ISSN 1392–124X (print), ISSN 2335–884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2016, T. 45, Nr. 2

# Automatic 3D Mesh-Based Centerline Extraction from a Tubular Geometry Form

**Bahia Yahya-Zoubir, Latifa Hamami, Llies Saadaoui**

*Ecole Nationale Polytechnique, Electronics department, Signal et Communications laboratory*
*10 Avenue Hassen Badi BP 182, El-Harrach 16200, Algiers, Algeria,*
*e-mail: byahyazoubir@gmail.com, latifa.hamami@enp.edu.dz, llies.saadaoui@enp.edu.dz*

**Rafik Ouared**

*University of Geneva, CUI, Informatics department, Battelle, Building A,*
*7 route de Drize, CH-1227 Carouge, Switzerland,*
*e-mail: Rafik.Ouared@unige.ch*

crossref http://dx.doi.org/10.5755/j01.itc.45.2.12162

**Abstract**. This paper presents a new automatic 3D mesh-based centerline extraction (3D-MCE) algorithm, which allows an accurate extraction of 3D centerline from a tubular geometry form, without manned intervention. The 3D-MCE does not require any input parameters and works on polygon mesh vertices producing a thin, connected and centered centerline, without needing pre or post-processing stages. In order to extract the centerlines by 3D-MCE algorithm, several regular and irregular 3D surface meshes of medical anatomy models, generated either from real CT colonoscopy datasets or numerically, have been processed. The validation of the 3D-MCE has been done on models of generic helical tubular geometry forms, where several parameters have been varied: curvature, thickness of tubes and density of surface nodes. Results show that 3D-MCE algorithm statistically extracts accurately and efficiently the centerline of the tubular geometry structures, independently from the density of surface nodes, curvature and thickness. The precision of 3D-MCE algorithm is confirmed by a comparison with the well-known Voronoi diagram centerline extraction method.

**Keywords**: 3D; Centerline; Medial axis; Polygon Mesh; Skeleton; Tubular geometry; Vertices

## 1. Introduction

Nowadays, the visualization and analysis of 3D models are becoming more widespread. 3D technology is rapidly emerging as time progresses. As a result, the need to develop new processing algorithms is increasing. Due to the large amount of data contained in 3D objects, their processing becomes difficult and the need to work with smaller sets with the same total amount of data becomes crucial. Centerline extraction algorithms deliver solutions that can handle this challenge. The centerline, also known as medial axis transform, is defined by Blum [1] as the center of maximal disks in 2D or balls in 3D contained in the shape. It provides along with compact description, significant geometrical and topological information of the 3D objects. The centerline is used in a large number of applications such as path planning for 3D navigation [2, 3], mesh generation [4, 5], animation [6, 7], data compression [8, 9], pattern recognition [10, 11], and images segmentation [12, 13], among others.

Several algorithms have been proposed to compute the centerline; most of them can be classified into four categories: topological thinning, distance transform, shortest path, and Voronoi diagrams.

The topological thinning is defined as iteratively boundary peeling (layer by layer). This procedure identifies and removes the simple points [14] that satisfy specific constraints, sequentially [15, 16] or in parallel [17-19], and then converts them into background points. A thinning procedure does not stop until no further simple point can be found. The result of thinning is a centerline that preserves the topology of the object, but it is not necessarily centered because of the sensitivity of the method to boundary complexity. In addition, such a method is computation time greedy.

The distance transform is applied to the whole geometry volume to assess distance maps [20-23]. The centerline is extracted from the local maxima of the distance map. However, this method needs further post-processing to produce the centerline because it doesn't generate a thin centerline directly and doesn't preserve the topology of the shape.

The shortest path is based on tracing back the centerline from the minimal path between the start and end points which are initially defined by the user [24, 25]. While the advantage of this procedure is its low time complexity, its drawback related to its inaccuracy as short paths are sought essentially.

The Voronoi diagrams imply that the internal edges and faces are extracted and then pruned according to some angles and length criteria [26-30]. Nevertheless, the objects with irregular boundaries are likely to have dense Voronoi diagrams, and hence, significant pruning of the medial surface is needed to produce a thin centerline. Added it is hard to implement.

All of the techniques described above need input parameters to extract the centerline, and don't usually produce a one voxel thick, connected and smooth centerline directly. Most of them suffer from thickness, centrality of centerline points and sensitivity to boundary complexity problems. Consequently, they further need post-processing stages to refine the centerline. In this paper, a new automatic 3D mesh-based centerline extraction (3D-MCE) algorithm for 3D non-branching tubular geometry forms is presented. This novel method is simple to implement and applies directly on the mesh domain. Centerline coordinates are measured from mesh connectivity analysis without any pre- or post-processing extra stages and this gives directly a one voxel thick centerline. This algorithm is not sensitive to boundary noise and does not require prior knowledge of the object shape. It is fully automatic. In order to evaluate the 3D-MCE algorithm, a comparison with the well-known and widely used Voronoi diagram algorithm is performed.

The proposed algorithm is explained in details in Section 2 and its results are analyzed in Section 3. Concluding remarks are given in Section 4.

## 2. Methodology

To extract the centerline, the 3D-MCE algorithm exploits triangle mesh connectivity [31]. In 3D modeling, polygon mesh is a group of vertices, edges and faces that defines the object [32, 33]. The most common representation of 3D surfaces is triangle mesh, which comprises a type of polygon mesh where faces are triangles connected by their common edges or vertices (refer to Figure 1).

The vertices are coordinates in a three dimensional Euclidean space of real numbers $R^3$. Edges are connections between vertices, every edge connects two vertices. Triangular face is a closed set of three edges. A group of $n$ vertices $V$ with theirs corresponding $k$ edges $E$ characterize a 3D mesh model $M = \{V, E\}$:

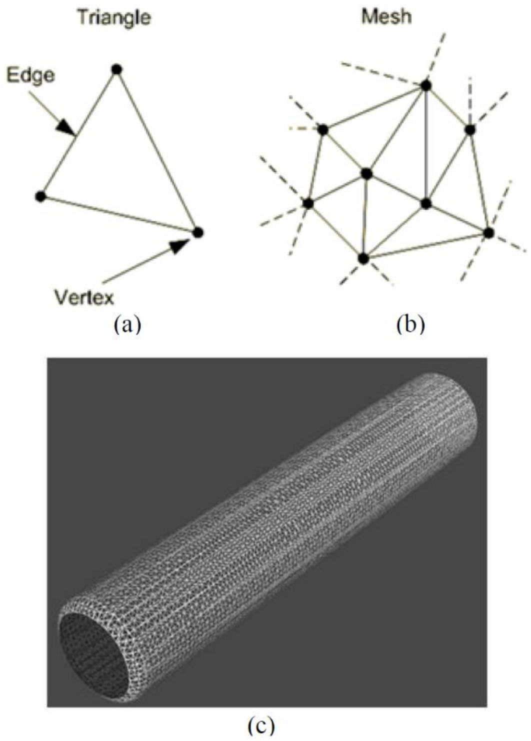

**Figure 1.** (a) Triangle face, (b) Triangle mesh, (c) Triangle mesh of 3D tubular model

$$V = \{(v_i^x, v_i^y, v_i^z) \in R^3, 1 \le i \le n\} \quad (1)$$

$$E = \{e_{ij}^q, 1 \le q \le k\} \quad (2)$$

where $e_{ij}^q$ denotes the $q^{th}$ edge that join the $i^{th}$ vertex with the $j^{th}$ one.

Using the triangle mesh $M$, 3D-MCE algorithm involves several steps to compute the centerline. The process operations are represented in the flowchart illustrated by Figure 2.

### 2.1. External Vertices Extraction

The extraction of the external vertices is an essential step in 3D-MCE algorithm. It allows the location of the external vertices from the input mesh $M$ by the exploration of edges connectivity.

Let $M$ be a triangle mesh $M = \{V, E\}$, where $V$ is the set of all mesh vertices and $E$ is the list of all mesh edges. The edge $e_{ij} \in E$, that connects the vertex $v_i$ with the vertex $v_j$, can be classified into two categories according to its location:

- Edges that bound two triangles are called internal edges. e.g. the triangles $\Delta_{(v_1,v_2,v_3)}$ and $\Delta_{(v_1,v_3,v_4)}$ of the triangle mesh represented in Figure 3 share the internal edge $e_{13}$; the triangles $\Delta_{(v_1,v_5,v_6)}$ and $\Delta_{(v_1,v_6,v_7)}$ share the internal edge $e_{16}$.
- Edges that bound exactly one triangle are called external edges. e.g. the external edge $e_{82}$ of the triangle mesh represented in Figure 3 belongs only

to the triangle $\Delta_{(v_1,v_8,v_2)}$, and the external edge $e_{56}$ belongs only to the triangle $\Delta_{(v_1,v_5,v_6)}$.

Using the two definitions given above, the set of all mesh $M$ edges $E$ can be divided into two subsets: the internal edges subset $E_{int}$ and the external edges subset $E_{ext}$:

$$E = \{E_{int} \cup E_{ext}\}. \qquad (3)$$

The external edge $e_{ij} \in E_{ext}$ of the mesh $M$ connects the external vertex $v_i$ with the external vertex $v_j$. Every external edge connects two external vertices. The external vertices extraction steps of 3D-MCE algorithm consist of looking for the external vertices $V_{ext}$ of the mesh $M$, that connect all external edges $E_{ext}$ (refer to Figure 4).

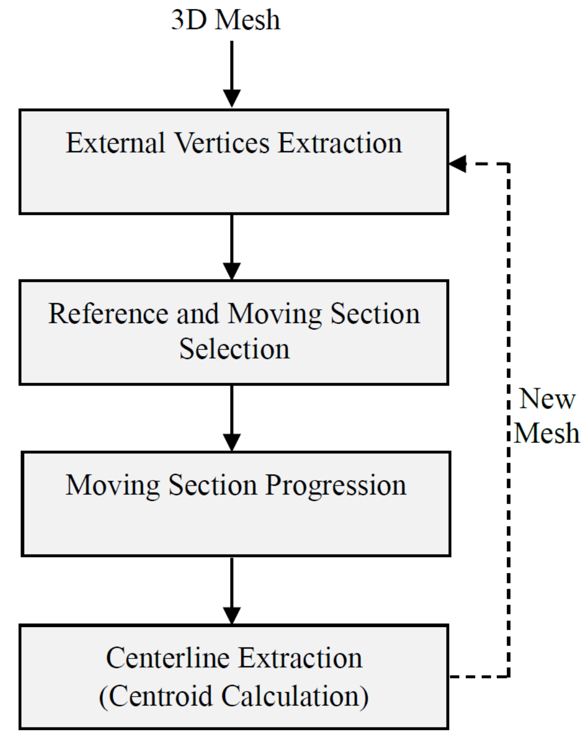
3D Mesh
External Vertices Extraction
Reference and Moving Section Selection
Moving Section Progression
Centerline Extraction (Centroid Calculation)
New Mesh

**Figure 2.** 3D-MCE flowchart

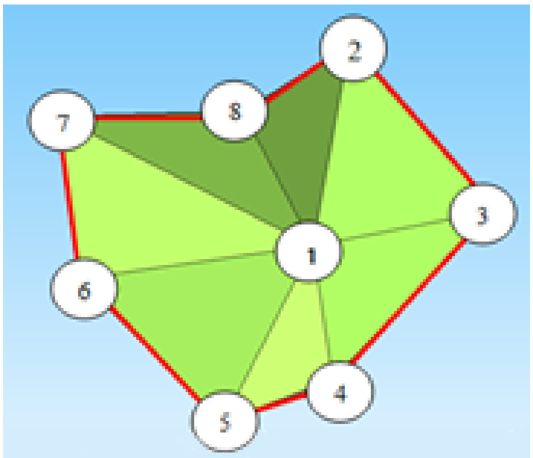

**Figure 3.** Triangle mesh (green color) and its external edges (red color)

## 2.2. Reference and Moving Section Selection

The aim of this step is to divide the external vertices $V_{ext}$ extracted in the previous step, into two subsections. The purpose is to make a distinction between the inlet and the outlet of the 3D tubular geometry form. To do so, the external vertices $V_{ext}$ are partitioned into two exclusive clusters using the Agglomerative Hierarchical clustering method [34-38]. The agglomerative hierarchical clustering method starts with each external vertex in its own cluster and proceeds progressively by joining the closest clusters until two different clusters are formed. The first cluster represents the reference section vertices $V_{ref}$, and the second one represents the moving section vertices $V_{mov}$ (refer to Figure 5):

$$V_{ext} = \{V_{mov} \cup V_{ref}\}. \qquad (4)$$

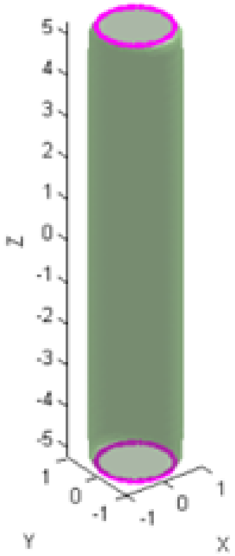

**Figure 4.** External vertices (magenta color) of a 3D tubular geometry form (green color)

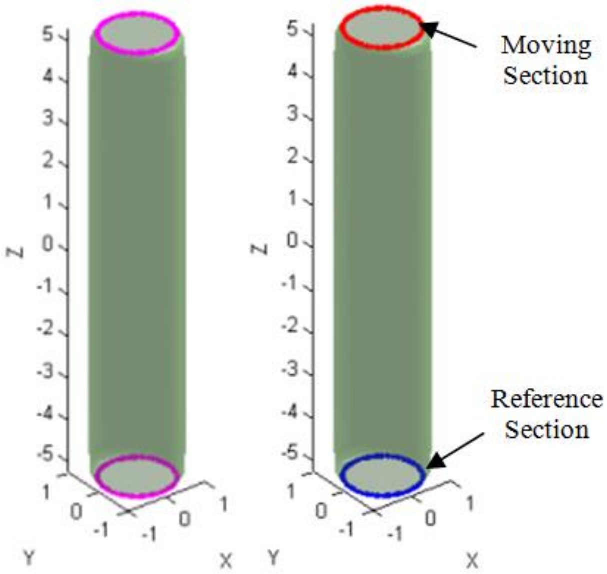

**Figure 5.** Reference and moving sections

### 2.3. Moving Section Progression

Once the first moving section is detected, it must travel along the tubular geometry form; until it reaches the reference section (refer to Figure 6). This progress is performed by looking at the adjacent vertices of the actual moving section, in order to find the nearest section. This task can be done by calculating the distance between the moving section vertices $V_{mov}$ and all mesh $M$ vertices, in order to select the vertices that minimize the distance. However, such a method is time consuming, that's why another approach is used in this paper to find the nearest section.

Assuming that the difference between two sets of vertices $A \in R^3$ and $B \in R^3$ is denoted $||$, and $a \in R^3$ is a set of vertices that belong to $A$ and don't belong to $B$:

$$A||B = \{a \in R^3, a \in A, a \notin B\}. \qquad (5)$$

The moving and reference sections, $V_{mov}$ and $V_{ref}$, are used to extract the new moving sections $V_{mov}^{new}$ through the following steps:

1. Mark all the moving section vertices $V_{mov}$ as visited vertices $V_{mov}^{\circlearrowleft}$.
2. Ignore the visited vertices $V_{mov}^{\circlearrowleft}$ in the mesh $M$ vertices. Accordingly, a new mesh is obtained using equation (5): $M^{new} = M||V_{mov}^{\circlearrowleft}$.
3. Extract the new external edges $E_{ext}^{new}$ and vertices $V_{ext}^{new}$ from the new mesh $M^{new}$.
4. The new moving section is obtained using equation (5): $V_{mov}^{new} = V_{ext}^{new}||V_{ref}$.
5. Repeat Steps 1, 2, 3, and 4 until $M^{new} = V_{ext}^{new}$.

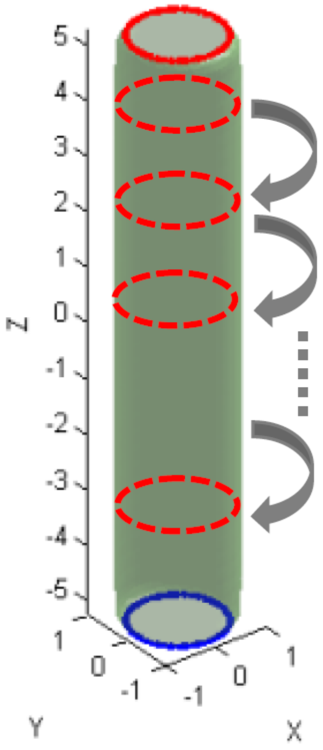

**Figure 6.** Moving section progression (red color)

### 2.4. Centerline Extraction

After the identification of the moving sections along the tubular geometry form, the centerline is defined as the union of all the centroids $C$ of the $\varphi$ existent moving sections $V_{mov_{1,\dots,\varphi}}$, from the first moving section to the reference section:

$$V_{mov_k} = \{(v_i^{x,k}, v_i^{y,k}, v_i^{z,k}) \in R^3, 1 \leq i \leq m\} \quad (6)$$

$$C_k = \begin{cases} c_{x,k} = \overline{v^{x,k}} = \frac{1}{m}\sum_{i=1}^{m} v^{x,k} \\ c_{y,k} = \overline{v^{y,k}} = \frac{1}{m}\sum_{i=1}^{m} v^{y,k} \\ c_{z,k} = \overline{v^{z,k}} = \frac{1}{m}\sum_{i=1}^{m} v^{z,k} \end{cases} \qquad (7)$$

$$3D - MCE\ centerline = C_{3D-MCE} = \{\bigcup_{k=1}^{\varphi} C_k\}. \ (8)$$

## 3. Experiments and Results

The 3D-MCE algorithm was tested on 3D generic models, the obtained centerlines can be seen in Figure 7 and, on a 3D colon model constructed from computed tomography CT colonoscopy dataset downloaded from the cancer imaging archive [39], the obtained centerline is shown in Figure 8.

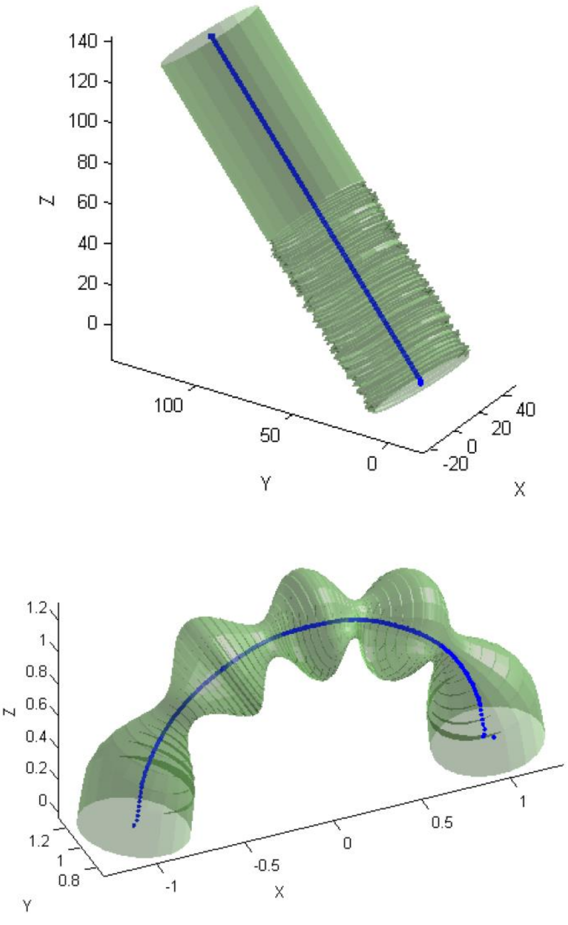

**Figure 7.** 3D-MCE centerline (blue color) of 3D generic models (green color)

In order to compare the 3D-MCE centerline with the widely used Voronoi diagram centerline [29, 30] integrated within the Vascular Modeling Toolkit open-source software package (VMTK) [40], ten 3D helical tubular models with irregular radius were generated by creating irregular circular helix curves of radius $a$ and pitch $2\pi b$ using the known parametric equation (9) (refer to Figure 9.a).

$$r(t) = \big(x(t), y(t), z(t)\big) = \begin{cases} x(t) = a\cos(t) \\ y(t) = a\sin(t) \\ z(t) = bt \end{cases} \qquad (9)$$

As shown in Tables 1 and 2, the 3D helical tubular models have different radius $a$, slope $b$ and dissimilar number of mesh triangles.

To evaluate the centerlines produced by 3D-MCE and Voronoi diagram algorithms on the ten created 3D models, two fits in two dimensional subspaces were done:

1. Iterative nonlinear least squares circular fit [41, 42] to estimate the radius $\hat{a}$ on the two dimensional subspaces $\{X, Y\}$ of the 3D centerline (refer to Figure 9.b).
2. On the $\{Y, Z\}$ plane of the 3D centerline, the distance between two points that make a complete revolution is used to find the slope $\hat{b}$ (refer to Figure 9.c).

The results of estimated radius $\hat{a}$ and slope $\hat{b}$ from the 3D-MCE and Voronoi diagram centerlines of the ten created 3D models are depicted in Tables 1 and 2.

The main contribution of 3D-MCE algorithm is its autonomy and accuracy; the 3D centerline is generated automatically without needing any input parameters, and it produces a centered, connected and thin centerline even if the 3D models are noisy (refer to Figures 7, 8 and 9.a), because it calculates, for each section of the 3D tubular form, a center of gravity, in an iterative manner, which gives directly a one voxel thick centerline with no need to any prune stages.

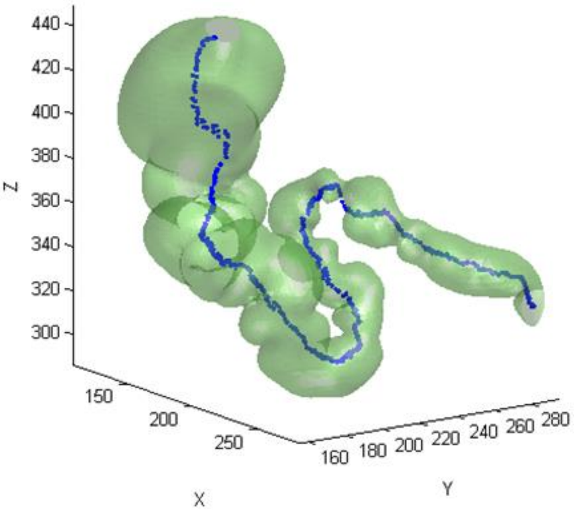

**Figure 8.** 3D-MCE centreline (blue color) of a 3D colon model (green color)

Compared to the 3D-MCE algorithm, the Voronoi diagram centerline extraction procedure is not fully automatic; in the Voronoi method, the user has to define two points: a seed point where the desired centerline starts, and a target point where the desired centerline ends. A small change in the position of those points can produce a strong effect on the generated centerline. Although the level of accuracy is close to the Voronoi diagram, the proposed method is fully automatic and it does not involve complicated mathematical tool.

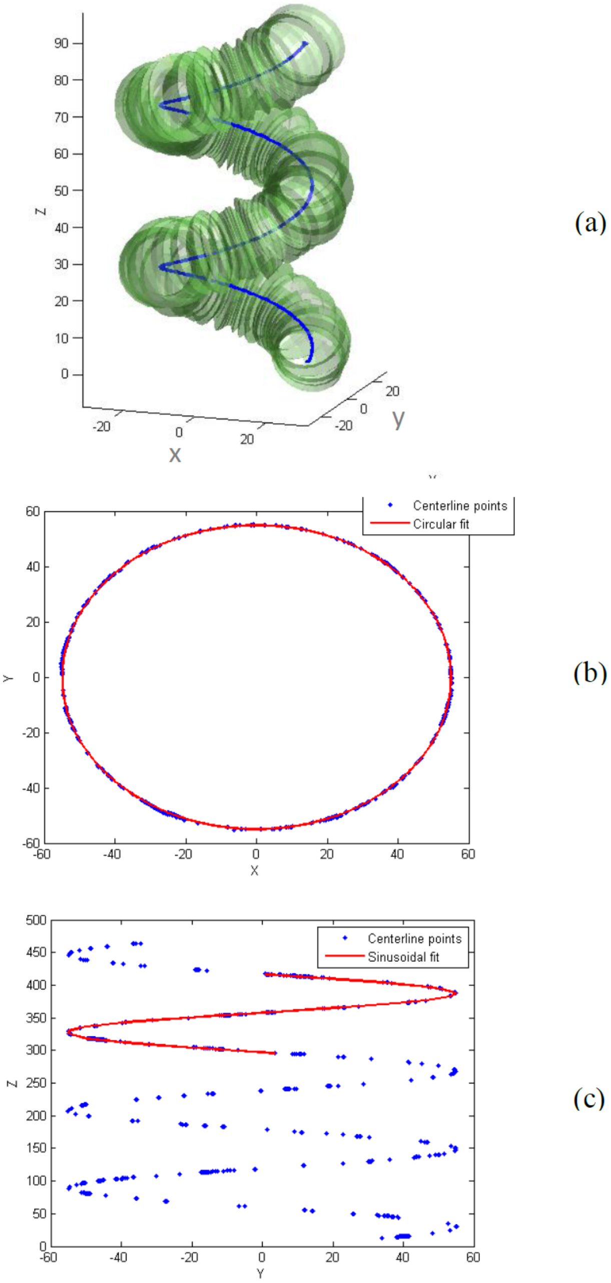

**Figure 9.** (a) 3D helical tubular model with irregular radius (green color) and its 3D-MCE centerline (blue color), (b) Nonlinear least squares circular fit (red color) of 3D-MCE centerline points (blue color), (c) Sinusoidal fit (red color) of 3D-MCE centerline points (blue color)

**Table 1.** Radius based comparison between 3D-MCE and Voronoi diagram methods

| | | | | Estimation of the Radius of the Models | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 3D-MCE method | | | | Voronoi method | | | |
| *Model* | *a* | *b* | *Number of triangles* | $\hat{a}$ | $\sigma$ | $\hat{a}_{max}$ *at 95%* | $\hat{a}_{min}$ *at 95%* | $\hat{a}$ | $\sigma$ | $\hat{a}_{max}$ at 95% | $\hat{a}_{min}$ at 95% |
| M1 | 20 | 9 | 20040 | 19,9779 | 0,0138 | 20,0049 | 19,9509 | 20,1038 | 0,1284 | 20,3556 | 19,8521 |
| M2 | 10 | 6 | 13320 | 9,9863 | 0,0379 | 10,0606 | 9,9121 | 10,1695 | 0,1359 | 10,4358 | 9,9033 |
| M3 | 43 | 7 | 9960 | 42,81 | 0,0154 | 42,8402 | 42,7798 | 42,902 | 0,036 | 42,9726 | 42,8314 |
| M4 | 30 | 15 | 22280 | 29,9737 | 0,0136 | 30,0003 | 29,9471 | 30,0466 | 0,0923 | 30,2274 | 29,8657 |
| M5 | 20 | 35 | 25080 | 19,9907 | 0,0371 | 20,0634 | 19,918 | 20,014 | 0,0443 | 20,1008 | 19,9271 |
| M6 | 9 | 15 | 18200 | 8,9896 | 0,0384 | 9,0649 | 8,9143 | 9,0832 | 0,0868 | 9,2532 | 8,9131 |
| M7 | 60 | 70 | 28680 | 59,9678 | 0,0134 | 59,9942 | 59,9415 | 59,9119 | 0,0205 | 59,9521 | 59,8717 |
| M8 | 27 | 11 | 33400 | 26,9891 | 0,0134 | 27,0154 | 26,9628 | 27,0633 | 0,1047 | 27,2686 | 26,858 |
| M9 | 100 | 11 | 40120 | 99,9729 | 0,0138 | 100 | 99,9459 | 99,8626 | 0,0309 | 99,9232 | 99,8019 |
| M10 | 55 | 19 | 50200 | 54,9994 | 0,0402 | 55,0781 | 54,9207 | 55,0264 | 0,1157 | 55,2532 | 54,7995 |

**Table 2.** Slope based comparison between 3D-MCE and Voronoi diagram methods

| | | | | Estimation of the Slope of the models | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | 3D-MCE method | | | | Voronoi method | | | |
| *Model* | *a* | *b* | *Number of triangles* | $\hat{b}$ | $\sigma$ | $\hat{b}_{max}$ at 95% | $\hat{b}_{min}$ at 95% | $\hat{b}$ | $\sigma$ | $\hat{b}_{max}$ at 95% | $\hat{b}_{min}$ at 95% |
| M1 | 20 | 9 | 20040 | 8,9934 | 0,0542 | 9,0997 | 8,8871 | 9,0163 | 0,25 | 9,5062 | 8,5263 |
| M2 | 10 | 6 | 13320 | 5,9953 | 0,034 | 6,0619 | 5,9288 | 6,0357 | 0,1878 | 6,4038 | 5,6677 |
| M3 | 43 | 7 | 9960 | 6,9917 | 0,0656 | 7,1203 | 6,8631 | 6,9916 | 0,0353 | 7,0608 | 6,9224 |
| M4 | 30 | 15 | 22280 | 15,0157 | 0,0432 | 15,1003 | 14,9311 | 14,9858 | 0,094 | 15,17 | 14,8016 |
| M5 | 20 | 35 | 25080 | 34,9873 | 0,0158 | 35,0183 | 34,9564 | 34,9715 | 0,5022 | 35,9558 | 33,9871 |
| M6 | 9 | 15 | 18200 | 15,0093 | 0,0705 | 15,1475 | 14,8711 | 14,9133 | 0,7154 | 16,3155 | 13,5111 |
| M7 | 60 | 70 | 28680 | 69,9275 | 0,2579 | 70,4329 | 69,422 | 69,9724 | 0,0896 | 70,1479 | 69,7969 |
| M8 | 27 | 11 | 33400 | 10,9947 | 0,0344 | 11,0621 | 10,9273 | 10,9807 | 0,1292 | 11,2339 | 10,7275 |
| M9 | 100 | 11 | 40120 | 11,0592 | 0,029 | 11,0592 | 10,9455 | 10,9743 | 0,0073 | 10,9886 | 10,9599 |
| M10 | 55 | 19 | 50200 | 18,9933 | 0,0012 | 18,9933 | 18,9887 | 19,2071 | 0,3941 | 19,9796 | 18,4347 |

As shown in Tables 1 and 2, the 3D-MCE algorithm gives a smaller 95% confidence interval compared to the Voronoi diagram, which means that the region around the true value is smaller for the 3D-MCE, compared with the Voronoi diagram method. Results show that for the both estimated parameters (radius and slope), the proposed 3D-MCE approach provides narrower 95 % confidence intervals compared to Voronoi diagram method.

In order to evaluate the sensitivity of 3D-MCE method to the density of triangles of the 3D models, fourteen helical models were generated using the equation (9), with constant radius $a = 25$ and slope $b = 12$ and with different number of triangles contained in the mesh. The 3D-MCE algorithm was performed on the generated models and, on each obtained centerline, two fits in two dimensional subspaces were done: iterative nonlinear least squares circular fit on $\{X, Y\}$ subspace to estimate the radius $\hat{a}$ and, sinusoidal fit on $\{Y, Z\}$ subspace to estimate the slope $\hat{b}$ (refer to Figures 9b and 9c). The obtained slopes and radii of the 3D models are represented in Figure 10. The results show that the accuracy of 3D-MCE is less sensitive to the density of triangles of the 3D model, and it is close to the true value, when the number of triangles contained in the mesh is dense.

## 4. Conclusion

In this paper, a new, automatic and parameters-free 3D-MCE extraction algorithm for tubular geometry form is presented. 3D-MCE algorithm provides many advantages compared to the existing methods, such as:

- It does not involve complicated mathematical tools.
- It is invariant with regard to model rotation, translation and geometric transformations.
- It gives directly a one voxel thick centerline without any further post-processing stages.

The 3D-MCE algorithm uses only the triangular mesh of the 3D tubular geometry form to calculate the centerline. Our experiments show that the centerline is centered, connected and thin even if the 3D models are noisy. Results show also that the 3D-MCE performs correctly for the tubular geometry forms, independently from mesh density. The 3D-MCE algorithm offers a high level of accuracy, besides, it is fully automatic and, easy to implement.

Our future work will address the extension of 3D-MCE algorithm to branched tubular geometry form.

## Acknowledgments

CT colonoscopy dataset used in this research was obtained from the cancer imaging archive (http://cancerimagingarchive.net/) sponsored by the cancer imaging program, DCTD/NCI/NIH.

Authors would like to thank Samira Bouchama and Cristina Vidrutiu for their careful review of this manuscript.

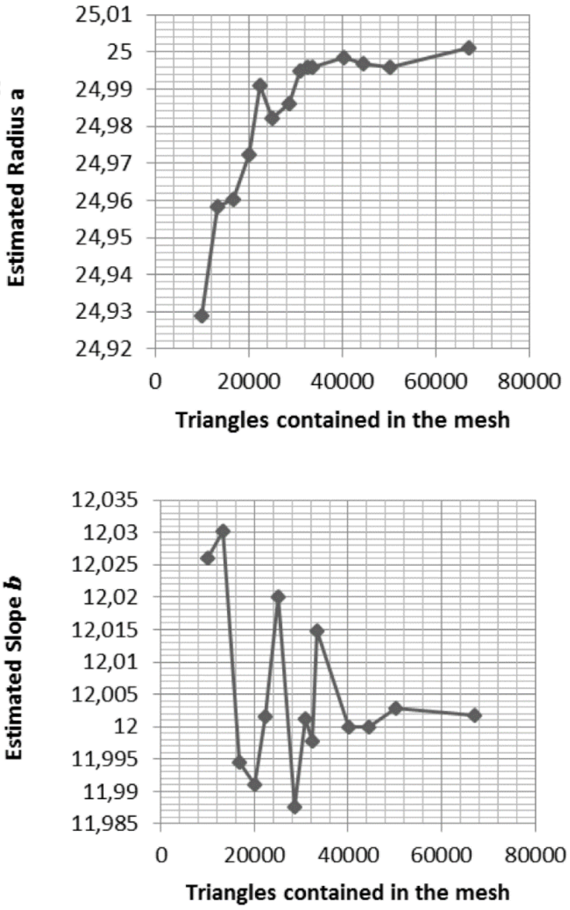

**Figure 10.** Sensitivity of the radius and slope of the tubular forms created using Eq. (9) with constant Radius $a = 25$ and Slope $b = 12$ to the number of triangles contained in the mesh

## References

[1] **H. Blum.** A transformation for extracting new descriptors of shape. In: *Proc. Symp. Models for the Perception of Speech and Visual Form*, MIT Press, Cambridge, MA, 1967, pp. 362-380.

[2] **J. Wang, T. Ohya, H. Liao, I. Sakuma, T. Wang, I. Tohnai, T. Iwai**. Intravascular catheter navigation using path planning and virtual visual feedback for oral cancer treatment. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 2011, Vol. 7, No. 2, 214-224.

[3] **J. D. Gibbs, M. W. Graham, W. E. Higgins.** 3D MDCT-based system for planning peripheral bronchoscopic procedures. *Computers in Biology and Medicine*, 2009, Vol. 39, 266-279.

[4] **E. Marchandise, C. Geuzaine, J. Remacle.** Cardiovascular and lung mesh generation based on centerlines. *International Journal for Numerical Methods in Biomedical Engineering*, 2013, Vol. 29, No. 6, 665-682.

[5] **W. R. Quadros, V. Vyas, M. Brewer, S. J. Owen, K. Shimada**. A computational framework for automating generation of sizing function in assembly meshing via disconnected skeletons. *Engineering with Computers*, 2010, Vol. 26, Issue 3, 231-247.

[6] **C. Wang, Q. Ma, D. Zhu, H. Chen, Z. Yang**. Real-time control of 3D virtual human motion using a depth-sensing camera for agricultural machinery training. *Mathematical and Computer Modelling*, 2013, Vol. 58, 782-789.

[7] **E. De Aguiar, C. Theobalt, S. Thrun, H. P. Seidel.** Automatic Conversion of Mesh Animations into Skeleton-based Animations. *Computer Graphics Forum*, 2008, Vol. 27, No. 2, pp. 389-397.

[8] **A. Firouzmanesh, I. Cheng, A. Basu.** Perceptually guided fast compression of 3-d motion capture data. *IEEE Transactions on Multimedia*, 2011, Vol. 13, pp. 829-834.

[9] **H. Peng, Z. Ruan, F. Long, J. H. Simpson, E. W. Myers**. V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets. *Nature Biotechnology*, 2010, Vol. 28, No. 4, 348-353.

[10] **A. W. Vieira, E. R. Nascimento, G. L. Oliveira, Z. Liu, M. F. Campos**. Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences. In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, ed: Springer, 2012, pp. 252-259.

[11] **H. Sundar, D. Silver, N. Gagvani, S. Dickinson**, Skeleton based shape matching and retrieval. In: *Proceedings of Shape Modeling International*, 2003, pp. 130-139.

[12] **C. Bauer, T. Pock, E. Sorantin, H. Bischof, R. Beichel.** Segmentation of interwoven 3d tubular tree structures utilizing shape priors and graph cuts. *Medical Image Analysis*, 2010, Vol. 14, 172-184.

[13] **S. Worz, H. von Tengg-Kobligk, V. Henninger, F. Rengier, H. Schumacher, D. Bockler, H.-U. Kauczor, K. Rohr**. 3-D quantification of the aortic arch morphology in 3-D CTA data for endovascular aortic repair. *IEEE Transactions on Biomedical Engineering*, 2010, Vol. 57, 2359-2368.

[14] **M. Couprie, G. Bertrand**. New characterizations of simple points in 2D, 3D, and 4D discrete spaces. *IEEE*

*Transactions on Pattern Analysis and Machine Intelligence*, 2009, Vol. 31, 637-648.

[15] **P. Kardos**. Sufficient Conditions for Order-Independency in Sequential Thinning. *Acta Cybern.*, 2011, Vol. 20, No. 1, 87-100.

[16] **K. Palágyi**. A Sequential 3D Curve-Thinning Algorithm Based on Isthmuses. In: *Advances in Visual Computing*, ed: Springer, 2014, pp. 406-415.

[17] **K. Palágyi, G. Németh, P. Kardos**. Topology preserving parallel 3D thinning algorithms. In: *Digital Geometry Algorithms*, ed: Springer, 2012, pp. 165-188.

[18] **K. Palágyi**. Parallel 3D 12-Subiteration Thinning Algorithms Based on Isthmuses. In: *Advances in Visual Computing*, ed: Springer, 2013, pp. 87-98.

[19] **P. Kardos, K. Palágyi**. On topology preservation for triangular thinning algorithms. In: *Combinatorial Image Analysis*, ed: Springer, 2012, pp. 128-142.

[20] **C. Arcelli, G. Sanniti di Baja, L. Serino**. Distance-driven skeletonization in voxel images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011, Vol. 33, No. 4, 709-720.

[21] **W. Liu, H. Jiang, X. Bai, G. Tan, C. Wang, W. Liu, K. Cai**. Distance transform-based skeleton extraction and its applications in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 2013, Vol. 24, No. 9, 1763-1772.

[22] **C. Arcelli, G. Sanniti di Baja, L. Serino**. A parallel algorithm to skeletonize the distance transform of 3D objects. *Image and Vision Computing*, 2009, Vol. 27, No. 6, 666-672.

[23] **E. Žitkevičius, D. Grigaitis, D. Navakauskas**. On skeletonization of blood vessels in angiographic MRI images of human brain. *Information Technology and Control*, 2007, Vol. 36, No. 4, 372-376.

[24] **H. E. Çetingül, M. A. Gülsün, H. Tek.** A unified minimal path tracking and topology characterization approach for vascular analysis. In: *Medical Imaging and Augmented Reality*, ed: Springer, 2010, pp. 11-20.

[25] **F. Benmansour, L. D. Cohen**. Tubular structure segmentation based on minimal path method and anisotropic enhancement. *International Journal of Computer Vision*, 2011, Vol. 92, No. 2, 192-210.

[26] **T. K. Dey, W. Zhao**. Approximating the medial axis from the Voronoi diagram with a convergence guarantee. *Algorithmica*, 2004, Vol. 38, No. 1, 179-200.

[27] **Y. J. Liu, Z. Chen, K. Tang.** Construction of iso-contours, bisectors, and Voronoi diagrams on triangulated surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011, Vol. 33, No. 8, 1502-1517.

[28] **F. Aurenhammer, R. Klein, D.-T. Lee**. Voronoi Diagrams and Delaunay Triangulations: World Scientific, 2013.

[29] **M. Piccinelli, A. Veneziani**, **D. A. Steinman, A. Remuzzi, L. Antiga**. A framework for geometric analysis of vascular structures: application to cerebral aneurysms. *IEEE Transactions on Medical Imaging*, 2009, Vol. 28, No. 8, 1141-1155.

[30] **L. Antiga, M. Piccinelli, L. Botti, B. Ene-Iordache, A. Remuzzi, D. A. Steinman**. An image-based modeling framework for patient-specific computational hemodynamics. *Medical & Biological Engineering & Computing*, 2008, Vol. 46, No. 11, 1097-1112.

[31] **F. Yu, Z. Lu, H. Luo, P. Wang**. Three-dimensional model analysis and processing. *Springer*, 2011.

[32] **M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, B. Lévy.** Polygon mesh processing. *CRC Press*, 2010.

[33] **J. Toriwaki, H. Yoshida**. Fundamentals of three-dimensional digital image processing. *Springer*, 2009.

[34] **Z. Bar-Joseph, D. K. Gifford, T. S. Jaakkola**. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, 2001, Vol. 17, No. S1, S22-S29.

[35] **M. B. Eisen, P. T. Spellman, P. O. Brown, D. Botstein**. Cluster analysis and display of genome-wide expression patterns. In: *Proceedings of the National Academy of Sciences*, 1998, Vol. 95, pp. 14863-14868.

[36] **J. L. DeRisi, V. R. Iyer, P. O. Brown**. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 1997, Vol. 278, No. 5338, 680-686.

[37] **T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri.** Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 1999, Vol. 286, No. 5439, 531-537.

[38] **M. Goncalves, M. Netto, J. Costa, J. Zullo Junior**. An unsupervised method of classifying remotely sensed images using Kohonen self-organizing maps and agglomerative hierarchical clustering methods. *International Journal of Remote Sensing*, 2008, Vol. 29, pp. 3171-3207.

[39] **K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle**. The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository. *Journal of Digital Imaging*, 2013, Vol. 26, No. 6, 1045-1057.

[40] **The Vascular Modeling Toolkit**. Available: http://www.vmtk.org/

[41] **J. J. Moré**. The Levenberg-Marquardt algorithm: implementation and theory. In: *Numerical Analysis*, ed: Springer, 1978, pp. 105-116.

[42] **T. F. Coleman, Y. Li.** An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 1996, Vol. 6, No. 4, 418-445.

Received April 2015.