Vision, Modeling, and Visualization (2013)
Michael Bronstein, Jean Favre, and Kai Hormann (Eds.)

A tracking approach for the skeletonization of tubular parts of 3D shapes

V. Garro and A. Giachetti

Department of Computer Science, University of Verona, Italy

Abstract

In this paper we propose a new simple and efficient method to characterize shapes by segmenting their elongated parts and characterizing them with their centerlines. We call it Tubular Section Tracking, because it consists of slicing the interested volume along different directions, tracking centroids of the extracted sections with approximately constant centroid position, area and eccentricity and refining the extracted lines with a post processing step removing bad branches and centering, joining and extending the relevant ones. We show that, even using just a few slicing directions (in some cases even just three perpendicular directions), the method is able to obtain good results, approximately pose independent and that the extracted lines can be more informative on the relevant feature of the objects than the classical skeletal lines extracted as subsets of the medial axis. Estimated lines can be used to segment shapes into meaningful parts and compute useful parameters (e.g. length, diameters).

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

1. Introduction

Finding compact geometrical representations of natural objects capturing their relevant properties not depending on orientation and articulated motion is extremely important for a variety of applications like shape matching and recognition, pose estimation, anthropometric measurements and more.

One of the most common ideas applied to characterize this kind of shapes is to exploit the presence of approximately tubular parts, i.e. surface parts that could be defined as regions where a plane moving rectilinearly in the space (or changing slowly direction) would cut the surface creating sections that are closed curves with smooth variations of centroid position and area.

Object parts with this feature are, in fact, extremely common in many objects (e.g. humans, animals, trees, furniture, buildings, etc.) and need often to be automatically measured, recognized or matched using computer based techniques. The number of approximately tubular parts and the relationship between them is a peculiar characteristic of different object classes and can be, therefore exploited for shape retrieval tasks. Segmentation of tubular structures is, moreover, extremely important in medical image processing, where the interpretation of 3D voxelized volumes created by the modern imaging modalities requires often the extraction of the geometry of vessels, lungs, colon, bones, etc.

Several algorithms have been therefore proposed for this task, with different approaches depending on the original shape data nature (point cloud, polygonal mesh, binary or grayscale voxel grid) and application fields. Most of them are related to the extraction of the "curve skeleton" of the shape, e.g. a connected line graph that is a subset of the medial surface (e.g. locus of centers of spheres that are tangent to the surfaces in two points or more).

In this paper we propose to use a feature tracking approach to extract centerline paths of approximately tubular parts of 3D shapes. The basic idea is to sweep planes along different directions creating sets of uniformly spaced slices, and track along these slice series the centroids of the connected components of the object sections when they maintain similar position, area and elongation. This procedure is sufficient to detect a set of lines approximately lying on the searched centerlines and, even using a small number of directions (e.g. 3 perpendicular axes), it is possible to reconstruct from this set a complete and accurate 1D medial

DOI: 10.2312/PE.VMV.VMV13.073-080

representation for all the approximately tubular parts of the object, reasonably independent on the original object orientation. The reconstruction is performed with a simple post processing step able to join lines, remove duplicated parts, center them and extending them at the extrema.

The rest of the paper is organized as follows: Section 2 presents an overview of different approaches found in literature and able to extract tubular object sections, Section 3 motivates and describes our approach and Section 4 presents experimental results, briefly discussed in the last section.

2. Related work

A classical approach to detect tubular parts is based on the use of surface operators characterizing local diameter or curvature clustering surface points to segment approximately cylindrical and extracting their features. Mortara et al. proposed, for example, the Plumber method [MPS*04] analyzing on the surface points the behavior of shape intersections with spheres of different radii centered on them. The method has been used, for example, to segment and measure the human body. Shapira et al. similarly introduced a surface function called Shape Diameter Function [SSCO08] computed by throwing rays inside the geometry and taking smartly averaged lengths of the segments joining the point and the intersections of rays with the closest internal surface point.

Another method to extract tubular parts has been proposed by de Goes et al. [dGGV08]: they used spectral analysis and the concept of medial structures, e.g. loops equidistant to the boundaries of tubular regions.

A lot of work has been then dedicated to the extraction of the curve skeleton of a 3D shape, e.g. a continuous 1D curve representing its local "direction" and center. Despite some problems in giving a satisfactory definition of the curve skeleton, well pointed out in [CM07], several methods able to extract reasonable curve skeletons, continuous and well centered have been recently developed. An effective method based on Gradient Vector flow has been proposed in [HF07], while in [SLSK07] the skeleton is obtained on the fly while reconstructing the object with a deformable surface. Cao et al. [CTO*10] applied Laplacian-based contraction working on generalized discrete geometry data and allowing moderate amounts of missing data. A computationally fast approach based on the projection of the object on couple of stereo views has been presented in [LGS12]. Recently, two unambiguous geometrical definition of curve skeleton have been proposed together with algorithms allowing their extraction. Dey and Sun [DS06] defined the curve-skeleton as the subset of the medial axis where a function called Medial Geodesic Function (depending also on geodesic distances between surface points) can be defined and is singular. A similar approach, but defined on voxelized volumes has been used by Reniers et al. [RvWT08]. Telea and Jalba [TJ12] defined the curve skeleton as the locus of points of the medial surface of the original object that are at equal geodesic distance from at least 2 points of its border.

Medial axes (and therefore their subsets) are not usually robust against surface noise and the algorithms applied for their computation present often limitations and drawbacks as pointed out, for example, in [SYJT13]. Furthermore, they do not provide directly a partitioning of tubular parts of the object. A non trivial post processing step, finding the parts of the skeletal tree actually corresponding to the object meaningful parts is necessary, as shown, for example in [RT08, LCG09]. Curve skeletons are not exactly corresponding to the idea of "centerline" of an object. They are stable and well characterized only in approximately cylindrical parts and do not follow the complete center of the tubular part at borders or near junctions (Fig.1A). Small bumps on surfaces create centerline paths not corresponding to actual "tubes" (Fig.1B). An intuitive idea of centerline of a structure like that shown in Fig.1 is rather the straight line shown in Fig.1C.

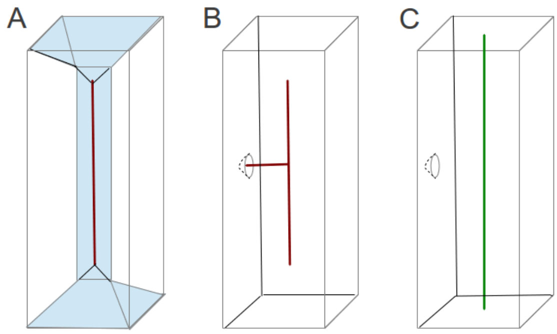

Figure 1: . *A: tubular structure with rectangular section, the medial axis includes non centered structures (light blue surface) and the curve skeleton is only a subset of the expected skeleton (C). B: Small bumps create large changes in the extracted curve skeletons, they are not expected to be found in the "ideal" skeleton (C).*

A tube, both intuitively and geometrically, is a solid shape with a constant section (in shape and area). It is reasonable, therefore, to extract tubular parts, not to rely on the geometric properties of the curve skeleton, but to this simple definition.

Obviously it is not easy to track the constant sections not knowing in advance their local centerline direction. Trying to detect from the object data the centerlines positions with some kind of multiscale vesselness analysis can be an idea, applied in some sense in [TZCO09, GL12], but can be time consuming and working only for circular sections. However, two observations can be made:

- For a tube with straight axis, if we sweep a plane along a direction even not perpendicular to the tube, and slice

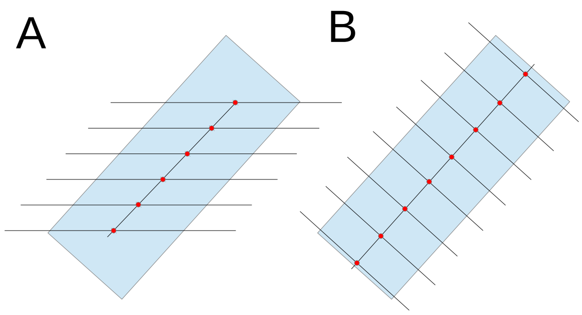

Figure 2: *A: 2D example showing that cutting the object with parallel lines and taking consecutive center points with similar sections and close position we can obtain a subset of the expected skeleton even if the lines create a large angle with its direction. B: if we now use lines perpendicular to the extracted skeleton and try to extend the skeleton at the extrema, we should be able to extract the complete skeleton.*

the tube with it at fixed distances, we obtain slices with centroids lying on the centerline of the tube (see Fig.2A).

- If sets of consecutive planes are sufficiently close, multiple sections are cut on curved tubes at distances such that the tube direction is only slightly changed. Under this condition we have sets of consecutive planes cutting approximately constant sections of the tube with centroids close to the tube centerline.

These observations can be used to design a simple skeletonization method extracting centerlines of "tubular" shapes.

3. Tracking approach to object skeletonization

The basic idea is to consider a set of different directions and sweep a plane perpendicularly to each one, computing object sections at equally spaced distances and connecting centroids of sections that are similar (in shape and position) in consecutive slices. This procedure, as discussed before, should be able to create a variable number of "centerline fragments" depending on the number of directions used and on the similarity criteria used to connect the points. These fragments can be characterized by associating to them features related to the corresponding sections (e.g. area, eccentricity). From these fragments, it is possible to reconstruct and expand lines with a subsequent post processing: a simple clustering/filtering phase, joining line fragments and rejecting bad lines, and a final refinement, centering and extending the extracted lines. The method is therefore in some sense related to algorithms based on section contouring, often applied in medical analysis [dD01], and to skeletonization methods based on Reeb Graphs [BGSF08], but, unlike previous approaches, it avoids computation of functions on surface points and allows control on sections' shape and behavior. Let us describe in detail the complete procedure.

3.1. Line fragments detection

The line fragment detection is performed computing the intersection of the object of a moving plane iteratively displaced of constant steps along each considered direction. On each slice the intersection will determine a number of connected components that can be characterized with their properties (centroid, area, elongation, etc.). At each slice location after the first one, we check if the extracted regions can be linked with those obtained at the previous position due to small centroid distance (less than a fixed percentage of the square root of the local area section) and similar area/eccentricity. We can exclude from this search sections with particular features (e.g. too big or too small area or eccentricity), using specific thresholds.

When a component extracted in the current slice is linked to one extracted in the previous one, its centroid is stored in a data structure with the associated section features. If the centroid of the previous region was already inserted in a line fragment (that is a vector of centroids), the new centroid is added to the vector, otherwise a new line fragment with the two points (previous and current centroids) is created. In the current implementation the intersection between the planes and the geometry is obtained by voxelizing the mesh on a three-dimensional grid with the desired resolution, and analyzing the resulting pixelized sections using OpenCV functions. In this way we can quickly perform the line fragment extraction in three perpendicular directions related to the three grid axes. To perform the intersection with further arbitrarily oriented sets of planes, we actually rotate the mesh of the desired angle before the discretization and apply the inverse rotation to the line fragments after the detection (potentially adding for each rotation three perpendicular tracking directions/line fragments extractions). In future implementations we will perform the discretization using the GPU pipeline.

The number of directions used is critical, in the sense that it changes the number of line fragments extracted. If we use few directions (e.g. those defined by three perpendicular axes), we can extract fragments that only partially represent the expected centerlines of the tubular parts of the object. However, we will show that, thanks to the subsequent post-processing, it is often possible, even in this case, to reconstruct the whole structure if the tolerance in accepting changes in consecutive sections is sufficiently large. This fact is shown in the example of Fig.3. Here we see a 2D "tube" with constant radius and variable direction, with a circular centerline with radius R.

If we cut a tube with horizontal lines and accept small variations of centroid position and size of the sections created, the tracking procedure generates the green line fragment in Fig.3A, a tracking of vertical sections would create the blue line fragment in Fig.3B. A clustering procedure like that described in the following subsection, searching for fragments intersection and merging longest subparts, would create a global centerline like the red one in Fig.3C. It is easy to see that the maximal error in this centerline is represented by the light red segment measuring the distance between the fragments intersection point and the true centerline.

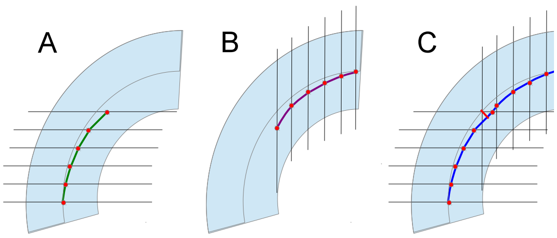

Figure 3: *If we cut a curved 2D tube with only lines perpendicular to the x and y axes, we are still able to extract the complete centerline if we tolerate a small change in the section change during the section tracking. A: Extraction of the centerline with lines perpendicular to the y-axis. Consecutive sections can be joined even with a small tolerance on centroid proximity and length difference of consecutive sections, creating the green line fragment. B: A similar procedure can create another line fragment. C: the subsequent clustering procedure would join the paths in an unique line, with a relatively small centering error (bright red segment, see text).*

3.2. Lines clustering and merging procedure

The first post-processing step is then a clustering of the extracted line fragments in sets that should correspond to different tubular part centerlines. Applying an agglomerative clustering approach we compute the minimum distance between each pairs of line fragments. If the distance is smaller than an adaptive threshold, proportional to the local section areas and if the two line have locally a similar direction, fragments are added to the same cluster. Then we apply a merging procedure to join closest pairs of line fragments that belong to the same cluster. In detail, we consider the detected link position in both the fragments and divide each fragment in two parts (see Fig.4). We reorient the line fragments in order to have positive dot product of the tangent vector in the link points, so we can compare the two half fragments before and after the joint position. If the directions of both longest half fragments are similar, so the angle between them is reasonably small we merge the fragments. The two shortest parts are checked: if their points are all approximately included in the volume defined by the sections defining the merged fragment they are completely removed.

Self-intersections are similarly found searching for couple of points of a same fragment closer than the tube radius but with far larger geodesic distance on the line. In this case a loop is created (a dedicated flag in the line structure is set) and external parts are removed (Fig.4D). The procedure is iterated until no more links can be established between line fragments of the same cluster. Lines fragments should cover maximally and with good precision centerlines of tubular parts if the number of chosen directions is high, however, it is possible to show that centerlines of tubular parts of simple objects can be traced rather well even using just three perpendicular directions.

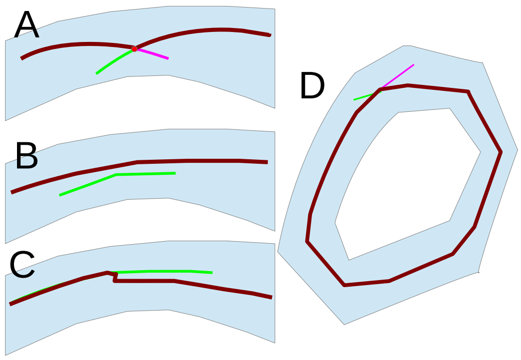

Figure 4: *Example of merging procedure. After splitting each line into two parts, the shortest parts are pruned. A: The two line fragments intersect in a point. B: One of the two line fragments is considerably smaller than the other and can be assimilated to the longest one. C: The two line fragments do not intersect but their smallest distance is below a fixed threshold. The splitting procedure will treat them as case A. D: Example of self intersection, external part are removed.*

3.3. Post processing

The set of lines obtained is then post processed in order to remove bad lines and adjust correct ones. Lines are first resampled in order to have a constant point distance. Local direction is computed along the lines using finite differences and smoothed. A local estimation of the true tubular section area and average radius is obtained throwing a set of radii (set to 16, with a constant angular sampling, in the current implementation), perpendicular to the local direction and checking intersections with the object boundaries. This estimation is used to reject lines not corresponding to an approximately tubular part according to the desired choices or to refine the branches removing, for example, bad points at the begin and at the end of the line. Currently we remove all the lines that are too short with respect to the local radius and we can remove as well (or break in subparts) lines with non uniform radius. In our implementation we also remove from the initial and final part of the lines points that are too different from the average tube size or parts with sudden increase or decrease of the radius that usually correspond to the regions where the tubular part is connected to the rest of the shape.

3.4. Centering, filtering and extension

Finally, the extracted lines are centered and extended to complete partially detected "tubes". The centering is obtained

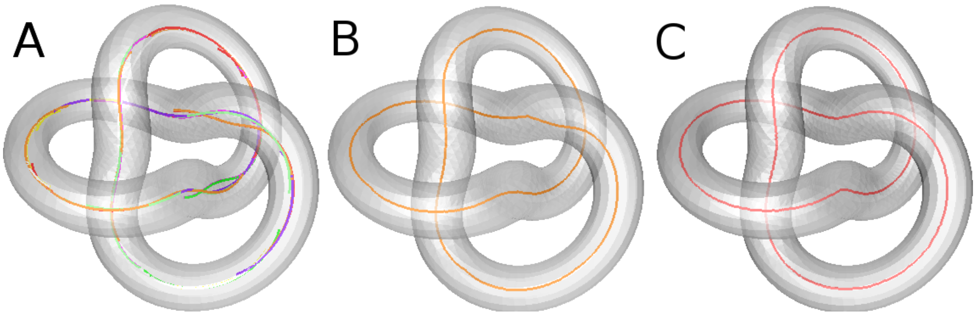

Figure 5: *The main steps of the skeletal lines extraction. A: Line fragments (shown by different colors) obtained tracking sections along nine different directions. B: Output of the clustering procedure, here a single loop. C: The centered skeleton.*

with an iterative procedure similar to those commonly used for active contour segmentation. At each iteration we recompute the local line direction for each point of the chain and throw a set of rays along sampled perpendicular directions, finding intersections with the object's surface. From these we can recompute estimated positions of the section centroids $\vec{x}_i^*$. These estimates are inserted in a snake-like equation, moving the actual line points smoothly in order to avoid bad behavior due to outliers and noise:

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \alpha(\vec{x}_i^* - \vec{x}_i(t)) + \beta(\vec{x}_{i-1}(t) - 2\vec{x}_i(t) + \vec{x}_{i+1}(t))$$

The displacement of endpoints is furtherly constrained making points move only perpendicularly to the estimated local direction in order to avoid line shrinking. The number of centering/smoothing iterations is currently fixed.

After this step, a refinement is added to adjust the endpoints position of the lines. This procedure tries to add new points along the local direction at distance s, checking if the position and feature of the new centroid differ less than the expected thresholds from the previous ones and repeating the procedure until new points can be added. This can complete tubular parts not detected due to the use of cutting planes not perpendicular to the tube direction in the fragments generation.

3.5. Selection of slicing directions and parameters

The use of sufficiently large thresholds allows to partially track curved sections even sectioning tubes with non perpendicular planes. Moreover, combined with the post processing step, it allows the method to obtain good results for simple shapes even if the centroids tracking is computed only along three perpendicular directions. Adding more directions sampling optimally the spherical angle would result in a better detection with reduced necessity of post processing and the possible use of smaller values for tracking tolerances. In our test, however, we have taken only a small number of directions, typically 9 or 13 simply adding three perpendicular directions defining the standard references the diagonals of resulting quadrants and octants. We plan to investigate the effect of optimally sampling the solid angle with a larger number of approximately equidistant directions and using smaller thresholds in the tracking step.

4. Experimental results

Fig.5 shows the effects of the different processing steps in the extraction of skeletal lines. Fig.5A shows the line fragments extracted using 9 sampled directions, Fig.5B the result of the clustering and merging procedure and Fig.5C the centered and smoothed result.

Here and in the following experiments, the plane section discretization is performed with a fixed pixel size s taken equal to a fraction (1/50) of the radius of the sphere with the same volume of the analyzed object, the same value is used for consecutive slice planes spacing. Centerlines are tracked if the distance between centroids in consecutive slices are within a distance equal to $2s$ and if the area and eccentricity differences of the consecutive sections are within $1/10$ of the average of the two. The values of α and β for the centering procedure were set using a trial and error procedure equal to 0.01 and 0.2 respectively.

The "approximately constant" section tracking clearly produces line fragments that depend on the cutting directions chosen. However, for a tubular object, a rather small number of sampled directions is enough and differences between results obtained with different cutting plane directions are negligible after the post processing. Fig.6A shows line fragments extracted from a toroidal shape tracking centroids along three perpendicular directions corresponding to the normals of the small cube superimposed. Fig.6C shows the line fragments similarly extracted using three slicing plane directions largely different from the previous ones. Fig.6B,D show the corresponding final extracted lines after clustering and merging. The result proves that for a close tubular loop the method works fine using just three directions and is almost independent on the choice of the directions.

If the shape is more complex centerlines extraction is

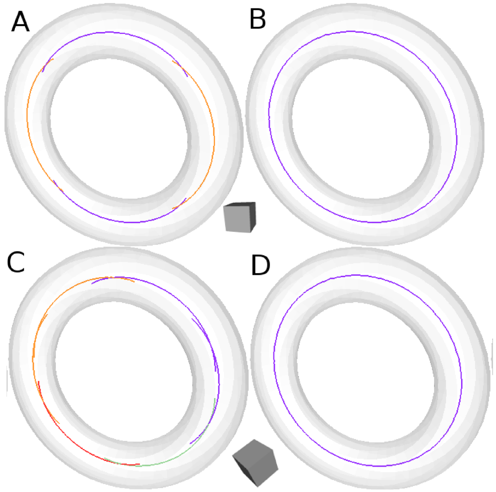

Figure 6: *A,C: Using set of planes parallel to the faces of the small cubes, differing for a large rotation, different line fragments are generated, all covering the complete torus. B,D: the corresponding results after clustering and post-processing are the same.*

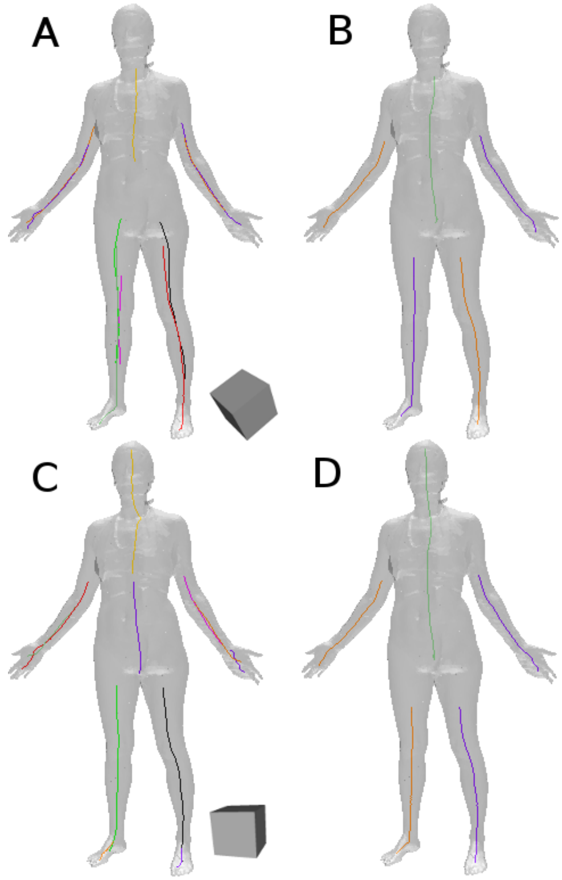

Figure 7: *A,C: In case of a more complex structure, with non uniform tubular parts, line fragments extracted tracking sections along three different mutually perpendicular set of planes (corresponding to the faces of the small cubes) are different, as expected, mainly due to missing parts and moved endpoints. B, D: Corresponding post processed lines: they are rather similar, except for the missing part in the head.*

more difficult due to the spurious structures that can be tracked at the end of tubular sections and in non-tubular parts. The sufficiently large thresholds used to detect curve tubes with a few sampled plane cutting directions creates some problems to the procedure that are still mostly fixed in the post processing phase. Fig.7A shows the line fragments extracted from a scanned human model slicing along the three perpendicular directions indicated by the small cube. Fig.7C shows the line fragments extracted from the same model slicing in the completely different directions indicated. Apart the head part missing, the extracted lines are similar. Fig.7B,D show the corresponding extracted lines after clustering and merging: the post processing improves the similarity. The results obtained in Fig.7D are rather close to those obtained with a larger set of slicing directions and detects well the symmetric part of the trunk and the limbs.

Fig.8 shows the results obtained on a series of different models with approximately tubular parts using 9 slicing directions.

Using a volume discretization as in the line extraction algorithm we can also efficiently propagate the information captured by tubular parts centerlines to the external surface mesh (doing the opposite of the approaches segmenting nodes then extracting tubes [MPS*04]). To do this, we rasterize the mesh using the same resolution used in the line tracking step and create an array of the same size for labelling. We sort the extracted lines by increasing average radius and we start label voxels corresponding to the different tubular parts as follows. Voxel including centerline points are labelled with a selected value, and immediately external points along line direction are labelled differently. A fast marching procedure is then started growing from the labelled points at constant speed up to a distance corresponding to the maximum radius of the centerline, inside the rasterized shape. Labels of "external" points are then removed and the procedure is then repeated for all the other extracted skeletal lines. The volume labelling can be easily transferred to mesh points with results like those shown in Fig.9.

4.1. Computational complexity

The complexity of the clustering step is quadratic in the in the number of points of the line fragments, while the previous steps depend on the volume discretization and can be sufficiently fast with a reasonable accuracy. Typical algorithms to extract tubular sections are quadratic in the number

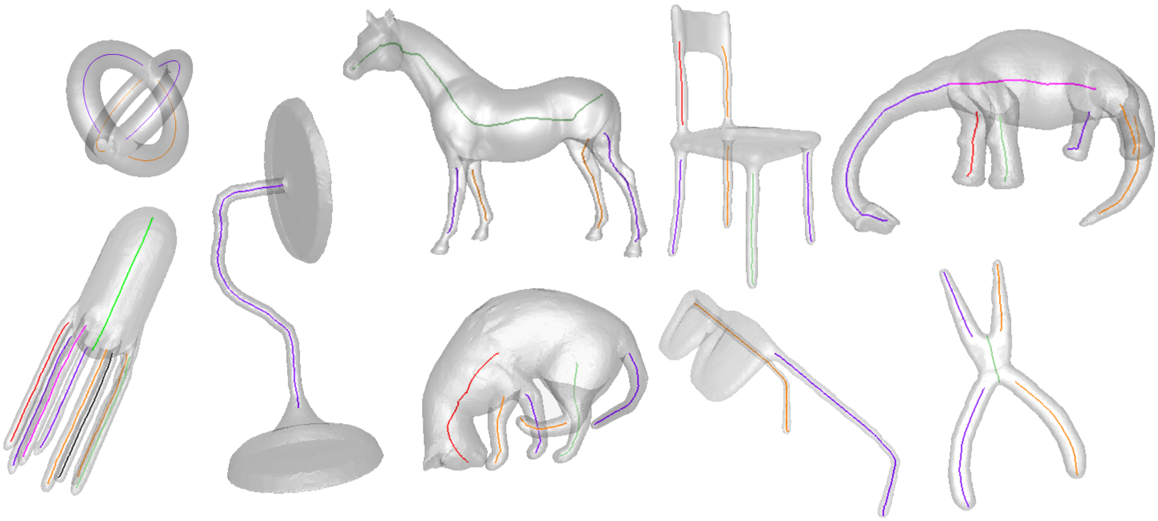

Figure 8: *Example of tubular sections centerlines extracted with the complete procedure.*

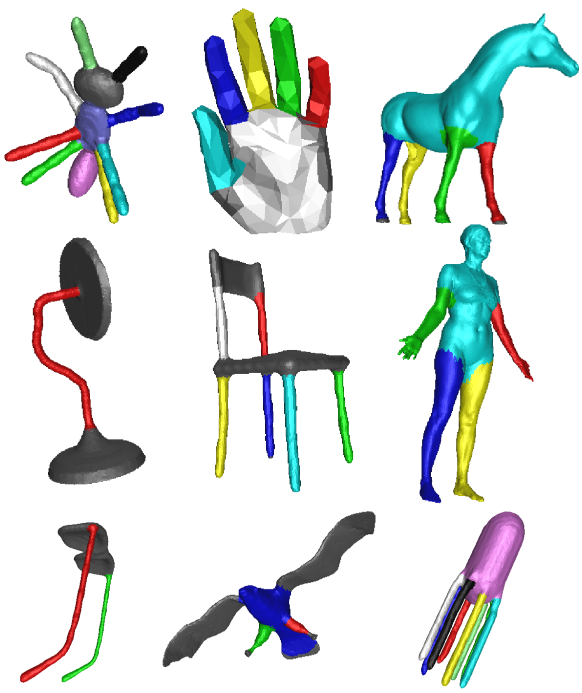

Figure 9: *Examples of tubular parts segmented on the surface by propagating the skeleton information inside the voxelized volume.*

of surface nodes and usually result in higher computation times [MPS*04].

The current CPU implementation takes a few seconds to process models of any resolution. We test the time performances of our implementation on a desktop PC with Intel Core i7 CPU (2.80 GHz, RAM 6 GB) using the models of SHREC'11 database [LGB*11] non-rigid watertight contest. Table 1 shows the average time spent for each processing step, using as slice pixel size and planes distancing value $1/50$ of the radius of a sphere with the same volume of the input mesh. Major computational bottlenecks of the implementation are slices rasterization and clustering that are directly related to the number of sample directions chosen. The slice rasterization step, that now mainly consists in rotating the mesh along the chosen direction and voxelizing it can be, however, easily and hugely speeded up by using the rasterization pipeline of graphics cards. We plan to evaluate different approaches for line clustering and merging in order better suited to deal with larger numbers of slicing directions.

5. Discussion

We presented a simple method to extract centerlines of tubular parts of shapes that is not based on the analysis of medial axis or on local surface features, but is based on the geometric definition of tube, e.g. a shape that presents a constant section if cut by a plane perpendicular to a straight or curved line. The algorithm is based on cutting the shape with sets of planes perpendicular to a few selected directions, allowing the extraction of line fragments that can be extended, joined and centered to create complete centerlines. The algorithm is sufficiently invariant against change in the sets of directions used, but depends on critical parameters (e.g. thresholds used in the tracking step, in the clustering step and in post-processing/rejection phase. This can be seen as a drawback of the method, but can be indeed a positive feature, allowing users to tune it in order to extract only selected kind of tubular parts (with strict or large tolerance on sections shape and variability). Tests on the dependency of the

	Sample directions		
	3	9	13
Sections rasterization	0.5990	1.7872	2.5394
Line fragments detection	0.1773	0.5654	0.7932
Line clustering	0.1074	1.2063	2.0899
Centering and filtering	0.1917	0.4266	0.4615
Total execution time	1.1013	4.0175	5.9907

Table 1: *Average execution time [sec] for the main steps of the algorithm choosing different number of planes sweeping directions.*

results on the different parameters are, however, planned as future work.

The method, avoiding the computation of surface parameters is rather fast and being discretized on grids is easily parallelizable. The current implementation is only a proof of concept, in future versions we plan to improve the clustering step in order to fit an optimal line through the extracted line fragments. With this choice and using a larger number of sampled cutting planes direction we could avoid the further post-processing steps and obtain a cleaner result. In the new implementation the rasterization step will be realized in the GPU pipeline. We also plan to extend the method joining centerlines to create a single skeletal tree and to make the method robust against holes. The current method is designed for watertight meshes, even if it can work on non-watertight ones by filling holes during the rasterization step with a classical signed distance method. Other more interesting ways to make the method detect centerlines of tubes also on non watertight surfaces could be obtained by tracking along the cutting planes different features instead of centroids of connected components. For example, it would be possible to track centroids of non closed lines, or maxima of 2D symmetry detectors (e.g. Fast Radial Symmetry [LZ03]).
Acknowledgements Many thanks to Marco Livesu for providing test 3D models.

References

[BGSF08] BIASOTTI S., GIORGI D., SPAGNUOLO M., FALCIDIENO B.: Reeb graphs for shape analysis and applications. *Theoretical Computer Science 392*, 1 (2008), 5–22. 3

[CM07] CORNEA N. D., MIN P.: Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics 13*, 3 (2007), 530–548. Member-Silver, Deborah. 2

[CTO*10] CAO J., TAGLIASACCHI A., OLSON M., ZHANG H., SU Z.: Point Cloud Skeletons via Laplacian Based Contraction. In *2010 Shape Modeling International Conference* (June 2010), IEEE, pp. 187–197. 2

[dD01] DE DIETRICH G.: A modular algorithm for automatic slice positioning in tubular organs. In *Proc. Int. Workshop on Medical Imaging and Augmented Reality* (2001), IEEE, pp. 163–167. 3

[dGGV08] DE GOES F., GOLDENSTEIN S., VELHO L.: A hierarchical segmentation of articulated bodies. In *SGP '08: Proceedings of the Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2008), Eurographics Association, pp. 1349–1356. 2

[DS06] DEY T. K., SUN J.: Defining and computing curve-skeletons with medial geodesic function. In *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing* (Aire-la-Ville, Switzerland, 2006), Eurographics Association, pp. 143–152. 2

[GL12] GIACHETTI A., LOVATO C.: Radial symmetry detection and shape characterization with the multiscale area projection transform. *Comp. Graph. Forum 31*, 5 (Aug. 2012), 1669–1678. 2

[HF07] HASSOUNA M. S., FARAG A. A.: On the extraction of curve skeletons using gradient vector flow. In *ICCV* (2007), pp. 1–8. 2

[LCG09] LOVATO C., CASTELLANI U., GIACHETTI A.: Automatic segmentation of scanned human body using curve skeleton analysis. In *MIRAGE '09: Proceedings of the 4th International Conference on Computer Vision/Computer Graphics CollaborationTechniques* (Berlin, Heidelberg, 2009), Springer-Verlag, pp. 34–45. 2

[LGB*11] LIAN Z., GODIL A., BUSTOS B., ET AL.: Shrec'11 track: shape retrieval on non-rigid 3d watertight meshes. In *Proceedings of the 4th Eurographics conference on 3D Object Retrieval* (Aire-la-Ville, Switzerland, Switzerland, 2011), Eurographics Association, pp. 79–88. 7

[LGS12] LIVESU M., GUGGERI F., SCATENI R.: Reconstructing the Curve-Skeletons of 3D Shapes Using the Visual Hull. *IEEE transactions on visualization and computer graphics 18* (Feb. 2012). 2

[LZ03] LOY G., ZELINSKY A.: Fast radial symmetry for detecting points of interest. *IEEE Trans. Pattern Anal. Mach. Intell. 25* (August 2003), 959–973. 8

[MPS*04] MORTARA M., PATANÈ G., SPAGNUOLO M., FALCIDIENO B., ROSSIGNAC J.: Plumber: a method for a multi-scale decomposition of 3d shapes into tubular primitives and bodies. In *Proceedings of the ninth ACM symposium on Solid modeling and applications* (Aire-la-Ville, Switzerland, Switzerland, 2004), SM '04, Eurographics Association, pp. 339–344. 2, 6, 7

[RT08] RENIERS D., TELEA A.: Hierarchical part-type segmentation using voxel-based curve skeletons. *The Visual Computer 24*, 6 (2008), 383–395. 2

[RvWT08] 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 (2008), 355–368. 2

[SLSK07] SHARF A., LEWINER T., SHAMIR A., KOBBELT L.: On-the-fly curve-skeleton computation for 3d shapes. *The Visual Computer 26*, 3 (2007). 2

[SSCO08] SHAPIRA L., SHAMIR A., COHEN-OR D.: Consistent mesh partitioning and skeletonisation using the shape diameter function. *Vis. Comput. 24*, 4 (2008), 249–259. 2

[SYJT13] SOBIECKI A., YASAN H., JALBA A., TELEA A.: Qualitative comparison of contraction-based curve skeletonization methods. In *Proc. ISMM* (2013), Springer. to appear. 2

[TJ12] TELEA A., JALBA A.: Computing curve skeletons from medial surfaces of 3d shapes. In *TPCG* (2012), pp. 99–106. 2

[TZCO09] TAGLIASACCHI A., ZHANG H., COHEN-OR D.: Curve skeleton extraction from incomplete point cloud. *ACM Trans. Graph. 28*, 3 (July 2009), 71:1–71:9. 2