Boundary Surface Shrinking – a Continuous Approach to 3D Center Line Extraction

Hartmut Schirmacher*, Malte Zöckler, Detlev Stalling, Hans-Christian Hege

Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)
Takustr. 7, D-14195 Berlin-Dahlem, Germany
Email: {schirmacher,zoeckler,stalling,hege}@zib.de

Abstract

We present a new algorithm for efficient and robust approximation of skeletons and construction of connected center line graphs from 3D image data. The algorithm is based on the idea of shrinking the boundary surface along the gradients of the object's distance map. After transforming the surface in that way, duplicate vertices and line segments are eliminated. If the skeleton contains no medial surfaces, a graph representation of the centerline can directly be extracted from the remaining edges. The algorithm has proved to perform well on several different datasets.

1 Introduction

Much work has been done in the field of skeleton-based shape analysis. One application of increasing interest is the extraction of center line graphs from skeletons in order to represent branching structures like for example neurons with their axons, dendrites, and spines. Existing 3D skeletonization methods suffer from several problems, the most severe ones being the discrepancy between discrete and continuous representation and the high sensibility to noise. This is a considerable drawback in the context of medical and biological applications.

We propose an alternative approach to skeletonization and immediate center line extraction which is based on the simple idea of *shrinking* the boundary surface of the object until it approximates the skeleton. This new algorithm is very robust with respect to noise and discretization and makes it possible to directly extract connected center line graphs in $\mathcal{R}^3$ from the skeleton edges.

2 Previous Work

In 1967, Blum introduced the *skeleton* to describe the symmetry axes of arbitrary shapes based on the *medial axis transform* [1, 2]. His definition is based on the locus points of grass fire fronts originating simultaneously from all points of the object boundary.

One category of skeletonization algorithms based on Blum's ideas compute a so-called *distance map* containing the minimal distance to the object boundary for each discrete point or cell. Several methods have been proposed to compute those maps and properly extract the skeleton points [3, 6, 5].

Another category of algorithms uses topological thinning methods to reduce the object to a skeleton-like shape by iteratively removing so-called *simple points*, the removal of which does not change the object's topology [7, 8, 4].

These first two categories of algorithms work in discrete space. The most popular *continuous* skeletonization methods in 2D and 3D are those based on Voronoi diagrams [11, 10].

All of the methods mentioned above suffer from a high sensibility to noise which usually produces many "spikes" that have to be removed in a post-processing step by pruning all branches which are shorter than a certain threshold. This does not seem adequate in many cases, e.g. when the object's local diameter varies strongly. Additionally, the discrete algorithms have to cope with a number of artifacts which come along with distance operations in discrete space [6], and they require a non-trivial step for constructing a graph

[*] Hartmut Schirmacher is currently with the computer graphics group at Erlangen University and has performed his research at the ZIB as part of his diploma thesis.

in $\mathcal{R}^3$ from the discrete skeleton. For Voronoi algorithms, similar discretization problems arise during the transition from the discrete data to $\mathcal{R}^3$ by sampling the object boundary.

In the next section, we show how to use the distance gradient in combination with the object's boundary surface for approximating the object's skeleton and extracting center lines. The algorithm uses a robust transition from discrete to continous space, allows for direct extraction of connected 3D graphs, and has proved to perform well even for large biological data as we will show in section 4.

3 The Shrinking Approach

The basic idea of our approach is to use the boundary surface as well as the *gradient* of the object's distance map. The surface vertices are then translated in direction of the gradient vectors pointing towards the center of the object (cf. Fig. 2). During this transition no edges are cut, i.e., the connectivity information of the boundary surface is fully preserved. After a final simplification step, the object's skeleton edges provide a graph description of the center lines.

3.1 The Boundary Surface

The starting point of the new algorithm is a boundary surface of the object, which can be obtained from the voxel boundaries of the segmented binary object or by applying the marching cubes algorithm [9]. Our approach yields the same result for different inital surfaces as long as they are of similar topology.

The initial surface consists of a set of vertices, edges, and faces in $\mathcal{R}^3$. This is a first means for operating in continous space rather than on discrete voxels.

3.2 Using the Distance Gradient

There are several efficient methods for computing the distance map for an arbitrary binary object in 2D or 3D (see Section 2). The distance map contains the minimal distance to the object's boundary for every voxel, measured in some suitable metric. Figure 1 shows the volume data visualization of a carrot-shape's distance map.

Interpreting the distance map as a 3D scalar field, we can compute its gradient at each point.

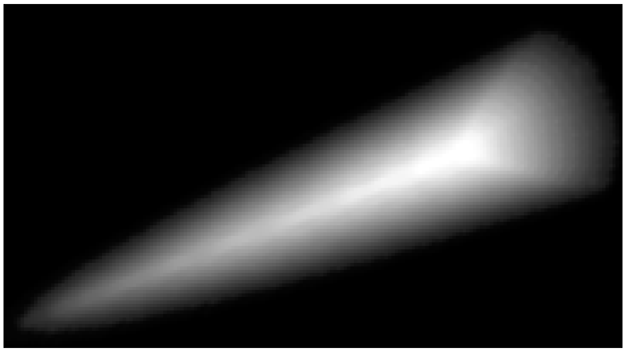

Figure 1: Distance map for a “carrot” shape. Higher distance values appear brighter.

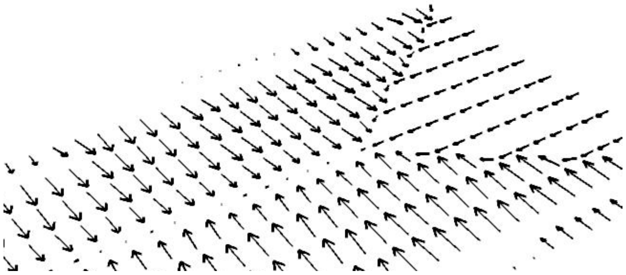

Figure 2: Closeup view of a cut through the distance gradient. The vectors point towards the medial axis and nearly vanish there.

The result is a 3D vector field as shown in Figure 2. The vectors point towards the medial axis of the object. In the center of symmetry they nearly vanish. In order to compute gradient vectors inside a voxel we use trilinear interpolation, which also works as a smoothing filter. This is also of some importance in context of noise sensibility. The gradient of the distance field at a point $\boldsymbol{x}$ will simply be denoted by $\nabla D(\boldsymbol{x})$ throughout this text.

In the next section we will discuss how to make use of the distance gradient for transforming the geometry of the surface constructed in Section 3.1.

3.3 The Shrinking Iteration

The key idea of the new algorithm is to iteratively translate the vertices of the inital surface along the gradient vectors at their current positions. The position of vertex i in iteration $n+1$ is thus given by:

$$\boldsymbol{x}_i^{(n+1)} := \boldsymbol{x}_i^{(n)} + h \, \nabla D(\boldsymbol{x}_i^{(n)}) \, , \qquad (1)$$

where h determines the size of a translation step. In order to prevent vertices from moving too far apart from their edge neighbours, we introduce *constraints.* The constraints must allow a pair of neighbours to approach each other, but prevent them from moving too far apart. In order to apply the desired constraints, we look at two vertices i and j defining the edge $\boldsymbol{r}_{i,j}^{(n)} = \boldsymbol{x}_j^{(n)} - \boldsymbol{x}_i^{(n)}$ with current length $\left|\boldsymbol{r}_{i,j}^{(n)}\right|$ and define

$$\boldsymbol{c}_{i,j}^{(n)} \; := \; \left(\left|\boldsymbol{r}_{i,j}^{(n)}\right| - \left|\hat{\boldsymbol{r}}_{i,j}^{(n)}\right|\right) \frac{\boldsymbol{r}_{i,j}^{(n)}}{\left|\boldsymbol{r}_{i,j}^{(n)}\right|} \qquad (2)$$

with the minimal length so far being

$$\left|\hat{\boldsymbol{r}}_{i,j}^{(n)}\right| \; := \; \min\left\{\left|\boldsymbol{r}_{i,j}^{(m)}\right| \text{with } 0 \leq m \leq n\right\} \; .$$

Using the minimal edge length accounts for the fact that a constraint should effect the current translation as soon as an edge would get longer. We incorporate Eq. 2 into Eq. 1 and get the following constrained shrinking scheme:

$$\begin{aligned} \boldsymbol{x}_i^{(n+1)} \; := \; \boldsymbol{x}_i^{(n)} \; & + \; h \nabla D(\boldsymbol{x}_i^{(n)}) \\ & + \; w \sum_{j \in N(i)} \boldsymbol{c}_{i,j}^{(n)} \; , \qquad (3) \end{aligned}$$

where $N(i)$ is the set of edge neighbours of vertex i. The weight factor w controls the importance of the constraints, while h still determines the size of a single shrinking step. We obtained good results using values from 0.1 to 0.5 for w and half the voxel diameter for h.

It has already been mentioned that the gradient vectors become very small on the medial axis of an object. However, in general they do not vanish exactly. Therefore, in order to prevent the vertices from shifting along the medial axis, we stop to propagate them as soon as the length of the gradient vector becomes less than a certain threshold. In practice we use a value of 0.05. The total number of required iterations depends on the greatest object radius, i.e., the maximum value of the distance map.

3.4 Simplification

One important benefit of our approach is that the connectivity inherent in the surface description can be used directly after the shrinking phase to extract graph edges for the skeleton. To reduce complexity, vertices with similar coordinates and edges with identical vertices are removed. We choose a thresholding distance ϵ (usually of subvoxel size) and sort the skeleton vertices into an octree which is constructed in such way that all vertices within a distance ϵ can be found in the 26-neighbourhood of a leaf node. All such vertices are then clustered into a single one.

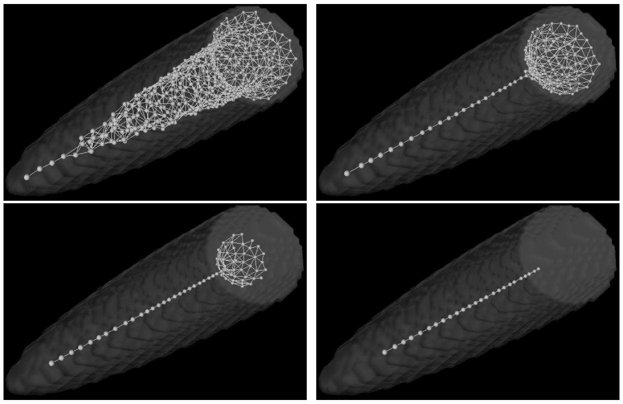

Figure 3: Initial surface and extracted graph of the "carrot" after 10, 20, 40, and 80 shrinking iterations. Note how the funnel at the bigger end of the shape disappears.

It is important to note that in contrast to the pruning step required in the postprocessing of other skeletonization approaches, our simplification step is just a graph-based elimination of approximate duplicates employing a much smaller threshold.

4 Results

We have implemented the shrinking and simplification steps within our visualization framework HyperPlan which also provided the necessary modules Isosurface and Gradient as well as visualization methods for volumes, surfaces, and line sets.

Figure 3 shows the boundary surface of the "carrot" example shape during various stages of the shrinking process, resulting in a simple connected graph representation. Figure 4 shows the graph extracted from a rather large neuron dataset. The skeleton contains medial surfaces where the cross-section's center of symmetry is not point-like.

The algorithm proved to be quite robust in several ways. It showed no noticable dependency on

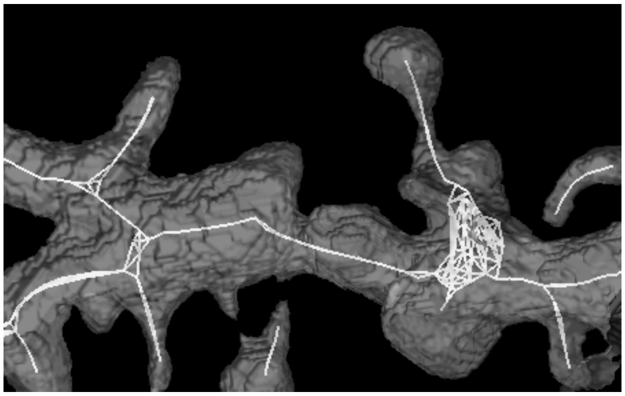

Figure 4: Graph extracted from a neuron dataset. Note the resulting medial surfaces in regions of extremely non-symmetric cross-sections

the choice of the initial boundary surface, and it performed well for different values of the free parameters mentioned in Sections 3.1 and 3.3 as well.

5 Conclusions and Future Work

We have presented an alternative approach for computing approximate skeletons and center line graphs from arbitrary 3D datasets. Due to the use of a gradient and a surface in continuous space, the new algorithm does not exhibit the usual discretization problems of previous approaches and has proved to perform well on several different datasets.

Future research on this topic will be done in several directions. The major remaining challenge for obtaining center lines is to remove the medial surfaces which occur where the object's cross-section is not point-symmetric. Second, the resulting graph nodes should be re-centered in a postprocessing step in order to correct for the error introduced by the clustering process mentioned in Section 3.4. While doing this, the graph can for example be attributed with the object radius (or with the two radii of an ellipsoid) in every node to allow for reconstruction of the object shape.

Acknowledgements. The authors would like to thank Martin Seebaß for the idea of looking at the distance gradients rather than at the scalar values. The neuron datasets have been provided by IFN Magdeburg.

References

[1] H. Blum. A transformation for extracting new descriptors of shape. In *Models for the Perception of Speech and Visual Form.* MIT Press, 1967.

[2] H. Blum. Biological shape and visual science: Part I. *J. Theoretical Biology*, 38:205–287, 1973.

[3] G. Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27:321–345, 1984.

[4] U. Eckhardt and L. Latecki. Invariant thinning and distance transform. *Computing Supplementum*, 11:21–36, 1996.

[5] N. Gagvani and D. Silver. Parameter controlled skeletonization of three dimensional objects. Technical Report CAIP-TR-216, Rutgers State University of New Jersey, June 1997.

[6] Y. Ge and J.M. Fitzpatrick. On the generation of skeletons from discrete euclidian distance maps. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(11):1055–1066, November 1996.

[7] L. Lam, S.-W. Lee, and C.Y. Suen. Thinning methodologies – a comprehensive survey. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(9), September 1992.

[8] T.-C. Lee, R.L. Kashyap, and C.-N. Chu. Building skeleton models via 3-D medial surface/axis thinning algorithms. *Graphical Models and Image Processing*, 56(6):462–478, November 1994.

[9] W.E. Lorensen and H.E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163, 1987.

[10] M. Näf, O. Kübler, R. Kikinis, M.E. Shenton, and G. Székely. Characterization and recognition of 3D organ shape in medical image analysis using skeletonization. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, San Francisco, June 1996.

[11] R.L. Ogniewicz. *Discrete Voronoi Skeletons.* PhD thesis, Swiss Federal Institute of Technology, Zurich, 1992. Dissertation No. 9876.