Centerlines of Tubular Volumes Based on Orthogonal Plane Estimation

Florent Grélard[1,2](✉), Fabien Baldacci[1,2], Anne Vialard[1,2], and Jean-Philippe Domenger[1,2]

[1] Univ. Bordeaux, LaBRI, UMR 5800, 33400 Talence, France
florent.grelard@labri.fr
[2] CNRS, LaBRI, UMR 5800, 33400 Talence, France

Abstract. In this article we present a new method to compute a centerline on tubular volumes. The curve-skeleton is central to many applications in discrete geometry, since it captures interesting geometrical and topological properties with respect to the initial volume. Although there are numerous algorithms for skeleton computation, they are not necessarily well-suited for tubular volume specificities, and can lead to unexpected results (faulty branches, not complete). Our method works on tubular-like volumes with junctions (tree structures) and varying diameter. It is based on the center of mass of cross-sections computed using a Voronoï covariance measure on the volume. Our method adapts its parameters to the shape of the tube. The results on both synthetic and real tubular structures with non-constant diameter illustrate the method's efficiency.

Keywords: Centerline · Voronoï covariance measure · Tubular structure analysis

1 Introduction

Simplification of 3D models is of major importance in shape analysis. By reducing dimensionality in particular, skeletonization algorithms applied to 3D objects yield a set of curves and/or surfaces. The particular problem of reducing a volume to a set of curves, known as centerline and curve-skeleton extraction, is key to many applications. For example, the centerline is used for the geometric analysis of 3D shapes [4,15], virtual endoscopy [3] and object matching [2]. Tubular volumes can be found in medical applications (vessels, airways, neurons and colons) but also in industrial applications [4,12]. The afore-mentioned tubes are characterized by elongation in one direction, varying diameter and varying aspect-ratio. They are topologically equivalent to a tree or a graph (see Fig. 1). In [5], the authors describe various desirable properties for a curve-skeleton, such as: same topology as the initial object, centeredness, invariance under isometric transformations, robustness to noise, thinness, junction-detection and smoothness.

N. Normand et al. (Eds.): DGCI 2016, LNCS 9647, pp. 427–438, 2016.
DOI: 10.1007/978-3-319-32360-2_33

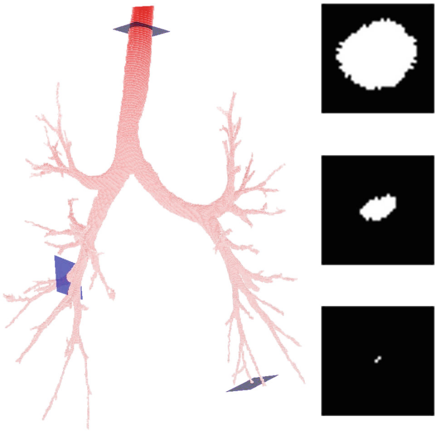


Fig. 1. Example of a tubular volume and three cross-sections, and their 2D equivalent on the right showing variations in diameter and ellipticity.

Satisfying all these wanted properties on any volume is not achievable, due to the data variability and complexity. However, given that our volumes are tubular and according to existing applications, only a subset of these properties are required. Indeed, for length and curvature estimation the curve-skeleton must be thin, for virtual endoscopy the points must be centered and the junctions detected, and for object matching the topology must be preserved. Moreover, since our volumes have varying diameter, the skeleton must be complete i.e. capture information at the lowest local levels. In this paper, an original approach for computing a centerline on any kind of tubes is proposed. It is based on the tracking of centers of mass which are extracted from orthogonal planes computed on the volume.

In Sect. 2 related works with different approaches towards curve-skeleton extraction are presented. Section 3 introduces orthogonal plane computation on a volume. In Sect. 4 the proposed method is explained. Finally, results obtained on both synthetic and real data are described and discussed in Sect. 5, with quantitative comparison with other methods.

2 Related Works

We focus here only on the skeletonization methods which process 3D voxel objects and compute curve-skeletons. According to the survey [5] we can classify the curve-skeleton algorithms for discrete objects in the following classes: (a) thinning algorithms and (b) distance field or general field based algorithms.

Among existing methods, we cite as examples [7,14] for the thinning class. The first one can handle any 3D voxel volume while the second is dedicated to three-dimensional tubular structures. The authors in [7] propose to compute the skeleton by filtering the medial axis with a new bisector function. The resulting filtered medial axis is used as constraint set for the computation of the euclidean skeleton by homotopic thinning. However, the skeleton computation depends on two parameters: r which allows to filter small maximal balls, and α, a threshold for the angle of the bisector function. This method, like other thinning methods, can create small faulty branches (see Fig. 2a), often removed by a pruning step, which are, in our case, difficult to distinguish from small branches that are representative of the volume.

As examples of the second class of curve-skeleton algorithms, we can cite the methods of [6,11] where the skeleton is extracted from a potential field computed on the volume. In [6], the authors compute a hierarchical skeleton based on a newtonian potential field. Each boundary point is considered as an electric charge, repelling interior points of the volume. From the potential vector field, critical points (points where the magnitude of the force vector vanishes) are extracted and connected thanks to paths, yielding a first hierarchy in the skeleton. Then, two levels of hierarchy are added through high divergence value and high-curvature points. These points are used as seeds to recompute new potential fields and extract details on the underlying shape. The main drawback of this kind of algorithm is that the resulting skeleton is not necessarily connected and can be incomplete (see Fig. 2b). Other methods can be found in the recent survey [17]. Among them, the method described in [16] computes a multiscale skeleton by defining an importance measure for each volume point. This method is not adapted to the previously described data: since the importance measure is based on an area computation, small branches in the volume do not appear in the skeleton.

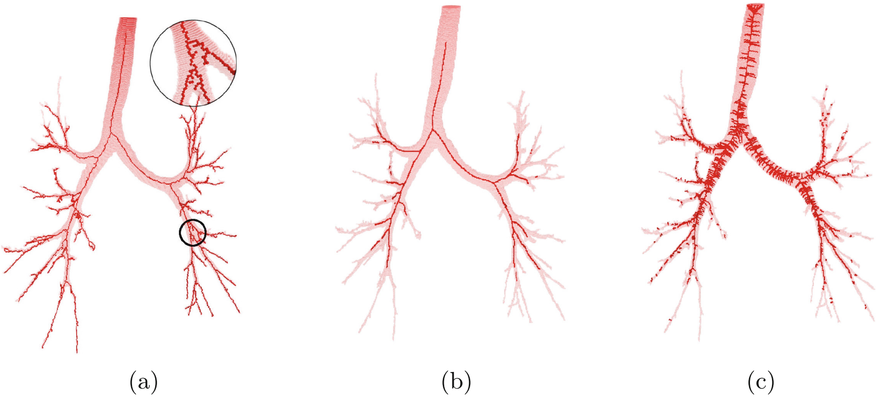


Fig. 2. (a) Skeleton computed using the method described in [7]: parameter values were chosen to obtain the most complete skeleton with the least amount of faulty branches, but there are some regardless, see closeup view. (b) (c) Skeleton computed using potential field [6]: two levels of hierarchy on a skeleton obtained with two different parameter values. In both cases, obtaining a skeleton containing all the branches in the volume as well as no faulty branches is not possible.

This paper addresses the shortcomings of the afore-mentioned methods, and proposes an accurate curve-skeleton extraction on tubular volumes which adapts its parameters to the shape of the tubular volume.

3 Orthogonal Plane Estimation

In [10], we described a way to compute orthogonal planes from a voxel set. Our method is based on a Voronoï covariance measure (VCM). The VCM was first introduced in [13] on point clouds to estimate surface normals, and extended in digital geometry in [8]. Moreover the VCM is proven to be multigrid convergent and resilient to Hausdorff noise. The principal direction of a Voronoï cell corresponds to the surface normal, defined in the computed covariance matrix by the eigenvector with the largest eigenvalue. The VCM computes at a point y in a digital object O a covariance matrix of row vectors between points in a domain of integration DI centered at y and the sites of their respective Voronoï cells, denoted by $p_O(x)$ (see Fig. 3a). The covariance matrix is given as:

$$\mathcal{V}_O(y) = \sum_{x \in DI_O(y)} (x - p_O(x))(x - p_O(x))^t$$

where DI is for example a ball of radius R.

In [10], we compute orthogonal planes on a volume by summing the contribution of all the normal cones $\mathcal{N}$ in the domain of integration (see Fig. 3b). The orthogonal plane is defined as the plane spanned by the two eigenvectors with the highest eigenvalues in the covariance matrix, i.e. those which define the normal cone.

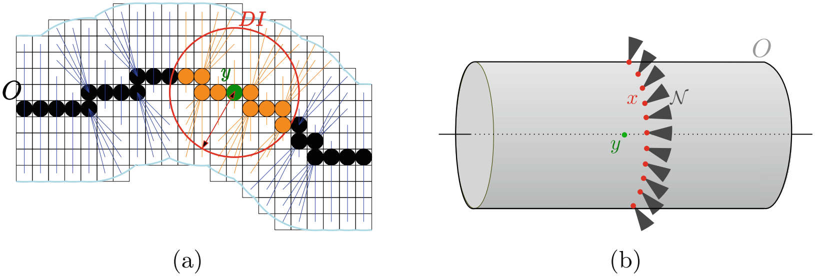


Fig. 3. (a) 2D digital object (in black, O) and computation of the VCM around y by integration of vectors (in orange) in the domain of integration, denoted by DI. (b) Computation of an orthogonal plane for a point (in green) inside the volume (cylinder), and relationship with flat normal cones (in dark gray, denoted by $\mathcal{N}$) at different points on the surface inside the domain of integration (in red). For sake of clarity, the normal cones are separated by blank areas, without these the union of all the normal cones corresponds to the expected orthogonal plane (Color figure online).

In this paper, a new approach based on orthogonal planes for curve-skeleton computation on tubular volumes is presented.

4 Proposed Method

Let $O \subseteq \mathbb{Z}^3$ be an object, $\mathcal{P}(p)$ an orthogonal plane at a point p, R the radius of the domain of integration of the VCM used for the orthogonal plane computation, as described in Sect. 3. The main idea of our algorithm is that the curve-skeleton C of O is the set of centers of mass of the orthogonal planes computed on the volume. C is obtained by tracking the centers of mass iteratively (as described in detail in the "Tracking" paragraph).

The parameter R must be "well-adjusted" to capture locally the shape of the object, while being robust to shape irregularities. In other words, the domain of integration should be adjusted such that it contains (a) surface points for which normal cones are aligned with the expected orthogonal plane and (b) the least amount of irrelevant surface points. In the following paragraph, using a ball as domain of integration, we describe how to obtain the minimal value for its radius R automatically.

The size of the domain of integration is initialized with the distance transform (DT) value. Obviously, this value does not correspond to the expected value for R in the case of a tube with elliptic cross-sections, or when the orthogonal plane is computed at a point p near the boundary of the object. The process is the following: an initial orthogonal plane $\mathcal{P}_0(p)$ is computed with $R = DT(p)$. If there is at least one point x in $O \cap \mathcal{P}(p)$, such that $d(p, x) > R$, then R is incremented. This process is repeated until no points in $O \cap \mathcal{P}(p)$ are found at $d > R$. It allows to converge towards both the expected orthogonal plane and the minimal value for R to include all the surface points in $O \cap \mathcal{P}(p)$.

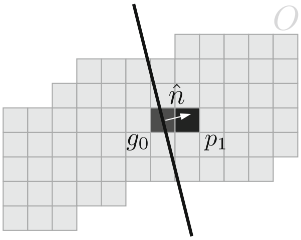


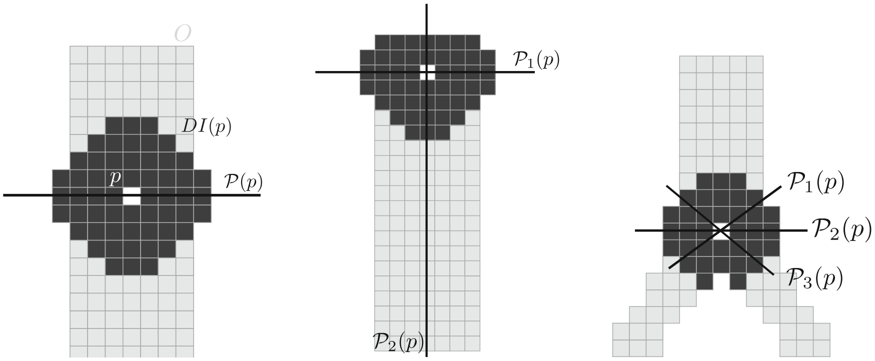
Fig. 4. Tracking performed by our algorithm. From a determined center of mass g_0, we propagate to the next point p_1 thanks to the plane normal $\hat{n}$.

Tracking. It is computationally expensive and redundant to go through all points in the volume. The intuitive idea to solve this problem is that the volume is processed layer by layer, orthogonal plane by orthogonal plane. The layer contains a unique curve-skeleton point so it does not need to be processed further. As a result, points belonging to each computed orthogonal plane are stored in a set M, which are not considered in the rest of the computation.

The starting point p_0 of the algorithm is the point in the volume which has the highest DT value, as it is centered. After the orthogonal plane computation at a point p_i we obtain a center of mass g_i. We then propagate to the next point p_{i+1} in the 26-neighborhood of g_i with the normalized plane normal:

$$p_{i+1} = \begin{cases} g_i + \boldsymbol{n}_{\boldsymbol{\mathcal{P}(p_i)}} & \text{if } p_{i+1} \notin M \\ \max_{q \in (O \setminus M)} DT(q) & \text{otherwise} \end{cases}$$

where $\boldsymbol{n}_{\boldsymbol{\mathcal{P}(p_i)}}$ is the plane normal vector. Since $g_i + \boldsymbol{n}_{\boldsymbol{\mathcal{P}(p_i)}} \approx g_{i+1}$ in regular (i.e. convex and flat) portions of the object, propagation is done either on the next center of mass or on its neighborhood (see Fig. 4). This ensures the algorithm prioritizes points which are well-centered and for which the orthogonal plane normal is close or equivalent to the plane normal after convergence. This process is done for the two normal directions ($p_0 + \boldsymbol{n}_{\boldsymbol{\mathcal{P}(p_0)}}$ and $p_0 - \boldsymbol{n}_{\boldsymbol{\mathcal{P}(p_0)}}$). If $p_{i+1} \in M$, the set of marked vertices, a new point p_{i+1} is assigned from the highest DT value.



(a) (b) (c)

Fig. 5. Computation of orthogonal lines on 2D objects (in light gray). Different configurations arise: (a) regular case in the tube, and resulting orthogonal line (b) endpoint: contribution of two orthogonal lines $\mathcal{P}_1(p)$ and $\mathcal{P}_2(p)$ and (c) junction area: contribution of three orthogonal lines (one per branch).

For some points $p \in O$, the orthogonal plane is not defined because there are several incompatible contributions on the surface. This is illustrated in Fig. 5b for endpoints, and in Fig. 5c for points in junctions. The following paragraphs describe how to specifically detect and handle these two types of points.

Junctions. A junction consists in the intersection of three or more tubes. The reason why orthogonal plane computation performs so poorly in such areas is because the branches forming the junctions have different orthogonal planes, which are all relevant, but cannot be computed in a unique operation. Junctions correspond to high-curvature points on the surface of the volume. In our method, high-curvature points are processed differently from the rest of the points.

Our approach was inspired from the sharp-edge detection method described in [13]. The purpose is to detect such high-curvature points, and more particularly the concave parts of the object. In such parts, and under good sampling conditions, the eigenvalues of every vector in the covariance matrix of the Voronoï cells are roughly the same (i.e. small, see Fig. 6a). As a result, we considered the following method to detect concave points: from the eigenvalues, we compute the concave feature given as:

$$f_c(p) = \frac{\lambda_0}{\sum_i \lambda_i}$$

where λ_0 is the lowest eigenvalue. This captures the fact that the lowest eigenvalue is in the same order of magnitude as the rest of the eigenvalues only for concave points. We consider p a high-curvature point if $f_c(p) > t$ where t is a threshold. This threshold can be detected automatically using the Otsu thresholding method, since the classes of concave and non-concave points are clearly separated. In order to deal with the junctions parts specifically, when the domain of integration of the VCM contains a high curvature point, no further processing is done and the tracking resumes (see "Tracking" paragraph above).

At this stage, the curve-skeleton consists in various connected curves. Each curve needs to be connected to another in a second pass, such that the result is one connected component, i.e. the resulting centerline. For this, we reconstruct sub-volumes taking into account two tubes (out of the three or more in the junctions) and use the main tracking algorithm on these to compute missing parts of the skeleton. The computation follows two steps. Firstly, each extremity of an individual curve is connected to another by a line segment (see white line in Fig. 6b). For a junction consisting in the intersection of three tubes, three points have to be linked. There are three possible line segments linking each pair of points, one of which is not oriented along the main axis of a branch. The line segment which goes "near" a branching point is discarded as follows. Let L be the line segment linking two extremities e_1 and e_2, and B the set of high-curvature points. A line segment is discarded if $\exists (l,b) \in L \times B$ s.t. $d(l,b) < \min(DT(e_1), DT(e_2))$. As a second step, tubular sub-volumes are created by intersecting balls of radius $r = DT(p)$ with the volume, centered in $p \in L$. The main tracking algorithm is then used on this sub-volume, which yields a connected and geometrically-sound curve-skeleton.

Endpoints. The endpoints of the volume must also be treated specifically. This is illustrated in Fig. 5b where the Voronoï cells at the end part of the tube have an orthogonal orientation compared with the expected plane. One way to detect these points was presented in [1], where the authors compute a metric graph and detect the degree of a node in the graph. They compute the shell at p, defined by:

$$S(p) = (B_r(p) \setminus B_R(p)) \cap O$$

where $B_R(p)$ denotes a ball of radius R centered at p, and $r > R$. Here the parameter R is the same as the one described for the domain of integration

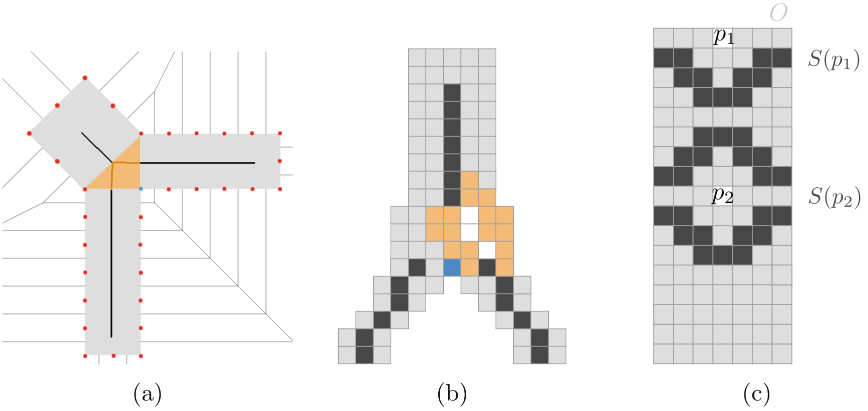


Fig. 6. (a) 2D Voronoï diagram showing concave areas (blue point) have a limited Voronoï cell and normal cone in all directions. The orange area is not considered in the first pass of the curve-skeleton computation. (b) Curves in the skeleton after the first pass (in dark gray), which are connected by a straight line (in white) avoiding a high-curvature point (in blue). From these points, a sub-volume is reconstructed (in orange) and used to compute the skeleton in junctions. (c) Shells (in dark grey) with one connected component corresponding to an endpoint (p_1) and with two components for a point in the volume (p_2) (Color figure online).

size. The number of connected components in $S(p)$ gives the degree $deg(p)$ of p (see Fig. 6c). When this number is equal to one, p is an endpoint, otherwise, when it is greater than one, p corresponds to a point inside the volume. Unlike high-curvature points, endpoints are not processed further because they do not provide relevant information for the curve-skeleton extraction. Theoretically, for junction points, the degree is equal to the number of branches. However this property cannot be used reliably for junction detection on varying-diameter tubes. In fact, given a junction, we want to set r and R such that $deg(p) = 3$. Nonetheless, for junctions where two branches have a very large diameter compared to a third branch limited in length, it is not possible to find solutions.

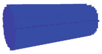
5 Results

In this section, our method's efficiency is evaluated by comparison to ground-truth curve-skeletons and to state-of-the-art methods namely euclidean skeleton thinning approach [7] and the potential field method [6] presented in Sect. 2. Although the limits of both these methods have been discussed, they satisfy interesting properties related to our applications and are tested within their field of application. Our method is tested on both synthetic and real data.

5.1 Synthetic Data

Various simple tubular volumes have been generated using a parametric curve: cylinder, torus, curved tube and curved tube with noise. Noise was added with

Table 1. Hausdorff distance (first column, in white) and F-measure (second column, in gray) for each method on different volumes.

Data	Euclidean		Potential field		Our method	
 Cylinder	**0**	0.684	1.41	0.658	**0**	**0.809**
Torus	**0**	**1**	1.41	0.494	1	0.817
Curved tube	1.41	0.590	**1**	**0.740**	1.41	0.635
Noisy curved tube	1.41	0.503	1.41	0.596	1.41	**0.683**

a simplified version of Kanungo's method, which switches the voxel's value with probability α^d where d is the distance to the boundary. For each volume, the expected curve-skeleton corresponds to the parametric curve used to generate the volume. The expected skeleton is compared to the computed skeletons using the Hausdorff distance. The Hausdorff distance allows to measure the dissimilarity between two curves. Here, it is defined by the maximal distance between a point in the computed skeleton and its nearest point in the expected skeleton, i.e.:

$$\mathrm{d}_H(T,C) = \max_{c\in c}\left\{\min_{t\in T}\mathrm{d}(t,c)\right\}$$

where T and C are the sets of points in the expected and computed skeleton respectively and d is the euclidean distance. The skeleton is centered if $d_H(T,C) \leq \sqrt{3}$, that is to say if the points in the computed skeleton are "connected" to those of the expected skeleton. Otherwise, the skeleton is thick or contains faulty branches. This distance does not allow to capture the completeness and sensitivity of our method. These two properties are estimated by the precision p and the recall r. The precision estimates whether our skeleton contains faulty points, and ranges from 0 (only faulty points) to 1 (no faulty points). The recall estimates whether the centerline is complete, and ranges from 0 (fully incomplete), to 1 (complete). The precision and recall allow to define the F-measure, which is a compromise between completeness and sensitivity. The results are presented in Table 1.

Various parameters for the state-of-the art methods are tested in order to obtain no faulty branches and the most complete skeleton. Only results with the best parameters are shown here.

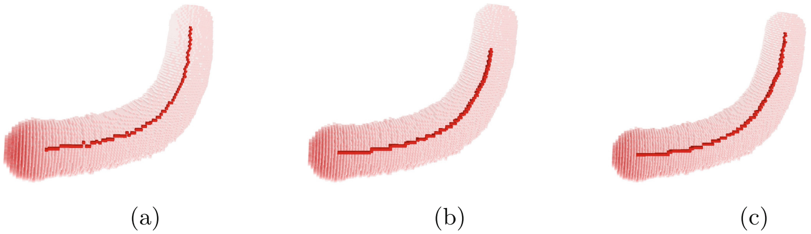


Fig. 7. Curve-skeletons computed on the curved tube using (a) thinning with euclidean skeleton (b) the potential field method and (c) our method.

For all volumes, our skeleton is centered ($d_H(T,C) \leq \sqrt{3}$), and that is the case of the other methods as well. Regarding the F-measure, our method produces the best results in average. The curve-skeleton is close to complete on the cylinder and the torus. Results on the curved tube are comparable to the euclidean skeleton. As for the noisy curved tube, the skeleton is more complete than both the potential field and the euclidean skeletons. Our skeleton is robust to deformations in the volume because our algorithm is designed to work on varying-diameter tubes. The resulting skeletons on these volumes are presented in Fig. 7.

5.2 Real Data

Our method was designed to work on varying-diameter tubular structures with junctions: results obtained from a human airway-tree acquired from CT-scan are shown. The airway-tree segmentation yields a volume topologically equivalent to a binary tree.

Figure 8a and b show the resulting skeletons on two different volumes. A general visual inspection shows the skeleton does not have the defects of the state-of-the-art methods illustrated in Sect. 2. Closeups show the resulting skeleton can capture local information (junctions, see Fig. 9a) and be exempt of faulty branches through the majority of the volume. Moreover, our skeleton is well centered and does not suffer from distortions. In addition to the fact the skeleton is obtained automatically, these properties are interesting in light of the various applications mentioned in Sect. 1. However, when the diameter is as small as one voxel, points might not be taken into account (see Fig. 9b) as they are marked and put in the set M wrongfully. For these particular small branches, the information is lost. Nonetheless, considering the level at which this problem appears, it does not seem like an issue: the branches are too small to provide relevant information. The skeleton was also computed on a tube acquired from laser scan (see Fig. 8c) with concave parts. By visual inspection, the skeleton corresponds precisely to what we would expect. More examples and images are available at [9].

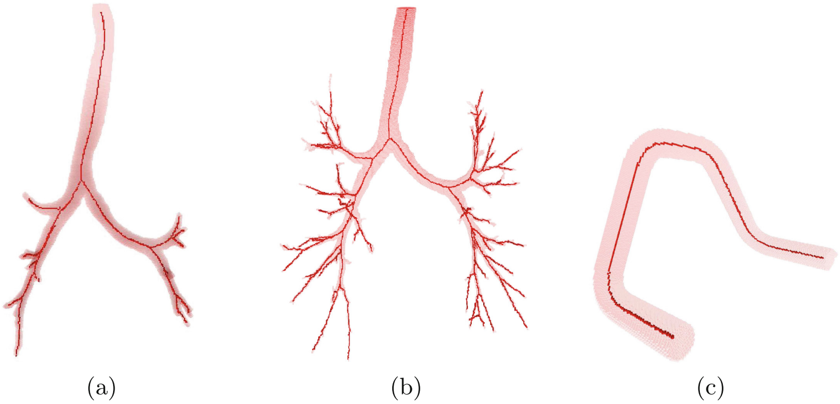


Fig. 8. (a) (b) Centerline extraction on two airway tree volumes. (c) Tube acquired from laser scan, courtesy of [12].

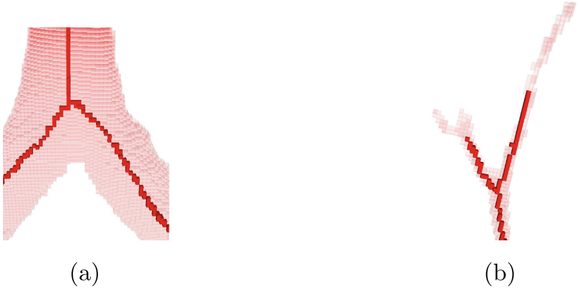


Fig. 9. Closeups of areas in the extracted centerline (a) example of a junction. (b) defect in the skeleton: the smaller branches on the left of the volume are not extracted.

6 Conclusion and Prospects

In this article, we have presented a new approach to estimate a centerline on a tubular volume. We have shown it possesses interesting properties in relation with various applications, is fully automatic, and has an interesting compromise between completeness and sensitivity. Moreover, it is centered and does not suffer from distortions. We have shown interesting results on tubular volumes, that is to say volumes which are elongated in one direction and restricted in the two other directions. This algorithm does not apply on non-tubular objects since orthogonal planes are not properly defined for these objects. This method will be developed further to ensure missing properties are satisfied in the future. For instance, connectivity is not ensured: three types of connectivity 6-, 18- and 26-connectivity are mixed in the skeleton as of yet. This property can be guaranteed easily by deleting or adding points in order to obtain the desired connectivity. We cannot ensure our method preserves topology for genus ≥ 1 objects. In particular, if there is a hole in a junction area, it will not be preserved in the resulting curve-skeleton. As another prospect, one could think about the relevance of choosing a ball as a domain of integration for the VCM. Only surface points yielding relevant information for orthogonal plane computation (i.e. points on the surface belonging to the plane) are interesting. The domain of integration could be defined as a structuring element depending on the local shape of the object.

References

1. Aanjaneya, M., Chazal, F., et al.: Metric graph reconstruction from noisy data. Int. J. Comput. Geom. Appl. **22**(04), 305–325 (2012)
2. Bai, X., Latecki, L.J.: Path similarity skeleton graph matching. IEEE Trans. PAMI **30**(7), 1282–1292 (2008)
3. Bauer, C., Bischof, H.: Extracting curve skeletons from gray value images for virtual endoscopy. In: Dohi, T., Sakuma, I., Liao, H. (eds.) MIAR 2008. LNCS, vol. 5128, pp. 393–402. Springer, Heidelberg (2008)
4. Coeurjolly, D., Svensson, S.: Estimation of curvature along curves with application to fibres in 3D images of paper. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 247–254. Springer, Heidelberg (2003)
5. Cornea, N.D., Silver, D., Min, P.: Curve-skeleton properties, applications, and algorithms. IEEE Trans. Vis. Comput. Graph. **13**, 530–548 (2007)
6. Cornea, N.D., Silver, D., Yuan, X., Balasubramanian, R.: Computing hierarchical curve-skeletons of 3D objects. Vis. Comput. **21**(11), 945–955 (2005)
7. Couprie, M., Coeurjolly, D., Zrour, R.: Discrete bisector function and Euclidean skeleton in 2D and 3D. Image Vis. Comput. **25**(10), 1519–1698 (2007)
8. Cuel, L., Lachaud, J.-O., Thibert, B.: Voronoi-based geometry estimator for 3D digital surfaces. In: Barcucci, E., Frosini, A., Rinaldi, S. (eds.) DGCI 2014. LNCS, vol. 8668, pp. 134–149. Springer, Heidelberg (2014)
9. Grélard, F.: January 2016. http://www.labri.fr/perso/fgrelard/research.html#skeleton
10. Grélard, F., Baldacci, F., Vialard, A., Lachaud, J.-O.: Precise cross-section estimation on tubular organs. In: Azzopardi, G., Petkov, N., Effenberg, A.O. (eds.) CAIP 2015. LNCS, vol. 9257, pp. 277–288. Springer, Heidelberg (2015). doi:10.1007/978-3-319-23117-4_24
11. Hassouna, M.S., Farag, A.A.: Variational curve skeletons using gradient vector flow. IEEE Trans. Pat. Anal. Mach. Intell. **31**(12), 2257–2274 (2009)
12. Kerautret, B., Krähenbühl, A., Debled-Rennesson, I., Lachaud, J.-O.: 3D geometric analysis of tubular objects based on surface normal accumulation. In: Murino, V., Puppo, E. (eds.) ICIAP 2015. LNCS, vol. 9279, pp. 319–331. Springer, Heidelberg (2015)
13. Mérigot, Q., Ovsjanikov, M., Guibas, L.: Voronoi-based curvature and feature estimation from point clouds. Vis. CG **17**(6), 743–756 (2011)
14. Palágyi, K., Tschirren, J., Hoffman, E., Sonka, M.: Quantitative analysis of pulmonary airway tree structures. Comput. Biol. Med. **36**(9), 974–996 (2006)
15. Postolski, M., Janaszewski, M., Kenmochi, Y., Lachaud, J.O.: Tangent estimation along 3D digital curves. In: 2012 21st International Conference on Pattern Recognition (ICPR), pp. 2079–2082. IEEE(2012)
16. Reniers, D., van Wijk, J.J., Telea, A.: Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure. IEEE Trans. Vis. Comput. Graph. **14**(2), 355–368 (2008)
17. Sobiecki, A., Jalba, A., Telea, A.: Comparison of curve and surface skeletonization methods for voxel shapes. Pattern Recogn. Lett. **47**, 147–156 (2014)