Skeletonizing Volume Objects
Part II: From Surface to Curve Skeleton

Gunilla Borgefors[1], Ingela Nyström[2] and Gabriella Sanniti di Baja[3]

[1] Centre for Image Analysis, Swedish University of Agricultural Sciences
Uppsala, Sweden
gunilla@cb.uu.se

[2] Centre for Image Analysis, Uppsala University
Uppsala, Sweden
ingela@cb.uu.se

[3] Istituto di Cibernetica, National Research Council of Italy
Arco Felice (Napoli), Italy
gsdb@imagm.cib.na.cnr.it

Abstract. Volume imaging techniques are becoming common and skeletonization has begun to prove valuable for shape analysis also in 3D. In this paper, a method to reduce solid volume objects to their 3D curve skeletons is presented. The method consists of two major steps. The first step is aimed at the computation of the surface skeleton, and is an improvement of a previous method. In the second step, the surface skeleton is further reduced to the 3D curve skeleton. Our skeletonization method preserves topology; no disconnections, holes or tunnels are created. It also preserves the general geometry of the object, especially in the case of elongated objects. Resulting skeletons for a number of synthetic and real images are presented.

Keywords: volume (3D voxel) images, surface skeleton, curve skeleton

1 Introduction

The 3D skeleton of a volume object is either a set of 3D surfaces and curves, or, in the case of solid objects, a set of only 3D curves. In any case, the 3D skeleton is a unit wide subset of the object centered within the object and characterized by the same topological properties as the object. The 3D skeleton is a promising tool for an increasing number of applications, especially in biomedical imagery. However, the articles published on 3D skeletonization are still not very numerous, compared to the literature on 2D skeletonization.

The general strategy for 3D skeletonization does not differ significantly from the strategy in the 2D case. Object voxels are changed to background voxels under the constraint that topology and geometry of the object are preserved. However, a number of essential problems are more difficult to solve efficiently in the 3D case. E. g., when designing topology preserving removal operations, besides preventing disconnections and creation of cavities, which must be done also in 2D, you must avoid the creation of tunnels and the "excavation" of unwanted deep cavities in complex surfaces. Although many concepts such as Euler characteristics, simple points and connectivity have been studied in the past years [1, 6, 7, 8, 15], the implementation of skeletonization methods based on their use is rather complicated.

Different approaches to 3D curve skeletonization have been proposed [2, 8, 9, 10, 17, 18]. Mostly, the only examples given are tiny test images, which makes it difficult to understand what the results would be for reasonably sized and/or real images. It seems that the definition of protrusions or end-points, are what separates existing algorithms; every author has his own criteria, resulting in very different skeletons.

This paper is a follow-up to a previous one, where a new method to compute the surface skeleton of volume objects was proposed [4]. We first improve the method for the computation of the surface skeleton (Sect. 3) and then introduce a method to reduce the surface skeleton to a curve skeleton (Sect. 4). We present resulting skeletons for a number of synthetic and real images (Sect. 5).

2 Definitions

Each voxel v has three types of neighbours among its 26 closest neighbours; 6 face-, 12 edge-, and 8 point-neighbours, that share a face, an edge, and a point with v, respectively.

In a binary image we define an object component as a 26-connected set of voxels. As a consequence of the 26-connectedness selected for the object, 6-connectedness must be used for the background. If the background consists of exactly one 6-connected component, then the object is termed a *solid* object.

A *border* voxel is an object voxel with a face-neighbour in the background. Object voxels, which are not border voxels, are *internal* voxels.

Voxels are characterized by the numbers of n-connected components of object and background in their neighbourhood. E. g., a voxel is a *break-point* voxel if it has more than one 26-connected object component in its 26-neighbourhood. The recursive algorithm in [5] can be used to count connected components efficiently.

N_f18 is defined as the number of 6-connected background components in the 18-neighbourhood of a voxel having the central voxel as a face-neighbour. On a 3D surface, an *outer* voxel has $N_f18 = 1$. If $N_f18 > 1$ it is called a *tunnel* voxel, because removing it would create a tunnel through the object. This classification of surface voxels is discussed in [11, 16].

The D^6 metric is obtained by counting the number of steps in the minimal 6-connected path between voxels. The D^6 distance is the 3D equivalent of the city-block distance in 2D [3].

3 Surface Skeleton

The surface skeleton is computed in two phases. First, an at most two voxels thick set of skeletal voxels is found, based on the notion of multiple voxels. In the second phase, the set is reduced to a unit-wide surface skeleton. Here, we sketch the general method and describe in detail only the first phase, which has been modified and improved with respect to [4].

Voxels are iteratively removed from the object, until no more voxels can be removed. Each iteration consists of four parallel steps (or scans through the image).

Step 1: .

1. Among voxels not already labelled, identify border voxels, and label them with the current iteration number.
2. Among internal voxels, identify voxels with an edge-neighbour in the background and label them with the current iteration number plus one.

Step 2: Among voxels labelled with the current iteration number, mark those that are multiple.

Step 3: Among non-multiple voxels labelled with the current iteration number, mark as tunnel voxels those for which $N_f18 > 1$. When computing N_f18 neighbouring non-multiple voxels labelled with the current iteration number are interpreted as background voxels, i. e. they are treated as if they were already removed.

Step 4: Remove all unmarked border voxels.

Labelling the voxels with an edge-neighbour in the background with the iteration number plus one, gives them the correct distance label (from the original background). Consider the internal voxels placed at the intersection of the two crossing planes in Fig. 1, and suppose that we do not label them as requested by Step 1.2. Removal of the border voxels in the intersection, exposes to the background other voxels in the intersection. These voxels would become removable in the next iteration. By iteration, the surface skeleton would finally have four planes linked to each other only by a single middle voxel. This would still be topologically correct, but not desirable as it would complicate analysis of the surface skeleton. If we instead consider that all voxels in the intersection have the *same* distance label in the D^6 distance transform of the object, it becomes apparent that they should all be checked for removal in the *same* iteration. Hence, the labelling of voxels introduced in Step 1.2.

In Step 2 we identify multiple voxels. For volume images, we define a border voxel of the current iteration, v, as multiple if any of Conditions A1–A3 is satisfied:

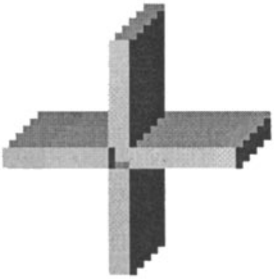

Fig. 1. Where two surfaces cross the outermost voxels of the crossing are removed, but creation of deep cavities is prevented by Step 1.2 in the surface skeletonization method.

Condition A1: No pair of opposite face-neighbours (aligned along one of the three principal planes) of v exists, such that one of them is an internal voxel and the other a background voxel.

Condition A2: A 2×2 neighbourhood of v (in any of the three principal planes) exists, such that the edge-neighbour of v is a border voxel (of the current or an earlier iteration), and the two face-neighbours are background voxels.

Condition A3: A $2 \times 2 \times 2$ neighbourhood of v exists, such that the point-neighbour of v is a border voxel (of the current or an earlier iteration), while the other six neighbours are background voxels.

Step 3 is necessary to prevent tunnel creation in thin complex objects [11, 16]. This problem is even more pronounced when reducing the surfaces to curves, and will be further described in Sect. 4.

The skeletal set obtained is 26-connected and at most two voxels thick. The voxels are labelled with the iteration number, which coincides with the D^6 distance transform of the object. As the skeletal set includes all the local maxima of the D^6 distance transform, object recovery is possible using the *reverse* D^6 distance transformation [14].

A few internal voxels might still remain where many surfaces and/or curves meet. Some of these voxels might not be labelled, i. e. they do not have an edge-neighbour in the background. Such voxels should be assigned a label equal to the minimum label in their 6-neighbourhood plus one, which corresponds to the label they would have in the D^6 distance transform.

The set can be reduced to unit thickness by applying *six* directional thinning processes sequentially. Using directional processes is necessary to prevent breaking connectedness and excessive shortening. After the thinning processes the skeleton consists of unit-wide surfaces (and curves).

4 Curve Skeleton

The surface skeleton can be further reduced to a curve skeleton. A thorough investigation of the strategies leading to the computation of the curve skeleton can be found in [12]. Voxels are iteratively removed from the unit-wide surface skeleton, until no more voxels can be removed. Each iteration consists of three steps (or scans through the image). Only the two first steps are performed in parallel.

Step 1: Identify outer voxels, $N_f18 = 1$, on the surface.

Step 2: Inspect all outer voxels (both from the current and earlier iterations). Mark a voxel as removable, if it has two 26-connected components of outer voxels, but only one 26-connected component of object in its 26-neighbourhood.

Step 3: Sequentially remove marked voxels, unless they are break-point voxels.

Step 1 identifies the voxels which are candidates for removal. Intuitively, interior surface voxels have more than one background component in their neighbourhood, and voxels on the border of a surface have one background component. As the background is 6-connected, its components are counted using 6-connectedness. In Fig. 2 the outer voxels are marked in grey. If counting background components in the 26-neighbourhood, the hatched voxels would also be identified as outer voxels. Removal of a hatched voxel would change the topology of the object as a tunnel would be created, therefore they must not be identified as outer voxels. The solution is to count only the 6-connected background components in the 18-neighbourhood having the central voxel as a face-neighbour [11, 16], N_f18.

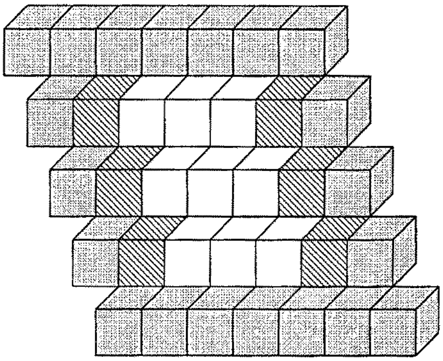

Fig. 2. A thin surface where the outer voxels are marked in grey. The hatched voxels are erroneously identified as outer voxels, if counting the 6-connected background components in the 26-neighbourhood; the 18-neighbourhood should be used.

Every outer voxel is checked for removal in Step 2. The voxel must have only one 26-connected object component, otherwise the object will be disconnected. Also, it must have two components of outer voxels, otherwise it may be the tip of a protrusion. With this definition of protrusion some voxels that are not really "protrusions" may remain, giving the curve skeleton for complex surfaces a "cloudy" look. This happens when the outer voxels do not constitute a connected set; some voxels only have one component of outer voxels, even if it is not a protrusion. Outer voxels identified in earlier iterations are also checked as the removal of neighbouring voxels may create two components of outer voxels, which allows removal.

The removal of any single voxel marked in Step 2 preserves connectedness, but the simultaneous removal of all of them may disconnect the object. Therefore, Step 3 must be sequential.

If the original surface is (locally) an even number of voxels broad, the curve skeleton becomes two voxels wide. It can be reduced to unit thickness by an iterative thinning process. Each iteration consists of three steps (or scans through the image). Only the two first steps are performed in parallel.

Step 1: Classify *tip-of-protrusion* voxels as those for which a "cap" (Fig. 3) of background voxels fits in any of the six directions. Classify *break-point* voxels, which have more than one component of object in their 26-neighbourhood.

Step 2: Among non-classified voxels, mark outer voxels, $N_f18 = 1$, whose removal will not create tunnels.

Step 3: Sequentially remove the outer voxels, unless the object is disconnected.

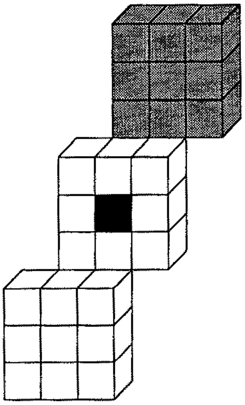

Fig. 3. A 26-neighbourhood shown slice by slice with a "cap". The central voxel is a tip-of-protrusion voxel. Background voxels are white. "Don't care" voxels are grey. Rotation of the "cap" gives the other five direction masks.

The result is a 26-connected unit-wide curve skeleton. The original object can thereafter not be recovered, but its topology and geometry are preserved. The skeletal voxels are labelled with their (D^6) distance to the original background, which might be useful information in quantitative shape analysis. For elongated objects, e. g. blood vessels, curve skeletonization preserves the essential information, especially since the curve voxels are marked with the current diameter. See [13] for a recent publication.

5 Examples and Discussion

Volume objects are synthesised in 128 × 128 × 128 images. In Fig. 4 a box of size 40 × 60 × 80 voxels is shown. The surface skeleton of the box can be seen in the middle. The original object can be recovered, except for some of the original border voxels. The curve skeleton of the box can be seen to the right, describing the shape of the original object well. The skeletonizations of a pyramid (base 85 × 85 and height 122 voxels) and of the same pyramid rotated 45° can be seen in Fig. 5. Remember that the underlying metric is the rotation dependent D^6 metric.

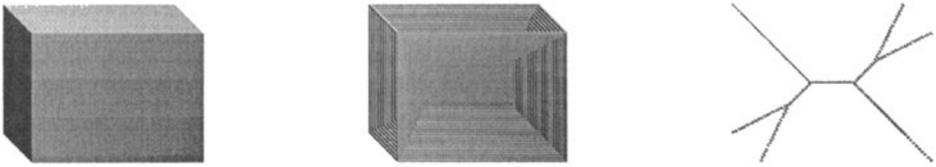

Fig. 4. A box (left), its surface skeleton of labelled voxels (middle), and its curve skeleton (right). [Thanks to Dr. Pieter Jonker, Delft University, The Netherlands, for letting us use the volume visualization software.]

A non-synthetic example is a hand in a 170 × 150 × 120 image (Fig. 6). The original hand includes more than 400 000 voxels. By representing the hand with its surface skeleton, 6.1 % of the original voxels are needed. By reducing the rather complex surfaces, a complex curve skeleton is obtained. In Fig. 6 the curve skeleton has been further pruned by a simple connectivity-preserving thresholding of the distance labels. Only 861 of the original voxels are needed to describe the topology and, to some extent, the shape of the hand.

The skeletonization takes on average a few minutes for objects in images of a few Mbyte size on ordinary (DEC Alpha) workstations.

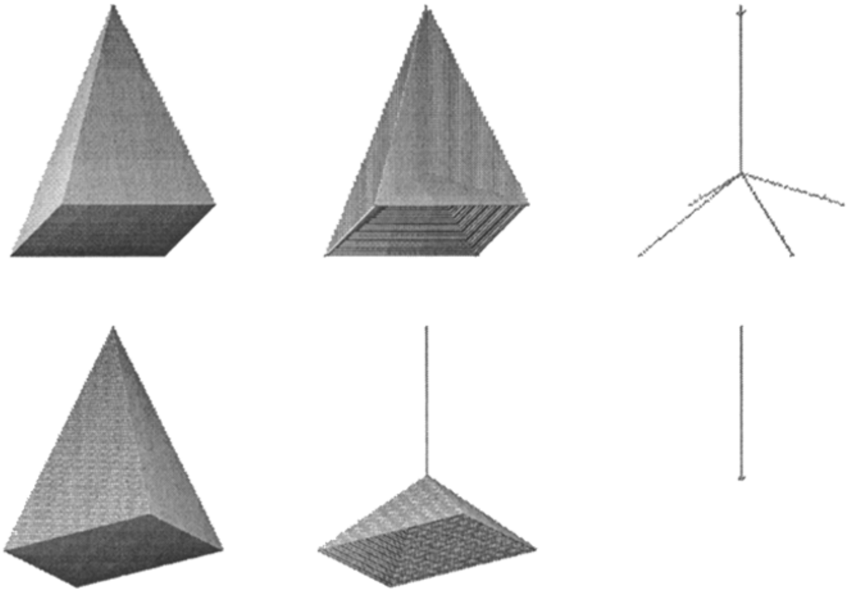

Fig. 5. Top: A pyramid (left), its surface skeleton (middle), and its curve skeleton (right). Bottom: The pyramid rotated 45° (left), its surface skeleton (middle), and its curve skeleton (right).

As the underlying metric used in the skeletonization is D^6, the skeletons produced suffer from rotation dependence. Also, objects with flat surfaces will produce much "nicer" skeletons than objects with curved surfaces. To solve these problems we need to develop skeletons based on more rotation independent metrics. The skeletons are also sensitive to noise. Every protrusion from a flat surface or a straight edge (in any rotation), that is not locally a corner of an octahedron oriented with the coordinate axes through its vertices, will generate a skeletal branch. Pre-processing in the form of morphological smoothing operations will alleviate this problem. All essential parts of the skeletons are there, so the task is to remove branches resulting from weak protrusions and noise. Developing good pruning strategies for 3D surface and curve skeletons is an important problem for further research.

Acknowledgements

Scientific support were given by Prof. Ewert Bengtsson and Dr. Bo Nordin, which is gratefully acknowledged.

References

1. G. Bertrand. Simple points, topological numbers and geodesic neighbourhoods in cubic grids. *Pattern Recognition Letters*, 15:1003–1011, Oct. 1994.

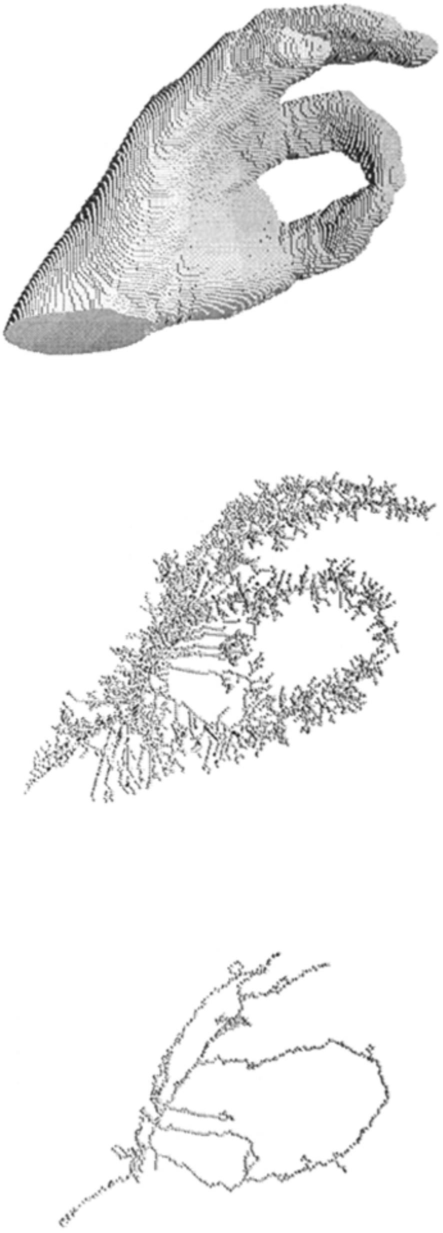

Fig. 6. A hand (top), its curve skeleton (middle), and the pruned curve skeleton (bottom). [The copyright holders of this hand are Prof. Jun-ichiro Toriwaki and Prof. Kazuhiro Katada, Nagoya University, Japan.]

2. G. Bertrand and Z. Aktouf. A three-dimensional thinning algorithm using subfields. In R. A. Melter and A. Y. Wu, editors, *Vision Geometry III*, pages 113–124. Proc. SPIE 2356, 1994.
3. G. Borgefors. On digital distance transforms in three dimensions. *Computer Vision and Image Understanding*, 64(3):368–376, Nov. 1996.
4. G. Borgefors, I. Nyström, and G. Sanniti di Baja. Surface skeletonization of volume objects. In P. Perner, P. Wang, and A. Rosenfeld, editors, *Proceedings of SSPR'96: Advances in Structural and Syntactical Pattern Recognition*, pages 251–259. Springer-Verlag, Berlin Heidelberg, 1996.
5. G. Borgefors, I. Nyström, and G. Sanniti di Baja. Connected components in 3D neighbourhoods. In *Proceedings of* 10th *Scandinavian Conference on Image Analysis, Lappeenranta, Finland*, pages 567–572, 1997.
6. T. Y. Kong. On topology preservation in 2-D and 3-D thinning. *International Journal of Pattern Recognition and Artificial Intelligence*, 9(5):813–844, 1995.
7. T. Y. Kong and A. Rosenfeld. Digital topology: Introduction and survey. *Computer Vision, Graphics, and Image Processing*, 48:357–393, 1989.
8. T.-C. Lee, R. L. Kashyap, and C.-N. Chu. Building skeleton models via 3-D medial surface/axis thinning algorithms. *CVGIP: Graphical Models and Image Processing*, 56(6):462–478, 1994.
9. S. Lobregt, P. W. Verbeek, and F. C. A. Groen. Three-dimensional skeletonization: Principle and algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2(1):75–77, Jan. 1980.
10. C. M. Ma and M. Sonka. A fully parallel 3D thinning algorithm and its applications. *Computer Vision and Image Understanding*, 64(3):420–433, Nov. 1996.
11. G. Malandain, G. Bertrand, and N. Ayache. Topological segmentation of discrete surfaces. *International Journal of Computer Vision*, 10(2):183–197, 1993.
12. I. Nyström. *On Quantitative Shape Analysis of Digital Volume Images*. PhD dissertation No. 288, Uppsala University, Uppsala, Sweden, May 1997. Available from the author.
13. I. Nyström. Skeletonization applied to magnetic resonance angiography images. In K. M. Hanson, editor, *Medical Imaging 1998: Image Processing*. Proc. SPIE 3338, Feb. 1998. Accepted for publication.
14. I. Nyström and G. Borgefors. Synthesising objects and scenes using the reverse distance transformation in 2D and 3D. In C. Braccini, L. DeFloriani, and G. Vernazza, editors, *Proceedings of ICIAP'95: Image Analysis and Processing*, pages 441–446. Springer-Verlag, Berlin Heidelberg, 1995.
15. P. K. Saha and B. B. Chaudhuri. Detection of 3-D simple points for topology preserving transformations with application to thinning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(10):1028–1032, Oct. 1994.
16. P. K. Saha and B. B. Chaudhuri. 3D digital topology under binary transformation with applications. *Computer Vision and Image Understanding*, 63(3):418–429, May 1996.
17. P. K. Saha, B. B. Chaudhuri, and D. D. Majumder. A new shape preserving parallel thinning algorithm for 3D digital images. *Pattern Recognition*, 30(12):1939–1955, Dec. 1997.
18. T. Saito and J.-I. Toriwaki. A sequential thinning algorithm for three dimensional digital pictures using the Euclidean distance transformation. In *Proceedings of* 9th *Scandinavian Conference on Image Analysis, Uppsala, Sweden*, pages 507–516, 1995.