Qualitative Comparison of Contraction-Based Curve Skeletonization Methods

Sobiecki, André; Yasan, Haluk C.; Jalba, Andrei C.; Telea, Alexandru C.

Published in:
EPRINTS-BOOK-TITLE

Publication date:
2013

Qualitative Comparison of Contraction-Based Curve Skeletonization Methods

André Sobiecki[1], Haluk C. Yasan[2], Andrei C. Jalba[2], and Alexandru C. Telea[1]

[1] Institute Johann Bernoulli, University of Groningen, The Netherlands
{a.sobiecki,a.c.telea}@rug.nl
[2] Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands
h.c.yasan@student.tue.nl, a.c.jalba@tue.nl

Abstract. In recent years, many new methods have been proposed for extracting curve skeletons of 3D shapes, using a mesh-contraction principle. However, it is still unclear how these methods perform with respect to each other, and with respect to earlier voxel-based skeletonization methods, from the viewpoint of certain quality criteria known from the literature. In this study, we compare six recent contraction-based curve-skeletonization methods that use a mesh representation against six accepted quality criteria, on a set of complex 3D shapes. Our results reveal previously unknown limitations of the compared methods, and link these limitations to algorithmic aspects of the studied methods.

Keywords: Curve skeletons, shape analysis, shape representation.

1 Introduction

Curve skeletons are among the most well-known, and widest used, descriptors for 3D shapes. They have been extensively used in applications such as shape matching and recognition, computer animation, virtual navigation, and shape processing [8,27].

Earlier methods for computing curve skeletons used mainly voxel-based 3D shapes. In recent years, several methods have been proposed to compute curve skeletons from meshed 3D shapes, using a *contraction* principle, where the input mesh is iteratively shrunk towards its local center. Such methods are highly computationally scalable, and can easily handle mesh shapes with considerable more details than voxel-based methods. However, their algorithmic complexity makes it harder to reason analytically about the properties of the produced skeletons. In particular, it is not fully clear how their results relate to desirable skeleton properties. Moreover, since such methods are typically compared with methods in the same class (mesh-based), it is unclear whether mesh-based methods are indeed always superior to voxel-based methods.

In this paper, we compare six mesh-contraction-based curve-skeletonization methods, all which are based on a collapse principle, against six accepted quality criteria: centeredness, homotopy to the input shape, invariance under isometric

C.L. Luengo Hendriks, G. Borgefors, R. Strand (Eds.): ISMM 2013, LNCS 7883, pp. 425–439, 2013.

transformations, detail preservation, smoothness, and independence from the input shape's sampling. Our work extends the earlier survey of Cornea *et al.* [8] by studying six mesh-based curve-skeletonization algorithms published after that survey was done. Our results reveal several limitations of the studied methods which, to our knowledge, have not been highlighted in the literature, and link these to algorithmic aspects of the studied methods.

The structure of this paper is as follows. Section 2 overviews related work in curve skeletonization, with a focus on contraction-based methods. Section 3 details the quality criteria used for the comparison. Section 4 presents the comparison results. Section 5 discusses our findings. Section 6 concludes the paper with future work directions.

2 Related Work

For a shape $\Omega \subset \mathbb{R}^3$ with boundary $\partial\Omega$, we first define its distance transform $DT_{\partial\Omega} : \mathbb{R}^3 \to \mathbb{R}^+$

$$DT_{\partial\Omega}(\mathbf{x} \in \Omega) = \min_{\mathbf{y}\in\partial\Omega} \|\mathbf{x} - \mathbf{y}\|. \tag{1}$$

The surface skeleton of Ω is next defined as

$$S(\Omega) = \{\mathbf{x} \in \Omega \,|\, \exists\, \mathbf{f}_1, \mathbf{f}_2 \in \partial\Omega, \mathbf{f}_1 \neq \mathbf{f_2}, \|\mathbf{x} - \mathbf{f}_1\| = \|\mathbf{x} - \mathbf{f}_2\| = DT_{\partial\Omega}(\mathbf{x})\} \tag{2}$$

where $\mathbf{f}_1$ and $\mathbf{f}_2$ are the contact points with $\partial\Omega$ of the maximally-inscribed ball in Ω centered at $\mathbf{x}$ [12,24], also called *feature transform* (FT) points [15]. Surface skeletons consist of several manifolds with boundaries which meet along a set of Y-intersection curves [9,17,7]. They can be computed by voxel-based or mesh-based methods [21,5,28,14,3,14]. A recent comparison of surface-skeleton extraction methods is given in [15].

In contrast to surface skeletons, curve skeletons are loosely defined as 1D structures "locally centered" within the input shape Ω. The lack of a unanimously accepted formal definition has led to many methods which compute curve skeletons following not necessarily identical definitions. This makes it hard to analytically compare, and reason about, the properties of the produced curve skeletons.

Tools from mathematical morphology [25] were among the first used to compute curve skeletons: The residue of openings, based on Lantuéjoul's formula [16], usually leads to disconnected skeleton branches, whereas methods based on homotopic thinning transformations [16,19,4,21] yield connected skeletons. Combining such techniques with distance-driven thinning further ensures the extraction of centered surface and curve skeletons [1].

Dey and Sun propose one of the first analytic definitions of curve skeletons based on the medial geodesic function (MGF), where the curve skeleton is defined as the locus of points having at least two equal-length shortest geodesics on $\partial\Omega$ between their feature points [10,23]. Reniers *et al.* extend the MGF to regularize curve skeletons by assigning each skeleton point an importance equal to the area bounded by such geodesics, in a voxel setting [24], inspired by the so-called 2D

collapse metric [20,32]. A GPU implementation of the above metric for mesh models is presented in [15].

Voxel-based methods typically require significant resources to store and process the large voxel volumes required to capture the fine details of complex 3D shapes. To be used on 3D meshes, such methods require a costly voxelization step. Mesh-based methods address these cost issues by working directly on a mesh representation of $\partial\Omega$. In recent years, several such mesh-based methods have been proposed based on a *contraction* principle, which shrinks the input mesh until the 1D curve-skeleton structure is reached, as follows. Au *et al.* shrink the mesh via Laplacian smoothing until its volume gets close to zero, followed by an edge-collapse (to extract the 1D curve skeleton) and a re-centering step (to correct shrinking errors) [2]. Cao *et al.* extend this idea to extract curve skeletons from incomplete point clouds [6]. The ROSA method defines, and extracts, curve skeletons using rotational, rather than positional, symmetry: $\partial\Omega$ is cut with planes, and curve-skeleton points are found as the centers of planes which minimize the variance between the plane's normal and $\partial\Omega$ normals along the cut curve [30]. Sharf *et al.* reverse the contraction direction: They find the curve skeleton as the centers of a set of competing fronts which evolve to approximate the input surface [26]. A similar method is presented by Hassouna and Farag [13]. Telea and Jalba define, and extract, curve-skeletons by contracting the surface skeleton $S(\Omega)$ (computed as in [18]) inwards, along the gradient of the 2D distance transform of $\partial S(\Omega)$, *i.e.* define the curve-skeleton as the result of a two-step skeletonization [31].

Mesh-contraction methods are currently deemed to be the state-of-the-art for extracting detailed curve skeletons from high-resolution shapes [29]. As 3D models become more complex, it is arguable that such methods will dominate the more costly voxel-based methods. Conceptually, such methods work very similarly to voxel-based thinning. However, there are few, if any, comparisons of contraction-based methods based on the accepted skeleton desirable criteria used for earlier voxel-based methods. Also, the algorithmic complexity of mesh-contraction methods makes a formal analysis thereof more complex than for voxel-based methods. All in all, it is not clear if mesh-contraction methods are indeed always superior to voxel-based methods, and if not, which are their specific weak points with respect to desirable skeleton criteria.

3 Comparison Criteria

The literature knows a well-accepted set of quality criteria that curve skeletons should conform to. For curve-skeletonization methods, such criteria are significantly more important than for surface skeletonization methods: While the latter can be rigorously checked against the formal surface skeleton definition (Eqn. 2), the former do not use a single curve-skeleton definition. As such, the only comparison available for curve skeletons is a qualitative one, from the perspective of desirable quality criteria. Following [8,15,27], we focus on the following generally-accepted quality criteria for a curve skeleton:

Homotopy: The curve skeleton is topologically-equivalent with the input shape, *i.e.* has the same number of connected components and tunnels.

Invariant: The curve skeleton should be invariant under isometric transformations of the input shape.

Thin: The curve skeleton should be as thin as the sampling model used allows it. Voxel-based curve skeletons should be one voxel thick. Mesh-based curve skeletons should contain only lines, and not polygons or loose points. Point-cloud based curve skeletons should ideally have zero local thickness in any direction orthogonal to the largest eigenvector of the covariance matrix of point neighborhoods.

Centered: This is the hardest criterion to quantify, since it is not uniquely defined when a curve is centered within a 3D shape. However, several weak forms of curve-skeleton centeredness exist: The curve skeleton should be a subset of the surface skeleton (since the latter is by definition centered within the shape); and in no case should the curve-skeleton exit the input shape.

Smoothness: As centeredness, smoothness is also hard to formally define. Surface skeleton manifolds are known to be at least $\mathcal{C}^2$ continuous [22,27]. Curve-skeletons are centered subsets thereof [29,31]. Hence, it is arguable that curve skeletons should be also piecewise, *i.e.* per branch, $\mathcal{C}^2$. In any case, curve skeletons should not exhibit curvature discontinuities induced by the sampling of either the input surface or curve skeleton representation.

Detail Preserving: Curve skeletons should be able to capture fine-scale details, such as bumps, of the input shape, in a user-controlled manner. In other words, the user should be able to select the scale of input shape details which the curve skeleton should capture (being significant) and the scale of details to ignore (being regarded as noise).

Sampling Robustness: Given two different samplings of an input shape (*e.g.* two different level-of-detail meshes), the difference between the two corresponding curve skeletons should be proportional with the difference of the two input meshes. In other words, small input-sampling differences should not cause large differences in the curve skeleton.

4 Comparison

Given our core question on how curve-skeletonization methods perform, we compared six such methods (further denoted in the paper by the abbreviations listed below):

Au *et al.* (AU) [2]: We included this method as it is arguably the best-known mesh-based skeletonization technique in existence [13,29,15].

Tagliasacchi *et al.* (ROSA) [30]: We chose this method given its advocated noise-resistance and since it works on point clouds, which is a different type of input than the other methods.

Cao *et al.* (CAO) [6]: We chose this method since it uses a contraction similar to [2], but works on point clouds, like [30].

Telea and Jalba (TJ) [31]: In contrast to all other curve-skeletonization methods, this technique contracts the surface skeleton, rather than the input mesh, to compute the curve skeleton. It produces a point cloud rather than a polyline curve-skeleton. For comparison fairness, we postprocessed the produced point cloud using the polyline reconstruction proposed in [2].

We also developed and tested two extensions of [2], as follows.

Au *et al.* Improved (AUI): A well-known limitation of Au *et al.* is its skeleton re-centering step [29]. As the input mesh is contracted, it can go off-center due to numerical and discretization inaccuracies of the Laplacian smoothing. To address this issue, we proceed as follows. During the Laplacian contraction and edge-collapse steps of the method, we maintain a backwards, skeleton-node-to-mesh-vertex mapping $\Pi : S \rightarrow \partial\Omega$, which can be used to identify those mesh vertices $\mathbf{v} \in \partial\Omega$ that 'collapsed' into a given skeleton node $\mathbf{s} \in S(\Omega)$. The re-centering step uses Π to compute the final position of each node $\mathbf{s}$ as a weighted average of the vertices in $\Pi(\mathbf{s})$, with weights given by the areas of the input-mesh triangles with vertices in $\Pi(\mathbf{s})$.

Au *et al.* Using Surface Skeletons (AUS): The improved re-centering outlined above cannot fully correct errors accumulated during the iterative contraction. To further reduce these, we start the Laplacian contraction from the surface skeleton, which is closer to the final target (curve skeleton) than the input mesh, along the idea proposed in [31].

Global Considerations: In our method choice, we focused on recent contraction-based techniques, not studied in the survey of Cornea *et al.* [8], proven by their authors on complex shapes, and which use different curve-skeleton detection principles. All studied methods satisfy the *invariance* criterion by construction, since they work in 3D vector space. All methods also directly satisfy the *thinness* criterion, since they model the curve-skeleton as a polyline. We used the original implementations provided by their authors, all running on a Windows PC with 4 GB RAM. Since not all studied methods claim computational efficiency, we excluded timings from the comparison.

Comparison Material: For comparison, we used a set of 21 3D shapes which are frequently encountered in the curve-skeleton literature (for details, see [34]). Figures 1, 2, 3 and 4 and show relevant samples from this set, within space

limitations. The models have between 20K and 300K vertices. We used MeshLab [33] to clean mesh models for normal orientation consistency, T-vertices, and duplicate vertices. To factor our parameter settings, we ran each method for uniformly-sampled values of all its documented parameters, and retained in our final comparisons the best results with respect to the quality criteria mentioned in Sec. 3.

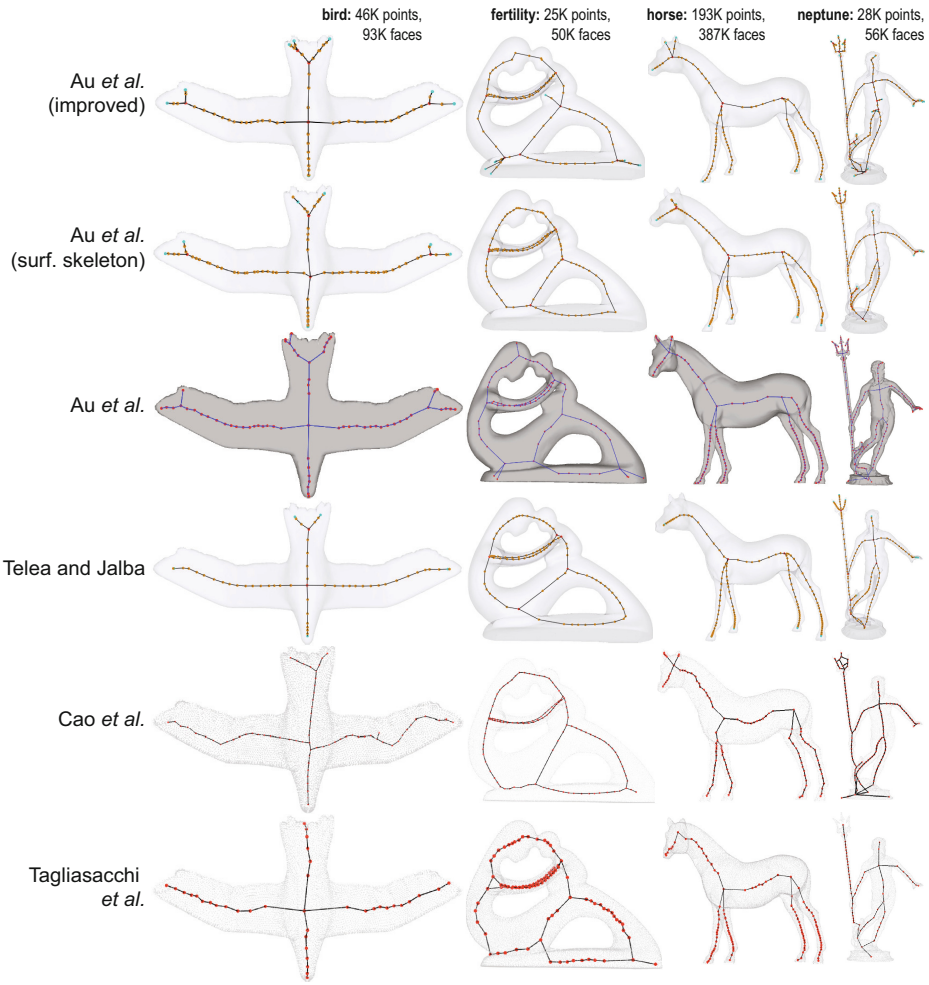

Fig. 1. Overview comparison of skeletonization methods

4.1 Overview

Figure 1 shows an overview of several curve skeletons extracted by the compared methods. Even at this level, we quickly notice that not all skeletons are equally well centered, equally smooth, and have the same number of terminal (detail) branches. We next zoom-in on each criterion and discuss our findings with respect to the studied methods.

4.2 Homotopy

For relatively simple shapes of genus 0 or higher, all studied methods behaved equally well, *i.e.* produced curve skeletons homotopic with the input shape (Fig. 1). This is due to the fact that all these methods start by contracting the input shape *and* change only the geometry, but not the topology, of this shape during the iterative contraction process. Still, detail differences exist between these methods. Skeleton junctions are not always identical, so the produced skeleton graph is different, see *e.g.* the marked limbs-to-body junctions of the *bird* model in Fig. 2 (left) and the *horse* model in Fig. 3 (right). Differences get larger for small-scale details, where curve skeleton terminal branches enter saliencies of the input shape, see *e.g.* Fig. 3 (*neptune*, *frog*). An extreme case happens when the input mesh has self-intersections, *e.g.* Fig. 2 (*frog*). Here, CAO and ROSA create curve skeletons whose topology is far from the input shape (fake loops and branches).

4.3 Centeredness

The methods AU, AUI, and AUS produce similar, well centered, results. Among these, AUS is the best: Since contraction starts from the surface skeleton, nodes go less off-center, as the surface skeleton is already centered by definition and closer to the curve skeleton than the input mesh. For mesh-based methods, TJ produced the best centering. This is due to the fact that TJ contracts the surface skeleton along the gradient field of its 2D distance transform, which is by definition tangent to the surface skeleton itself, so the curve skeleton stays inside the surface skeleton by construction. In contrast, AU, AUI, and AUS contract in the direction of the shrunken surface's normals. These are delicate to estimate as the shape shrinks and develops singularities (creases). The different re-centering steps performed by these methods alleviate, but cannot fully correct, these problems.

ROSA's results are quite poorly centered in several areas. As mentioned in [30], orientation information is unreliable around junctions, where the input shape has many points with diverse orientations. To overcome this, ROSA treats junctions specially. This works well for junctions whose branches correspond to tubular shape parts of similar size. However, we discovered that junctions where shape parts of very different sizes and shapes meet create problems, see *e.g.* Fig. 2 for the *bird* model (wings joining rump) and *neptune* (arm-torso junction).

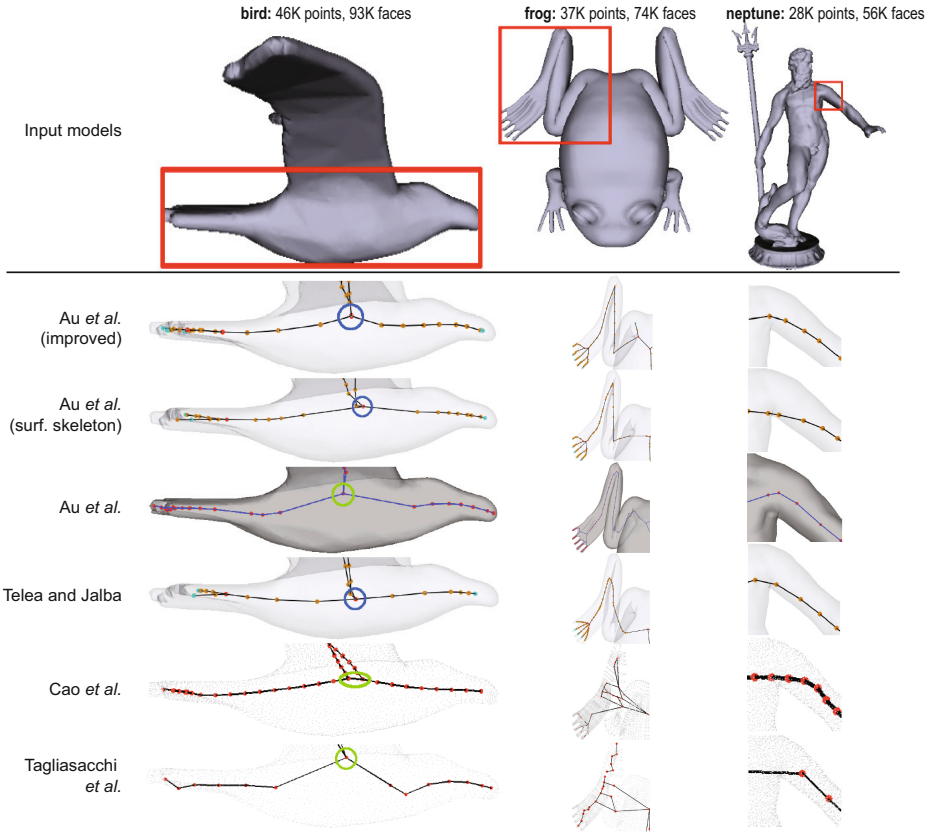

Fig. 2. Centeredness comparison. Details show areas marked by red insets.

The *frog* model (Fig. 2) reveals two other challenges. First, the model has several very sharp bends around the leg joint. Secondly, in the same area, the mesh has several self-intersections. Meshless methods (CAO, ROSA) generate seriously erroneous skeletons here, and even skeleton disconnections. In these areas, TJ still creates a smooth skeleton, but cannot handle centeredness perfectly. This is due to the fact that the surface skeleton it starts from has errors in self-intersecting areas, since the technique used to compute it [15] cannot handle self-intersecting surfaces. In contrast, AU, AUI, and AUS generate very similar, relatively well-centered, skeletons in these challenging areas.

The *neptune* model (Fig. 2) highlights the situation where a relatively thin object part (arm) joins a thick one (torso). In such areas, curve (and surface) skeletons exhibit so-called ligature branches which connect the skeleton branches of the two parts [22]. If the two parts form an angle different from 90°, like in our case, the ligature branch has to rapidly turn [27]. This turn is best captured by AU. In contrast, all other methods emphasize smoothness too much, which results in clearly off-centered skeletons close to the armpit.

4.4 Detail Preservation

Detail preservation refers to the generation of separate curve-skeleton terminal branches for all input shape bumps, or salient convexities, at a user-specified scale. Detail preservation is important for applications such as shape matching, retrieval, and reconstruction [8,24]. Large details, such as the limbs of shapes in Fig. 1, are well captured by skeleton branches by all studied methods. For smaller-scale details, the situation is different, see Fig. 3 left. The problem is that all studied methods include explicit actions to smooth the computed skeletons. Although desirable (see next Sec. 4.5), such smoothing will remove some small-scale branches.

AU and AUI preserve small-scale, detail, branches best. In contrast, AUS and TJ find detail branches of long protrusions (*e.g.* Fig. 3, *neptune* and *frog* fingers) quite well, but fail to find branches for shallower bumps, such as *gargoyle*'s wing-tips. Upon closer analysis, we found that this is caused by the fact that the surface skeletons that both AUS and TJ start from, fail to capture such details. Hence, these details cannot appear further in the curve skeleton.

CAO and ROSA perform the worst for this criterium. These methods fail finding most detail skeleton branches found by the other studied methods. Moreover, when found, small-scale terminal skeleton branches seem to be arbitrary, as Fig. 3 shows for all three models on the left.

Small-scale noise is ignored equally well by all methods. For all the studies mesh-based methods, this is an effect to their built-in smoothing, which appears to work well at small scale.

4.5 Smoothness

As outlined earlier, curve-skeleton branches should be at least $\mathcal{C}^2$ continuous curves (Sec. 3). Hence, skeletonization methods should follow this property as well as possible. Voxel methods are inherently constrained here by the sampling resolution. In contrast, mesh-based methods which model the curve skeleton as a polyline should distribute the computed skeletal points, or sample the skeleton, to optimally approximate the desired smooth curve. Hence, for these methods, the issue of skeleton smoothness is implicitly connected to the skeletal curve sampling.

Contraction-based methods, as the ones we studied, have an additional challenge here. As the input mesh is contracted, the local point density naturally increases in convex areas and decreases in concave ones. This potentially leaves too few nodes to approximate well the curve skeleton in concave areas. Ligature branches are an extreme case hereof. An example are the ligature branches that connect the horse's leg-skeletons to its rump-skeleton (Fig. 3 right). Here, CAO, ROSA, and up to some extent AU, clearly show a lower point density – see branches meeting at the marked junctions. This in turn creates spurious kinks in the rump's curve skeleton. In contrast, AUS, AUI, and TJ create smoother skeletons. The skeletons of TJ and AUS follow the rump's curvature best. This is explained by the fact that their contraction is constrained to stay on

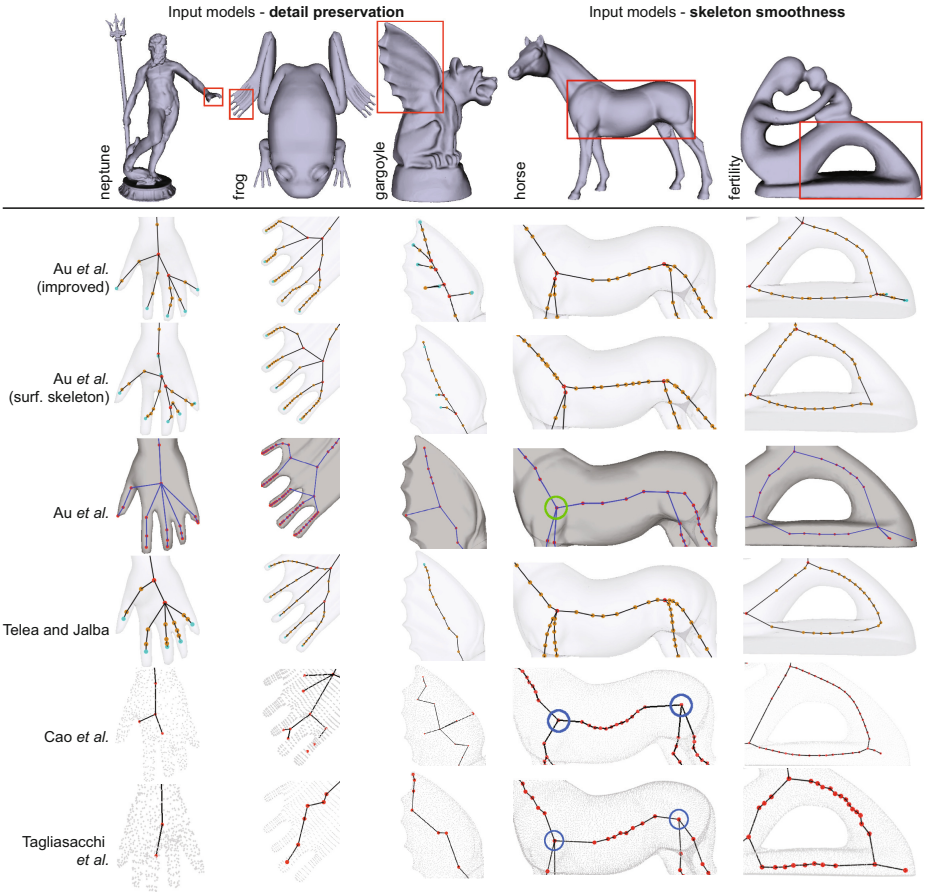

Fig. 3. Comparison for detail preservation comparison (left) and skeleton smoothness (right)

the surface skeleton, whose shape already captures the input shape's curvature. AU and AUI both fail capturing the rump's curvature, since they have no such constraint. The same non-uniform skeletal point distribution is also observed for the *fertility* model (Fig. 3 right). Here, again, AUS and AUI yield the most uniform point distribution, and ROSA and AU the least uniform one (which leads to unnatural kinks).

4.6 Sampling Robustness

Sampling robustness refers to the relation between the resolution of the input shape and changes in its curve skeleton. Ideally, we would like that when the former changes slightly, the curve skeleton also changes only slightly. This property

is closely related to the concept of *regularization*, which states that small changes in the input shape Ω should only yield small changes in its skeleton [32,24,15].

To study this, we produced three versions of the *dragon* model (see Fig. 4), using the Yams mesh resampling tool [11]. Next, we ran the studied skeletonization methods on these datasets, and analyzed the results. In the comparison, we had to exclude CAO and ROSA, as the provided implementations of these methods were too slow to complete, even in several hours, for the largest-resolution meshes.

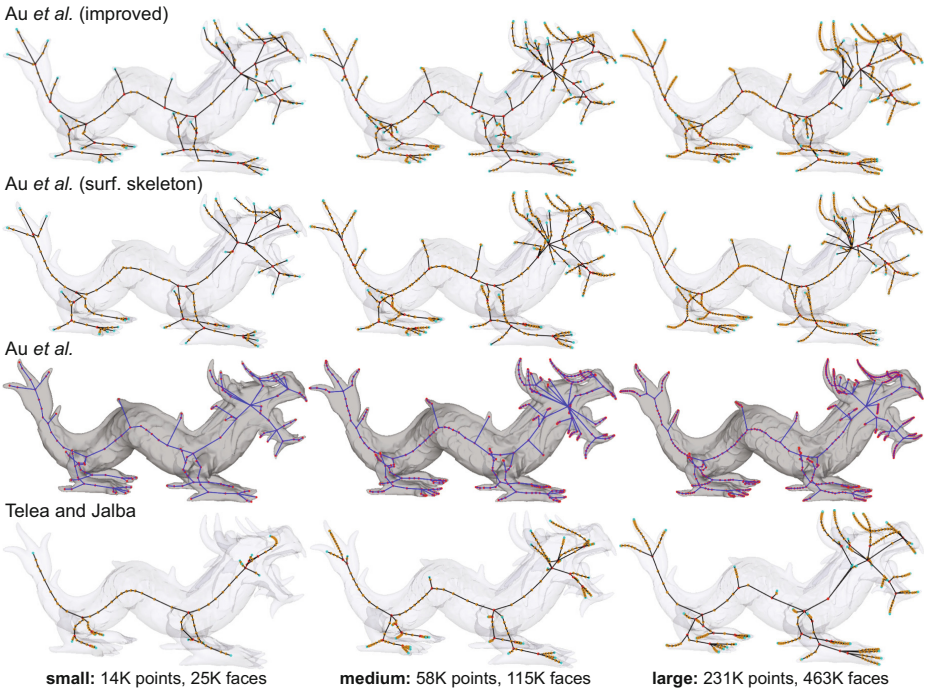

Fig. 4. Sampling robustness comparison

The method AU is quite sensitive to the mesh sampling. Looking at Fig. 4, we see that, in the dragon head area, the small and large resolution models produce relatively similar skeletons, but the medium-resolution model yields a very different skeleton topology. Given that higher resolution can only potentially add extra details, but not remove existing ones, we expect to get an increasingly rich curve skeleton (in terms of terminal branches), but the core structure of this skeleton should not change significantly. This is not the case, which hints to an important instability of the method with respect to mesh resolution.

In contrast, AUS and AUI show a much stabler curve skeleton with respect to mesh resolution. Although these methods do not produce identical skeletons for the same resolution, the changes of their respective skeletons as the resolution

changes, are quite small. Both methods find more terminal skeleton branches as the resolution increases, which is expected since higher-resolution models capture more surface details.

The TJ method is the most sensitive to sampling. For the low-resolution model, the method simply fails to extract many significant branches. Although more branches are found for the high resolution model, many significant surface details, like the upper spikes on the back and tail, fail to generate branches. This can be directly traced to the quality of the surface skeleton: The underlying method used to compute it [18] produces as many skeleton points as surface points. To accurately capture the surface skeletal structure, very densely-sampled models are required [15]. Less densely sampled surface skeletons will in turn create a noisy distance-transform gradient, which will contract the skeleton mesh in the wrong directions.

5 Discussion

Contrary to our initial belief, based on the studied contraction-based skeletonization literature, all contraction methods studied here appeared to be much more sensitive in terms of *all* studied quality criteria (except homotopy) than implied by the examples in the literature. The CAO and ROSA methods performed significantly under expectations. The AU method performed relatively well for smooth shapes, but showed limitations for centeredness and smoothness for more complex shapes. This is the main reason for us having designed the two improved variants AUI and AUS. The trade-off between these variants is as follows: While AUS yields smoother skeletons, AUI delivers a better centeredness. The TJ method dominates all others in terms of smoothness, but has clear centeredness problems in ligature areas, and requires a very high input mesh sampling to generate even moderately-detail skeleton branches, due to its usage of the surface skeleton.

A key question is whether voxel-based skeletonization methods can overcome the above limitations. Although answering this deserves a separate study, we outline below several observations in this respect:

Homotopy: Voxel-based skeletonization methods are not, by definition, homotopy preserving. For example, Reniers *et al.* can occasionally create small disconnected components [24]. However, thinning methods can enforce homotopy relatively easily [21,1].

Invariance: Like for homotopy, voxel methods are not invariant under isometric transformations by construction, as mesh-based methods are. Using truly Euclidean distance transforms helps invariance [14], but does not guarantee it [27].

Thinness: For voxel methods, this criterion translates to creating one-voxel-thin skeletal manifolds and curve skeletons. Thinning methods are best suited to enforce thinness [21,1], whereas general-field methods cannot guarantee it [14,24].

Applying a thinning postprocessing step (in line with the former methods) to general-field methods appears to be the optimal combination.

Centeredness: Just as thinness, centeredness for voxel methods is limited by sampling resolution. Using an Euclidean distance transform to drive thinning [1] or, alternatively, to find the skeleton by directly applying Eqn. 2[10,24], guarantees thinness under this sampling limitation. In contrast, all contraction methods studied here do not use a volumetric distance transform. Hence, their centeredness is subject to accumulated errors during the iterative contraction process, which arguably makes them less accurate than voxel methods.

Detail preservation: This requirement is not fundamentally linked to the type of object discretization (mesh or voxel-based). Using a global importance metric can guarantee detail preservation for both mesh [15], voxel [24], or hybrid [10] representations.

Smoothness: Voxel models can be preferable to mesh contraction models here. Indeed, while the former typically treat each skeleton voxel separately, the latter enforce local constraints on the surface and curve skeletons. This forces mesh-based skeletons to be either smooth (but not well centered), or well centered (but not smooth).

An additional desirable property of skeletons is *reconstructability*, *i.e.*, the ability to reconstruct the input shape from its skeleton. Surface skeletons should obey this property, by definition, as the medial axis transform (MAT) is a dual of the input shape [27]. Curve skeletons can obey this property only partially. Although reconstructability is sometimes studied as a separate property [8,1], it can be traced directly to a combination of centeredness and detail preservation.

The main challenge we find for voxel methods is, however, scalability: Voxelizing complex meshes to resolutions over 1000^3 voxels, and further processing such volumes to extract curve or surface skeletons, is much slower, and more memory demanding, than using mesh-based methods. For instance, a highly optimized parallel implementation of [24] processes the 700^3 *dragon* model (Fig. 4) in around 15 minutes; the equivalent mesh model (463K faces) is processed in under a minute by all studied mesh-based methods. Moreover, the memory consumption of voxel methods is at least an order of magnitude larger than for mesh-based methods. If efficient data representation and GPU parallelization schemes were designed to reduce this overhead, voxel-based methods may in the end be a very strong competitor to mesh-based methods.

6 Conclusions

In this paper, we have presented a qualitative comparison of six contraction-based curve-skeletonization methods that use a mesh representation of the input shape to be skeletonized. The methods were compared from the perspective of

several accepted quality criteria: homotopy, thinness, centeredness, detail preservation, smoothness, and robustness to sampling. In contrast to recent insights from the mesh skeletonization literature, the studied mesh-based methods appeared to perform less optimal than expected.

Although our comparison is far from exhaustive, it raises a number of important points about the current state of mesh-based curve skeletonization techniques. First and foremost, the question is raised whether such methods can outclass earlier voxel-based skeletonization methods (if we ignore computational resources). A more critical more critical quantitative and qualitative evaluation of such algorithms against each other and also against voxel-based skeletonization methods is needed to answer this question. Finally, we believe that our comparison will generate increased attention towards the development of efficient algorithms that exploit the desirable properties of voxel-based skeletonization techniques.

References

1. Arcelli, C., Sanniti di Baja, G., Serino, L.: Distance-driven skeletonization in voxel images. IEEE TPAMI 33(4), 709–720 (2011)
2. Au, O., Tai, C., Chu, H., Cohen-Or, D., Lee, T.: Skeleton extraction by mesh contraction. Proc. ACM SIGGRAPH, 44:1–44:10 (2008)
3. Bai, X., Latecki, L.J., Liu, W.-Y.: Skeleton pruning by contour partitioning with discrete curve evolution. IEEE TPAMI 3(29), 449–462 (2007)
4. Beucher, S.: Digital skeletons in Euclidean and geodesic spaces. Signal Processing 38(1), 127–141 (1994)
5. Bouix, S., Siddiqi, K., Tannenbaum, A.: Flux driven automatic centerline extraction. Med. Imag. Anal. 9(3), 209–221 (2005)
6. Cao, J., Tagliasacchi, A., Olson, M., Zhang, H., Su, Z.: Point cloud skeletons via laplacian based contraction. In: Proc. SMI, pp. 187–197 (2010)
7. Chang, M., Leymarie, F., Kimia, B.: Surface reconstruction from point clouds by transforming the medial scaffold. CVIU (113), 1130–1146 (2009)
8. Cornea, N., Silver, D., Min, P.: Curve-skeleton properties, applications, and algorithms. IEEE TVCG 13(3), 530–548 (2006)
9. Damon, J.: Global medial structure of regions in $\mathbf{R}^3$. Geometry and Topology 10, 2385–2429 (2006)
10. Dey, T., Sun, J.: Defining and computing curve-skeletons with medial geodesic function. In: Proc. SGP, pp. 143–152 (2006)
11. Frey, P.: YAMS: a fully automatic adaptive isotropic surface remeshing procedure. tech. rep. 0252, INRIA (November 2001), `http://www.ann.jussieu.fr/frey`
12. Giblin, P., Kimia, B.: A formal classification of 3D medial axis points and their local geometry. IEEE TPAMI 26(2), 238–251 (2004)
13. Hassouna, M., Farag, A.: Variational curve skeletons using gradient vector flow. IEEE TPAMI 31(12), 2257–2274 (2009)
14. Hesselink, W., Roerdink, J.: Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform. IEEE TPAMI 30(12), 2204–2217 (2008)
15. Jalba, A., Kustra, J., Telea, A.: Surface and curve skeletonization of large meshes on the GPU. IEEE TPAMI (2012), doi:10.1109/TPAMI.2012.212

16. Lantuéjoul, C.: La squelettisation et son application aux mesures topologiques de mosaiques polycristallines. PhD thesis, School of Mines, Paris (1979)
17. Leymarie, F., Kimia, B.: The medial scaffold of 3D unorganized point clouds. IEEE TVCG 29(2), 313–330 (2007)
18. Ma, J., Bae, S., Choi, S.: 3D medial axis point approximation using nearest neighbors and the normal field. Vis. Comput. 28(1), 7–19 (2012)
19. Meyer, F.: Skeletons and perceptual graphs. Signal Processing 16(4), 335–363 (1989)
20. Ogniewicz, R.L., Kubler, O.: Hierarchic Voronoi skeletons. Patt. Recog. (28), 343–359 (1995)
21. Palágyi, K., Kuba, A.: Directional 3D thinning using 8 subiterations. In: Bertrand, G., Couprie, M., Perroton, L. (eds.) DGCI 1999. LNCS, vol. 1568, pp. 325–336. Springer, Heidelberg (1999)
22. Pizer, S., Siddiqi, K., Szekely, G., Damon, J., Zucker, S.: Multiscale medial loci and their properties. IJCV 55(2-3), 155–179 (2003)
23. Prohaska, S., Hege, H.C.: Fast visualization of plane-like structures in voxel data. In: Proc. IEEE Visualization, pp. 29–36 (2002)
24. Reniers, D., van Wijk, J.J., Telea, A.: Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure. IEEE TVCG 14(2), 355–368 (2008)
25. Serra, J.: Image Analysis and Mathematical Morphology. Academic Press, London (1982)
26. Sharf, A., Lewiner, T., Shamir, A., Kobbelt, L.: On-the-fly curve-skeleton computation for 3d shapes. CGF 26(3), 323–328 (2007)
27. Siddiqi, K., Pizer, S.: Medial Representations: Mathematics, Algorithms and Applications. Springer (2009)
28. Sud, A., Foskey, M., Manocha, D.: Homotopy-preserving medial axis simplification. In: Proc. SPM, pp. 103–110 (2005)
29. Tagliasacchi, A., Alhashim, I., Olson, M., Zhang, H.: Mean curvature skeletons. CGF 31(5), 1735–1744 (2012)
30. Tagliasacchi, A., Zhang, H., Cohen-Or, D.: Curve skeleton extraction from incomplete point clouds. ACM TOG 28(3), 71:1–71:9 (2009)
31. Telea, A., Jalba, A.: Computing curve skeletons from medial surfaces of 3D shapes. In: Proc. TPCG, pp. 137–145 (2012)
32. Telea, A., van Wijk, J.J.: An augmented fast marching method for computing skeletons and centerlines. In: Proc. VisSym, pp. 251–259 (2002)
33. Univ. of Pisa. MeshLab mesh processing toolkit (2012), `http://meshlab.sourceforge.net`
34. Yasan, H.: Contraction-based curve skeletonization of 3D meshes, MSc thesis, Dept. of Computer Science, TU Eindhoven, the Netherlands (2012), `http://repository.tue.nl/741246`