Computing 3D Medial Axis for Chamfer Distances

Eric Remy and Edouard Thiel

Laboratoire d'Informatique de Marseille
LIM, Case 901, 163 av Luminy, 13288 Marseille Cedex 9, France
{Eric.Remy,Edouard.Thiel}@lim.univ-mrs.fr

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. Computing medial axis with chamfer distances has been discussed in the literature for some simple cases, mainly in 2D. In this paper we give a method to compute the medial axis for any chamfer distance in 2D and 3D, by local tests using a lookup table. Our algorithm computes very efficiently the lookup tables and, very important, the neighbourhood to be tested.

Keywords: Medial axis, Centres of Maximal Disks, Chamfer Distances, Lookup Table Transform, Shape Representation.

1 Introduction

A digital image is a set of colored points in a space, for instance the voxels on the cubic grid. A shape in a binary image, is a subset of the points of the image, sharing the same color. The points of the shape, taken individually, do not provide any global information for the shape description.

Given a family of disks and a shape, we consider the disks completely included in the shape. If the smallest disk in the family is a single point, then the shape can be completely covered by such disks. Coding their positions and sizes is sufficient to reconstruct the original shape.

Now consider a covering of the shape by maximal disks, defined as follows:

Definition 1. A disk is maximal in the shape if the disk is not included in any single other disk in the shape.

The maximal disks are centred in the shape. Note that a maximal disk can be included in the union of other maximal disks; so the covering by maximal disk, which is unique by construction, is not always a minimal covering.

Definition 2. The Medial Axis (MA) of a shape is the set of centres of maximal disks in the shape.

The medial axis is a reversible coding of the shape; it is a global representation, centred in the shape, which allows shape description, analysis, simplification or compression. The medial axis is often not thin and disconnected. It is a step for weighted skeleton computation [12] and implicit surface reconstruction [7].

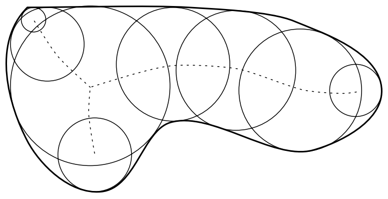

Fig. 1. Medial Axis with circles

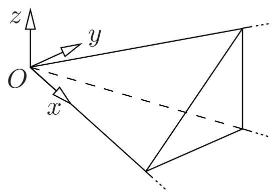

Fig. 2. The 48^{th} of space $\mathcal{S}$

The aspect of the medial axis and its computation depend on the geometry of the family of disks, which can be circles, rectangles, distance balls, etc. For instance, figure 1 schematizes a medial axis covering a shape with circles.

The medial axis is especially attractive when the disks are distance balls, for computation cost and for description purpose. A distance is positive defined, symetric and respects the triangular inequality. Given a distance d, the distance ball B_d of centre p and radius r, is the set of points

$$B_d(p,r) = \{\, q \,:\, d(p,q) \leq r \,\} \tag{1}$$

which is centre-symetric. There exists innumerable distances functions; since digital images are the most often encoded in integer arrays, we focus on discrete distances, which are distances working with integers.

2 Chamfer Distances

Chamfer distance d_C can be defined in the following way: a chamfer mask $\mathcal{M}_C$ is a set of legal displacements in a neigbourhood, each displacement being weighted by an integer cost; the chamfer distance d_C between two points is the cost of the path of least cost joining them, formed with legal displacements in the mask.

Borgefors popularizes chamfer distances in [2], in any dimension. Afterwards many optimization methods have been proposed, to approximate the Euclidean distance d_E; the major contribution is due to Verwer in 2 and 3 dimensions [13]. One can find in [12] a complete history of chamfer distance, the comparison of different optimization methods and crossing formulas between them. More recently, we propose in [8] new results in the 3D case, using Farey triangulations.

Chamfer distances have many advantages, which justify their success in applications. They are local distances, that is to say, which permit to deduce a distance from the distances of close neighbours, unlike d_E. All computations are done using integers and linear 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, denoted DT, due to Rosenfeld et al. [10]. The DT consists in labeling each point of a shape to its distance to the complementary. The DT is global, and operates in 2 passes 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 it's medial axis, also in 2 passes.

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

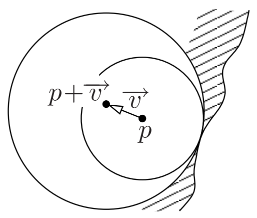

Fig. 4. Balls inside the shape

Our work space is the cubic grid, associated with the fundamental lattice Λ of $\mathbb{Z}^3$. The cubic grid implies the symmetry towards planes of axes and bissectrices, called 48-symmetry : it divides $\mathbb{Z}^3$ into 48 sub-spaces ($48 = 2^3.3!$ with 2^3 sign combinations and 3! coordinates permutations), versus 8 octants in $\mathbb{Z}^2$. We denote $\mathcal{S}$ the 48^{th} of space represented in figure 2:

$$\mathcal{S} = \left\{ 0 \le z \le y \le x \,,\ (x,y,z) \in \mathbb{Z}^3 \right\} \ . \tag{2}$$

We call weighting $M(\vec{v}, w)$ a displacement $\vec{v}(x,y,z) \in \mathbb{Z}^3$ associated with a weight $w \in \mathbb{N}^*$, also denoted $W[\vec{v}]$. In our work space, a chamfer mask $\mathcal{M}_C$ is a 48-symmetric set of m weightings

$$\mathcal{M}_C = \{ M_i(x_i, y_i, z_i, w_i) \,,\ 1 \le i \le m \} \ . \tag{3}$$

The generator $\mathcal{M}_C^g$ of a mask $\mathcal{M}_C$ is the part $\mathcal{M}_C \cap \mathcal{S}$, from which are deduced all other weightings by the 48-symmetry. The cardinal of $\mathcal{M}_C^g$ is denoted m_g. Given a displacement $\vec{v}$ in $\mathcal{M}_C$, we name $\vec{v}^g$ the corresponding displacement in $\mathcal{M}_C^g$ by the 48-symmetry.

A 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 $\mathcal{M}_C^g$ should only be formed of points such that $\gcd(x, y, z) = 1$. The points having this property are said visible (from the origin, see [5]). The set of visible points of $\mathcal{S}$ can be obtained with a sieve upon the periods of visible points, by scanning $\mathcal{S}$ on x, y, z. Visible points are named a, b, c, ... in the sieve order. We give figure 3 the cartesian coordinates of the first visible points in $\mathcal{S}$. Properties of the choice of visible point subsets in a chamfer mask are studied in [8,9].

3 Existing Methods to Extract MA

3.1 Local Maxima

After the DT, each shape point p is labeled to its distance $DT[p]$ to the complementary. Let $\overrightarrow{v}$ be a displacement of the mask $\mathcal{M}_C$. The point $p+\overrightarrow{v}$ is deeper inside the shape than p (see figure 4) if $DT[p+\overrightarrow{v}] > DT[p]$. Because of the definition of the chamfer distances, the greatest possible value of $DT[p+\overrightarrow{v}]$ is $DT[p] + W[\overrightarrow{v}]$. If this hapens, then the point p propagates to $p+\overrightarrow{v}$ the distance information during the DT. We deduce that the disk centered in $p+\overrightarrow{v}$ completely overlaps the disk centered in p (figure 4), thus $p \notin$ MA.

On the contrary, if p does not propagate any weighting, then p is called a Local Maximum. Such a point verifies

$$DT[p+\overrightarrow{v}] < DT[p] + W[\overrightarrow{v}] \quad , \quad \forall \overrightarrow{v} \in \mathcal{M}_C \tag{4}$$

which we name Local Maximum Criterion (LMC). The set of points detected by the LMC, includes the MA by construction. Rosenfeld and Pfaltz showed in [10] that for the basic distances such that $\mathsf{a} = 1$ (d_4 and d_8 in 2D, d_6, d_{18} and d_{26} in 3D), the LMC set is exactly MA.

3.2 Equivalent Disks

This is no more true from the moment that $\mathsf{a} > 1$, since the LMC detects the MA plus erroneous points, which are not center 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 MA completely unusable for applications.

The trouble comes from the check in (4) in the difference between disk radii on DT. In fact, if $\mathsf{a} > 1$, then several radii may correspond to a same disk, ie. a given disk may have an interval of equivalent integer radii. The computation of the equivalence classes is related to the Frobenius problem [11,6]. For instance, the class of equivalence of the single pixel ball is $[1\mathinner{\ldotp\ldotp}\mathsf{a}]$. The corollary is that (4) is inadequate for $\mathsf{a} > 1$.

Arcelli and Sanniti di Baja 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 ; the LMC is then exact on the modified DT. For instance, $d_{3,4}$ simply needs to bring the 3 down to 1 and the 6 down to 5 [1]. 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 [12].

3.3 Lookup Tables

The most general and efficient solution is the method of the lookup tables, 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 $B_d(q, DT[q])$ entirely overlaps the ball $B_d(p, DT[p])$. The presence of q forbids p to be a MA point. Suppose that (i) it is sufficient to search q in a local neigbourhood of p and (ii) that we know for each $DT[p]$ the minimal value $DT[q]$, stored in a lookup table Lut, which forbids p in direction $\overrightarrow{v} = \overrightarrow{pq}$.

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

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

Finally we have the following criterion:

$$p \in \mathrm{AM} \quad \Longleftrightarrow \quad DT[p+\overrightarrow{v}] < Lut[\overrightarrow{v}^g][DT[p]] \ , \quad \forall \overrightarrow{v} \in \mathcal{M}_{Lut} \ . \tag{5}$$

The first use of lookup tables is due to Borgefors, Ragnemalm and Sanniti di Baja in [4], for d_E in 2D. The lookup tables are computed with exhaustive search; the combinatory is enormous, but the computations are done once for all. The tables are given for radii less than $\sqrt{80}$. Borgefors gives in [3] the lookup tables for the 2D distance $d_{5,7,11}$, which entries differs from the LMC for radii less than 60; but she does not generalize her lookup table computation method.

One of the authors proposes in [12] an efficient algorithm to compute the lookup tables for any chamfer mask in 2D, assuming that $\mathcal{M}_{Lut} = \mathcal{M}_C$. But he points out that for large masks, erroneous points may be detected: a contradiction, and a fondamental question about the validity of the whole method.

We have recently discovered that the assumption $\mathcal{M}_{Lut} = \mathcal{M}_C$ actualy is the error. In fact, the two masks often completely differ, and we propose in the following the correct and efficient algorithm which computes both $\mathcal{M}_{Lut}$ and Lut in 3D. The following method is immediately applicable to the 2D case by skipping any reference to z.

4 Proposed Method to Compute Lookup Tables

Let us come back to the meaning of the labels on the DT image. A point p is labeled to its distance $r = DT[p]$ to the complementary. If we apply the RDT on p, it generates the reverse ball denoted $B_d^{-1}(p,r)$, which is the set of points

$$B_d^{-1}(p,r) = \{\, q \,:\, r - d(p,q) > 0 \,\} \tag{6}$$

and which is the greatest ball inside the shape, centred in p. From (1) and (6) it comes immediately the following lemma:

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

While the sets are the same, it is important to note that resulting labels have different values on DT and RDT, as demonstrated figure 5.

The computation of an entry $Lut[\overrightarrow{v}][r]$ in the lookup table for $r = DT[p]$ in a direction $\overrightarrow{v}$, consists in finding the smallest radius R of a ball $B_d^{-1}(p+\overrightarrow{v},R)$

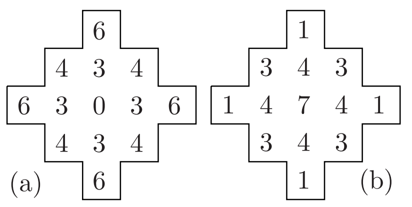

Fig. 5. Difference of labeling between (a) $B_{d_{3,4}}(6)$ and (b) $B^{-1}_{d_{3,4}}(6+1)$

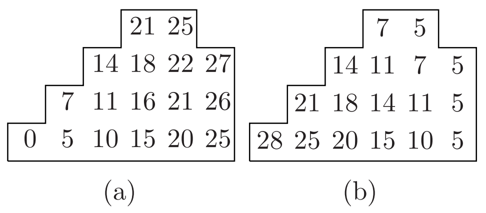

Fig. 6. Difference between (a) $B_{d_{5,7,11}}(27) \cap \mathcal{S}$ produced by CT^g and (b) its DT^g

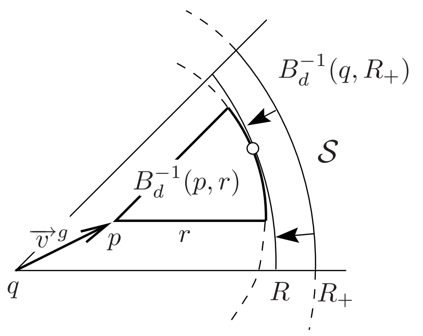

Fig. 7. Overlapping test on two balls restricted to $\mathcal{S}$

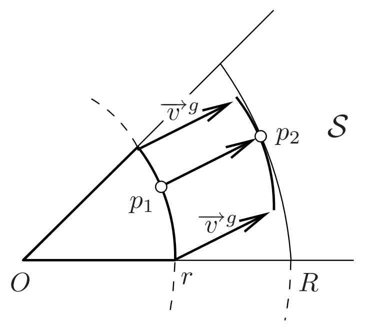

Fig. 8. Translated overlapping test on DT^g

which overlaps completely $B^{-1}_d(p, r)$ (see figure 4). As illustrated figure 7, a method to find R is to decrease the radius R_+ while the ball overlaps, but each step needs a RDT, and the cost is prohibitive. The lemma 1 is the starting point of our method: we will show that it is sufficient to compute the DT in $\mathcal{S}$ only a single time at the beginning; the lemma 1 allows to test the overlappings on $\mathcal{S}$.

4.1 Computing an Entry of the Lookup Table

We choose to restrict our study to balls which are convex and symetric, ie. to distances inducing a norm (see [8]). Therefore, we can limit the overlapping test by restricting the two balls to $\mathcal{S}$, which gives the figure 7. This loss of generality should not be annoying since non-norm distances are not homogeneous (the shortest path may zigzag over the grid). Note however that it is still possible not to restrict to $\mathcal{S}$ leading to a greater computationnal load.

Instead of reducing the radius R_+ while the ball overlaps in figure 7, we propose to translate both $B^{-1}_d(p, r)$ and $B^{-1}_d(q, R)$ to the origin as shown in figure 8. We scan each point p_1 of $B^{-1}_d(O, r) \cap \mathcal{S}$, which by translation of vector $\overrightarrow{v}^g$ gives p_2. Finally, we find

$$R = \max \left\{ d(O, p_2) \; : \; p_2 = p_1 + \overrightarrow{v}^g \, , \; p_1 \in B^{-1}_d(O, r) \cap \mathcal{S} \right\} \tag{7}$$

$$= \max \left\{ d(O, p_1 + \overrightarrow{v}^g) \; : \; p_1 \in B^{-1}_d(O, r) \cap \mathcal{S} \right\} . \tag{8}$$

This process can be very efficiently implemented, using a correspondence between figure 8 and the image CT^g, which is the result of the Cone Transform. It gives for any point of $\mathcal{S}$ its distance from the origin, see figure 6.a. For any chamfer distance that induces a norm (see [9,8]), a fast algorithm (given in figure 9) exists that computes CT^g in a single pass, only using $\mathcal{M}^g_C$.

```
PROCEDURE Compute$CT^g$ ( $L$, $\mathcal{M}^g_C$, $Cone$ ) ;
$Cone[(0,0,0)] = 0$ ;
FOR $x = 1$ TO $L-1$ , FOR $y = 0$ TO $x$ , FOR $z = 0$ TO $y$ DO
 {
   $min = +\infty$ ;
   FOREACH $\overrightarrow{v}^g$ IN $\mathcal{M}^g_C$ DO
    {
      $(x',y',z') = (x,y,z) - \overrightarrow{v}^g$ ;
      IF $(x',y',z') \in \mathcal{S}$ AND $Cone[(x',y',z')] + W(\overrightarrow{v}^g) < min$ THEN
           $min = Cone[(x',y',z')] + W(\overrightarrow{v}^g)$ ;
    }
   $Cone[(x,y,z)] = min$ ;
 }
```

Fig. 9. Fast Cone Distance Transform Algorithm. Input: L the side length, $\mathcal{M}^g_C$ the generator of the d_C mask. Output : $Cone$ the L^3 distance image containing the cone

```
    PROCEDURE ComputeLutColumn ( $Cone$, $L$, $\overrightarrow{v}^g$, $R_{max}$, $Lut[\overrightarrow{v}^g]$ ) ;
 1  FOR $r = 1$ TO $R_{max}$ DO $Lut[\overrightarrow{v}^g][r] = 0$ ;          // Initializes $Lut[\overrightarrow{v}^g]$ to 0
 2  FOR $x = 1$ TO $L-1$ , FOR $y = 0$ TO $x$ , FOR $z = 0$ TO $y$ DO
 3   {
 4     $r_1 = Cone[(x,y,z)] + 1$ ;              // radius of the ball where $p_1$ is located
 5     $r_2 = Cone[(x,y,z) + \overrightarrow{v}^g] + 1$ ;                           // same for $p_2$
 6     $Lut[\overrightarrow{v}^g][r_1]$ = MAX $\{Lut[\overrightarrow{v}^g][r_1], r_2\}$ ;
 7   }
 8  $r_2 = 0$ ;
 9  FOR $r_1 = 0$ TO $R_{max}$ DO
10     IF $Lut[\overrightarrow{v}^g][r_1] > r_2$ THEN $r_2 = Lut[\overrightarrow{v}^g][r_1]$ ELSE $Lut[\overrightarrow{v}^g][r_1] = r_2$ ;
```

Fig. 10. *Lut* Computation Algorithm. Input : *Cone* the cone, L the side length, $\overrightarrow{v}^g$ the column of *Lut* to be computed, R_{max} the greatest radius value seekable in *Lut*. Output : $Lut[\overrightarrow{v}^g]$ is filled with the correct values

The implementation of *Lut* computation shown in figure 10, consists in a scan of $\mathcal{S}$ (line 2–7). For each point p_1, we look for the corresponding radius r_1 which is $CT^g[p_1] + 1$ because of lemma 1. We then look for the radius r_2 of the ball passing by point p_2; its value is $CT^g[p_2] + 1 = CT^g[p_1 + \overrightarrow{v}^g] + 1$, also because of lemma 1. During the scan, we keep the greatest value found for r_2, which at the end, is R by (8).

Due to the discrete nature of the balls, one can observe in Lut, cases where for instance $r_1 < r_2$ while $Lut[\overrightarrow{v}^g][r_1] > Lut[\overrightarrow{v}^g][r_2]$, which means that the ball overlapping $B_d(r_1)$ is bigger than the ball overlapping $B_d(r_2)$. These artefacts of discrete distances should not happen because it has been proved in [13], p. 20, that any d_C is a distance function, and thus

$$\forall r_1, r_2 \quad r_1 < r_2 \Rightarrow B_d(r_1) \subseteq B_d(r_2)\,. \tag{9}$$

We therefore have to correct the table by assuming that in this case, $Lut[\overrightarrow{v}^g][r_2]$ should at least equals $Lut[\overrightarrow{v}^g][r_1]$ (figure 10, lines 8–10).

4.2 Computing the Mask of the Lookup Table

We now focus on the computation of the set of weightings $\mathcal{M}^g_{Lut}$.

We assume that a given set $\mathcal{M}^g_{Lut}$ is sufficient to extract correctly the MA from any DT image which values does not exceed R_{Known}, ie. this $\mathcal{M}^g_{Lut}$ enables to extract from any ball $B_d(O, r)$ where $r \leq R_{Known}$, a medial axis which is (by definition 2) the sole point O (the origin). At the beginning, $\mathcal{M}^g_{Lut}$ is empty and $R_{Known} = 0$.

We propose to test each ball $B_d(O, r)$, where $r > R_{Known}$, 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 $B_d(r)$. If r reaches R_{Target}, then we know that $\mathcal{M}^g_{Lut}$ enables to extract MA correctly, for any DT containing values lower or equal to R_{Target}.

On the contrary, if we reach the case where (at least) one extra point p is found in MA, it means that $\mathcal{M}^g_{Lut}$ is not sufficient to correctly extract MA. We propose to introduce the new weighting $(p, CT^g[p])$ in $\mathcal{M}^g_{Lut}$. If this new mask is sufficient to remove p then the process continues. If not, there is no possible Lut for this distance and the process stops, reporting the problem (this should never happen if d actualy induces a norm).

4.3 Related Algorithms

The full algorithm given figure 11, uses two other algorithms given figures 12 and 13. They are dedicated versions (limited to $\mathcal{S}$ and thus 48 times faster) of the classical Distance Transform and Medial Axis Extraction.

Note that the use of DT^g (figure 11, line 7) is mandatory, since the MA is extracted from the DT to the complementary (see §2 and §3). In fact, a simple threshold on image CT^g to the radius R gives only the $B_d(O, R) \cap \mathcal{S}$ set, but not the correct labels (see figure 6, where values of (a) differ from (b)).

```
PROCEDURE ComputeAndVerifyLut ( $L$, $\mathcal{M}^g_{Lut}$, $R_{Known}$, $R_{Target}$ ) ;
1  Compute$CT^g$($L$, $\mathcal{M}^g_C$, $Cone$) ;
2  $R_{max}$ = MAX $\{Cone[p] : p \in Cone\}$ ;
3  FOREACH $\overrightarrow{v}^g$ IN $\mathcal{M}^g_{Lut}$ DO ComputeLUTColumn ( $\overrightarrow{v}^g$, $R_{max}$, $Lut$ ) ;
4  FOR $R = R_{Known} + 1$ TO $R_{Target}$ DO
5   {
6     $\forall p,\ DT^g[p] = \begin{cases} 1 \ if\ Cone[p] \le R \\ 0 \ else \end{cases}$ ;        // Copy $B_d(R) \cap \mathcal{S}$ to $DT^g$
7     Compute$DT^g$($\mathcal{M}^g_C$, $DT^g$) ;
8     FOR $x = 1$ TO $L$ , FOR $y = 0$ TO $x$ , FOR $z = 0$ TO $y$ DO
9       IF IsMA ( $(x,y,z)$, $\mathcal{M}^g_{Lut}$, $Lut$, $DT^g$ ) THEN
10        {
11          $M = (x,y,z,Cone[(x,y,z)])$ ;                // Build a new weighting $M$
12          $\mathcal{M}^g_{Lut} = \mathcal{M}^g_{Lut} \cup \{M\}$ ;                    // Add $M$ to $\mathcal{M}^g_{Lut}$
13          ComputeLUTColumn( $(x,y,z)$, $R_{max}$, $Lut[(x,y,z)]$ ) ;    // and $Lut$
14          IF IsMA ( $(x,y,z)$, $\mathcal{M}^g_{Lut}$, $Lut$, $DT^g$ ) THEN ERROR ;
15        }
16   }
17 $R_{Known} = R_{Target}$ ;
```

Fig. 11. Full *Lut* Computation Algorithm with determination of $\mathcal{M}^g_{Lut}$. Input: L the side length of the images, $\mathcal{M}^g_{Lut}$ the generator of the *Lut* neighbourhood, R_{Known} the last verified radius, R_{Target} the maximum radius to be verified. Output: *Lut* the lookup table, R_{Known} the new verified radius

5 Results

While the computation of the *Lut* array in figure 10 is very fast (less than a second[1]), the computation of $\mathcal{M}^g_{Lut}$ in figure 11, involving its verification, is rather slow, as shown below, and its result should hence be saved for further usage. It takes approximatively six minutes (370s[1]) to compute the $\mathcal{M}^g_{Lut}$ for distance $d_{11,16,19,\mathsf{j}=45}$ for $L = 100$ and from $R_{Known} = 0$ to $R_{Target} = 1100$ (the radius of the biggest $B_d \cap \mathcal{S}$ possible in the L^3 sized image CT^g). This load is explained by the systematic test of 1100 balls $B_d(r)$. Each of them involves computations (CT^g and MA extraction) on $o(r^3)$ points. It is therefore much more interesting to use chamfer distances with small values of $W[\mathsf{a}]$ 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 by more weightings in $\mathcal{M}^g_C$. For example, the extraction of $\mathcal{M}^g_{Lut}$ for distance $d_{7,10,12,\mathsf{d}=16,\mathsf{e}=17}$ for $L = 100$ from $R_{Known} = 0$ to $R_{Target} = 700$ is 33% faster, while achieving a better approximation of d_E.

We give in figure 14 some examples of *Lut* arrays where $\mathcal{M}^g_C = \mathcal{M}^g_{Lut}$. The tables show only the values which differ from the LMC (ie. $R + W[\overrightarrow{v}^g] \neq Lut[\overrightarrow{v}^g][R]$, see (4) and (5)), and thus represent the irregularities.

[1] On a SGI Octane/IRIX 6.5, Mips R10000 180 MHz

```
PROCEDURE Compute$DT^g$ ( $DT^g$, $L$, $\mathcal{M}^g_C$ ) ;
FOR $z = L-1$ TO 0 , FOR $y = L-1$ TO $z$ , FOR $x = L-1$ TO $y$ DO
   IF $DT^g[(x,y,z)] \neq 0$ THEN
    {
     $min = +\infty$ ;
     FOREACH $\overrightarrow{v}^g$ IN $\mathcal{M}^g_C$ DO
      {
        IF $(x,y,z) + \overrightarrow{v}^g \in \mathcal{S}$ THEN $w = DT^g[(x,y,z) + \overrightarrow{v}^g] + W(\overrightarrow{v}^g)$
                                 ELSE $w = +\infty$ ;
        $min =$ MIN $\{w, min\}$ ;
      }
     $DT^g[(x,y,z)] = min$ ;
    }
```

Fig. 12. Fast Distance Transform in $\mathcal{S}$. Input: DT^g the shape (limited to $\mathcal{S}$), L the side length, $\mathcal{M}^g_C$ the generator of the d_C mask. Output: DT^g the distance map

```
FUNCTION IsMA ( $p$, $\mathcal{M}^g_{Lut}$, $Lut$, $DT^g$ ) ;
FOREACH $\overrightarrow{v}^g$ IN $\mathcal{M}^g_{Lut}$ DO
   IF $DT^g[p + \overrightarrow{v}^g] < Lut[\overrightarrow{v}^g][DT^g[p]]$ THEN RETURN FALSE ;
RETURN TRUE ;
```

Fig. 13. Fast extraction of MA points from $B_d \cap \mathcal{S}$. Input: p the point to test, $\mathcal{M}^g_{Lut}$ the generator of the Lut neighbourhood, Lut the lookup 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

The figure 15 shows a full example of both the computed mask $\mathcal{M}^g_{Lut}$ and the Lut array for distance $d_{11,16,19,\mathsf{j}=45}$. One must note the difference between $\mathcal{M}^g_C$ and $\mathcal{M}^g_{Lut}$ and the presence of point 3e which is not a visible point, and thus would not have appeared in $\mathcal{M}^g_C$ as seen in section 2. We finally give in figure 16 the $\mathcal{M}^g_{Lut}$ for distance $d_{19,27,33,\mathsf{d}=43,\mathsf{e}=47}$, which is as in figure 15, very different from its corresponding $\mathcal{M}^g_C$.

6 Conclusion

In this paper, we give the solution to the fundamental problem pointed out in [12], where the computed lookup tables detected erroneous MA points for large masks. We introduce a new mask $\mathcal{M}_{Lut}$, different from the mask $\mathcal{M}_C$ used to compute DT. We present and justify our algorithm to compute $\mathcal{M}_{Lut}$ and Lut. The correctness of the method is verified during the computation of $\mathcal{M}_{Lut}$ and thus ensures that the medial axis is free of the error points encountered in [12]. We use the symetry of the ball of chamfer norms to minimize computations. Finally, we give results for various chamfer norms given in [8].

$d_{3,4,5}$

R	a	b	c
3	4	5	6

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

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

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

R	a	b	c	e
7	8	11	14	19
10	15	18	19	26
13	18	21	24	29
14	19			

R	a	b	c	e
17	22	25	27	33
18			29	
20	26	29	32	37
23	29	32	34	40

R	a	b	c	e
24	29			
25			37	
26			37	
27	33			

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

R	c
35	47
36	47
38	50
43	55

R	c
46	58
48	60
56	68

Fig. 14. Examples of *Lut* of 4 d_C for which $\mathcal{M}^g_C = \mathcal{M}^g_{Lut}$

Acknowledgements

This work was performed within the framework of the joint incentive action "Beating Heart" of the research groups ISIS, ALP and MSPC of the French National Center for Scientific Research (CNRS).

References

1. C. Arcelli and G. Sanniti di Baja. Finding local maxima in a pseudo-euclidean distance transform. Computer Vision, Graphics and Image Processing, 43:361–367, 1988. 421
2. G. Borgefors. Distance transformations in arbitrary dimensions. Computer Vision, Graphics and Image Processing, 27:321–345, 1984. 419
3. G. Borgefors. Centres of maximal disks in the 5-7-11 distance transform. In 8th Scandinavian Conf. on Image Analysis, pages 105–111, Tromsø, Norway, 1993. 422
4. G. Borgefors, I. Ragnemalm, and G. Sanniti di Baja. The Euclidean Distance Transform : finding the local maxima and reconstructing the shape. In 7th Scandinavian Conf. on Image Analysis, volume 2, pages 974–981, Aalborg, Denmark, 1991. 422
5. G. H. Hardy and E. M. Wright. An introduction to the theory of numbers. Oxford University Press, fifth edition, October 1978. §3.1. 420

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

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			121	136	151		192	
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. 15. $\mathcal{M}^g_{Lut}$ and Lut for $d_{11,16,19,\mathsf{j}=45}$

6. M. Hujter and B. Vizvari. The exact solutions to the Frobenius problem with three variables. Ramanujan Math. Soc., 2(2):117–143, 1987. 421
7. J. L. Mari and J. Sequeira. Using implicit surfaces to characterize shapes within digital volumes. In RECPAD'00, pages 285–289, Porto, Portugal, May 2000. 419
8. E. Remy and E. Thiel. Optimizing 3D chamfer masks with distance constraints. In 7^{th} IWCIA, Int. Workshop on Combinatorial Image Analysis, pages 39–56, Caen, July 2000. 419, 420, 423, 424, 427
9. E. Remy and E. Thiel. Structures dans les sphères de chanfrein. In $12^{ème}$ RFIA, congrès Reconnaissance des Formes et I.A, volume 1, pages 483–492, Paris, Fev 2000. 420, 424

$$\mathcal{M}^g_C = \left\{ \begin{array}{l} \mathsf{a} = (1,0,0,19) \\ \mathsf{b} = (1,1,0,27) \\ \mathsf{c} = (1,1,1,33) \\ \mathsf{d} = (2,1,0,43) \\ \mathsf{e} = (2,1,1,47) \end{array} \right\} \qquad \mathcal{M}^g_{Lut} = \left\{ \begin{array}{lr} \mathsf{a} = (1,0,0,19) & \mathsf{d} = (2,1,0,43) \\ \mathsf{b} = (1,1,0,27) & \mathsf{i} = (3,2,0,70) \\ \mathsf{c} = (1,1,1,33) & v_{17} = (5,3,0,113) \\ \mathsf{e} = (2,1,1,47) & \end{array} \right\}$$

Fig. 16. $\mathcal{M}^g_{Lut}$ for $d_{19,27,33,\mathsf{d}=43,\mathsf{e}=47}$

10. A. Rosenfeld and J. L. Pfaltz. Sequential operations in digital picture processing. Journal of ACM, 13(4):471–494, 1966. 420, 421
11. J. Sylvester. Mathematicals questions with their solutions. Educational Times, 41:21, 1884. 421
12. E. Thiel. Les distances de chanfrein en analyse d'images : fondements et applications. PhD thesis, UJF, Grenoble, Sept 1994. `http://www.lim.univ-mrs.fr/~thiel/these` . 419, 421, 422, 427
13. J. H. Verwer. Distance transforms: metrics, algorithms and applications. PhD thesis, Technische Universiteit, Delft, 1991. 419, 425