LOCALLY ADAPTABLE MATHEMATICAL MORPHOLOGY

Olivier Cuisenaire

Signal Processing Institute
Ecole Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland

ABSTRACT

We investigate how common binary mathematical morphology operators can be adapted so that the size of the structuring element (SE) can vary across the image. We show that when the SE 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.

1. INTRODUCTION

Mathematical morphology [1] 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 translation invariance is theoretically useless and often unpractical. For instance, when analyzing traffic control camera images, 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. Roerdink has published several theoretical papers laying down the background for mathematical morphology that is not based on translation-invariant transformations of the Euclidean space. He defines polar morphology, 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 [2].

In this paper 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 allows us to propose efficient algorithms for the adaptable erosion, dilation, opening and closing, at a computational cost similar to the most efficient Euclidean mathematical morphology algorithms. Figure 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 why the algorithm reaches 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. Finally, section 6 discusses the computational cost of the method and 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 [3]. For instance, Ragnelmam [4] proposed an algorithm 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

$$B_d = \{\mathbf{h} : \|\mathbf{h}\| < d\} \tag{1}$$

The dilation operation is then defined as

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

The later expression uses the distance transformation D_X defined as

$$D_X(\mathbf{p}) = \min_{\mathbf{x} \in X}(\|\mathbf{p}-\mathbf{x}\|) \tag{3}$$

for which numerous efficient algorithms exist [5, 6]. 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 metric

$$\|\mathbf{v}\| = \sqrt{v_x^2 + v_y^2} \tag{4}$$

and therefore the balls B_d are circular. Nevertheless, other shapes of structuring elements can be obtained using different definitions of the distance such as the city-block, chessboard and chamfer metrics, or metrics defined on anisotropic grids, and the following developments still hold.

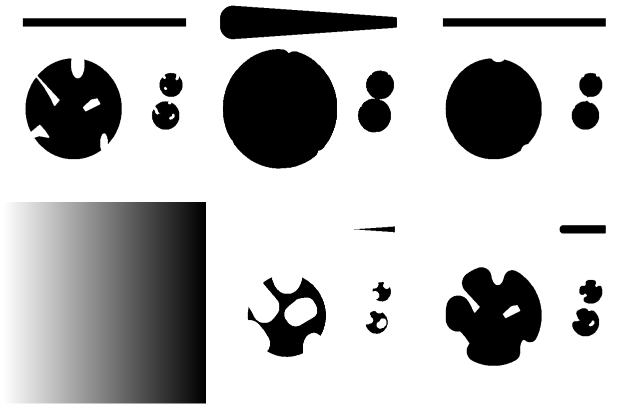

Fig. 1. **(top-left)** Original image I of size 512×512 with object X in black. **(bottom-left)** Structuring element image S chosen as $S(\mathbf{p}) = (512 - p_x)/14$. **(top-center)** Dilation $X \oplus S$. **(bottom-center)** Erosion $X \ominus S$ **(top-right)** Closing X^S **(bottom-right)** Opening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

3. DILATION AND EROSION

We want to extend the above method to allow the size of the ball used as structuring element to vary over the image. Instead of a single ball size d, we consider a whole image S of local structuring element sizes. Similarly to (2), we define the adaptable dilation of object X by this image S of SE sizes as

$$X \oplus S = \{\mathbf{x} + \mathbf{h} : \mathbf{x} \in X, \|\mathbf{h}\| < S(\mathbf{x})\} \tag{5}$$

It can be implemented as efficiently as before using a modified distance measure. Indeed, we have

$$\begin{aligned} X \oplus S &= \{\mathbf{x} + \mathbf{h} : \mathbf{x} \in X, \|\mathbf{h}\| - S(\mathbf{x}) < 0\} \\ &= \{\mathbf{y} : \exists\, \mathbf{x} \in X, \|\mathbf{y} - \mathbf{x}\| - S(\mathbf{x}) < 0\} \\ &= \{\mathbf{y} : D_{X,S}(\mathbf{y}) < 0\} \end{aligned} \tag{6}$$

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

$$D_{X,S}(\mathbf{p}) = \min_{\mathbf{x} \in X}(\|\mathbf{p} - \mathbf{x}\| - S(\mathbf{x})) \tag{7}$$

The algorithm to compute $D_{X,S}$ takes as inputs both the binary image I in which resides an object X, and the image of structural element sizes S. It computes both $D_{X,S}$ and a vectorial image $\mathbf{V}$ such that $\mathbf{V}(\mathbf{p})$ is the object pixel that minimizes this expression, i.e.

$$\mathbf{V}(\mathbf{p}) = \arg\min_{\mathbf{x} \in X}(\|\mathbf{p} - \mathbf{x}\| - S(\mathbf{x})) \tag{8}$$

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

First, we initialize 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.

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 $p_y = 1 \to N$ **do**
  for $p_x = 1 \to M$ **do**
    check($(p_x, p_y), (-1, 0)$); check($(p_x, p_y), (0, -1)$)
  for $p_x = M \to 1$ **do**
    check($(p_x, p_y), (1, 0)$)
for $p_y = N \to 1$ **do**
  for $p_x = M \to 1$ **do**
    check($(p_x, p_y), (1, 0)$); check($(p_x, p_y), (0, 1)$)
  for $p_x = 1 \to M$ **do**
    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 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 $X \oplus S$

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

$$\|\mathbf{p} - \mathbf{V}(\mathbf{p} + \mathbf{n})\| - S(\mathbf{V}(\mathbf{p} + \mathbf{n})) < D_{X,S}(\mathbf{p}) \tag{9}$$

This propagation is implemented similarly to the 4SED Euclidean DT algorithm of Danielsson [7] 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

$$X \ominus S = (X^c \oplus S)^c \tag{10}$$

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 [7] followed by a threshold. The rationale for Danielsson's algorithm is that while the definition (3) of the DT 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

$$T(\mathbf{x}) = \{\mathbf{p} : \mathbf{V}(\mathbf{p}) = \mathbf{x}\} \tag{11}$$

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.

In the general case, the tiles are more complex, but we can prove a property that is sufficient for the propagation to reach all pixels in the tiles: the tiles are star-shaped, i.e.

$$\mathbf{p} \in T(\mathbf{x}) \Rightarrow \forall \alpha \in [0,1], \mathbf{x} + \alpha.(\mathbf{p} - \mathbf{x}) \in T(\mathbf{x}) \qquad (12)$$

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

$$\|\mathbf{p} - \mathbf{x}\| - S(\mathbf{x}) \leq \|\mathbf{p} - \mathbf{y}\| - S(\mathbf{y}) \qquad (13)$$

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

$$\|\mathbf{p} - \mathbf{y}\| \leq \|\mathbf{p} - \mathbf{q}\| + \|\mathbf{q} - \mathbf{y}\| \qquad (14)$$

The definition of $\mathbf{q}$ also gives

$$\begin{aligned} \|\mathbf{p} - \mathbf{q}\| &= \|\mathbf{p} - \mathbf{x} - \alpha.(\mathbf{p} - \mathbf{x})\| \\ &= (1-\alpha).\|\mathbf{p} - \mathbf{x}\| && (15) \\ \|\mathbf{q} - \mathbf{x}\| &= \alpha.\|\mathbf{p} - \mathbf{x}\| \\ &= \|\mathbf{p} - \mathbf{x}\| - \|\mathbf{p} - \mathbf{q}\| && (16) \end{aligned}$$

which are valid for $0 \leq \alpha \leq 1$. By adding (13) and (14), then using (16), we get

$$\begin{aligned} \|\mathbf{p} - \mathbf{x}\| - \|\mathbf{p} - \mathbf{q}\| - S(\mathbf{x}) &\leq \|\mathbf{q} - \mathbf{y}\| - S(\mathbf{y}) \\ \|\mathbf{q} - \mathbf{x}\| - S(\mathbf{x}) &\leq \|\mathbf{q} - \mathbf{y}\| - S(\mathbf{y}) && (17) \end{aligned}$$

which is valid for any object pixel $\mathbf{y}$, and therefore we have $\mathbf{q} \in T(\mathbf{x})$. On a continuous image plane, this property would guarantee that there is a propagation path from $\mathbf{x}$ to all the pixels in $T(\mathbf{x})$. Unfortunately, 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 [7] does with the 8SED algorithm.

5. OPENING AND CLOSING

In Euclidean morphology [8], the closing X^B and opening X_B of an object X by a structuring element B are defined respectively as

$$\begin{aligned} X^B &= (X \oplus \check{B}) \ominus B && (18) \\ X_B &= (X \ominus \check{B}) \oplus B && (19) \end{aligned}$$

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 (5), 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

$$X \oplus \check{S} = \{\mathbf{x} + \mathbf{h} : \mathbf{x} \in X, \|\mathbf{h}\| < \check{S}(\mathbf{x})\} \qquad (20)$$

A naive approach would be to consider that since we work with circular balls, we can assume $\check{S} = S$ and define the closing X^S as

$$(X \oplus S) \ominus S \qquad (21)$$

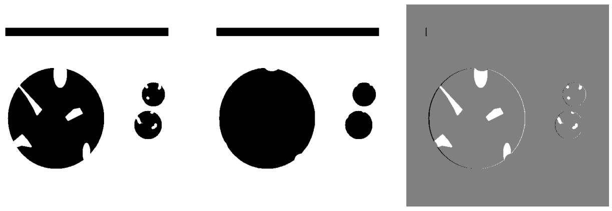

Fig. 2. **(left)** Original image with object X **(center)** Naive implementation of the closing: $(X \oplus S) \ominus S$. **(right)** $X \setminus ((X \oplus S) \ominus S)$. Black pixels belong to X but not $(X \oplus S) \ominus S$.

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

$$X_S \subseteq X \subseteq X^S \qquad (22)$$

as illustrated at Fig. 2. The problem with (21) 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. 2.

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

$$X \mathbin{\check{\oplus}} S = \{\mathbf{y} : \exists\, \mathbf{y} - \mathbf{h} \in X, \|\mathbf{h}\| < S(\mathbf{y})\} \qquad (23)$$

This expression is similar to (20), 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 $X \mathbin{\check{\oplus}} S$. From this, we can compute the closing as

$$X^S = (X \mathbin{\check{\oplus}} S) \ominus S \qquad (24)$$

Implementing this closing is straightforward once we notice that (23) can be written as

$$\begin{aligned} X \mathbin{\check{\oplus}} S &= \{\mathbf{y} : \exists\, \mathbf{y} - \mathbf{h} \in X, \|\mathbf{h}\| < S(\mathbf{y})\} \\ &= \{\mathbf{y} : \exists\, \mathbf{x} \in X, \|\mathbf{y} - \mathbf{x}\| < S(\mathbf{y})\} \\ &= \{\mathbf{y} : D_X(\mathbf{y}) < S(\mathbf{y})\} && (25) \end{aligned}$$

which uses the classical distance transform D_X defined as

$$D_X(\mathbf{p}) = \min_{\mathbf{x} \in X}(\|\mathbf{p} - \mathbf{x}\|) \qquad (26)$$

For city-block or chessboard metrics, this is easily computed with simple algorithms [9]. For the Euclidean metric, it can also be computed efficiently, as in [6] for instance. Finally, from $Y = X \mathbin{\check{\oplus}} S$, 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 as

$$X_S = ((X^c)^S)^c \qquad (27)$$

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, anti-extensive and idempotent Similarly, the opening X_S is increasing, extensive and idempotent. These properties are proven in [10].

Compute $D_X(\mathbf{p}) = \min_{\mathbf{x}\in X}(\|\mathbf{p} - \mathbf{x}\|)$ using the Euclidean DT algorithm in [6]

for all $\mathbf{p} \in I$ **do**
  if $D_X(\mathbf{p}) < S(\mathbf{p})$ **then**
    $\mathbf{p} \in Y = X \check{\oplus} S$
  else
    $\mathbf{p} \in Y^c = (X \check{\oplus} S)^c$

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

Algorithm 2: computes X^S

Let us note that the choice of the order of operations in algorithm 2 is arbitrary. Instead of computing the classical distance transform, comparing $D_X(\mathbf{p})$ to $S(\mathbf{p})$, then applying the adaptable erosion, one could apply an adaptable dilation, then compute the distance transform 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 the closing

$$X^{S*} = (X \oplus S) \check{\ominus} S \tag{28}$$

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

$$X \check{\ominus} S = (X^c \check{\oplus} S)^c \tag{29}$$

One can prove that this operation is also a closing, i.e. that it is increasing, anti-extensive and idempotent. Nevertheless, this alternative definition creates an operator of less practical interest.

6. DISCUSSION

The computational complexity of the algorithms presented here is 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. For openings and closings, the exact Euclidean distance transform [6] requires 3 scans of the image, and has a complexity similar to the dilation algorithm above. Initializations and set complementations add a negligible overhead.

The complexity is essentially independent of the size of the local structuring elements used. 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.

The methods and algorithms of this paper can be used as efficient implementation of several of Roerdink's mathematical morphologies on non Euclidean spaces [2]. 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 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 even 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 know 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.

7. 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.

8. REFERENCES

[1] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, New York, 1982.

[2] J.B.T.M Roerdink, "Group morphology," *Pattern Recognition*, vol. 33, pp. 877–895, 2000.

[3] E.R. Dougherty, *Mathematical Morphology in Image Processing*, Marcel Dekker, Inc., New York, 1992.

[4] I. Ragnelmam, "Fast erosion and dilation by contour processing and thresholding of distance maps," *Pattern Recognition Letters*, vol. 13, pp. 161–166, 1992.

[5] 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.

[6] O. Cuisenaire and B. Macq, "Fast and exact signed euclidean distance transformation with linear complexity," in *Proc. ICASSP'99*, Phoenix (AZ), March 1999, vol. 6, pp. 3293–3296.

[7] P.E. Danielsson, "Euclidean distance mapping," *Computer Graphics and Image Processing*, vol. 14, pp. 227–248, 1980.

[8] J.B.T.M Roerdink and H.J.A.M. Heijmans, "Mathematical morphology for structures without translation symmetry," *Signal Processing*, vol. 15, no. 3, pp. 271–277, 1988.

[9] A. Rosenfeld and J.L. Pfaltz, "Distance functions on digital pictures," *Pattern Recognition*, vol. 1, no. 1, pp. 33–61, 1968.

[10] O. Cuisenaire, "Locally adaptable binary mathematical morphology using distance transformations," Technical Report ITS-2004.30, Signal Processing Institute, EPFL, CH-1015 Lausanne, Switzerland, December 2004.