Extracting Skeletal Curves from 3D Scattered Data

Anne Verroust, Francis Lazarus

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Extracting Skeletal Curves from 3D Scattered Data

Anne Verroust & Francis Lazarus

N´ 3250

septembre 1997

THEME 3

ISSN 0249-6399

INRIA
ROCQUENCOURT

Extracting Skeletal Curves from 3D Scattered Data

Anne Verroust* & Francis Lazarus†

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet SYNTIM

Rapport de recherche n° 3250 — septembre 1997 — 13 pages

Abstract: We introduce a method for the construction of skeletal curves from an unorganized collection of scattered data points lying on a surface. These curves may have a tree like structure to capture branching shapes such as blood vessels. The skeletal curves can be used for different applications ranging from surface reconstruction to object recognition.

As an input, the algorithm takes a set of 3D points. It returns a set of curves arranged in a tree structure. The only interaction needed is the selection of a data point which represent the root of the tree. A neighborhood graph is constructed over the set of points to compute geodesic distances between the root point and the other points. Connected level sets of the distance map are then extracted and organized in a tree structure. The centers of these levels sets constitute the skeletal curves.

Key-words: visualization, skeletal curve, cylindrical decomposition, generalized cylinders, reconstruction

(Résumé : tsvp)

* Anne.Verroust@inria.fr
† I.R.C.O.M.-S.I.C. (U.R.A. C.N.R.S. 356) SP2MI Boulevard 3, Teleport 2, B.P. 179 86960 Futuroscope Cedex, France

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : (33) 01 39 63 55 11 – Télécopie : (33) 01 39 63 53 30

Extraction de courbes squelette à partir de nuages de points

Résumé : Nous proposons une méthode construisant des courbes squelettes à partir de la donnée d'un nuage de points approximant une surface 3D. Ces courbes permettent de construire une approximation cylindrique de la surface de l'objet. Elles peuvent avoir une structure arborescente pour modéliser des embranchements comme dans le cas des vaisseaux sanguins.

A partir d'un ensemble de points 3D, l'algorithme fournit un ensemble de courbes structurées sous forme d'arbre. La sélection préalable d'un des points permet de définir la racine de l'arbre. Un graphe de voisinage construit sur l'ensemble des points est utilisé pour calculer les distances géodésiques entre le point racine et les autres points. Des ensembles de niveaux connexes sur la carte de distance sont extraits et structurés sous forme d'arbre. Les courbes squelettes sont alors construites à partir des centres de ces niveaux.

Mots-clé : courbe squelette, décomposition cylindrique, cylindres généralisés, reconstruction

1 Introduction.

The description of objects with cylindrical shapes has been extensively used in the computer vision community [AB73, SAB81, SB84, OBFG94, BRLS96]. In a first approximation it is often convenient to model a shape as a generalized cylinder. This is particularly relevant in the context of reconstruction of anatomical shapes from CT data or scanner [PWMZ95, WW95, NKK+96]. Numerous works have been done for object description using tree-like structures [SES93, Ebe93, NCS94, Naz96, NCS96] or cylindrical representations [BRLS96]). What makes the use of cylindrical descriptions and generalized cylinders so popular is their relative simplicity. A generalized cylinder is parameterized by a spin curve together with a set of cross sections [AB73]. Depending on the application, the recovery of such parameters usually requires less work than it would be for a more general description of a shape.

Modeling of shapes with implicit surfaces defined by a skeleton has also been exploited in Graphics [BW90, BS91]: the surfaces of the objects are then defined as iso-surfaces of field functions generated by the geometric primitives that form the skeleton. An implicit reconstruction of shapes from a set of 3D scattered points using a geometric skeleton has been proposed in [BTG95, FCGA96]. In their approach, the geometric skeleton is built using the Voronoi graph of the data points.

Most of the studies done in medical imaging start from volume images so that the traditional medial axis transform can be applied [NKK+96]. Pisupati & al. [PWMZ95] use a different approach to compute a central axis inside pulmonary tree structures. Their algorithm uses a recursive region growing to go down the tree volume. A branching region is encountered when the unvisited set of voxels split into two disjoin sets.

As opposed to voxelized data we start from an unorganized cloud of points sampled on a surface and we do not restrict ourselves to medical applications. To use a cylindrical model to represent the surface, a 3D curve inside the cloud of points representing the object must be specified. When the set of 3D points is given as a set of 2D planar scans, the curve defined by the centroids of each 2D section can be used (as in [BRLS96]). However, this implies a consistent orientation of the cross-sectional planes relatively to the position of the object. Figure 1 shows two possible relative orientations. Clearly the first configuration (a) is not adapted. Sometimes, because of the bend shape of the object, it is even impossible to find a valid orientation for the sectional planes, as for the horse of Figure 10.

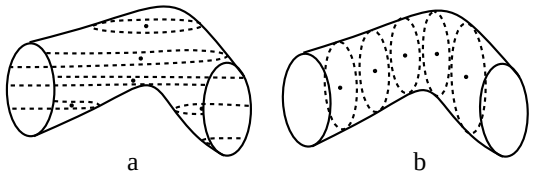

Figure 1: Two sets of 2D sections sampled on a same surface.

Although it is essential to find a suitable axis in order to design a generalized cylinder, this issue was rarely discussed. In practical cases this task can be tedious as mentioned by Nazarian & al. [NCS94, Naz96, NCS96]. Nazarian & al. propose a few techniques to automate the determination of the axis. They mainly present two methods:

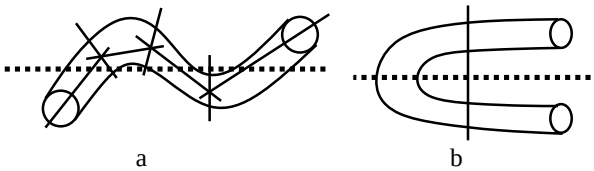

Figure 2: a: recursive subdivision using main axes of inertia (Nazarian & al); the first main axis is shown in dashed line; The main axes of inertia and the perpendicular splitting planes for the second level of recursion are shown in plain lines. b: The first axis of inertia and the perpendicular splitting plane are not suitable for the subdivision algorithm.

- the first one consists in a recursive subdivision of the set of data points. At each step the subsets of data points are split by a plane perpendicular to their main axis of inertia and passing through their barycenter. The resulting axis is a polyline composed of main axes of parts of the initial data set of points

(cf. Figure 2.a). The authors recognize their method does not always work and show a counterexample where the main axis of inertia, represented in dashed line in Figure 2.b, cannot be used to find the central axis.

- the second method works on set of data points given by planar 2D scans and it uses the triangulation computed by Boissonnat's method [Boi88] as an approximation of the tubular shape.

In this paper, we present a method which automatically computes a set of 3D curves in a cluster of points. When the points lie on a surface these curves may be used as axes for a cylindrical approximation of the surface. These curves may form a tree structure if the surface can be seen as a "tree of generalized cylinders". The method works independently of the bending of the surface. As a consequence it will return a correct result for the shape shown on Figure 2.b. In order to obtain satisfying results we make some hypothesis on the input cluster of points:

1. The data points should be sampled on a certain (unknown) surface.
2. The sampling should be done in such a way that the distance between neighbor points is small with respect to the width of the tubular parts of the surface.

The computation of the axes is based on a cylindrical decomposition of the surface representing the set of 3D points. We introduced the cylindrical decomposition of a shape in [Laz95, LV96] to compute axes inside polyhedra. A single axis will be obtained for each cylindrical part associated to this decomposition. As an input parameter, the cylindrical decomposition takes a single point called the *source point*. This source point may be interactively designated by the user. We are also currently working on a default selection based on the diameter of a shape [CEGS93]. The choice of the source point is critical since it will determine the cylindrical decomposition. Hopefully, as was pointed out in [LV96] the decomposition is not very sensitive to the precise location of the source point. Intuitively, the source point correspond to a polar extremity of the shape as shown on Figure 3.

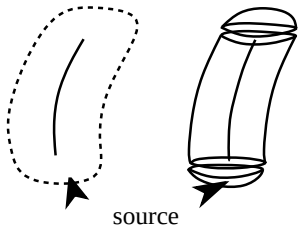

Figure 3: The source point and the corresponding cylindrical decomposition.

2 Overview

The computation of the 3D curves is made in four steps, as illustrated in Figure 4. Given a set of data points,

1. We build the *neighborhood graph* of order n. This is a non oriented graph whose vertices are the input data points. The edges have the form (v,v'), where v' is one of the n nearest neighbors of the vertex v or v is one of the n nearest neighbors of the vertex v'. In general, it will not be possible to determine a suitable polygonal surface from this graph as it may be non planar.
2. From this neighborhood graph and a source point, we construct a distance map and an oriented subgraph called the *geodesic graph*. The geodesic graph is obtained by computing shortest paths of the neighborhood graph between all vertices and the source point. The distance map is simply the length of these shortest paths.

   The construction of the neighborhood graph, the distance map and the geodesic graph are described in details in section 3.

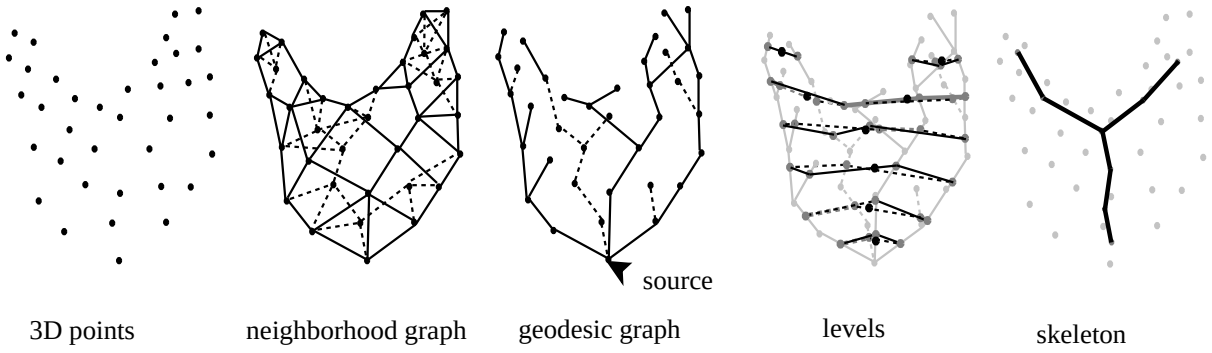

Figure 4: The main steps of our method.

3. We further compute k level sets of the distance map using the geodesic graph, where k is a user defined value. The level sets are composed of points at a constant distance from the source. We connect these points on a proximity criterion and identify the connected components. For each level, we also compute the centroids of every of its connected components.

4. The axes are finally obtained by connecting the centroids of successive levels as previously computed. When the levels contains several connected components, the connection of the centroids are based upon the paths of the geodesic graph.

   Steps 3 and 4 are discussed in section 4 and examples are provided in section 5.

The computation of the neighborhood graph is the most expensive step. Since the neighborhood graph does not depend on the choice of the source point, the user can select different source points, and the corresponding axes computation will only involve step 2, 3 and 4. Similarly, as the number k of discretization points is used only in step 3 and 4, different values of k may be tried without actually recomputing steps 1 and 2.

3 The Geodesic Graph

We are given a set of data points $X = \{x_1, ..., x_m\}$ sampled on an unknown surface $\mathcal{S}$ and an integer n. The neighborhood graph $\mathcal{N}_{(X,n)}$ is an undirected graph defined as follow:

- the vertices of $\mathcal{N}_{(X,n)}$ are the points of X
- a pair (x_i, x_j) is an edge of $\mathcal{N}_{(X,n)}$ if either x_i is one of the n nearest points of x_j in X or x_j is one of the n nearest points of x_i in X.

In order to compute the n nearest points of each vertex, we use a subdivision of the space into voxels. We subdivide the enclosing box of the data points in $N \times N \times N$ voxels[1] and associate each data point with the voxel containing it. The search of the n nearest neighbors of a data point x is made by visiting the voxels close to the one containing x and moving away progressively until n points $(v_1,...v_n)$ are found. Let $d = \max_{1 \leq i \leq n}(\text{distance } (v_i, x))$. As we have already found n points $(v_1,...v_n)$ located at a distance lower or equal to d from x, we are sure that the n nearest neighbors of x belong to the sphere $S(x, d)$ centered in x and of radius d. Then we search all the data points belonging to $S(x, d)$ and we extract the n nearest points on this set, using a quick sort on the distance values.
The number n should be chosen such that $\mathcal{N}_{(X,n)}$ is a connected graph. This choice depends on the regularity of the sampling of the data points on the surface. In the examples presented in Figure 10 we used $n = 5$. When the data points are not uniformly spread over the surface larger values may be necessary. For instance, in the case of 2D planar scans with a smaller resolution inside the scan planes than between the cross-sections we found that a value of n as large as 20 was necessary to obtain a connected graph. This was the case for the object shown in Figure 15.

[1] The number N depends on the number of input data points.

Even though $\mathcal{N}_{(X,n)}$ may be non planar we expect the neighborhood graph to be a fair approximation of the unknown surface. That is the edges of $\mathcal{N}_{(X,n)}$ should lie on the surface.

Together with the selected source point x_s, the neighborhood graph is used to compute the geodesic graph $\mathcal{G}_{(X,n,x_s)}$. The geodesic graph is composed of geodesic paths of the neighborhood graph emanating from the data points and joining the source point. The length of an edge of the neighborhood graph is taken as the Euclidean distance between its endpoints. We use the Dijkstra algorithm [CLR90] to efficiently compute these paths and their length. Precisely, the Dijkstra algorithm runs in $\mathcal{O}(n \log n)$ and compute the geodesic distance from any point to the source as well as a pointer to the next point in a geodesic path joining the source. Note that there may exist several geodesic path from a point to the source. We simply use the path returned by the Dijkstra algorithm. The distance map introduced in section 2 is defined for every data point to be its geodesic distance to the source. It is extended to the edges of the geodesic graph with a simple linear interpolation of the endpoint distance values of the edges.

As we supposed that the edge of the neighborhood graph should lie on the surface, the geodesic paths previously computed are intended to approximate geodesic paths on the surface. The computation of geodesic paths for polyhedral surfaces have been the subject of numerous publications [MMP87, OP94, Ped95]. The algorithm described in [MMP87] computes exact geodesic distances but is also expensive. To obtain a faster algorithm, [Laz95, LV96] have chosen to compute an approximation of the geodesic distance on a polyhedron, considering only the graph composed of its vertices and its edges. As noticed in the previous section, the neighborhood graph is not a polyhedral approximation of the surface described by the set of data points, so that we can only compute approximations of the geodesic distances.

Note that $\mathcal{G}_{(X,n,x_s)}$ is a connected subgraph of $\mathcal{N}_{(X,n)}$ since all the geodesic paths share the source point. An example of a geodesic graph is shown Figure 5 for a set of data points registered on a femur[2].

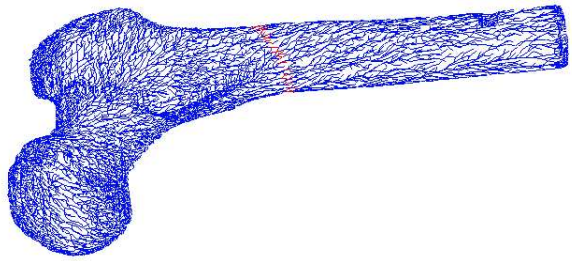

Figure 5: The geodesic graph of a femur. Edges marked in red are located at mid-distance of the source point and its most distant point.

4 Levels and branches

After computing the geodesic graph and the distance map, we extract k level sets. The number k is given by the user to specify the number of points used to approximate the longest axis. The choice of k must be made considering the number and the locations of the branches and of the smoothness of the skeletal curves to be computed. Thus several trials can be necessary to obtain the best value of k. The k levels $d_1,\ldots, d_k$ are uniformly distributed over the interval $[\alpha d_{max}, \beta d_{max}]$, where d_{max} is the extremum of the distance map and α and β are two percentage coefficients. We used $\alpha = 3\%$ and $\beta = 97\%$ on Figure 15. The ith level set $\mathcal{S}_i$ is the set of points on the geodesic graph[3] whose (extended) distance is d_i.

Next, we need to partition each level set $\mathcal{S}_i$ into subsets corresponding to the different branches of the surface. To do so, we first construct the neighborhood graph $\mathcal{N}_{(\mathcal{S}_i,2)}$, linking every point of $\mathcal{S}_i$ with its two closest neighbors. Then, we remove from $\mathcal{N}_{(\mathcal{S}_i,2)}$ the edges longer than a certain length w_i. w_i is determined using the hypothesis (2) on the data points: the distance between points is small with respect to the width of the tubular parts. In practice w_i is computed by using the median of the distribution of the distance values of the second nearest neighbor of each point inside $\mathcal{S}_i$. Finally, we partition this graph into its connected components.

[2]data, courtesy of Jai Menon, IBM T. J. Watson Research Center.

[3]We have chosen to use the geodesic graph instead of the neighboring graph because we need to have the geodesic paths when branches are detected.

This can be done in time nearly linear using the union-find algorithm [CLR90]. We also compute the centroid of each connected component by taking the barycenter of its points.

The skeletal curves are constructed by connecting the centroids of successive level sets. When the level sets have several connected components we use the geodesic paths to connect the centroids: if a level set $\mathcal{S}_i$ is partitioned into l_i connected components ($\mathcal{S}c_{i,1}, ..\mathcal{S}c_{i,l_i}$, for each $j \leq l_i$, we select a vertex of the geodesic graph whose edge contains a point of $\mathcal{S}c_{i,j}$ and we follow its geodesic path to the source point until we encounter an edge containing a point of $\mathcal{S}_{i-1}$. This point belongs to the connected component of $\mathcal{S}_{i-1}$ which is either associated to the branch containing $\mathcal{S}c_{i,j}$ or located at the birth of this branch.

In the first case, the corresponding centroids are connected.

In the second case, the branching node is backed up one level, as illustrated in Figure 6: if the connected component is the location of the birth of l new branches, it is decomposed into l new components using the geodesic graph and the l connected components of $\mathcal{S}_i$ corresponding to these branches. l new centroids are computed at level $i-1$ and are connected to the centroids of the corresponding connected components of level i. With this process, we ensure the fact that the skeletal curves stay inside the object whose surface is approximated by the 3D data points in most of the cases.

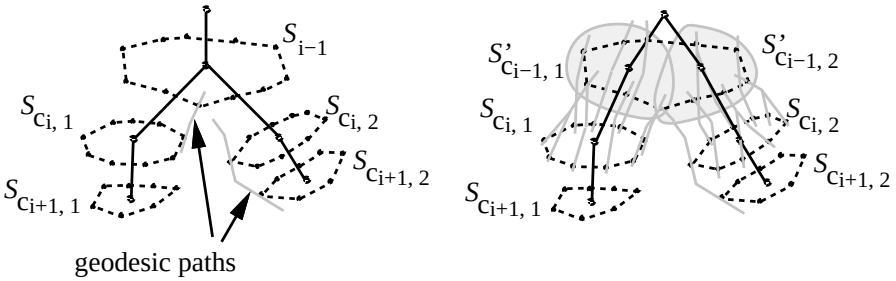

Figure 6: Left: $\mathcal{S}_i$ is located at the birth of a branch. Right: $\mathcal{S}_i$ is decomposed into two components: $\mathcal{S}'c_{i,1}$ and $\mathcal{S}'c_{i,2}$ and the corresponding centroids are connected with the two centroids of level $i+1$.

Eventually we may approximate the axes with spline curves as shown for the femur on Figure 7 and 8[4]. We use a simple fitting procedure from [Sch90, Eil94] to compute these splines from the connected centroids.

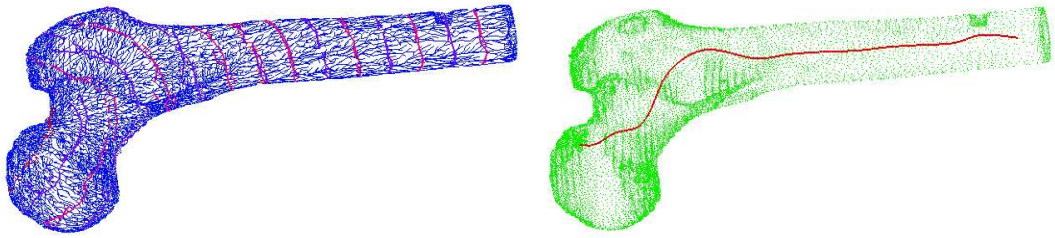

Figure 7: Left: 15 level sets computed on the geodesic graph of a femur. Right: the corresponding axis.

Each axis of the tree structure correspond to a tubular part of the surface. The union of these tubular parts form a decomposition of the the surface. We first introduced this decomposition for polyhedral surfaces in a previous work [Laz95, LV96]. In this work a simplified discrete version of the Morse theory is developed (cf. [Ker92, KCM94] for use of Morse Theory in visualization). The geodesic distance to a source vertex on a polyhedron is computed, and a vertex is classified as regular or critical according to the configuration of distances of its neighbors (see Figure 9). The cylindrical decomposition is obtained by cutting through the level sets associated to the critical points. The level sets of the geodesic distance are efficiently extracted using the adjacency between faces of the polyhedron. In our case this adjacency relationship is not available. As a result points cannot be classified as regular or critical which would in turn provide the branching points. The only

[4]One can notice that, in this case, the axis curvature is very low and 15 levels are sufficient to obtain the skeletal curve: the three axes of Figure 8 are very close and the smoother one presented in Figure 7 is the best one.

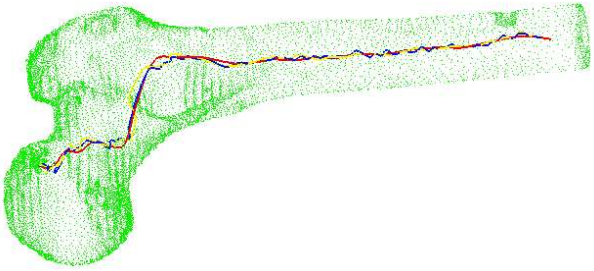

Figure 8: Different number of levels used to compute the axis of the femur : 30 (red curve), 60 (yellow curve) and 90 (blue curve).

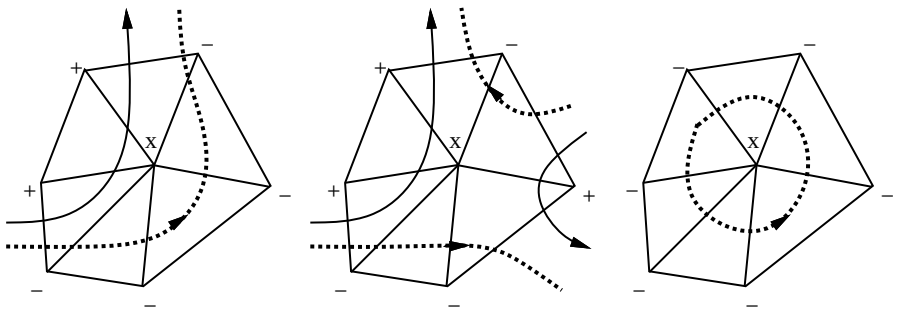

Figure 9: the leftmost vertex is regular and the other are critical (the + (respectively −) associated to a vertex indicates that the geodesic distance between the vertex and the source is greater (respectively lower) than the geodesic distance between the source and x).

topological information available are the neighborhood graph and the geodesic graph. In order to detect critical points we could introduce a local ordering around a vertex with an approximate tangent plane associated as in [HDD+92]. In this case we would have to enforce a global coherence of this local ordering on the whole surface. This can be as expensive as enforcing the global consistency of the tangent plane orientations as done in [HDD+92]. Moreover this would not be very reliable since neither the neighborhood graph nor the geodesic graph are planar. Also, since our purpose is to built skeletal curves and not to reconstruct the surface it is reasonable to provide a procedure less expensive than a reconstruction method.

5 Results

We applied our skeletonization method on a set of examples presented from Figure 10 to Figure 15:

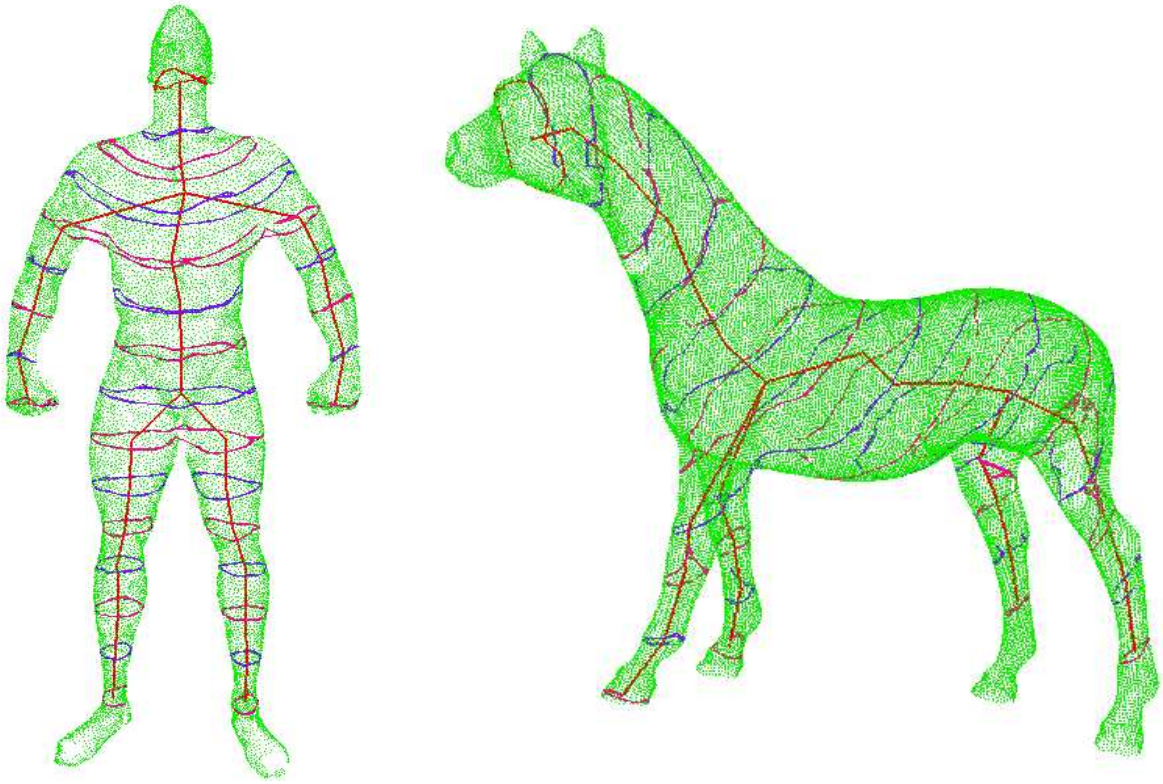

Figure 10: 15 level sets and the corresponding skeletal curves on a man and on a horse.

- Figure Figure 10 shows two models of a man and a horse. These models are public domain models found on the Web pages of Cyberware[5]:
  - The man has the source point located at the top of his head.
  - The horse was scanned from many orientations resulting in 210 linear scans which consisted of a series of scans of different regions (views) of the object. The source point is at the extremity of its jaw.

  In these two cases, the branches are correctly located and a tree-like axial structure is built.
- The shoe presented in Figure 11 and 12 is issued from a synthetic model. The points describing the shoe are structured in 2D plans. One can notice that the levels obtained with the source point located at the front of the shoe are nearly planar. Our method is little sensitive to the location of the source points: in Figure 12, two different points located in front of the the shoe have been used to compute the axial

[5] http://www.cyberware.com/models/modelIndex.html

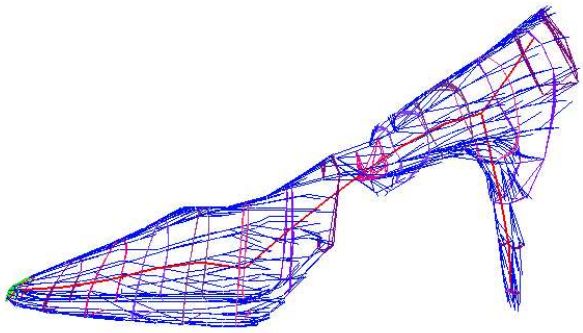

Figure 11: 15 level sets and the corresponding skeletal curves on the geodesic graph of a shoe. The edges adjacent to the source point are drawn in green.

structures. The corresponding axial structures are drawn in red and yellow. One can see that they are nearly the same.

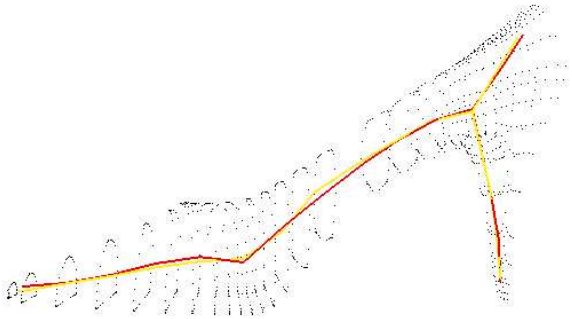

Figure 12: Two skeletal curves associated to two different source points.

- The pelvis bone the blood vessels shown respectively in Figure 13 and Figure 14 are described by a set of 2D scans. These data come from the database of Gill Barequet of the Department of Computer Science of Tel Aviv University [6].

  For the blood vessels, the main ramifications are detected, but when the vessels are too close, as illustrated in the right part of Figure 14, the data points belonging to distinct vessels are connected in the neighborhood graph and in the geodesic graph which leads to an incorrect result for the skeletal curves in this area.

- The axis of the synthetic helicoidal structure presented in Figure 15 cannot be found using Nazarian's subdivision method. Here, as the data points are regularly distributed inside each planar scan, the curve is very smooth.

Table 1 shows the CPU time required to compute the axial structure for the models shown Figures 7 to 15. The computing time depends mainly on the number of data points and on the order of neighborhood graph. In fact, as noticed previously, the computation of the neighborhood graph is one of the most expensive part of our method.

[6]Web site : ftp://ftp.math.tau.ac.il/pub/barequet/psdb/data/

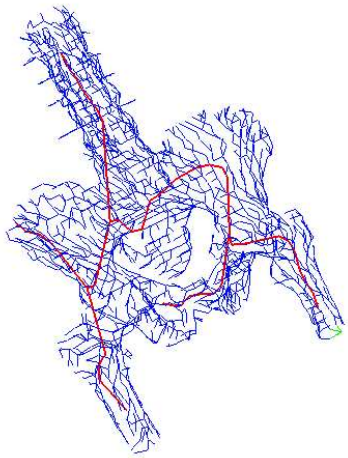

Figure 13: A pelvis bone (the geodesic graph and the skeletal curves associated).

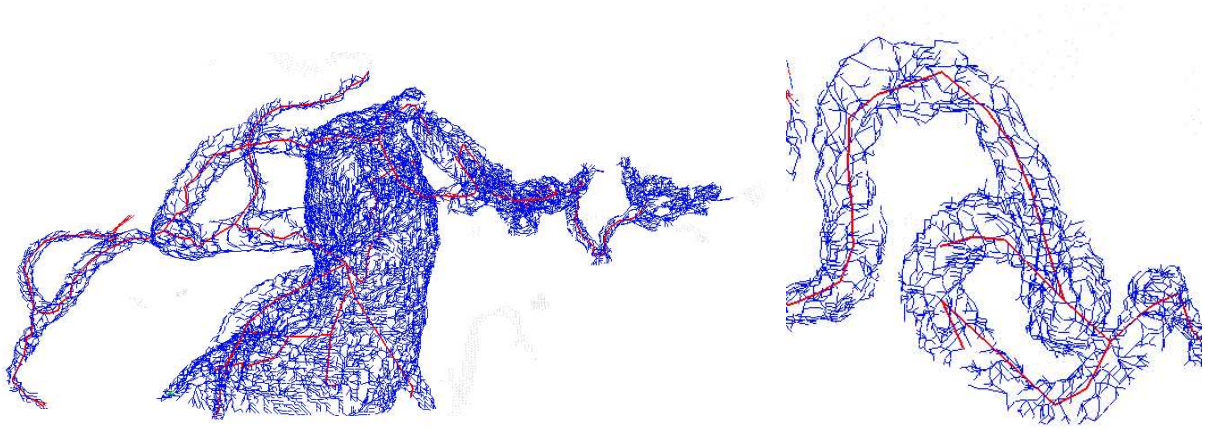

Figure 14: Blood vessels (the geodesic graph and the skeletal curves associated).

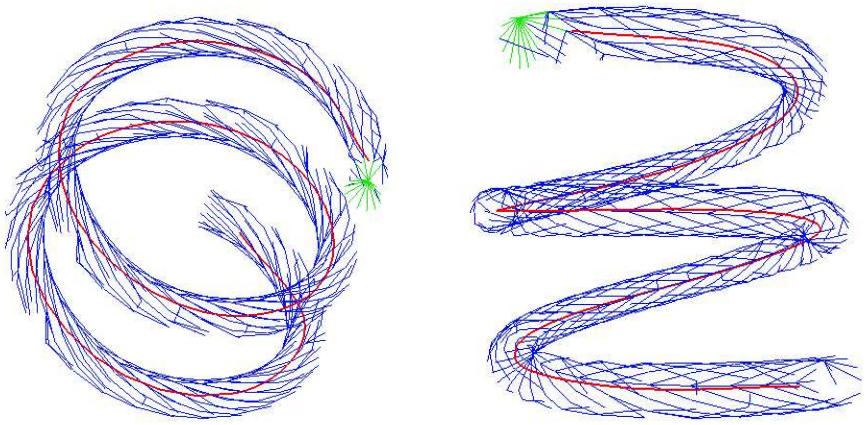

Figure 15: the geodesic graph and the skeletal curve associated.

reference of figures	number of data points	number of levels	order of the neighborhood graph	CPU time in seconds on an Indy
7	16781	15	5	27.83
8	16781	30	5	33.20
8	16781	60	5	39.50
8	16781	90	5	44.83
10 (man)	21500	15	5	52.21
10 (horse)	48486	15	6	132.61
11	1056	15	20	30.18
13	2278	30	4	4.72
14	21985	100	4	48.68
15	769	15	15	3.51

Table 1: Statistics for Figures 7 to 15.

6 Conclusion

We have presented an original technique to associate an axial structure to a set of scattered data points. The interaction is reduced to the selection of a source point. The main steps for the construction of the axes are:

– 1 – the computation of the neighborhood graph and of the geodesic graph, using Dijkstra's algorithm,

– 2 – the computation of geodesic levels and the location of branching nodes.

Our method is rapid and robust with respect to the location of the source point but the "quality" of the resulting skeletal curves depends on the appropriate choice of the order of the neighborhood graph.

Work in progress include attempts to model more precisely the location of the branching nodes and an automatic selection of a source point using the diameter of the neighborhood graph. A more robust construction of the neighborhood graph is also under consideration. An adaptive order for the neighborhood graph based on the local density of data points could probably help.

Acknowledgments

We wish to thank Marcel Bosc for his implementation of the computation of the n nearest neighbors of a point in a set of 3D data points.

References

[AB73] G.I. Agin and T.O. Binford. Computer description of curved objects. In *Third International Conferemce on Artificial Intelligence*, 1973.

[Boi88] J.-D. Boissonnat. Shape reconstruction from planar cross sections. *Computer Vision, Graphics and Image Processing*, 44, 1988.

[BRLS96] V. Burdin, C. Roux, C. Lefèvre, and E. Stindel. Modeling and analysis of 3-D elongated shapes with application to long bone morphometry. *IEEE Transactions on Medical Imaging*, 15(1):79–91, February 1996.

[BS91] J. Bloomenthal and K. Shoemake. Convolution surfaces. In *Computer Graphics (SIGGRAPH '91)*, volume 25, pages 251–256, July 1991.

[BTG95] E. Bittar, N. Tsingos, and M.P. Gascuel. Automatic reconstruction of unstructured 3D data: Combining a medial axis and implicit surfaces. In *Eurographics'95*, pages 457–468, September 1995.

[BW90] J. Bloomenthal and B. Wyvill. Interactive techniques for implicit modeling. *Computer Graphics (1990 Symposium on Interactive 3D Graphics)*, 24(2):109–116, March 1990.

[CEGS93] B. Chazelle, H. Edelsbrunner, L.J. Guibas, and M. Sharir. Diameter, width, closest line pair, and parametric searching. *Discrete & Computational Geometry*, 10, 1993.

[CLR90] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. McGraw-Hill, New York, NY, 1990.

[Ebe93] R. Ebel. Reconstruction interactive d'éléments anatomiques à l'aide de surfaces de forme libre. Thèse de doctorat en sciences, Ecole Nationale Supérieure des Télécommunications, Paris, January 1993.

[Eil94] P.H. Eilers. Smoothing and interpolation with finite differences. In P. Heckbert, editor, *Graphics Gems IV*, pages 241–250. Academic Press, Boston, 1994.

[FCGA96] E. Ferley, M-P. Cani-Gascuel, and D. Attali. Skeletal reconstruction of branching shapes. In *Implicit Surfaces'96*, pages 127–142, Eindhoven, October 1996. Eurographics.

[HDD+92] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Computer Graphics (SIGGRAPH '92)*, volume 26, pages 71–78, July 1992.

[KCM94] D.B. Karron, J. Cox, and B. Mishra. New findings from the SpiderWeb algorithm: Towards a digital Morse theory. In R.A. Robb, editor, *Visualization in Biomedical Computing '94*, pages 643–657, Rochester, October 1994. SPIE Proceedings.

[Ker92] Y.L. Kergosien. Topology and visualization: from generic singularities to combinatorial shape modelling. In Y. Shinagawa T.L. Kunii, editor, *International workshop on modern geometric computing for visualization*, Tokyo, Japan, 1992. Springer.

[Laz95] F. Lazarus. Courbes, cylindres et métamorphoses pour l'image de synthèse. Thèse de doctorat en sciences, Université PARIS VII, December 1995.

[LV96] F. Lazarus and A. Verroust. Décomposition cylindrique de polyèdre et courbe squelette. *Revue Internationale de CFAO et d'Infographie*, 11(4), November 1996.

[MMP87] J.S.B. Mitchell, D.M. Mount, and C.H. Papadimitriou. The discrete geodesic problem. *SIAM J. Comput.*, 16:647–668, 1987.

[Naz96] B. Nazarian. Reconstruction automatique de structures anatomiques tubulaires à l'aide de surfaces de forme libre. Thèse de doctorat en sciences, Faculté des Sciences de Luminy, Marseille, France, September 1996.

[NCS94] B. Nazarian, C. Chédot, and J. Sequeira. Interactivity and Delaynay triangulation for the reconstruction of tubular anatomical structures. In *16th annual conference of IEEE/Engineering in Medecine and Biology Society*, November 1994.

[NCS96] B. Nazarian, C. Chédot, and J. Sequeira. Automatic reconstruction of irregular tubular structures using generalized cylinders. *MICAD'96 - Revue Internationale de CFAO et d'Infographie*, 11(1-2), 1996.

[NKK+96] M. Näf, O. Kübler, R. Kikinis, M.E. Shenton, and E. Székely. Characterisation of 3D organ shapes in medical image analysis using skeletonization. In *Workshop on Mathematical Methods on Biomedical Image Analysis*, pages 139–150. IEEE PAMI, June 1996.

[OBFG94] T. O'Donnell, T.E. Boult, X.S. Fang, and A. Gupta. The extruded generalized cylinder: A deformable model for object recovery. In *Computer Vision and Pattern Recognition*, pages 174–181, 1994.

[OP94] K. Opitz and H. Pottmann. Computing shortest paths on polyhedra: applications in geometric modeling and scientific visualization. *International Journal of Computational Geometry and Applications*, 4(2):165–178, 1994.

[Ped95] H.K. Pedersen. Decorating implicit surfaces. In *Computer Graphics (SIGGRAPH '95)*, pages 291–300, August 1995.

[PWMZ95] C. Pisupati, L. Wolf, W. Mitzner, and E. Zerhouni. A central axis algorithm for 3D bronchial tree structure. In *International Symposium on Computer Vision*, pages 259–264. IEEE PAMI, November 1995.

[SAB81] B. Soroka, R. Andersson, and R. Bajcsy. Generalized cylinders from local aggregation of sections. *Pattern Recognition*, 13(5), 1981.

[SB84] U. Shani and D. Ballard. Splines as embeddings for generalized cylinders. *Computer Vision, Graphics and Image Processing*, 27(2), August 1984.

[Sch90] P.J. Schneider. An algorithm for automatically fitting digitized curves. In Andrew S. Glassner, editor, *Graphics Gems*. Academic Press, Boston, 1990.

[SES93] J. Sequeira, R. Ebel, and F. Schmitt. Three-dimensional modeling of tree-like anatomical structures. *Computerized Medical Imaging and Graphics*, 17(4), November 1993.

[WW95] J. P. Williams and L. B. Wolff. Form from function: a vector field based approach to the analysis of ct images of the vascular tree. In *Workshop on Physic-based Modeling in Computer Vision*, pages 70–77, Cambridge, MA, June 1995. IEEE Computer Society Press.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399