

Chapter 6

Mathematical morphology

Morphology is the study of shape. *Mathematical morphology* mostly deals with the mathematical theory of describing shapes using sets. In image processing, mathematical morphology is used to investigate the interaction between an image and a certain chosen *structuring element* using the basic operations of *erosion* and *dilation*. Mathematical morphology stands somewhat apart from traditional linear image processing, since the basic operations of morphology are non-linear in nature, and thus make use of a totally different type of algebra than the linear algebra.

Although the mathematics behind this¹ is fascinating, treating it here would be beyond the scope of this book. In practical image processing, it is sufficient to know that morphology can be applied to a finite set P if

1. we can *partially order* its elements, (where the ordering is denoted by “ \leq ”), i.e., for all $a, b, c \in P$

$$a \leq a$$

$$(a \leq b, b \leq a) \Rightarrow a = b$$

$$(a \leq b, b \leq c) \Rightarrow a \leq c,$$

and

2. each non-empty subset of P has a maximum and minimum.

Example

Any finite set of real or integer numbers is a suitable set P

The ordering “ \leq ” is defined as in ordinary calculus ($3 \leq 4$, $4 \leq 18$, *etc.*). The maximum and minimum are also defined in the usual sense (e.g., $\max\{5, 3, 4\} = 5$).

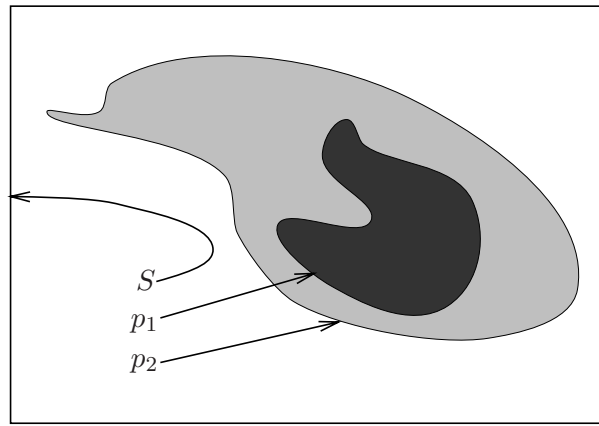
¹The mathematics of *lattices*.

This means we can apply morphology to grey-valued images (or subsets of images), because the collection of grey values can be viewed as a finite set P with ordering, maximum, and minimum well defined.

Example

The set of all subsets of a superset S is a suitable set P

The ordering “ \leq ” is defined by the subset relation “ \subset ”. The figure below shows an example, with $p_1 \subset p_2$, i.e., $p_1 \leq p_2$.



The maximum and minimum are defined by the union (\cup) and intersection (\cap) operators respectively. In this example, $\max\{p_1, p_2\} = \cup\{p_1, p_2\} = p_2$, and $\min\{p_1, p_2\} = \cap\{p_1, p_2\} = p_1$.

Besides finite sets of real and integer numbers and the set of subsets, more abstract sets P can be constructed. Mathematical morphology can be applied to any of these sets. In this chapter we are only concerned with the two sets mentioned in the examples above.

We can extend the partial ordering to digital images by applying the rules to the individual pixels. For instance, for two images f and g , the relation $f \leq g$ holds if:

$$f \leq g \iff \forall x : (f(x) \leq g(x)),$$

where “ $\forall x$ ” refers to all possible pixel locations. The “maximum image” and “minimum image” of two images can also be defined on a pixelwise basis:

$$\begin{aligned} (\max\{f, g\})(x) &= \max\{f(x), g(x)\} \\ (\min\{f, g\})(x) &= \min\{f(x), g(x)\}. \end{aligned}$$

The concepts of ordering, maximum, and minimum are fundamental in mathematical morphology. In fact, the maximum and minimum operator are the *only* basic operators used in mathematical morphology applied to images! And the only useful morphological operators are operators that somehow retain the order of images (*increasing* operators, see next section), or retain their maxima or minima.

6.1 Complement and operator properties

The *complement* X^c of a set X is defined as all elements not belonging to X . For an image f , the complement f^c is defined as f mirrored in a central grey-value line, as shown in figure 6.1. For an image with a range $\{L, \dots, M\}$ the complement can be written as

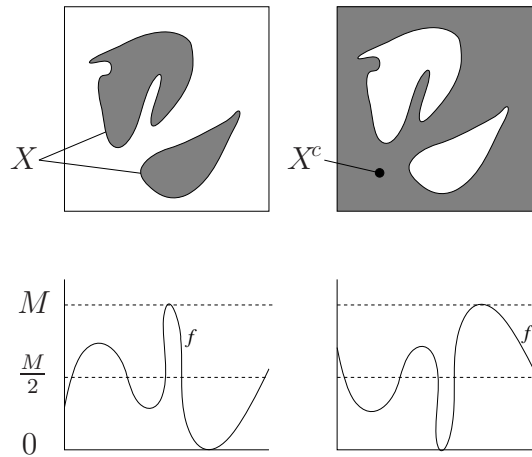


Figure 6.1 The complement of a set X (top row) and of an image or function f (bottom row).

$f^c(x, y) = L + M - f(x, y)$. Complementing twice results in the original set or function, *i.e.*, $(X^c)^c = X$, and $(f^c)^c = f$.

Example

If f equals

0	1	5
4	6	3
7	2	8

then the range is $\{0, \dots, 8\}$, and the complement f^c can be computed by $f^c(x, y) = 8 - f(x, y)$. So f^c equals

8	7	3
4	2	5
1	6	0

A fast way of verifying that the two images are complementary is to check that the sum of two corresponding pixels is always 8. Complementing the complement again will yield the original image.

Notice that the complement definitions for sets and images are consistent: if we represent the set X in figure 6.1 as a function (an image) f , *e.g.*, by

$$f(x, y) = \begin{cases} 1 & \text{if } (x, y) \in X \\ 0 & \text{elsewhere} \end{cases},$$

then applying the complement as defined for images will give the same result as applying the complement to the set X . Since we can convert a set to a binary image as in the formula above, we will only discuss images in the remainder of this section.

The operators φ_1 and φ_2 are called *dual* if applying φ_1 to the complement of f equals complementing the result of applying φ_2 to f , *i.e.*, if the following holds:

$$\forall f : \quad \left(\varphi_1(f^c) = (\varphi_2(f))^c \right).$$

Example

The \max and \min operators applied to images are dual.

For an example, suppose f equals

0	0	1
4	2	3
3	3	4

then f^c equals

4	4	3
0	2	1
1	1	0

We see that $\max(f^c) = 4$ and that $\min f = 0$. Complementing the latter expression gives us $(\min f)^c = 4$. So \max and \min are dual in this example:

$$\max(f^c) = (\min f)^c.$$

Many of the pairs of morphological operators presented in the next sections are dual, e.g., erosion and dilation, and opening and closing. Some operators are self-dual ($\varphi(f^c) = (\varphi(f))^c$). For example, the median filter from the previous chapter is self-dual.

Example

Given the images f and f^c :

0	0	1
4	2	3
3	3	4

4	4	3
0	2	1
1	1	0

Then the median values are 3 and 1 respectively. Since $(1) = (3)^c$ here, the median is self-dual:

$$\begin{aligned} 1 &= (3)^c \\ \text{median}(f^c) &= (\text{median}(f))^c \end{aligned}$$

An operator φ is called *increasing* if it does not alter the order of images:

$$\forall f, g : (f \leq g \implies \varphi(f) \leq \varphi(g)).$$

An operator φ is called *extensive* if

$$\forall f : (f \leq \varphi(f)),$$

i.e., the output is always “larger” than the input. See figure 6.2. If the reverse holds, then φ is *anti-extensive*:

$$\forall f : (\varphi(f) \leq f).$$

Increasingness and (anti-)extensivity are desired properties in a large number of operations. They are also a necessity in many theoretical considerations concerning morphological operators.

An operator is called *idempotent* if applying the operator more than once has no effect:

$$\forall f : (\varphi(\varphi(f)) = \varphi(f)).$$

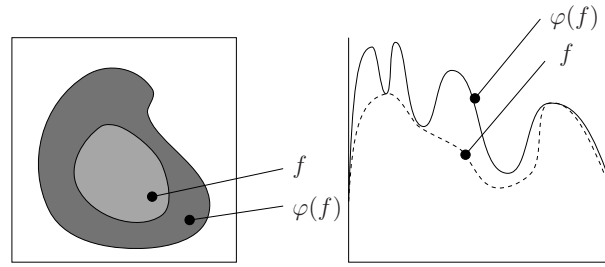


Figure 6.2 Example of an extensive operation φ applied to a set (a binary image) and a function (an image).

6.2 Relation between sets and images

Many of the morphological operators in the next section are only defined for sets. Because we want to use these operators on images, we need to define a relationship between sets and images, so we can transfer these set-operators to images. In the previous section we already defined the relation between a binary image and a set X : X is defined by all the pixels with grey value 1, the set $\{(x, y) | f(x, y) = 1\}$. This idea can be expanded so that we can define a relation between grey valued images and sets: we regard each specific grey level in the image as a set; we regard the image f as a stack of sets F with the relation

$$F(c) = \{(x, y) | f(x, y) \geq c\}.$$

This is demonstrated by figure 6.3.

We are now able to “translate” any operators defined only for sets to apply to images: we apply the operator to all of the level sets $F(c)$ that together make up the image. This seems tedious, but in practice the resulting operations can often be simplified a great deal.

6.3 Erosion and dilation

Morphological operations describe the interaction of an image with a *structuring element* S . The structuring element is usually small relative to the image. In the case of digital images, we typically use simple binary structuring elements like a cross or a square, such as the ones in figure 6.4².

²Note: whenever we display *binary* images in this book we reverse the normal convention “bright pixel = high grey value, and dark pixel = low grey value”. In the case of binary images we use the convention that *object* pixels have grey value 1 and are displayed *black*, while *background* pixels have grey value 0 and are displayed *white*. Although this may be somewhat confusing in the beginning, it usually enhances the clarity of binary images.

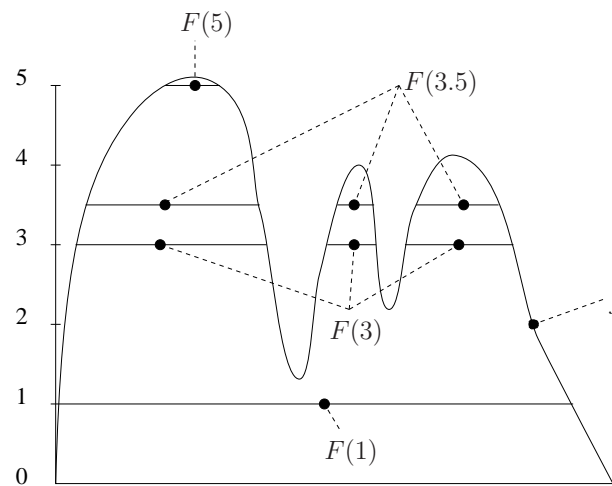


Figure 6.3 An image or function f can be viewed as a stack of sets $F(c)$. The level sets $F(1)$, $F(3)$, $F(3.5)$, and $F(5)$ are shown. See text for details.

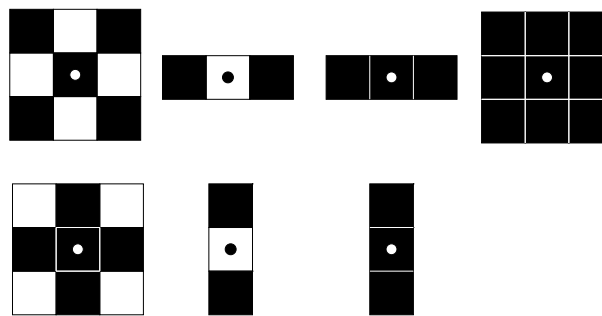


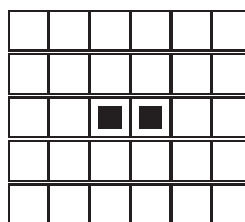
Figure 6.4 Example of simple binary structuring elements. Black pixels are part of the element, white ones are not. The pixel with the circle marks the origin.

The *dilation*³ $\delta(X)$ of a set (binary image) X by the structuring element S is defined by

$$\delta(X) = \{x + s \mid x \in X \wedge s \in S\}.$$

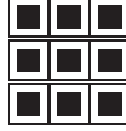
Example

Suppose the set X is represented by the marked pixels in this binary image:



³Also called Minkowski addition.

i.e., the set $X = \{(2, 2), (3, 2)\}$. Suppose S is a square structuring element S with the origin at its center:



so $S = \{(-1, -1), (0, -1), (1, -1), \dots, (1, 1)\}$. The dilation δ of X by S equals $\delta(X) = \{x+s | x \in X \wedge s \in S\}$, i.e., the set of all possible additions of an element of X and an element of S . For example: $(2, 2) + (-1, -1) = (1, 1)$ or $(3, 2) + (0, 1) = (3, 3)$. This results in the following set:

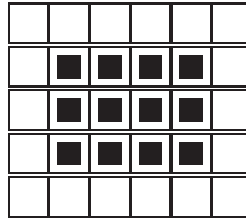


Figure 6.5 shows another example of a dilation of a set. The dilation by a *symmetrical*⁴

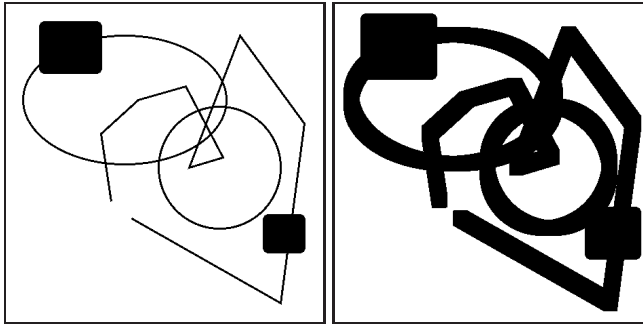


Figure 6.5 Example of the dilation (right) of a set (binary image, left) by a square structuring element (shown on the far right).

structuring element is described more intuitively by:

- Place the structuring element anywhere in the image
- Does it hit the set? Then the origin of the structuring element is part of the dilated set.

⁴Symmetrical in the origin.

If the structuring element S is not symmetrical we must use its *transpose* \check{S} in the above procedure. The transpose is the structuring element mirrored in the origin:

$$\check{S} = \{s | (-s) \in S\}.$$

The dual operation of dilation is called *erosion*⁵, and the erosion $\varepsilon(X)$ of a set X by a structuring element S is defined by

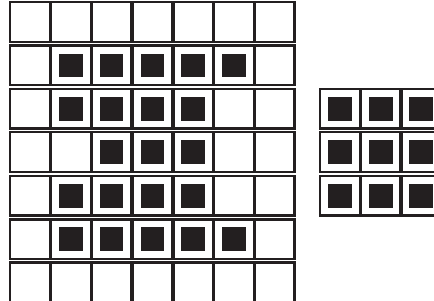
$$\varepsilon(X) = \{x | \forall s \in S, x + s \in X\}.$$

More intuitively, it is described by:

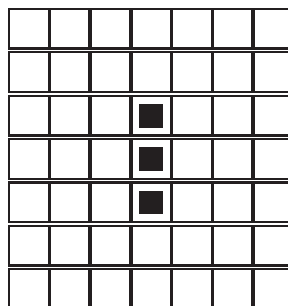
- Place the structuring element anywhere in the image
- Is it fully contained by the set (i.e., a subset)? Then the origin of the structuring element is part of the eroded set.

Example

Given the set X and the structuring element S with the origin at its center:



then the erosion $\varepsilon(X)$ equals:



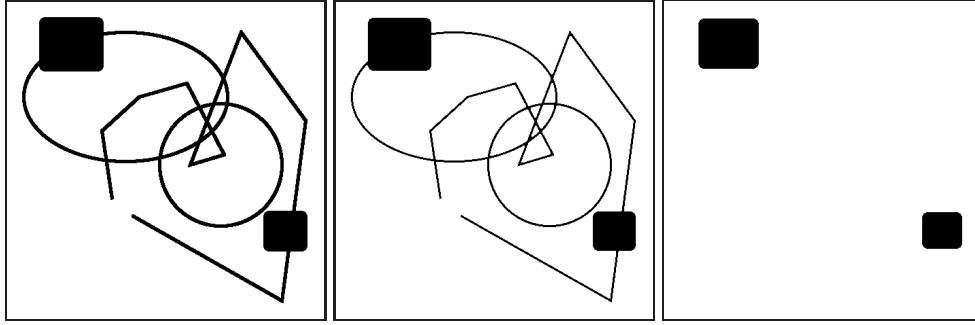


Figure 6.6 Example of erosion of a set (binary image) with a square structuring element. On the left the original set is shown. In the middle is the result after one erosion, and to the right the result after another erosion with the same structuring element.

Figure 6.6 shows another example of an erosion of a set.

Using the relation between sets and images described in the previous section we can also give formulas for erosion ε and dilation δ of a digital image f with a structuring element S :

$$(\varepsilon(f))(x) = \min_{s \in S} f(x + s)$$

$$(\delta^*(f))(x) = \max_{s \in S} f(x - s).$$

Where x and s are vector quantities if f is an image. The minus sign in the definition of the dilation is counter-intuitive in most practical use. To avoid this, we will often use the alternative definition

$$(\delta(f))(x) = \max_{s \in S} f(x + s).$$

Although it has taken a number of theoretical steps to arrive at the above formulas for dilation δ and erosion ε , their application to digital images is straightforward, as the next example shows.

Example

Given the image f and a 3 pixel wide cross for a structuring element S :

1	1	0	0	1	2
2	2	1	0	0	1
3	3	3	1	0	1
3	3	1	0	1	2
3	1	0	1	2	3
1	0	1	2	3	4

	■	
■	■	■
	■	

⁵Also called Minkowski subtraction.

we can compute the erosion ε of f at each pixel (x, y) by centering S at (x, y) , and then taking the minimum of all pixels of f that are ‘hit’ by S . For example, the erosion at $(2, 2)$ equals the minimum of the values $\{1, 3, 3, 1, 1\}$, *i.e.*, 1.

The dilation is computed in the same way, except that the maximum is now taken. The erosion $\varepsilon(f)$ and dilation $\delta(f)$ are respectively:

1	0	0	0	0	1
1	1	0	0	0	0
2	2	1	0	0	0
3	1	0	0	0	1
1	0	0	0	1	2
0	0	0	1	2	3

2	2	1	1	2	2
3	3	3	1	1	2
3	3	3	3	1	2
3	3	3	1	2	3
3	3	1	2	3	4
3	1	2	3	4	4

where we used padding with values of $-\infty$ and ∞ in the cases of dilation and erosion respectively.

The figures 6.7 and 6.9 show examples of grey valued dilation and erosion applied to real images. Figure 6.8 clearly shows that the dilation is a maximum of a neighborhood defined by the structuring element.

Intermezzo

Different definitions of erosion, dilation, and other morphological operations may be found in other books. For example, the dilation is usually defined with the minus sign rather than the plus sign. This has interesting theoretical properties, like making operator duality straightforward, but has drawbacks like muddling the intuitive nature of the operation.

When comparing morphological operations from different texts, verify if their definitions match, or –if not– what the differences are.

Differences in definitions usually boil down to transposition (mirroring) of either the structuring element or the image. We have seen this problem before at the definition of discrete convolution: the formal definition $(*)$ has nice arithmetic and other theoretical properties, but the “working” definition (correlation; \star) is easier to understand in practical applications.

Note that all differences in definitions vanish if we use structuring elements or kernels that are *symmetrical* in the origin. Since the structuring element then equals its own transpose, there is no difference in the result, whatever the formal definition of an operation is. For example, the \star and $*$ operations have the same result if we use a symmetrical kernel.

In this book we will make use of the practical definitions, and avoid unintuitive transpositions as much as possible.

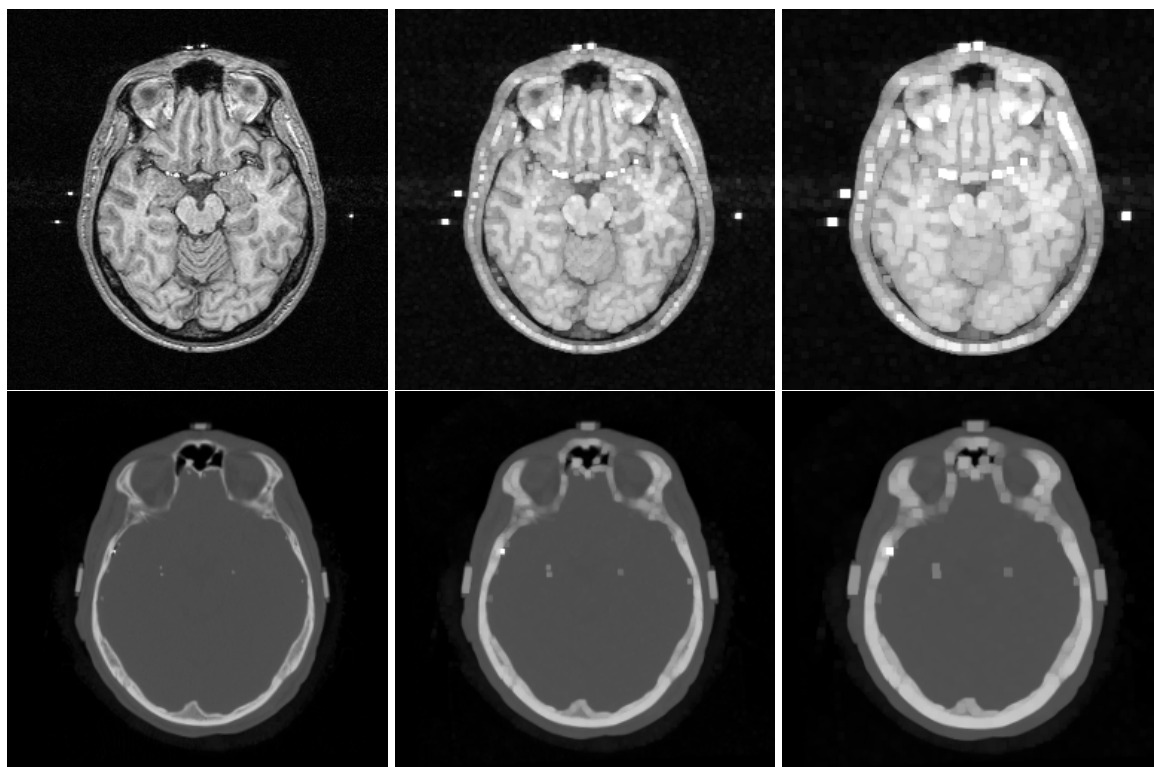


Figure 6.7 Example of dilation by a 3×3 square structuring element (middle column) and a 5×5 square structuring element (right column), applied to a 256×256 MR image (top row) and a 256×256 CT image (bottom row).

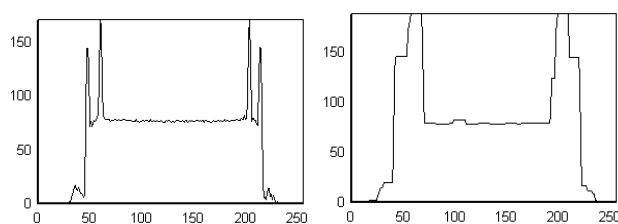


Figure 6.8 The left graph shows a plot of grey values along a horizontal line obtained from the CT image in the previous figure. The right graph shows the same plot after dilation; each local grey value at coordinate x corresponds to the maximum grey value in a small neighborhood around x in the original plot.

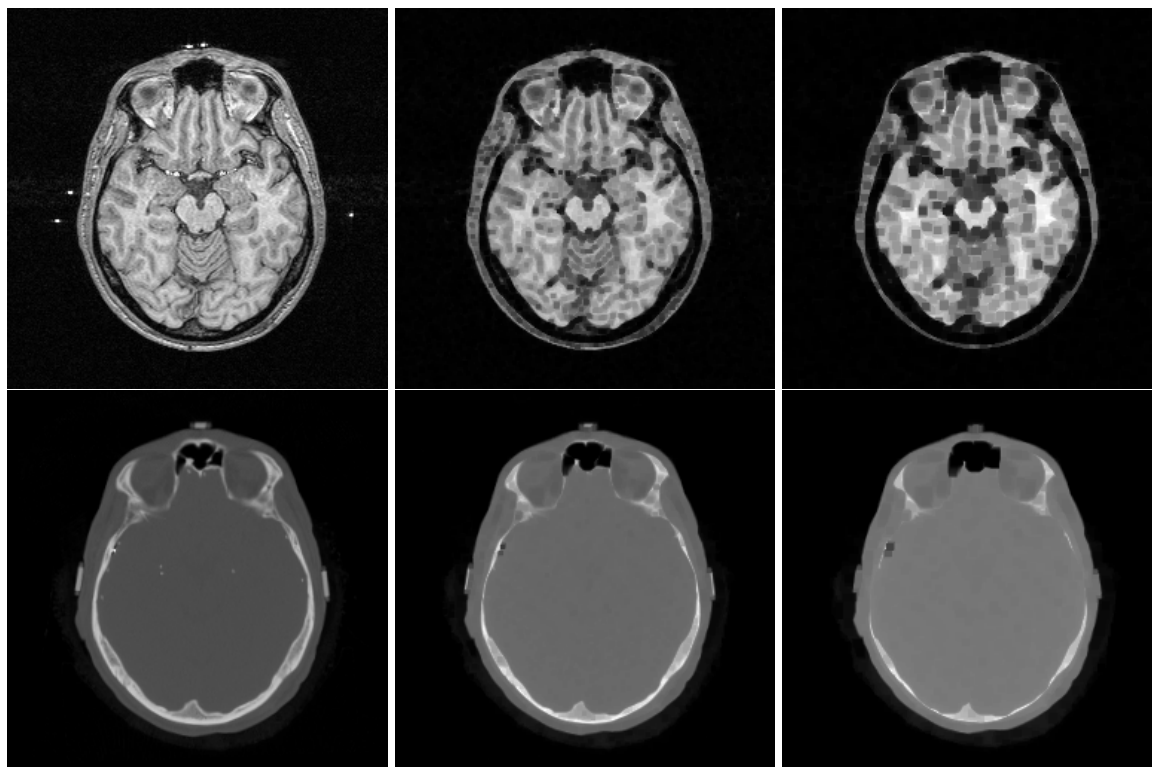


Figure 6.9 Example of erosion by a 3×3 square structuring element (middle column) and a 5×5 square structuring element (right column), applied to a 256×256 MR image (top row) and a 256×256 CT image (bottom row).

6.3.1 Properties of erosion and dilation

When examining the figures showing erosion and dilation, it will be obvious that the names *erosion* and *dilation* describe the effect of the operations well. The erosion operation “eats” chunks away at the boundaries of objects, while the dilation operation does the opposite. It is however noteworthy that –owing to the non-linear character of the operations– erosion is *not* the inverse of dilation. The inverse of either operation does not exist. For instance: many different input images can have the same erosion, so this operation cannot be reversed.

Many special properties hold for erosion and dilation. Assuming a *symmetrical* structuring element for simplicity:

Duality: erosion and dilation are dual operations:

$$(\varepsilon(f))^c = \delta(f^c).$$

Increasingness: Both the erosion and dilation are increasing operations:

$$f \leq g \Rightarrow \begin{cases} \varepsilon(f) \leq \varepsilon(g) \\ \delta(f) \leq \delta(g) \end{cases}$$

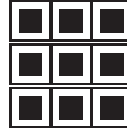
Extensivity: If the origin is part of the structuring element, the dilation is extensive, and the erosion is anti-extensive:

$$0 \in S \Rightarrow \begin{cases} \varepsilon(f) \leq f \\ f \leq \delta(f) \end{cases}$$

Separability: The symmetrical structuring element can be separated in one-dimensional parts. The erosion or dilation can be carried out using one-dimensional erosions and dilations:

$$S = \delta_{S_1}(S_2) \Rightarrow \begin{cases} \varepsilon_S(f) = \varepsilon_{S_1}(\varepsilon_{S_2}(f)) \\ \delta_S(f) = \delta_{S_1}(\delta_{S_2}(f)) \end{cases}$$

For example: dilation by



is equivalent to dilation by



followed by dilation with



6.4 Opening and closing

The composition of the erosion and dilation operations has interesting properties. The morphological *opening* γ and *closing* φ are defined by:

Opening: $\gamma(f) = \delta(\varepsilon(f))$

Closing: $\varphi(f) = \varepsilon(\delta(f))$.

To understand what *e.g.*, a closing operation does: imagine the closing applied to a set; the dilation will expand object boundaries, which will be partly undone by the following erosion. Small, (*i.e.*, smaller than the structuring element) holes and thin tube-like structures in the interior or at the boundaries of objects will be filled up by the dilation, and not reconstructed by the erosion, inasmuch as these structures no longer have a boundary for the erosion to act upon. In this sense the term 'closing' is a well chosen one, as the operation removes holes and thin cavities. In the same sense the opening opens up holes that are near (with respect to the size of the structuring element) a boundary, and removes small object protuberances. Examples of opening and closing can be seen in the example below and the figures 6.10 and 6.11.

Example

Consider the image f :

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

then the erosion $\varepsilon(f)$ and the dilation $\delta(f)$ by a cross-shaped structuring element



using padding with $+\infty$ and $-\infty$ as before, are

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

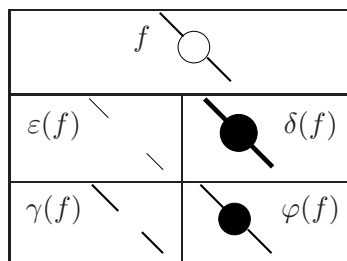
1	1	1	0	0	0	0
1	1	1	1	1	0	0
1	1	1	1	1	1	0
1	1	1	1	1	1	0
1	1	1	1	1	1	1
1	1	1	1	1	1	1
0	1	1	1	1	1	1
0	0	0	0	1	1	1

The opening $\gamma(f) = \delta\varepsilon(f)$ and closing $\varphi(f) = \varepsilon\delta(f)$ are

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

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

These abstractions of the images show that the names *erosion*, *dilation*, *opening*, and *closing* actually do what their names suggest:



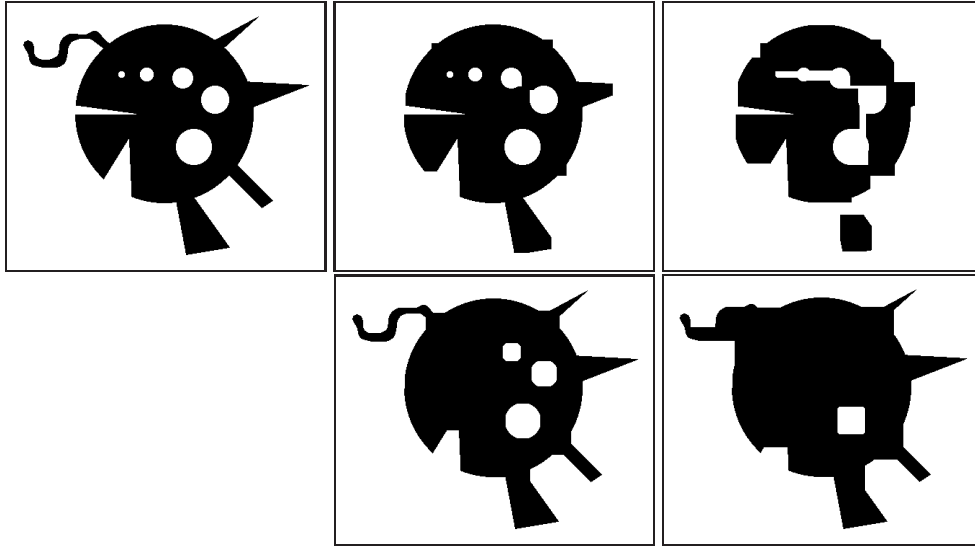


Figure 6.10 Example of opening and closing on a set (binary image). The top row shows the original, and two openings with a square structuring element of increasing size. The bottom row shows the corresponding closings.

6.4.1 Properties of opening and closing

Assuming a symmetrical structuring element:

Duality: Opening and closing are dual operations.

Increasing: Since the opening and closing are compositions of increasing operations (erosion and dilation), the opening and closing are also increasing.

Extensive: The opening is anti-extensive, the closing is extensive.

Idempotent: Both the opening and closing are idempotent operations, *i.e.*, applying them to an image twice gives us the same result as applying them only once:

$$\gamma\gamma(f) = \gamma(f)$$

$$\varphi\varphi(f) = \varphi(f)$$

6.5 Geodesic operations and reconstruction

Dilation is an extensive operation; structures “grow” at their boundaries. It is often useful to constrain this growth in a way such that structures do not grow outside of some pre-defined boundaries. A way to achieve this is to use *geodesic dilation* $\delta_g(f)$, which is defined as the minimum of the ordinary dilation $\delta(f)$ and a control image g that contains the restraining boundaries:

$$\delta_g(f) = \min(\delta(f), g).$$

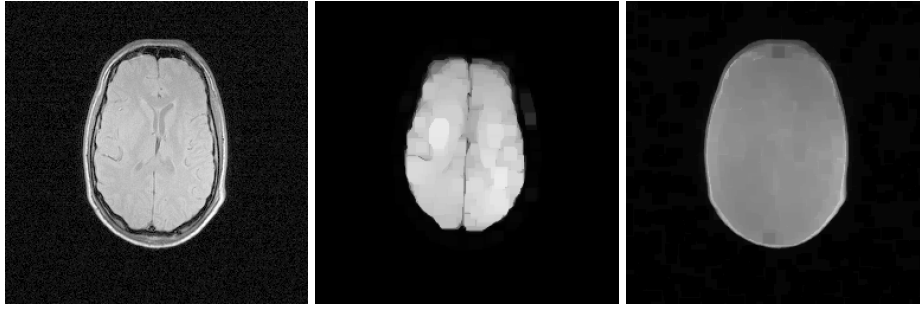


Figure 6.11 Example of opening (middle) and closing (right) by a square structuring element on a grey valued MR image (left).

Its counterpart, geodesic erosion, can be defined in a similar manner by

$$\varepsilon_g(f) = \max(\varepsilon(f), g).$$

Example

Given f and g by

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

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

Then $\delta(f)$ and $\delta_g(f)$ using a 5×5 square structuring element are

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

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

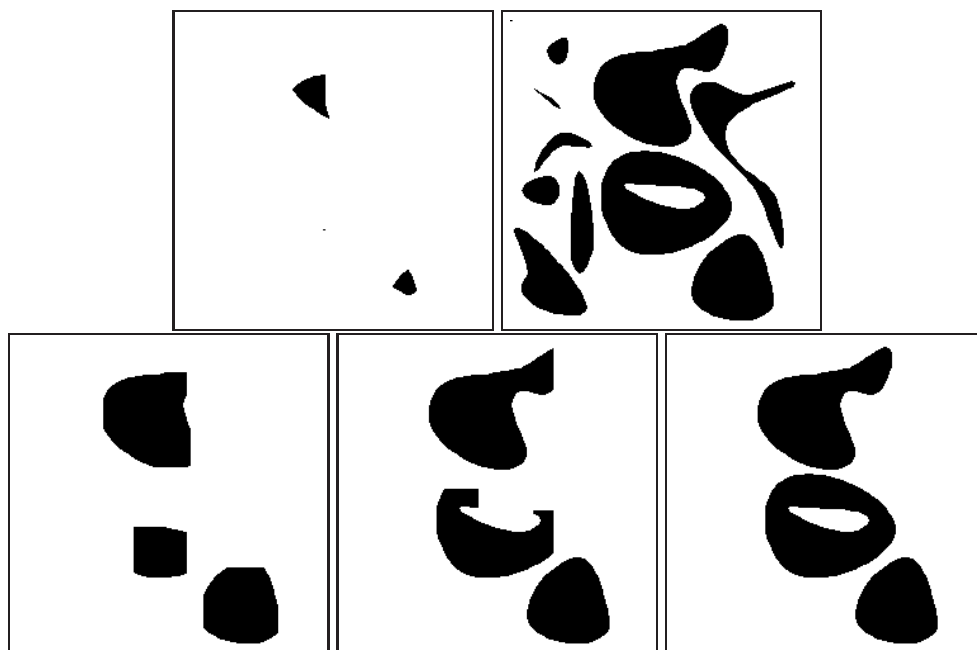
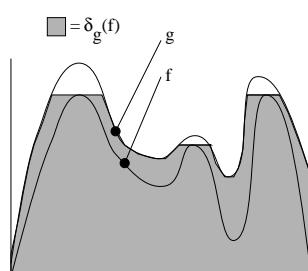


Figure 6.12 Example of geodesic dilation of a binary image. The top row shows the original image f and a control image g . The bottom row shows three geodesic dilations with a structuring element of increasing size.

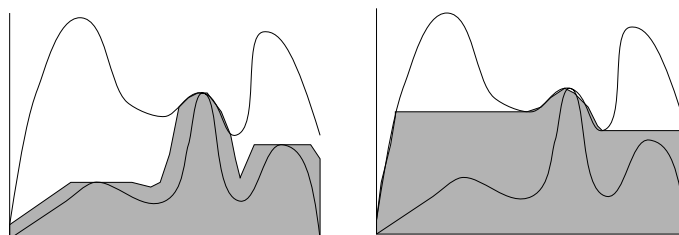
A more visual example is shown in figure 6.12.

Example



The above figure shows a one-dimensional example of a geodesic dilation $\delta_g(f)$. The boundaries of g stop f from “growing” outside of the region defined by g .

The below figure shows a geodesic dilation with a different choice of f . The left image shows one geodesic dilation step, the right image shows the result after a large number of geodesic dilations.



6.5.1 Opening by reconstruction of erosion

It can be shown that iterated application of geodesic dilation or erosion to an image will converge to a stable result. If we iterate the operations until stability occurs, this is called *reconstruction by dilation* and *reconstruction by erosion* respectively. As we know, the composition of erosion and dilation is called an opening. The special composition of erosion and reconstruction by dilation is called *opening by reconstruction of erosion* or just an *opening by reconstruction*.

Opening by reconstruction is a very effective way of removing small structures from (especially binary) images: the erosion step removes small structures from an original image. It will also erode the boundaries of larger structures, but parts of these structures remain present in the eroded image. If we now do a reconstruction by geodesic dilation *with the original image for a control image*, these parts will grow back to their original size. The small structures that were completely removed by the erosion will not grow back. Effectively, the opening by reconstruction has removed small structures, while keeping the larger structures completely intact. The figures 6.13 and 6.14 show examples of this.

The size of the structuring element in the erosion step determines what structures are removed by the opening by reconstruction. The larger we choose this element, the more structures will be removed. In figure 6.13, we can note that it is also possible to remove the *large* structures from an image rather than the small ones: this can be achieved by taking the difference image of the left and right images respectively.

6.5.2 Other openings by reconstruction

Reconstructions can be made of erosions of images, as in the previous section. Another common procedure is to reconstruct an image from an image that contains only a set of markers.

Example_____

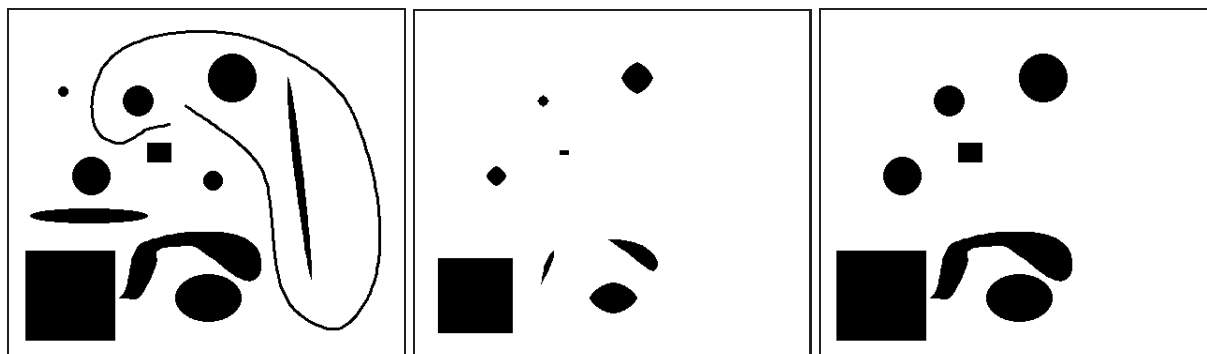
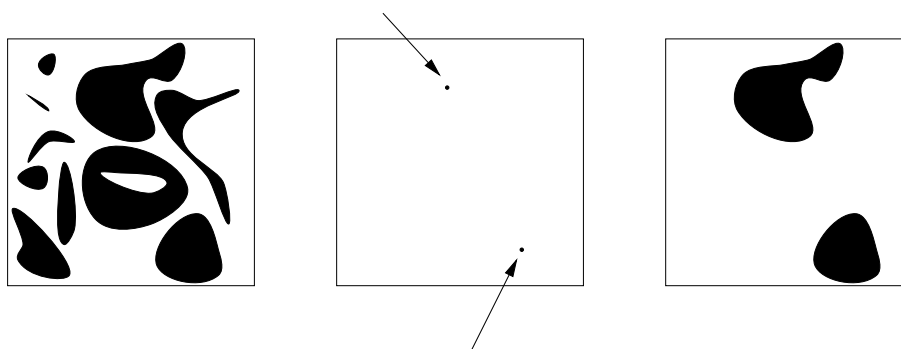


Figure 6.13 Example of opening by reconstruction on a binary image. On the left the original image is shown, in the middle the image after erosion, and on the right after opening by reconstruction using the original image for a control image. The larger structures from the original image that are still “marked” in the eroded image have been completely reconstructed. The smaller structures that were completely removed by the erosion remain absent in the reconstructed image.



Suppose we have an original image (left), and obtained a marker image (middle). Then the opening by reconstruction of the marker image using the original image for a control image is the image on the right. Only the “marked” objects have been reconstructed.

The marker image can *e.g.*, be obtained by “user clicks” in a computer application, or can be the result of some image processing routine run on the original image.

Example

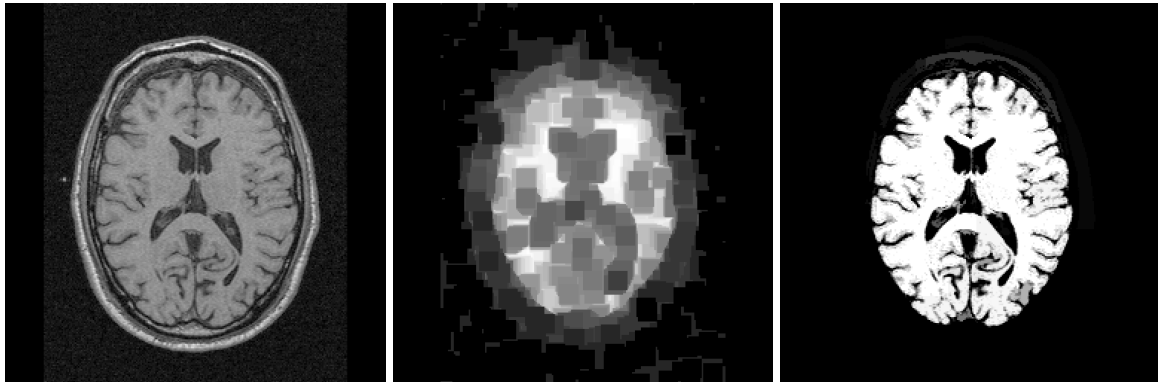
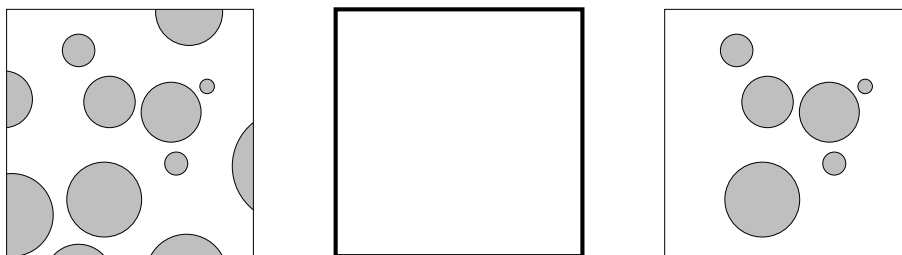


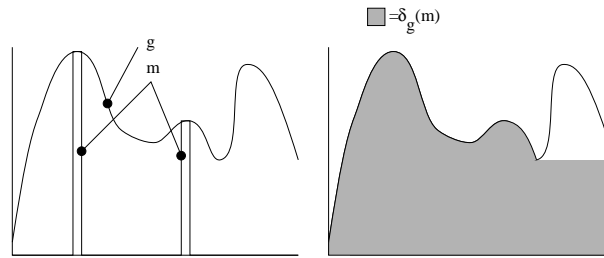
Figure 6.14 Example of opening by reconstruction of an MR image. To the left the original image is shown, in the middle the image after erosion, and on the right after opening by reconstruction –with the original image for a control image– and thresholding.



An application of reconstruction from a marker image can be the removal of objects that intersect the image boundary in a binary image (left image). If we choose the image boundary to be the marker image (middle image), then the reconstruction of the original image from this marker image will contain only those objects that intersect the boundary. If we subtract this reconstructed image from the original, we retain only those objects that do *not* intersect the boundary, as seen in the right image.

Example

Another example of an application of reconstruction from markers is the removal of unwanted maxima (and their neighborhood) from images, as shown in the image below.



The left image shows the original image and the marker image: two out of the three image maxima are marked. The right image shows the reconstruction from the marker image using the original image for a control image. The unmarked maximum and its neighborhood have vanished.

6.6 Residues

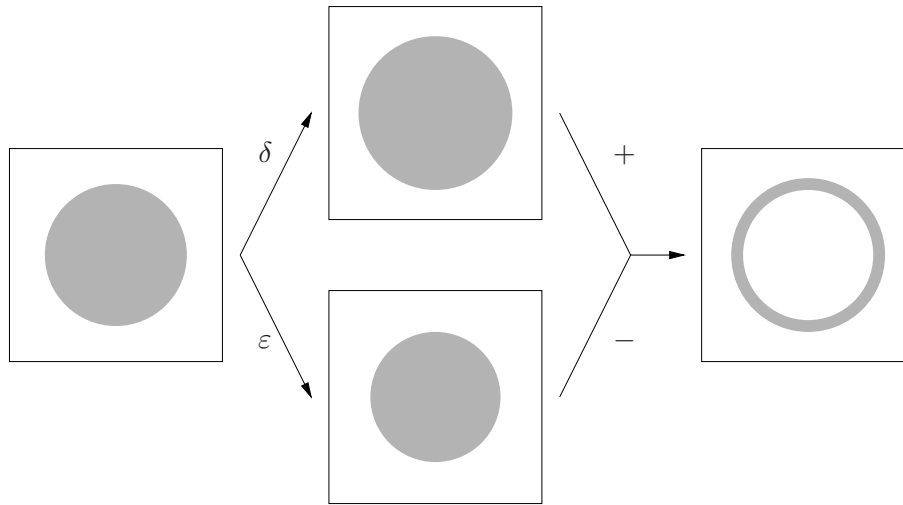
Many interesting morphological filters can be formed using *residues*, *i.e.*, the *differences* of two or more common operations. Examples of residue operations on an image f are $\delta(f) - \varepsilon(f)$ or $f - \gamma(f)$.

We can distinguish three types of residue, based on what type of operations are used in the difference

1. Using a difference of two primitives, *e.g.*,
 - Morphological gradient
 - Morphological Laplacian
 - Top hat filter
2. Using differences of two families of primitives, *e.g.*,
 - Skeleton
 - Ultimate erosion
3. Using the hit-or-miss transformation (see section 6.6.4) in the difference, *e.g.*,
 - Thinning and thickening

6.6.1 Morphological gradient filters

We have seen that the erosion and dilation operations act only on the edges of objects: the erosion “eats” chunks away, while the dilation does the opposite; making objects “grow” at their edges. Using these notions, we can detect edges by examining the difference between an original image and its erosion and dilation.

Example


By taking the difference of the dilation and erosion of an image, we can detect edges of objects in the original image.

The *morphological gradient* $g(f)$ of an image f is defined by

$$g(f) = \delta(f) - \varepsilon(f).$$

This “thick” gradient, that sticks out on two sides of the actual edges, can be decomposed into two “half” gradients: $g(f) = g^+(f) + g^-(f)$ with

$$\begin{aligned} g^-(f) &= f - \varepsilon(f) \\ g^+(f) &= \delta(f) - f, \end{aligned}$$

where the “inner” gradient $g^-(f)$ adheres to the inside⁶ of objects, and the “outer” gradient $g^+(f)$ adheres to the outside of objects. Examples can be seen in the figures 6.15 and 6.16.

As there is a morphological equivalent of the (length of the) gradient, there is also an equivalent of the Laplacian: the morphological Laplacian $\Delta(f)$ is defined as the residue of the outer and inner gradient:

$$\Delta(f) = g^+(f) - g^-(f).$$

An example can be seen in figure 6.17.

⁶i.e., the bright side of an edge in a grey valued image.

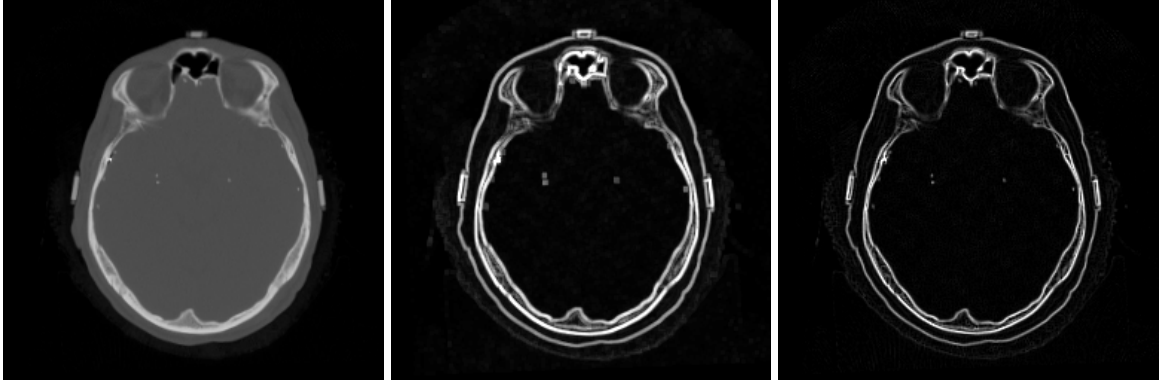


Figure 6.15 Example of the morphological gradient (middle image) and the inner gradient (right image) of a CT image (left image) with a 256×256 resolution using a 3×3 square structuring element.

6.6.2 Top hat filters

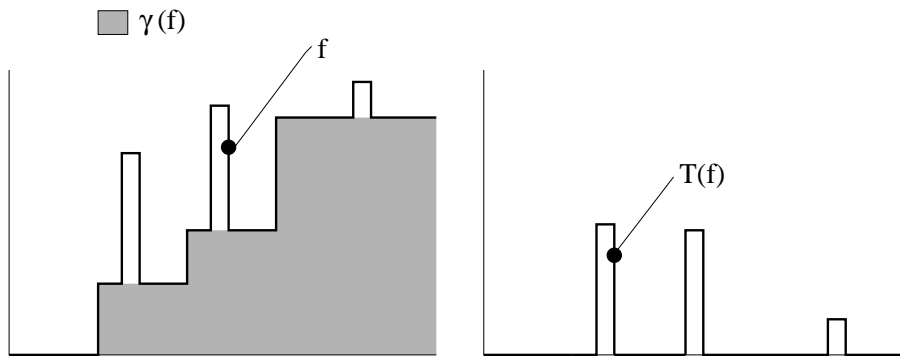
In previous sections, we have seen that an opening can be used to remove structures smaller than a certain size⁷ from an image, while not –or rather, as little as possible– altering larger structures. The dual operation, the closing, can be used to close up holes and cavities that are smaller than a certain size.

If an opening removes small structures, then the difference of the original image and the opened image should bring them out. This is exactly what the *white top hat* $T(f)$ filter does, which is defined as the residue of the original and opening:

$$T(f) = f - \gamma(f).$$

Example

The figure below shows an image f and an opening $\gamma(f)$. The opening removes small structures in f . The white top hat $T(f) = f - \gamma(f)$ extracts just these structures.



⁷Where “size” (and shape) is relative to the structuring element used.

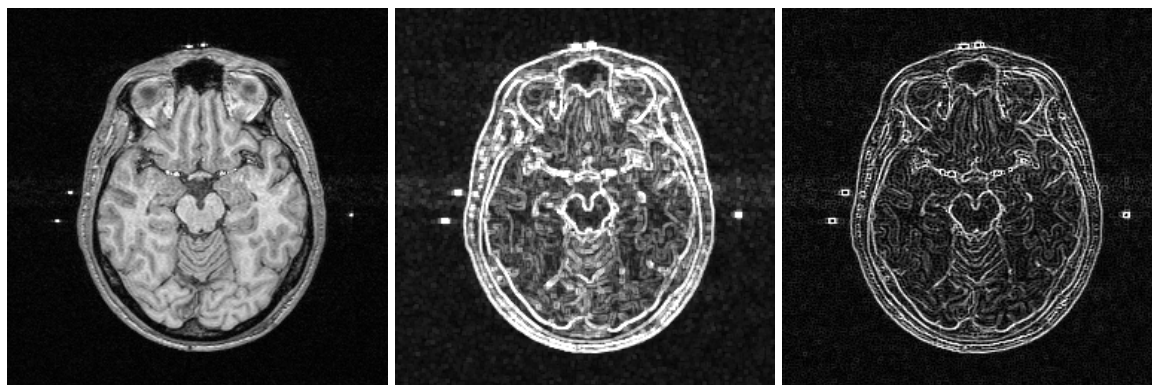


Figure 6.16 Example of the morphological gradient (middle image) and the outer gradient (right image) of an MR image (left image) with a 256×256 resolution using a 3×3 square structuring element.

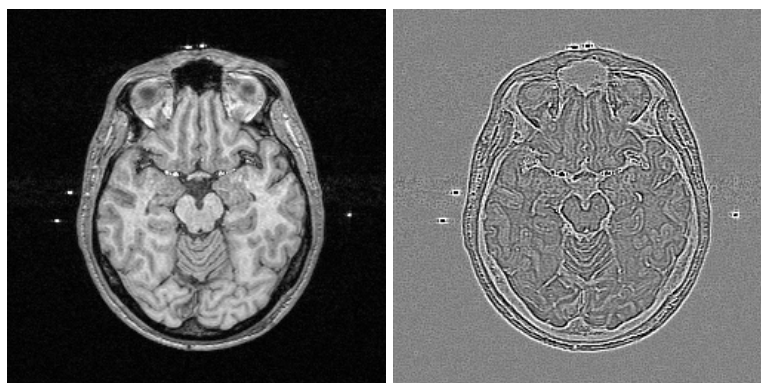


Figure 6.17 Example of the morphological Laplacian (right) of an MR image (left). The computations were done using a 3×3 square structuring element.

The above example shows another aspect of the white top hat: not only does it show small structures, but it shows them with a grey value that is *relative to the local background*: the grey value of the extracted small structures is relative to the local grey value in the neighborhood in the original image. This observation gives rise to another application of the white top hat: if we use a large enough structuring element, we effectively extract *all* structures in the original image, but with a grey value relative to the local background grey value. In this way, we can remove slow grey value variations in the background of an image, e.g., an illumination gradient in a photograph. Figure 6.18 shows an example of this use of the white top hat.

The counterpart of the white top hat is the *black top hat filter* $T^*(f)$ which is defined by

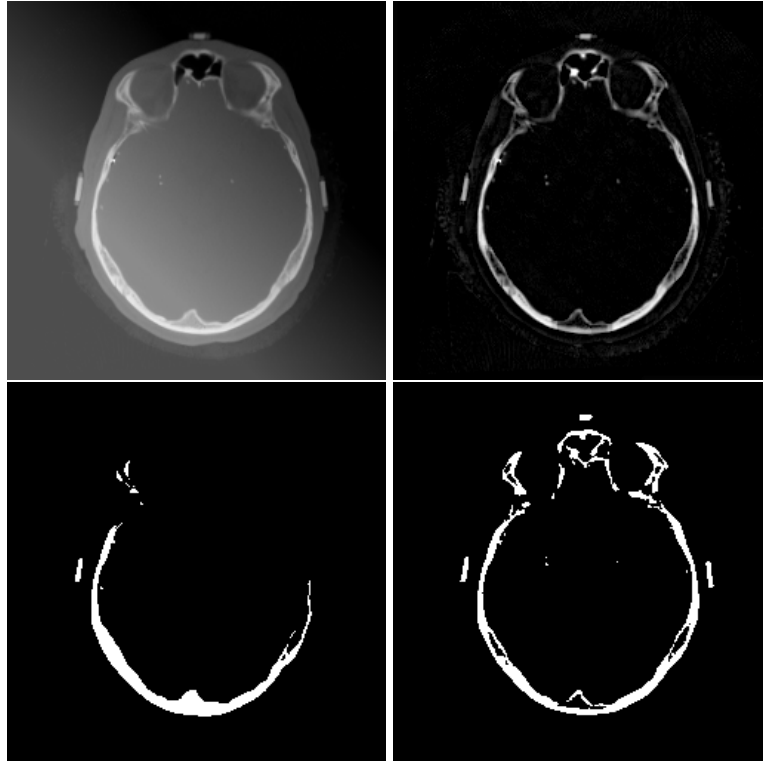


Figure 6.18 The top left image shows a 256×256 CT image with an added background grey value variation. As the image on the bottom left shows, we can no longer use thresholding to extract the bone from this image, because of the variation in grey values in the bony structures. The top right image shows the CT image with added variation after applying a white top hat filter using a square structuring element of 7×7 , a size which should be large enough to extract all the bony structures. As the bottom right image shows, thresholding *this* image gives us the desired result.

the residue of closing and the original:

$$T^*(f) = \varphi(f) - f.$$

As the white top hat extracts small “white” (larger grey value than the background) structures, the black top hat extracts small dark structures, *i.e.*, holes and cavities. Both the white and black top hat filters are idempotent.

Intermezzo

Many variations on the top hat filters exist. In most cases, the standard opening (or closing for the black top hat) is replaced by a more elaborate opening (closing),

e.g., an opening by reconstruction. Another example is the following variation of a white top hat:

$$T_{\text{var}}(f) = f - \min(\gamma(\varphi(f)), f).$$

A variation which is less sensitive to noise than the standard top hat. The standard top hat –since it extracts all structures small relative to the structuring element– also extracts noise from the image.

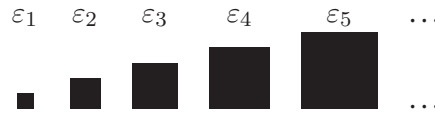
6.6.3 The skeleton and ultimate erosion

The skeleton and ultimate erosion are concepts that are most intuitive in the context of sets and binary images. Although there are definitions that apply to grey valued images, we will only discuss binary versions of the skeleton and ultimate erosion here.

In the previous sections, we have discussed filters that are the residue of two primitive operations. In this section, we will discuss residues of *families* of primitives. A *family* $\{\psi_i\}$ is a set of morphological operators with a structuring element that depends on the parameter i .

Example

A family of erosions $\{\varepsilon_i\}$ by square structuring elements is for instance the following family of erosions by the structuring elements:

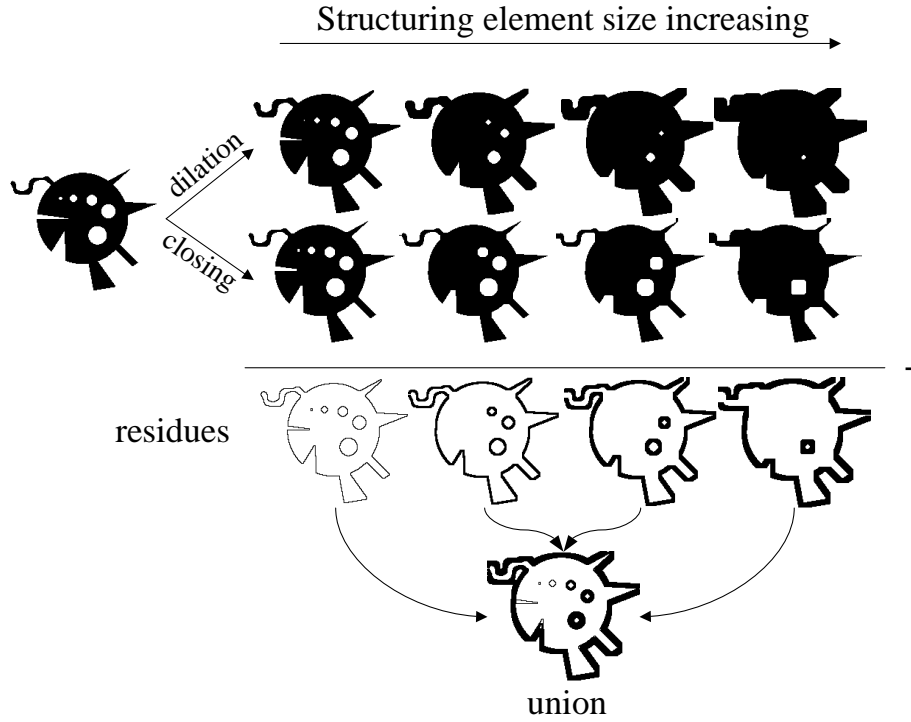


For a set (or binary image) S , the residue R of two families of primitives $\{\psi_i\}$ and $\{\varphi_i\}$ is defined as the union of the residues for each value of i :

$$R_{\{\psi_i\}, \{\varphi_i\}}(S) = \bigcup_i (\psi_i(S) - \varphi_i(S)).$$

Example

The graphical equivalent of the residue of a family of dilations and a family of closings (for four values of i) is shown below.



Intermezzo

Formally, a better definition of the residue of two families (purely in terms of sets) would be

$$R_{\{\psi_i\}, \{\varphi_i\}}(S) = \bigcup_i \left(\psi_i(S) \setminus \varphi_i(S) \right),$$

where \setminus is the “set minus” symbol, so $\psi_i(S) \setminus \varphi_i(S)$ means the set $\psi_i(S)$ *without* the set $\varphi_i(S)$. This definition avoids negative values that may occur in our definition.

We can easily adapt our definition to this set definition by setting all negative values to zero. In practice, the families $\{\psi_i\}$ and $\{\varphi_i\}$ are often chosen such that no negative values can occur.

6.6.3.1 Ultimate erosion

The *ultimate erosion* U of a set is defined as a series of erosions that leaves a single “marker” for each component of the set untouched. That is, each component is eroded

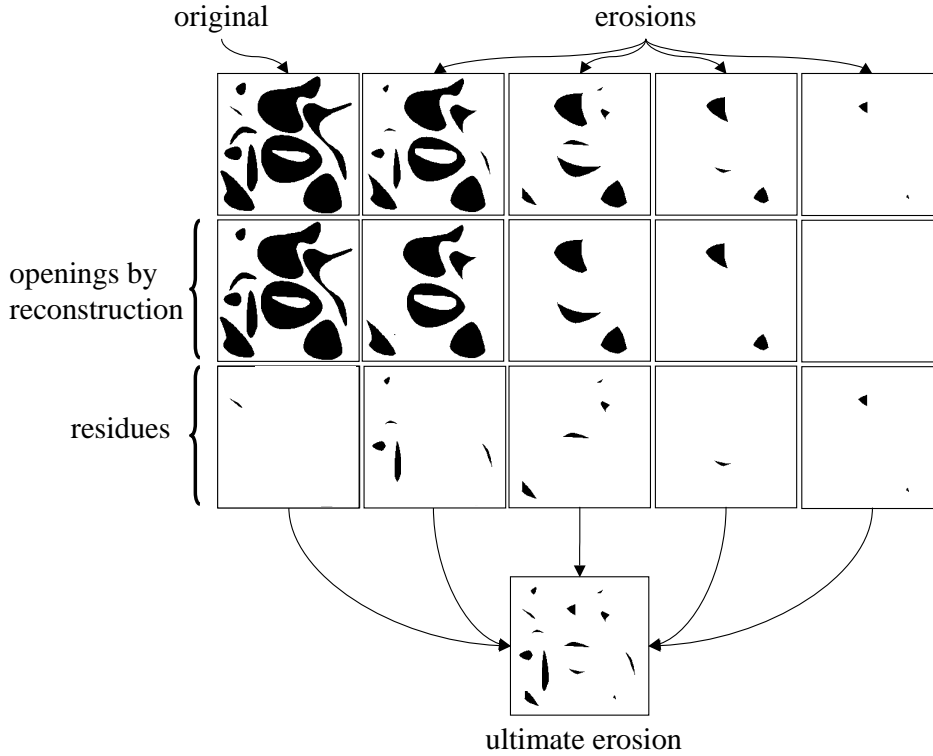
until one more erosion would remove the component entirely. The part of the component remaining before this final erosion step is retained in the ultimate erosion. We can compute the ultimate erosion by using the families of erosion and opening by reconstruction:

$$U(S) = R_{\{\varepsilon_i\}, \{\gamma_{\text{rec}, i}\}}(S),$$

where $\gamma_{\text{rec}, i}$ is an opening by reconstruction using $\varepsilon_{i+1}(S)$ for the erosion and ε_i for the control image.

Example

How does this formula for $U(S)$ work? Take a look at this figure:



The erosions will remove structures, but with the ultimate erosion we are not allowed to remove an object entirely. So we must include an eroded object in the ultimate erosion set just *before* it will be removed entirely in the next erosion step. So we must know if an object will be removed in the next step erosion step. We can do this using an opening by reconstruction (which was an erosion followed by a reconstruction using geodesic dilation). In the opening by reconstruction, objects that disappear in the next erosion step will not be reconstructed. We can find

these objects by taking the difference (residue) of the current image before and after opening by reconstruction. This residue contains exactly those remnants of objects that will disappear in the next erosion step, so this residue is part of the ultimate erosion.

Note that the ultimate erosion is not perfect in the sense that it always leaves exactly one marker for each object in the original image. In the example above, thirteen markers are placed for ten original objects. More than one marker per object appears if the original object contains relatively thin bridges between outer parts. If a bridge 'breaks' in the erosion process, this may result in two or more markers associated with the original object. It is possible to adapt the definition of ultimate erosion to force only one marker per object, but in practice the above definition is more commonly used, because it assigns multiple markers to objects that –although connected in the image– are in fact overlapping separate objects in reality.

6.6.3.2 The skeleton

The *skeleton* of a set is a stick figure representing the basic shape of the set. It is best explained using the “grass fire” analogy: imagine the objects in your set to be patches of grass. Now we set fire to all the grass at the boundaries of the objects, and assume the fire to burn with constant speed until all of the grass is gone. The skeleton of the set is formed by all points where the wave fronts of the fire meets. Figure 6.19 shows some examples of grass fire skeletons.

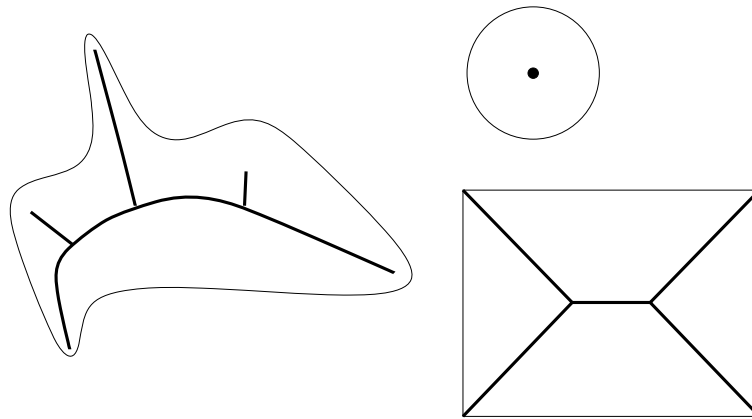


Figure 6.19 Some sets and their corresponding grass fire skeletons.

In the example on ultimate erosion (section 6.6.3.1 above), the ultimate erosion $U(S)$ is the union of five component residues, $U(S) = \cup\{U_0, U_1, \dots, U_4\}$, with $U_i = \varepsilon_i(S) -$

$\gamma_{\text{rec},i}(S)$. An interesting observation is that the dilation of U_i with δ_i results in a so-called *maximal ball*. The maximal ball is a set that fits completely inside the original set S . It is maximal in the sense that dilation with a larger element than δ_i results in a set that no longer fits inside S .

Formally, we define as the *ball* $B_i(x)$ the dilation of the point x with δ_i . x is the center of the ball. $B_i(x)$ is a maximal ball if:

$$B_i(x) \text{ is maximal} \Leftrightarrow (\nexists y, k) | B_i(x) \subset B_k(y) \subset S.$$

In words: there is no other ball in S containing $B_i(x)$. Figure 6.20 shows some examples of maximal balls.

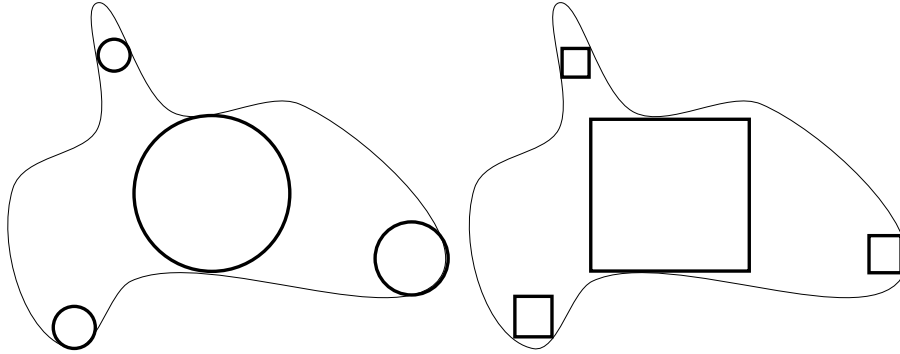


Figure 6.20 Examples of maximal balls in sets. On the left, a circular structuring element is used. On the right, the “balls” are formed by a square structuring element.

We can compute the centers of the maximal balls of size i of a set S by computing the residue $\text{MB}_i = \varepsilon_i(S) - \gamma((\varepsilon_i)(S))$, where the opening γ uses the smallest possible structuring element. The *skeleton by maximal balls* $S_{\text{MB}}(S)$ of a set S is defined as the set of centers of all maximal balls of S :

$$S_{\text{MB}} = \bigcup_{i=0}^{\infty} \{\text{MB}_i\}.$$

For this definition to be useful in the case of digital images, we must include the “trivial” 1×1 structuring element in the computation, and choose for the opening γ the next smallest structuring element.

Example

When computing the skeleton by maximal balls using the family of square structuring elements in a digital image, we use for structuring elements

and *always* a 3×3 square for the opening operation γ .

Given the image $f =$

then $\varepsilon_0(f) = f$, and $\gamma\varepsilon_0(f)$ and MB_0 equal

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

[illegible]



Figure 6.21 In the middle, a skeleton by maximal balls is shown. This skeleton was obtained using a square structuring element. On the right, a skeleton obtained by thinning is shown.

Figure 6.21 shows an example of a skeleton obtained in this way.

A disadvantage of the skeleton by maximal balls is that the skeleton may be disconnected within an object. In practice, instead of the skeleton by maximal balls a skeleton formed by *thinning* (see section 6.6.5) is often used.

When computing the skeleton by maximal balls, we know the size i of the maximal ball around each point in the skeleton. It is therefore easy to extend the binary skeleton image to a grey value image, where each point of the skeleton gets a grey value equal to the size of the local maximal ball. The resulting ‘function’ is called a *quench* function. Figure 6.22 shows an example of a quench function. The quench function gives us a very compact representation of the original image: given only the quench function we can reconstruct the original image by dilating each point with the structuring element of the appropriate size. It is also possible to compute all openings of the original image by dilating only those points with quench values above a certain threshold. The quench function also compactly gives shape information of the original: the “center” lines of the objects, and the value of the quench function gives the morphological distance to the nearest object boundary.

6.6.3.3 The SKIZ

In a binary image, we can also compute the skeleton of the background (the complement) of the image. This skeleton is often called the *SKIZ*, the *skeleton by influence zones*, because it shows the influence zones (the collection of points closest to an object) of the foreground objects. An example of a SKIZ can be seen in figure 6.23.

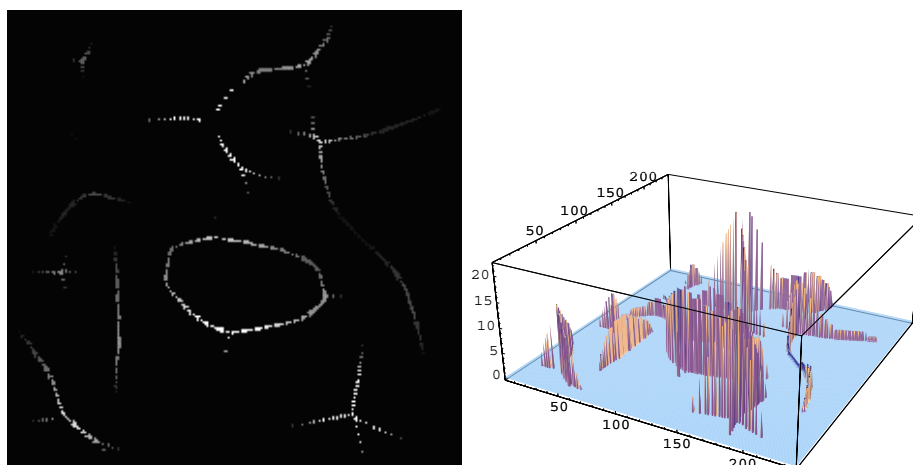


Figure 6.22 Example of a quench function obtained from the skeleton by maximal balls in the previous figure.

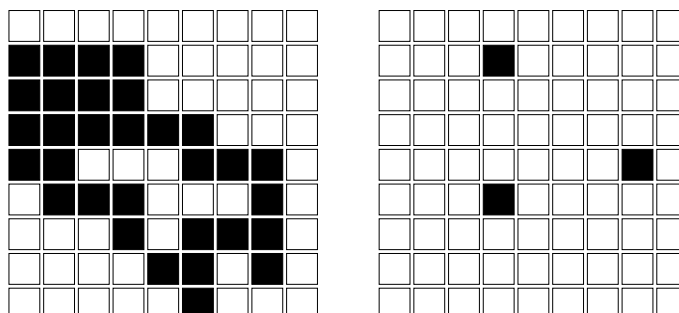
6.6.4 Hit-or-miss transformation

The hit-or-miss transformation is special in binary morphology because it uses a *composite* structuring element $X = (X_1, X_2)$. The hit-or-miss transformation of a set S is defined as all points x where X_1 fits in S , and X_2 fits in the background S^c .

Example

Given a structuring element $X_1 = \begin{bmatrix} \square & \square & \square \\ \blacksquare & \bullet & \square \\ \square & \blacksquare & \square \end{bmatrix}$ (with the circle marking the origin) and a structuring element $X_2 = \begin{bmatrix} \blacksquare & \blacksquare & \blacksquare \\ \square & \bullet & \blacksquare \\ \square & \square & \blacksquare \end{bmatrix}$, then the composite structuring element $X = (X_1, X_2) = \begin{bmatrix} \blacksquare & \blacksquare & \blacksquare \\ \square & \bullet & \blacksquare \\ \square & \square & \blacksquare \end{bmatrix}$. If we do a hit-or-miss transformation with X on a set S , then we are looking for points where X_1 fits in an object, and X_2 fits in the background. So in this case, we are looking for a type of upper-right corner.

For example: a set and its hit-or-miss transformation:



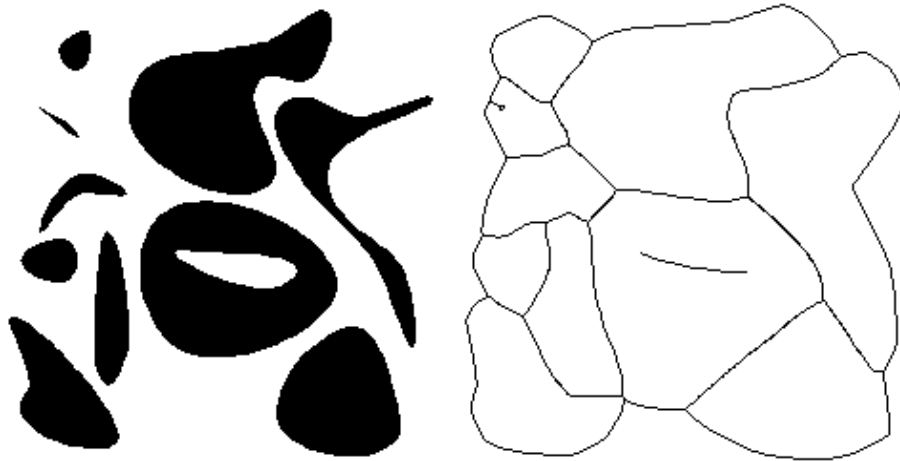


Figure 6.23 Example of a SKIZ.

Formally, we can define the hit-or-miss transformation H of a set S with a composite structuring element $X = (X_1, X_2)$ by

$$H_X(S) = \varepsilon_{X_1}(S) \cap \varepsilon_{X_2}(S^c).$$

6.6.5 Thinning and thickening

The *thinning* $\text{thin}_X(S)$ of a set S is the residue of S and its hit-or-miss transformation with the composite structuring element X :

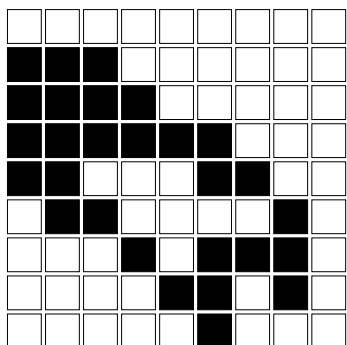
$$\text{thin}_X(S) = S - H_X(S).$$

Its dual counterpart is called *thickening* and is defined by the union of S and its hit-or-miss transformation:

$$\text{thick}_X(S) = S \cup H_X(S).$$

Example

The example in the previous section shows the hit-or-miss transformation $H_X(S)$ of a set S by X . The thinning $\text{thin}_X(S)$ equals

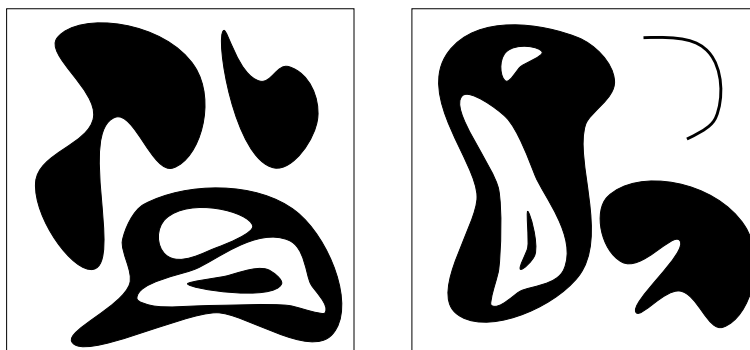


As all the elements of the hit-or-miss transformation lie inside S , the thickening $\text{thick}_X(S)$ equals S itself.

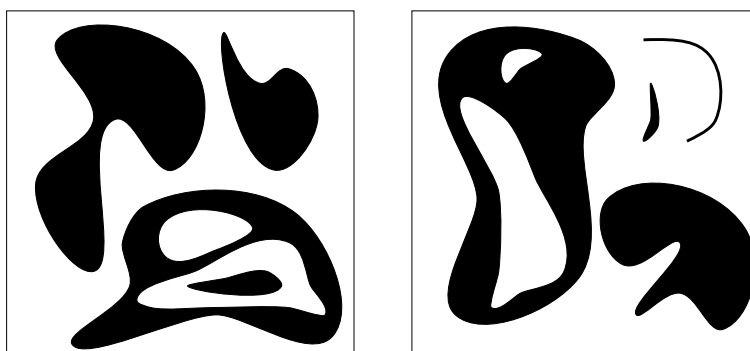
Thinning and thickening are useful operations since they allow the formulation of *homotopic* operations, *i.e.*, operations that preserve the homotopy of a set; that do not break object “connections”.

Intermezzo

Two sets are homotopic if they contain the same number of objects and each object contains the same number of holes and enclosed objects. For example: these two sets are homotopic:



And these two are *not*:



Another definition of homotopy is that the two sets can be continuously deformed to each other.

The structuring elements that are most useful can be found in the so-called *Golay* alphabet. Here, we will only discuss the L Golay element, which is most used. The L element is shown in figure 6.24. What is called the “thinning” with structuring element L is in

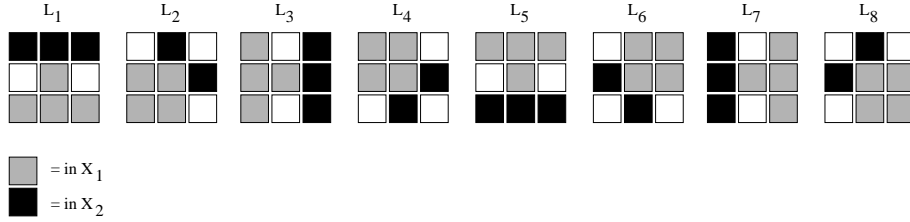


Figure 6.24 The L Golay element. Shown are the basic element L_1 and rotated version as they are used in thinning and thickening. Note that the rotations of L are not exact.

fact a sequence of thinnings with the rotated elements L_1, \dots, L_8 :

$$\text{thin}_L(S) = \text{thin}_{L_1}(\text{thin}_{L_2}(\dots(\text{thin}_{L_8}(S))))).$$

Iterated thinning of a set with L until the set no longer changes results in a skeleton of the set. An example of such a skeleton has already been shown in figure 6.21. The advantage of this skeleton is that it is homotopic with the original set. The skeleton by maximal balls does not have this property.

6.7 Applications

6.7.1 Alternating sequential filters

The ultimate erosion and skeleton are examples of structures that can be obtained by the residue of two families of operators. Another way of combining two families of operators $\{\varphi_i\}$ and $\{\psi_i\}$ is by creating an *alternating sequential filter* (ASF), which is a composition of operators such as:

$$\text{ASF1}_i = \psi_i \varphi_i \dots \psi_2 \varphi_2 \psi_1 \varphi_1$$

$$\text{ASF2}_i = \varphi_i \psi_i \dots \varphi_2 \psi_2 \varphi_1 \psi_1$$

$$\text{ASF3}_i = \psi_i \varphi_i \psi_i \dots \psi_2 \varphi_2 \psi_2 \psi_1 \varphi_1 \psi_1$$

$$\text{ASF4}_i = \varphi_i \psi_i \varphi_i \dots \varphi_2 \psi_2 \varphi_2 \varphi_1 \psi_1 \varphi_1.$$

In practice an ASF is only a useful operator if $\{\psi_i\}$ is decreasing as i rises, $\{\varphi_i\}$ is increasing as i rises, and $\psi_1(f) \leq \varphi_1(f)$, so if

$$\psi_i(f) \leq \dots \leq \psi_1(f) \leq \varphi_1(f) \leq \dots \leq \varphi_i(f).$$

Openings and closings are often used for the two families. ASF's are sometimes useful for dealing with very noisy images, as shown in figure 6.25.

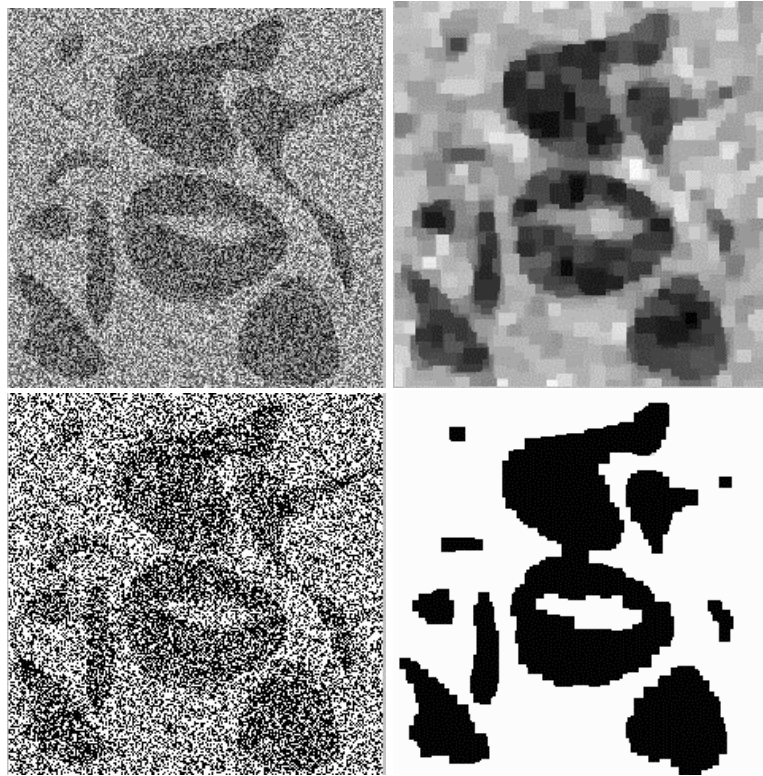


Figure 6.25 The top left shows a very noisy 256×256 image. In this case thresholding (bottom left) will not give us the objects that are vaguely visible in the original to the human observer. However, after application of an ASF using openings and closings with square structuring elements of sizes 3 to 7 (top right), thresholding *does* give us approximately the desired result (bottom right).

6.7.2 Granulometry

It is often useful to know the *size distribution* of an image, *i.e.*, *e.g.*, to know what fraction of an image is filled with objects or object parts of a certain size i . The study of the size distribution in an image is called *granulometry*. In previous sections, we saw

that an opening removes objects related to the size of the structuring element from an image, so it seems logical to use openings (and closings) of various sizes to study the size distribution of an image. A practical approach to measuring a size distribution of an image f is

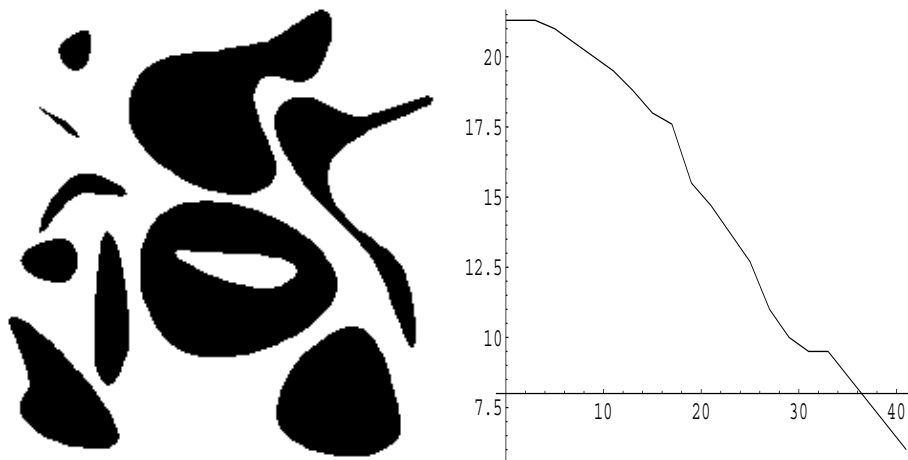
1. Open the image with a structuring element of a certain size.
2. Do a size measurement on the current image, e.g., the area of all object pixels, or the sum of all grey values.
3. Enlarge the structuring element.
4. Repeat from 1 until the opening removes the entire image.

This is in fact an example of a granulometry with the family of openings $\{\gamma_i\}$. In general, any decreasing⁸ family $\{\psi_i\}$ with the following property can be used for a granulometry: $\psi_i\psi_j(f) = \psi_{\max\{i,j\}}(f)$, i.e., applying the “strongest” ψ has the same effect as applying both ψ_i and ψ_j .

The size measurements together make up a granulometric curve.

Example

An example of a granulometric curve of an image is:



The x axis shows the size of the opening by a square structuring element. The y axis shows the number of pixels ($\cdot 10^3$) contained in the opening. The original image contains 57121 (239×239) pixels.

The granulometric curve obtained with granulometry by opening is often extended using closings, as figure 6.26 shows.

⁸ $\psi_i(f) \leq \psi_j(f)$ if $i > j$.

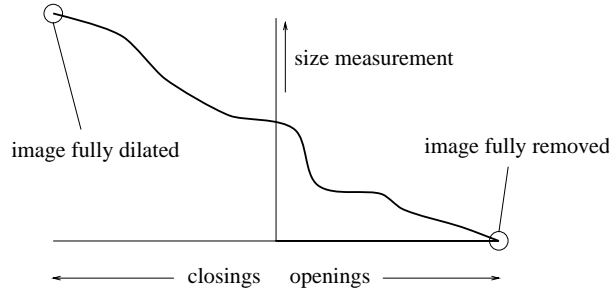


Figure 6.26 Extension of a granulometric curve with closings. The closing measurements are shown on the left side of the origin, moving to the left increases the closing structuring element size. On the right of the origin are the opening measurements. Moving to the right increases the opening structuring element size.

6.7.3 Toggle mapping

Toggle mapping is a morphological technique to enhance contrast in images. Given an anti-extensive transformation ψ_1 (e.g., erosion) and an extensive transformation ψ_2 (e.g., dilation) of an image f , the toggle map f_{TM} is defined by

$$f_{\text{TM}} = \begin{cases} \psi_1(f) & \text{if } f - \psi_1(f) < \psi_2(f) - f \\ \psi_2(f) & \text{otherwise.} \end{cases}$$

In other words, f is replaced by the transformation value $\psi_i(f)$ that is closest to f .

A type of toggle mapping that is also known as *morphological deblurring* uses erosion for ψ_1 and dilation for ψ_2 . So in this operation each image value of f is either replaced by the eroded value or the dilated value of f , whichever one is closest to the original value of f . The figures 6.27 and 6.28 show examples of deblurring applied to real images.

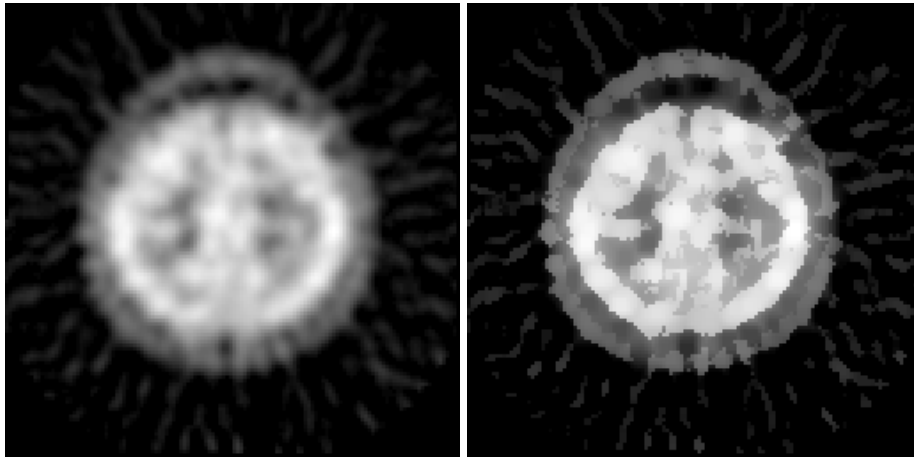


Figure 6.27 Example of morphological deblurring using a 3×3 square structuring element. The original image has a resolution of 128×128 .

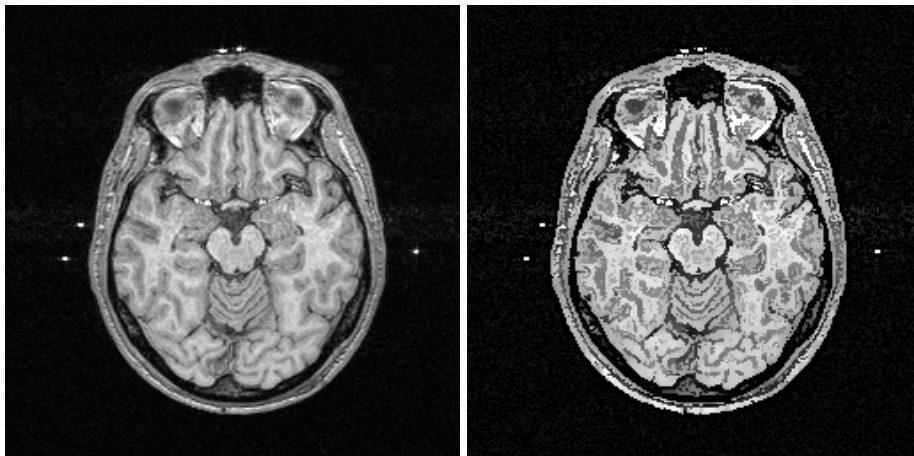


Figure 6.28 Example of morphological deblurring using a 5×5 square structuring element. The original image has a resolution of 256×256 .

