ELSEVIER

Pattern Recognition Letters 23 (2002) 649–661

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Medial axis for chamfer distances: computing look-up tables and neighbourhoods in 2D or 3D

E. Remy, E. Thiel *

Laboratoire d'Informatique de Marseille, LIM, Faculté des Sciences de Luminy, Case 901, 163 av Luminy, 13288 Marseille Cedex 9, France

Abstract

Medial axis, also known as centres of maximal disks, is a representation of a shape, which is useful for image description and analysis. Chamfer or weighted distances are discrete distances which allow to approximate the Euclidean distance with integers. Medial axis extraction for chamfer distances is discussed in the literature, but only for simple cases. The principle is to use local tests and look-up tables. In this paper, we give an algorithm which computes for any chamfer distance in 2D or 3D, the look-up table and, very important, the neighbourhood to be tested. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Medial axis; Centres of maximal disks; Chamfer distances; Distance transform; Shape representation

1. Introduction

In a digital shape $\mathscr{S}$, a disk is *maximal* if it is not completely covered by any single other disk in the shape. The *medial axis* MA of $\mathscr{S}$ is the set of centres of maximal disks; an example is given in Fig. 1. If the radii of the disks are kept, MA is a reversible coding of $\mathscr{S}$; it is a global representation, centred in $\mathscr{S}$, allowing shape description, analysis, simplification or compression (see Rosenfeld and Kak, 1982). One attractive solution to detect MA is to use a *distance transform*, denoted DT. In a distance transform on $\mathscr{S}$, each pixel is labelled with its distance to the background; it is also the radius of the largest disk in $\mathscr{S}$, centred on the pixel. The shape of MA depends on the distance function used. The detection of MA has been studied in the literature on some distance transforms; the algorithms are tailored to each case.

Rosenfeld and Pfaltz (1966) have shown for the basic city block and chessboard distances d_4 and d_8 that it is sufficient to detect the local maxima on DT. For chamfer distances using the 3×3 neighbourhood, such as $d_{3,4}$, Arcelli and Sanniti di Baja (1998) proved that some labels have to be lowered on the DT before identifying the local maxima; but their solution cannot be extended to larger neighbourhoods. Borgefors (1993) presented a method to extract MA for the distance $d_{5,7,11}$, using a look-up table. A partial look-up table was also given in (Borgefors et al., 1991) for the Euclidean distance d_E. The principle is general: the *look-up table* gives for each distance value the minimum value of the

* Corresponding author.
E-mail addresses: Eric.Remy@lim.univ-mrs.fr (E. Remy), Edouard.Thiel@lim.univ-mrs.fr (E. Thiel).

PII: S0167-8655(01)00141-6

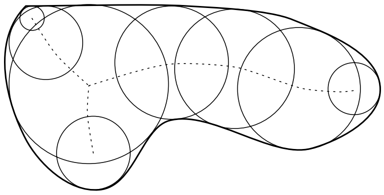

Fig. 1. Medial axis with circles.

neighbours which forbids a point to be in MA. The remaining problem is to compute the look-up table associated with a distance function.

In this paper, we present an efficient algorithm which computes the look-up table for any chamfer distance in 2D or 3D. Moreover, we show that the local neighbourhood to test can be completely different from the chamfer mask. Our algorithm computes the neighbourhood of the look-up table, and certifies that this neighbourhood is sufficient up to a given radius. We recall in Section 2 some basic notions and definitions. In Section 3 we study the existing methods to extract MA. We present and justify in Section 4 our method to compute both the look-up table and the neighbourhood. Finally results are given in Section 5 in the 2D and 3D cases.

2. Basic notions

2.1. Maximal disks

The medial axis transform (MAT) is an image representation scheme proposed by Blum (1967). The essential idea was to find a minimal set of upright squares whose union corresponds exactly to a shape $\mathcal{S}$. Pfaltz and Rosenfeld (1967) have introduced the notion of maximal disks in $\mathcal{S}$ (named "maximal neighbourhoods"). They have shown that the union of the maximal disks is a covering of $\mathcal{S}$. The families of disks they have considered are the distance balls from d_4 and d_8 (lozenges and squares).

The definition of maximal disk can be extended to any family of disks with a radius and a centre such that any centre is included in its disk, and the disk of radius 0 equals its centre. For instance, Jenq and Sahni (1992) have developed sequential and parallel algorithms to compute MAT with rectangular disks; geometric properties are extracted in (Wu et al., 1986; Cordella and Sanniti di Baja, 1989).

A maximal disk can be included in the union of other maximal disks; so the covering by maximal disks, which is unique by construction, is not always a minimal covering. Minimizing the set of maximal disks while preserving reversibility can be interesting for compression. Davies and Plummer (1980) have worked on iterative algorithms with d_6 on the hexagonal grid. More recently, Nilsson and Danielsson (1997) have presented an algorithm with d_E in 2D, with a relation table for the pixel coverage on the border of the disks. Borgefors and Nyström (1997) have described a comparable method for d_E in 2D, and for chamfer distances in 2D and 3D.

Sanniti di Baja and Svensson (2000) have introduced the notion of maximal geodesic disks in a surface. They present a distance transform for surfaces in 3D images. It is used to identify the set of centres of maximal geodesic disks, with the criterion of local maxima. The method is shown for the 3D distances d_6, d_{26} and $d_{3,4,5}$.

The medial axis is a powerful tool in image analysis; while it is a reversible coding, centred in the shape, it is often not thin and disconnected. Further treatments are generally applied to achieve shape analysis. In this way, the medial axis is an important step for weighted skeleton computation (see Sanniti di Baja and Thiel, 1996) and implicit surface reconstruction (see Mari and Sequeira, 2000).

2.2. Chamfer distances

Chamfer distances can be defined in the following way: a *chamfer mask* $\mathcal{M}_C$ is a set of vectors, each of them associated with an integer weight (also called "local distance"). The *chamfer distance* d_C between two points is the cost of the path of least cost joining them, only formed with vectors of the mask.

Borgefors (1984) has popularized chamfer distances in any dimension. Afterwards, many optimization methods were proposed to approximate the Euclidean distance d_E; a major contribution is due to Verwer (1991a) in two and three dimensions. A comparison of different optimization methods and transformation formulas between them can be found in (Thiel, 1994). More recently, new results have been obtained in 3D (see Borgefors, 1996; Remy and Thiel, 2000) and in 4D (see Borgefors, 2000).

Chamfer distances have many advantages, which justify their success in applications. They are local distances, i.e. they permit to deduce a distance from the distances of close neighbours, unlike d_{E}. All computations are performed on integers using elementary operations $\{+,-,<\}$. As we will see, the computation of the medial axis can also be done by local tests. The major attraction is the high speed – and simplicity – of the distance transform algorithm (DT), due to Rosenfeld and Pfaltz (1966). The DT is global, and operates in two scans on the image, independently of the thickness of the shape in the image, and of the dimension. The reverse distance transform (RDT) allows to recover a shape from its medial axis, also in 2 scans.

2.3. Distances and norms

Let E be $\mathbb{Z}^2$ or $\mathbb{Z}^3$. Consider a function d: $E \times E \to \mathbb{N}$ and the properties

positive: $$d(p,q) \geqslant 0 \quad \forall p,q \in E, \tag{1}$$

definite: $$d(p,q) = 0 \iff p = q \quad \forall p,q \in E, \tag{2}$$

symmetric: $$d(p,q) = d(q,p) \quad \forall p,q \in E, \tag{3}$$

triangle inequality:
$$d(p,q) \leqslant d(p,r) + d(r,q) \quad \forall p,q,r \in E, \tag{4}$$

translation invariant:
$$d(p+r,q+r) = d(p,q) \quad \forall p,q,r \in E, \tag{5}$$

homogeneity:
$$d(\lambda p, \lambda q) = |\lambda| \cdot d(p,q) \quad \forall \lambda \in \mathbb{Z},\ \forall p,q \in E. \tag{6}$$

The function d will be called a *discrete distance* if it satisfies (1)–(4), and a *discrete norm* if it satisfies (1)–(6). The *direct ball* B_d and the *reverse ball* B_d^{-1} of centre $p \in E$ and radius $r \in \mathbb{N}$ are the sets of points

$$B_d(p,r) = \{q \in E : d(p,q) \leqslant r\}, \tag{7}$$

$$B_d^{-1}(p,r) = \{q \in E : r - d(p,q) > 0\}, \tag{8}$$

which are central-symmetric if d is a distance. An important property is that the balls are convex if d is a norm.

Any chamfer mask induces a discrete distance (see Verwer, 1991a, p. 20). On the other hand, a chamfer mask does not necessarily induce a discrete norm; some conditions must be fulfilled on the choice of vectors of $\mathscr{M}_{\mathrm{C}}$ in one side, and on the choice of associated weights on the other side. Having a norm is important in applications for the homogeneity in DT, the convexity of the chamfer balls and to ensure that any shortest path is monotonic. We have established exact conditions for a chamfer mask to induce a norm in 2D (Thiel, 1994) and in 3D (Remy and Thiel, 2000).

Borgefors (1996, 2000) has established criteria of validity for another notion, called *regularity*. Kiselman (1996) has compared the regularity and the notion of norm (called "positively homogeneous subadditive function"). While the two notions are very close, we point out that norm implies regularity, but not the contrary (homogeneity on the grid is not guaranteed).

2.4. Weightings and generator

Our workspace is the cubic (resp. square) grid, associated with the fundamental lattice Λ of $\mathbb{Z}^3$ (resp. $\mathbb{Z}^2$). The cubic grid is symmetric with respect to planes of axes and bisectors; these planes divide $\mathbb{Z}^3$ into 48 cones ($48 = 2^3 \cdot 3!$ with 2^3 sign combinations and 3! coordinates permutations) versus 8 octants in $\mathbb{Z}^2$. We call *48-symmetry* (resp. *8-symmetry*) this set of symmetries, and we denote the 48th of space (resp. 8th of plane) represented in Fig. 2 by:

$$\frac{1}{8}\mathbb{Z}^2 = \left\{(x,y) \in \mathbb{Z}^2 : 0 \leqslant y \leqslant x\right\}, \tag{9}$$

and

$$\frac{1}{48}\mathbb{Z}^3 = \left\{(x,y,z) \in \mathbb{Z}^3 : 0 \leqslant z \leqslant y \leqslant x\right\}. \tag{10}$$

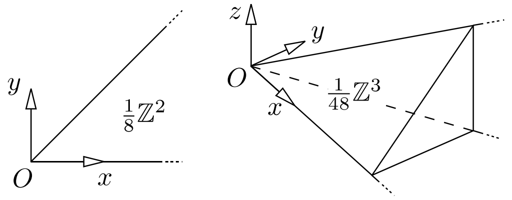

Fig. 2. The regions $\frac{1}{8}\mathbb{Z}^2$ and $\frac{1}{48}\mathbb{Z}^3$.

a	$(1,0,0)$
b	$(1,1,0)$
c	$(1,1,1)$
d	$(2,1,0)$
e	$(2,1,1)$
f	$(2,2,1)$
g	$(3,1,0)$
h	$(3,1,1)$
i	$(3,2,0)$
j	$(3,2,1)$
k	$(3,2,2)$
l	$(3,3,1)$
m	$(3,3,2)$

Fig. 3. First visible points in 3D.

In the following, we use equally $\frac{1}{8}\mathbb{Z}^2$ or $\frac{1}{48}\mathbb{Z}^3$, depending on the context (2D or 3D), as they represent the same notion; for instance, 2D figures use $\frac{1}{8}\mathbb{Z}^2$ whereas the text and the algorithms use $\frac{1}{48}\mathbb{Z}^3$.

We call *weighting* $M = (\vec{v}, w)$ a vector $\vec{v} = (x, y, z) \in \mathbb{Z}^3$ associated with a weight $w \in \mathbb{N}^*$, also denoted $W[\vec{v}]$. In our workspace, a chamfer mask $\mathscr{M}_{\mathrm{C}}$ is a 48-symmetric set of m weightings

$$\mathscr{M}_{\mathrm{C}} = \{M_i(x_i, y_i, z_i, w_i),\ 1 \leqslant i \leqslant m\}. \quad (11)$$

The *generator* $\mathscr{M}_{\mathrm{C}}^g$ of a mask $\mathscr{M}_{\mathrm{C}}$ is the part $\mathscr{M}_{\mathrm{C}} \cap \frac{1}{48}\mathbb{Z}^3$, from which are deduced all other weightings by the 48-symmetry. The cardinal of $\mathscr{M}_{\mathrm{C}}^g$ is denoted by m_{g}. Given a vector $\vec{v}$ in $\mathscr{M}_{\mathrm{C}}$, we name $\vec{v}^g$ the corresponding vector in $\mathscr{M}_{\mathrm{C}}^g$ by the 48-symmetry.

When defining chamfer norms, each weighting (x, y, z, w) generates by translation the *periods* $(2x, 2y, 2z, 2w)$, $(3x, 3y, 3z, 3w)$, etc. For the sake of efficiency during DT, it is self-evident that $\mathscr{M}_{\mathrm{C}}^g$ should only be formed of points such that $\gcd(x, y, z) = 1$. The points having this property are said to be *visible* from the origin (see Hardy and Wright, 1978, Chapter 3). The set of visible points of $\frac{1}{48}\mathbb{Z}^3$ can be obtained with a sieve upon the periods of visible points by scanning $\frac{1}{48}\mathbb{Z}^3$ on x, y, z. Visible points are named $\mathsf{a}, \mathsf{b}, \mathsf{c}, \ldots$ in the sieve order; we also denote by $\mathsf{a}, \mathsf{b}, \mathsf{c}, \ldots$ their corresponding weights, following Borgefors, since no confusions may arise between directions and weights. We give Fig. 3 the Cartesian coordinates of the first visible points in $\frac{1}{48}\mathbb{Z}^3$. Properties of the choice of visible point subsets in a chamfer mask are studied in (Remy and Thiel, 2000). Chamfer distances are named using their constituent weights, e.g. $d_{7,10,13,\mathsf{e}=18}$ refers to the mask $\mathscr{M}_{\mathrm{C}}^g = \{(\mathsf{a}, 7),\ (\mathsf{b}, 10),\ (\mathsf{c}, 13),\ (\mathsf{e}, 18)\}$.

3. Existing methods to extract MA

3.1. Local maxima

After the DT, each shape point p is labelled to its distance $\mathrm{DT}[p]$ to the background. $\mathrm{DT}[p]$ is also the radius of the largest disk in the shape, centred in p, which is by definition of DT and (8), the reverse ball $B_d^{-1}(p, \mathrm{DT}[p])$.

Let $\vec{v}$ be a vector of the mask $\mathscr{M}_{\mathrm{C}}$. The point $p + \vec{v}$ is deeper inside the shape than p (see Fig. 4) if $\mathrm{DT}[p + \vec{v}] > \mathrm{DT}[p]$. Because of the definition of the chamfer distances, the greatest possible value of $\mathrm{DT}[p + \vec{v}]$ is $\mathrm{DT}[p] + W[\vec{v}]$. If this happens, then the point p propagates to $p + \vec{v}$ the distance information during the DT. We deduce that the disk centred in $p + \vec{v}$ completely covers the disk centred in p (see Fig. 4), thus $p \notin \mathrm{MA}$.

On the contrary, if p does not propagate any weighting, then p is called a *local maximum*. Such a point verifies

$$\mathrm{DT}[p + \vec{v}] < \mathrm{DT}[p] + W[\vec{v}] \quad \forall \vec{v} \in \mathscr{M}_{\mathrm{C}}, \quad (12)$$

which we name local maximum criterion (LMC). The set of points detected by the LMC includes the MA by construction. Rosenfeld and Pfaltz (1966) showed that for the basic distances such that $\mathsf{a} = 1$

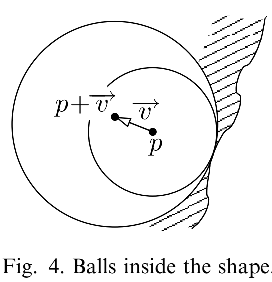

Fig. 4. Balls inside the shape.

(d_4 and d_8 in 2D, d_6, d_{18} and d_{26} in 3D), the LMC set is *exactly* the MA.

The LMC set is no more the MA from the moment that $\mathsf{a} > 1$, since the LMC detects the MA *plus* erroneous points, which are not centres of maximal disks. The LMC set is still reversible; but the erroneous points are generally numerous, in particular close to the border of the shape, and they make the LMC set completely unusable for applications. The detection of erroneous points comes from the presence of equivalent disks.

3.2. Equivalent disks

Two disks of radii r and r' are *equivalent* if the sets of pixels $B_d(O, r)$ and $B_d(O, r')$ are the same (even if the labels of the pixels on the DT are generally different). The *equivalence class* of a disk is the interval of radii for which the disks are equivalent to it. The *conductor* χ is the lowest radius from which the cardinal of all the equivalence classes is 1, i.e. from which all the disks are different. The computation of the conductor χ is related to the Frobenius problem (see Sylvester, 1884; Hujter and Vizvari, 1987). It is obvious that if $\mathsf{a} = 1$ then $\chi = 1$. The equivalence class of the single pixel reverse ball is $[1..\mathsf{a}]$, hence $\chi \geqslant \mathsf{a}$.

The LMC checks in (12) the difference between disk radii; this test is biased by the equivalence classes. Thus, the LMC is inadequate if at least one of the radii is lower than or equal to χ, which happens on the DT if $\mathsf{a} > 1$.

Arcelli and Sanniti di Baja (1998) showed in the 2D case for 3×3 masks that it is sufficient to bring down each value on the DT to the lowest term in its equivalence class; then the LMC is exact on the modified DT. For instance, $d_{3,4}$ simply needs to bring each 3 down to 1 and each 6 down to 5. Their method is inappropriate for masks greater than 3×3 in 2D and $3 \times 3 \times 3$ in 3D, because of the appearance of influence cones in chamfer balls (see Borgefors, 1993; Thiel, 1994). Nacken (1994) showed how to compute the medial axis for the 2D distance $d_{5,7,11}$ with mathematical morphology; but his approach is rather complex and cannot be easily extended to larger masks of either higher dimensions. Arcelli and Frucci (1992) have used a medial axis for $d_{5,7,11}$, but they do not details to how they identify the medial axis.

3.3. Look-up tables

The most general and efficient solution in the literature is the method of the *look-up tables* (LUT), which stores the corrections to the LMC.

A shape point p is a maximal centre if there is no other shape point q such that the ball $B_d^{-1}(q, \mathrm{DT}[q])$ entirely covers the ball $B_d^{-1}(p, \mathrm{DT}[p])$. The presence of q *forbids* p to be an MA point. Suppose that (i) it is sufficient to search q in a local neighbourhood of p and (ii) that we know for each $\mathrm{DT}[p]$ the minimal value $\mathrm{DT}[q]$, stored in a look-up table Lut, which forbids p in direction $\vec{v} = \overrightarrow{pq}$.

(i) The local neighbourhood of vectors to be tested is denoted by $\mathcal{M}_{\mathrm{Lut}}$ and is 48-symmetric. The generator of $\mathcal{M}_{\mathrm{Lut}}$ is denoted by $\mathcal{M}^g_{\mathrm{Lut}}$. Given $\vec{v} \in \mathcal{M}_{\mathrm{Lut}}$, we name $\vec{v}^g$ the corresponding vector by the 48-symmetry in $\mathcal{M}^g_{\mathrm{Lut}}$.

(ii) The minimal value for p and $\vec{v}$ is stored in $\mathrm{Lut}[\vec{v}][\mathrm{DT}[p]]$. Because of the 48-symmetry, it is sufficient to store only the values in $\mathcal{M}^g_{\mathrm{Lut}}$; hence the minimal value for p and $\vec{v}$ is accessed using $\mathrm{Lut}[\vec{v}^g][\mathrm{DT}[p]]$.

Finally we have the following criterion:

$$p \in \mathrm{MA} \iff \mathrm{DT}[p + \vec{v}] < \mathrm{Lut}[\vec{v}^g][\mathrm{DT}[p]] \quad \forall \vec{v} \in \mathcal{M}_{\mathrm{Lut}}. \tag{13}$$

The first use of LUT is due to Borgefors et al. (1991) for d_{E} in 2D. The look-up table is computed via an exhaustive search; the complexity is huge, but the computations are done once for all. The table is given for radii less than $\sqrt{80}$. Afterwards, Borgefors (1993) has given the look-up table for the 2D distance $d_{5,7,11}$, whose entries differ from the LMC for radii less than $\chi = 60$; but she did not generalize her look-up table computation method.

In a previous work, Thiel (1994) has proposed an efficient algorithm to compute the LUT for any chamfer mask in 2D, assuming that $\mathcal{M}_{\mathrm{Lut}} = \mathcal{M}_{\mathrm{C}}$. But he pointed out that for large masks, erroneous points are still detected, which cast doubt over the validity of the whole method.

We have recently discovered that the assumption $\mathscr{M}_{\mathrm{Lut}} = \mathscr{M}_{\mathrm{C}}$ actually is false. In fact, the two masks often completely differ, and we propose in the following, a correct and efficient algorithm which computes both $\mathscr{M}_{\mathrm{Lut}}$ and Lut in 3D. Our method is immediately applicable to the 2D case by skipping any reference to z (and also to the 4D case by adding a fourth coordinate t).

4. Proposed method to compute Lut and $\mathscr{M}_{\mathrm{Lut}}$

4.1. Starting point

The computation of an entry $\mathrm{Lut}[\vec{v}][r]$ in the look-up table for $r = \mathrm{DT}[p]$ in direction $\vec{v}$ consists in finding the smallest radius R of a ball $B_d^{-1}(p+\vec{v}, R)$ which completely covers $B_d^{-1}(p, r)$ (see Fig. 4). One can find R, as illustrated in Fig. 7, by decreasing the radius R_+ while keeping the ball $B_d^{-1}(q, R_+)$ covering the ball $B_d^{-1}(p, r)$, where $q = p + \vec{v} = p - \vec{v}^g$ by symmetry. Unfortunately, each decreasing step needs a prohibitive RDT. To avoid this, we introduce the following lemma which links the results of DT and RDT, i.e. balls and reverse balls. From (7) and (8) we have:

Lemma 1. $B_d(p, r) = B_d^{-1}(p, r+1)$.

While the sets are the same, it is important to note that the resulting labels have different values on DT and RDT, as shown in Fig. 5. Lemma 1 is the starting point of our method: we will show that it is sufficient to compute the DT in $\frac{1}{48}\mathbb{Z}^3$ only once at the beginning and to test all the coverings on the resulting image.

(a)

		6		
	4	3	4	
6	3	0	3	6
	4	3	4	
		6		

(b)

		1		
	3	4	3	
1	4	7	4	1
	3	4	3	
		1		

Fig. 5. Difference of labeling between (a) $B_{d_{3,4}}(6)$ and (b) $B^{-1}_{d_{3,4}}(6+1)$.

4.2. Cone transform

We restrict to distances defined by a *norm*, as presented in Section 2.3, and whose triangulation is a refinement of triangle $(\mathsf{a}, \mathsf{b}, \mathsf{c})$ (see also Remy and Thiel, 2000). Thus, all the considered balls are (iii) convex and (iv) 48-symmetric and such that (v) if $r_1 \leqslant r_2$ then $B_d(O, r_1) \subseteq B_d(O, r_2)$; we have also (vi) all paths involved in distance computation of $d(O, p)$, where $p \in \frac{1}{48}\mathbb{Z}^3$, remain in $\frac{1}{48}\mathbb{Z}^3$. Hence with (iii)–(vi), we can limit the covering test by restricting the two balls to $\frac{1}{48}\mathbb{Z}^3$, which gives Fig. 7.

We denote CT^g, the image resulting from the cone transform, which gives for any point of $\frac{1}{48}\mathbb{Z}^3$ its distance from the origin (see Fig. 6(a)). The CT^g can be obtained for any chamfer norm satisfying (vi), using the fast algorithm given in Fig. 9; it computes CT^g in a single scan and only using $\mathscr{M}^g_{\mathrm{C}}$.

4.3. Computing an entry of Lut

The covering of the ball $B_d^{-1}(q, R_+)$ over $B_d^{-1}(p, r)$ can be tested by simply scanning CT^g;

(a)

			21	25	
		14	18	22	27
	7	11	16	21	26
0	5	10	15	20	25

(b)

			7	5	
		14	11	7	5
	21	18	14	10	5
28	25	20	15	10	5

Fig. 6. Difference between (a) CT^g and (b) DT computed on $B_{d_{5,7,11}}(27) \cap \frac{1}{8}\mathbb{Z}^2$.

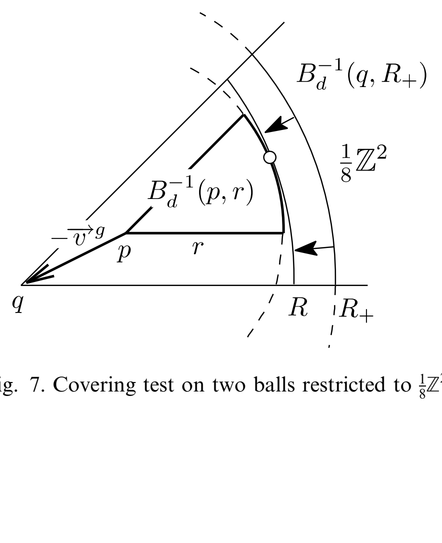

Fig. 7. Covering test on two balls restricted to $\frac{1}{8}\mathbb{Z}^2$.

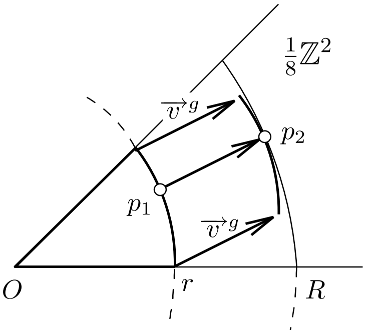

Fig. 8. Translated covering test on CT^g.

moreover, the smallest radius R can be read in CT^g during the scan. We propose to translate both $B_d^{-1}(p,r)$ and $B_d^{-1}(q,R)$ to the origin as shown in Fig. 8. We scan each point p_1 of $B_d^{-1}(O,r) \cap \frac{1}{48}\mathbb{Z}^3$, which by translation of vector $\vec{v}^g$ gives p_2. Values $d(O,p_1)$ and $d(O,p_2)$ are read in CT^g. We have

$$R = \max\left\{ d(O,p_2) : p_2 = p_1 + \vec{v}^g, \; p_1 \in B_d^{-1}(O,r) \cap \frac{1}{48}\mathbb{Z}^3 \right\}, \quad (14)$$

so

$$R = \max\left\{ d(O,p_1 + \vec{v}^g) : p_1 \in B_d^{-1}(O,r) \cap \frac{1}{48}\mathbb{Z}^3 \right\} \quad (15)$$

This process can be efficiently implemented (see Fig. 10), because all the covering relations (r,R) in a direction $\vec{v}^g$ can be detected during the same scan (lines 2–7). To remain in the bounds of the CT^g image, the x scan is limited to $L - v_x^g - 1$ (where v_x^g is the x component of $\vec{v}^g$). For each point p_1, we look for the corresponding radius r_1 which is $CT^g[p_1] + 1$ by Lemma 1. Then we look for the radius r_2 of the ball passing via the point p_2. Its value is $CT^g[p_2] + 1 = CT^g[p_1 + \vec{v}^g] + 1$ by Lemma 1. During the scan, we keep in $\mathrm{Lut}[\vec{v}^g][r_1]$ the greatest value found for r_2, which at the end, is R by (15).

At this stage, our algorithm gives a set of local covering relations, which stands for a partial ordering on the covering of balls. One can observe in Lut, cases where $r_a < r_b$ while $\mathrm{Lut}[\vec{v}^g][r_a] > \mathrm{Lut}[\vec{v}^g][r_b]$, which means that the ball covering $B_d^{-1}(O,r_a)$ is bigger than the ball covering $B_d^{-1}(O,r_b)$. But any d_C is a distance function (see Verwer, 1991a, p. 20), and thus for any d_C, the property (v) in Section 4.2 holds. We therefore correct the table by assuming that in this case, $\mathrm{Lut}[\vec{v}^g][r_b]$ should at least equal $\mathrm{Lut}[\vec{v}^g][r_a]$, building this way a compatible total order (Fig. 10, lines 8–10).

4.4. Computing $\mathscr{M}_{\mathrm{Lut}}$

We now focus on the computation of the set of weightings $\mathscr{M}^g_{\mathrm{Lut}}$, which gives by symmetry $\mathscr{M}_{\mathrm{Lut}}$.

```
Procedure ComputeCT^g ( L, M^g_C, CT^g ) ;
CT^g[(0,0,0)] = 0 ;
for x = 1 to L−1 , for y = 0 to x , for z = 0 to y do
  {
    min = +∞ ;
    for each v^g in M^g_C do
      {
        (x', y', z') = (x, y, z) − v^g ;
        if (x', y', z') ∈ 1/48 Z^3 and CT^g[(x', y', z')] + W[v^g] < min
              then min = CT^g[(x', y', z')] + W[v^g] ;
      }
    CT^g[(x, y, z)] = min ;
  }
```

Fig. 9. Fast cone distance transform algorithm. Input: L the side length, $\mathscr{M}^g_C$ the generator of the d_C mask. Output: CT^g the L^3 distance image to the origin.

```
Procedure ComputeLutCol ( CT^g, L, v^g, R_max, Lut[v^g] ) ;
for r = 0 to R_max do Lut[v^g][r] = 0 ;     // Initializes Lut[v^g] to 0
for x = 0 to L - v^g_x - 1 , for y = 0 to x , for z = 0 to y do
{
  r_1 = CT^g[(x, y, z)] + 1 ;     // radius of the ball where p_1 is located
  r_2 = CT^g[(x, y, z) + v^g] + 1 ;                          // same for p_2
  if r_1 <= R_max and r_2 > Lut[v^g][r_1] then Lut[v^g][r_1] = r_2;
}
r_b = 0 ;
for r_a = 0 to R_max do
  if Lut[v^g][r_a] > r_b then r_b = Lut[v^g][r_a] else Lut[v^g][r_a] = r_b ;
```

Fig. 10. Lut column computation algorithm. Input: CT^g the cone, L the side length, $\vec{v}^g$ the direction of the search, $R_{\max}$ the greatest radius value to be verified in Lut. Output: the column $\mathrm{Lut}[\vec{v}^g]$ is filled with the correct values.

Let us assume that a given set $\mathscr{M}^g_{\mathrm{Lut}}$ is sufficient to extract correctly the MA from any DT image whose values does not exceed R_{Known}. This means that $\mathscr{M}^g_{\mathrm{Lut}}$ enables to extract, from any ball $B_d(O,R)$ where $R \leqslant R_{\mathrm{Known}}$, a medial axis which is by definition the sole point O (the origin). At the beginning, $\mathscr{M}^g_{\mathrm{Lut}}$ is empty and $R_{\mathrm{Known}} = 0$.

So as to increase R_{Known} to R_{Target}, we propose to test each ball $B_d(O,R)$, where $R > R_{\mathrm{Known}}$, each time extracting its DT and then its MA, until whether R reaches R_{Target} or a point different from O is detected in the MA of $B_d(O,R)$. If R reaches R_{Target}, then we know that $\mathscr{M}^g_{\mathrm{Lut}}$ enables to extract the MA correctly for any DT containing values lower than or equal to R_{Target}. Thus this value R_{Target} must be kept as the new R_{Known}.

On the contrary, if one extra point p is found in MA during the scan, then $\mathscr{M}^g_{\mathrm{Lut}}$ is not sufficient to properly extract the MA, since by construction $B_d(O,R)$ covers $B_d^{-1}(p, \mathrm{DT}[p])$. In this case we use:

Lemma 2. *Adding the weighting $(p, \mathrm{CT}^g[p])$ in $\mathscr{M}^g_{\mathrm{Lut}}$ is necessary and sufficient to remove p from the MA of the ball $B_d(O,R)$.*

Proof. The current $\mathscr{M}^g_{\mathrm{Lut}}$ is validated until $R-1$, thus it enables to find all the direct balls covering $B_d^{-1}(p, \mathrm{DT}[p])$ of radii lower than or equal to $R-1$. So, the only direct ball which is not tested is the only ball of radius R: $B_d(O,R)$ itself. This ball is in direction $\overrightarrow{pO}$ from p and must be searched by $\mathscr{M}^g_{\mathrm{Lut}}$ to remove p. Since $\mathscr{M}_{\mathrm{Lut}}$ is symmetric, $B_d(O,R)$ is detected by adding $\overrightarrow{Op}$ in its generator, i.e. by adding the weighting $(p, \mathrm{CT}^g[p])$ in $\mathscr{M}^g_{\mathrm{Lut}}$. □

After having added the weighting, we compute the corresponding new column in Lut. Then, we ensure that this new mask is sufficient to remove p. This is a consistency test of the Lut column computation algorithm in Fig. 10, because we know that the new mask is correct. Failing this test may only come from the impossibility to construct a Lut column enough restrictive to remove p. In this case, there are no possible $\mathscr{M}^g_{\mathrm{Lut}}$ and Lut for this distance.

Once p has been removed, we resume the scan for current R. Other extra points p may be detected sequentially, each time giving a new weighting and Lut column. The computation of $\mathscr{M}^g_{\mathrm{Lut}}$ is finished when R reaches R_{Target}.

Finally, if the chosen distance satisfies (iii)–(vi) in Section 4.2, and if a sufficient $\mathscr{M}^g_{\mathrm{Lut}}$ exists for any given R_{Target}, then our algorithm computes it with its corresponding Lut. If not, this means that the whole method of the LUT of Section 3.3 is incorrect for this distance, which is also detected by our algorithm. Note however that if any of (iii)–(vi) does not hold, results are unpredictable.

4.5. Related algorithms

The full algorithm presented in Fig. 11 uses two other algorithms given in Figs. 12 and 13. They are

```
   Procedure ComputeAndVerifyLut(M_C^g, L, M_Lut^g, R_Known, R_Target, Lut);
1  ComputeCT^g(L, M_C^g, CT^g) ;
2  for each v^g in M_Lut^g do ComputeLutCol(CT^g, v^g, R_Target, Lut ) ;
3  for R = R_Known + 1 to R_Target do
4   {
5    for x = 0 to L − 1 , for y = 0 to x , for z = 0 to y do
6      if CT^g[(x, y, z)] ≤ R
7        then DT^g[(x, y, z)] = 1
8        else  DT^g[(x, y, z)] = 0 ;        // Copy B_d(R) ∩ (1/48)Z^3 to DT^g
9    ComputeDT^g(M_C^g, L, DT^g) ;
10   for x = 1 to L − 1 , for y = 0 to x , for z = 0 to y do
11     if DT^g[(x, y, z)] ≠ 0 and IsMA^g( (x, y, z), M_Lut^g, Lut, DT^g) then
12      {
13       M = (x, y, z, CT^g[(x, y, z)]) ;       // Build a new weighting M
14       M_Lut^g = M_Lut^g ∪ {M} ;              // Add M to M_Lut^g and Lut
15       ComputeLutCol(CT^g, L, (x, y, z), R_Target, Lut[(x, y, z)] ) ;
16       if IsMA^g ( (x, y, z), M_Lut^g, Lut, DT^g ) then error ;
17      }
18  }
```

Fig. 11. Full Lut computation algorithm with determination of $\mathscr{M}^g_{\mathrm{Lut}}$. Input: $\mathscr{M}^g_{\mathrm{C}}$ the generator of the chamfer mask, L the side length of the images, $\mathscr{M}^g_{\mathrm{Lut}}$ the generator of the Lut neighbourhood, R_{Known} the last verified radius, R_{Target} the maximum radius to be verified. Output: Lut the look-up table, R_{Target} the new verified radius. At first call, $\mathscr{M}^g_{\mathrm{Lut}}$ and R_{Known} must be set to $\emptyset$ and 0, respectively. On successful exit, R_{Known} must be set to R_{Target} to memorize that the new $\mathscr{M}^g_{\mathrm{Lut}}$ is valid until R_{Target}.

```
Procedure ComputeDT^g ( M_C^g, L, DT^g ) ;
for z = L − 1 to 0 , for y = L − 1 to z , for x = L − 1 to y do
  if DT^g[(x, y, z)] ≠ 0 then
   {
    min = +∞ ;
    for each v^g in M_C^g do
     {
      (x', y', z') = (x, y, z) + v^g ;
      if x' < L and DT^g[(x', y', z')] + W[v^g] < min
        then min = DT^g[(x', y', z')] + W[v^g] ;
     }
    DT^g[(x, y, z)] = min ;
   }
```

Fig. 12. Fast distance transform in $\frac{1}{48}\mathbb{Z}^3$. Input: $\mathscr{M}^g_{\mathrm{C}}$ the generator of the d_{C} mask, L the side length, DT^g the shape (limited to $\frac{1}{48}\mathbb{Z}^3$). Output: DT^g the distance map to the border of the shape.

adapted versions of the distance transform and medial axis extraction, whose average time is 48^2 times shorter: they work on $\frac{1}{48}\mathbb{Z}^3$ with the generators $\mathscr{M}^g_{\mathrm{C}}$ and $\mathscr{M}^g_{\mathrm{Lut}}$ of the masks, using a single scan. As for the computation of CT^g, these algorithms are only correct when assertions (iii)–(vi) in Section 4.2 hold.

Note that the computation of DT (Fig. 11, line 9) is mandatory, since the MA is extracted from the DT to the background. In fact, a simple

```
Function IsMA^g ( p, M^g_Lut, Lut, DT^g ) ;
for each v^g in M^g_Lut do
   if p − v^g ∈ (1/48)Z^3 then                          // Test only in (1/48)Z^3
      if DT^g[p − v^g] ≥ Lut[v^g][DT^g[p]] then return false ;
return true ;
```

Fig. 13. Fast extraction of MA points from $B_d \cap \frac{1}{48}\mathbb{Z}^3$. Input: p the point to test, $\mathcal{M}^g_{\text{Lut}}$ the generator of the Lut neighbourhood, Lut the look-up table, DT^g the distance transform of the section of the ball. Output: returns *true* if point p is detected as MA in DT^g.

threshold on image CT^g to the radius R gives only the $B_d(O,R) \cap \frac{1}{48}\mathbb{Z}^3$ set, but not the correct DT labels (see Fig. 6, where values of (a) differ from (b)).

5. Results

We give in Figs. 14–17 some LUT computed with our algorithm in 2D and 3D. A sample usage of look-up table using formula (13) and the first line of Fig. 14 is: a point valued 5 on DT is not an MA point, if it has at least a a-neighbour greater than or equal to 6 or a b-neighbour greater than or equal to 8 or a c-neighbour greater than or equal to 12. We show in the tables only the values which differ from the LMC (i.e. $\text{Lut}[\vec{v}^g][r] \neq r + W[\vec{v}^g]$, see (12) and (13)), and thus which represent the irregularities. The tables are also compressed by showing only the radii r which are possible in DT; they may be detected in a single scan on CT^g.

We give in Fig. 14 the result of our algorithm on the well-known 2D distance $d_{5,7,11}$. Values in Fig. 14 are different from (Borgefors, 1993) because we compute the smallest radius in each equivalence class, instead of the greatest. Since all disks are equivalent between these two radii, the two tables must be understood as identical.

$$\mathcal{M}^g_C = \mathcal{M}^g_{Lut} = \left\{ \begin{array}{l} \mathsf{a} = (1,0,5) \\ \mathsf{b} = (1,1,7) \\ \mathsf{c} = (2,1,11) \end{array} \right\}$$

R	a	b	c	R	a	b	c	R	a	b	c
5	6	8	12	21		27		38		44	
7	11	12	17	25	28	30	34	39		45	
10	12	15	19	27		33		40	44		
11		17		28		34		42		48	
14	17	19	23	29	33			46		52	
15	19			30	34			49		55	
16		22		31		37		53		59	
18	22	23	28	32		38		60		66	
20	23	26	30	35	39	41	45				

Fig. 14. $\mathcal{M}^g_{\text{C}}$, $\mathcal{M}^g_{\text{Lut}}$ and Lut for the 2D distance $d_{5,7,11}$.

$$\mathcal{M}^g_C = \left\{ \begin{array}{l} \mathsf{a} = (1,0,14) \\ \mathsf{b} = (1,1,20) \\ \mathsf{c} = (2,1,31) \\ \mathsf{d} = (3,1,44) \end{array} \right\}$$

$$\mathcal{M}^g_{Lut} = \left\{ \begin{array}{r} \mathsf{a} = (1,0,14) \\ \mathsf{b} = (1,1,20) \\ \mathsf{c} = (2,1,31) \\ \mathsf{d} = (3,1,44) \\ 2\mathsf{c} = (4,2,62) \\ \mathsf{i} = (5,2,75) \end{array} \right\}$$

Fig. 15. $\mathcal{M}^g_{\text{C}}$ and $\mathcal{M}^g_{\text{Lut}}$ for the 2D distance $d_{14,20,31,44}$.

Fig. 15 shows the differences between $\mathcal{M}^g_{\text{C}}$ and $\mathcal{M}^g_{\text{Lut}}$ which caused the errors observed in (Thiel, 1994, p. 81) when using the 2D distance $d_{14,20,31,44}$ from Borgefors (1986). One must notice the presence of point 2c which is not a visible point, and thus would not have appeared in $\mathcal{M}^g_{\text{C}}$ as seen in Section 2.4.

The 3D distances used in Figs. 16 and 17 are optimal norms for d_{E} approximation (see Borgefors, 1984; Verwer, 1991b; Remy and Thiel, 2000). Fig. 16 shows some examples of Lut arrays for 3D distances where $\mathcal{M}^g_{\text{C}} = \mathcal{M}^g_{\text{Lut}}$. We give in Fig. 17 a full example of both the computed mask $\mathcal{M}^g_{\text{Lut}}$ and the Lut array for distance $d_{11,16,19,\text{j}=45}$. One must notice the difference between $\mathcal{M}^g_{\text{C}}$ and $\mathcal{M}^g_{\text{Lut}}$ and, as in Fig. 15, the presence of a non-visible point (3e).

While the computation of the Lut array in Fig. 10 is very fast (less than a second [1]), the

[1] On a PC/Linux-2.2.16 PentiumIII 650 MHz.

$d_{3,4,5}$

R	a	b	c
3	4	5	6

$d_{4,6,7,\mathsf{d}=9,\mathsf{e}=10}$

R	a	b	c	d	e
4	5	7	8	10	11
6	9	10	11	14	15
7	10				
8	11				
9		14	15		
12	15	17	18	20	21
16	19				

$d_{19,27,33}$

R	a	b	c
19	20	28	34
27	39	47	53
33	47	55	61
38	53	61	67
46	58	66	72
52	66	74	80
54	72	80	86
57	74	82	88
60	77	85	91
65	80	88	94
71	86	94	100
73	91	99	105

R	a	b	c
76	93	101	107
79	96	104	110
81	99	107	113
84	101	109	115
87	105	113	119
90	107	115	121
92	110	118	124
95	113	121	127
98	115	123	129
103	120	128	134
106	124	132	138
108	126	134	140

R	a	b	c
111	129	137	143
114	132	140	146
117	134	142	148
122	140	148	154
125	143	151	157
130	148	156	162
135	153	161	167
141	159	167	173
144	162	170	176
149	167	175	181
168	186	194	200

$d_{3,4,5,\mathsf{e}=7}$

R	a	b	c	e
3	4	5	6	8
4			8	
6	8			
7			11	

$d_{7,10,13,\mathsf{e}=18}$

R	a	b	c	e
7	8	11	14	19
10	15	18	19	26
13	18	21	24	29
14	19			
17	22	25	27	33
18			29	

R	a	b	c	e
20	26	29	32	37
23	29	32	34	40
24	29			
25			37	
26			37	
27	33			

R	a	b	c	e
28			40	
30	36	39	42	47
33			45	
34	40			
35			47	
36			47	

R	c
38	50
43	55
46	58
48	60
56	68

Fig. 16. Examples of Lut for 3D distances for which $\mathcal{M}^g_{\mathrm{C}} = \mathcal{M}^g_{\mathrm{Lut}}$.

computation of $\mathcal{M}^g_{\mathrm{Lut}}$ in Fig. 11, involving its verification, is slower, and its result should thus be saved for further usage. Computing the $\mathcal{M}^g_{\mathrm{Lut}}$ takes 41s[1] for the 3D distance $d_{11,16,19,\mathrm{j}=45}$ for $L = 100$ and from $R_{\mathrm{Known}} = 0$ to $R_{\mathrm{Target}} = 1066$. This load is explained by the systematic test of 1066 balls $B_d(O, r)$. Each of them involves computations (CT^g and MA extraction) on $\mathrm{O}(r^3)$ points. It is therefore much more interesting to use chamfer distances with small weight values of a since this gives fewer balls to test and thus a faster result. In this scope, in most cases, it is interesting to compensate for the quality loss in d_{E} approximation by more weightings in $\mathcal{M}^g_{\mathrm{C}}$. For example, the extraction of $\mathcal{M}^g_{\mathrm{Lut}}$ for distance $d_{7,10,12,\mathrm{d}=16,\mathrm{e}=17}$ for $L = 100$ from $R_{\mathrm{Known}} = 0$ to $R_{\mathrm{Target}} = 685$ is faster (29s[1]), while achieving a better approximation of d_{E}. Another way to reduce computation cost is to test only once each equivalent ball. But in the case of many chamfer distances, the gain is not noticeable since equivalent balls only appear for small radii where the verification is fast.

6. Conclusion

The computation of the medial axis (MA) from a distance transform (DT) is detailed for any chamfer norm in 2D and 3D. The MA extraction process using LUT was already published with tables corresponding to some common distances (d_{E} and $d_{5,7,11}$), but no general method to compute them was given. We introduce the new mask $\mathcal{M}_{\mathrm{Lut}}$, which stores the test neighbourhood used during the MA extraction. We present and justify an efficient algorithm which computes both Lut and $\mathcal{M}_{\mathrm{Lut}}$ for any chamfer norm satisfying assertions (iii)–(vi) in Section 4.2, in 2D or 3D (and higher dimensions with minor changes). Our algorithm certifies that $\mathcal{M}_{\mathrm{Lut}}$ is sufficient up to a given ball radius. We give results for various chamfer norms proposed in the literature.

Our experimentations show that generally speaking, the neighbourhood $\mathcal{M}_{\mathrm{Lut}}$ to test is completely different from the chamfer mask $\mathcal{M}_{\mathrm{C}}$. This is not surprising because $\mathcal{M}_{\mathrm{Lut}}$ comes from

$$\mathcal{M}^g_C = \left\{ \begin{array}{l} \mathsf{a} = (1,0,0,11) \\ \mathsf{b} = (1,1,0,16) \\ \mathsf{c} = (1,1,1,19) \\ \mathsf{j} = (3,2,1,45) \end{array} \right\} \qquad \mathcal{M}^g_{Lut} = \left\{ \begin{array}{ll} \mathsf{a} = (1,0,0,11) & \mathsf{k} = (3,2,2,49) \\ \mathsf{b} = (1,1,0,16) & \mathsf{d} = (2,1,0,27) \\ \mathsf{c} = (1,1,1,19) & \mathsf{3e} = (6,3,3,90) \\ \mathsf{f} = (2,2,1,35) & \mathsf{j} = (3,2,1,45) \end{array} \right\}$$

R	a	b	c	f	k	d	3e	j
11	12	17	20	36	50	28	91	46
16	23	28	31	46	61	39	102	57
19	28	33	36	52	65	44	106	62
22	31	36	39	55	69	46	110	65
27	34	39	42	57	72	50	113	68
30	39	44	46	62	76	55	117	73
32	42	46	50	65	80	57	121	76
33					81		121	
35	45	50	53	68	83	61	124	79
38	46	52	55	71	84	62	125	81
41	50	55	58	74	88	66	129	84
43	53	57	61	76	91	68	132	87
44			62	78	91		132	
45				79				
48	57	62	65	81	95	73	136	91
49					97		136	
51	61	66	69	84	99	77	140	95
52	62				100	78	140	
54	64	68	72	87	102	79	143	98
55					103		144	
56				90				
57				91				
59	69	74	77	93	107	84	148	103
60			78	94	107		148	
61				95				
63	73	78	81	97	110	89	151	107
64		79		98		90		

R	a	b	c	f	k	d	3e	j
66			84	100	114		155	
67				101				
70	80	84	88	103	118	95	159	114
71					119		159	
72				106				
73			91	107	121		162	
74	84				122	100	163	
75		90		109		101		
76				110				
79				113				
80		95		114		106		
81					129		170	
82			100	116	129		170	
83				117				
85	95	100	103	119	133	111	174	129
86		101		120		112		
88				122				
89				123				
92			110	126	140		181	
93					141		182	
95				129				
96						122		
98				132				
99				133				
101				135				
102				136				
104			122	138	152		193	

R	k
105	139
108	142
111	145
114	148
117	151
118	152
120	154
121	155
124	158
127	161
130	164
133	167
137	171
140	174
143	177
146	180
149	183
156	190
159	193
162	196
165	199
175	209
178	212
194	228

Fig. 17. $\mathscr{M}^g_{\mathrm{Lut}}$ and Lut for the 3D distance $d_{11,16,19,\mathrm{j}=45}$.

inclusions of discrete chamfer balls. The geometry of the borders of these balls is rather irregular, which makes the analysis of inclusions much more complicated than in the continuous case. Even if we can reasonably conjecture that $\mathscr{M}_{\mathrm{Lut}} = \mathscr{M}_{\mathrm{C}}$ for 3×3 and $3 \times 3 \times 3$ neighbourhoods, a further work needs to be done to get a better understanding of the discrete phenomenons involved, and to find arithmetical rules, if any. For instance, some masks have a finite set of irregularities (see Section 5); finding an upper bound, i.e. the greatest radius where an irregularity occurs, would limit the verification process; its computation could be related to the notion of conductor χ (see Section 3.2).

7. Electronic appendix

Some more examples and program sources (in C language) to compute Lut and $\mathscr{M}^g_{\mathrm{Lut}}$ for dimensions 2, 3 and 4 are available at http://www.lim.univ-mrs.fr/~thiel/PRL2001/ or http://www.dil.univ-mrs.fr/~thiel/PRL2001/

References

Arcelli, C., Frucci, M., 1992. Reversible skeletonization by (5–7–11)-erosion. In: Arcelli, C., et al. (Eds.), Visual Form Analysis and Recognition. Plenum Press, New York, pp. 21–28.

Arcelli, C., Sanniti di Baja, G., 1998. Finding local maxima in a pseudo-euclidean distance transform. Comput. Vision Graphics Image Process. 43, 361–367.

Blum, H., 1967. A transformation for extracting new descriptors of shape. In: Wathendunn, W. (Ed.), Models for the Perception of Speech and Visual Form. MIT Press, Cambridge, MA, pp. 362–380.

Borgefors, G., 1984. Distance transformations in arbitrary dimensions. Comput. Vision Graphics Image Process. 27, 321–345.

Borgefors, G., 1986. Distance transformations in digital images. Comput. Vision Graphics Image Process. 34, 344–371.

Borgefors, G., 1993. Centres of maximal disks in the 5–7–11 distance transform. In: Proc. 8th Scandinavian Conf. on Image Analysis, Tromsø, Norway, pp. 105–111.

Borgefors, G., 1996. On digital distance transforms in three dimensions. Computer Vision and Image Understanding 64 (3), 368–376.

Borgefors, G., 2000. Some weighted distance transforms in four dimensions. In: Proc. 9th Discrete Geometry for Computer Image, Uppsala, Sweden. Lecture Notes in Computer Science, Vol. 1953, pp. 325–336.

Borgefors, G., Nyström, I., 1997. Efficient shape representation by minimizing the set of centres of maximal discs/spheres. Pattern Recognition Lett. 18, 465–472.

Borgefors, G., Ragnemalm, I., Sanniti di Baja, G., 1991. The Euclidean Distance Transform: finding the local maxima and reconstructing the shape. In: Proc. 7th Scandinavian Conf. on Image Analysis, Aalborg, Denmark, Vol. 2, pp. 974–981.

Cordella, L.P., Sanniti di Baja, G., 1989. Geometric properties of the union of maximal neighborhoods. IEEE Trans. PAMI 11 (2), 214–217.

Davies, E.R., Plummer, A.P.N., 1980. A new method for the compression of binary picture data. In: 5th ICPR, Miami, pp. 1150–1152.

Hardy, G.H., Wright, E.M., 1978. An Introduction to the Theory of Numbers, fifth ed. Oxford University Press, Oxford.

Hujter, M., Vizvari, B., 1987. The exact solutions to the Frobenius problem with three variables. Ramanujan Math. Soc. 2 (2), 117–143.

Jenq, J.F., Sahni, S., 1992. Serial and parallel algorithms for the Medial Axis Transform. IEEE Trans. PAMI 14 (12), 1218–1224.

Kiselman, C.O., 1996. Regularity properties of distance transformations in image analysis. Computer Vision and Image Understanding 64 (3), 390–398.

Mari, J.L., Sequeira, J., 2000. Using implicit surfaces to characterize shapes within digital volumes. In: RECPAD'00, Porto, Portugal, pp. 285–289.

Nacken, P.F.M., 1994. Image analysis methods based on hierarchies of graphs and multi-scale mathematical morphology. Ph.D. Thesis, Amsterdam.

Nilsson, F., Danielsson, P.E., 1997. Finding the minimal set of maximum disks for binary objects. Graphical Models Image Process. 59 (1), 55–60.

Pfaltz, J.L., Rosenfeld, A., 1967. Computer representation of planar regions by their skeletons. Comm. ACM 10 (February), 119–125.

Remy, E., Thiel, E., 2000. Optimizing 3D chamfer masks with norm constraints. In: Proc. 7th IWCIA, Internat. Workshop on Combinatorial Image Analysis, Caen, pp. 39–56.

Rosenfeld, A., Kak, A.C., 1982. In: Digital Image Processing. Academic Press, New York.

Rosenfeld, A., Pfaltz, J.L., 1966. Sequential operations in digital picture processing. J. ACM 13 (4), 471–494.

Sanniti di Baja, G., Svensson, S., 2000. Detecting centres of maximal geodesic discs on the distance transform of surfaces in 3D images. In: Proc. 9th Discrete Geometry for Computer Image, Uppsala, Sweden. Lecturers Notes in Computer Science, Vol. 1953, pp. 443–452.

Sanniti di Baja, G., Thiel, E., 1996. A skeletonization algorithm running on path-based distance maps. Image and Vision Comput. 14 (1), 47–57.

Sylvester, J., 1884. Mathematical questions with their solutions. Educ. Times 41, 21.

Thiel, E., 1994. Les distances de chanfrein en analyse d'images: fondements et applications. Ph.D. Thesis, Université Joseph Fourier, Grenoble 1, September, http://www.lim.univ-mrs.fr/~thiel/these.

Verwer, J.H., 1991a. Distance transforms: metrics, algorithms and applications. Ph.D. Thesis, Technische Universiteit, Delft.

Verwer, J.H., 1991b. Local distance for distance transformations in two and three dimensions. Pattern Recognition Lett. 12, 671–682.

Wu, A.Y., Bhaskar, S.K., Rosenfeld, A, 1986. Computation of geometric properties from the Medial Axis Transform in $O(n \log n)$ time. Comput. Vision Graphics Image Process. 34, 76–92.