

Locally adaptable mathematical morphology using distance transformations

Olivier Cuisenaire

Signal Processing Institute (ITS)

Swiss Federal Institute of Technology (EPFL)

CH-1015 Lausanne, Switzerland

Phone: +41 21 6934712

Fax: +41 21 6937600

Abstract

We investigate how common binary mathematical morphology operators can be adapted so that the size of the structuring element can vary across the image pixels. We show that when the structuring elements are balls of a metric, locally adaptable erosion and dilation can be efficiently implemented as a variant of distance transformation algorithms. Opening and closing are obtained by a local threshold of a distance transformation, followed by the adaptable dilation.

Key words: mathematical morphology, distance transformation, adaptive filtering

Email address: olivier.cuisenaire@epfl.ch (Olivier Cuisenaire).

URL: itswww.epfl.ch/~cuisenai (Olivier Cuisenaire).

1 Introduction

Mathematical morphology [1–3] on binary images is a set oriented approach to image processing. Typically, it relies on a small set B of pixel locations, called structuring element (SE), that is translated over the image I . Logical operations about whether the pixels in the translated B belong or not to an object X define operations such as the dilation $X \oplus B$, erosion $X \ominus B$, opening X_B and closing X^B .

Defined this way, mathematical morphology operators have translation invariance, notwithstanding boundary effects. But, as Serra points out in [4], translation invariance is more cumbersome than helpful, and theoretically useless. A typical example - from chapter 4 of [3] - is the analysis of images from traffic control cameras. Because of the perspective effect, vehicles at the bottom of the image are closer and appear larger than those higher in the image. Hence, the SE size should be modulated by the perspective function, i.e. vary linearly with the vertical position in the image.

More complex variations of the relevant structuring element size are possible. In [5], Roerdink considers a photograph of the trees in a forest, taken by putting the camera at ground level and aiming towards the sky. This case, explicitly excluded in [2] from the application field of Euclidean morphology, requires a polar structure and SE sizes increasing with the distance to the center of the image. Verly [6] applies mathematical morphology to range imagery, a modality where the value of each pixel is the distance to the imaging device. Thus, in order to take perspective into account the SE size should be adapted locally to the image content.

In [7], Chen considers using structuring elements of variable sizes to filter a one-dimensional signal. Statistical analysis shows that such a method outperforms morphological filters with a space-invariant SE. Masayasu [8] defines morphological operators on grey-level images with constant SE support but uses non-flat structuring elements for which the SE function can vary according to the image content. While these operators prove useful to process ultrasound images, they do not respect all the required properties of morphological filters.

Roerdink has published several papers laying down the theoretical background for a mathematical morphology that is not based on translation-invariant transformations of the Euclidean space. He defines polar morphology [5], constrained perspective morphology, spherical morphology, translation-rotation morphology, projective morphology and differential morphology. All those morphologies are brought together in the general framework of group morphology [9].

Finally, Charif-Chefchaoui and Schonfeld [10] propose a comprehensive theory of spatially-variant binary mathematical morphology where the structuring elements can vary both in size and shape. While they prove important theoretical properties of these operators, they offer no efficient way to implement them.

Indeed, while much work has been spent on developing the mathematical tools to handle the above problems, relatively little has been done on developing efficient implementations of these tools, a fundamental issue in order to ensure their practical use. In this paper we focus explicitly on implementability. We define morphological operators with structuring elements whose size can vary

over the image without any constraint. On the other hand, the SE shape is identical for the whole image plane and has to be a ball of a distance metric. This SE shape constraint allows us to propose efficient algorithms for size-adaptable erosion, dilation, opening and closing, at a computational cost similar to the most efficient Euclidean distance transformation algorithms. Fig. 1 shows an example of locally adaptable morphological operations where the size of the structuring elements varies.

This paper is organized as follows. In section 2, we recall how classical mathematical morphology can be implemented using a distance transformation when structuring elements are balls of a metric. Section 3 extends this approach to balls of varying size and proposes an algorithm for the adaptable dilation and erosion operators. Section 4 shows that the algorithm does reach the desired result. Section 5 considers closings and openings. It shows that combining an adaptable dilation with an adaptable erosion does not necessarily give a closing, but offers an alternative method that replaces the first adaptable dilation by the threshold of a distance map. Section 6 proves that this operator is indeed a closing. Furthermore, it shows that replacing the dilation part was an arbitrary choice and that another closing can be defined where the adaptable erosion is replaced by the threshold of a distance map. Section 7 analyses the computational complexity of the algorithms and compares experimental CPU time measures with the brute force alternative. Section 8 considers how these methods can be extended to images in more than 2 dimensions, as well as to grey-scale images. Finally, section 9 discusses how the operators defined in this paper relate to Charif-ChefChaouni's and Roerdink's works, as well as possible applications.

2 DT-based Morphology

Distance transformations (DT) have been widely used to implement binary mathematical morphology operations efficiently, as discussed by Vincent in chapter 8 of [1]. For instance, Ragnelmam [11] and Cuisenaire [12] proposed algorithms based on the Euclidean distance transformation by ordered propagation to implement the morphological dilation of an object X by a circular structuring element B in a time proportional to the number of pixels in $(X \oplus B) \setminus X$.

For this purpose, we must restrict ourselves to structuring elements that are balls of a given metric, i.e. structuring elements B_d such that

$$B_d = \{\mathbf{h} : \|\mathbf{h}\| < d\} \quad (1)$$

The dilation operation is then defined as

$$\begin{aligned} X \oplus B_d &= \{\mathbf{x} + \mathbf{h} : \mathbf{x} \in X, \mathbf{h} \in B_d\} & (2) \\ &= \{\mathbf{x} + \mathbf{h} : \mathbf{x} \in X, \|\mathbf{h}\| < d\} \\ &= \{\mathbf{y} : \exists \mathbf{x} \in X, \|\mathbf{y} - \mathbf{x}\| < d\} \\ &= \left\{ \mathbf{y} : \min_{\mathbf{x} \in X} (\|\mathbf{y} - \mathbf{x}\|) < d \right\} \\ &= \{\mathbf{y} : D_X(\mathbf{y}) < d\} & (3) \end{aligned}$$

The later expression uses the distance transformation D_X defined as

$$D_X(\mathbf{p}) = \min_{\mathbf{x} \in X} (\|\mathbf{p} - \mathbf{x}\|) \quad (4)$$

for which numerous efficient algorithms exist [12–16]. Hence, the dilation is implemented by computing the distance transformation, then applying a threshold by the value d that defines the size of the ball B_d . Erosion, opening and

closing are obtained by combining dilations with set complementations.

In what follows, we consider the Euclidean distance transformation and therefore balls that are circular, i.e. the metric is

$$\|\mathbf{v}\| = \sqrt{v_x^2 + v_y^2} \quad (5)$$

Nevertheless, other shapes of structuring elements can be obtained using different definitions of the distance. Indeed, the city-block metric

$$\|\mathbf{v}\| = |v_x| + |v_y| \quad (6)$$

defines balls of diamond shape. The chessboard metric

$$\|\mathbf{v}\| = \max(|v_x|, |v_y|) \quad (7)$$

defines square balls. The chamfer metric [17]

$$\|\mathbf{v}\| = \max(|v_x|, |v_y|) + \frac{1}{3} \min(|v_x|, |v_y|) \quad (8)$$

defines octagonal balls. Also, anisotropic metrics can be used to define rectangular or ellipsoidal balls. All considerations that follow can easily be extended to any of these metrics and ball shapes.

3 Dilation and Erosion

In this paper, we extend the above method to allow the size of the ball used as structuring element to vary over the image. Instead of using the same ball B_d for all pixels, we consider different balls $B_{S(\mathbf{x})}$ with varying radiuses defined in an image S of local structuring element sizes. Extending (2), we define the adaptable dilation of object X by this image S of SE sizes as

$$\begin{aligned}
X \oplus S &= \{\mathbf{x} + \mathbf{h} : \mathbf{x} \in X, \|\mathbf{h}\| \in B_{S(\mathbf{x})}\} \\
&= \{\mathbf{x} + \mathbf{h} : \mathbf{x} \in X, \|\mathbf{h}\| < S(\mathbf{x})\}
\end{aligned} \tag{9}$$

It can be implemented as efficiently as before using a modified distance measure. Indeed, similarly to (3), we have

$$\begin{aligned}
X \oplus S &= \{\mathbf{x} + \mathbf{h} : \mathbf{x} \in X, \|\mathbf{h}\| - S(\mathbf{x}) < 0\} \\
&= \{\mathbf{y} : D_{X,S}(\mathbf{y}) < 0\}
\end{aligned} \tag{10}$$

where we define $D_{X,S}$ as

$$D_{X,S}(\mathbf{p}) = \min_{\mathbf{x} \in X} (\|\mathbf{p} - \mathbf{x}\| - S(\mathbf{x})) \tag{11}$$

Note that $D_{X,S}$ is not a distance *stricto sensu* since in general it respects none of the axioms of a metric. Computing $D_{X,S}$ is relatively straightforward. The algorithm, illustrated at Fig. 2, takes two inputs. First, the binary image I in which resides an object X (Fig. 2.1). Secondly the image of structural element sizes S (Fig. 2.2). We aim to compute $D_{X,S}$ for all pixels $\mathbf{p} \in I$, and a vectorial image \mathbf{V} such that $\mathbf{V}(\mathbf{p})$ is the object pixel that minimizes this expression, i.e.

$$\mathbf{V}(\mathbf{p}) = \arg \min_{\mathbf{x} \in X} (\|\mathbf{p} - \mathbf{x}\| - S(\mathbf{x})) \tag{12}$$

\mathbf{V} is the Voronoi partition of the image for the modified distance $D_{X,S}$.

Firstly, we initialize (Fig. 2.3) all objects pixels with $D_{X,S}(\mathbf{p}) = -S(\mathbf{p})$ and $\mathbf{V}(\mathbf{p}) = \mathbf{p}$. For background pixels, we should set $D_{X,S}(\mathbf{p})$ to ∞ and leave $\mathbf{V}(\mathbf{p})$ unassigned. Practically, we set $D_{X,S}(\mathbf{p})$ to 0 for background pixels and compute $\min(D_{X,S}, 0)$, which limits the amount of computations and does not affect the final computation of $X \oplus S$ which involves a threshold by 0 anyway.

Secondly, we propagate this information from neighbor to neighbor. The values

of $D_{X,S}$ and \mathbf{V} at pixel \mathbf{p} are modified by its neighbor $\mathbf{p} + \mathbf{n}$ if we have

$$\|\mathbf{p} - \mathbf{V}(\mathbf{p} + \mathbf{n})\| - S(\mathbf{V}(\mathbf{p} + \mathbf{n})) < D_{X,S}(\mathbf{p}) \quad (13)$$

There are several ways to implement this propagation. The most intuitive method would be to use a dynamic list of propagating pixels to scan the image by order of increasing values of $D_{X,S}(\mathbf{p})$, adapting the algorithms of Ragnelmam [18,11] or Cuisenaire [14] for the Euclidean DT. Nevertheless, this is needlessly complex. Instead, we adapt the original 4SED Euclidean DT algorithm of Danielsson [19] which uses a particular type of raster scanning. The first scan operates line by line from top to bottom. Each line is first scanned from left to right considering the up and left direct neighbors, then from right to left considering the right neighbor. The second scan operates similarly from bottom to top and from right to left. Algorithm 1 formalizes the method for an image of size $M \times N$.

As in the case of classical mathematical morphology, the erosion is the dual operation of dilation and can be obtained as

$$X \ominus S = (X^c \oplus S)^c \quad (14)$$

where $X^c = \{\mathbf{x} : \mathbf{x} \notin X\}$ is the complement set of X .

4 Analysis of the algorithm

In the special case where S has a constant value for all pixels, algorithm 1 is identical to Danielsson's Euclidean DT [19] followed by a threshold. The rationale for Danielsson's algorithm is that while the definition (4) of the DT

```

for all  $\mathbf{p} \in I$  do
  if  $\mathbf{p} \in X$  then
     $D_{X,S}(\mathbf{p}) \leftarrow -S(\mathbf{p})$  ;  $\mathbf{V}(\mathbf{p}) \leftarrow \mathbf{p}$ 
  else
     $D_{X,S}(\mathbf{p}) \leftarrow 0$  ;  $\mathbf{V}(\mathbf{p}) \leftarrow (\infty, \infty)$ 

for  $p_y = 1 \rightarrow N$  do
  for  $p_x = 1 \rightarrow M$  do
     $\text{check}((p_x, p_y), (-1, 0))$ 
     $\text{check}((p_x, p_y), (0, -1))$ 
  for  $p_x = M \rightarrow 1$  do
     $\text{check}((p_x, p_y), (1, 0))$ 
for  $p_y = N \rightarrow 1$  do
  for  $p_x = M \rightarrow 1$  do
     $\text{check}((p_x, p_y), (1, 0))$ 
     $\text{check}((p_x, p_y), (0, 1))$ 
  for  $p_x = 1 \rightarrow M$  do
     $\text{check}((p_x, p_y), (-1, 0))$ 

for all  $\mathbf{p} \in I$  do
  if  $D_{X,S}(\mathbf{p}) < 0$  then
     $\mathbf{p} \in X \oplus S$ 
  else
     $\mathbf{p} \in (X \oplus S)^c$ 

procedure  $\text{check}(\mathbf{p}, \mathbf{n})$ 
  if  $\mathbf{p} + \mathbf{n} \in I$  then
     $\mathbf{v} \leftarrow \mathbf{V}(\mathbf{p} + \mathbf{n})$ 
    if  $\mathbf{v} \neq (\infty, \infty)$  then
       $d \leftarrow \|\mathbf{p} - \mathbf{v}\| - S(\mathbf{v})$ 
      if  $d < D_{X,S}(\mathbf{p})$  then
         $D_{X,S}(\mathbf{p}) \leftarrow d$  ;  $\mathbf{V}(\mathbf{p}) \leftarrow \mathbf{v}$ 

```

Algorithm 1: computes $X \oplus S$

is global, the influence of an object pixel \mathbf{x} is essentially local. Indeed, the tiles of the Voronoi partition \mathbf{V} , i.e the sets

$$T(\mathbf{x}) = \{\mathbf{p} : \mathbf{V}(\mathbf{p}) = \mathbf{x}\} \quad (15)$$

defined for each object pixel \mathbf{x} , are convex polygons around \mathbf{x} . Therefore, every pixel in $T(\mathbf{x})$ is reachable from \mathbf{x} in two scans. Practically, this is only true in the continuous case. In a digital image, $T(\mathbf{x})$ may be disconnected when a corner of the polygon is thinner than the image grid. Fortunately, this only leads to a few isolated errors where the distance value is slightly overestimated. If a perfect Euclidean DT is needed, it can be achieved by a post-processing of \mathbf{V} as in [13].

In the general case where S can vary arbitrarily, the shape of the tiles in \mathbf{V} is of course more complex. Let us consider the simplest case where the object $X = \{\mathbf{a}, \mathbf{b}\}$ is made of two pixels. The limit between the tiles $T(\mathbf{a})$ and $T(\mathbf{b})$ is the locus of pixels \mathbf{p} that respect

$$\|\mathbf{p} - \mathbf{a}\| - S(\mathbf{a}) = \|\mathbf{p} - \mathbf{b}\| - S(\mathbf{b}) \quad (16)$$

When $\|\mathbf{a} - \mathbf{b}\| \geq |S(\mathbf{a}) - S(\mathbf{b})|$, this is one arc of the hyperbola with foci in \mathbf{a} and \mathbf{b} and eccentricity

$$e = \frac{\|\mathbf{a} - \mathbf{b}\|}{|S(\mathbf{a}) - S(\mathbf{b})|} \quad (17)$$

Otherwise - when $\|\mathbf{a} - \mathbf{b}\| < |S(\mathbf{a}) - S(\mathbf{b})|$ - the tile of the object pixel \mathbf{x} with the largest $S(\mathbf{x})$ covers the whole image plane and the tile of the other one is empty. In the special case where $S(\mathbf{a}) = S(\mathbf{b})$, the hyperbola degenerates into a straight line.

When X has more than two pixels, the tiles are areas separated by segments

of lines or hyperbolae. Thus, the tiles are not necessarily convex, the basic assumption for Danielsson's algorithm. Fortunately, we can prove a weaker property that is sufficient for the propagation to reach all pixels in the tiles. The tiles are star-shaped, i.e.

$$\mathbf{p} \in T(\mathbf{x}) \Rightarrow \forall \alpha \in [0, 1], \mathbf{x} + \alpha \cdot (\mathbf{p} - \mathbf{x}) \in T(\mathbf{x}) \quad (18)$$

Indeed, let us consider a pixel $\mathbf{p} \in T(\mathbf{x})$. For any other object pixel $\mathbf{y} \in X$, we have

$$\|\mathbf{p} - \mathbf{x}\| - S(\mathbf{x}) \leq \|\mathbf{p} - \mathbf{y}\| - S(\mathbf{y}) \quad (19)$$

Let us then consider a pixel $\mathbf{q} = \mathbf{x} + \alpha \cdot (\mathbf{p} - \mathbf{x})$. By the triangular inequality, we have

$$\|\mathbf{p} - \mathbf{y}\| \leq \|\mathbf{p} - \mathbf{q}\| + \|\mathbf{q} - \mathbf{y}\| \quad (20)$$

The definition of \mathbf{q} also gives

$$\begin{aligned} \|\mathbf{p} - \mathbf{q}\| &= \|\mathbf{p} - \mathbf{x} - \alpha \cdot (\mathbf{p} - \mathbf{x})\| \\ &= (1 - \alpha) \cdot \|\mathbf{p} - \mathbf{x}\| && \text{if } \alpha \leq 1 \end{aligned} \quad (21)$$

$$\|\mathbf{q} - \mathbf{x}\| = \alpha \cdot \|\mathbf{p} - \mathbf{x}\| \quad \text{if } \alpha \geq 0 \quad (22)$$

$$= \|\mathbf{p} - \mathbf{x}\| - \|\mathbf{p} - \mathbf{q}\| \quad \text{if } 0 \leq \alpha \leq 1 \quad (23)$$

By adding (19) and (20), then using (23), we get

$$\begin{aligned} \|\mathbf{p} - \mathbf{x}\| - \|\mathbf{p} - \mathbf{q}\| - S(\mathbf{x}) &\leq \|\mathbf{q} - \mathbf{y}\| - S(\mathbf{y}) \\ \|\mathbf{q} - \mathbf{x}\| - S(\mathbf{x}) &\leq \|\mathbf{q} - \mathbf{y}\| - S(\mathbf{y}) \end{aligned} \quad (24)$$

which is valid for any object pixel \mathbf{y} , and therefore we have $\mathbf{q} \in T(\mathbf{x})$.

In the continuous image plane, this property guarantees that there is a direct

propagation path from \mathbf{x} to all the pixels in $T(\mathbf{x})$. Similarly to Danielsson's algorithm, there is no such guarantee in the discrete case when the corner of a tile can be thinner than the grid step. This can lead to occasional small errors where $D_{X,S}$ is slightly overestimated. Thus, a few pixels at the edge of $X \oplus S$ can be mistakenly considered as belonging to $(X \oplus S)^c$. In most practical cases this is of no consequence, and it can be partially corrected by using a larger neighborhood in the raster scanning algorithm, as Danielsson [19] does with the 8SED algorithm.

5 Opening and Closing

In Euclidean morphology [5], the closing X^B and opening X_B of an object X by a structuring element B are defined respectively as

$$X^B = (X \oplus \check{B}) \ominus B \tag{25}$$

$$X_B = (X \ominus \check{B}) \oplus B \tag{26}$$

where $\check{B} = \{-\mathbf{b} : \mathbf{b} \in B\}$ is the reflected set of B . In the case of a symmetric structuring element, we have $\check{B} = B$.

When it comes to locally adaptable opening and closing, things are more complicated. In (9), S is not a set of pixel locations, but an image of SE sizes. Thus, there is no obvious way to compute a reflected \check{S} which would give

$$X \oplus \check{S} = \{\mathbf{x} + \mathbf{h} : \mathbf{x} \in X, \|\mathbf{h}\| < \check{S}(\mathbf{x})\} \tag{27}$$

A naive approach would be to consider that since the circular balls we use are

symmetric, we can assume $\check{S} = S$ and define the closing X^S as

$$(X \oplus S) \ominus S \quad (28)$$

Unfortunately, if we do so, the resulting operations do not respect important properties of opening and closing. In particular, we do not have idempotence, nor the extensivity of X_S and anti-extensivity of X^S , i.e.

$$X_S \subseteq X \subseteq X^S \quad (29)$$

as illustrated at the top of Fig. 3. The problem with (28) is that at the dilation step, we consider values $S(\mathbf{p})$ at pixels $\mathbf{p} \in X$, while at the erosion step, we use values $S(\mathbf{p})$ at different pixels $\mathbf{p} \in (X \oplus S)^c$. Thus, the dilation and erosion steps use different local structuring element sizes, which leads to the problems of Fig. 3. In general, it is impossible to define a reflected \check{S} . Fortunately, it is instead possible to define a reflected dilation operation $\check{\oplus}$ as

$$X \check{\oplus} S = \{\mathbf{y} : \exists \mathbf{y} - \mathbf{h} \in X, \|\mathbf{h}\| < S(\mathbf{y})\} \quad (30)$$

This expression is very similar to (27), but instead of considering the value of an hypothetical \check{S} for the pixels in X , we use the value of S itself on the pixels of the result $X \check{\oplus} S$. From this, we can compute the closing as

$$X^S = (X \check{\oplus} S) \ominus S \quad (31)$$

as illustrated at the bottom of Fig. 3 Implementing the closing is straightforward once we notice that (30) can be written as

$$X \check{\oplus} S = \{\mathbf{y} : D_X(\mathbf{y}) < S(\mathbf{y})\} \quad (32)$$

Compute $D_X(\mathbf{p}) = \min_{\mathbf{x} \in X} (\|\mathbf{p} - \mathbf{x}\|)$ using the Euclidean DT algorithm in [13]

for all $\mathbf{p} \in I$ **do**

if $D_X(\mathbf{p}) < S(\mathbf{p})$ **then**

$\mathbf{p} \in Y = X \oplus S$

else

$\mathbf{p} \in Y^c = (X \oplus S)^c$

Compute $X^S = (Y^c \oplus S)^c$ using algorithm 1

Algorithm 2: computes X^S

which uses the classical distance transform D_X defined as

$$D_X(\mathbf{p}) = \min_{\mathbf{x} \in X} (\|\mathbf{p} - \mathbf{x}\|) \quad (33)$$

For city-block or chessboard metrics, this is easily computed with simple algorithms [20]. For the Euclidean metric, it can also be computed efficiently, as in [13] for instance. Finally, from $Y = X \oplus S$, we compute $X^S = Y \ominus S$ using the algorithm of the previous section. This is summarized in algorithm 2. The opening is obtained by duality.

$$X_S = ((X^c)^S)^c \quad (34)$$

6 Properties

In order to check that X^S and X_S are respectively a morphological closing and opening, we need to check that X^S is increasing, i.e.

$$X_1 \subseteq X_2 \Rightarrow X_1^S \subseteq X_2^S \quad (35)$$

anti-extensive, i.e.

$$X \subseteq X^S \quad (36)$$

and idempotent, i.e.

$$(X^S)^S = X^S \quad (37)$$

Similarly, the opening X_S is increasing, extensive and idempotent. These properties are proved in appendix A.

Let us note that the choice of the order of operations in algorithm 2 is arbitrary. Instead of computing the distance transformation, comparing $D_X(\mathbf{p})$ to $S(\mathbf{p})$, then applying the adaptable erosion, one could apply an adaptable dilation, then compute the DT from $(X \oplus S)^c$ and finally compare $D_{(X \oplus S)^c}(\mathbf{p})$ to $S(\mathbf{p})$ for all pixels. This leads to an alternative definition of X^S

$$X^{S*} = (X \oplus S) \check{\ominus} S \quad (38)$$

with the reflected erosion $\check{\ominus}$ defined as

$$X \check{\ominus} S = (X^c \check{\oplus} S)^c \quad (39)$$

One can prove that this operation is also a closing, i.e. that it is increasing, anti-extensive and idempotent. On the other hand, it leads to a different result than X^S for non trivial S . Fig. 4, with the extreme example of a white random S , illustrates that choosing the definition of the previous section leads to a much more intuitive result.

```

 $X \oplus S = \emptyset$ 
for all  $\mathbf{x} \in X$  do
  for all  $\mathbf{h} \in B_{S(\mathbf{x})}$  do
     $(X \oplus S) \ni (\mathbf{x} + \mathbf{h})$ 

```

Algorithm 3: Brute-force computation of $X \oplus S$

7 Computational Complexity

The computational complexity of the algorithms presented here is very low. The core of algorithm 1 requires two scans over the image for a total of 6 comparisons per pixel. Furthermore, as long as the size of a line is small enough to fit entirely in cache memory, pixels are only fetched twice from the main memory. On the other hand, for the circular balls that use the Euclidean metric, each comparison also requires the computation of a square root operation. For openings and closings, the exact Euclidean distance transform [13] requires 3 scans of the image, and has a complexity similar to the dilation algorithm. Initializations and set complementations add a negligible overhead.

Typically, the closings that illustrate this paper require approximately 130 ms for a 512×512 image on a 1.8 GHz pentium 4 computer. CPU time scales linearly with the number of pixels in the image and are independent of the size of the local structuring elements used. This should be compared with the brute force approach which implements the following equations

$$X \oplus S = \{\mathbf{x} + \mathbf{h} \mid \mathbf{x} \in X, \mathbf{h} \in B_{S(\mathbf{x})}\} \tag{40}$$

$$X \ominus S = \{\mathbf{x} \mid \exists \mathbf{h} \in B_{S(\mathbf{x})}, \mathbf{x} + \mathbf{h} \in X\} \tag{41}$$

Algorithm 3 summarizes the brute force approach for the adaptable dilation, and similarly algorithm 4 implements the reflected dilation.

```

 $X \overset{\checkmark}{\oplus} S = X$ 
for all  $\mathbf{x} \in X^c$  do
  for all  $\mathbf{h} \in B_{S(\mathbf{x})}$  do
    if  $(\mathbf{x} + \mathbf{h}) \in X$  then
       $(X \overset{\checkmark}{\oplus} S) \ni \mathbf{x}$ 
      jump to next  $\mathbf{x}$ 

```

Algorithm 4: Brute-force computation of $X \overset{\checkmark}{\oplus} S$

The two approaches are compared at Fig. 5 for the 512×512 synthetic image used throughout this paper and randomly sized circular structuring elements with a mean radius that varies between 1 and 50 pixels. Apart from the trivial case of radius 1, the DT-based approach is always significantly faster - up to 50 times faster in this example - than the brute force alternative. Fig. 5 also illustrates that the computational cost of the DT-based approach does not depend on the size of the structuring elements used.

The comparative advantage of the DT-based approach is even more dramatic in the 3D case discussed in the next section. For a typical application - the closing of a $512 \times 512 \times 198$ CT image of $0.42 \times 0.42 \times 0.8$ mm³ voxel size by structuring elements of average radius 15mm - the DT-based approach of this paper typically requires 2 minutes, while the brute force algorithm takes more than 13 hours to complete, which makes it approximately 400 times slower.

8 Extensions

8.1 Higher dimensional images

While the methods here have been described for 2D images, extending them to 3 or more dimensions is relatively straightforward. Appropriate neighborhoods

and scanning directions for algorithm 1 can be found in [21]. It includes a 4 scans algorithm for 3D images and 2^D scans algorithms for D dimensions.

For the openings and closings, the Euclidean DT algorithm in [13] cannot be used on images of more than 2 dimensions, but there are other linear time algorithms [15,16] that work in 3 and higher dimensions.

8.2 Gray-level images

There are two conceptual approaches to develop gray-level MM from binary MM. The first consists of using local maximum and minimum operators for the dilation and erosion. The second consists of considering an image with G gray-levels as $G - 1$ binary images, applying the binary MM operators on each of these images, then recombining the results into a gray-level image. Both approaches give equivalent results.

Using the first approach, we get the following definitions for the adaptable dilation δ_S and the erosion ε_S of a gray-level image I by a map of structuring element sizes S , as well as the other operators.

$$\delta_S(I)(\mathbf{p}) = \max(I(\mathbf{p} + \mathbf{h}) : \|\mathbf{h}\| < S(\mathbf{p} + \mathbf{h})) \quad (42)$$

$$\varepsilon_S(I)(\mathbf{p}) = \min(I(\mathbf{p} + \mathbf{h}) : \|\mathbf{h}\| < S(\mathbf{p} + \mathbf{h})) \quad (43)$$

$$\check{\delta}_S(I)(\mathbf{p}) = \max(I(\mathbf{p} + \mathbf{h}) : \|\mathbf{h}\| < S(\mathbf{p})) \quad (44)$$

$$\check{\varepsilon}_S(I)(\mathbf{p}) = \min(I(\mathbf{p} + \mathbf{h}) : \|\mathbf{h}\| < S(\mathbf{p})) \quad (45)$$

$$I^S = \varepsilon_S(\check{\delta}_S(I)) \quad (46)$$

$$I_S = \delta_S(\check{\varepsilon}_S(I)) \quad (47)$$

Let us note that these definitions are equivalent to equations (9) and (30) if the image I is binary and defined such that $I(\mathbf{p}) = 1$ if $\mathbf{p} \in X$ and $I(\mathbf{p}) = 0$ otherwise.

```

for all  $\mathbf{p} \in I$  do
     $\delta_S(I)(\mathbf{p}) \leftarrow -\infty$ 
for all  $\mathbf{p} \in I$  do
    for all  $\mathbf{h} \in B_{S(\mathbf{p})}$  do
        if  $\delta_S(I)(\mathbf{p} + \mathbf{h}) < I(\mathbf{p})$  then
             $\delta_S(I)(\mathbf{p} + \mathbf{h}) \leftarrow I(\mathbf{p})$ 

```

Algorithm 5: Brute-force grey-level dilation $\delta_S(I)$

```

for all  $\mathbf{p} \in I$  do
     $\check{\delta}_S(I)(\mathbf{p}) \leftarrow -\infty$ 
for all  $\mathbf{p} \in I$  do
    for all  $\mathbf{h} \in B_{S(\mathbf{p})}$  do
        if  $\check{\delta}_S(I)(\mathbf{p}) < I(\mathbf{p} + \mathbf{h})$  then
             $\check{\delta}_S(I)(\mathbf{p}) \leftarrow I(\mathbf{p} + \mathbf{h})$ 

```

Algorithm 6: Brute-force grey-level reversed dilation $\check{\delta}_S(I)$

One can implement these operators by directly implementing their definitions. For this purpose, equations (42) and (43) must be implemented using a write-mechanism, as in algorithm 5. On the other hand, equations (44) and (45) must be implemented using a read-mechanism, as in algorithm 6. The computational cost of this approach is obviously $\mathcal{O}(\sum_{\mathbf{p} \in I} S(\mathbf{p})^D)$ for D dimensional images since at each pixel we consider the $\mathcal{O}(S(\mathbf{p})^D)$ neighbors in the ball $B_{S(\mathbf{p})}$ around \mathbf{p} . This becomes unpractical when using large local structuring elements.

An alternative method consists of decomposing the gray-level image in as many binary images as there are gray levels, then applying binary morphological operators on each binary image and then recombining the gray-level image from the resulting level sets. For a gray-level image with G levels, i.e. with a range of $[0, G - 1]$, we have $G - 1$ images I_1 to I_{G-1} defined as

$$I_g(\mathbf{p}) = (I(\mathbf{p}) \geq g) \quad (48)$$

The gray-level dilation is computed from the $G - 1$ binary dilations as

$$\delta_S(I)(\mathbf{p}) = \sum_{g=1}^{G-1} (I_g \oplus S)(\mathbf{p}) \quad (49)$$

The computational cost of this method is approximately G times the computational cost of the binary algorithm. It becomes relevant for heavily quantized images for which G is small, or for large structuring elements, i.e. in all cases where

$$\sum_{\mathbf{p} \in I} S(\mathbf{p})^D \gg \sum_{\mathbf{p} \in I} G \quad (50)$$

9 Discussion

The mathematical morphology operators defined in this paper have strong links with previous works by Charif-Chefchaoui [10] and Roerdink [5,9], as discussed hereafter.

9.1 Link with Charif-Chefchaoui's work

In [10], Charif-Chefchaoui and Schonfeld propose a general framework for spatially varying mathematical morphology. The structuring element around a pixel \mathbf{p} is the set $\theta(\mathbf{p})$ and the spatially-varying erosion ε_θ , dilation δ_θ , opening Γ_θ and closing Φ_θ are defined as

$$\varepsilon_\theta(X) = \{ \mathbf{p} : \theta(\mathbf{p}) \in X \} \quad (51)$$

$$\delta_\theta(X) = \{ \mathbf{p} : \theta(\mathbf{p}) \cap X \neq \emptyset \} \quad (52)$$

$$\Gamma_\theta(X) = \delta_{\theta'}(\varepsilon_\theta(X)) \quad (53)$$

$$\Phi_\theta(X) = \varepsilon_{\theta'}(\delta_\theta(X)) \quad (54)$$

with

$$\theta'(\mathbf{q}) = \{ \mathbf{p} : \mathbf{q} \in \theta(\mathbf{p}) \} \quad (55)$$

The operators defined in this paper can be expressed using the same formalism provided we choose

$$\theta(\mathbf{p}) = \{ \mathbf{q} : \|\mathbf{p} - \mathbf{q}\| < S(\mathbf{p}) \} \quad (56)$$

and

$$\begin{aligned} \theta'(\mathbf{q}) &= \{ \mathbf{p} : \mathbf{q} \in \theta(\mathbf{p}) \} & (57) \\ &= \{ \mathbf{p} : \|\mathbf{p} - \mathbf{q}\| < S(\mathbf{p}) \} & (58) \end{aligned}$$

Let us note that this actually leads to the alternative definition of the opening and closing of equation (38). For the more intuitive operators defined here, one needs to swap θ and θ' in equations (53) and (54). Nevertheless, all properties proved in [10] also hold for the operators of this paper.

Also, while (55) provides a way to compute the reflected variable structuring elements, we still need both algorithms 1 and 2 for fast implementations of openings and closings, since choosing θ such that $\theta(\mathbf{p})$ respects (1) for all pixels - a necessary condition to use algorithm 1 for dilations - does not guarantee that θ' will.

9.2 Link with Roerdink's work

The methods and algorithms of this paper can be used as efficient implementation of several of Roerdink's mathematical morphologies on non Euclidean spaces. For instance, in [5], he develops mathematical morphology on a polar structure. When points are described by their polar coordinates (r, θ) , the

group operation is

$$(r_1, \theta_1) * (r_2, \theta_2) = (r_1 \cdot r_2, \theta_1 + \theta_2) \quad (59)$$

He considers dilations by structuring element B , a circle of center $(1, 0)$ and radius δ . The reflected structuring element \check{B} used to define closings is the circle centered on $((1 - \delta^2)^{-1}, 0)$ of radius $\delta \cdot (1 - \delta^2)^{-1}$. In appendix B, we show that these operations are strictly identical to those described in this paper when we specify

$$S(\mathbf{p}) = \delta \cdot \|\mathbf{p}\| \quad (60)$$

Obviously the methods of this paper can also be applied to implement some of the other group morphologies defined in [9].

9.3 Applications

There is a vast area of possible application fields where the image acquisition process involves a projection of the imaged object onto the image plane that is not properly modelled as a parallel projection along an axis perpendicular to this plane. This includes applications mentioned in the introduction such as traffic control cameras, but also others such as weather satellite images where the curvature of the earth is not negligible.

A major advantage of the method of this paper over those of Roerdink is that one does not need to develop a new brand of morphology adapted to the projection geometry for each new problem. Also, it does not require that we

have an analytical description of the projection geometry from which we derive an analytical description of S . Instead, S can be calibrated experimentally by imaging objects of a known size.

Because there is no constraint on S , it can be made dependent on the image content. This becomes adaptive mathematical morphology for which range imagery is an obvious application. In medical imaging, prior anatomical knowledge could be used to specify S appropriately.

Ultimately, this paper does not say how S should be chosen, but states that whatever the choice, it can be used to define morphological operators that can be computed efficiently. Finding the optimal S is an open issue that needs to be addressed on an application dependent basis.

10 Conclusion

In this work, we have defined morphological operators using structuring elements with a fixed shape, but sizes $S(\mathbf{p})$ that can vary arbitrarily at each pixel location \mathbf{p} . We have presented efficient algorithms using two raster scans for the adaptable dilation and erosion, and using five raster scans for opening and closing. We have presented a few applications where S is defined by the image acquisition process or the image content, but ultimately left the optimal choice of S an open question.

References

- [1] E. R. Dougherty, *Mathematical Morphology in Image Processing*, Marcel Dekker, Inc., New York, 1992.
- [2] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, New York, 1982.
- [3] J. Serra, *Image Analysis and Mathematical Morphology. Vol 2: Theoretical Advances*, Academic Press, New York, 1988.
- [4] J. Serra, Morphological filtering: an overview, *Signal Processing* 38.
- [5] J. Roerdink, H. Heijmans, Mathematical morphology for structures without translation symmetry, *Signal Processing* 15 (3) (1988) 271–277.
- [6] J. G. Verly, R. L. Delanoy, Adaptive mathematical morphology for range imagery, *IEEE Transactions on Image Processing* 2 (2) (1993) 272–275.
- [7] C. S. Chen, J. L. Wu, Y. P. Hung, Statistical analysis of space-varying morphological openings with flat structuring elements, *IEEE Transactions on Signal Processing* 44 (4) (1996) 1010–1014.
- [8] I. Masayasu, T. Masayoshi, N. Akira, Morphological operations by locally variable structuring elements and their applications to region extraction in ultrasound images, *Systems and Computers in Japan* 34 (3) (2003) 33–43.
- [9] J. B. T. M. Roerdink, Group morphology, *Pattern Recognition* 33 (2000) 877–895.

- [10] M. Charif-Chefchaoui, D. Schonfeld, Spatially-variant mathematical morphology, in: IEEE International Conference on Image Processing, 1994, pp. 555–559.
- [11] I. Ragnelmam, Fast erosion and dilation by contour processing and thresholding of distance maps, *Pattern Recognition Letters* 13 (1992) 161–166.
- [12] O. Cuisenaire, Distance transformations: fast algorithms and applications to medical image processing, Ph.D. thesis, Université catholique de Louvain (UCL), Louvain-la-Neuve, Belgium (October 1999).
- [13] O. Cuisenaire, B. Macq, Fast and exact signed euclidean distance transformation with linear complexity, in: Proc. IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP99), Vol. 6, Phoenix (AZ), 1999, pp. 3293–3296.
- [14] O. Cuisenaire, B. Macq, Fast euclidean distance transformation by propagation using multiple neighborhoods, *Computer Vision and Image Understanding* 76 (2) (1999) 163–172.
- [15] C. R. Maurer Jr, R. Qi, V. Raghavan, A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (2) (2003) 265–270.
- [16] A. Meijster, J. B. T. M. Roerdink, W. H. Hesselink, A general algorithm for computing distance transforms in linear time, in: J. Goutsias, L. Vincent, D. Bloomberg (Eds.), *Mathematical Morphology and its applications to image and signal processing*, 2000, pp. 331–340.

- [17] G. Borgefors, Distance transformations in digital images, *Computer Vision, Graphics, and Image Processing* 34 (1986) 344–371.
- [18] I. Ragnelmam, Neighborhoods for distance transformations using ordered propagation, *CVGIP, Image Understanding* 56 (3) (1992) 399–409.
- [19] P. E. Danielsson, Euclidean distance mapping, *Computer Graphics and Image Processing* 14 (1980) 227–248.
- [20] A. Rosenfeld, J. L. Pfaltz, Distance functions on digital pictures, *Pattern Recognition* 1 (1) (1968) 33–61.
- [21] I. Ragnelmam, The euclidean distance transformation in arbitrary dimensions, *Pattern Recognition Letters* 14 (1993) 883–888.

About the author

OLIVIER CUISENAIRE was born in Lobbes, Belgium, in 1973. He received the M.S. and Ph.D. degrees in electrical engineering from Université catholique de Louvain (UCL), Louvain-la-Neuve, Belgium, in 1995 and 1999, respectively. His Ph.D. thesis focused on efficient implementations of the Euclidean distance transformation and its applications to medical image processing. He was a Visiting Student at the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain in spring 1995, and at the Surgical Planning Laboratory, Harvard Medical School, Boston, MA, in the summer of 1999. He then joined the Signal Processing Institute at the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland in 2000 as Postdoctoral Researcher, and was appointed First Assistant in 2001 and Lecturer in 2003.

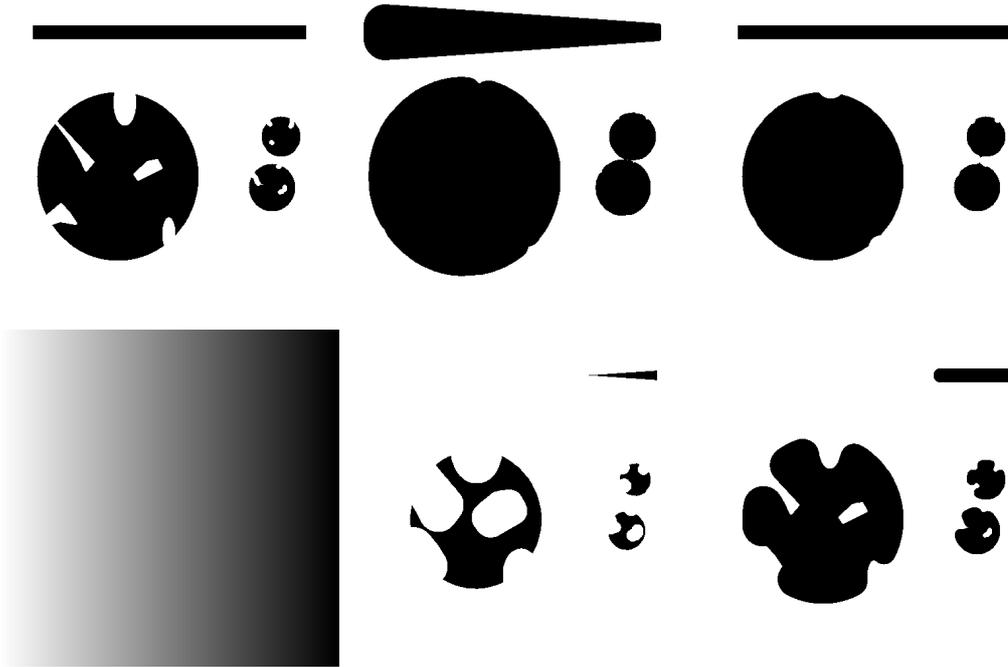


Fig. 1. **(top-left)** Original image I of size 512×512 with object X in black. **(bottom-left)** Structuring element image S chosen as $S(\mathbf{p}) = (512 - p_x)/14$. **(top-center)** Dilation $X \oplus S$. **(bottom-center)** Erosion $X \ominus S$ **(top-right)** Closing X^S **(bottom-right)** Opening X_S .

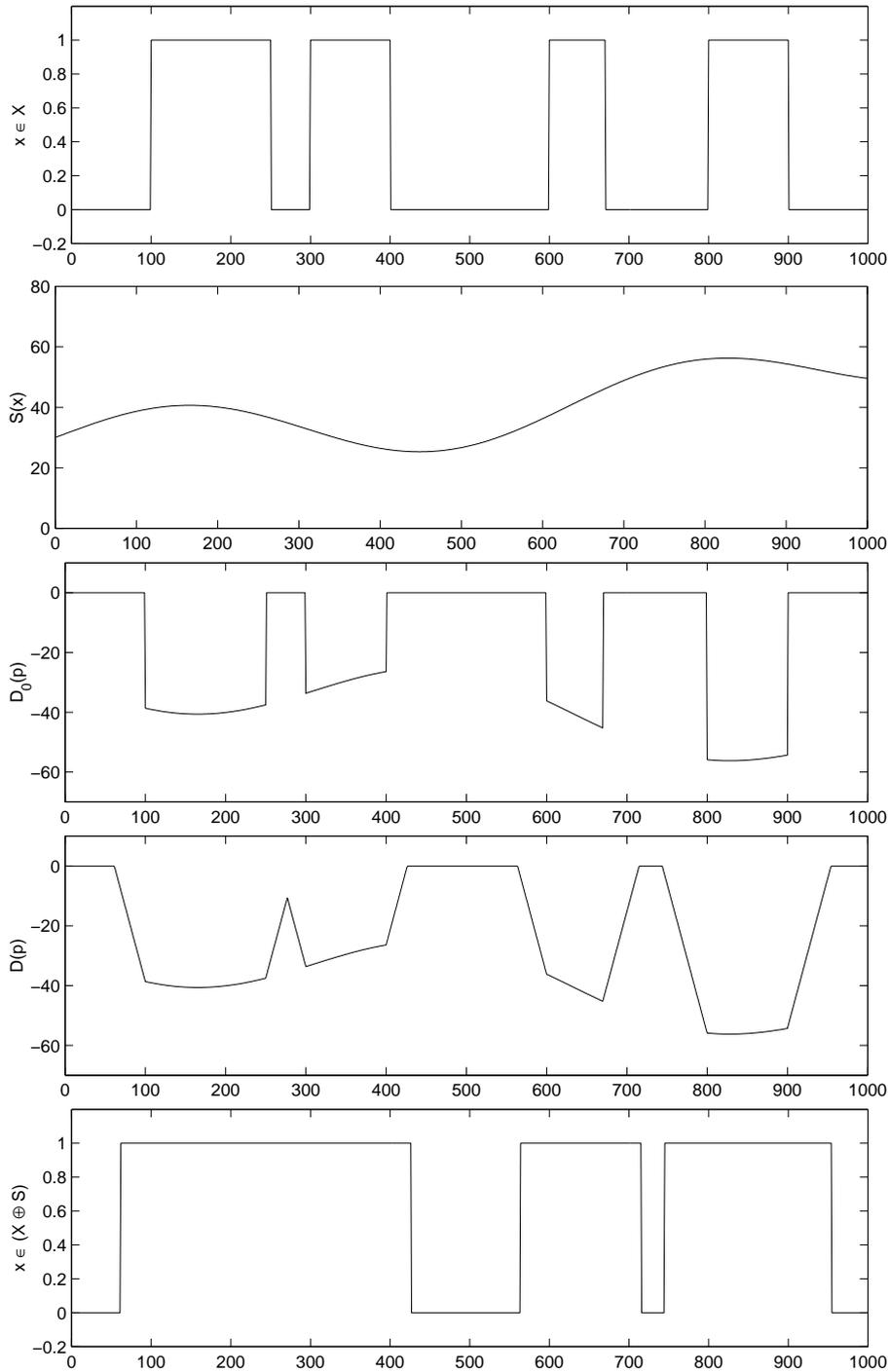


Fig. 2. Adaptive dilation algorithm in one dimension. **(1)** Object X . **(2)** Local structuring element sizes S . **(3)** Initialization of $D_{X,S}$ **(4)** $D_{X,S}$ after raster scan propagation **(5)** Resulting $X \oplus S$

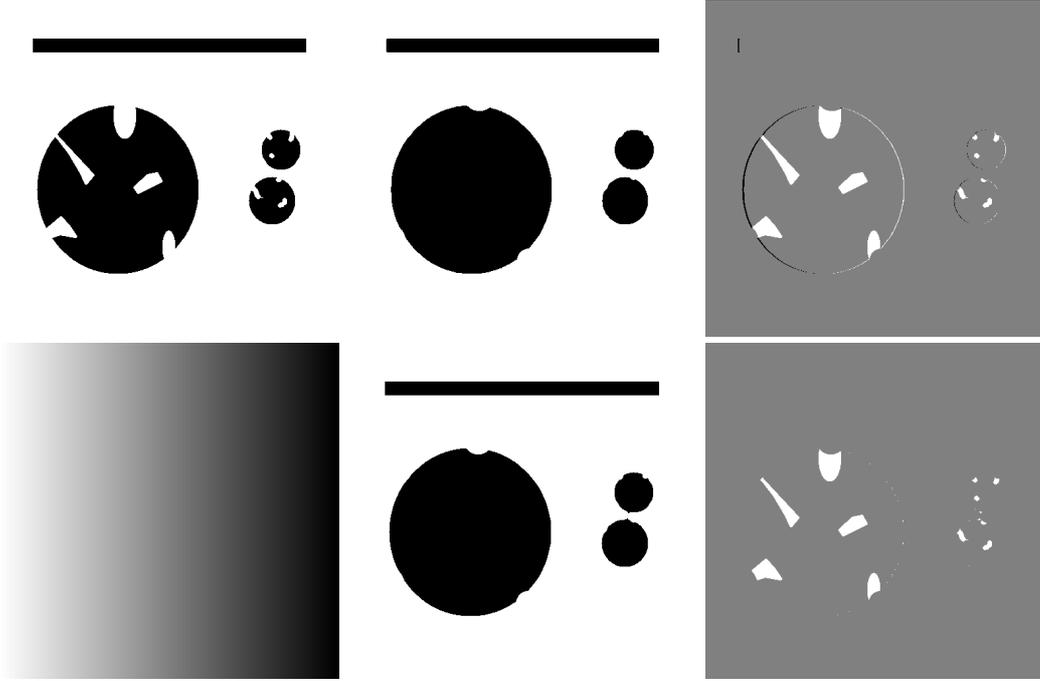


Fig. 3. **(top-left)** Original image I of size 512×512 with object X in black. **(bottom-left)** Structuring element image S chosen as $S(\mathbf{p}) = (512 - p_x)/10$. **(top-center)** Naive implementation of the closing: $(X \oplus S) \ominus S$. **(bottom-center)** Proper implementation of the closing $X^S = (X \ominus S) \oplus S$. **(top-right)** $X \setminus ((X \oplus S) \ominus S)$. Black pixels belong to X but not $(X \oplus S) \ominus S$. **(bottom-right)** $X \setminus X^S$. All pixels in X also belong to X^S .

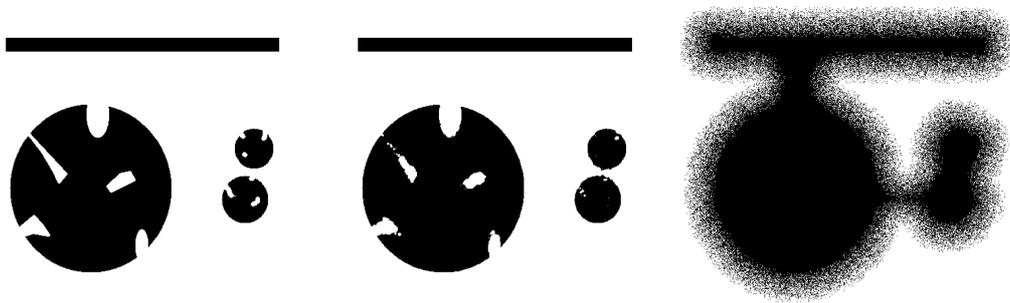


Fig. 4. **(left)** Original image I of size 512×512 with object X in black. S is a random image where each pixel is chosen independently and uniformly between 0 and 50
(center) Closing $X^S = (X \check{\oplus} S) \ominus S$ **(right)** Alternative closing $X^{S^*} = (X \oplus S) \check{\ominus} S$.

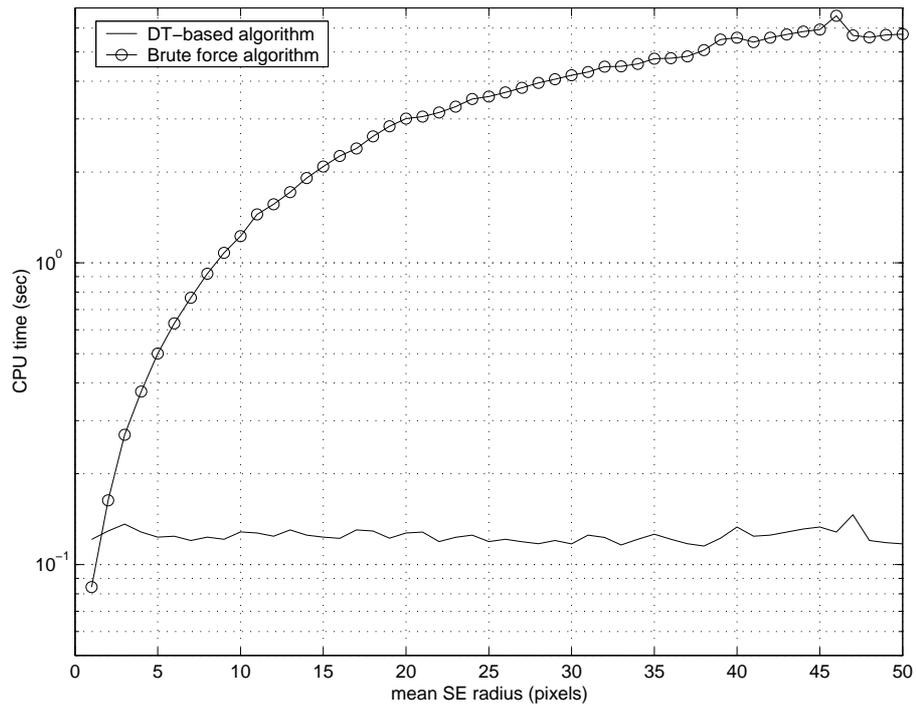


Fig. 5. Computational times for the adaptable closing of the 512×512 synthetic image of the previous figures using uniformly distributed random structuring element sizes of mean radius between 2 and 50 pixels.

A X^S is a closing

In order to prove that the X^S operation (31) is a closing, we need to show that it is increasing, anti-extensive and idempotent, i.e

$$X_1 \subseteq X_2 \Rightarrow X_1^S \subseteq X_2^S \quad (\text{A.1})$$

$$X \subseteq X^S \quad (\text{A.2})$$

$$(X^S)^S = X^S \quad (\text{A.3})$$

A.1 X^S is increasing

First, let us prove that both the adaptable dilation $X \oplus S$ and the reflected dilation $X \check{\oplus} S$ are increasing. We consider two objects X_1 and X_2 such that $X_1 \subseteq X_2$. We have

$$\begin{aligned} X_1 \oplus S &= \{\mathbf{x} + \mathbf{h} : \mathbf{x} \in X_1, \|\mathbf{h}\| < S(\mathbf{x})\} \\ &\subseteq \{\mathbf{x} + \mathbf{h} : \mathbf{x} \in X_2, \|\mathbf{h}\| < S(\mathbf{x})\} \\ &= X_2 \oplus S \end{aligned} \quad (\text{A.4})$$

and

$$\begin{aligned} X_1 \check{\oplus} S &= \{\mathbf{p} : \min_{\mathbf{x} \in X_1} \|\mathbf{p} - \mathbf{x}\| < S(\mathbf{x})\} \\ &\subseteq \{\mathbf{p} : \min_{\mathbf{x} \in X_2} \|\mathbf{p} - \mathbf{x}\| < S(\mathbf{x})\} \\ &= X_2 \check{\oplus} S \end{aligned} \quad (\text{A.5})$$

The corresponding erosions are also increasing. Indeed, using the property that $X_1 \subseteq X_2 \Rightarrow X_1^c \supseteq X_2^c$, we have

$$\begin{aligned}
& X_1 \subseteq X_2 \\
\Rightarrow & X_1^c \supseteq X_2^c \\
\Rightarrow & X_1^c \oplus S \supseteq X_2^c \oplus S \\
\Rightarrow & (X_1^c \oplus S)^c \subseteq (X_2^c \oplus S)^c \\
\Rightarrow & X_1 \ominus S \subseteq X_2 \ominus S
\end{aligned} \tag{A.6}$$

and a similar proof for the reflected erosion $X \check{\ominus} S$. It follows immediately that

$$\begin{aligned}
& X_1 \subseteq X_2 \\
\Rightarrow & X_1 \check{\ominus} S \subseteq X_2 \check{\ominus} S \\
\Rightarrow & (X_1 \check{\ominus} S) \ominus S \subseteq (X_2 \check{\ominus} S) \ominus S \\
\Rightarrow & X_1^S \subseteq X_2^S
\end{aligned} \tag{A.7}$$

A.2 X^S is anti-extensive

Let us suppose that X is not a subset of X^S . Then, there exists a pixel \mathbf{x} such that $\mathbf{x} \in X$ and $\mathbf{x} \in (X^S)^c$. We consider the pixels \mathbf{y} that do not belong to $X \check{\ominus} S$, i.e. the set $Y = (X \check{\ominus} S)^c$. By the definition of $X \check{\ominus} S$ (30), $\mathbf{x} \in X$ implies that

$$\forall \mathbf{y} \in Y, \|\mathbf{x} - \mathbf{y}\| \geq S(\mathbf{y}) \tag{A.8}$$

On the other hand, we have $\mathbf{x} \in (X^S)^c = Y \oplus S$. By the definition of the adaptive dilation (9), this means that

$$\exists \mathbf{y} \in Y, \|\mathbf{x} - \mathbf{y}\| < S(\mathbf{y}) \tag{A.9}$$

Since this contradicts (A.8), no pixel \mathbf{x} can belong to both X and $(X^S)^c$, and

therefore $X \subseteq X^S$.

A.3 X^S is idempotent

In order to prove idempotence, we first notice that since X^S is anti-extensive, we have

$$X^S \subseteq (X^S)^S \quad (\text{A.10})$$

It remains to be proven that

$$(X^S)^S \subseteq X^S \quad (\text{A.11})$$

$$(X^S \check{\oplus} S) \ominus S \subseteq (X \check{\oplus} S) \ominus S \quad (\text{A.12})$$

Since adaptable erosion is increasing, it is sufficient to prove that

$$X^S \check{\oplus} S \subseteq X \check{\oplus} S \quad (\text{A.13})$$

Let us proceed ab absurdo. Suppose there is a pixel $\mathbf{y} \in (X^S \check{\oplus} S)$ and $\mathbf{y} \in (X \check{\oplus} S)^c$. By the definition of the reflected dilation (30), $\mathbf{y} \in (X^S \check{\oplus} S)$ means that

$$D_{X^S}(\mathbf{y}) = \min_{\mathbf{x} \in X^S} (\|\mathbf{x} - \mathbf{y}\|) < S(\mathbf{y}) \quad (\text{A.14})$$

or in other words,

$$\exists \mathbf{x} \in X^S : \|\mathbf{x} - \mathbf{y}\| < S(\mathbf{y}) \quad (\text{A.15})$$

On the other hand, according to the definition of the adaptable dilation (9), since $(X^S)^c = (X \overset{\check{}}{\oplus} S)^c \oplus S$, $\mathbf{y} \in (X \overset{\check{}}{\oplus} S)^c$ means that

$$\forall \mathbf{x} \in X^S : \|\mathbf{x} - \mathbf{y}\| \geq S(\mathbf{y}) \quad (\text{A.16})$$

Since this contradicts (A.15), no pixel \mathbf{y} can belong to both $(X^S \overset{\check{}}{\oplus} S)$ and $(X \overset{\check{}}{\oplus} S)^c$, we have $X^S \overset{\check{}}{\oplus} S \subseteq X \overset{\check{}}{\oplus} S$ and thus $(X^S)^S \subseteq X^S$.

B Polar morphology

In this annex we show that by specifying

$$S(\mathbf{p}) = \delta \cdot \|\mathbf{p}\| \quad (\text{B.1})$$

we get an operation that is equivalent to Roerdink's polar morphology. In [5], points are described by their polar coordinates (r, θ) , the group operation is

$$(r_1, \theta_1) * (r_2, \theta_2) = (r_1 \cdot r_2, \theta_1 + \theta_2) \quad (\text{B.2})$$

Roerdink considers dilations by structuring element B , a circle of center $(1, 0)$ and radius $\delta < 1$. If we consider an object $X = \{\mathbf{p}\}$ that consists of a single pixel \mathbf{p} , the dilation $X \oplus B$ is the disk of center \mathbf{p} and radius $\delta \cdot \|\mathbf{p}\|$. This is clearly equivalent to $X \oplus S$ with S defined above.

In [5], the reflected structuring element \check{B} used to define the closing

$$X^B = (X \oplus \check{B}) \ominus B \quad (\text{B.3})$$

is the disk \check{B} centered on $((1 - \delta^2)^{-1}, 0)$ of radius $\delta.(1 - \delta^2)^{-1}$. It results that, for the single pixel X above, the dilation $X \oplus \check{B}$ is the disk of center

$$\mathbf{c} = \frac{1}{1 - \delta^2} \mathbf{p} \quad (\text{B.4})$$

and radius

$$R = \frac{\delta}{1 - \delta^2} \|\mathbf{p}\| \quad (\text{B.5})$$

Let us consider the pixels q in this disk. They respect

$$\|\mathbf{q} - \mathbf{c}\| < R \quad (\text{B.6})$$

$$\|\mathbf{q} - \frac{1}{1 - \delta^2} \mathbf{p}\| < \frac{\delta}{1 - \delta^2} \|\mathbf{p}\| \quad (\text{B.7})$$

Expressed in cartesian coordinates, the square of this expression is

$$(q_x - \frac{1}{1 - \delta^2} p_x)^2 + (q_y - \frac{1}{1 - \delta^2} p_y)^2 < \frac{\delta^2}{(1 - \delta^2)^2} (p_x^2 + p_y^2) \quad (\text{B.8})$$

By expanding the squares and multiplying by $1 - \delta^2$, then grouping the terms in p_x^2 and p_y^2 , we get

$$\begin{aligned} (1 - \delta^2)q_x^2 - 2p_x q_x + p_x^2 + (1 - \delta^2)q_y^2 - 2p_y q_y + p_y^2 &< 0 \\ q_x^2 - 2p_x q_x + p_x^2 + q_y^2 - 2p_y q_y + p_y^2 &< \delta^2 q_x^2 + \delta^2 q_y^2 \\ (q_x - p_x)^2 + (q_y - p_y)^2 &< \delta^2 (q_x^2 + q_y^2) \\ \|\mathbf{q} - \mathbf{p}\| &< \delta \|\mathbf{q}\| \\ D_X(\mathbf{q}) &< S(\mathbf{q}) \end{aligned} \quad (\text{B.9})$$

This is clearly the definition of $X \check{\oplus} S$ with $S(\mathbf{p}) = \delta \cdot \|\mathbf{p}\|$. Having proved that both approaches are equivalent for an object X made of any single pixel \mathbf{p} in the image, we have proven the equivalence for all objects since we have

$$(X \cup Y) \oplus B = (X \oplus B) \cup (Y \oplus B) \quad (\text{B.10})$$

and similar properties for set \check{B} and operation $\check{\oplus}$.