

# Surface Skeletonization of Volume Objects

Gunilla Borgefors<sup>1</sup>, Ingela Nyström<sup>2</sup> and Gabriella Sanniti di Baja<sup>3</sup>

<sup>1</sup> Centre for Image Analysis, Swedish University of Agricultural Sciences,  
Lägerhyddv. 17, S-75237 Uppsala, Sweden,  
gunilla@cb.uu.se

<sup>2</sup> Centre for Image Analysis, Uppsala University,  
Lägerhyddv. 17, S-75237 Uppsala, Sweden,  
ingela@cb.uu.se

<sup>3</sup> Istituto di Cibernetica, National Research Council of Italy,  
Via Toiano 6, I-80072 Arco Felice (Napoli), Italy,  
gsdb@imagn.na.cnr.it

**Abstract.** Tools for quantitative analysis of volume images are becoming more important, as volume images are becoming more common in a number of application fields, but especially in biomedical tomographic images at different scales. Here we present a method for reducing a volume (3D) object to a surface skeleton. The original object can be recovered from its skeleton. The method is based on the notion of “multiple voxels,” derived from that of “multiple pixels” in the 2D case. It consists of two phases. During the first phase non-multiple voxels are iteratively removed. During the second phase, the remaining set of voxels is thinned to a set of one-voxel thick surfaces and curves. This skeletonization method requires only a small number of local ( $3 \times 3 \times 3$  neighbourhood) operations per voxel, no extra memory and no look-up tables. It is suited both for sequential and parallel implementation. We exemplify the results of the method on a number of  $128 \times 128 \times 128$  images.

## 1 Introduction

Thinning denotes the process where objects are reduced to structures of lower dimension. Thinning is referred to as skeletonization if the process is reversible. Reversibility is important when processing and characterizing objects, otherwise too much of the shape information is lost. The only situation where reversibility is not so important is when processing band-like objects with known, constant width, such as letters. Skeletonization reduces planar objects in images to a set of planar curves, see [5], and objects in volume images to a set of 3D surfaces and curves, see [3, 7, 10] for recent examples. In volume images, skeletonization might furthermore compress the skeleton to a set of 3D curves, see [6, 11]. However, this would imply a considerable loss of shape information and would definitely prevent the reversibility of the thinning process. Moreover, a curve skeleton can be obtained only when dealing with solid objects without cavities.

The skeleton concept has proved very useful for processing the binary shape of planar objects. A large number of papers dealing with extracting or using the skeleton can be found in the literature. Although there have been several efforts

to generalise skeletonization to volume images, the literature on this subject is still in its early stages. Also, in previous papers on skeletonization of volume objects, the methods have mostly been exemplified using quite tiny test images, that makes it difficult to understand what the results would be in more application oriented examples. However, skeletonizing volume images is a promising approach for quantification and manipulation of volumetric shape. A reduction of volume data could also be useful for decreasing the storage requirements significantly.

In this paper we describe a method for reducing a volume (voxel) object to a surface skeleton. The original object can be recovered from its skeleton. Similarly to [7], our approach is based on the notion of "multiple voxels," derived from that of "multiple pixels" in the 2D case. The multiple pixels on the object contour have been proved to be equivalent to the pixels that cannot be removed during skeletonization, [1]. Skeletonization based on multiple pixel/voxel detection can not be applied to irregular objects whose border consists entirely of multiple elements, i.e. having extremely jagged outlines. These objects can generally be created only artificially, thus we will not consider them in this paper.

Our skeletonization algorithm consists of two phases. During the first phase non-multiple voxels are iteratively removed until an at most two-voxel thick set of skeletal voxels is identified (Sect. 2). During the second phase, this set is reduced to a set of one-voxel thick surfaces and curves (Sect. 3). We exemplify the results of the method on a number of  $128 \times 128 \times 128$  images (Sect. 4). Differently from other algorithms addressing the same problem, our skeletonization method requires only a small number of local operations per voxel, no extra memory and no look-up tables. It is easily implementable both in sequential and in parallel fashion. The method presented here can find applications in many fields, especially for different biomedical tomographic images, which is an ever increasing field, in great need of tools for volume image processing.

## 2 Identification of Multiple Voxels

An object is a black 26-connected set of voxels. The background is white and 6-connected (which is a necessary consequence of the object being 26-connected). Each voxel  $v$  has three types of neighbours, called face-, edge-, and point-neighbours, depending on whether they share a face, an edge, or a point with  $v$ . A *border* voxel is a black voxel with a white face-neighbour. Black voxels, which are not border voxels, are *internal* voxels.

The skeletal voxels are identified by an iterative algorithm, where each iteration consists of three steps.

**Step 1:** Identify all border voxels, and label them with the iteration number, if they are not already labelled.

**Step 2:** Inspect all voxels labelled with the current iteration number, and mark those that are multiple.

**Step 3:** Remove all unmarked border voxels.

The process terminates when all border voxels are multiple, and hence no more voxels can be removed. A few internal voxels might still be present at this stage, in regions where surfaces and/or curves meet. These internal voxels are not yet labelled, and are at this stage assigned a label equal to the final iteration number plus one. The labels assigned to the skeletal voxels can be used for recovery purposes.

Steps 1 and 3 do not need any further explanation. To identify multiple voxels, we have generalised an algorithm that, in the 2D case, identifies the multiple pixels on the contour of an 8-connected pattern, see [2]. The resulting skeletal pixels are labelled as they would be on the city block distance transform of the pattern. Moreover, it can be shown that the multiple pixels include all the pixels that are local maxima of the city block transform.

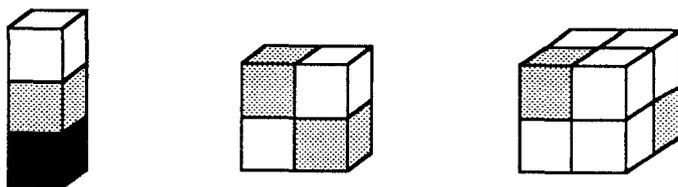
For volume images, we define a border voxel  $v$  as multiple if any of the following conditions is satisfied:

**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  $2 \times 2$  neighbourhood of  $v$  (in any of the three principal planes) exists such that the edge-neighbour of  $v$  is a border voxel, 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, while the other six neighbours are background voxels.

The conditions are illustrated in Fig. 1. Condition A1 means that the configuration to the left in the figure does not occur in any of the three principal planes. Condition A2 means that the configuration in the middle, or any of its rotations, occurs and Condition A3 means that the configuration to the right, or any of its rotations, occurs. Condition A1 prevents the creation of cavities and all three conditions prevents disconnecting the object. Condition A3 is unique to the three-dimensional case and is necessary as the object is defined as 26-connected.



**Fig. 1.** The voxel configurations in Conditions A1–A3. Internal voxels are black, border voxels are grey, and background voxels are white. Rotations of the configurations results in six cases for Condition A1 (left), twelve cases for Condition A2 (middle), and eight cases for Condition A3 (right).

The skeletal voxels found by this algorithm constitute a 26-connected set of voxels. The set is at most two-voxel thick. The voxels are labelled with the iteration number, which means that they have the same label as they would have if the  $D^6$  distance transformation had been applied to the object. The  $D^6$  distance is the volume image equivalent of the city block distance, see [4]. This equivalence with the  $D^6$  distance depends on the definition of border voxel, which determines that each successive border layer is 6-connected to the previous layer, exactly as the layers are ordered in the  $D^6$  distance transform. Due to Condition A1, the skeletal voxels include all the voxels that are local maxima of the  $D^6$  distance transform of the object. A local maximum is defined as a voxel with label larger than or equal to that of its face-neighbours. In our case, the label is the iteration number, but as we have already remarked, this is equal to the  $D^6$  distance value. None of the face-neighbours of a local maximum can then be an internal voxel (an internal voxel would be labelled with the next iteration number!) and thus Condition A1 holds for the local maximum. As local maxima are not removed, object recovery is possible, using the reverse  $D^6$  distance transformation, see [8].

### 3 The Surface Skeleton

The set obtained using the algorithm described in the previous section can be reduced to unit thickness applying a thinning process, that has to be split into three directional processes, each of them applied once. Using directional processes is necessary to avoid breaking connectedness and excessive shortening. Only three directional processes, rather than six, are used, since the set of the skeletal voxels is already at most two-voxel thick.

The three processes occur in the directions Top-Down, Left-Right, and Front-Back, see Fig. 2. A  $Z$ -voxel is defined as a skeletal voxel with a background voxel as the face-neighbour in the  $Z$ -direction. We describe the Top-Down process below (the other two are similar).

In the Top-Down process Top-voxels are candidates for removal. A Top-voxel  $v$  is removed if all the following three conditions are satisfied:

**Condition B1:** The face-neighbour of  $v$  in the down position is a Down-voxel.

(The corresponding four-voxel configuration is illustrated in Fig. 3.)

**Condition B2:** Condition A2 does not occur.

**Condition B3:** Condition A3 does not occur.

Condition B1 guarantees that the current Top-voxel belongs to a portion of the set of the skeletal voxels which is two-voxel thick in the Top-Down direction. For example, see Fig. 4, illustrating a set that is two-voxel thick in the Front-Back direction. The Top-voxels of the set do not satisfy Condition B1, so we avoid removing them during the Top-Down process and therefore unwanted "shrinking" of the set is avoided.

Condition B2 prevents breaking the connectedness of the set when the connection occurs between edge-neighbours only. Only four of the twelve rotational

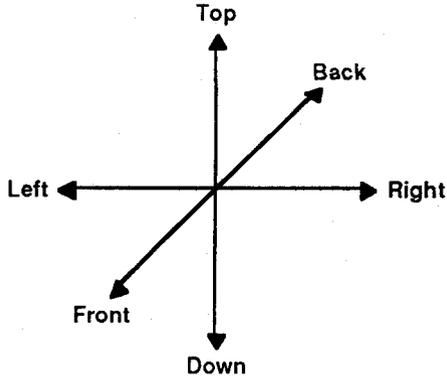


Fig. 2. Directions in voxel space.

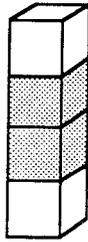


Fig. 3. Configuration for removing Top-voxels (see text). Border voxels are grey, background voxels are white.

cases of Condition A2 (see middle of Fig. 1) have to be checked, as Condition B1 guarantees connection in the other cases. The cases to be checked are those where the edge-neighbour connects to one of the four top edges of the voxel.

Condition B3 prevents breaking the connectedness of the set when the connection occurs between point-neighbours only. As Condition B1 guarantees connection in four cases, Condition A3 has to be checked only for the cases where the point-neighbour connects to one of the four top corners of the voxel (see the right of Fig. 1).

After the three directional processes have been applied, the skeleton consists of thin surfaces and curves. Some of the local maxima of the implied distance transform have been removed during this thinning phase, so complete object recovery by the reverse distance transformation is no longer possible. However, the voxels that are not recovered are all border voxels of the original object, so the distortion of the shape is not great.

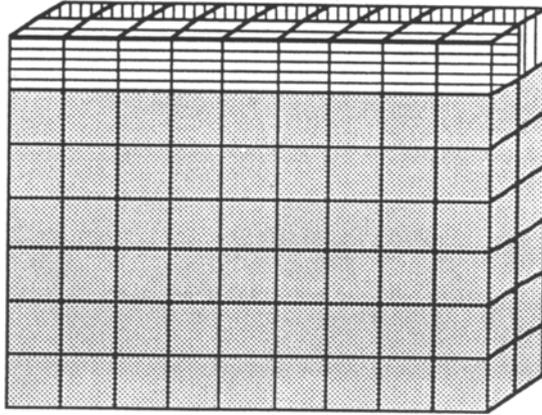


Fig. 4. A voxel set, two voxels thick in the Front-Back direction. The Top-voxels (hatched) are not removed in the Top-Down process, but the horizontally hatched Top-voxels are also Front-voxels and will be removed during the Front-Back process.

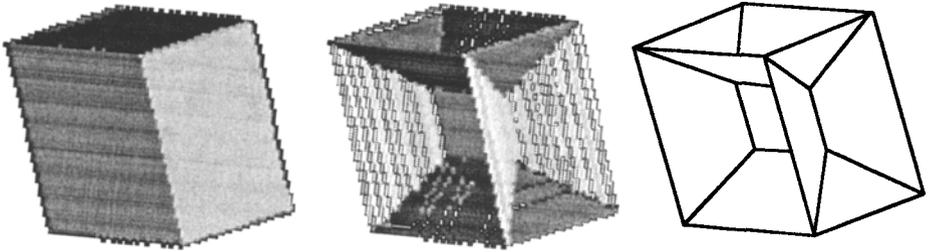
## 4 Examples

Our examples are volume objects synthesised in  $128 \times 128 \times 128$  images. For the visualisation we are using *ANALYZE<sup>TM</sup>*, [9]. The authors are well aware of the difficulty in interpreting a single projection of the thin surface skeletons. A better way to visualise this kind of images is a rotation sequence of projections. To alleviate the problem to some degree, we have made line drawings representing the skeletons (when possible). We assure the reader, that all skeletons *are* connected and without holes, even though the figures could sometimes be misleading in this respect. (Synthesised test images like these ones can be easily achieved using the  $D^{26}$  reverse distance transformation on a few seed labels, see [8].)

A first example of the results of our method is illustrated in Fig. 5. A binary volume object, a box of size  $40 \times 60 \times 80$  voxels, is shown to the left. The surface skeleton of the box is found in the middle. The CAD-like skeleton representation is shown to the right. The skeleton consists of 5.4 % of the original voxels, and contains almost all the information present in the original image. The original object can thus be recovered, except for some of the original surface voxels.

In Fig. 6 the result of the skeletonization of a pyramid can be seen.

Figure 7 illustrates how the algorithm handles an object with a tunnel. In this case, the skeleton does not consist only of planes. To make the skeleton clearer, a cut was made perpendicularly through the middle of the  $31 \times 31 \times 81$  block, see the bottom of the figure. The various skeleton surfaces are rather complex at the ends of the tunnel.



**Fig. 5.** A box (left), its surface skeleton of labelled voxels (middle), and a line drawing representing the skeleton (right).

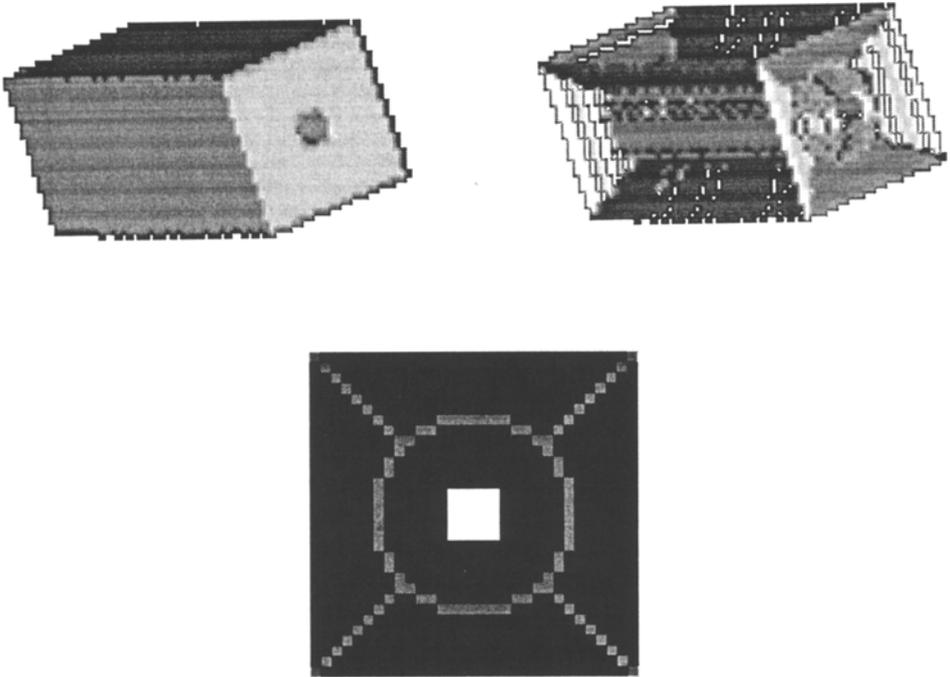


**Fig. 6.** A pyramid (left), its surface skeleton (middle), and a line drawing representing the skeleton (right).

## 5 Conclusion

We have developed a tool that promises to be at least as useful for volume images as the corresponding tool has been for 2D images. A reduction of data is of even greater importance for volume images than for 2D images. And as volume images are becoming more common, especially in different biomedical fields, there is a need for volume image processing tools. Note that reduction of data could immediately be achieved by simply identifying the local maxima in the  $D^6$  distance transform. However, skeletonization is preferable, since the set of local maxima does not reflect the topological properties of the shape. In fact, not even the number of components can be determined.

Our skeletonization algorithm is based on simple  $3 \times 3 \times 3$  neighbourhood operations (except for the thinning operation that also uses a  $4 \times 1 \times 1$  neighbourhood). It allows almost complete object recovery and is topology preserving. It



**Fig. 7.** A tunnel (upper left), its surface skeleton (upper right), and a cut through the middle of the tunnel, where skeleton voxels are grey (bottom).

is suited both for sequential and parallel implementation. We have implemented the sequential algorithm. We observe that this skeletonization method is not invariant under object rotation as the digital distance metric implicit in the iterated process used for finding the multiple voxels is the  $D^6$  metric, which is quite rotation dependent. Future work will be directed to achieving more rotation independent results.

## Acknowledgements

This work was financially supported by the Wenner-Gren Center Foundation for Scientific Research, Stockholm. Scientific support were given by Prof. Ewert Bengtsson and Dr. Bo Nordin. Both types of support are gratefully acknowledged.

## References

1. Arcelli, C., Sanniti di Baja, G.: A contour characterization for multiply connected figures. *Pattern Recognition Letters* **6** (1987) 245-249
2. Arcelli, C., Sanniti di Baja, G.: A one-pass two-operations process to detect the skeletal pixels on the 4-distance transform. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **11** (1989) 411-414
3. Bertrand, G.: A parallel thinning algorithm for medial surfaces. *Pattern Recognition Letters* **16** (1995) 979-986
4. Borgefors, G.: Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing* **27** (1984) 321-345
5. Lam, L., Lee, S. W., Suen, C. Y.: Thinning methodologies - A comprehensive survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **14** (1992) 869-885
6. Lee, T., Kashyap, R. L., Chu, C.: Building skeleton models via 3-D medial surface/axis thinning algorithms. *CVGIP: Graphical Models and Image Processing* **56**(6) (1994) 462-478
7. Miguet, S., Marion-Poty, V.: A new 2-D and 3-D thinning algorithm based on successive border generations. *Proc. 4<sup>th</sup> Discrete Geometry for Computer Imagery*. Grenoble, France (1994) 195-206
8. Nyström, I., Borgefors, G.: Synthesising objects and scenes using the reverse distance transformation in 2D and 3D. *Proc. 8<sup>th</sup> International Conference on Image Analysis and Processing*, Sanremo, Italy (1995) 441-446
9. Robb, R. A., Barillot, C.: Interactive display and analysis of 3-D medical images *IEEE Transactions on Medical Imaging*. **8**(3) (1989) 217-226
10. Saha, P. K., Chaudhuri, B. B.: Detection of 3-D simple points for topology preserving transformations with application to thinning. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **16**(10) (1994) 1028-1032
11. Saito, T., Toriwaki, J.: A sequential thinning algorithm for three dimensional digital pictures using the Euclidean distance transformation. *Proc. 9<sup>th</sup> Scandinavian Conference on Image Analysis*, Uppsala, Sweden (1995) 507-516