Fast skeletonization of volumetric objects

Jorge Rodríguez*, Lluís Ros, Dolors Ayala, Federico Thomas
e-mail: jrodri@lsi.upc.es, llros@iri.upc.es,
dolorsa@lsi.upc.es, fthomas@iri.upc.es

Abstract

Many skeletonization algorithms for 3D solids have been proposed in recent years. The one given herein is surprisingly simple compared to most of them, but it is still very fast and has theoretically favorable properties. Actually, it provides a connected surface skeleton that allows shapes to be reconstructed with bounded error. Such skeleton is discretized, but can be computed at any desired resolution. In addition, the algorithm is also very attractive because it allows discrete skeletons to be obtained directly from solid objects in many representations without converting them to a voxel model. In other words, we present the first concise arbitrary-resolution skeletonization algorithm that does not require to necessarily work at voxel level. Our algorithm is a generalization of the one presented for 2D objects. It is based on the application of directional erosions, while retaining those regions whose elimination would introduce disconnections.
*keywords:*3D skeletons, solid modelling, volume rendering, object tracking, animation, medical imaging.

1 Introduction

The Medial Axis Transform (MAT) is well defined for any object as the closure of the set of centres of maximal balls which can be fit inside the object, where a ball is maximal if it is not contained in any other such ball [5]. Note that the definition of the MAT is general, and applies to objects of any dimension. The word *skeleton* is usually understood in 2D to mean the medial axis of a given shape. The *medial surface* of a 3D solid is defined similarly to its 2D counterpart.

The skeleton of an object is useful in computational geometry and geometric modeling. It has been used, for example, in modeling growth and in the analysis of symmetries, in path planning, feature recognition, and in finite element mesh generation

*This author has been supported by a grant from FONACIT (Venezuelan Council of Research in Science and Technology)

[18]. It has also been proposed as an alternative to boundary representation or constructive solid geometry models in a design and interrogation system, since it also provides a complete representation of a solid [23] and [8]. More recently, Skeletonization has also been used as an useful tool in medical imaging, providing an efficient method for visualization and analysis, such as interpretation of images [24], virtual navigation [12], [10], [4], surface extraction [21] and representation model [22], [15], [25].

Although there have been numerous studies on this technique in computer vision and pattern recognition [16], [5], [19], most research focuses on 2D images [6], incompatible with easy 3D extension.

The computation of the medial surface for arbitrary solids is a complex problem. So far, it has only been solved exactly for polyhedra [18]. Nevertheless, the MAT *per se* is not the unique skeleton which can be used. There are others, more easily derived skeletons, which share the critical properties of the MAT. Sudhalkar et al. [20] infer these properties as the followings:

- The skeleton should have no interior (thinness).
- The skeleton should have *homotopic equivalence* to the object. That is, there sould be holes and enclosed voids in the skeleton corresponding to those in the solid (connectivity).
- The location and relative proportions of features such as ribs, bosses, holes, etc. should reflect those in the solid.
- There should be thickness information associated with each point in the skeleton (reconstructability).

Unfortunately, as recognized by [17], in the discrete plane these requirements become mutually incompatible, so that practical skeleton methods are invariably a compromise between them. In this sense, thinness may be incompatible with preservation of connectedness, or with reconstructability.

Although other alternatives are possible [26], two main methods to obtain skeletons in discrete spaces have been proposed: boundary-based thinning methods (also called boundary peeling, erosion, etc.)and distance coding (also called distance transform).

Thinning methods iteratively peel off the boundary of a solid layer-by-layer, identifying "simple points", where removal does not affect the topology of the object. Up to our knowledge, the latest algorithm for thinning solids appeared in [13]. On the other hand, methods based on distance coding first convert the volume, which consist of foreground and background voxels, into a volume where every foreground voxel is labeled with a value corresponding to the minimum distance to the background. Different types of metrics for discrete solids are used, aiming to approximate the Euclidean distance. Then, the ridges of the induced scalar field constitute the skeleton. The ideal distance coding-based methods has three steps: 1) Approximate the minimum distance field, 2) detect all local maxima in terms of distance value, and 3) reconnect the local maxima to generate skeletons [26]. Up to our knowledge, the most recent distance coding-based skeletonization algorithm appeared in [14]. A more recent work based on a variant of distance coding methods, called the grass fire transform, can be found in [26].

Recently, a very simple thinning algorithm for 2D was proposed in [9] which falls inside the boundary-based thinning approach and it has been proved successfully over run-length encoded images. The algorithm we propose here is an extension of that former one.

2 Background

Let Z^3 be the discrete space. Let $X \subset Z^3$ be a 3D solid. Let $\bar{X} = Z^3 \setminus X$ denote the background of X. The connectivity used herein is (26,6)-connectivity, which means 26-connectivity for the solid and 6-connectivity for the background. Each of the 26 neighbors of a voxel in the solid defines a direction which will be numbered as shown in figure 1.

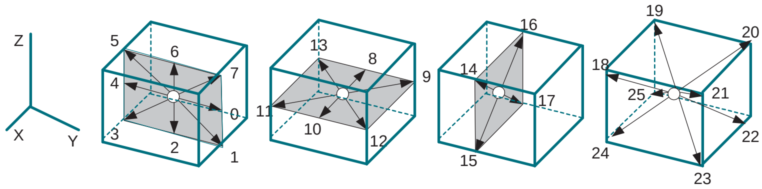

Figure 1: Directional offset in Z^3.

Mathematical morphological operations apply to sets of any dimensions, such as Euclidean R^n or its discrete equivalent, the set of n-tuples of integers Z^n. Dilation and erosion are the primary operations of mathematical morphology. Erosion of a binary region X using structuring element B is defined as $X \ominus B = \{y | \forall b \in B, y + b \in X\}$, the dilation using the same structuring element as $X \oplus B = \{y | y = x + b, x \in X, b \in B\}$ and its opening as $XoB = ((X \ominus B) \oplus B)$.

If X_b denotes the translation of X in the direction associated with $b \in B$, then it can be shown that $X \ominus B = \bigcap_{b \in B} X_{-b}$. That is, erosion can be accomplished by taking the intersection of all the translates of X, where the shifts in the translates are negated members of B seen as vectors.

An especially interesting case for B is that in which B consists of two voxels, where one is centered in the origin. Then, the erosion of X using B can be computed simply by $X \ominus B = X \cap X_{-b}$ and its opening by $XoB = (X \cap X_{-b}) \cup (X \cap X_{-b})_b$

Since $(B_1 \ominus B_2) \ominus B_3 = B_1 \ominus (B_2 \oplus B_3)$ then, if $B = B_1 \oplus B_2 \oplus ... \oplus B_k$, one concludes that $X \ominus B = (...[(X \ominus B_1) \ominus B_2] \ominus ... \ominus B_k)$

Thus, if a structuring element can be broken down to a chain of dilations of smaller substructuring elements, the desired operations may be performed as a string of suboperations.

3 Extreme Vertices Model

The natural redundancy in 3D images can be exploited to speed up morphological operation, often dramatically, through a spatial encoding scheme such as the Extreme Vertices Model (EVM) introduced in [2]. The EVM allows representing orthogonal polyhedra (OPP) - i.e., polyhedra with all their edges and faces oriented in three orthogonal directions - in a very concise form. This model also enables the development of simple and robust algorithms for performing the most usual and demanding tasks on solid modelling, such as closed and regularized boolean operations, solid splitting, other set-membership classification and measure operations on OPPs.

All the geometric and topological relations concerning faces, edges and vertices of an OPP can be obtained from its EVM. Actually, an EVM is a complete non-ambiguous implicit boundary model for OPPs, so an explicit boundary representation (B-rep) can be obtained from the EVM for visualization purposes.

3.1 Definitions and properties

Let P be an OPP, a *brink* is the maximal uninterrupted segment built out of a sequence of collinear and contiguous two-manifold edges of P. The ending vertices of a brink are called *extreme vertices* (EV). Figure 2(a) shows an OPP with a brink from vertex A to vertex E (both extreme vertices). In this brink, vertices B, C and D are non-extreme vertices.

The *EVM* is a representation scheme in which any OPP is described by its (an only its) set of EV. A *plane of vertices* (*plv*) is the set of vertices lying on a plane perpendicular to a main axis of P. A *slice* is the region between two consecutive planes of vertices, that is, the interior of P. A *section* (S) is the resulting polygon from the intersection between a slice and an orthogonal plane. Each slice has its representing section. Figure 2(b) shows an OPP with its planes of vertices and sections perpendicular to the X axis.

All these definitions can be extended to any dimension [7]. In this paper we are concerned with dimension ≤ 3. Planes of vertices and sections obtained from a 3D object are, then, 2D orthogonal polygons. From them, we can obtain their 1D lines of vertices and their 2D slices with their corresponding 1D sections. Finally, lines of vertices and 1D sections are 1D objects which are composed of one or several brinks.

Let $plv_i(P)$ be the *ith* plane of vertices and $S_i(P)$ be the *ith* section of EVM(P), where $i = 1 \ldots n$ with n the total number of planes of vertices. Sections can be

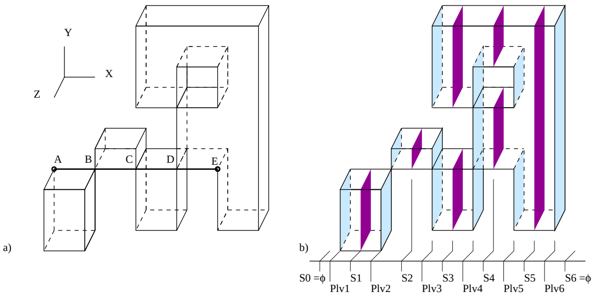

Figure 2: (a) An OPP with three non-manifold edges and a non-manifold vertex and a marked brink from vertex A to vertex E. (b) Planes of vertices and sections perpendicular to the X axis are shown in dark and light grey respectively.

computed from planes of vertices and vice-versa:

$$\begin{aligned} S_0(P) =& \quad S_n(P) = \emptyset \\ \overline{S_i(P)} =& \quad \overline{S_{i-1}(P)} \otimes^* \overline{plv_i(P)}, \qquad (1) \\ \overline{plv_i(P)} =& \quad \overline{S_{i-1}(P)} \otimes^* \overline{S_i(P)}, \qquad (2) \end{aligned}$$

where $\overline{plv_i(P)}$ and $\overline{S_i(P)}$ denote the projections of $plv_i(P)$ and $S_i(P)$ onto a main plane orthogonal to a coordinate axis and $\otimes^*$ denotes the regularized XOR operation. Note that in order to operate with the projections we need not take into account the coordinate of the extreme vertices that corresponds to the projecting plane.

General Boolean operations between OPP can be carried out by applying recursively the same Boolean operation over the corresponding OPP sections. This algorithm is presented in a previous work [3] and consists in a geometric merge between the EVM of both operands which involve as a basic operation the XOR between EV.

In addition, efficient morphological operations of 3D objects have been implemented through the EVM, shrinking or elongating the inner sections of the object. The algorithms can be found in [11].

4 3D skeletonization using EVM

The algorithm we propose here is an extension to 3D solids of a previous one for 2D binary images presented in [9]. We use the *Extreme Vertices Model (EVM)* as the spatial scheme of representation for the 3D solids. The EVM encoding avoids the natural redundancy in 3D voxelized objects speeding up morphological operations often dramatically and allows implementing the thinning algorithm indistinctly onto images and orthogonal solids with improved performances. In addition, thinning of both 3D voxelized objects and orthogonal solids is possible, and 3D skeletons of variable thickness can be produced using the EVM.

The proposed method satisfies the requirements given above in the following order of priority: connectedness, thinness and reconstructability . The proposed procedure can be outlined as follows: Those voxels whose deletion by a directional erosion might destroy the connectedness are retained and classified as *gaps*, then the region is eroded and the corresponding *residual* computed. Gaps and residuals are retained in the image, and this process is repeated until no progress is made. Following the reasoning depicted in [9] the skeleton is obtained by the union of *residuals* and *gaps*.

4.1 Residuals

$X \perp B$, is the set made of those points in X which do not belong to its opening using the structuring element B, that is, $X \perp B = X \setminus (XoB)$. Then, in terms of EVM, a disconnected skeleton of an object X can be defined as the set of all the residuals of the successive erosions of $EVM(X)$ (see Fig. 4(b)).

```
PROCEDURE algorithmS1 (INPUT $X : EVM$, $B : EVM$, OUTPUT $S : EVM$)
VAR $E : EVM$ ENDVAR
    $S \leftarrow \emptyset$;
    While $X \neq \emptyset$; do
        $E \leftarrow X \ominus B$;
        $S \leftarrow S \cup X \backslash (E \oplus B)$;
        $X \leftarrow E$;
    endWhile
end.
```

Now, let us assume that B is a centered $3 \times 3 \times 3$ cubic structuring element which can be broken down into a chain of six dilations of two-voxel elements in the directions 0,2,4,6,8 and 10. Then, the above algorithm can be rewritten as follows:

```
PROCEDURE algorithmS2 (INPUT $X : EVM$, $B : box$, OUTPUT $S$ : Array of EVM)
VAR $i : int, E : EVM$ ENDVAR
    $S \leftarrow \emptyset$; $i \leftarrow 1$;
    While $X \neq \emptyset$ do
        $E \leftarrow X \cap X_0 \cap X_2 \cap X_4 \cap X_6 \cap X_8 \cap X_{10}$;
        $S[i] \leftarrow X \backslash E \cup E_{10} \cup E_8 \cup E_6 \cup E_4 \cup E_2 \cup E_0$;
        $X \leftarrow E$; $i \leftarrow i + 1$;
    endWhile
end.
```

Then, each cell of S stores a part of the skeleton computed at the *ith* iteration, where each value of i will be associated with a grey-level in the final image which guaranties the desired property of reconstructability. The main advantage of this algorithm over the previous one is that it only involves directional erosions and dilations along the coordinate axes. Nevertheless, the skeleton thus obtained is neither unit-width nor connectivity preserving. A similar disconnected skeleton, but an unit-width one, can be easily obtained by slightly modifying the above algorithm as follows (see Fig. 4(c)):

```
PROCEDURE algorithmS3 (INPUT $X : EVM$, $B : EVM$, OUTPUT $S$ : Array of EVM)
VAR $E, A : EVM; d, i : int$ ENDVAR
    $S \leftarrow \emptyset$; $A \leftarrow \emptyset$; $i \leftarrow 1$;
    While $X \neq \emptyset$ do
        for $(d \leftarrow 0; d \leq 10; d \leftarrow d + 2)$
            $E \leftarrow X \cap X_d$;
            if $(0 \leq d \leq 6)$ then
                $A \leftarrow A \cup (X \setminus (E \cup E_{(d+4) mod 8}))$;
            else
                if $(d = 8)$ then $A \leftarrow A \cup (X \setminus (E \cup E_{10}))$;
                else $A \leftarrow A \cup (X \setminus (E \cup E_8))$;
                endIf
            endIf
            $X \leftarrow E$;
        endFor
        $S[i] \leftarrow A$; $i \leftarrow i + 1$;
    endWhile
end.
```

Unfortunately, as it has been already said, thinness and reconstructability are mutually incompatible goals in the discrete sets, so the original shape can only be nearly reconstructed. Also, the connectivity among the residuals must be solved.

4.2 Directional gaps

Those voxels to ensure connectivity in the final skeleton will be part of a set of disjoined regions that we will call gaps. Contrary to what one might expect, when considering only directional erosions, gaps can be easily computed. For example, the directional gap of a binary region X in direction 0 (the coordinate y axis in the positive region direction) can be obtained by computing:

$$\begin{aligned} X \setminus X_0 \cap ((X_7 \setminus X_6) \cup (X_1 \setminus X_2) \cup (X_{20} \setminus X_{16}) \cup (X_{22} \setminus X_{17}) \\ \cup (X_{21} \setminus X_{14}) \cup (X_{23} \setminus X_{15}) \cup (X_{12} \setminus X_{10}) \cup (X_9 \setminus X_8)) \end{aligned} \tag{3}$$

This boolean formula detects the voxel configurations shown in figure 3 and all $\pi/2$ rotated versions around the y axis. While grey cubes correspond to points in the foreground, the transparent cubes correspond to points in the background.

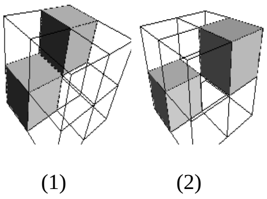

Figure 3: Gaps in Z^3.

Gaps along the other coordinates axes, either in positive or negative directions, can analogously be obtained. The above expression is obtained as a generalization of the two-dimensional case. It is worth nothing that the concept of gaps, first introduced in [9], is closely related to the set of βn templates recently presented in [1].

The EVM encoding allows us to represent and manipulate images and 3D voxelized objects indistinctly. Then, skeletonization of 3D objects using EVM is obtained as a simple generalization of the two-dimensional case.

4.3 The thinning algorithm

The implementation of the thinning algorithm following this approach consist in two steps repeated iteratively until no more progress is made. The first step is to retain those voxels whose deletion by a directional erosion might destroy the connectedness, which are classified as gaps. Next, the region is effectively eroded and the corresponding residual is computed. The following algorithm implements this process and work for both images and 3D voxelized objects using the EVM as spatial encoding.

```
**PROCEDURE** algorithmS4 (INPUT $X : EVM$,$B : EVM$ ,OUTPUT $S : EVM$)
**VAR** $A, G, E : EVM; d, i, loop : int$ **ENDVAR**
    $S \leftarrow \emptyset$; $A \leftarrow \emptyset$; $S \leftarrow X$;
    **if** (dim=2) **then** loop=6 **else** loop=10
    **do**
        $X \leftarrow S$;
        **for** $(d \leftarrow 0; d \leq loop; d \leftarrow d + 2)$
            $G \leftarrow Gaps(S, dim, d)$; $E \leftarrow S \cap S_d$;
            **if** $d \neq 8$ or $d \neq 10$ **then**
                $R \leftarrow S \setminus (E \cup E_{(d+4)mod8})$;
            **else**
                **if** $(d = 8)$ **then** $R \leftarrow S \setminus (E \cup E_{10})$;
```

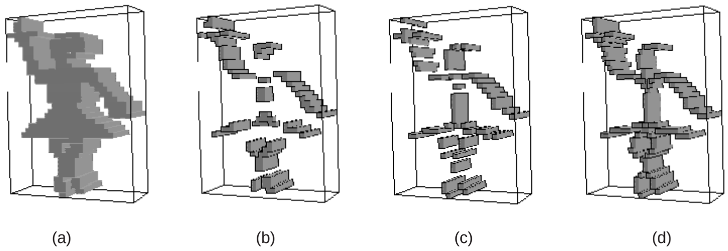

Figure 4: (a) original object. (b) residuals by S1. (c) Unit-width residuals by S2. (d) 3D skeleton by S4.

```
                else $R \leftarrow S \setminus (E \cup E_8)$;
                endIf
            endIf
            $A \leftarrow A \cup R \cup G$; $S \leftarrow A \cup E$;
        endFor
    while $(X \neq S)$
end.
```

where $Gaps(S, dim, d)$ implements the expression 3 for every direction of translate (indicated for the parameter d). Figure 4 shows the original doll (a), the residuals obtained by algorithmS1 (b), the unit-width residuals obtained by algorithmS2 (c) and the final skeleton (d) obtained by applying the general thinning algorithm (algorithmS4).

5 Acceleration technique based on decomposed EVM operations

The process to obtain the skeleton depicted above consist in successive boolean operations between the input object, X, and displaced versions of itself, X_d, $x = 0, 1..., 25$. It is easy to prove that $EVM(X_d) = (EVM(X))_d$, because it is a simple ordered set of points, so it is translation invariant. Then, all the planes of vertices of $EVM(X_d)$ can be computed from $EVM(X)$ displaced conveniently. Let $plv_i(X)$ be the ith plane of vertices of the $EVM(X)$, then

$$plv_i(X_d) = (plv_i(X))_d \tag{4}$$

$$S_i(X_d) = plv_i(X_d) \otimes plv_{i+1}(X_d) = (plv_i(X))_d \otimes (plv_{i+1}(X))_d = (S_i(X))_d \tag{5}$$

By the other side, X_d is always displaced one unit respect of X in the corresponding direction d (see Fig. 5). The same is true for every pair of operands involved in the skeletonization because the definition of gaps and residuals (see definitions above), so the spatial coherence can be profited. Let X and X_d be a pair of operands in the skeletonization process and let **op** be the operation between them, one of the following is always true:

$$\begin{aligned} S_i(X \text{ op } X_d) &= S_i(X) \text{ op } S_i(X_d) && \text{if } d \in \{2,6,14,15,16,17\} \\ S_i(X \text{ op } X_d) &= S_i(X) \text{ op } S_{i+1}(X_d) && \text{if } d \in \{3,4,5,7,13,18,19,24,25\} \\ S_i(X \text{ op } X_d) &= S_i(X) \text{ op } S_{i-1}(X_d) && \text{if } d \in \{0,1,7,9,12,20,21,22,23\} \end{aligned} \tag{6}$$

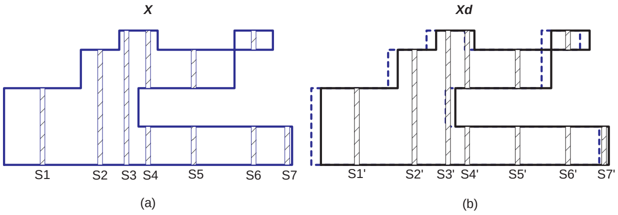

Figure 5: (a) EVM of X. (b) EVM of X_0 overlapped on X.

In the conventional implementation of skeletonization, all the displacements of X involved in the expression have to be computed previously and afterward, each resulting EVM is operated with each other until the total expression is completed. For example, computing expression 3 would require to build 22 3D-EVMs before to solve the final expression.

Profiting the invariance on translations and the spatial coherence between operands, an incremental version of boolean operations is proposed in order to avoid computing the EVM of all displacements of X. In this approach, an expression composed of several operations is decomposed to solve it section by section, i.e. each section is operated against all displaced versions involved in the expression, so all the intermediate operands are solved through 2D-EVM and only the 3D-EVM of the final result is builded. This strategy saves multiple computing of sections and EVM traversals which are very time consuming. Using the equations 4 and the fact of every operation involves two operands displaced in at most one unit of each other (equations 6), the $EVM(R)$ resulting of operating X and X_0, for example, can be expressed as:

$$EVM(R) = \{S_i(R) \backslash S_i(R) = S_i(X) \text{ op } S_{i-1}(X)_d, i = 0 \dots n\} \tag{7}$$

6 Results

This section presents some results of our proposal. The testing datasets are a synthetic model of a truck, and an human vertebra obtained by CT (see Fig. 6). All the algorithms have been run on a Sun Ultra Sparc 60 workstation. Images show, per each object, the original model in grey, its yellow skeleton overlapped onto the transparent original and a zoom-in of the previous composition. Table shows, per each dataset, their number of voxels and the processing time required to obtain the skeleton. The first one corresponds to the Thomas et al.'s implementation (unencoded-based), the second one corresponds to our proposal (EVM-based) and the last row shows the times with the acceleration technique (improved-EVM).

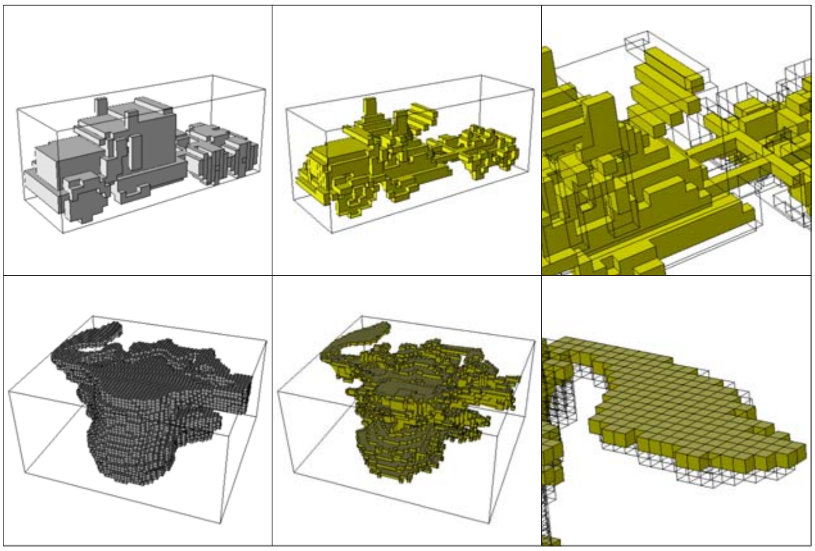

Figure 6: original, skeleton and detail for truck (up) and vertebra (bottom)

	truck	vertebra
# of voxels	5910	22263
unencoded-based	758	2644
EVM-based	266	920
EVM-improved	118	377

Table 1: Skeletonization processing time (in seconds)

7 Acknowledgments

This work has been partially supported by a CICYT grant TIC99-1230-C02-02 and MAT2002-04297-C03-02.

References

[1] F. Prêteux A. Manzanera, T.M. Bernard and B. Longuet. Medial faces from a concise 3D thinning algorithm. *7th IEEE Conf. on Computer Vision*, 1, 1999.

[2] A. Aguilera. *Orthogonal Polyhedra: Study and Application*. PhD thesis, LSI-Universitat Politècnica de Catalunya, 1998.

[3] A. Aguilera and D. Ayala. Orthogonal Polyhedra as Geometric Bounds in Constructive Solid Geometry. In C. Hoffmann and W. Bronsvort, editors, *ACM SM'97. Atlanta (USA)*, pages 56 – 67, 1997.

[4] S.R. Aylward. Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction. *IEEE Transactions on Medical Imaging*, 21(2):61–75, 2002.

[5] H. Blum. A transformation for extracting new descriptors of shape. *Models for the perception of speech and visual form*, pages 362–380, 1967.

[6] G. Borgefors, G. Ramell, and G. di Baja. Hierarchical decomposition of multiscale skeletons. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1296–1312, 2001.

[7] O. Bournez, O. Maler, and A. Pnueli. Orthogonal Polyhedra: Representation and Computation. In *Hybrid Systems: Computation and Control*, LNCS 1569, pages 46 – 60. Springer, 1999.

[8] J.W. Brandt. Describing a solid with the three-dimensional skeleton. *SPIE Curves and Surfaces in Computer Vision and Graphics III*, 1830:258 – 269, 1992.

[9] R. Cardoner and F. Thomas. Residual + directional gaps = skeletons. *Pattern recognition letters*, 18:343 – 353, 1997.

[10] N. Gagvani. Skeletons and volume thinning in visualization. Master's thesis, Rutgers University, New Brunswick, N.J., June 1997.

[11] J.Rodriguez and D.Ayala. Erosion and dilation of 2D and 3D images using the EVM. In *CEIG 2001*, pages 203–214, 2001.

[12] Y.C. Wei A. Viswambharan M. Wax L. Hong, A. Kaufman and Z. Liang. 3D virtual colonoscopy. *IEEE Computer Graphics and Applications*, 21(4):26–32, 1995.

[13] Ch. Lohou and G. Bertrand. A new 3D 6-subiteration thinning algorithm based on p-simple points. In *Discrete Geometry for Computer Imagery, 10th international conference*, pages 102 – 113, 2002.

[14] G. Malandain and S. Fernandez-Vidal. Euclidean skeletons. *Image and Vision Computing*, 16:317 – 327, 1998.

[15] A. Puig, D. Tost, and I. Navazo. An interactive cerebral blood vessel exploration system. *IEEE Visualization'97*, 1997.

[16] A. Rosenfeld and J.L. Pfaltz. Sequential operations in digital picture processing. *Journal of the ACM*, 13(4):471 – 494, 1966.

[17] J. Serra. *Image Analysis and Mathematical Morphology.* Academic Press, London, 1992.

[18] E. Sherbrooke, N. Patrikalakis, and E. Brisson. An algorithm for the medial axis transform of 3D objects. *IEEE Transactionsn on Visualization and Computer Graphics*, 2(1):44–61, 1996.

[19] R.W. Smith. Computer processing of line images: A survey. *Pattern Recognition*, 20(1):7–15, 1987.

[20] A. Sudhalkar, L. Gürxöz, and F. Prinz. Box-skeletons of discrete solids. *Computer-aided Design*, 28(6/7):507 – 517, 1996.

[21] Y. Yamaguchi T. Itoh and K. Koyamada. Volume thinning for automatic isosurface propagation. In *Proc. IEEE Visualization '96*, pages 303–310, 1996.

[22] P.J. Vermeer. *Medial axis tranform to boundary representation conversion.* PhD thesis, Purdue University, May 1994.

[23] F.-E. Wolter. *Cut Loci in borderd and unbordered Riemannian manifolds.* PhD thesis, Purdue university, May 1994.

[24] P.J. Yim, P.L. Choyke, and R.M. Summers. Gray-scale skeletonization of small vessels inn magnetic resonance angiography. *IEEE Transaction on Medical Imaging*, 19:568–576, 2001.

[25] Y. Zhou, A. Kaufman, and A.W. Toga. 3D skeleton and centerline generation based on an approximate minimum distance field. *The Visual Computer*, 14(7):303–314, 1998.

[26] Y. Zhou and A.W. Toga. Efiicient skeleton of volumetric objects. *IEEE Transaction on Visualization and Computer Graphics*, 5(3):196–209, 1999.