Convex Structuring Element Decomposition for Single Scan Binary Mathematical Morphology

Nicolas Normand

Convex Structuring Element Decomposition for Single Scan Binary Mathematical Morphology

Nicolas Normand

IRCCyN/IVC UMR CNRS 6597

Abstract. This paper presents a structuring element decomposition method and a corresponding morphological erosion algorithm able to compute the binary erosion of an image using a single regular pass whatever the size of the convex structuring element.
Similarly to classical dilation-based methods, the proposed decomposition is iterative and builds a growing set of structuring elements. The novelty consists in using the set union instead of the Minkowski sum as the elementary structuring element construction operator. At each step of the construction, already-built elements can be joined together in any combination of translations and set unions. There is no restrictions on the shape of the structuring element that can be built. Arbitrary shape decompositions can be obtained with existing genetic algorithms with an homogeneous construction method. This paper, however, addresses the problem of convex shape decomposition with a deterministic method.

```
@inproceedings{normand2003dgci,
  Author = {Normand, Nicolas},
  Booktitle = {Discrete Geometry for Computer Imagery},
  Date = {2003-11-18},
  Doi = {10.1007/b94107},
  Editor = {Nystr{\"o}m, Ingela and Sanniti di Baja, Gabriella and Svensson, Stina},
  Keywords = {Morphologie math{\'e}matique},
  Month = nov,
  Pages = {154-163},
  Publisher = {Springer Berlin / Heidelberg},
  Series = {Lecture Notes in Computer Science},
  Title = {Convex Structuring Element Decomposition for
    Single Scan Binary Mathematical Morphology},
  Volume = {2886},
  Year = {2003}
}
```

The original publication is available at www.springerlink.com

Convex Structuring Element Decomposition for Single Scan Binary Mathematical Morphology

Nicolas Normand

IRCCyN-IVC (CNRS UMR 6597)
École polytechnique de l'université de Nantes
La Chantrerie, rue Christian Pauc, BP 50609, 44306 Nantes Cedex 3 France

Abstract. This paper presents a structuring element decomposition method and a corresponding morphological erosion algorithm able to compute the binary erosion of an image using a single regular pass whatever the size of the convex structuring element.
Similarly to classical dilation-based methods [1], the proposed decomposition is iterative and builds a growing set of structuring elements. The novelty consists in using the set union instead of the Minkowski sum as the elementary structuring element construction operator. At each step of the construction, already-built elements can be joined together in any combination of translations and set unions. There is no restrictions on the shape of the structuring element that can be built. Arbitrary shape decompositions can be obtained with existing genetic algorithms [2] with an homogeneous construction method. This paper, however, addresses the problem of convex shape decomposition with a deterministic method.

1 Introduction

Mathematical morphology operators are time-consuming with large structuring elements and brute force algorithms. In the past, several methods have been described to reduce the cost of these operators. Two main approaches exist, the first one uses a decomposition of a large structuring element into a set of smaller ones. The result is obtained by a series of operations with small structuring elements. The overall cost is then directly connected to the number of operations and depends on the size of the initial structuring element. The second one consists in binarizing a distance map, which can be computed in a fixed number of image scans. It requires the structuring element to be expressed as a distance disk but then the computational cost is constant whatever the size of the structuring element.

The method proposed here is based on a new generalized distance transform (GDT). The algorithm is quite similar to local distance propagation algorithms but in our case, distance increments are not constant over disks size, which allows for much more flexibility in the disk construction. As an example, we describe an algorithm to decompose any convex 2D polygon in a series of pseudo-distance disks for single scan distance map computation.

The overall cost of the mathematical morphology operators derived from this GDT is constant with the size of the structuring element. Moreover, they can be used in a pipeline fashion and have very low memory requirements. In section 2 existing structuring element decomposition methods will be recalled.

2 Distances and Structuring Elements

2.1 Mathematical Morphology Operators

Let A and B be two sets of points in the discrete grid E with origin O, the neutral element for the symetry in E (p symetric element is denoted as $\check{p}$). The erosion of A by the structuring element B is defined as:

$$A \ominus \check{B} = \{p | (B)_p \subseteq A\} \tag{1}$$

where $(B)_p$ is B translated by p: $(B)_p = \{x + p | x \in B\}$.

The erosion dual operator, the dilation can be defined as:

$$A \oplus \check{B} = (A^c \ominus \check{B})^c = \{p | (B)_p \cap A \neq \emptyset\}. \tag{2}$$

The notation $\oplus$ denotes the Minkowski sum of two sets *i.e.* the set of sums of two elements, one taken from the first set the other from the second one.

These basic operators lead to a great variety of image transformations [3]. However, the algorithm directly derived from the fundamental definition given in (eq. 1) is not efficient for a large structuring element B, as it requires the exploration of all translated points of B for each point of the image.

2.2 Distance Map

The distance transform associates to any point x the smallest distance to a point outside the set X:

$$d_X(x) = \min_{y \notin X}(d(x, y)) \tag{3}$$

The distance map is linked to the mathematical morphology erosion by the fact that the set of points whose distance map values are at least r is the eroded set of X by $D(r)$:

$$A \ominus \check{D}(r) = \{p | d_X(p) \geq r\} \tag{4}$$

with $D(r) = \{p | d(O, p) < r\}$. Eroding a shape from a distance map consists in thresholding distance values, so the erosion cost depends only on the cost of the distance transform. Since algorithms exist to compute a distance map in a fixed number of scans [4–6], they can be used to erode with a constant cost whatever the size of the structuring element.

For usual distances, each disk is constructed by the dilation of the previous disk with a basic structuring element as illustrated in fig. 1.a. In a sequential distance map computation, the symetric neighborhood is divided in two halves which are passed over the image once, in reverse order scans [4].

By moving the morphological center of the disks to the last scanned pixel, some one-pass algorithms can be obtained [7]. We can not refer to these disks as distance disks since the symetry property of distances is not verified anymore. Hence the transform is called a generalized distance transform (GDT). However, there is a strong constraint on the shape of the disks since the basic structuring element is unique for the whole set of disks, so this method will only apply to very specific structuring elements.

By mixing different structuring elements, distances like the octagonal distance add some variability: each disk can be built from the previous by a different structuring element (fig. 1.b). Any shape that is decomposed in a series of dilations can be constructed with this method. However, distance transfom algorithms are only known for some specific cases (for instance when two building structuring elements are used periodically [7, 8]).

On the other side, another way of mixing different neighborhoods is used in chamfer distances (fig. 1.c): each disk is built from different-size disks according to local distances described in a neighborood mask.

2.3 Structuring Element Decomposition

The structuring element decomposition methods rely on the fact that a series of erosions with a set of structuring elements is equivalent to a single erosion with the Minkowski sum of the structuring elements:

$$(A \ominus \check{B}) \ominus \check{C} = A \ominus (\check{B} \oplus \check{C}). \tag{5}$$

Decomposition methods generally use a series of basic structuring elements computable by specific hardware machines in one clock cycle. The shape of the building structuring elements depends on the hardware platform and convex polygon decomposition algorithms were presented for instance for linear shaped building structuring elements [9] and for 4 and 8-neighborhood parallel machines [1]. These decompositions lead to optimal morphological operator implementations for parallel or pipeline architectures, but conversely to distance-based methods, the computational complexity depends on the size of the structuring element.

Since some convex polygonal structuring elements can not be decomposed by Minkowski sums, an extra final set union can be needed as displayed in fig. 1.d [10]. In this case, the initial decomposition can be obtained from a single scan GDT. However, the complexity of the last step depends on the shape of the structuring element. Other methods use a fixed number of scans, but are still restricted to simple shapes such as lines [11] or rectangles [12] and also need combination for other kinds of elements [13]. In order to deal with arbitrary shapes, combinatorial and genetic algorithms have been proposed [2].

3 Convex Polygon Decomposition for Single Scan Erosion

The proposed method is the combination of a construction scheme used to recursively build structuring elements (section 3.1), a generalized distance transform

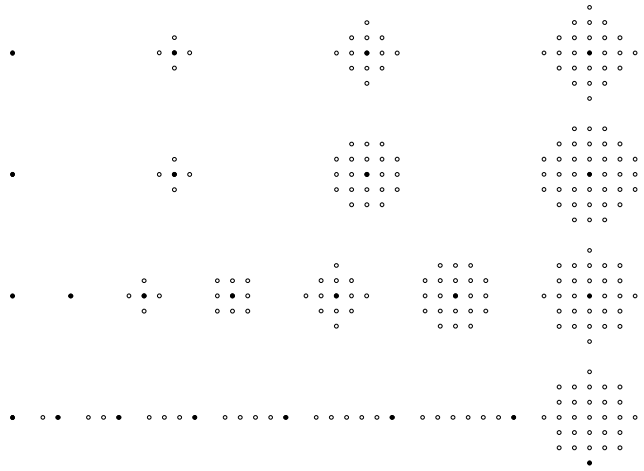

Fig. 1. Some disk and structuring element construction examples. a: d_4 disks (first row). b: octagonal distance disks (second row). c: chamfer distance $d_{2,3}$ disks (third row). d: line elements obtained by a GDT are gathered in the last step (last row)

(section 3.2) and a decomposition algorithm (section 3.3) which determines how structuring elements have to be assembled to obtain a given convex polygon.

A sample polygon $\mathcal{P}$ is shown in fig. 2. It is convex since it is equal to the intersection of all the half-planes supported by its sides. The aim of the method is to obtain structuring element B, the discrete counterpart of $\mathcal{P}$. B is the set of discrete points of the square grid included in the closed polygon $\mathcal{P}$. The construction is directed by a series of increasing polygons $\{\mathcal{P}(i)\}_{i\in[2..N]}$ used as templates for the structuring elements assembling. Each structuring element $B(i)$ is the discrete counterpart of its corresponding polygon $\mathcal{P}(i)$, defined in the continuous plane.

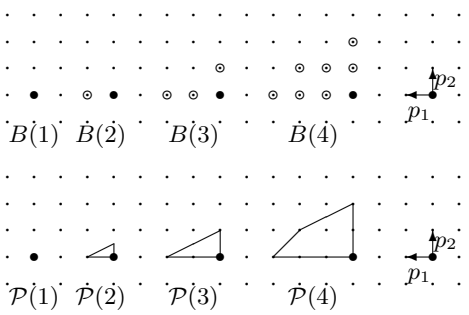

Fig. 2. Series of structuring elements (top), series of polygons (bottom)

Table 1. Structuring element construction table (see text concerning column 1).

i	1	2	3	4
$I_1(i)$	0	1	2	3
$I_2(i)$	0	0	1	3

3.1 Structuring Element Construction

Like the methods recalled in the previous section, the proposed structuring element construction scheme recursively builds a family of increasing elements. However, each structuring element can be built from different smaller elements (conversely to dilation-based construction) and size increments are not fixed for each neighborhood (conversely to chamfer disks). This method operation can be compared to local distance increment with varying weights.

Each structuring element $B(i)$ is the union of smaller structuring elements translated according to a set of neighbors $\{p_k\}$. For instance, in fig. 2,

$$\begin{aligned} B(2) &= B(1) \cup (B(1))_{p_1} \\ B(3) &= B(2) \cup (B(2))_{p_1} \cup (B(1))_{p_2} \\ B(4) &= B(3) \cup (B(3))_{p_1} \cup (B(3))_{p_2} \end{aligned}$$

where $B(1)$ is the simplest element, only containing the origin $\{O\}$.

A general expression is given by introducing $I_k(i)$, the index of the element used in neighborhood p_k for $B(i)$, $B(0)$ the empty set and neighbor p_0 the origin:

$$\forall i = 2 \dots N \; B(i) = \bigcup_{k \in [0,K]} (B(I_k(i)))_{p_k} \tag{6}$$

The values of $I_k(i)$ are summarized in a construction table. Such a table is shown in table 1 for fig. 2 structuring elements. Despite $B(1)$ is not built *stricto sensu*, an extra column 1 is however added for later computing purposes.

Disk increase. By adding $p_0 = O$ with $I_0(i) = i - 1$, we have $(B(I_k(i)))_{p_0} = B(i-1)$, so $B(i-1)$ is always a subset of $B(i)$. Without loss of generality, we can assume that each I_k table contains increasing values ($I_k(i) \geq I_k(i-1)$).

Comparison with Other Methods. This construction scheme generalizes the disk or structuring element construction methods previously recalled. Chamfer distances use constant local distance increments which correspond to a fixed difference between a constructed disk size i and the included disk size $I_k(i)$. Chamfer distance $d_{a,b}$ is obtained with $I_k(i) = i - a$ or $I_k(i) = i - b$ depending on p_k. Dilation series are obtained by taking $I_k(i) = i - 1$ for each p_k belonging to the structuring element used to build $B(i)$.

Each p_k can be any point in the discrete plane. The neighbor set is determined from the shape of the structuring element (section 3.3).

3.2 Single Pass Generalized Distance Transform and Erosion

The value of the distance map at point x is the index of the largest structuring element centered in x contained in X. It is built from elements located on x neighbors: $\{x + p_k\}$. The current element size is the greatest one that contains all the neighbor elements:

$$\begin{aligned} d_X(x) &= \max\{i | \forall k, I_k(i) \leq d_X(x + p_k)\} \\ &= \min_k\{\max\{i | I_k(i) \leq d_X(x + p_k)\}\} \end{aligned}$$

In order to speed up the distance transform computation, we introduce $M_k(j)$:

$$M_k(j) = max(i | I_k(i) \leq j). \tag{7}$$

The distance transform is then:

$$d_X(x) = \begin{cases} \min_k(M_k(d_X(x + p_k))) \text{ if } x \in X \\ 0 \text{ otherwise} \end{cases} \tag{8}$$

$M_k(j)$ represents the index of the largest element $B(i)$ that can be built with $B(j)$ in the neighborhood k. $M_k(j)$ is at least equal to 1 due to column 1 filled with 0 in table 1. M_k can be computed once from the construction table I_k. table 2 shows the M_k values corresponding to the example construction values displayed in table 1

The overall complexity is linear with the number of image pixels like all GDT. Furthermore, if all the neighborhoods are chosen to be causal (all p_k precede O in the scan order) then only one image scan is needed. While it is also true for some GDT for few restricted shape classes, this GDT works with any convex polygonal shape as it will be shown in next section.

The erosion of X by $B = B(N)$ is finally:

$$p \in X \ominus \check{B} \Leftrightarrow d_X(p) = N$$

The causality hypothesis implies that the last vertex in scan order must be equal to the origin O. The single-scan algorithm structure permits to use it in a pipeline chain, with one stage for each morphological operation (for instance, a morphological opening requires two pipeline stages). A first implementation has been realized on a Xilinx Spartan IIE FPGA educational card fed with a PAL video signal. The FPGA handles the input synchronization signal and regenerates it on the output. Due to the low cost of the algorithm, at least 8 morphological pipeline stages with different structuring elements can be handled at video rates without extra resources (only in-chip memory is used). The input-output delay is only a fraction of a pixel for each stage and an extra delay can be introduced in the output synchronization signal to compensate the translation of the structuring element center.

Table 2. Generalized distance transform table M_k. $M_1(1) = 2$ because disk 2 can be built with disk 1 in neighborhood 1 but disk 3 can not

j	0	1	2	3	4
$M_1(j)$	1	2	3	4	4
$M_2(j)$	2	3	3	4	4

Table 3. Half-plane location table

i	0	1	2	3	4
$\mathcal{A}_{0,-1}(\mathcal{P}(i))$	$-\infty$	0	0	0	0
$\mathcal{A}_{-1,2}(\mathcal{P}(i))$	$-\infty$	0	1	2	4
$\mathcal{A}_{-2,1}(\mathcal{P}(i))$	$-\infty$	0	1	2	3
$\mathcal{A}_{1,0}(\mathcal{P}(i))$	$-\infty$	0	0	0	0

3.3 Convex Structuring Element Decomposition

The proposed structuring element construction evokes the anisotropic growth of a single crystal in which epitaxial layers of atoms are successively deposited on a crystal seed. The shape of the crystal is influenced by the physical properties of atoms which constrain the orientations and by the speed of the deposit which may differ from an orientation to another. The orientation of its sides remain constant during the growth. The shape of the structuring element is controlled by artificial constraints which maintains the direction of its sides. However, as the discrete plane produces orientation artifacts especially for small structuring element sizes, the growth is proceeded on a family of continuous polygons which are then used as templates for the structuring elements.

The decomposition method is able to process any convex polygon *i.e.* any closed shape that can be obtained from the intersection of half-planes. For instance, $\mathcal{P}(4)$ shown in fig. 2 is bounded by the following half-planes:

$$(x,y) \in \mathcal{P}(4) \Leftrightarrow \begin{cases} -y \leq 0 \\ -x + 2y \leq 4 \\ -2x + y \leq 3 \\ x \leq 0 \end{cases} \Leftrightarrow \begin{cases} \mathcal{A}_{0,-1}(\mathcal{P}(4)) \leq 0 \\ \mathcal{A}_{-1,2}(\mathcal{P}(4)) \leq 4 \\ \mathcal{A}_{-2,1}(\mathcal{P}(4)) \leq 3 \\ \mathcal{A}_{1,0}(\mathcal{P}(4)) \leq 0 \end{cases} \tag{9}$$

with:

$$\mathcal{A}_{p,q}(X) = \max_{(x,y)\in X}(px + qy)$$

The decomposition algorithm consists in moving the planes from their initial seed position (tangent at the origin with $\mathcal{A}_{p_l,q_l} = 0$) to their final position. A series of positions is computed for all half-planes as displayed in table 3 for fig. 2 polygons. Half-planes locations are set in such a way that the sides of intermediate polygons $\mathcal{P}(i)$ have a constant orientation and an increasing length:

$$\forall l, \forall i, \forall j, \det[v_{l+1,i} - v_{l,i}; v_{l+1,j} - v_{l,j}] = 0 \tag{10}$$

$$\forall i > 0, \forall j \geq i, ||v_{l+1,i} - v_{l,i}|| \geq ||v_{l+1,j} - v_{l,j}|| \tag{11}$$

Structuring elements from polygons The index of the structuring element used in neighborhood p_k is determined as the largest polygon translated by p_k that is included in $\mathcal{P}(i)$:

$$\begin{aligned} I_k(i) &= \max(i : (\mathcal{P}(I_k(i)))_{p_k} \subseteq \mathcal{P}(i) \\ &= \max(i : \forall l, \mathcal{A}_{p_l,q_l}\mathcal{P}(I_k(i)) + \mathcal{A}_{p_k} \leq \mathcal{A}_{p_l,q_l}\mathcal{P}(i)) \end{aligned}$$

This expression of $I_k(i)$ ensures that every structuring element $B(i)$ is a subset of the corresponding polygon $\mathcal{P}(i)$: $\forall i, B(i) \subseteq \mathcal{P}(i)$.

Polygon set As a result of the polygon side properties (constant orientation and increasing length, eq. 10, 11), the series of polygons can be iteratively constructed by Minkowski sums in the continous plane [14]. A direct consequence on the construction is that:

$$\begin{aligned} v_{l,i} = v_{l,j} + p_k &\Rightarrow I_k(i) \geq j \\ &\Rightarrow (v_{l,j} \in B(j) \Rightarrow v_{l,i} \in B(i)) \end{aligned}$$

Therefore, if the set of intermediate vertex v_l positions $\{v_{l,i}\}_{i \in [i..N]}$ contains a path from O to v_l using neighbor moves (plus extra non discrete positions), then v_l is necessarily contained in $\mathcal{P}$. Algorithm 1 takes this point into consideration. Half-plane positions are guided by the movement of vertices. Each vertex is initially located at the origin and follows a path to its final position using the two neighbors of its influence cone. The algorithm ensures that each position in the path is correctly reached by the half-plane, *i.e.* that half-plane boundaries meet exactly at the vertex intermediate positions.

Neighbor selection This phase is actually the first in the decomposition process, it must ensure that the obtained structuring elements are convex and that paths to vertices can be obtained with polygons of increasing side length (eq. 11). Each pair of successive neighbors defines an influence cone that have some similarities with chamfer disks geometry [15]. In an influence cone, each pixel is reached by a series of moves along the two neighbors. The main difference with chamfer distances is that the vertices of the structuring element do not necessarily belong to boundaries between cones. Therefore the number of needed neighbors is generally less than the number of vertices in $\mathcal{P}$. There are two constraints on the pair of neighbors (p_k, p_{k+1}):

i all pixels from the influence cone must be reachable from the neighbors.
A necessary and sufficient condition is that p_k and p_{k+1} form a regular cone ($\det[p_k, p_{k+1}] = 0$) [15].

ii all pathes to a point must be included in the structuring element.
If $p \in B(i)$ is in the cone (p_k, p_{k+1}) and $p = ap_k + bp_{k+1}$ then all the points in the parallelogram (O, ap_k, p, bp_{k+1}) must be in $B(i)$.

Algorithm 1 Half-planes shift computation

$i \leftarrow 0$
while $\exists l : v_{i,l} \neq v_i$ **do**
  {Update reached vertices intermediate position}
  for $l \leftarrow 1$ to L **do**
    {Test of vertex $v_{l,i}$}
    if $\forall m, \mathcal{A}_{p_m,q_m}(\{v_{l,i}\}) \leq \mathcal{A}_{H_m}$ **then**
      {All half-planes contain $v_{l,i}$, move it to the next intermediate location}
      choose neighbor k
      $v_{l,i+1} \leftarrow v_{l,i} + p_k$
    else
      $v_{l,i+1} \leftarrow v_{l,i}$
    end if
  end for
  {Half-plane shift}
  for $l \leftarrow 2$ to $L-1$ **do**
    {Reach the closest vertex v_l or v_{l+1}}
    $\mathcal{A}_{H_l} \leftarrow \min(\mathcal{A}_{p_l,q_l}\{v_l, v_{l+1}\})$
  end for
  $i \leftarrow i + 1$
end while

Algorithm 2 Determination of neighbors

{Selection of the two initial neighbors}
$p_1 = (v_{2x}/\gcd(v_{2x}, v_{2y}), v_{2y}/\gcd(v_{2x}, v_{2y}))$
$p_2 = (v_{Lx}/\gcd(v_{Lx}, v_{Ly}), v_{Ly}/\gcd(v_{Lx}, v_{Ly}))$
{Neighbor insertion for condition i}
while $k < K$ **do**
  if $det(p_k, p_{k+1}) \neq 1$ **then**
    {(p_k,p_{k+1}) is not a regular cone}
    find a et b with extended Euclide's algorithm such that $bp_{k_x} - ap_{k_y} = 1$
    $n \geq \frac{bp_{k+1_x} - ap_{k+1_y}}{p_{k_x}p_{k+1_y} - p_{k_y}p_{k+1_x}} > n-1$
    insert $(a + np_{k_x}, b + np_{k_y})$ after k (indices above k are shifted)
  end if
  $k \leftarrow k + 1$
end while
{Neighbor insertion for condition ii}
for $n = 1$ to L **do**
  {Detection of the cone (p_k,p_{k+1}) containing v_n}
  while v_n is not atteignable **do**
    {division of the cone}
    insert neighbor p_k+p_{k+1} after k
    if v_n is in the second half-cone p_{k+1}+p_{k+2} (indices after insertion) **then**
      $k \leftarrow k + 1$
    end if
  end while
end for

4 Conclusion

We have introduced a unified structuring element construction scheme, the corresponding generalized distance transform algorithm and a convex polygon decomposition method. Eroding an image only requires a single regular scan of the image pixels which differs from the classical chamfer distance transform by table lookups instead of constant local distant increments. The computational properties of these algorithms allow their use in a pipeline manner, optimizing time and memory consumption for series of morphological operations.

References

1. Xu, J.: Decomposition of convex polygonal morphological structuring elements into neighborhood subsets. IEEE trans. on PAMI **13** (1991) 153–162
2. Anelli, G., Broggi, A., Destri, G.: Decomposition of arbitrarily shaped binary morphological structuring elements using genetic algorithms. IEEE trans. on PAMI **20** (1998) 217–224
3. Serra, J.: Image analysis and mathematical morphology. Academic Press London (1982)
4. Rosenfeld, A., Pfaltz, J.: Distances functions on digital pictures. Pattern Recognition Letters **1** (1968) 33–61
5. Yokoi, S., Toriwaki, J., Fukumura, T.: On generalized distance transformation of digitized pictures. PAMI **3** (1981) 424–443
6. Borgefors, G.: Distance transformations in digital images. CVGIP **34** (1986) 344–371
7. Wang, X., Bertrand, G.: An algorithm for a generalized distance transformation based on minkowski operations. In: ICPR. (1988) 1164–1168
8. Wang, X., Bertrand, G.: Some sequential algorithms for a generalized distance transformation based on minkowski operations. IEEE Trans. on PAMI **14** (1992) 1114–1121
9. Gong, W.: On decomposition of structure element for mathematical morphology. In: ICPR. (1988) 836–838
10. Ji, L., Piper, J., Tang, J.: Erosion and dilation of binary images by arbitrary structuring elements using interval coding. Pattern Recognition Letters (1989) 201–209
11. van Herk, M.: A fast algorithm for local minimum and maximum filters on rectangular and octogonal kernels. Pattern Recognition Letters **13** (1992) 517–521
12. Van Droogenbroeck, M.: Algorithms for openings of binary and label images with rectangular structuring elements. In Talbot, H., Beare, R., eds.: Mathematical morphology. CSIRO Publishing, Sydney, Australia (2002) 197–207
13. Soille, P., Breen, E., Jones, R.: Recursive implementation of erosions and dilations along discrete lines at arbitrary angles. IEEE Transactions on Pattern Analysis and Machine Intelligence **18** (1996) 562–567
14. Ohn, S.: Morphological decomposition of convex polytopes and its application in discrete image space. In: ICIP. Volume 2. (1994) 560–564
15. Thiel, E., Montanvert, A.: Chamfer masks: Discrete distance functions, geometrical properties and optimization. In: ICPR. Volume III. (1992) 244–247