The Visual Computer manuscript No.
(will be inserted by the editor)

Extracting curve-skeletons from digital shapes using occluding contours

Marco Livesu · Riccardo Scateni

Received: date / Accepted: date

Abstract Curve-skeletons are compact and semantically relevant shape descriptors, able to summarize both topology and pose of a wide range of digital objects. Most of the state-of-the-art algorithms for their computation rely on the type of geometric primitives used and sampling frequency. In this paper we introduce a formally sound and intuitive definition of curve-skeleton, then we propose a novel method for skeleton extraction that rely on the visual appearance of the shapes. To achieve this result we inspect the properties of occluding contours, showing how information about the symmetry axes of a 3D shape can be inferred by a small set of its planar projections. The proposed method is fast, insensitive to noise, capable of working with different shape representations, resolution insensitive and easy to implement.

Keywords curve-skeleton · perceptual shape analysis

1 Introduction

Digital objects are flooding our environments: whether they are reproductions of real existing objects or produced by artists and designers they are more and more complexes bearing fine details. Skeletons and subdivisions in parts (segmentations) are compact and semantically sound approximations of the digital objects very useful in several different fields like, to name a few, computer animation, medical imaging, mechanical design, and shape archival, matching and retrieval. Focusing on skeletons of three-dimensional objects we can distinguish between surface-skeletons and curve-skeletons, where the former want to adhere more directly to definition of collecting all the equidistant points from the boundaries of the shape, while the latter are collections of curves and, thus, are, by definition, more compact and simple to manipulate than the firsts. Especially when dealing with the animation of digital characters, the curve-skeleton is an extremely important feature since it is the best guidance for changing the pose of the character and performing interpolations among poses.

Digital machineries (e.g. range scanners) are very good at capturing, even in the finer details, the surface of a shape, while human sight and perception are enormously more efficient in capturing the essential of a shape: how many significant parts compose it, where are the joints, and so on. In other words, a good way to compute the curve-skeleton of a digital object is to try to mimic what the human mind does when looking at an object. This assumption is the basis of our approach as described in the rest of the paper.

2 Related work

Previous methods for curve-skeleton extraction can be sorted depending on the shape representation used as input. There are algorithms that are able to work on triangle meshes (e.g.: [8] [3] [27] [11] [28]) which is the usual representation found in computer animation, others well suited to process point clouds (e.g.: [7] [29])

M. Livesu · R. Scateni
Dipartimento di Matematica e Informatica
University of Cagliari
via Ospedale, 72
09124 – Cagliari, Italy
Tel.: +39 070 675 8540
Fax: +39 070 675 8504
E-mail: marco.livesu@unica.it
E-mail: riccardo@unica.it

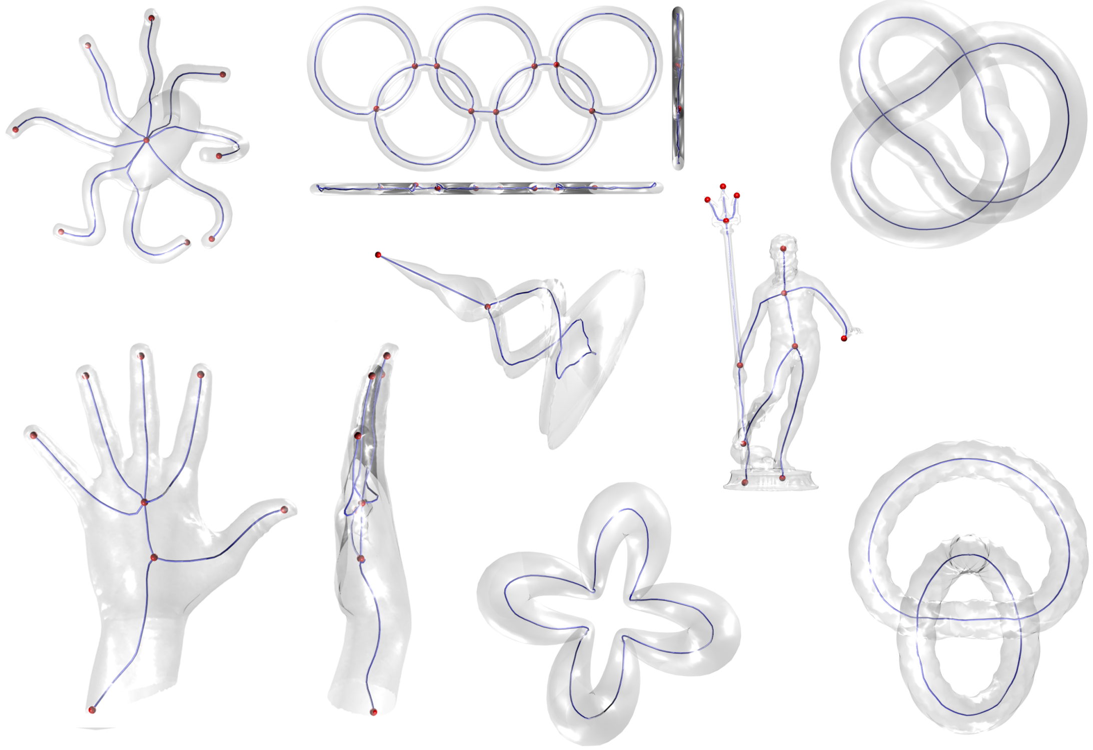


Fig. 1 Some result achieved with our method. The algorithm is able to perform with shapes of any genus, and with multiple connected components. The topology of the shape is preserved as long as the occluding contours used to extract the skeletons carry enough informations to observe it.

which range scanners produce as output with no further post-processing, and others with discrete volumes (e.g.: [17] [31] [10] [12]) since this is the format of acquisition for machinery like CAT and MRI. Most of these algorithms are able to perform well only when the model resolution is fine enough, leading to unstable results when they are applied to coarse models. As skeletons are supposed to be high-level descriptors, the difference between the descriptors computed starting from high and low resolution models should be negligible. These are the biggest drawbacks of the state-of-the-art skeleton extraction algorithms: they are too tightly coupled with geometric primitives and sampling frequency. They are working more on the machine side, focusing on *primitives* and *resolution*, than on the human side, focusing on *appearance*.

2.1 Main contribution

We give a formal definition of curve-skeleton of digital objects built as unions of Generalized Cones. We extend the contour interpretation to partially occluded silhouettes, providing a definition of locally unique symmetry point and moving the early visual perception theory from the global to the local setting. We formulate an algorithm to extract curve-skeletons from a set of occluding contours.

3 Theoretical background

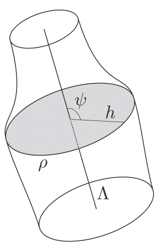
Shape analysis and recognition problems can be approached from another point of view, focusing more on appearance and less on primitives. We largely based our work on the perception based approach developed during the 1970s. People from M.I.T. Artificial Intelligence Laboratory, in particular David Marr, developed the theory of *early visual perception*, a study of how the human brain behave while looking to an image containing the projection of a real object. We briefly recall the background of this theory with our contributions before entering in the details of our proposed algorithm for skeleton extraction.

3.1 The generalized cones theory

According to the early perception theory, in [20] Marr and Nishihara stated that only the shapes belonging to the class of *generalized cones* can be fruitfully analysed. A generalized cone is *the surface swept out by moving* *a cross-section of constant shape but smoothly varying size along an axis* [4], or more formally

Definition 1 (Generalized cone) Let $\rho(r,\theta)$ be a simple closed planar curve twice continuously differentiable, and let h be a twice continuously differentiable positive real function. Let Λ be a line at some angle ψ to the plane containing ρ, and denote positions along Λ with z. Then, the surface $\mathcal{GC} = h \times \rho$ is a *generalized cone* with axis Λ, cross-section ρ, scaling function h, and eccentricity ψ.

There are strong links between generalized cones and curve-skeletons. First of all, only objects that can be described in terms of generalized cones can be well described by curve-skeletons. It is really difficult, for instance, to imagine the skeleton of a mug, a door, or a crumpled newspaper because the natural axis of these shapes are either too weak to describe them or external to the shape. In [9] Cornea et al suggested that the skeleton of a shape having a cavity should contain at least one loop around it but this would completely break down the topological connection between an object and its skeleton. To describe shapes containing tunnels or deep cavities non mono-dimensional shape descriptors like [21] or [22] would be more suitable.



We can express the relation between the generalized cones primitives and curve-skeletons formalizing their descriptions. Given a real object $\mathcal{O}$

$$\mathcal{O} = \bigcup_{i=1}^{n} \mathcal{GC}_i(\Lambda_i, \rho_i, h_i, \psi_i) \ ,$$

composed by n generalized cones, according with the ideas expressed in [20] we define the skeleton of $\mathcal{O}$ as the union of the axes of each generalized cone, that is

$$\mathsf{Skel}\,(\mathcal{O}) = \bigcup_{i=1}^{n} \Lambda_i \ .$$

Sticking to this definition of skeleton, in the remainder of the paper we will introduce a new algorithm for curve-skeleton extraction that makes use of the tools provided by the early visual perception framework to catch the axes of the generalized cones composing a shape.

In [19] Marr formally proved that, under few hypothesis, the axes Λ_i of an object $\mathcal{O}$ may be found just by analysing one its *occluding contour* (see [30]). A similar result, for a narrower set of shapes, have been achieved some years later by Rao and Medioni. In [24] they proved that *the contour of a solid of revolution is symmetric about the projection of its axis for any view.* These results can be therefore used to find the component axes of 3D objects without any *a priori* knowledge about their shapes.

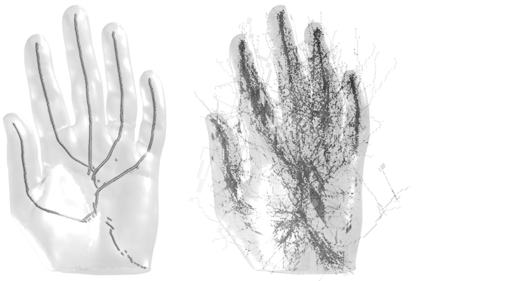


Fig. 2 Two raw point clouds produced by our method (left) and [18] (right) for the Olivier hand model. The same set of occluding contours has been used to make the comparison. Back-projecting only occlusion free locally unique symmetry points we can fully get rid of the noise, without any cleaning or post-processing. Moreover, working in the continuous the skeleton paths suggested by the cloud are naturally smooth and appealing.

The main drawbacks of the perceptual approach arise in the analysis of axes which are either foreshortened or hidden behind another part of the shape (i.e. occluded). To overcome the foreshortening problem we decided to feed our algorithm with a set of silhouettes gathered from different points of view, making sure that every single axis is not foreshortened in at least one of them. To handle occlusions we moved Marr theory from the global to the local setting, so that we have been able to get as much information as possible from every single contour, either containing occlusions or not. The multi-view approach is also justified by the fact that shapes belonging to the class of unions of generalized cones can have complex topology or pose. For these shapes a view point that ensures that all the components are not occluded or foreshortened sometimes does not exists at all.

In literature there are several examples of centreline extraction from multiple views, especially in the Computer Vision field. Bullitt et al [6] used stereo views to extract centerlines from medical datasets. Yoon et al [32] employed a set of real cameras to catch many different silhouettes of a human, gather them together to compose a discrete volume, and then apply a curve-skeleton extraction algorithm based on Gradient Vector Flow. However, the theory of occluding contours has not ever been taken into account until recently. In a previous work [18] we assumed that the medial axis of a set of silhouettes of a 3D model were projections of the curve-skeleton of the shape. Then, we gathered the medial axis points in the discrete 3D space in order to reconstruct

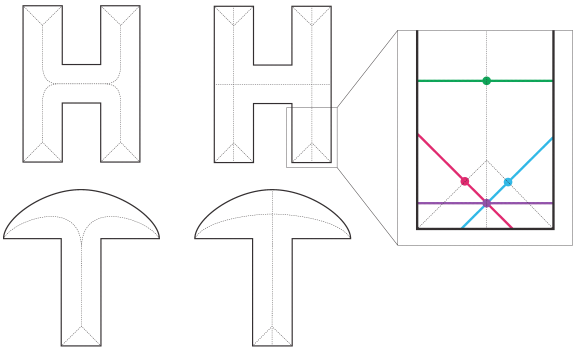


Fig. 3 A comparison between Medial Axis Transform (left) and Smoothed Local Symmetry (right). SLS is more powerful as it is able to capture all the symmetries of the figured shapes. Symmetry points are locally unique when their contact lines with the boundary do not intersect each other at any other symmetry point (close-up): red, blue and violet symmetry points are not unique, as their segments intersect each other, while the green point is locally unique.

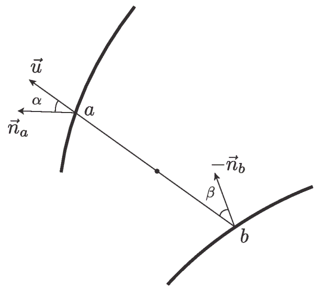
the skeleton with an ad hoc heuristic. Even though our assumption did not stand for every medial axis point of each silhouette, we have been able to achieve good results for a wide class of shapes of different topology and genus. Unfortunately, the drawbacks of our approach became evident when the amount of occlusions was high. In these cases the medial axis projected in the 3D space will bring a high amount of noise, making the skeleton extraction difficult and unstable. Nothing can be done to distinguish between noise and skeleton points because they both project into every single considered contour. Moreover, our method did not guarantee the skeleton paths to be centered because a discrete grid was used during the extraction and the smoothness was achieved only in post processing, thus deviating the curves from the medial lines of the shape. The method proposed in this paper is able to perform better with any kind of shape, producing incredibly noise free point clouds (see Figure 2) from which medial and naturally smooth skeleton curves can be easily computed, without requiring any further post-processing.

The advantages of the perceptual approach are several. Firstly, it is usually faster than the state of the art counterparts; secondly, it is completely unrelated with the geometric primitives used to describe the shape so that it is possible to extract coherent skeletons from any kind of representation (e.g. polygon meshes, implicit surfaces, parametric surfaces). Moreover, as long as the appearance of the object is preserved, resolution and noise has a negligible impact on the final result.

Most natural and artificial objects are unions of smooth elongated parts, and can, thus, be described as unions of generalized cone primitives.

3.2 Inspecting symmetries

The symmetry set of a domain $\Omega \in \mathbb{R}^2$ is the set of the centres of circles tangent to the boundary $\partial\Omega$ at at least two distinct points. We here describe the *Smoothed Local Symmetry*, a local shape descriptor introduced in [5].



Let a, b be two points on the boundary $\partial\Omega$, and $\mathbf{u}$ the unit vector in the direction $\overline{ab}$. By definition, the midpoint of the segment $\overline{ab}$ belongs to SLS(Ω) if and only if the angle α between $\mathbf{u}$ and the outward normal at a is equal to that β between $\mathbf{u}$ and the inward normal at b.

SLS is a very powerful shape descriptor as it is able to catch *all* the local symmetries of a contour; to have an example one can look at Figure 3, where a comparison with the Medial Axis Transform (MAT) is provided. This is particularly interesting because makes easier the characterization of locally unique symmetry points. A symmetry point $p = (a+b)/2 \in \text{SLS}(\Omega)$ is said to be *locally unique* if and only if

$$\overline{ab} \cap \text{SLS}(\Omega) = p\,. \qquad (1)$$

For example, looking at Figure 3, one can note that the most of the SLS points in the surroundings of the sharp corners of the boundary are not locally unique. When a point is locally unique, the distance $\|a-b\|$ can be used as an approximation of the local thickness of the shape. This particular property will be discussed in the remainder of the paper.

3.3 Analysis of occluding contours

When humans look at a silhouette, they perceive it as a particular 3D shape even though such silhouette could, in theory, be generated by an infinite number of shapes. In this section we will briefly introduce the theory of early visual perception, and we will discuss some improvements of it, useful to analyse occluded silhouettes.

Let $\mathcal{GC}$ be a generalized cone, and let Ω be its silhouette as seen from a viewpoint v, with π_v be the linear projection which define the mapping $\pi_v : \mathcal{GC} \rightarrow \Omega$. We call *occluding contour* the boundary $\partial\Omega$, and *contour generator* ($\mathcal{GC}_{\partial\Omega}$) the set of points $p \in \mathcal{GC}$ that project onto $\partial\Omega$. In [19] Marr proved that, given a generalized cone $\mathcal{GC}\,(\Lambda, \rho, h, \psi)$ and a projection function π_v, if the axis of symmetry of the projection $\pi_v\,(\mathcal{GC})$ is unique, then it is the actual projection of the axis of symmetry Λ. To prove it he assumed the following restrictions to be globally satisfied

R1: each point on the contour generator projects to a different point on the contour, that is, $\mathcal{GC}$ is convex as seen from v or, in other words, the inverse $\pi_v^{-1} : \partial\Omega \to \mathcal{GC}_{\partial\Omega}$ is one-valued

R2: nearby points on the contour arise from nearby points on the contour generator, that is, the mapping $\pi_v : \mathcal{GC}_{\partial\Omega} \to \partial\Omega$ is continuous

R3: the contour generator is planar

The first thing we observed is that there is an interesting link between Marr's theory and the Smoothed Local Symmetry (SLS). When restrictions *R1-R3* are satisfied the following relation is satisfied as well

$$\pi_v\,(\Lambda) \subseteq SLS\,(\Omega)\ . \tag{2}$$

This is straightforward to prove because, by construction, each symmetry point in Ω belongs to SLS(Ω). Therefore, if Λ projects to the axis of symmetry of Ω, it has to belong to SLS$\,(\Omega)$. It is important to notice that, if the axis of symmetry of Ω is unique, then $\pi_v\,(\Lambda) = \text{SLS}\,(\Omega)$. In any other case, at least one symmetry point p such that $p \in \text{SLS}(\Omega)$ and $p \notin \pi_v\,(\Lambda)$ must exist. One should note that the relation above is true only for medial descriptors able to catch every possible symmetry. For example, for the Medial Axis Transform (MAT) it would be false in the surroundings of the sharp corners of the shapes in Figure 3.

What happens when the silhouette is partially occluded? Should we discard it? Or we can still get some good information from it? The first thing to do is to get rid of the symmetry points that have been affected by occlusions. Occlusion-free symmetry points can be locally characterized exploiting the locality properties of SLS. To do this we define a function $\Phi : \Omega \to \mathbb{I}^{+}$ that assign to each point of the silhouette the number of points projected over it by π_v. Since each symmetry point $p = (a+b)\backslash 2 \in \text{SLS}(\Omega)$ depends only on the behaviour of the boundary restricted to a and b we can state that p is *occlusion-free* if and only if each point in $\overline{ab}$ is occlusion-free, that is

$$\forall\, q \in \overline{ab},\ \Phi(q) \leq 2. \tag{3}$$

Restrictions *R1-R3* can be formulated in local fashion too so now we are ready to define an occlusion-aware equivalent of the Marr's theory. Let $p = (a+b)\backslash 2$ be an inner point of SLS(Ω), if the symmetry is locally unique and occlusion-free in p, and if restrictions *R1-R3* locally hold, then

$$p \in \pi(\Lambda)\ . \tag{4}$$

Firstly, if *R1-R3* are satisfied in a and b then p will always be a SLS point, regardless the behaviour of the rest of the boundary $\partial\Omega$. Secondly, let Ω_1, Ω_2 be the connected components of $\Omega\backslash\overline{ab}$. By construction SLS is connected, hence two points $\lambda_1 \in \Omega_1, \lambda_2 \in \Omega_2$ always exists. Moreover, as the axis Λ and the mapping π are linear, $\pi(\Lambda)$ and $\overline{ab}$ have exactly one intersection. Therefore, for (1) and (2), they intersect in p.

4 Curve-skeleton extraction

We propose here a novel curve-skeleton extraction algorithm that exploits the theory presented in the previous section to extract curve-skeletons of a 3D shape just by looking at multiple silhouettes of it. The idea is, at high-level, very simple and intuitive. Firstly, we gather together multiple occluding contours of a 3D shape as seen from different viewpoints. Secondly, we extract from each contour the symmetry points that are projection of the axes of the generalized cones composing the object. We eventually match such symmetry points among the collected views in order to discover the axes in the 3D space using basic computer vision tools.

4.1 Silhouette analysis

The usage of a local analysis of the occluding contours is the major innovation of this paper and also the most innovative part of the proposed method over our approach previously presented in [18]. Basically, it consists in taking silhouettes, calculate their symmetries, and filter all the symmetry points that are not unique or occlusion free. To compute the SLS an algorithm is proposed in [5]. While uniqueness can be easily checked during the SLS calculation, to check if a point is occlusion free or not a discrete version of the function Φ must be implemented. To do this we employed the OpenGL stencil buffer, with the following setting

```
glEnable(GL_STENCIL_TEST);
glStencilFunc(GL_ALWAYS, 0x1, 0x1);
glStencilOp(GL_INCR, GL_INCR, GL_INCR);
```

The code above makes sure that each time a object's primitive is projected over one pixel, the stencil buffer entry corresponding to it will be increased by one.

4.2 View collection

The choice of the viewpoints is the core factor in the construction of the perceptual skeleton of the object. In [14] Laurentini stated that the number of silhouettes necessary to optimally describe a polyhedra with n faces is: unbounded if the viewpoints are not allowed to lie inside the convex-hull of the object; $O\left(n^5\right)$ if the viewpoints are allowed to stay into the convex hull. Moreover, even if the views choice is optimal, some problems may occur, for example in case the object contains cavities (see [13]). In our method silhouettes are gathered just by rotating along the most important axis given by the Principal Component Analysis (PCA) of the 3D shape with a step of 3°. We used at most 60 silhouettes for complex models, thus covering a rotation of 180° around the object (e.g. fertility and knots) and fewer views for simple models (e.g. olympics). This choice proved to be sufficient in most of our experiments. However, the method does not depend on the particular camera positioning. Some heuristics, like [25] [23] and [26] would accommodate better contours for some shapes.

4.3 Scanline matching

In all our experiments we used parallel projections in order to produce *rectified* sequences of contours. This choice makes the point matching problem very easy to solve because candidate matches always lie in the same scanline.

We used a two-step matching algorithm. Let $p^{(i)}$ and $p^{(j)}$ be two symmetry points belonging respectively to the i^{th} and j^{th} occluding contours. To have a positive match $p^{(i)}$ and $p^{(j)}$ must belong to adjacent views, lie in the same scanline, and their distance along the scanline must be lower than a predefined threshold. In our simplified environment this can be stated as

$$\left|\, i-j \,\right| = 1 \quad \wedge \quad p_r^{(i)} = p_r^{(j)} \quad \wedge \quad \left|\, p_c^{(i)} - p_c^{(j)} \,\right| \le \delta \;, \quad (5)$$

where δ is the maximum allowed displacement between two consecutive observations of the same point of a generalized cones axis (in all our tests $\delta = 2$ pixels). This first matching has the effect to group together all the subsequent observations of a symmetry point. We refer to this sets as *bundles*. However, the same point can be observed for a certain number of views, disappear due to an occlusion, and then appear again. In the second matching step we merge together all the bundles that have been generated by projections of the same skeleton point. To do this we apply the correspondence search algorithm proposed in [16]. Grouping in a single set all the observations of a skeleton point makes the algorithm more robust and also decrease the overall size and redundancy of the cloud (see Figure 4).

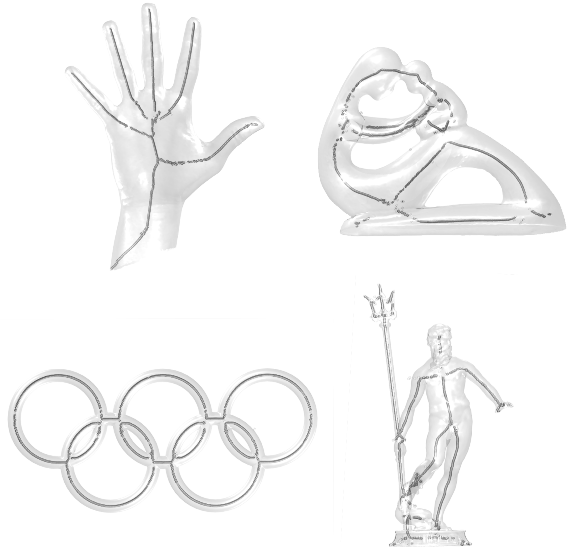


Fig. 4 Four raw point clouds produced back-projecting 2D symmetry points with our method. For each model we used a set of contours gathered by rotating around the most important axis given by the PCA. From each silhouette we select only the occlusion-free medial points, thus producing incredibly noise free clouds for many different kinds of objects.

4.4 Back-projection

After all the 2D symmetry points projections have been grouped in coherent bundles, the next step consists in projecting the points back to the shape space, in order to discover the skeleton paths. Every symmetry point in the bundle defines a projective ray in the space; the intersection of all these lines defines a candidate skeleton point in $\mathbb{R}^3$. A projective ray is a line and it can be expressed as the intersection of two planes in the space. Let $p \in \mathbb{R}^3$ be the coordinates of a general point belonging to the ray, and $\mathbf{d}$ be the Direction Of Projection (DOP) of that ray. We can express it with the following linear system

$$\begin{cases} \mathbf{d}_y x - \mathbf{d}_x y = \mathbf{d}_y p_x - \mathbf{d}_x p_y \\ \mathbf{d}_z x - \mathbf{d}_x z = \mathbf{d}_z p_x - \mathbf{d}_x p_y \end{cases}$$

For any bundle we therefore set up a dense linear system $A\mathbf{x} = \mathbf{b}$ composed by $2n$ equations, where n is the number of symmetry points in the bundle. The matrix A contains the directions of projection of the rays generating the symmetry points, the unknowns of the problem are the xyz coordinates of a generalized cones' axes point. To increase the overall robustness we discard bundles having less than 3 points, therefore the system is always overdetermined and can be solved in the least square sense, according to the normal equations $\tilde{\mathbf{x}} = \left(A^T A\right)^{-1} A^T \mathbf{b}$, such that

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\| \mathbf{b} - A\mathbf{x} \right\|^2 .$$

An example of the point clouds produced by our method can be seen in Figure 4. As one can note, due to the power of the SLS filtering and the robustness of the scanline matching, even for complex models like Fertility and Neptune, the clouds are almost noise free and the skeleton paths are quite clean.

4.5 Shape thickness

Beside the xyz coordinates of the skeleton points, another important information can be inferred from the SLS of the occluding contours. As stated in Subsection 3.2 when a symmetry point is unique, the distance from its closest contact points can be thought as an approximation of the local thickness of the shape as seen from a viewpoint. For each skeleton point we therefore set the local thickness as the lower distance observed among the SLS points involved in the back-projection step. This information will be really useful in the following step, where the skeleton paths will be reconstructed starting from the point cloud just created. Thickness information can also be used in a lot of applications. For example in collision detection, where a coarse representation of a model can be really useful to detect collisions between articulated objects, drastically reducing the complexity of the problem. In Figure 7 an example of reconstruction achieved using about 50 maximal balls have been produced for Fertility and Olivier hand models.

Method	model	avg	std
		displ.	dev
Our method	Eight	0.059918	0.009305
Dey and Sun [11] ($\theta = 0.0$)		0.063134	0.012655
Dey and Sun [11] ($\theta = 0.5$)		0.065679	0.012750
Tagliasacchi et al [28]		0.001762	0.011715
Livesu et al [18]		0.065188	0.025373
Our method	Knot 1	0,044990	0.022698
Dey and Sun [11] ($\theta = 0.0$)		0.046228	0.023781
Dey and Sun [11] ($\theta = 0.5$)		0.050125	0.024415
Tagliasacchi et al [28]		0.001570	0.012924
Livesu et al [18]		0.092037	0.027562
Our method	Knot 1	0.052322	0.008020
Dey and Sun [11] ($\theta = 0.0$)		0.054011	0.008043
Dey and Sun [11] ($\theta = 0.5$)		0.054063	0.008049
Tagliasacchi et al [28]		0.001742	0.013306
Livesu et al [18]		0.079337	0.018083

Table 1 Numerical comparisons with three state of the art skeletonization algorithms. We considered a set of synthetic shapes with convex cross section. We firstly sub-sampled the skeletons and then, for each sample, we measured its distance from the centroid of the local cross section. Mean displacements are normalized respect to the diagonal of the axis aligned bounding box containing the shape.

4.6 Curve extraction

The last step of our algorithm consists in the creation of the skeleton paths starting from the point cloud produced at the previous step. To do this we used an approach firstly proposed in [15], and also used in [29]. As can be noted in Figure 4 point clouds have a very thin structure along the branches, while points are a bit scattered near the joints. We then employ a 1D moving least square (MLS) approach for the branches, iteratively projecting points onto their corresponding locally best fitting lines via principal component analysis (PCA). At each iterative step we select a subset of points lying in a small neighbourhood. Since the cloud has been produced by the discrete representation of the contours, we use as neighbourhood size a quantity proportional to the distance between two points that would project in two adjacent pixels. However, as near joints points are too scattered, this neighbourhood measure is not able to work properly to infer branch connectivity. To thin the cloud and reveal how the branches connect each other we therefore apply Laplacian smoothing, using our estimation of the local thickness of the shape to infer point connectivity. It is now clear how important thickness is: in [29] the authors retrieved point connectivity with the help of the Mahalanobis distance. This was the most time consuming task of their algorithm; for 10K points they needed about 3 minutes of computations.

5 Results and comparisons

We implemented our methods in C++, using the [2] for the manipulation of geometric data structures and [1] for numerical computations. Experiments were run on a iMac equipped with 2.66GHz Intel Core 2 Duo and 4GB RAM. The application runs on a single core. We used silhouettes of size 500 × 500 in all our tests; this choice proved to be a good trade-off between efficiency and precision. If necessary, higher resolution would permit higher precision.

We discuss here the skeletons produced with our algorithm according to the properties listed in [9]. **Homotopy** is observed but not guaranteed. It depends on the considered contours; we can guarantee it as long as any cone axis is projected without occlusions enough times to generate a bundle. **Centeredness** is also observed but not guaranteed, in the sense that the least square solution may move some skeleton point slightly

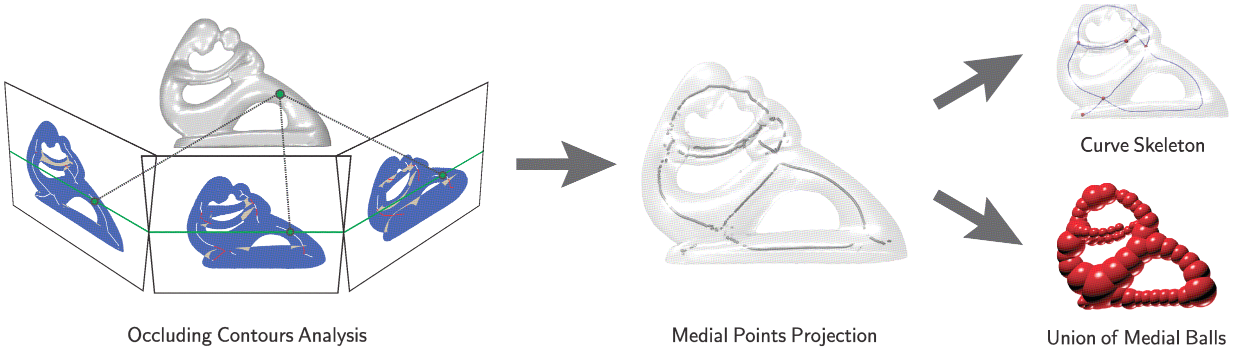


Fig. 5 The pipeline of the whole process: we first collect a set of occluding contours from neighbouring viewpoints; the result is a cloud of point sampling the axes of the generalized cones composing the shape; we eventually reconstruct the curve-skeleton, and an approximation of the shape given by the union of the medial balls.

far from the cone's axis. Being based on the visual appearance of the objects the algorithm is very **robust**. It is almost insensitive to noise and missing parts (see figure 6). Moreover, thickness informations allow a rough **reconstruction** of the shape (see Figure 7).

In Table 1 we compared our method with three state of the art algorithms. In this comparison we considered three synthetic shapes (a double torus and two knot models) with convex cross section everywhere. We evenly sampled the skeletons and, for each skeleton point, we cut the mesh with a plane centered in it and having as normal direction the direction normal to the skeleton curve. We then measured the distance between the skeleton point and the centroid of the cross section, normalizing it respect to the diagonal of the axis aligned bounding box containing the shape. As can be noted from the table our results are comparable with the results achieved in [11] and [28] while [18] behaves slightly worse, probably because the curves are extracted from a voxel grid and then smoothed in post processing, thus deviating from the middle of the shape.

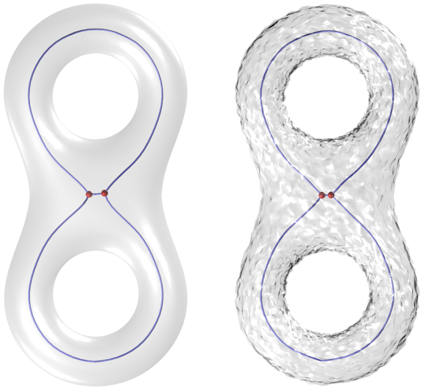


Fig. 6 Being based on shape appearance rather than on geometry our method is able to handle noise successfully. In this image two curve skeletons extracted respectively from a noise-free and a noise-affected double torus.

In Table 2 we report running times and number of considered contours for some models we tested. Our method runs faster than the state of the art counterparts, especially for high resolution models. For instance, on models with approximatively 60,000 faces, we compute the skeleton in half a second, [28] takes few seconds, [3] few minutes, and [11] almost half an hour. Moreover, since the most time consuming task is the rasterization, times may be further lowered using smart rendering techniques.

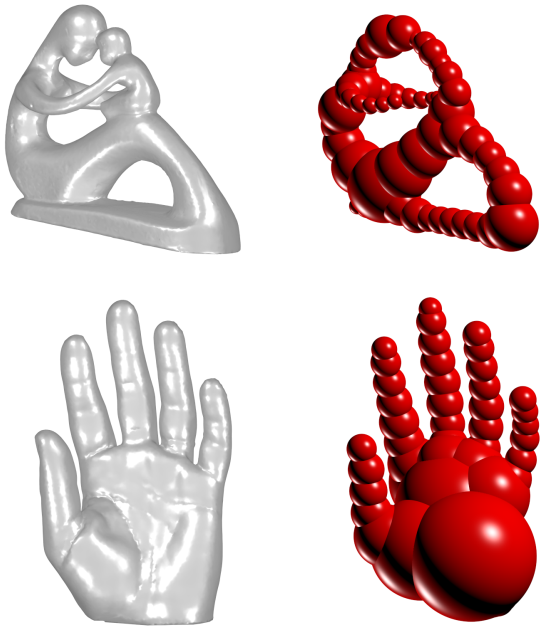
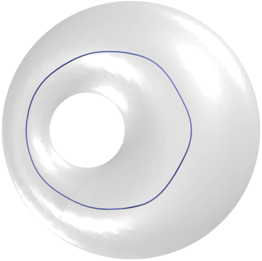


Fig. 7 Shape approximation of the Fertility model (55 balls) and Olivier Hand model (42 balls). Maximal balls are spanned along the skeleton paths, creating a good approximation of the original shape. This representation can be useful for applications like hole filling, surface reconstruction and collision detection.

5.1 Implicit Surfaces

Triangle meshes are the most common data structure for surface representation and all the state of the art algorithms we compared with are able to deal with them. However, they have two big drawbacks: they are only $\mathcal{C}^0$ continuous and their topology must be checked explicitly in order to avoid self intersections that are not found in real world objects.



When one wants to overcome this drawbacks uses other surface representations like parametric or implicit surfaces. One of the greatest advantages of our method (as well as [18]) is the possibility to deal with this representations without any additional effort: the computation of the silhouettes can be merely reduced to the computation of the projection of the surface onto the projection planes. Meshing an implicit surface is instead a time consuming task, and this is why usually the surface is calculated on-the-fly at raster time, without any explicit representation. To the best of our knowledge, contour based approaches are the only techniques available in literature for the direct skeletonization of implicit surfaces. In the figure above we show an example of the curve skeleton extracted from a Dupin's cyclide with equation $693x^4+1386x^2y^2-18880x^2y+1386x^2z^2+59000x^2+693y^4-18880y^3+1386y^2z^2+187000y^2-18880yz^2-800000y+693z^4+67000z^2+1250000=0$. To compute the projections, raster the silhouettes, and, finally, compute the skeleton, we considered the zero set of the equation.

6 Limitations and further works

It is worth to clearly remind that the method we describe here is meant to work only for objects that can be represented by a union of a finite number of generalized joined cones. To describe shapes with weakly defined symmetry axes, like mugs, busts or buildings, one should probably use a different descriptor.

About the multi-view based approach, beside the advantages, it carries some limitations. Back-projected points are not guaranteed to be inside the shape. Even if the matching algorithm is extremely robust, it can happen that when there are a lot of occlusions some points may be projected outside the shape due to a wrong matching. Moreover, as we are dealing with contours, we can only guarantee that the skeleton lie inside the *Visual Hull* of the object rather than the object itself. We lack a stop rule for the silhouette acquisition system. Our method uses a set of silhouettes constructed by rotating around the most important PCA direction; however, as one can note by looking at Table 2, some models need few contours and some others need more. A stop rule to automatically determine how many views we need should be used. About shape representations, our method is able to work with any polygon mesh and any discrete volume representation. Every kind of representation that allow the calculation of occlusions can be used interchangeably. We are currently working to extend the work to point clouds. This can be done by inferring the occlusion map (the discretized Φ function) by post processing the information contained in the stencil buffer. We are also working to extend this 2D-to-3D paradigm to real objects moving in front of a camera or multiple views of a static object.

7 Conclusions

In this paper we introduced an intuitive definition of curve skeleton and we showed how the axes of a complex generalized cone can be inferred just by looking their planar projections from few viewpoints. We extended the classical theory of early visual perception in order to deal with occlusions, proposing a skeletonization algorithm which is fast, insensitive to noise and missing data and easy to implement. Moreover, we showed that the results produced by our algorithm is qualitatively comparable with the state of the art counterparts but it is more versatile because it can be used with any kind of surface representation. We also showed that the centeredness of the skeletons produced by our algorithm is higher than previous contour-based approaches, because our skeleton curves are naturally smooth and do not need any post processing.

References

1. `http://eigen.tuxfamily.org/`
2. `http://vcg.sourceforge.net/`
3. Au, O.K.C., Tai, C.L., Chu, H.K., Cohen-Or, D., Lee, T.Y.: Skeleton extraction by mesh contraction. In: ACM Transactions on Graphics (TOG), vol. 27, p. 44. ACM (2008)
4. Binford, T.O.: Visual perception by computer. In: IEEE conference on Systems and Control, vol. 261, p. 262 (1971)
5. Brady, M., Asada, H.: Smoothed local symmetries and their implementation. The International Journal of Robotics Research **3**(3), 36–61 (1984)
6. Bullitt, E., Liu, A., Pizer, S.: Three-dimensional reconstruction of curves from pairs of projection views in the presence of error. i. algorithms. Medical Physics **24**, 1671 (1997)

Model	Faces	Occluding contours	Rasterization time (*ms*)	Skeletonization time (*ms*)	Total time (*ms*)
Olivier Hand	49,586	37	646	249	895
Hand	273,060	6	247	74	321
Neptune	56,112	32	321	250	571
Double torus	1,536	6	22	79	101
Fertility	50,000	60	546	391	937
Twirl	10,402	20	112	175	287
Knot #1	4,160	60	202	382	584
Knot #2	6,400	60	227	396	623
Knot #3	11,520	60	266	302	568
Olympics	7,862	6	24	155	179
Spider	23,808	60	357	442	799

Table 2 Time splitting (in milliseconds) of the skeleton extraction pipeline.

7. Cao, J., Tagliasacchi, A., Olson, M., Zhang, H., Su, Z.: Point cloud skeletons via laplacian based contraction. In: Shape Modeling International Conference (SMI), 2010, pp. 187–197. IEEE (2010)
8. Chuang, M., Kazhdan, M.: Fast mean-curvature flow via finite-elements tracking. In: Computer Graphics Forum, vol. 30, pp. 1750–1760. Wiley Online Library (2011)
9. Cornea, N., Silver, D., Yuan, X., Balasubramanian, R.: Computing hierarchical curve-skeletons of 3D objects. The Visual Computer **21**(11), 945–955 (2005)
10. Cornea, N.D., Silver, D., Min, P.: Curve-skeleton applications. In: IEEE Visualization, pp. 95–102 (2005)
11. Dey, T.K., Sun, J.: Defining and computing curve-skeletons with medial geodesic function. In: Proceedings of the fourth Eurographics symposium on Geometry processing, SGP '06, pp. 143–152. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2006)
12. Hassouna, M.S., Farag, A.A.: Robust centerline extraction framework using level sets. In: CVPR '05, vol. 1, pp. 458–465 (2005)
13. Laurentini, A.: How far 3D shapes can be understood from 2D silhouettes. Pattern Analysis and Machine Intelligence, IEEE Transactions on **17**(2), 188–195 (1995)
14. Laurentini, A.: How many 2D silhouettes does it take to reconstruct a 3D object? Computer Vision and Image Understanding **67**(1), 81–87 (1997)
15. Lee, I.K.: Curve reconstruction from unorganized points. Comput. Aided Geom. Des. **17**(2), 161–177 (2000)
16. Leordeanu, M., Hebert, M.: A spectral technique for correspondence problems using pairwise constraints. In: Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 2, ICCV '05, pp. 1482–1489. IEEE Computer Society, Washington, DC, USA (2005)
17. Liu, L., Chambers, E.W., Letscher, D., Ju, T.: A simple and robust thinning algorithm on cell complexes. Comput. Graph. Forum **29**(7), 2253–2260 (2010)
18. 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**(11), 1891–1901 (2012)
19. Marr, D.: Analysis of Occluding Contour. Proceedings of the Royal Society of London. Series B, Biological Sciences **197**(1129), 441–475 (1977)
20. Marr, D., Nishihara, H.K.: Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes. Proceedings of the Royal Society of London **200**(1140), 269–294 (1978)
21. Martin, T., Chen, G., Musuvathy, S., Cohen, E., Hansen, C.: Generalized swept mid-structure for polygonal models. Computer Graphics Forum **31**, 805–814 (2012)
22. Miklos, B., Giesen, J., Pauly, M.: Discrete scale axis representations for 3D geometry. ACM Trans. Graph. **29**, 101:1–101:10 (2010)
23. Pujari, A.: Volume intersection for shape from silhouettes. Sadhana **18**, 325–336 (1993). 10.1007/BF02742664
24. Rao, K., Medioni, G.: Useful geometric properties of the generalized cone. In: Computer Vision and Pattern Recognition, 1988. Proceedings CVPR '88., Computer Society Conference on, pp. 276 –281 (1988)
25. Secord, A., Lu, J., Finkelstein, A., Singh, M., Nealen, A.: Perceptual models of viewpoint preference. ACM Transactions on Graphics **30**(5) (2011)
26. Shanmukh, K., Pujari, A.K.: Volume intersection with optimal set of directions. Pattern Recognition Letters **12**(3), 165 – 170 (1991)
27. Sharf, A., Lewiner, T., Shamir, A., Kobbelt, L.: On-the-fly curve-skeleton computation for 3D shapes. Computer Graphics Forum **26**(3), 323–328 (2007)
28. Tagliasacchi, A., Alhashim, I., Olson, M., Zhang, H.: Mean curvature skeletons **31**(5), 1735–1744 (2012)
29. Tagliasacchi, A., Zhang, H., Cohen-Or, D.: Curve skeleton extraction from incomplete point cloud. ACM Trans. Graph. **28**(3), 1–9 (2009)
30. Waltz, D.: Understanding line drawings of scenes with shadows. In: P. Winston (ed.) The Psychology of Computer Vision, pp. 19–91. McGraw-Hill (1975)
31. Wang, Y.S., Lee, T.Y.: Curve-skeleton extraction using iterative least squares optimization. IEEE Transactions on Visualization and Computer Graphics **14**(4), 926–36 (2008)
32. Yoon, S.M., Malerczyk, C., Graf, H.: 3D Skeleton Extraction from Volume Data Based on Normalized Gradient Vector Flow. In: The 17^{th} International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, pp. 177–182. Plzen, Czech Republic (2009)