

Discrete bisector function and Euclidean skeleton in 2D and 3D

Michel Couprie^(a,b) and Rita Zour^(a,b)

(a) Laboratoire A2SI, Groupe ESIEE
BP99, 93162 Noisy-le-Grand Cedex France

(b) IGM, Unité Mixte de Recherche CNRS-UMLV-ESIEE UMR 8049
m.couprie@esiee.fr

Abstract. In this paper, we propose a new definition and an exact algorithm for the discrete bisector function, which is an important tool for analyzing and filtering Euclidean skeletons. We also introduce a new thinning algorithm which produces homotopic discrete Euclidean skeletons. These algorithms, which are valid both in 2D and 3D, are integrated in a skeletonization method which is based on exact transformations, allows the filtering of skeletons, and is computationally efficient.

1 Introduction

The notion of skeleton plays a major role in shape analysis. It has been introduced by Blum [5] in 1961 and is the subject of an abundant literature, which deals with both metrical and topological aspects (see e.g. [1, 2, 6, 11, 12, 14, 15, 17, 19, 23, 25]). In this paper, we focus on skeletons in the discrete grid \mathbb{Z}^2 or \mathbb{Z}^3 , which are centered in the shape with respect to the Euclidean distance, and which have the same topology as the original shape.

Introduced by Talbot and Vincent [23] and generalizing a notion proposed by Meyer [16], the bisector function can play an important role in analyzing and filtering skeletons [1, 2, 17]. Informally, the bisector function associates to each object point x the maximal angle formed by x (as the vertex) and the points of the background which are nearest from x . Until now, the algorithms proposed to compute the bisector function in \mathbb{Z}^2 were based on the use of vectors produced by distance transformation algorithms (e.g., [7]). To each object point, with such algorithms, only one vector indicates the location of a closest background point, and some other points at the same distance may be ignored. In fact, the aim of previous approaches was to compute an approximation of the bisector function as defined in a continuous framework.

The medial axis of an object X is composed by the centers of the balls which are included in X but which are not included in any other ball included in X . This set of points is, by nature, centered in the object with respect to the distance which is used to define the notion of ball. In discrete spaces, the medial axis has not, in general, the same topology as the original object. Thus, the use of guided and constrained discrete homotopic transformations has been

proposed by several authors to obtain an homotopic skeleton which contains the medial axis. Among them, Talbot and Vincent [23] propose an algorithm for the 2D case which guarantees that the skeleton branches follow lines of steepest slope of the Euclidean distance map of the object, a property which is verified in the continuous framework. This algorithm cannot be straightforwardly extended to the 3D case.

This paper contains two original contributions. First, we propose a new definition and an exact algorithm to compute a discrete bisector function. This algorithm was inspired by the methods recently introduced [6, 19] to compute the exact Euclidean medial axis, and is also based on a pre-computed look-up table. Second, we introduce a new thinning method which produces homotopic discrete Euclidean skeletons. Unlike previously proposed approaches, this method is also valid in 3D.

Using these two original algorithms, we propose an integrated skeletonization method for objects in \mathbb{Z}^2 and \mathbb{Z}^3 which is based on exact metrical and topological transformations, allows the filtering of the skeleton under the control of two parameters, and is computationnaly efficient.

2 Basic notions

In this section, we recall some basic metrical and topological notions for binary images [9, 14]. For the sake of simplicity, we limit this presentation to the 2D case.

We denote by \mathbb{Z} the set of integers, by \mathbb{N} the set of nonnegative integers, and by \mathbb{N}^* the set of strictly positive integers. We denote by E the discrete plane \mathbb{Z}^2 . A point x in E is defined by (x_1, x_2) with x_i in \mathbb{Z} . Let $x, y \in E$, we denote by $d^2(x, y)$ the square of the Euclidean distance between x and y , that is, $d^2(x, y) = (x_1 - y_1)^2 + (x_2 - y_2)^2$. Let $Y \subset E$, we denote by $d^2(x, Y)$ the square of the Euclidean distance between x and the set Y , that is, $d^2(x, Y) = \min\{d^2(x, y); y \in Y\}$. Let $X \subset E$ (the ‘‘object’’), we denote by D_X^2 the map from E to \mathbb{N} which associates, to each point x of E , the value $D_X^2(x) = d^2(x, \overline{X})$, where \overline{X} denotes the complementary of X (the ‘‘background’’). The map D_X^2 is called the (*squared Euclidean*) *distance map of X* . Let $x \in E, r \in \mathbb{N}^*$, we denote by $B_r(x)$ the *ball of (squared) radius r centered on x* , defined by $B_r(x) = \{y \in E, d^2(x, y) < r\}$. Notice that, for any point x in X , the value $D_X^2(x)$ is precisely the radius of a ball centered on x and included in X , which is not included in any other ball centered on x and included in X .

Let us recall the notion of medial axis (see also [18, 23]). Let $X \subseteq E, x \in X, r \in \mathbb{N}^*$. A ball $B_r(x) \subseteq X$ is *maximal for X* if it is not strictly included in any other ball included in X . The *medial axis of X* , denoted by $\text{MA}(X)$, is the set of the centers of all the maximal balls for X (see Fig. 1d, see also Figs. 2, 3).

Efficient algorithms have been proposed to compute exact squared Euclidean distance maps [20, 21], and also to extract the exact Euclidean medial axis of a shape, from an exact squared Euclidean distance map and using pre-computed look-up tables [6, 19].

In discrete spaces, it is well known that the topology of the medial axis is generally not the same as the topology of the original object. In particular, if X is connected, $\text{MA}(X)$ is generally not connected (see Fig. 1d). Let us now introduce some topological notions in \mathbb{Z}^2 that we use in the sequel.

We consider the two adjacency relations Γ_4 and Γ_8 defined by, for each point $x \in E$: $\Gamma_4(x) = \{y \in E; |y_1 - x_1| + |y_2 - x_2| \leq 1\}$, $\Gamma_8(x) = \{y \in E; \max(|y_1 - x_1|, |y_2 - x_2|) \leq 1\}$. In the following, we will denote by n a number such that $n = 4$ or $n = 8$. We define $\Gamma_n^*(x) = \Gamma_n(x) \setminus \{x\}$. The point $y \in E$ is n -adjacent to $x \in E$ if $y \in \Gamma_n^*(x)$; it is n -adjacent to a subset X of E if it is n -adjacent to at least one point of X . An n -path is a sequence of points $x_0 \dots x_k$ with x_i n -adjacent to x_{i-1} for $i = 1 \dots k$.

Let X be a nonempty subset of E . We say that two points x, y of X are n -connected in X if there is an n -path in X between these two points. This defines an equivalence relation. The equivalence classes for this relation are the n -connected components of X , or n -components in short. The set X is said to be n -connected if it consists of exactly one n -connected component. The set composed of all n -connected components of X which are n -adjacent to a point x is denoted by $C_n[x, X]$.

In order to have a correspondence between the topology of X and the topology of \bar{X} , we have to consider two different kinds of adjacency for X and \bar{X} [14]: if we use the n -adjacency for X , we must use the \bar{n} -adjacency for \bar{X} , with $(n, \bar{n}) = (8, 4)$ or $(4, 8)$. In the sequel, we assume that the adjacency pair $(n, \bar{n}) = (8, 4)$ has been chosen and we do not write the subscripts n, \bar{n} unless necessary; but the results also hold for $(n, \bar{n}) = (4, 8)$.

Informally, a simple point p of a discrete object X is a point which is “inessential” to the topology of X . In other words, we can remove the point p from X without “changing the topology of X ”. The notion of simple point is fundamental to the definition of topology-preserving transformations in discrete spaces. We now give a definition and a local characterization of simple points in $E = \mathbb{Z}^2$. For the 3D case, see [3].

The point $x \in X$ is *simple (for X)* if each n -component of X contains exactly one n -component of $X \setminus \{x\}$ and if each \bar{n} -component of $\bar{X} \cup \{x\}$ contains exactly one \bar{n} -component of \bar{X} . Let $X \subseteq E$ and $x \in E$, the two *connectivity numbers* are defined as follows ($\#X$ stands for the cardinality of X):

$$T(x, X) = \#C_n[x, \Gamma_8^*(x) \cap X]; \quad \bar{T}(x, X) = \#C_{\bar{n}}[x, \Gamma_8^*(x) \cap \bar{X}].$$

The following property allows us to locally characterize simple points [14, 3], hence to implement efficiently topology preserving operators:

$$x \in E \text{ is simple for } X \subseteq E \Leftrightarrow T(x, X) = 1 \text{ and } \bar{T}(x, X) = 1.$$

Let X be any finite subset of E . The subset Y of E is an *homotopic thinning of X* if $Y = X$ or if Y may be obtained from X by iterative deletion of simple points. We say that Y is an *ultimate homotopic skeleton of X* if Y is an homotopic thinning of X and if there is no simple point for Y .

Let C be a subset of X . We say that Y is an *ultimate homotopic skeleton of X constrained by C* if $C \subseteq Y$, if Y is an homotopic thinning of X and if there is no

simple point for Y in $Y \setminus C$ (see e.g. [11, 25]). The set C is called the *constraint set* relative to this skeleton.

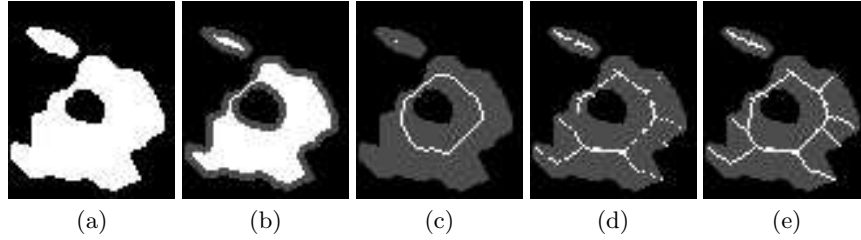


Fig. 1. (a): a set X (in white); (b): an homotopic thinning of X ; (c): ultimate homotopic skeleton of X ; (d): medial axis of X ; (e): ultimate homotopic skeleton of X constrained by the medial axis of X . In (b,c,d,e) the original set X appears in gray for comparison.

3 The bisector function: new definition and exact algorithm

Let X be a nonempty subset of E , and let $x \in X$. The *downstream of x in X* , denoted by $Ds(x, X)$ or by $Ds(x)$ when no confusion may occur, is the set of points y of \overline{X} which are at minimal distance from x ; more precisely, $Ds(x, X) = \{y \in \overline{X}, \forall z \in \overline{X}, d^2(y, x) \leq d^2(z, x)\}$. For example in Fig. 2, we have $Ds(x) = \{a, b\}$ and $Ds(y) = \{c\}$. The bisector angle of a point x in X can be defined informally, in the continuous framework, as the maximal unsigned angle formed by x (as the vertex) and any two points in $Ds(x)$. In Fig. 2, the bisector angle of x is α , and the bisector angle of y is 0.

In the continuous framework, a point x which belongs to the medial axis is characterized by the fact that $Ds(x)$ contains strictly more than one point, in other words, its bisector angle is nonzero. In the discrete case such a characterization is not possible, due to configurations like Fig. 3, which are very common. At least, we want such medial axis points to have a nonzero bisector angle; this motivates the introduction of the following notion.

Let $X \subset E$, and let $x \in X$. The *extended downstream of x in X* , denoted by $EDs(x, X)$ or by $EDs(x)$ when no confusion may occur, is the union of the sets $Ds(y, X)$, for all y in $\Gamma_4(x)$.

Now we can propose our definition of the discrete bisector function.

Let $X \subset E$, and let $x \in X$. The *bisector angle of x in X* , denoted by $\theta_X(x)$, is the maximal unsigned angle between the vectors \vec{xy}, \vec{xz} , for all y, z in $EDs(x)$. In particular, if $\#EDs(x, X) = 1$, then $\theta_X(x) = 0$. The *bisector function of X* , denoted by θ_X , is the function which associates to each point x of X , its bisector angle in X .

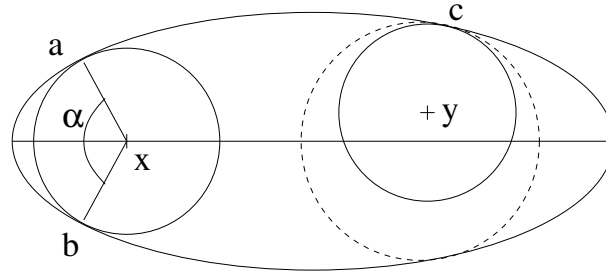


Fig. 2. A set X (full ellipsis, represented by its border) and its medial axis (horizontal line segment), a point x and its downstream $\{a, b\}$, a point y and its downstream $\{c\}$. Notice that y does not belong to the medial axis, since no ball centered on y and included in X is maximal for X (the ball with dashed contour contains them all).

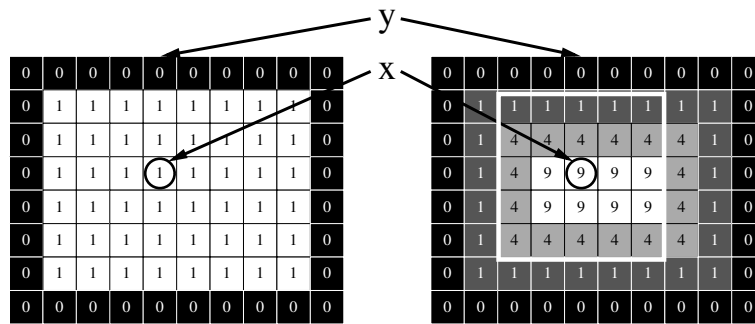


Fig. 3. A set X (left, in white) and its distance map D_X^2 (right). The point x belongs to the medial axis of X (it is the center of a maximal ball for X , delineated in white). The downstream of x is the singleton $\{y\}$.

With this definition, we can easily check that the bisector angle of the point x in Fig. 3 is equal to π . More generally, we may wonder if we can prove that any point of the medial axis has a nonzero bisector angle. It is unfortunately not the case, as shown by the counter-example of Fig. 4. Similar counter-examples can be found if we replace Γ_4 by Γ_8 in the definition of the extended downstream. On the other hand, we will see in the sequel that points of the medial axis which have the smallest bisector angle are the best candidates for elimination in a bisector-based filtering process. Those points which have a null bisector angle will thus not belong to any medial axis filtered using this criterion.

We now propose a method to exactly and efficiently compute the bisector function, with the help of a squared Euclidean distance map. The following property, which may be easily established, is the key of the method.

| | | | | | | | | | | | | | | | | | | | |
|----|-----|----|-----|----|-----|----|-----|----|-----|-----|----|----|-----|-----|----|-----|-----|-----|-----|
| 45 | 40 | 37 | 34 | 29 | 25 | 20 | 17 | 13 | 10 | 8 | 5 | 4 | 2 | 1 | 0 | 0 | 0 | 0 | 0 |
| 58 | 53 | 50 | 45 | 40 | 34 | 29 | 25 | 20 | 17 | 13 | 10 | 8 | 5 | 2 | 1 | 1 | 0 | 0 | 0 |
| 73 | 68 | 65 | 58 | 52 | 45 | 40 | 34 | 29 | 25 | 20 | 17 | 13 | 8 | 5 | 4 | 2 | 1 | 0 | 0 |
| 90 | 85 | 80 | 73 | 65 | 58 | 52 | 45 | 40 | 34 | 29 | 25 | 18 | 13 | 10 | 8 | 5 | 2 | 1 | 1 |
| 97 | 104 | 97 | 89 | 80 | 73 | 65 | 58 | 52 | 45 | 40 | 32 | 25 | 20 | 17 | 13 | 8 | 5 | 4 | 2 |
| 80 | 89 | 97 | 106 | 97 | 89 | 80 | 73 | 65 | 58 | 50 | 41 | 34 | 29 | 25 | 18 | 13 | 10 | 8 | 5 |
| 65 | 73 | 80 | 89 | 97 | 106 | 97 | 89 | 80 | 72 | 61 | 52 | 45 | 40 | 32 | 25 | 20 | 17 | 13 | 8 |
| 52 | 58 | 65 | 73 | 80 | 89 | 97 | 106 | 97 | 85 | 74 | 65 | 58 | 50 | 41 | 34 | 29 | 25 | 18 | 13 |
| 40 | 45 | 52 | 58 | 65 | 73 | 80 | 89 | 97 | 100 | 89 | 80 | 72 | 61 | 52 | 45 | 40 | 32 | 25 | 18 |
| 29 | 34 | 40 | 45 | 52 | 58 | 65 | 73 | 80 | 89 | 100 | 97 | 85 | 74 | 65 | 58 | 50 | 41 | 32 | 25 |
| 20 | 25 | 29 | 34 | 40 | 45 | 52 | 58 | 65 | 74 | 65 | 58 | 50 | 41 | 34 | 29 | 25 | 18 | 13 | 10 |
| 13 | 17 | 20 | 25 | 29 | 34 | 40 | 45 | 52 | 61 | 72 | 85 | 97 | 106 | 97 | 85 | 72 | 61 | 52 | 45 |
| 8 | 10 | 13 | 17 | 20 | 25 | 29 | 34 | 41 | 50 | 61 | 72 | 80 | 89 | 100 | 98 | 85 | 74 | 65 | 58 |
| 4 | 5 | 8 | 10 | 13 | 17 | 20 | 25 | 32 | 41 | 50 | 58 | 65 | 74 | 85 | 98 | 100 | 89 | 80 | 72 |
| 1 | 2 | 4 | 5 | 8 | 10 | 13 | 18 | 25 | 32 | 40 | 45 | 52 | 61 | 72 | 85 | 98 | 106 | 97 | 85 |
| 0 | 1 | 1 | 2 | 4 | 5 | 8 | 13 | 18 | 25 | 29 | 34 | 41 | 50 | 61 | 72 | 85 | 98 | 113 | 97 |
| 0 | 0 | 0 | 1 | 1 | 2 | 5 | 8 | 13 | 17 | 20 | 25 | 32 | 41 | 50 | 61 | 72 | 85 | 98 | 106 |
| 0 | 0 | 0 | 0 | 0 | 1 | 2 | 5 | 8 | 10 | 13 | 18 | 25 | 32 | 41 | 50 | 61 | 72 | 85 | 98 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | 4 | 5 | 8 | 13 | 18 | 25 | 32 | 41 | 50 | 61 | 72 | 85 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | 5 | 8 | 13 | 18 | 25 | 32 | 41 | 50 | 61 | 72 | 85 |

Fig. 4. A portion of the squared Euclidean distance map of a set X (in white and gray). The point x , circled in black, belongs to the medial axis of X (in gray); and the point y , circled in white, belongs to \overline{X} . Knowing the following decompositions: $85 = 9^2 + 2^2 = 7^2 + 6^2$, $74 = 7^2 + 5^2$, $72 = 6^2 + 6^2$, $100 = 10^2 + 0^2 = 8^2 + 6^2$; we can check that the extended downstream of x is reduced to $\{y\}$, and thus the bisector angle of x is null.

Property 1 *Let $X \subset E$, let $x \in X$, and let $y \in E$. The point y belongs to the downstream of x if and only if $d^2(x, \overline{X}) = d^2(x, y)$ and $d^2(y, \overline{X}) = 0$.*

Now, observe that for any given point $x \in X$, the value $R = d^2(x, \overline{X}) = D_X^2(x)$ can be read in a pre-computed squared Euclidean distance map. The positions of the points y such that $R = d^2(x, y)$ can be found by solving the diophantine equation $(x_1 - y_1)^2 + (x_2 - y_2)^2 = R$, or more simply $\alpha^2 + \beta^2 = R$. This amounts to compute the different decompositions of a given integer into a sum of two squares (or three squares in 3D), a problem which has been studied extensively (see e.g. [13]).

Furthermore, these decompositions can also be pre-computed and stored in a look-up table. The program which computes this look-up table is very simple (see Annex). Once all decompositions are known, one must take into account the different symmetries of the discrete space (their number is 8 for \mathbb{Z}^2 , 48 for \mathbb{Z}^3) and check those positions y which satisfy the second condition of prop. 1: $d^2(y, \overline{X}) = 0$, again using the distance map.

The following algorithm summarizes the computation of the extended downstream of a given point $x \in X$.

Procedure ExtendedDownstream (Input D_X^2, x , Output EDs)

01. $EDs \leftarrow \emptyset$

02. **ForEach** $v \in \Gamma_4(x)$

03. $T \leftarrow \{(z_1, z_2) \in \mathbb{Z}^2; z_1 \geq z_2 \geq 0; z_1^2 + z_2^2 = D_X^2(v)\}$ (from LUT)

04. $T' \leftarrow \{(z_1, z_2) \in \mathbb{Z}^2; (|z_1|, |z_2|) \text{ is a permutation of an element of } T\}$
 05. **ForEach** $z \in T'$ **Do**
 06. **If** $D_X^2(v+z) = 0$ **Then** $EDs \leftarrow EDs \cup \{v+z\}$

The last step to obtain the bisector angle consists in the computation of the maximum unsigned angle between all the pairs of vectors $\{\vec{xy}, \vec{xz}\}$ for all y, z in $EDs(x)$. If we denote by k the number of points in $EDs(x)$, the number of such pairs is quadratic with respect to k , more precisely, it is equal to $k(k-1)/2$. By normalizing all these vectors, we can easily see that the problem of finding a maximum angle reduces to the problem of finding a maximum diameter of a convex polygon in 2D. This last problem has been solved in 1978 by Shamos [22], who provided a simple linear-time algorithm (that is, in $O(k)$). In 3D, the problem is more complicated but some efficient algorithms (in $O(k \log k)$ or less) have been proposed for the maximal diameter of a set of points, see e.g. [4]. However, in practice, the mean cardinal of the extended downstream for a given shape is usually quite small. For example, it is close to 1.9 for the 2D object of Fig. 7, and close to 4.9 for the 3D object of Fig. 9. If we consider only the extended downstream of medial axis points, these values are respectively of 2.6 and 5.4. Thus, the straightforward algorithm which considers all the pairs of points in the extended downstream is the best choice in most cases.

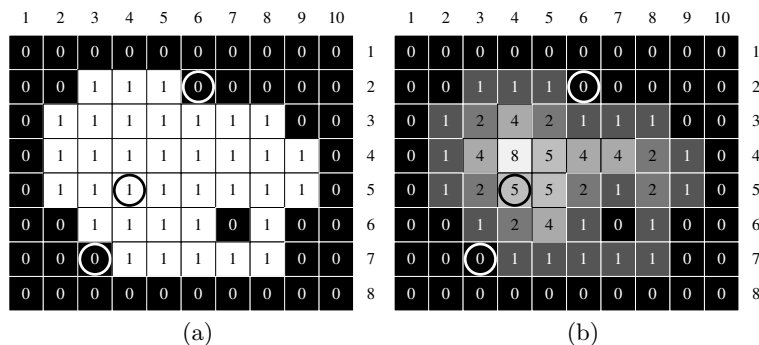


Fig. 5. (a): a set X (in white); (b): the squared Euclidean map of X . The point $x = (4, 5)$ is circled in black, the points $y_1 = (6, 2)$ and $y_2 = (3, 7)$ are circled in white.

We illustrate this procedure using the example of Fig. 5. We consider the point $x = (4, 5)$, which is circled in the figure. Thus, $T_4(x) = \{x, x_1, x_2, x_3, x_4\}$ with $x_1 = (5, 5)$, $x_2 = (4, 4)$, $x_3 = (3, 5)$, $x_4 = (4, 6)$. Let us begin with $x = (4, 5)$, and $D_X^2(x) = 5$. From the Look-up table (see 15), we see that the only decomposition of 5 into a sum of two squares is $2^2 + 1^2$. Applying the 8 symmetries of \mathbb{Z}^2 , we find the offsets $\{(2, 1), (-2, 1), (-2, -1), (2, -1), (1, 2), (-1, 2), (-1, -2), (1, -2)\}$, hence the points $\{(6, 6), (2, 6), (2, 4), (6, 4), (5, 7),$

$(3, 7), (3, 3), (5, 3)\}$. Among these points, only $(2, 6)$ and $(3, 7)$ are in \overline{X} , hence, in the downstream of x . Carrying on with the points x_1, x_2, x_3, x_4 , we find that $\text{EDs}(x) = \{(2, 6), (3, 7), (7, 6), (2, 2), (6, 2)\}$. Let $y_1 = (6, 2)$ and $y_2 = (3, 7)$, it may be easily checked that the maximum unsigned angle corresponds to the couple of vectors $\{\vec{xy}_1, \vec{xy}_2\}$ and is near to 3.02 rad.

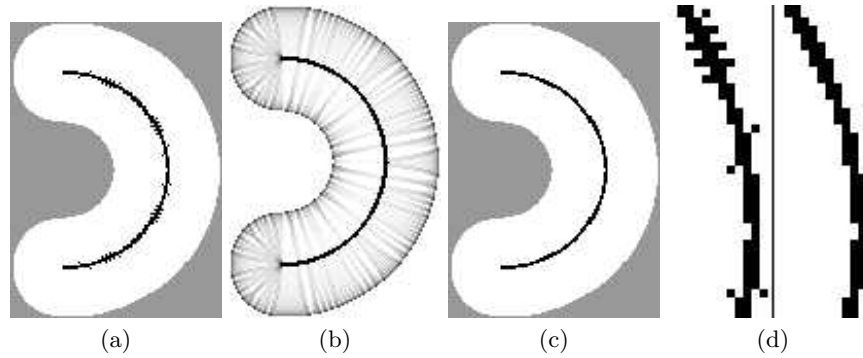


Fig. 6. (a): a set X and its medial axis (in black); (b): the bisector function θ_X (dark colors correspond to wide angles); (c): filtered medial axis, based on the values of θ_X ; (d): detail of the non-filtered and filtered medial axis.

In Fig. 6, we show a set X together with its medial axis (a) and the bisector function θ_X (b). We illustrate the use of this function to eliminate spurious points of the medial axis: in (c), we show the points of the medial axis (in black) which have a bisector angle greater than 0.7 rad. A zoomed detail of both axes is shown in (d). Notice that only the bisector angles of the medial axis points need to be computed for this application. Fig. 7 shows the bisector function of a more complex 2D shape.

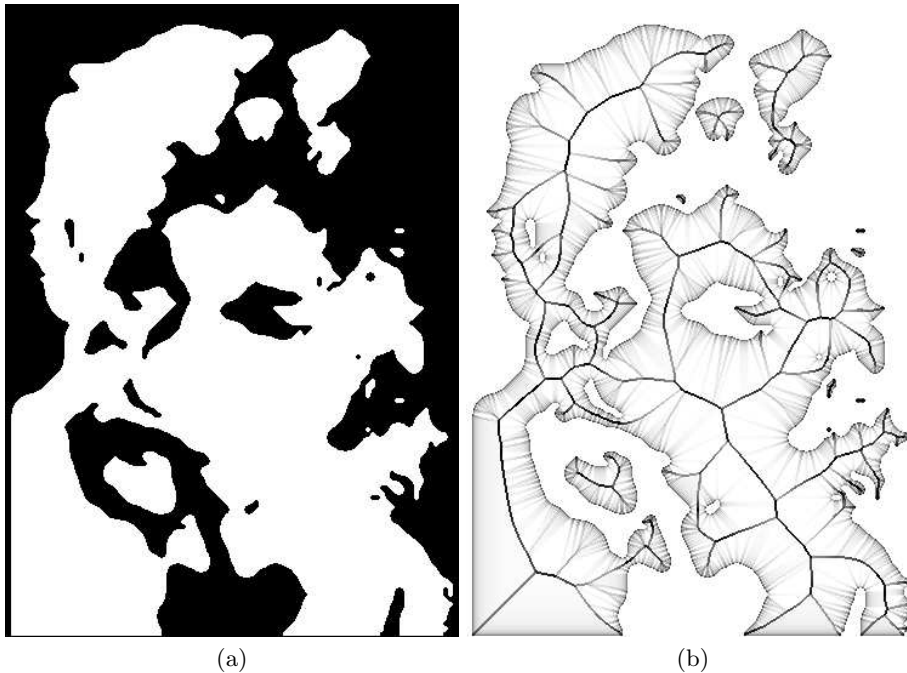


Fig. 7. (a): a set X (in white); (b): the bisector function of X .

Let us analyse the differences between the Talbot's discrete bisector function [24] and ours. Let $X \subset E, x \in X$, and let $V(x)$ denote a point arbitrarily chosen in $Ds(x)$. We set $B(x) = \max\{\text{angle}(\vec{y_1 z_1}, \vec{y_2 z_2}), y_1 \in \Gamma_4(x), y_2 \in \Gamma_4(x), z_1 = V(y_1), z_2 = V(y_2)\}$ (notice that the definition of Talbot is restricted to the points of the skeleton, and corresponds actually to $B(x)/2$). A first difference lies in the arbitrary choices done by the algorithm which finds a closest background point for each object point, formalized here by the notation $V(x)$. In our case, all the points of $Ds(x)$ are considered, thus no choice is done. The figure 8 illustrates a second difference. Observe that depending on the contour orientation, (e.g., horizontal, vertical or diagonal) Talbot's bisector angles can be very different, even for points which are not located on the contour. On the other hand, our definition and algorithm provide a more homogeneous result, whatever the contour orientation (see also Fir. 7). This is due to the choice of considering the point x itself as the only vertex for all the angles which are considered to compute $\theta_X(x)$. On the other hand, the definition of $B(x)$ is based on angles with different vertices in $\Gamma_4(x)$.

To conclude this section, we present in Fig. 9 an illustration of our bisector function of a three-dimensional object (a vertebra). The size of the original image is $122 \times 144 \times 53 = 931104$ points, among which 103302 are object points. On a standard PC, computing the bisector function on all object points takes 2.9 s, and 0.5 s for only medial axis points.

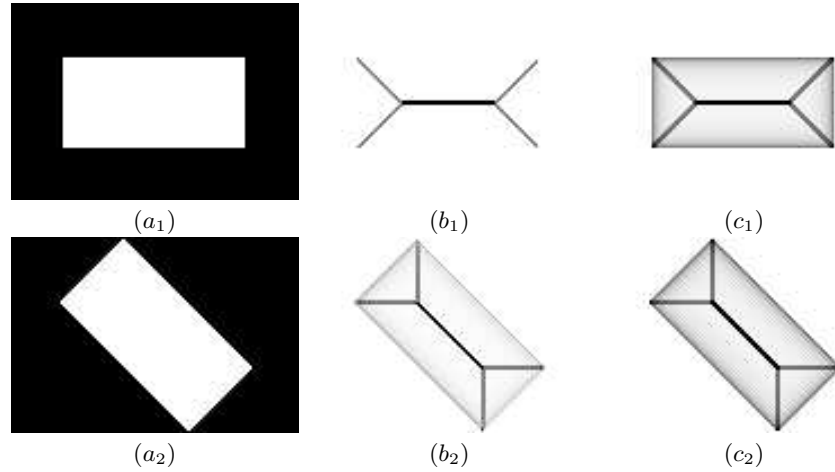


Fig. 8. (a_1, a_2) : a set X (in white); (b_1, b_2) : Talbot's bisector function; (c_1, c_2) : our bisector function.

4 New distance-guided homotopic thinning algorithm

All the methods which have been proved to guarantee that the skeleton and the original object have the same topology are based on homotopic thinnings, in the sense of section 2, or on some variants. The simplest such method consists in computing an ultimate homotopic skeleton of the object X constrained by the medial axis of X , that is, removing iteratively simple points from X which do not belong to $MA(X)$, taking the distance map as a priority function in order to select first the points which are closest to the background. This can be done using the following procedure, with $P = D_X$ and $Y = MA(X)$.

Procedure UltimateSkeleton (Input X, P, Y , Output Z)

01. $Z \leftarrow X$
02. $Q \leftarrow \{(P(x), x); \text{ where } x \text{ is any point of } X \setminus Y\}$
03. **While** $Q \neq \emptyset$ **Do**
04. choose (p, x) in Q such that p is minimal
05. **If** x is simple for Z **Then**
06. $Z \leftarrow Z \setminus \{x\}$
07. $Q \leftarrow Q \cup \{(P(y), y); \text{ where } y \in F(x) \cap (Z \setminus Y)\}$

The drawback of this method has been well analyzed in [23]. Roughly speaking the method does not guarantee that points of the homotopic skeleton outside the medial axis are “well centered” in the object; more precisely, such a point may have a null or quasi-null bisector angle.

Let us consider the object X in Fig. 10a, and the constraint set Y in Fig. 10c, which idealizes a filtered medial axis of X . The distance map D_X is shown in

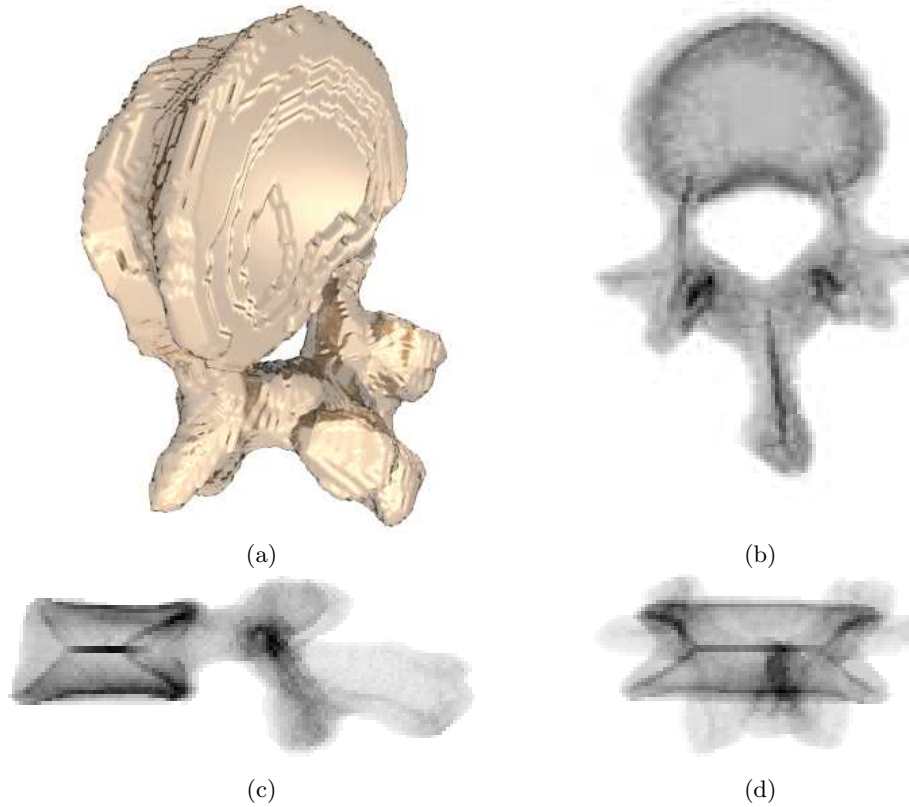


Fig. 9. (a): a view of a subset X of \mathbb{Z}^3 (vertebra), generated thanks to a topologically sound “Marching Cubes-like” algorithm [8]; (b,c,d): the bisector function, illustrated in an “X-ray” manner: the gray level of a point corresponds to the average of the bisector angles on a straight line parallel to one of the three axes.

Fig. 10b), and the result of $\text{UltimateSkeleton}(X, D_X, Y)$ is depicted in Fig. 10d. The bisector function of X is depicted in Fig. 10e for comparison.

To understand what happens, let us concentrate on a detail of the above example, depicted in Fig. 11. The numbers correspond to the distance map values. The circled point with value 1 is one of the points belonging to the constraint set Y . Suppose that all the points with a value below 8 have been processed by the homotopic thinning algorithm. At this step, the points in gray are still in X , as well as the two circled points (the point at 1 because it belongs to Y , and the one at 4 because it is not a simple point). All other points are not in X . Obviously, the point v at 8 adjacent to z at 4 will be selected before its neighbor w at 9, and since it will be a simple point at this stage, it will be deleted. Such a behaviour will be reproduced at later stages, generating a diagonal branch of the skeleton, and is in contradiction with a property of the

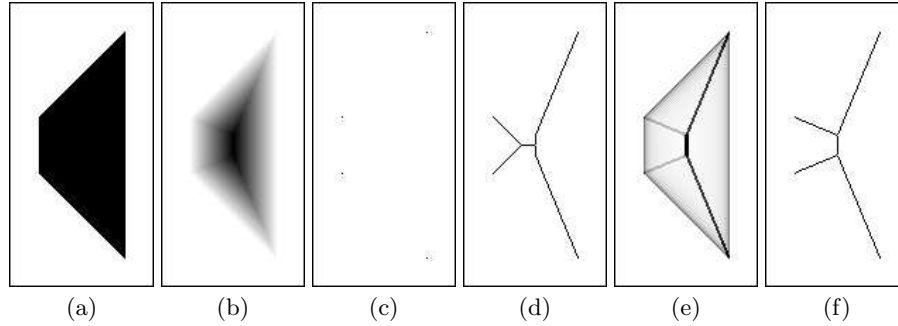


Fig. 10. (a): a set X (in black); (b): the squared Euclidean map D_X^2 ; (c): a constraint set Y (four black points); (d): homotopic skeleton of X guided by D_X^2 and constrained by Y ; (e): the bisector function of X ; (f): the result of our algorithm `EuclideanSkeleton`.

| | | | | | | |
|---|---|---|---|----|----|-----|
| 0 | 0 | 1 | 2 | 5 | 8 | v |
| 0 | 1 | 2 | 5 | 8 | 13 | |
| 0 | 1 | 4 | 8 | 13 | 18 | |
| 0 | 1 | 4 | 9 | 16 | 25 | w |
| 0 | 1 | 4 | 9 | 16 | 25 | |

Fig. 11. A detail of the preceding example.

skeleton in the continuous framework: informally, skeleton branches follow lines of steepest slope of the Euclidean distance map. Let us compute the slopes of zv and zw in our example: $(\sqrt{8} - \sqrt{4})/1 \approx 0.83$, and $(\sqrt{9} - \sqrt{4})/\sqrt{2} \approx 0.71$. Thus, the point v should be kept in the skeleton and the point w deleted according to this criterion.

In [23], Talbot and Vincent propose the following strategy to cope with this problem. During the thinning process guided by the distance map, having detected a point x which belongs to the skeleton (either a point of the constraint set or a non-simple point), the neighbor of x which corresponds to the steepest ascending slope is dynamically added to the constraint set. Although this method gives satisfactory results in 2D, it cannot be extended to the 3D case.

We propose another strategy which gives equivalent results in 2D and which also applies to the 3D case. The idea is to define a priority function which takes into account both the distance map and an auxiliary function defined in the neighborhood of each dynamically detected skeleton point. Let x be such a point, then to any neighbor y of x which is still in X and not in the constraint set Y , we associate the value $p_y = D_X(x) + (D_X(y) - D_X(x))/d(x, y)$, with $D_X(x) = \sqrt{D_X^2(x)}$ and $d(x, y) = \sqrt{d^2(x, y)}$. The new priority function, for the point y , is defined as $\min(p_y, D_X(y))$. We see that $(D_X(y) - D_X(x))/d(x, y)$ is the slope of xy , thus the neighbors of x will be examined in increasing order of

slope, since the value p_y is always less or equal to the corresponding distance value $D_X(y)$ (for all x, y in \mathbb{Z}^2 or \mathbb{Z}^3 with $x \neq y$, we have $d(x, y) \geq 1$).

For example, in the above situation, we have $D_X(v) = \sqrt{8} \approx 2.83$, $D_X(w) = 3$, $p_v = \sqrt{4} + (\sqrt{8} - \sqrt{4})/1 = \sqrt{8}$ and $p_w = \sqrt{4} + (\sqrt{9} - \sqrt{4})/\sqrt{2} \approx 2.71$; thus the point w will be selected before v with this strategy. Our algorithm is given below.

Procedure EuclideanSkeleton (Input X, D_X, Y , Output Z)

01. $Z \leftarrow X$
02. $Q \leftarrow \{(D_X(x), x); \text{ where } x \text{ is any point of } X \setminus Y\}$
03. $R \leftarrow \{(p_x, x); \text{ where } x \text{ is any point of } X \setminus Y \text{ adjacent to } Y,$
04. and where $p_x = \min\{D_X(z) + (D_X(x) - D_X(z))/d(x, z), z \in Y\}$
05. **While** $Q \neq \emptyset$ **Or** $R \neq \emptyset$ **Do**
06. choose (p, x) in $Q \cup R$ such that p is minimal
07. **If** $x \in Z \setminus Y$ **Then**
08. **If** x is simple for Z **Then**
09. $Z \leftarrow Z \setminus \{x\}$
10. **Else**
11. $Y \leftarrow Y \cup \{x\}$
12. $R \leftarrow R \cup \{(p_y, y); \text{ where } y \in \Gamma(x) \cap (Z \setminus Y)$
13. and where $p_y = D_X(x) + (D_X(y) - D_X(x))/d(x, y)\}$

In Fig. 10f, we see the result of this algorithm applied to the preceding example. Compare the shape of this skeleton to the bisector function of X depicted in 10e. The complexity of this algorithm depends on the data structure used to represent the sets Q and R . To be more precise, this data structure must allow to perform the choice in line 6 efficiently, and also the insertions in lines 11 and 12. Using for example balanced binary trees [10], the overall complexity of the algorithm is in $O(n \log n)$, where n is the number of image points.

5 Filtered Euclidean skeletons in 2D and 3D

In [2], Attali and Montanvert show that an efficient filtering of a skeleton can be performed by removing two kinds of medial axis points: those which correspond to maximal balls of small radius, and those which have a small bisector angle (see also [1]).

Based on the methods described in the previous sections, we propose a parametrized filtered Euclidean skeleton procedure which has the following qualities:

- flexible (it allows the removal of unwanted skeleton parts under the control of two parameters),
- general (it applies to 2D and 3D objects with arbitrary topology),
- exact (it is based on exact distance transform, exact medial axis and exact bisector function, and the topology preservation is guaranteed by the use of homotopic transformations),
- efficient in terms of processing speed.

Procedure FilteredSkeleton (Input X, r, α , Output Z)

01. $D_X \leftarrow \mathbf{ExactSquaredEuclideanDistanceMap}(X)$
02. $M \leftarrow \mathbf{ExactMedialAxis}(X, D_X)$
03. $Z \leftarrow \mathbf{EuclideanSkeleton}(X, D_X, M)$
04. $\theta \leftarrow \mathbf{ExactBisector}(M, D_X)$
05. $Y \leftarrow \{x \in M; D_X(x) \geq r \text{ and } \theta(x) \geq \alpha\}$
06. $Z \leftarrow \mathbf{UltimateSkeleton}(Z, D_X, Y)$

The exact squared Euclidean map may be computed in linear time, both in 2D and 3D, by the Hirata’s algorithm [21]. It only has to be computed once and is used by procedures **ExactMedialAxis**, **ExactBisector** and **EuclideanSkeleton**. Procedure **ExactMedialAxis** can be efficiently implemented in 2D and 3D thanks to the method of Rémy and Thiel [19]. Procedures **ExactBisector** and **EuclideanSkeleton** are described in sections 3 and 4, respectively. The bisector function θ need only to be computed on points of the medial axis.

Fig. 12 illustrates the **FilteredSkeleton** procedure in 3D. The original object (a full ellipsoid, depicted as transparent in the three views) is connected, has no cavity and no tunnel. Besides, its medial axis (a) has several small isolated components. The Euclidean skeleton has exactly the same topology as the original image, thus we can verify on b that the result is connected. A filtered skeleton is shown in c .

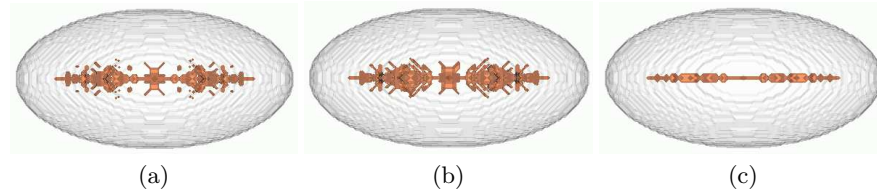


Fig. 12. (a): a view of the medial axis $M(X)$ of the set X (full 3D ellipsoid, shown as transparent on the three views); (b): a view of $\mathbf{EuclideanSkeleton}(X, D_X, M(X))$; (c): a view of $\mathbf{FilteredSkeleton}(X, 1, 1.5)$.

Fig. 13 illustrates the **FilteredSkeleton** procedure in 2D. Notice that, by construction, we have the guarantee that all the filtered skeletons are included in the non-filtered skeleton, this is indeed the reason why we use two separate homotopic thinning steps (lines 3 and 6).

Fig. 14 illustrates the **FilteredSkeleton** procedure on a “real world” 3D image (the vertebra of Fig. 9) which is connected, has no cavity and has only one tunnel. Its medial axis has several small isolated components and small tunnels (see b and the upper part of d). Again, we can verify on c and the lower part of d that the small tunnels have disappeared and that the result is connected. In e , we can judge the effectiveness of the filtering on this particular object.

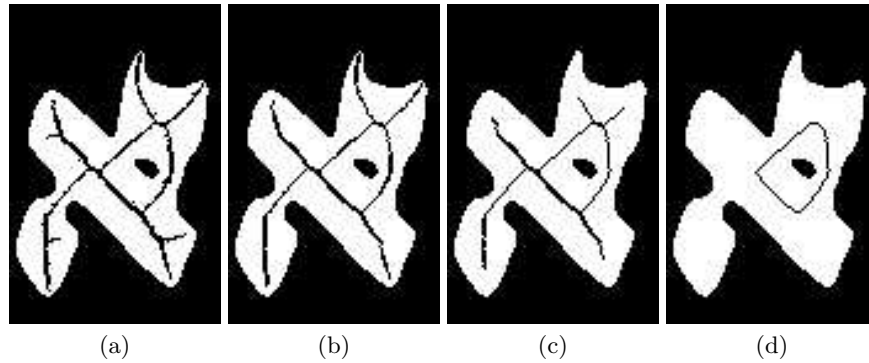


Fig. 13. (a): original shape and non-filtered skeleton; (b): filtered skeleton with parameter values $r = 0$, $\alpha = 2.0$; (c): $r = 64$, $\alpha = 2.2$; (d): $r = 100$, $\alpha = 3.14$.

6 Conclusion

We introduced a new definition and an exact algorithm for the discrete bisector function, and proposed a new thinning algorithm which produces homotopic discrete Euclidean skeletons. Both constitute significant improvements with respect to previous approaches, and apply to the 2D and 3D cases. Using these two original algorithms, together with methods for computing exact squared Euclidean distance transform [21] and exact Euclidean medial axis [19], we proposed a skeletonisation method which is flexible (with two parameters for filtering), general (for 2D and 3D), exact, and efficient. It is, to our knowledge, the first method which possesses these four qualities.

References

1. D. Attali, J.O. Lachaud: “Delaunay Conforming Iso-surface, Skeleton Extraction and Noise Removal”, *Computational Geometry: Theory and Applications*, Vol. 19, pages 175-189, 2001.
2. D. Attali, A. Montanvert: “Modelling noise for a better simplification of skeletons”, *Procs. International Conference on Image Processing*, Vol. 3, pp. 13-16, 1996.
3. G. Bertrand: “Simple points, topological numbers and geodesic neighborhoods in cubic grids”, *Pattern Recognition Letters*, Vol. 15, pp. 1003-1011, 1994.
4. S.N. Bespamyatnikh, “An efficient algorithm for the three-dimensional diameter problem”, *Procs. of the ninth annual ACM-SIAM symposium on Discrete algorithms*, pp. 137-146, 1998.
5. H. Blum, “An associative machine for dealing with the visual field and some of its biological implications”, *Biological prototypes and synthetic systems*, Vol. 1, pp. 244-260, 1961.
6. G. Borgefors, I. Ragnemalm, G. Sanniti di Baja, “The Euclidean distance transform: finding the local maxima and reconstructing the shape”, *Procs. of the 7th Scand. Conf. on image analysis*, Vol. 2, pp. 974-981, 1991.

7. P.E. Danielsson: "Euclidean distance mapping", *Computer Graphics and Image Processing*, 14, pp. 227-248, 1980.
8. X. Daragon, M. Couprie, G. Bertrand: "Discrete frontiers", Springer (Ed.), *Discrete Geometry for Computer Imagery*, Springer LNCS, Vol. 2886, pp. 236-245, 2003.
9. J.M. Chassery, A. Montanvert: *Géométrie discrète*, Hermès, 1991.
10. T.H. Cormen, C.E. Leiserson, R.L. Rivest: *Introduction to algorithms*, MIT Press, 1990.
11. E.R. Davies, A.P.N. Plummer: "Thinning algorithms: a critique and a new methodology", *Pattern Recognition*, Vol. 14, pp. 53-63, 1981.
12. Y. Ge, J.M. Fitzpatrick: "On the generation of skeletons from discrete Euclidean distance maps", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 11, pp. 1055-1066, 1996.
13. G.H. Hardy, E.M. Wright: *An introduction to the theory of numbers*, Oxford University Press, 1938.
14. T. Yung Kong, A. Rosenfeld: "Digital topology: introduction and survey", *Computer Vision, Graphics and Image Processing*, Vol. 48, pp. 357-393, 1989.
15. L. Lam, S-W. Lee, C.Y. Suen: "Thinning methodologies - a comprehensive survey", *IEEE PAMI*, Vol. 14, No. 9, pp. 869-885, 1992.
16. F. Meyer, "Cytologie quantitative et morphologie mathématique", PhD thesis, École des mines de Paris, 1979.
17. G. Malandain, S. Fernández-Vidal, "Euclidean Skeletons", *Image and vision computing*, Vol. 16, pp. 317-327, 1998.
18. A. Rosenfeld, A.C. Kak: *Digital Image processing*, Academic Press, 1982.
19. E. Rémy, E. Thiel: "Exact Medial Axis with Euclidean Distance", to appear in *Image and Vision Computing*, 2004.
20. T. Saito, J.I. Toriwaki: "New algorithms for Euclidean distance transformation of an n -dimensional digitized picture with applications", *Pattern Recognition*, Vol. 27, pp. 1551-1565, 1994.
21. T. Hirata: "A unified linear-time algorithm for computing distance maps", *Information Processing Letters*, Vol. 58(3), pp. 129-133, 1996.
22. M.I. Shamos: *Computational geometry*, PhD thesis, Yale University, 1978.
23. H. Talbot, L. Vincent: "Euclidean skeletons and conditional bisectors", *Proceedings of VCIP'92, SPIE*, Vol. 1818, pp. 862-876, 1992.
24. H. Talbot: "Analyse morphologique de fibres minérales d'isolation", PhD thesis, École des mines de Paris, 1993.
25. L. Vincent: "Efficient Computation of Various Types of Skeletons", *Proceedings of Medical Imaging V, SPIE*, Vol. 1445, pp. 297-311, 1991.

Annex: Look-up table for the bisector function

We give here the algorithms to build the lookup tables which are needed for computing the bisector function, and discuss the size of these tables with regard to the maximal squared radius considered.

Procedure Build2dLUT (Input N , Output LUT)

01. $n \leftarrow \lceil \sqrt{N} \rceil$; **For** i **From** 0 **To** N **Do** LUT[i] $\leftarrow \emptyset$
02. **For** x **From** 0 **To** n **Do**
02. **For** y **From** 0 **To** x **Do**
03. $i \leftarrow x^2 + y^2$

```

04.           If  $i \leq N$  Then
05.             LUT[i]  $\leftarrow$  LUT[i]  $\cup$  {(x, y)}

```

Procedure Build3dLUT (Input N , Output LUT)

```

01.  $n \leftarrow \lceil \sqrt{N} \rceil$ ; For  $i$  From 0 To  $N$  Do LUT[i]  $\leftarrow$   $\emptyset$ 
02. For  $x$  From 0 To  $n$  Do
03.   For  $y$  From 0 To  $x$  Do
04.     For  $z$  From 0 To  $y$  Do
05.        $i \leftarrow x^2 + y^2 + z^2$ 
06.       If  $i \leq N$  Then
07.         LUT[i]  $\leftarrow$  LUT[i]  $\cup$  {(x, y, z)}

```

Let $n \in \mathbb{N}, k \in \mathbb{N}^*$, we say that n is *k-decomposable* if there exists k integers $x_1 \dots x_k$ such that $n = x_1^2 + \dots + x_k^2$. Let n be a *k-decomposable* integer, let $x_1, \dots, x_k \in \mathbb{N}$, we say that (x_1, \dots, x_k) is a *primary k-decomposition of n* if $n = x_1^2 + \dots + x_k^2$, and if $x_i \geq 0$ for all $i = 1 \dots k$, and if $x_1 \geq \dots \geq x_k$. Let $n \in \mathbb{N}$, we denote by $r_k(n)$ the number of distinct primary *k-decompositions* of n , by $D_k(n)$ the number of decomposable positive integers not greater than n , by $R_k(n)$ the total number of distinct primary *k-decompositions* of positive integers not greater than n , and by $\tilde{r}_k(n)$ the mean number of distinct primary *k-decompositions* of positive integers not greater than n . In other words, $D_k(n) = \text{card}(\{i \in \mathbb{N}; i \leq n; r_k(i) > 0\})$; $R_k(n) = \sum_{0 < i \leq n} r_k(i)$; and $\tilde{r}_k(n) = R_k(n)/D_k(n)$. The number $R_k(n)$ gives the size of a lookup table for dimension k and for a maximum squared radius n , and $\tilde{r}_k(n)$ gives an ‘‘average’’ of the number of decompositions for a decomposable integer below n . In Fig. 17, we show the graphs of $R_2(n)$, $\tilde{r}_2(n)$, $R_3(n)$ and $\tilde{r}_3(n)$ for various values of n .

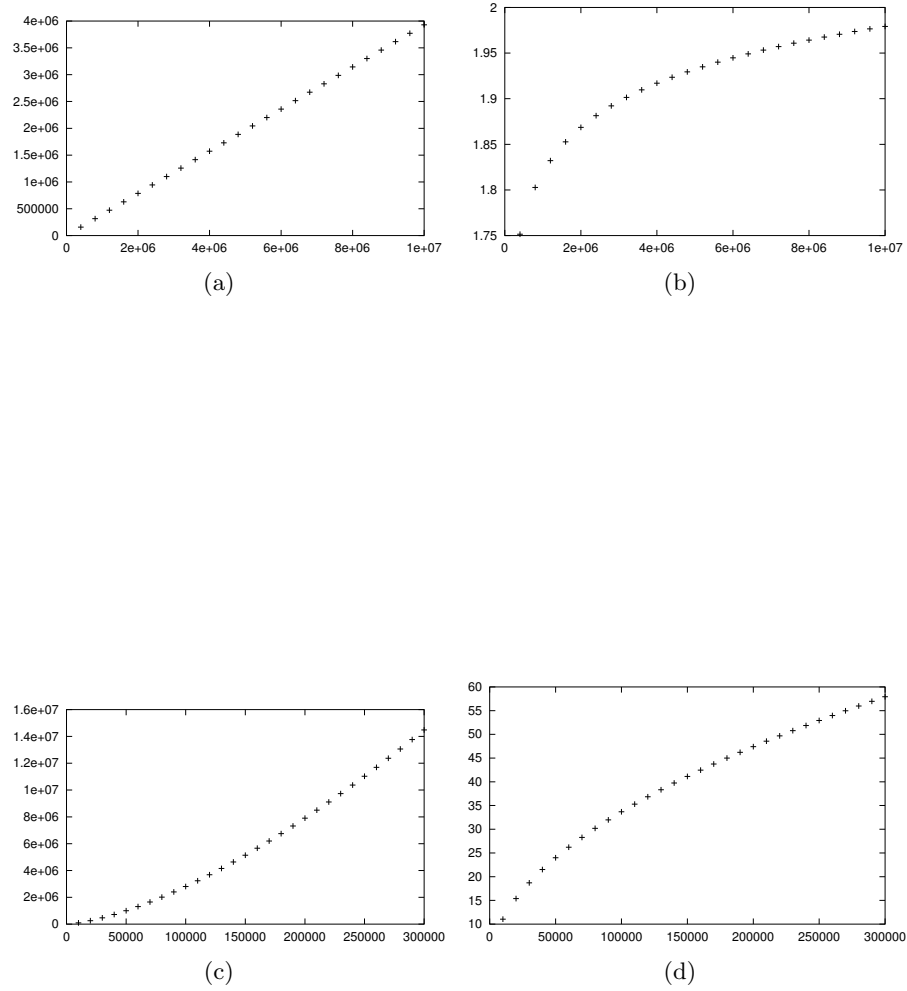


Fig. 17. (a): $R_2(n)$, with $0 < n \leq 10^7$. (b): $\tilde{r}_2(n)$, with $0 < n \leq 10^7$. (c): $R_3(n)$, with $0 < n \leq 3 \times 10^5$. (d): $\tilde{r}_3(n)$, with $0 < n \leq 3 \times 10^5$.

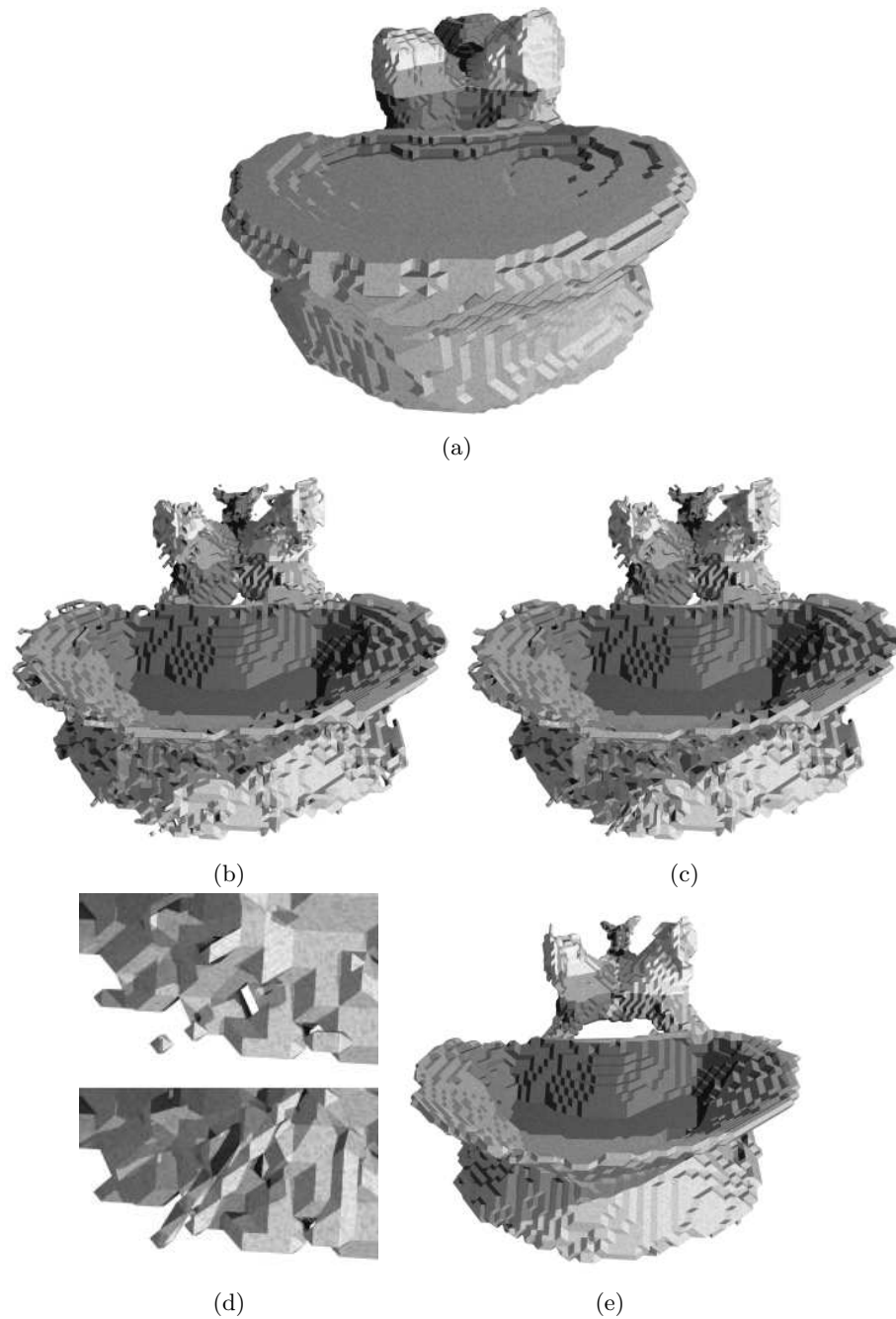


Fig. 14. (a): a view of the set X (vertebra). (b): a view of the medial axis $M(X)$ of the set X . (c): a view of **EuclideanSkeleton** $(X, D_X, M(X))$. (d): detail of (b) and (c). (e): a view of **FilteredSkeleton** $(X, 15, 1.5)$.

| | | | | |
|-----------|------------|--------------------|------------|--------------------|
| 0: (0, 0) | 8: (2, 2) | 17: (4, 1) | 29: (5, 2) | 40: (6, 2) |
| 1: (1, 0) | 9: (3, 0) | 18: (3, 3) | 32: (4, 4) | 41: (5, 4) |
| 2: (1, 1) | 10: (3, 1) | 20: (4, 2) | 34: (5, 3) | 45: (6, 3) |
| 4: (2, 0) | 13: (3, 2) | 25: (4, 3), (5, 0) | 36: (6, 0) | 49: (7, 0) |
| 5: (2, 1) | 16: (4, 0) | 26: (5, 1) | 37: (6, 1) | 50: (5, 5), (7, 1) |

Fig. 15. Look-up table (2D case): numbers 0 to 50

| | | | |
|--------------|--------------------------|--------------------------|--------------------------|
| 0: (0, 0, 0) | 9: (2, 2, 1), (3, 0, 0) | 18: (3, 3, 0), (4, 1, 1) | 27: (3, 3, 3), (5, 1, 1) |
| 1: (1, 0, 0) | 10: (3, 1, 0) | 19: (3, 3, 1) | 29: (4, 3, 2), (5, 2, 0) |
| 2: (1, 1, 0) | 11: (3, 1, 1) | 20: (4, 2, 0) | 30: (5, 2, 1) |
| 3: (1, 1, 1) | 12: (2, 2, 2) | 21: (4, 2, 1) | 32: (4, 4, 0) |
| 4: (2, 0, 0) | 13: (3, 2, 0) | 22: (3, 3, 2) | 33: (4, 4, 1), (5, 2, 2) |
| 5: (2, 1, 0) | 14: (3, 2, 1) | 24: (4, 2, 2) | 34: (4, 3, 3), (5, 3, 0) |
| 6: (2, 1, 1) | 16: (4, 0, 0) | 25: (4, 3, 0), (5, 0, 0) | 35: (5, 3, 1) |
| 8: (2, 2, 0) | 17: (3, 2, 2), (4, 1, 0) | 26: (4, 3, 1), (5, 1, 0) | 36: (4, 4, 2), (6, 0, 0) |

Fig. 16. Look-up table (3D case): numbers 0 to 36