



## Homotopy Semilattices

Renato Keshet  
HP Laboratories Israel  
HPL-2004-1  
January 2<sup>nd</sup>, 2004\*

E-mail: renato@hpli.hpl.hp.com

mathematical  
morphology,  
complete  
inf-semilattices,  
erosion, opening,  
topographic  
distance,  
reconstruction,  
fillhole

A new, quasi-self-dual approach for morphological image processing is introduced. This approach is based on a complete semilattice framework. The related morphological erosion, for instance, shrinks connected components in an image, regardless to whether they are bright or dark.

In the binary case, the morphological operators are based on a complete inf-semilattice that is related to the homotopy tree. Two grayscale generalizations are investigated. The first is based on the topographic-distance concept, whereas the second one utilizes the reconstruction-based fillhole operation.

# 1 Introduction

Morphological operators are very efficient for image analysis and processing. One classical disadvantage of most of these operators, however, is the fact that they are usually not self-dual. As a consequence, bright and dark “objects” are usually not treated similarly, which is often an undesirable feature.

Consider the problem of denoising the binary image in Fig. 1(a). A simple erosion operation (as a first step of an opening) does remove the positive component of the noise, but the negative component is actually dilated, as seen in Fig. 1(b). The standard approaches of opening-closing or closing-opening [shown in Figs. 1(c) and 1(d), respectively] also are less than satisfactory in many situations.

Consider the binary image in Fig. 2(a) as well, where one can see the outline of blood cells, and suppose one is interested in performing granulometry analysis of these cells. Here also simple erosions fail; the external part of the cells do get smaller, but the internal parts are dilated, interfering with the process [see Figs. 2(b) and 2(c)].

We would like to be able to design a sound morphological operator that responds satisfactorily in the above situations.

The study of self-dual morphological operators is somewhat limited. See for instance the works of Serra, [1, chapter 8], Heijmans in [2], and Mehnert & Jackway in [3]. These operators are hard to analyze because they lack important morphological properties such as extensivity or anti-extensivity, and distributivity w.r.t. the infimum and the supremum.

Another approach is based on the extension of mathematical morphology from complete lattices to complete semilattices [4, 5]. In certain complete inf-semilattices, it was shown to be possible to define erosions that are self-dual, and therefore anti-extensive, increasing, and distributive w.r.t. the infimum. Moreover, they have an adjoint dilation and an associated (self-dual) opening operator. The problem with the particular semilattices that were defined in [4, 5], however, is the fact that they assume the existence of a “reference” signal, which is often hard to produce.

This work proposes a novel, quasi-self-dual morphological image processing approach that is based on a new non-reference semilattice. First, the idea is developed in the context of binary images only. Then, we generalize the approach to grayscale images in two different, independent ways.

## 2 Binary Homotopy Scheme

In this section, we derive our proposed scheme for binary images. The basic idea here is to use the data in the *homotopy tree*. Homotopy tree was defined by Serra in [1, page 89,

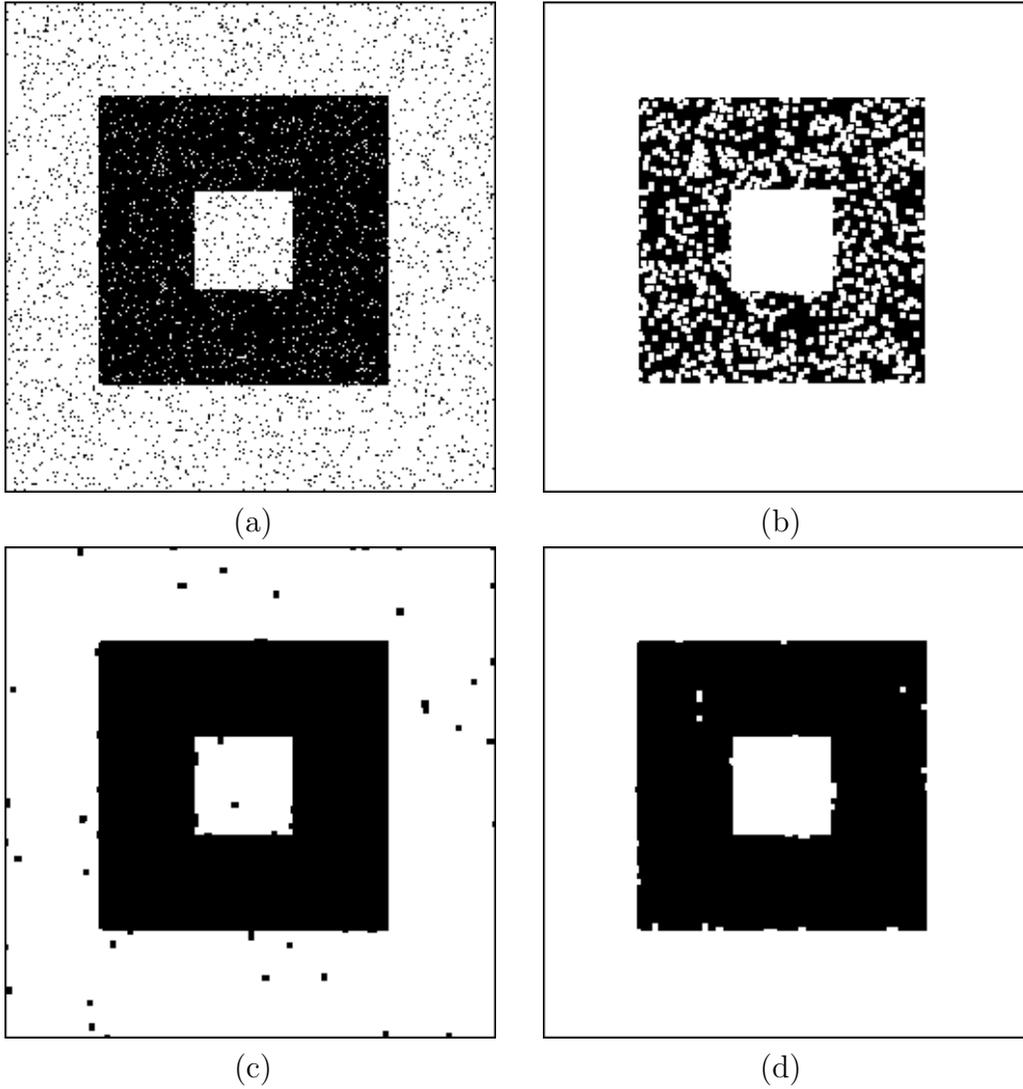


Figure 1: (a) A noisy binary image—the foreground is assumed black and the background white, (b) erosion by a  $3 \times 3$  squared structuring element, (c) the result of opening-closing with the same s.e., and (d) the result of closing-opening with the same s.e.

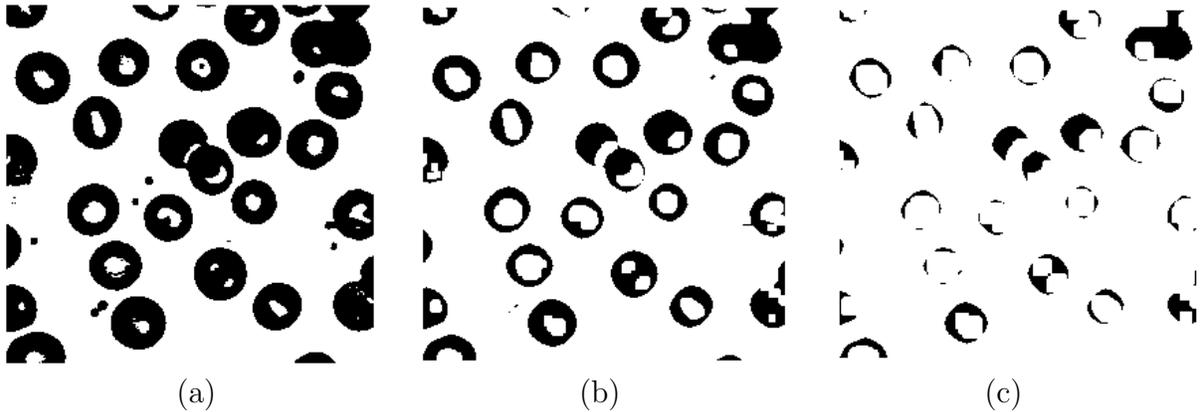


Figure 2: (a) A binary image showing the contours of blood cells—here again, the foreground is black and the background white, (b) erosion by a  $5 \times 5$  squared structuring element, (c) a further erosion step. One can see that the internal parts dilate, interfering with the erosion of the cell contours.

Figure III.10]. If  $X$  is a bounded input binary image in an Euclidean space  $E$ , then the root of the homotopy tree is the infinite connected component of  $X^c$ . The first level nodes of the tree are those connected components of  $X$  that are adjacent to the root. The second level of nodes are the connected components of  $X^c$  that are adjacent to the first level of nodes, and so on. See an example in Figs. 3(a) and (b).

## 2.1 The Binary Homotopy Algorithm

Let us associate a grayscale function  $f_X$  to the homotopy tree of a given binary image  $X$  as follows. The value  $f_X(x)$  of every pixel in  $x \in E$  is the level of the connected component that it belongs to in the homotopy tree. See Fig. 3(c), for instance.

We refer to the mapping  $\mathcal{H} : X \mapsto f_X$  as the *homotopy transform*. The inverse mapping  $\mathcal{H}^{-1} : f \mapsto X$  is given by

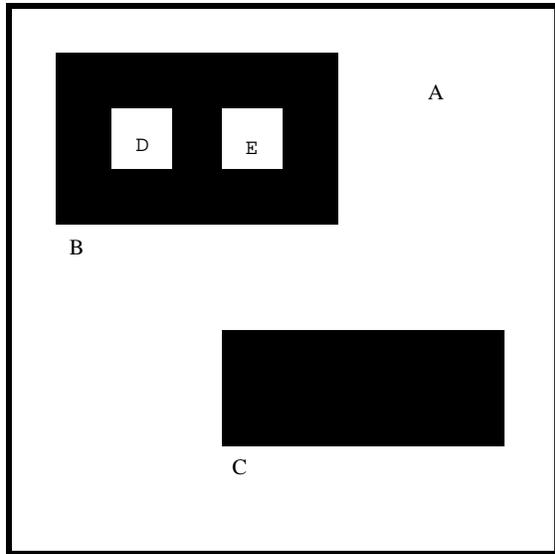
$$X = \{x \in E \mid \text{mod}(f(x), 2) = 1\}, \quad (1)$$

where  $\text{mod}(\cdot, 2)$  is the base-2 module. The homotopy transforms of the binary images in Fig. 1(a) and Fig. 2(a) are shown in Fig. 4(a) and 4(b), respectively.

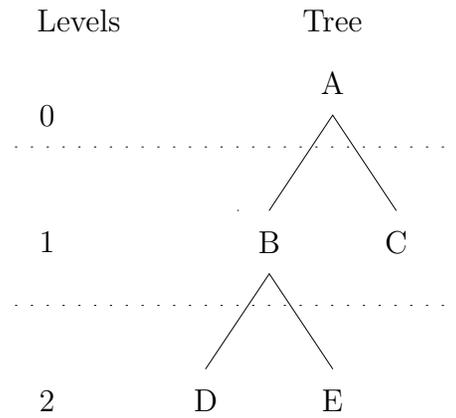
Not every grayscale function  $f$  is the homotopy transform of some binary image. However, it can be shown that if  $f_X = \mathcal{H}\{X\}$ , and  $b$  is a flat structuring element, then there exists a binary image  $Y$  such that  $\mathcal{H}\{Y\} = f_X \ominus b$ .

Now, consider the following algorithm:

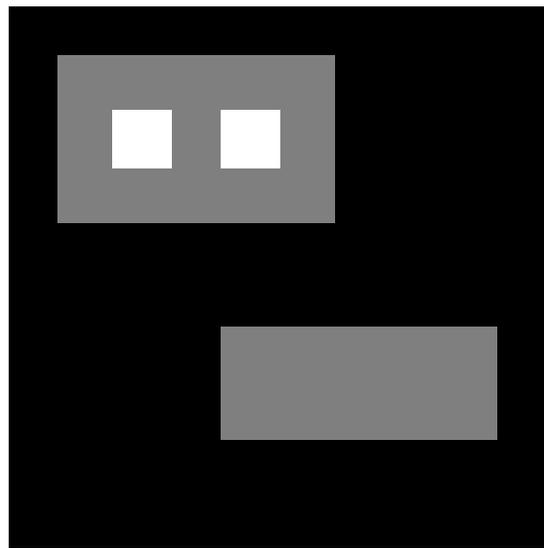
1. Let  $X$  be an input binary image, and calculate  $f_X = \mathcal{H}\{X\}$ .



(a)



(b)



(c)

Figure 3: (a) A binary image. Each letter corresponds to a connected component. A is the background component, B and C correspond to the two black connected components, and D and E correspond to the two white connected components inside B. (b) the homotopy tree of (a), and (c) its homotopy transform.

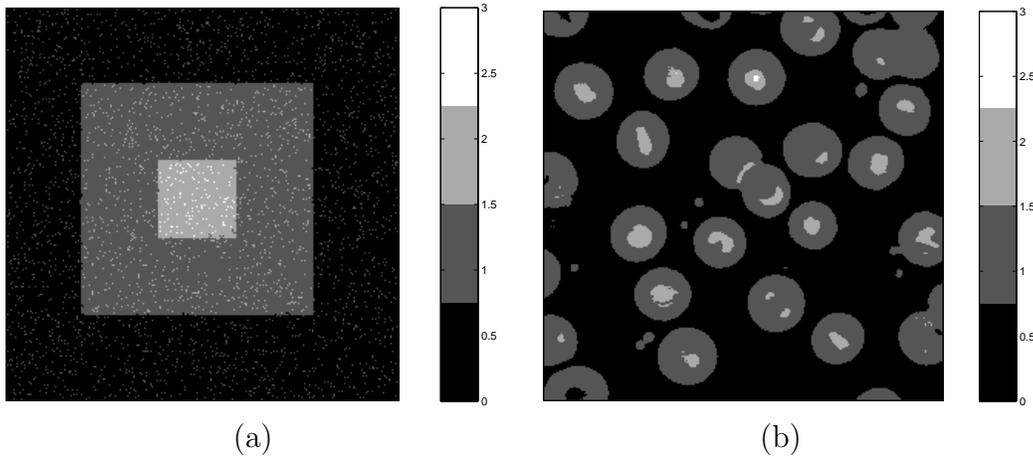


Figure 4: (a) and (b): The homotopy transforms of Fig. 1(a) and Fig. 2(a), respectively.

2. Erode  $f_X$  with a flat structuring element  $b$  to obtain a new function  $g \triangleq f_X \ominus b$ ,
3. Apply the inverse homotopy transform to  $g$  to obtain a binary output image  $Y = \mathcal{H}^{-1}\{g\}$ .

Running the above algorithm on Fig. 1(a) yields the results seen in Fig. 5(a). All noise (except on the edges) has been removed because the negative noise of the binary input is “unfolded” in the homotopy transform, and becomes positive, thus being removed by the grayscale erosion. The result of the algorithm on Fig. 2(a), for the  $5 \times 5$  and  $9 \times 9$  structuring elements, are shown in Fig. 5(b) and (c), respectively. Notice that the internal components of the cells are also eroded.

Judging from the above examples, this algorithm provides a useful, quasi-self-dual operation. However, it unfortunately does not possess some desired morphological properties: It is not distributive w.r.t. the intersection, and therefore it is not an erosion in the boolean lattice of binary images. If it is not an erosion, then theoretically we cannot speak of an adjoint dilation and an associated opening.

However, is there another framework (perhaps a semilattice) where the above algorithm does represent an erosion? The answer is yes, as seen next.

## 2.2 Binary Homotopy Semilattice

Define the following relation, for all  $X, Y \subseteq E$ :

$$X \preceq Y \iff f_X(x) \leq f_Y(x), \forall x \in E, \quad (2)$$

where  $f_X$  and  $f_Y$  are the homotopy transforms of  $X$  and  $Y$ , respectively.

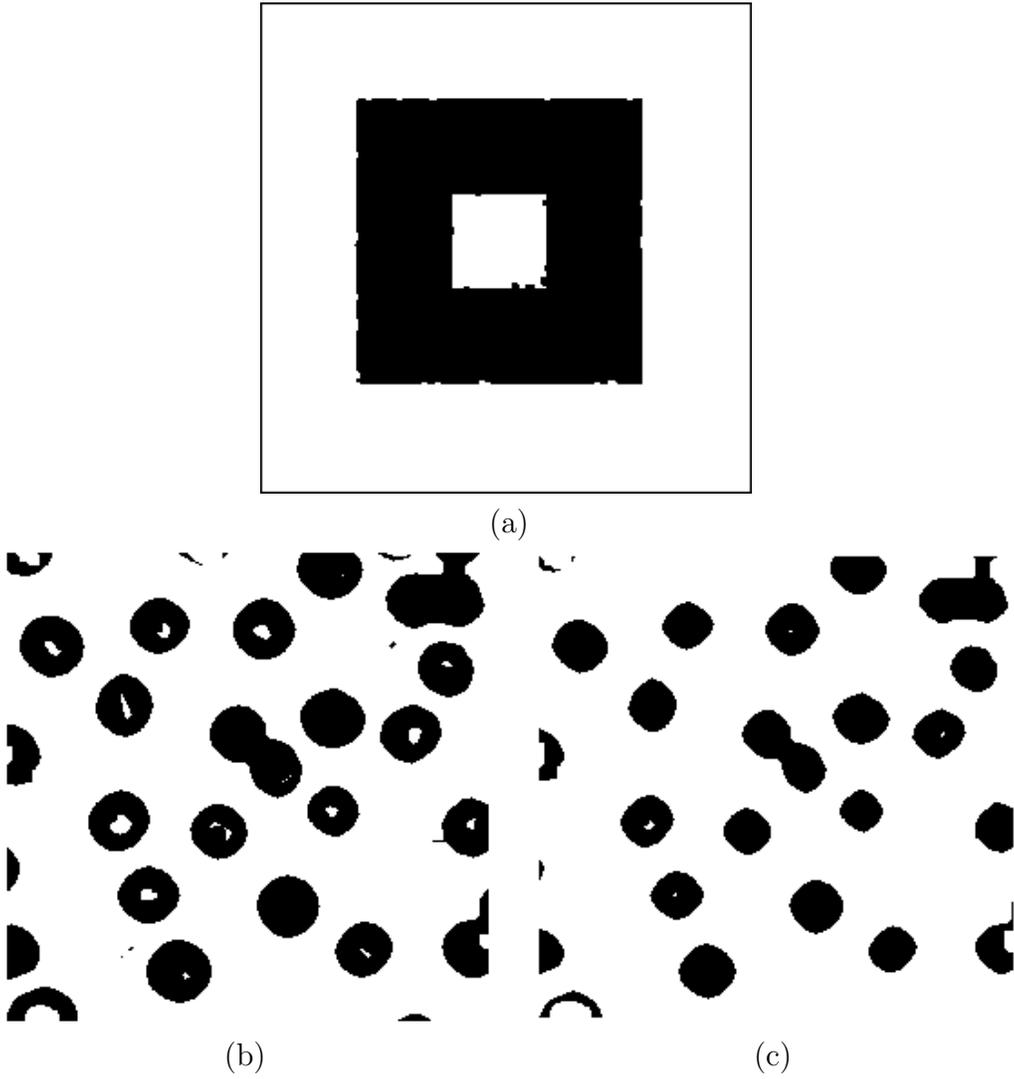


Figure 5: Results of running the binary homotopy algorithm on the test images. (a) Output of the algorithm on Fig. 1(a) with  $3 \times 3$  s.e., (b) and (c) outputs of the algorithm on Fig. 2(a) with  $5 \times 5$  and  $9 \times 9$  structuring elements, respectively.

The relation  $\preceq$  defined in (2) is a partial ordering, and  $(\mathcal{P}(E), \preceq)$  is a complete inf-semilattice<sup>1</sup>, with the infimum of an arbitrary collection of binary images  $\{X_i\}$  being given by:

$$\bigwedge_i X_i = \mathcal{H}^{-1} \left\{ \inf_i f_{X_i} \right\}, \quad (3)$$

where  $\mathcal{H}^{-1}$  denotes inverse homotopy transform, and  $\inf$  is the point-wise infimum operator.

One can show that every pair of binary images does have a supremum w.r.t.  $\preceq$ , and therefore  $(\mathcal{P}(E), \preceq)$  is a lattice. However, the existence of a supremum for an infinite set of binary images is not assured (in particular, there does not exist a greatest binary image w.r.t.  $\preceq$ ), and therefore  $(\mathcal{P}(E), \preceq)$  is not a *complete* lattice.

Let  $B$  be a structuring element in  $E$ . The operator

$$\varepsilon_B(X) \triangleq \bigwedge_{y \in B^s} X_y \quad (4)$$

is an erosion in  $(\mathcal{P}(E), \preceq)$ . Here,  $X_y$  denotes the translation of  $X$  to the point  $y$ :  $X_y = \{x + y \mid x \in X\}$ .

One can verify that, when the flat structuring element  $b$  has the shape of the binary structuring element  $B$ , then the binary homotopy algorithm in Section 2.1 returns the same result as  $\varepsilon_B(X)$ . The conclusion is that the binary homotopy algorithm is an erosion in  $(\mathcal{P}(E), \preceq)$ .

According to the mathematical morphology theory on semilattices [4],  $\varepsilon_B$  is adjoint to a unique dilation  $\delta_B$ , and the associate opening is given by  $\gamma_B = \delta_B \varepsilon_B$ . The adjoint dilation  $\delta_B$  is given by:

$$\delta_B(X) = \bigvee_{y \in B} X_y = \mathcal{H}^{-1} \left\{ \sup_{y \in B} f_{X_y} \right\}. \quad (5)$$

Opening results on the test images are shown in Fig. 6.

### 3 Grayscale Scheme—Boundary Topographic Distance

In this section and in Section 4, we address the issue of generalizing the binary homotopy operators to grayscale images. This is not a trivial task, since the homotopy tree has been defined only for binary images.

Two grayscale generalizations of the binary homotopy scheme are developed in this article. In this section, an approach based on a boundary topographic distance transform is addressed. In Section 4, a reconstruction-based scheme is proposed.

---

<sup>1</sup>The set  $\mathcal{P}(E)$  is called the power set of  $E$  and consists of the collection of all subsets of  $E$ .

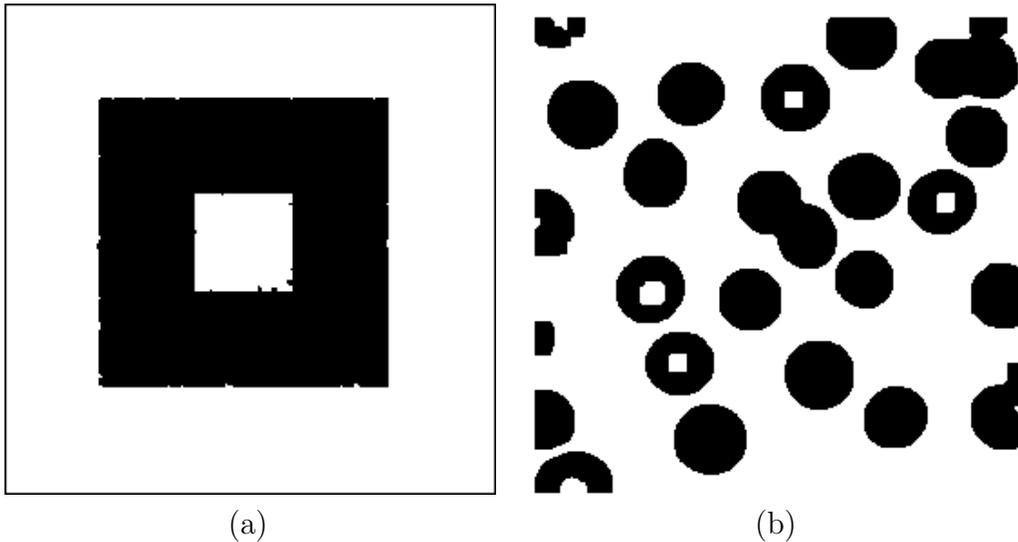


Figure 6: Opening  $\gamma_B$  of (a) Fig. 1(a) with  $3 \times 3$  s.e., and (b) Fig. 2(a) with  $9 \times 9$  s.e.

The basic idea behind the boundary topographic distance scheme, developed in this section, is based on the following observation: The value of  $f_X(x)$  is equal to the smallest number of edges that one has to cross in order to move from  $x$  to the boundary of the image on a connected path. For instance, consider a point  $x$  inside the connected component  $D$  in Fig. 3(a). In order to move from  $x$  to the boundary of the image, one has to cross at least two edges—the interface between  $D$  and  $B$  and the interface between  $B$  and  $A$ . And indeed  $f_X(x) = 2$ .

### 3.1 Boundary Topographic Distance (BTD)

It turns out that the above description is highly related to that of a *topographic distance function* w.r.t. the boundary.

Topographic distances were defined by Meyer in [6]. We utilize a simplified definition of topographic distance in this article<sup>2</sup>. The (simplified) topographic distance between two pixels  $x$  and  $y$  of an image  $f$  is the least total variation needed to walk from  $x$  to  $y$  (or vice-versa) on the topographic relief defined by  $f$ . For instance, suppose that a connected path between  $x$  and  $y$  has the following values on  $f$ :  $f(x = p_1, p_2, \dots, p_6, p_7 = y) = (3, 5, 7, 4, 2, 3, 5)$ . The total variation on that path is given by the sum of absolute differences between consecutive values, which is equal to 12 in this case. If no other connected path linking  $x$  to  $y$  has smaller total variation, then 12 is the topographic distance between  $x$  and  $y$ .

We define *boundary topographic distance* (BTD) of a pixel  $x$  w.r.t. a *bounded* image  $f$  as the

<sup>2</sup>We assume the cost of walking on a topographic function  $f$  between two consecutive points  $p_1$  and  $p_2$  to be simply equal to  $|f(p_1) - f(p_2)|$ .

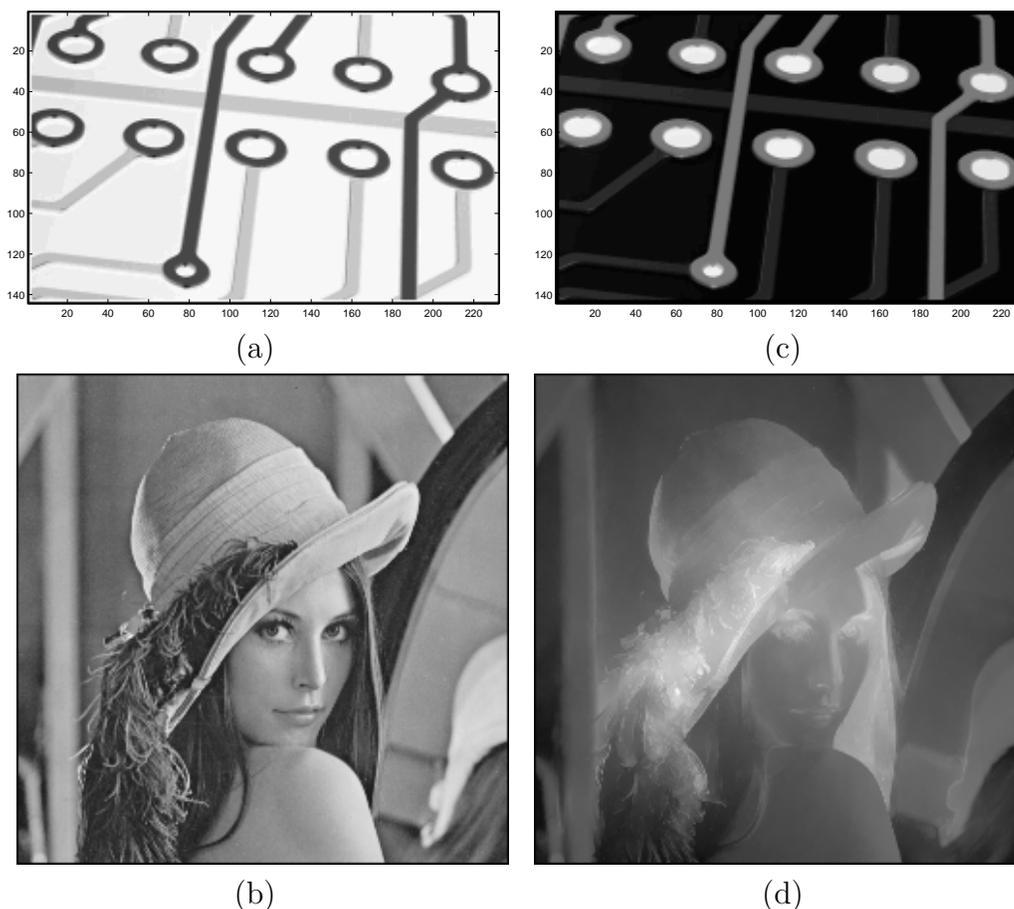


Figure 7: (a) 5-bit grayscale image, and (b) a 8-bit grayscale image. (c)-(d) The boundary topographic distance (BTD) functions of (a) and (b), respectively.

topographic distance (on the topography of  $f$ ) between  $x$  and the boundary of  $f$ . That is, the BTD of  $x$  is the least topographic distance between  $x$  and any point on the boundary of  $f$ . The BTD function,  $BT_f$ , is the mapping from each pixel  $x$  to its BTD,  $BT_f(x)$ .

It turns out that the BTD function associated to a *binary* image  $X$  is exactly its homotopy transform  $f_X$ . Happily, the BTD function applies also to grayscale images, and therefore we can consider it a generalization of the homotopy transform. Fig. 7 shows a couple of test images and their BTD functions.

It would make sense then to generalize the binary homotopy algorithm, given on page 5, by just replacing homotopy and inverse homotopy transforms by BTD function and inverse BTD function calculations, respectively. However, there does not exist an inverse BTD function operator. In fact, there are many different grayscale images that may have the same BTD function  $BT_f$  as a given image  $f$  (for instance, the  $BT_f$  image itself).

The problem is that, differently from the binary case, the BTD does not retain the “unfolding” pattern, needed to “fold” a processed BTD function back to the original domain. A mechanism for retaining this pattern is needed, and one is developed next.

### 3.2 Boundary-Topographic-Variation (BTV) Transform

Let us denote by  $\pi_f(x)$  a connected path linking the boundary of an image  $f$  to a pixel  $x$ . Let  $\hat{\pi}_f(x)$  be one such path, with least topographic distance between  $x$  and the boundary. I.e., the topographic distance (total variation) on  $\hat{\pi}_f(x)$  is exactly  $BT_f(x)$ . We call  $\hat{\pi}_f(x)$  a minimal-BTD path for  $x$ . Given a pixel  $x$ , there may be more than one minimal-BTD path for  $x$ .

We associate to a minimal-BTD path an alternating sequence that describes the “ups and downs” that occur on that path. We call these “ups and downs” *path variation*. For instance, consider a grayscale image  $f$ , and suppose that a minimal-BTD path  $\hat{\pi}_f(x)$  from the boundary to a pixel  $x$  has the following function values:

$$f(\hat{\pi}_f(x)) = (0, 3, 4, 7, 3, 1, 5, 6, 6, 2, 7). \quad (6)$$

The corresponding path variation is given by:

$$V(\hat{\pi}_f(x)) = \{7, -6, 5, -4, 5\}, \quad (7)$$

meaning that one has to climb 7 gray levels [to follow the monotonic sub-sequence (0, 3, 4, 7)], then go down 6 [to follow the next monotonic sub-sequence (7, 3, 1)], and so on for the whole path.

If we assume that  $f(x) \geq 0, \forall x$ , and that the boundary of  $f$  has null values, then the first element of a path variation is always non-negative.

We call the mapping  $\mathcal{V} : f \mapsto V_f$ , where  $V_f(x) \triangleq V(\hat{\pi}_f(x))$ , the *boundary-topographic-variation (BTV) transform*. Because it is based on the topographic distance, the BTV transform can be calculated fast, by modifying a fast (topographic-distance-based) implementation of the watershed algorithm.

As we noted before, the minimal boundary path is not necessarily unique, which means that there maybe more than one BTV transform for a given image  $f$ . In practice, different BTV transforms are identical up to a relatively small number of pixels, which we call *skeleton pixels* or *watershed pixels*.

Table 1 presents a simple 1-D case for illustration.

The original image  $f$  can be obtain back from  $V_f(x)$  as follows:

$$f(x) = \sum_i V_f(x)_i, \quad (8)$$

$x$	$f(x)$	$BT_f(x)$	$\hat{\pi}_x$	$[\mathcal{V}(f)](x)$
0	0	0	(0)	{0}
1	3	3	(0, 3)	{3}
2	5	5	(0, 3, 5)	{5}
3	4	6	(0, 3, 5, 4)	{5, -1}
4	9	11	(0, 3, 5, 4, 9)	{5, -1, 5}
5	13	15	(0, 3, 5, 4, 9, 13) (0, 6, 11, 10, 13)	{5, -1, 9} {11, -1, 3}
6	10	12	(0, 6, 11, 10)	{11, -1}
7	11	11	(0, 6, 11)	{11}
8	6	6	(0, 6)	{6}
9	0	0	(0)	{0}

Table 1: The BTV transform of a 1-D function  $f$ . The boundary here are the first and last elements of the function. Notice that the point  $x = 5$  is a skeleton (or watershed) point of the transform, with two different transform possibilities.

where  $\{V_f(x)_i\}$  are the elements of the alternating sequence  $V_f(x) = [\mathcal{V}(f)](x)$ . Equation (8) represents the inverse BTV transform,  $\mathcal{V}^{-1}$ .

It is also simple to obtain the BTD function from  $\mathcal{V}(f)$ :

$$BT_f(x) = \sum_i |V_f(x)_i|. \quad (9)$$

### 3.3 Semilattice in the BTV Domain

In order to define morphological operations in the BTV transform domain, let us define a complete inf-semilattice of variations (alternating sequences) by means of the following partial ordering. Let  $V_1$  and  $V_2$  be two alternating sequences with lengths  $L_1$  and  $L_2$ , respectively.

$$V_1 \sqsubseteq V_2 \iff \begin{cases} (V_1)_i = (V_2)_i, \quad \forall i < L_1, \\ |(V_1)_{L_1}| \leq |(V_2)_{L_1}|. \end{cases} \quad (10)$$

For instance,  $\{7, -3, 2\} \sqsubseteq \{7, -3, 4, -1\}$  but  $\{7, -3, 2\} \not\sqsubseteq \{7, -4, 1\}$ .

The infimum operation associated to the above partial ordering is the common prefix, followed by the the weakest of the next elements. More precisely,

$$V_1 \sqcap V_2 = \{P(V_1, V_2), \text{med}[(V_1)_{L_P+1}, (V_2)_{L_P+1}, 0]\}, \quad (11)$$

where  $P(V_1, V_2)$  is the common prefix of  $V_1$  and  $V_2$ ,  $L_P$  is the length of  $P(V_1, V_2)$ , and  $\text{med}$  is the median operation. For instance  $\{6, -1, 7\} \sqcap \{6, -1, 4, -2, 3\} = \{6, -1, 4\}$ . Also  $\{3, -1, 2, -3\} \sqcap \{3, -2, 6, -4\} = \{3, -1\}$  and  $\{4, -3, 6\} \sqcap \{2, -1, 4\} = \{2\}$ .

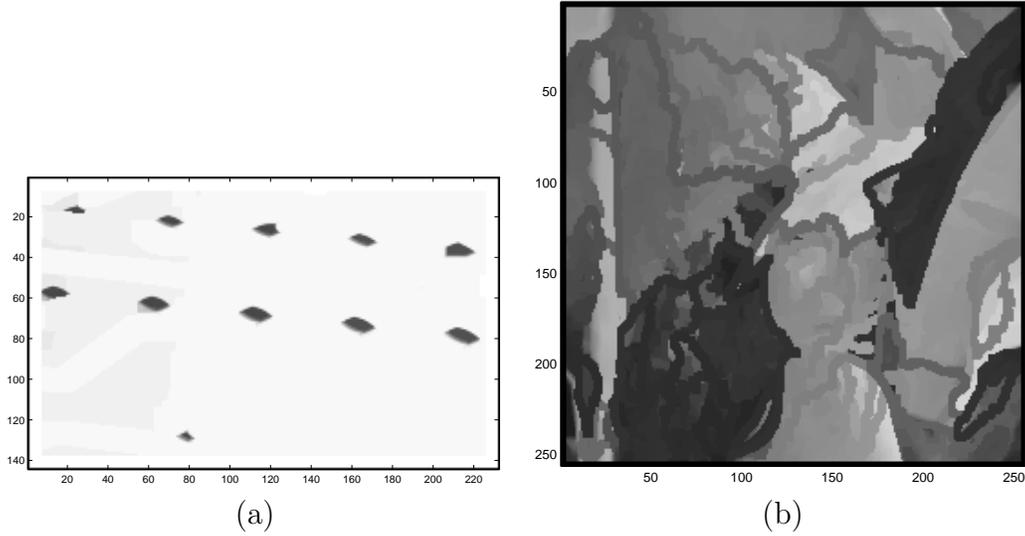


Figure 8: Results of BTV-based erosion  $\hat{\varepsilon}_B$ . (a) Erosion of Fig. 7(a) for a  $11 \times 11$  structuring element, and (b) erosion of Fig. 7(b) for a  $5 \times 5$  structuring element.

The following operator can now be defined:

$$\hat{\varepsilon}_B(f) = \mathcal{V}^{-1} \{ \sqcap_{y \in B^s} \mathcal{V}\{f_y\} \}, \quad (12)$$

where  $f_y$  denotes the translation of  $f$  by  $y$ :  $f_y(x) = f(x - y)$ . It can be shown that the above operator is an erosion in the complete inf-semilattices of BTV-transform images. The adjoint dilation is given by

$$\hat{\delta}_B(f) = \mathcal{V}^{-1} \{ \sqcup_{y \in B} \mathcal{V}\{f_y\} \}, \quad (13)$$

where  $\sqcup$  is the trivial supremum:

$$V_1 \sqcup V_2 = \begin{cases} V_1, & V_2 \subseteq V_1, \\ V_2, & V_1 \subseteq V_2, \\ \bar{A}, & \text{otherwise.} \end{cases} \quad (14)$$

Like for any erosion in a complete inf-semilattice, the opening operator  $\hat{\gamma}_B$  associated to  $\hat{\varepsilon}_B$  is well defined and given by  $\hat{\gamma}_B = \hat{\delta}_B \hat{\varepsilon}_B$ .

### 3.4 Results and Discussion

Fig. 8 shows the output of the erosion  $\hat{\varepsilon}_B$  on our test images.

Notice that the results on the 32-bit image [Fig. 8(a)] is satisfactory, where we see that the large sockets are eroded less than the lines, despite their holes.

However, the results on the 8-bit image is very strange; the output shows many “trenches” that do not have any justification to exist.

These are typical results. When the number of gray levels is small, and the image is not very complex, the BTV domain usually characterizes meaningfully the homotopic relationships in the image. As a result, the BTV-based erosion returns useful results. However, when the number of gray levels is high, or if the image is very complex, the BTV transform usually fails. The reason for that is the existence of skeleton (watershed) points in the BTV data, which have neighbors with potentially very dissimilar alternating sequences; when eroded, a “trench” may open on these points, because the infimum of the dissimilar alternating sequences yields a sequence that does not characterize any of the neighbors.

A solution to this problem is still being sought.

## 4 Grayscale Scheme—Fillhole

Here, a second grayscale generalization of the homotopy morphological operators is developed. The basic idea behind this second approach is based on the observation that the *threshold sets* [1, page 433] (see also [7]) of the homotopy transform  $f_X$  can be calculated with the help of the *reconstruction* operator [8, section 6.3]. More specifically, the thresholding can be computed using the *fillhole* algorithm  $\text{FILL}(\cdot)$  described in [8, section 6.4.2], which has the effect of closing all the holes in the input image. This is significant because grayscale erosion of  $f_X$  (which is the heart of the binary homotopy erosion) is equivalent to binary erosion of the threshold sets of  $f_X$ , and thus generalizing them will open the way to generalizing the whole approach.

### 4.1 Review of Fillhole and Thresholding

Mathematically, the fillhole operator is defined as follows:

$$\text{FILL}(f) = R_{f_m}^*(f), \quad (15)$$

where  $R_{f_m}^*(f)$  is the reconstruction by erosion (inverse reconstruction) of a grayscale image  $f$  from

$$f_m(x) \triangleq \begin{cases} f(x), & \text{if } x \text{ lies on the boundary of } f, \\ \infty, & \text{otherwise.} \end{cases} \quad (16)$$

The  $\text{FILL}(\cdot)$  function can be applied to binary images (subsets of the Euclidean space  $E$ ) as well, if we consider the input grayscale image to be the indicator function of the binary image (which assumes the value 1 for  $x \in X$  and 0 otherwise).

The threshold set  $T_n(f)$  of a function  $f$ , where  $n$  is a given threshold, is given by

$$T_n(f) \triangleq \{x \in E \mid f(x) \geq n\}. \quad (17)$$

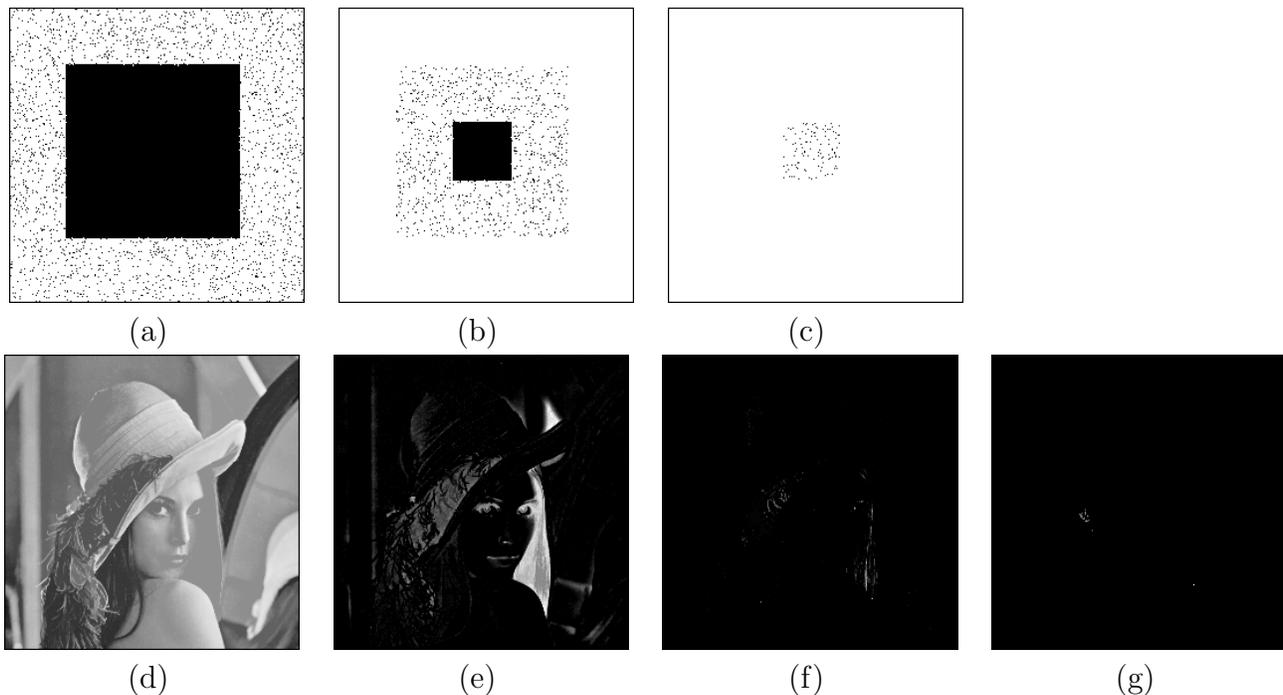


Figure 9: Thresholding and fillhole decomposition. (a)-(c) The sets  $\{T_n(f_X)\}$  associated to the image  $X$  in Fig. 1(a). Recall that background is assumed white. (d)-(g) The fillhole decomposition of the image in Fig. 7(b).

## 4.2 Fillhole Decomposition

As we stressed above, we can calculate the threshold sets of  $f_X$  using the FILL function. Indeed, if  $n \geq 1$  is a given threshold, then:

$$T_n(f_X) = \text{FILL}(Z_n), \quad (18)$$

where

$$\begin{cases} Z_1 \triangleq X, \\ Z_n \triangleq T_{n-1}(f_X) - Z_{n-1}. \end{cases} \quad (19)$$

Fig. 9(a)-(c) shows the sets  $\{T_n(f_X)\}$  associated to the image in Fig. 1(a).

Happily, the  $\text{FILL}(\cdot)$  operator is defined on grayscale images, and therefore, the above procedure can be generalized. Let  $f(x)$  be a given grayscale image, and define the set of images  $\{F_n(x)\}$  as follows:

$$F_n = \text{FILL}(z_n), \quad n = 1, 2, \dots \quad (20)$$

where

$$\begin{cases} z_1 \triangleq f, \\ z_n \triangleq F_{n-1} - z_{n-1}. \end{cases} \quad (21)$$

We call the mapping  $\mathcal{F} : f \mapsto \{F_n\}_{n \in \mathbb{Z}_+}$  the *fillhole decomposition*. The inverse mapping  $\mathcal{F}^{-1}$  is given by

$$f(x) = \sum_{n=1}^{\infty} (-1)^{n-1} F_n(x). \quad (22)$$

Fig. 9(d)-(g) show the fillhole decomposition of the test image in Fig. 7(b).

### 4.3 Fillhole Semilattice

Let us define the following image relation via fillhole decomposition:

$$f \preceq g \iff F_n(x) \leq G_n(x), \forall n \in \mathbb{Z}_+, \forall x \in E, \quad (23)$$

where  $\{F_n\} = \mathcal{F}(f)$  and  $\{G_n\} = \mathcal{F}(g)$ .

The above relation is a partial ordering that gives a complete inf-semilattice structure to the set of grayscale images. The infimum is given by:

$$f \wedge g = \mathcal{F}^{-1}\{\inf(F_n, G_n)\}_{n \in \mathbb{Z}_+}. \quad (24)$$

As in the binary case, the supremum of an infinite collection of images is not guaranteed to exist. Therefore,  $\preceq$  does not produce a complete semilattice.

We finally can define the following operator:

$$\tilde{\varepsilon}_B(f) \triangleq \bigwedge_{y \in B^s} f_y. \quad (25)$$

This operator is in fact an erosion in the above complete inf-semilattice, and it consists of a grayscale generalization of the binary homotopy erosion.

The adjoint dilation is given by

$$\tilde{\delta}_B(f) \triangleq \bigvee_{y \in B} f_y. \quad (26)$$

where

$$f \vee g = \mathcal{F}^{-1}\{\sup(F_n, G_n)\}. \quad (27)$$

As usual, the associated opening is  $\tilde{\gamma}_B = \tilde{\delta}_B \tilde{\varepsilon}_B$ .

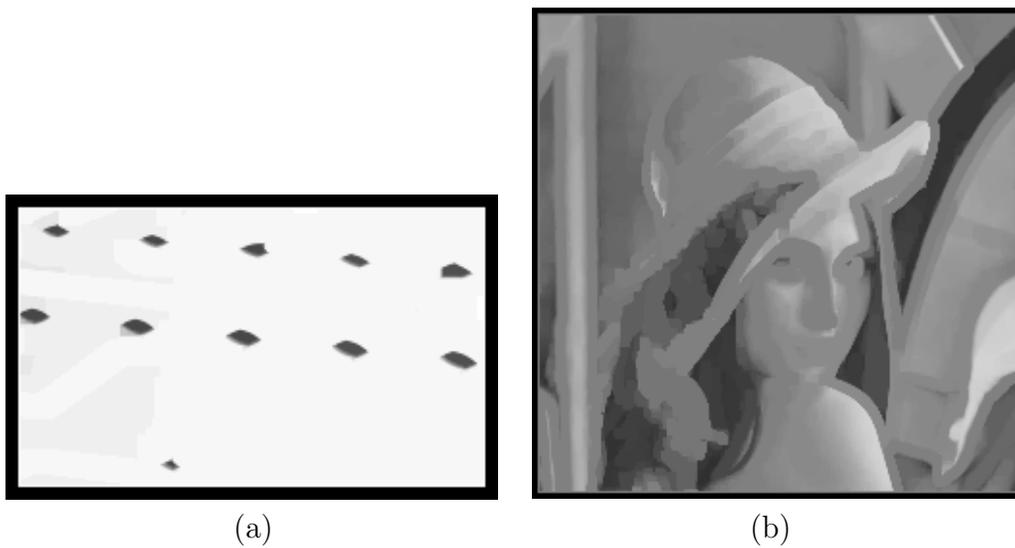


Figure 10: Results of fillhole-decomposition erosion  $\tilde{\epsilon}_B$ . (a) Erosion of Fig. 7(a) for a  $11 \times 11$  structuring element, and (b) erosion of Fig. 7(b) for a  $5 \times 5$  structuring element. For improved results, the value of the boundary pixels were set to 128 in the image in Fig. 7(b).

## 4.4 Results and Discussion

Fig. 10 shows the results of applying the fillhole-decomposition based erosion on the test images in Fig. 7. Notice that Fig. 10(a) is very similar to Fig. 8(a). However, the output in Fig. 10(b) is significantly more stable than that in Fig. 8(b). Notice that all “objects” in the image shrink, regardless to whether they are bright or dark. When bright and dark regions are adjacent, both shrink, and an “average” value fills the gap in between.

In Fig. 11, one can observe a series of fillhole-decomposition openings of Fig. 7(b). Notice that both bright and dark objects are removed at each iteration, while the main edges remain on their original place.

We should note that even though the fillhole operation is fast, various iterations are necessary in order to calculate the various components of the fillhole decomposition. This makes the fillhole-decomposition morphological operations not so fast as the BTV-based ones.

## 5 Conclusion

A new, quasi-self-similar approach for morphological image processing has been proposed.

Its binary version is based on the homotopy tree, and the resulting morphological erosion has the effect of shrinking all the elements of the image, regardless to whether they belong to foreground or background.

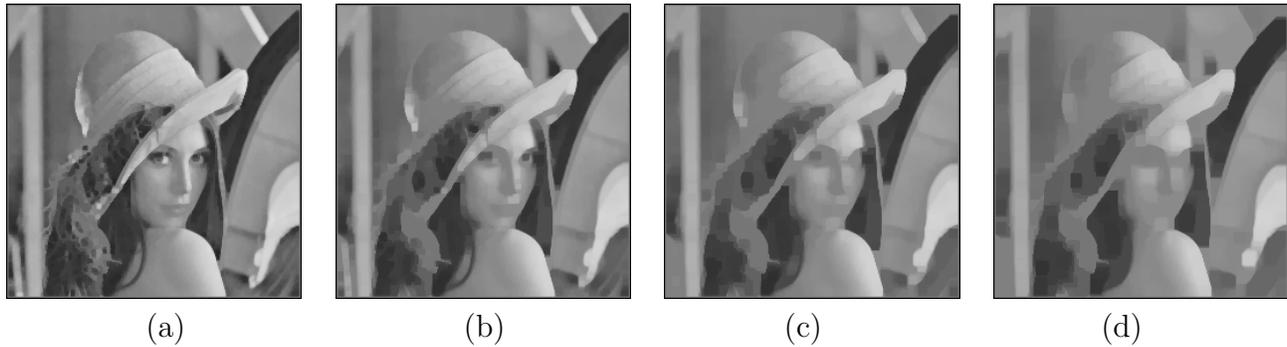


Figure 11: Granulometry of “Lena” using the fillhole-decomposition opening, with (a)  $3 \times 3$ , (b)  $5 \times 5$ , (c)  $7 \times 7$ , and (d)  $9 \times 9$  structuring elements.

Two different grayscale generalizations were investigated. The first one generalizes the homotopy transform to a boundary-topographic-distance (BTD) function. However, BTD alone is not enough, and alternating sequences that represent the minimal path variation w.r.t. the topographic distance is used instead. This representation is called the boundary-topographic-variation (BTV) transform. A complete inf-semilattice of alternating sequences was defined. The resulting morphological operations can be computed fast, but on high-bit or complex images they produce annoying “trench” artifacts.

The second grayscale generalization is based on the fillhole operation. Fillhole decomposition was shown to generalize the threshold sets of the homotopy transform, and it is used to define another complete inf-semilattice. The resulting morphological operations are conceptually simpler and easier to implement than the BTV-based ones, and are free of the “trench” artifacts. However, the BTV-based operators run significantly faster, and therefore they can be a good alternative for low-complexity, low-bit images.

## 6 References

- [1] J. Serra, *Image Analysis and Mathematical Morphology, Vol. 1*, London: Academic Press, 1982.
- [2] H.J.A.M. Heijmans, “Self-dual morphological operators and filters,” *Journal of Mathematical Imaging and Vision* 6, 1 (1996), pp. 15-36.
- [3] A.J.H. Mehnert and P.T. Jackway, “Folding induced self-dual filters,” *Mathematical Morphology and its Applications to Image and Signal Processing*, J. Goutsias, L. Vincent, and D.S. Bloomberg, Eds., Boston: Kluwer Academic Publishers, pp. 99-108, 2000.
- [4] R. Keshet, “Mathematical Morphology on Complete Semilattices and its Applications to Image Processing,” *Fundamenta Informaticæ* 41 (2000), pp. 33-56.

- [5] H.J.A.M. Heijmans and R. Keshet, "Inf-semilattice approach to self-dual morphology," *J. Math. Imaging Vision* 17 (1), pp. 55-80, 2002.
- [6] F. Meyer, "Topographic distance and watershed lines," *Signal Processing* 38 (1994), pp. 113-125.
- [7] H.J.A.M. Heijmans, "Theoretical Aspects of Gray-Level Morphology," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 13, No. 6, pp. 568-582, June 1991.
- [8] P. Soille, *Morphological Image Analysis: Principles and Applications*, NY: Springer, 1999.