1

Computing Center-Lines: An Application of Vector Field Topology

Thomas Wischgoll

Wright State University, thomas.wischgoll@wright.edu

Summary. Curve-skeletons of 3-D objects are medial axes shrunk to a single line. There are several applications for curve-skeletons. For example, animation of 3-D objects, such as an animal or a human, as well as planning of flight paths for virtual colonoscopy. Other applications are the extraction of center lines within blood vessels where center lines are used to quantitatively measure vessel length, vessel diameter, and angles between vessels. The described method computes curve-skeletons based on a vector field that is orthogonal to the object's boundary surface. A topological analysis of this field then yields the center lines of the curve-skeletons. In contrast to previous methods, the vector field does not need to be computed for every sampled point of the entire volume. Instead, the vector field is determined only on the sample points on the boundary surface of the objects. Since most of the computational time was spent on calculating the force field in previous methods, the proposed approach requires significantly less time compared to previous vector-based techniques while still achieving a better accuracy and robustness compared to methods based on Voronoi tessellations.

1.1 Introduction

Curve-skeletons describe the very basic features of an object. They describe a thinned version of the object represented as some type of stick model resulting in the center-lines of the object. Therefore, the use of curve-skeletons can prove useful for applications, such as animation [43] or flight planning for virtual colonoscopy [19]. Similarly, accurate curve-skeleton methods can be used for extracting quantitative measurements from computed tomography (CT) scanned vascular structures. Here, the curve-skeleton describes the center lines of the vessels. These can then be used to measure vessel radius, vessel lengths, and angles between vessels within the volumetric data set retrieved by using the CT scanner. This is the application that motivated the development of the algorithm described in this paper. In order to derive these measurements from the volumetric data set, an accurate extraction method for curve-skeletons is desirable. For example, thinning-based techniques that

work in the voxel space of the volumetric data set tend to generate jagged lines which are in no way suitable for determining angles between vessels. Similarly, inaccuracies can occur when computing the radii of the vessels. Hence, an approach that only uses the volumetric data set in order to identify the boundary surface of the contained object is more promising.

The algorithm described in this paper is exactly of this type. It is capable of extracting the boundary surface of an object that is defined by a volumetric data set at sub-voxel level. For this, it determines the location of the maximal gradient within the volumetric data set similar to Canny's [9] maxima-suppression technique but extended to three dimensions. Since the algorithm only relies on points extracted from the volumetric data set but not on its underlying structured grid, it can also be applied to objects defined by a point set without any restrictions.

Techniques used for computing the topological graph of a vector field are applied to determine the curve-skeleton. First, for all points on the object's boundary vectors are computed that are orthogonal to the boundary surface. There are different options for computing these vectors. They either can be derived by determining the normal vector of a plane that is defined by a least-square fit of the point and its neighbors. Or – in case of the object being defined by a volumetric data set – the image gradients determined in the previous step can be used. In both cases, the normal vectors can be determined in such a way that they are facing inwards with respect to the object. The entire vector field can then be determined by computing a tetrahedrization of the entire point cloud and then linearly interpolating within the tetrahedra. In order to ensure that only the curve-skeleton inside the object is extracted, all tetrahedra that are located outside the object are removed based on the normal vectors.

A topological analysis of the vector field within the faces of every tetrahedron yields points on the curve-skeleton. By following the topology of the tetrahedrization, points on the curve-skeleton within neighboring tetrahedra can be connected resulting in the entire curve-skeleton.

A detailed description of the algorithm can be found in section 1.3. The next section illustrates related work and compares it to the described approach. Subsequently, the theoretical background with regard to the topological analysis of vector fields is explained. Section 1.4 shows results of the algorithm applied to various data sets, followed by conclusions and future work.

1.2 Related Work

Several approaches for extracting curve-skeletons or medial axes can be found in the literature. A very good overview of available techniques can be found in the paper by Cornea et al. [11].

Some methods start with all voxels of a volumetric data set and use a thinning technique to shrink down the object to a single line. Directional

thinning approaches use a specific order in which voxels are removed. For example, directions, such as up or down, are used to define this order and conditions are used to identify endpoints [3, 8, 18, 23, 25, 30, 31, 34, 42]. Since these methods are sensitive with respect to the order in which the voxels are removed the resulting curve-skeleton may not be centered. Non-directional methods [6, 39] or fully parallel approaches [14, 27, 29] do not suffer from this disadvantage. Ideally, the topology of the object should be observed. Such an approach was proposed by Lobregt et al. [24] which is the basic technique used in commercial software systems, such as Analyze^{TM} developed by the Mayo Clinic. The disadvantage of this approach is that it tends to produce jagged lines which do not allow accurate measurements of angles between parts of the object, such as individual vessels of a vascular structure. Other approaches [40] classify the voxels in different groups, such as edge, inner, curve, or junction and re-classify after removal of a voxel. A similar algorithm is proposed by Palagyi et al. [30]. The disadvantage of thinning algorithms is that they can only be applied to volumetric data sets due to the nature of these algorithms.

To avoid this disadvantage, other approaches deploy the distance transform [17] or distance field in order to obtain a curve-skeleton. For each point inside the object, the smallest distance to the boundary surface is determined. For this, the Euclidian metric or the <3,4,5> metric [5] can be used. Also, fast marching methods [36, 41] can be deployed to compute the distance field. Voxels representing the center lines of the object are identified by finding ridges in the distance field. The resulting candidates must then be pruned first. The resulting values are then connected using a path connection or minimum span tree algorithm [38, 44, 48]. Methods used to identify points on the ridges include distance thinning [10, 15, 16, 33], divergence computing [7], gradient searching [4], thresholding the bisector angle [28], geodesic front propagation [32], or shrinking the surface along the gradient of the distance field [35]. The distance field can also be combined with a distance-from-source field to compute a skeleton [49]. Based on an anisotropic diffusion applied to the image gradients, Yu et al [47] extract skeletons from 2D images.

Techniques based on Voronoi diagrams [2, 13] define a medial axis using the Voronoi points. Since this approach usually does not result in a single line but rather a surface shaped object, the points need to be clustered and connected in order to obtain a curve-skeleton. Voronoi-based methods can be applied to volumetric data sets as well as point sets. Due to the fact that clustering of the resulting points is required these approaches can lack some accuracy.

1.3 Methodology

It is assumed that the reader is familiar with singularities in vector fields and 2-D vector field topology. If necessary, a good overview of these topics can be found in [20, 45, 46].

The algorithm for determining the curve-skeleton consists of several steps. If the object is given as a volumetric data set the object's boundary has to be extracted first. Then, a vector field is computed that is orthogonal to the object's boundary surface. Once the vector field is computed, the curve-skeleton can be determined by applying a topological analysis to this vector field. The following subsections explain these steps in detail.

1.3.1 Extracting the Boundary of the Object

If the object is given as a set of points, for example measured by a laser scanner, the object's boundary is already defined. If the object is defined by a volumetric data set, for example from a CT scan, the boundary of the object has to be determined first. A volumetric data set consists of voxels aligned along a regular, three-dimensional grid. Since it is generally not likely that the boundary of the original object is exactly located at these voxels, better precision can be achieved by finding the exact location between a set of voxels. Since an accurate representation of the object's boundary is crucial to the algorithm, improving the precision is an essential step. The method used within the described system uses similar techniques as described by Canny's non-maxima suppression [9, 21] but extended to three dimensions.

First, the image gradients are computed. Using a fixed threshold, all voxels with a gradient length below this threshold are neglected. Then, the gradients of the remaining voxels are compared to its neighbors to identify local maxima along the gradient. In 3-D, the direct neighborhood of a single voxel generally consists of 26 voxels forming a cube that surrounds the current voxel. In order to find the local maximum along the current gradient, the gradients in the neighborhood in positive and negative direction of the current gradient have to be determined. The current implementation of the described system uses tri-linear interpolation. Based on these three gradient values, i.e. the original image gradient and the two interpolated values, the location of the maximal gradient can be identified, resulting in a more accurate representation of the object's boundary and therefore a more precise center line. Hence, this step improves the accuracy of the resulting curve-skeleton. However, the algorithm would still work using the original voxels identified.

Once all points on the boundary are extracted from the volumetric data set using this gradient approach with sub-voxel precision, the resulting point cloud can be further processed in order to identify the skeleton.

1.3.2 Computing the Vector Field

The described method computes a curve-skeleton by applying a topological analysis to a vector field that is determined based on the geometric configuration of the object of which the curve-skeleton is to be determined. The vector field is computed in such a way that the vectors are orthogonal to the object's boundary surface. The vectors inside the object are then interpolated linearly.

Different approaches are possible for calculating such a vector field. A repulsive force field can be determined that uses the surrounding points on the object's boundary surface as used by Cornea et al. [12]. The repulsive force is defined similarly to the repulsive force of a generalized potential field [1, 22]. The basic idea is to simulate a potential field that is generated by the force field inside the object by charging the object's boundary. Another way is to define a normal vector by using the neighboring points in addition to the current one and then defining a plane that is approximated by these points. The normal of this plane then defines the vector corresponding to the current point.

If the data is given as a volumetric data set the image gradients can be used to define the vectors on the object's boundary surface. These image gradients are computed already in the previous step as they are needed for extracting the object's boundary and determining the sub-voxel precision. The image gradients are computed using the derivative of a Gaussian which also results in a smoothing of the gradient vectors to address noise that may be present in the data. In our experiments, the resulting gradient vectors were sufficiently accurate to determine an accurate center line as shown by the validation of the algorithm. Note that all three methods result in vectors pointing to the inside of the object.

1.3.3 Determining the Curve-Skeleton

In order to determine the curve-skeleton of the object, a tetrahedrization of all points on the object's boundary is computed first. For this, Si's [37] very fast implementation of a Delaunay tetrahedrization algorithm is used. By using the previously computed vectors that point to the inside of the object, outside tetrahedra can be distinguished from tetrahedra that are located inside the object. This way, all outside tetrahedra can be removed, leaving a Delaunay tetrahedrization of the inside of the object only. Since vectors are known for each vertex of every tetrahedron, the complete vector field can be computed using this tetrahedrization by interpolating linearly within each tetrahedron. This vector field is then used to identify points of the curve-skeleton which are connected which each other later on.

Since the vector field is now defined within the entire object, one could use an approach similar to the one used by Cornea et al. [12] at this point and compute the 3-D topological skeleton of the vector field which yields the curve-skeleton of the object. However, since singularities are very rare in a

3-D vector field Cornea et al. had to introduce additional starting points for the separatrices, such as low divergence points and high curvature points, in order to get a good representation of the curve-skeleton. Therefore, a different approach is described in this paper that analyzes the vector field on the faces of the tetrahedra. In order to be able to perform a topological analysis on the faces of the tetrahedra, the vector field has to be projected onto those faces first. Since linear interpolation is used within the tetrahedra, it is sufficient to just project the vectors at the vertices onto each face and then interpolate linearly within the face using these newly computed vectors. Based on the resulting vector field, a topological analysis can be performed on each face of every tetrahedron.

Points on the curve-skeleton can then be identified by computing the singularities within the vector field interpolated within each and every face of the tetrahedrization. For example, for a perfectly cylindrical object, the vectors computed at the cylinder's boundary point directly at the center of the cylinder. When looking at the resulting vector field at a cross section of the cylinder, a focus singularity is located at the center of the cylinder within this cross section. The location of this focus singularity resembles a point on the curve-skeleton of the cylinder. Hence, a singularity within a face of a tetrahedron resembles a point of the curve-skeleton. Since the vectors at the object's boundary point inwards, only sinks need to be considered in order to identify the curve-skeleton. Due to the fact that not all objects are cylindrical in shape and due to numerical errors and tolerances, points on the curve-skeleton can be identified by looking for sinks that resemble focus and spiral singularities.

Obviously, only faces that are close to being a cross section of the object should be considered to identify points on the curve-skeleton. In order to determine tetrahedra whose faces resemble a cross-section of the object, the vectors at the vertices can be used. If the vectors at the vertices, which are orthogonal to the object's boundary, are approximately coplanar with the face, then this face describes a cross section of the object. As a test, the scalar product between the normal vector of the face and the vector at all three vertices can be used. If the result is smaller than a user-defined threshold this face is used to determine points on the curve-skeleton. A fixed threshold works for most data sets. However, in some cases a sub-optimal choice of this threshold can result in false bifurcations where small segments (usually just a single line segment) appears to be branching off of the main skeleton. Computing the singularity on one of the faces fulfilling the threshold criterion then results in a point which is part of the curve-skeleton. Since linear interpolation is used within the face, only a single singularity can be present in each face.

Once individual points of the curve-skeleton are computed by identifying the focus and spiral singularities within the faces of the tetrahedra, this set of points needs to be connected in order to retrieve the entire curve-skeleton. Since the tetrahedrization describes the topology of the object, the connectivity information of the tetrahedra can be used. Thus, identified points of the curve-skeleton of neighboring tetrahedra are connected with each other

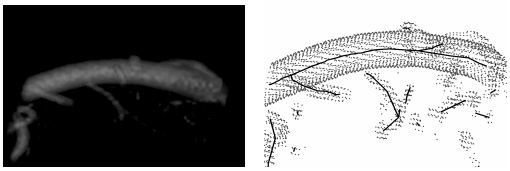


Fig. 1.1. Sub-section of the porcine heart data set visualized as a volume rendered image (left) and the extracted curve-skeleton of the same sub-section of the porcine heart data set (right).

forming the entire curve-skeleton. In some occurances, an additional gap closing step is required. Particularly, this is sometimes necessary at bifurcations where a smaller vessel branches off of a comparably large one. An example can be seen in figure 1.1 where the connection between the major vessel and the vessel branching off at the center of the image is missing. This can be easily detected by looking at the terminal nodes of the graph representing the center lines. For all these nodes, the algorithm checks if there is another graph it can be connected to. If this connection is within a vessel, the connection is added. To check this, the tetrahedrization can be used since the connection is only inside a vessel if it is entirely covered by tetrahedra.

1.4 Results

The algorithm was tested on several different data sets. It was mainly designed for extracting center lines from CT scanned volumetric data sets of porcine hearts where the arterials were previously perfused with a contrast enhancing polymer and computing the vessel radii as the distance between the center line and the vessel boundary. Figure 1.2(left) shows an example of such a data set.

The described algorithm is capable of extracting the curve-skeleton from such a volumetric data set in order to identify the center lines of the arterial vessels. The resulting curve-skeleton is depicted in figure 1.2(right). The figure shows the curve-skeleton extracted by the described algorithm before the gap closing step was applied as well as the point set representing the vessel boundaries.

Due to the densely located vessels of the right coronary arterial (RCA) tree, the extracted curve-skeleton seems rather cluttered and it is difficult to distinguish between lines at different depths. However, the extracted curve-skeleton describes the center lines of the arterial vessels found within the data set very well as illustrated below. When using a sub-section of the porcine heart data set, it can be seen that the curve-skeleton is located at the center of the arterial vessels, as shown in figure 1.3.

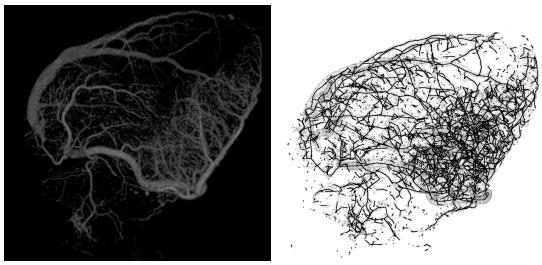


Fig. 1.2. Volume rendering (with shading enabled) of a previously perfused porcine heart which was scanned using a standard hospital CT scanner (left) and curve-skeleton of the porcine heart data set using the described algorithm beofer the gap closing step; to enhance visibility in the paper the thickness of the lines and points within the volumetric data set is increased (right).

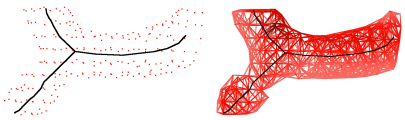


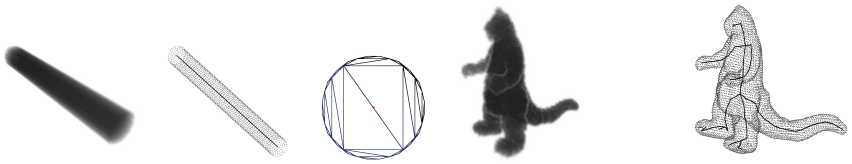
Fig. 1.3. Sub-section of the porcine heart data set showing the extracted center line (left) and the underlying tetrahedrization used for identifying the center line (right).

In order to validate the accuracy of the computed center lines, the vessel radii were computed for the main trunk of the arterial branches of a series of five porcine heart data sets as the distance between the center line and the vessel boundary and then compared to manual optical measurements. The optical measurements were performed by a domain expert after digesting away the tissue leaving only the contrast enhancing polymer used to fill the arteries. Based on the cast formed by the polymer, radii and distance to the most proximal vessel were then measured. Figure 1.7 (color plates) shows a comparison of the two different measurements for a typical data set. The agreement between the measurements is very good, with an error of $0.06mm$ (scan resolution was 0.6mm), which underlines the accuracy of the center lines. For the three major branches (LAD, LCX, and RCA) of the five procine hearts, the root mean square error between the two measurements is $0.16mm$ and the average deviation is $0.13mm$.

The described algorithm for extracting curve-skeletons has some definite advantages over, for example, Voronoi-based approaches. Voronoi-based approaches define a medial axis that is not necessarily a line, but more like a set of points defining a surface. In a post-processing step, these points need to be shrunk down to define a line. Even though the arterial vessels are rather round due to the fact that they are pressurized, a Voronoi-based algorithm only determines a fuzzy line around the actual center line of the vessels. In

addition, Voronoi-based approaches tend to generate significantly more false bifurcations in form of small, short vessels branching off. The root mean square error of the measurements computed using the presented technique of $0.16mm$ are more precise compared to other techniques found in the literature[26], where the root mean square error ranges from 0.2mm to 0.6mm with similar scan resolutions. The performance of the algorithm was compared to another vector-based algorithm by Conrea et al.[11]. The computation of the potential field as the first step of that algorithm would have taken several months. The presented algorithm requires only little more than an hour for the entire analysis of the data set.

The described algorithm works well with other types of data sets. The first example is a pure cylindrical shaped data set. The cylinder is perfectly round; hence, the algorithm should find a straight line as the curve-skeleton. As can be seen in figure 1.4(a), the algorithm generates the correct curve-skeleton for this data set exactly. The tetrahedrization of a small slice shown on the right illustrates the principles of the algorithm. There are two large triangles cutting through the cylinder with which a point on the center line is identified marked with a dot at the center of the image.



(a) (b) (c)

Fig. 1.4. Cylinder data set visualized as a volume rendered image (a) and the extracted curve-skeleton of the same data set (center); to enhance visibility in the paper the thickness of the lines and points within the image is increased. To illustrate the algorithm, the tetrahedrization within a small slice is shown as well (b). Monster data set visualized as a volume rendered image (left) and the extracted curve-skeleton of the same data set (right); to enhance visibility in the paper the thickness of the lines and points within the image is increased.

Figure 1.4(c) depicts the results of the next example, the monster data set. Again, both the volume rendered image and the curve-skeleton including the point describing the object's boundary are shown. This example shows that the algorithm works well even with non-tubular objects. More examples can be found on the color plates.

1.5 Conclusions and Future Work

In this paper, an algorithm for extracting a curve-skeleton from data sets given as point clouds or volumetric data sets was presented. The described

algorithm is based on a topological analysis of a vector field derived from the configuration of the point set describing the object's boundary contained in the data set. Due to the fact that it is no longer necessary to compute the vector field on a multitude of points but instead only for points on the object's boundary the described algorithm is significantly faster while still preserving a high accuracy of the extracted curve-skeleton. It took the algorithm a few seconds to extract the curve-skeleton for the smaller data sets. For the porcine heart data set, the algorithm needed a little more than an hour to determine the curve-skeleton. This is comparably fast, considering that in our tests Cornea's et al. [12] algorithm would have required several months to compute the potential field alone.

It is planned to use the described algorithm for deriving precise quantitative measurements from CT scanned specimens, such as vascular structures. In order to be able to measure vessel lengths, vessel diameters, and bifurcation angles an accurate representation of the center lines of the vessels is required. These center lines can be determined by the described algorithm for extracting curve-skeletons applied to the volumetric data set generated by a CT scanner.

1.6 Acknowledgments

The author would like to thank Ghassan Kassab and Jenny Choy-Zorrilla for providing the porcine heart data sets, including the manual measurements. I would also like to thank the Wright State University and the Ohio Board of Regents for supporting this research as well as Hang Si and Nicu Cornea for making their source code publicly available and providing data sets through their web pages.

References

[1] N. Ahuja, J.-H. Chuang, *Shape Representation Using a Generalized Potential Field Model*. IEEE Trans. Pattern Analysis and Machine Intelligence, 19(2):169-176, 1997.

[2] N. Amenta, S. Choi, R.-K. Kolluri, *The Power Crust,* Proc. of 6th ACM Symp. on Solid Modeling, pp 249-260, 2001.

[3] G. Bertrand and Z. Aktouf, *A three-dimensional thinning algorithm using subfields*, Vision Geometry III, 2356:113-124. SPIE, 1994.

[4] I. Bitter, A. E. Kaufman, M. Sato, *Penalized-Distance Volumetric Skeleton Algorithm*, IEEE Trans. Visualization and Comp. Graphics, 7(3), 2001.

[5] G. Borgefors, *On Digital Distance Transforms in Three Dimensions*, Computer Vision and Image Understanding 64(3):368-376, 1996.

[6] G. Borgefors, I. Nyström, G. S. Di Baja, *Computing skeletons in three dimensions*, Pattern Recognition, 32(7), 1999.

[7] S. Bouix, K. Siddiqi, *Divergence-Based Medial Surfaces,* ECCV 1842:603-618, Springer-Verlag, 2000.
[8] D. Brunner, G. Brunnett, *Mesh Segmentation Using the Object Skeleton Graph*, Proc. IASTED International Conf. on Computer Graphics and Imaging, 48-55, ACTA Press 2004.
[9] J. F. Canny, *A Computational Approach to Edge Detection*. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No. 6, pp. 679-698, 1986.
[10] M. Couprie and R. Zrour, *Discrete Bisector Function and Euclidean Skeleton*, Lecture Notes in Computer Science, vol. 3429, Springer-Verlag, 2005.
[11] N. D. Cornea, D. Silver, P. Min, *Curve-Skeleton Applications*. In Proceedings IEEE Visualization, pp. 95-102, 2005.
[12] N. D. Cornea, D. Silver, X. Yuan, R. Balasubramanian, *Computing Hierarchical Curve-Skeletons of 3D Objects*. The Visual Computer 21(11):945-955, Springer-Verlag, 2005.
[13] T. K. Dey and S. Goswami. *Tight Cocone: A water-tight surface reconstructor*. Proc. 8th ACM Sympos. Solid Modeling Applications, 127–134. Journal version in J. of Computing and Infor. Sci. Engin. Vol. 30, 2003, 302–307.
[14] U. Eckhardt, G. Maderlechner, *Invariant Thinning*, Pattern Recognition and Artificial Intellicgence (7):1115-1144, 1993.
[15] N. Gagvani and D. Silver, *Parameter Controlled Volume Thinning*, Graphical Models and Image Processing, 61(3):149-164, 1999.
[16] N. Gagvani and D. Silver, *Animating volumetric models*, Academic Press Professional 63(6):443-458, 2001.
[17] P. Golland, W.E.L. Grimson, *Fixed Topology Skeletons*, IEEE CVPR, 2000.
[18] W. Gong and G. Bertrand, *A simple parallel 3d thinning algorithm*. Proc. IEEE Pattern Recognition, 188-190, 1990.
[19] T. He, L. Hong, D. Chen, Z. Liang, *Reliable Path for Virtual Endoscopy*: Ensuring Complete Examination of Human Organs, IEEE Trans. Visualization and Comp. Graphics, 7(4):333-342, 2001.
[20] M. W. Hirsch, S. Smale, *Differential Equations, Dynamical Systems and Linear Algebra*, Academic Press, 1974.
[21] R. Jain, R. Kasturi, B. G. Schunck. *Machine Vision*. McGraw-Hill,Inc., New York, 1995.
[22] A. Kanitsar, D. Fleischmann, R. Wegenkittl, P. Felkel, E. Gröller, *CPR: Curved Planar Reformation*. Proc. IEEE Visualization, pp. 37-44, 2002.
[23] T. Lee and R.L. Kashyap, *Building skeleton models via 3-d medial surface/axis thinning algorithms*. CVGIP: Graphical Models and Image Processing, 56(6):462-478, November 1994.
[24] S. Lobregt and P. W. Verbeek and F. C. A. Groen, *Three-Dimensional Skeletonization: Principle and Algorithm*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2(1): 75-77, 1980.
[25] C. Lohou and G. Bertrand, *A 3D 12-subiteration thinning algorithm based on P-simple points*, Discrete Applied Mathematics 139:171-195, Elsevier, 2004.
[26] V. Luboz,X. Wu, K. Krissian, C. F. Westin, R. Kikinis, S. Cotin, S. Dawson, *A segmentation and reconstruction technique for 3D vascular structures*, MICCAI 2005, Lecture Notes in Computer Science 3749:43-50, 2005.
[27] C. M. Ma and M. Sonka, *A fully parallel 3d thinning algorithm and its applications*. Computer Vision and Image Understanding, 64(3):420-433, 1996.
[28] G. Malandain, S. Fernandez-Vidal, *Euclidean Skeletons, Image and Vision Computing*, vol. 16:317-327, 1998.

[29] A. Manzanera, T. Bernard, F. Preteux, B. Longuet, *A unified mathematical framework for a compact and fully parallel n-D skeletonization procedure*, Vision Geometry VIII, Vol. 3811: 57–68, SPIE, 1999.
[30] K. Palagyi, A. Kuba, *Directional 3D Thinning using 8 Subiterations,* Proc. Discrete Geometry for Computer Imagery, Lecture Notes in Computer Science 1568:325-336., 1999.
[31] K. Palagyi and A. Kuba. *A parallel 3d 12-subiteration thinning algorithm.* Graphical Models and Image Proc., 61(4):199-221, 1999.
[32] D. Perchet, C. I. Fetita, F. Preteux, *Advanced navigation tools for virtual bronchoscopy*, Proc. SPIE Conf. on Image Processing: Algorithms and Systems III, vol. 5298, 2004.
[33] C. Pudney, *Distance-Ordered Homotopic Thinning: A Skeletonization Algorithm for 3D Digital Images*, Computer Vision and Image Understanding, 72(3):404-413, 1998.
[34] P.K. Saha, B.B. Chaudhuri, D. Dutta Majumder, *A new shape preserving parallel thinning algorithm for 3d digital images.* Pattern Recognition, 30(12):1939-1955, 1997.
[35] H. Schirmacher, M. Zöckler, D. Stalling, H. Hege, *Boundary Surface Shrinking - a Continuous Approach to 3D Center Line Extraction*, Proc. of IMDSP, 25-28, 1998.
[36] J.A. Sethian, *Fast Marching Methods*, SIAM Review, 41(2):199-235, 1999.
[37] H. Si, *TetGen, A Quality Tetrahedral Mesh Generator and Three-Dimensional Delaunay Triangulator*, WIAS Technical Report No. 9, 2004.
[38] H. Sundar, D. Silver, N. Gagvani, S. Dickinson, *Skeleton Based Shape Matching and Retrieval*, Proc. Shape Modeling Int'l, 2003.
[39] K. Suresh, *Automating the CAD/CAE Dimensional Reduction Process*, ACM Symp. On Solid Modeling and Applications, 2003.
[40] S. Svensson, I. Nystrom, G. Sanniti di Baja, *Curve Skeletonization of Surface-like Objects in 3D Images Guided by Voxel Classification*, Pattern Recognition Letters, 23 (12):1419-1426, 2002.
[41] A. Telea, A. Vilanova, *A robust level-set algorithm for centerline extraction*, Eurographics/IEEE Symp. On Data Visualization, pp. 185-194, 2003.
[42] Y. F. Tsao and K. S. Fu, *A parallel thinning algorithm for 3d pictures.* Computer Vision, Graphics and Image Proc., 17:315-331, 1981.
[43] L. Wade, R.E. Parent, *Automated generation of control skeletons for use in animation*, The Visual Computer 18(2):97-110, 2002.
[44] M. Wan, F. Dachille, A. Kaufman, *Distance-Field Based Skeletons for Virtual Navigation*, IEEE Visualization 2001, pp. 239-246, 2001.
[45] T. Wischgoll, Gerik Scheuermann, *Detection and Visualization of Planar Closed Streamlines*, IEEE Transactions on Visualization and Computer Graphics, 7(2): 165-172, 2001.
[46] T. Wischgoll, *Closed Streamlines in Flow Visualization*, Ph.D. Thesis, Universität Kaiserslautern, Germany, 2002.
[47] Z. Yu, C. Bajaj, *A Segmentation-Free Approach for Skeletonization of Gray-Scale Images via Anisotropic Vector Diffusion*, CVPR 2004, pp. 415-420, 2004.
[48] Y. Zhou, A. Kaufman, A. W. Toga, *Three-dimensional Skeleton and Centerline Generation Based on an Approximate Minimum Distance Field*, The Visual Computer, 14, pp. 303-314, 1998.
[49] Y. Zhou, A. W. Toga, *Efficient skeletonization of volumetric objects*, IEEE Trans. Visualization and Comp. Graphics, 5(3):196-209, 1999.

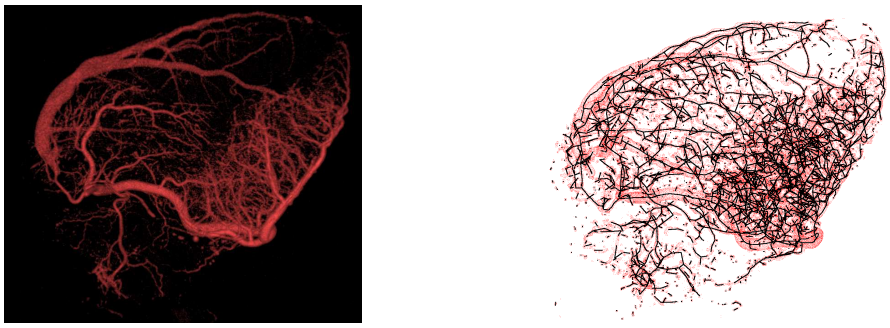


Fig. 1.5. Volume rendering (with shading enabled) of a previously perfused porcine heart which was scanned using a standard hospital CT scanner (left) and curve-skeleton of the porcine heart data set using the described algorithm; to enhance visibility in the paper the thickness of the lines and points within the image is increased (right).

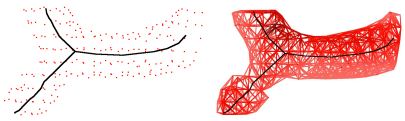


Fig. 1.6. Sub-section of the porcine heart data set showing the extracted center line (left) and the underlying tetrahedrization used for identifying the center line (right).

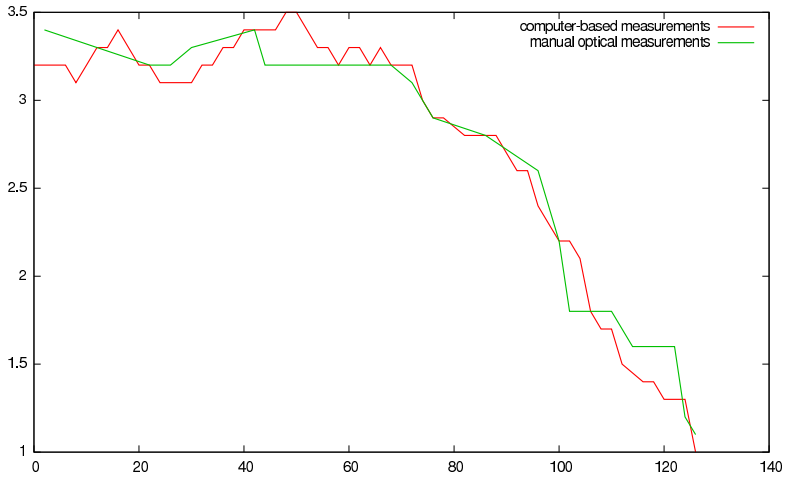


Fig. 1.7. Comparison of computed vessel radii (red) and manual optical measurements (green) for a typical specimen.

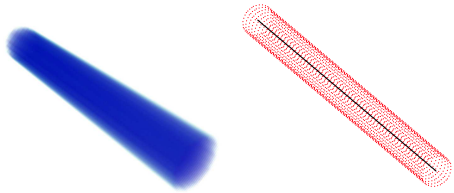


Fig. 1.8. Cylinder data set visualized as a volume rendered image (left) and the extracted curve-skeleton of the same data set (right); to enhance visibility in the paper the thickness of the lines and points within the image is increased.

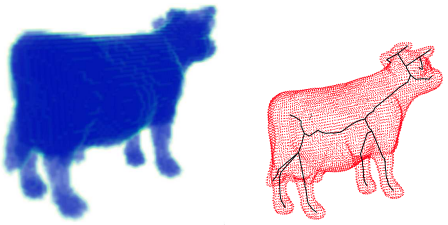


Fig. 1.9. Cow data set visualized as a volume rendered image (left) and the extracted curve-skeleton of the same data set (right); to enhance visibility in the paper the thickness of the lines and points within the image is increased.

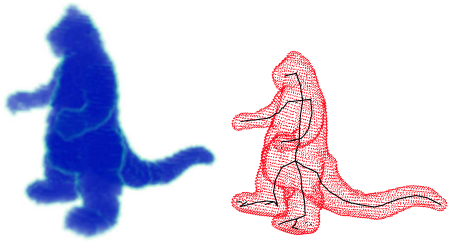


Fig. 1.10. Monster data set visualized as a volume rendered image (left) and the extracted curve-skeleton of the same data set (right); to enhance visibility in the paper the thickness of the lines and points within the image is increased.

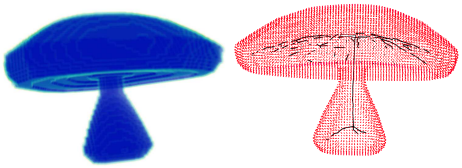


Fig. 1.11. Mushroom data set visualized as a volume rendered image (left) and the extracted curve-skeleton of the same data set (right); to enhance visibility in the paper the thickness of the lines and points within the image is increased.