Reconstructing the Curve-Skeletons of 3D Shapes Using the Visual Hull

Marco Livesu, Fabio Guggeri, and Riccardo Scateni

Abstract—Curve-skeletons are the most important descriptors for shapes, capable of capturing in a synthetic manner the most relevant features. They are useful for many different applications: from shape matching and retrieval, to medical imaging, to animation. This has led, over the years, to the development of several different techniques for extraction, each trying to comply with specific goals. We propose a novel technique which stems from the intuition of reproducing what a human being does to deduce the shape of an object holding it in his or her hand and rotating. To accomplish this, we use the formal definitions of epipolar geometry and visual hull. We show how it is possible to infer the curve-skeleton of a broad class of 3D shapes, along with an estimation of the radii of the maximal inscribed balls, by gathering information about the medial axes of their projections on the image planes of the stereographic vision. It is definitely worth to point out that our method works indifferently on (even unoriented) polygonal meshes, voxel models, and point clouds. Moreover, it is insensitive to noise, pose-invariant, resolution-invariant, and robust when applied to incomplete data sets.

Index Terms—Curve-skeleton, stereoscopic vision, visual hull

1 Introduction

Shape analysis focuses on the definition of mesh descriptors that can capture the main features of an object. This can give useful insights and knowledge on the shapes to provide quantitative means of measuring the similarity between two objects. Among the various descriptors proposed in the literature, *skeletons* have been found to be of great importance in many fields due to their versatility. For instance, the thinness and centricity features obtained by skeletons make them suitable to be used for motion planning; also, their ability to accurately summarize the topology of an object is used in shape retrieval, data compression, and so on. The original 2D definition [1] states that a *topological skeleton* or *medial axis* of a shape is the locus of centers of the maximal inscribed disks. During the decades, the term *skeleton* has been applied to every kind of thin, monodimensional, graph-like structure that best represents a given shape. Especially in the 3D case, the direct extension of the definition (i.e., the centers of the maximal inscribed balls), does not guarantee the thinness criterion. Even if Dey and Sun [2] provide a formal definition for *curve skeletons* in 3D, its exact computation is hard and unstable, thus making preferable an ad hoc skeleton extraction method that satisfies the desired goal-specific criteria.

We propose a novel method based on the observation of human capabilities in terms of shape recognition. Many works have been carried out toward the understanding of how human observers parse and interpret shapes [3], [4], [5]. These studies have shown the importance of silhouettes in early perception, demonstrating that we are able to give a 3D interpretation of a shape with no information other than the contour [6], [7]. Despite the background in perception, the idea of computing a curve-skeleton starting from 2D data is unexplored in the field of geometry processing. While the paradigm of recovering 3D shape information using 2D data is well known in fields like Medical Imaging (e.g., see [8]) or Computer Vision techniques (e.g., [9] and [10]), no studies have been carried out toward the connection between 2- and 3D skeletons of synthetic shapes. The cited techniques are based on real-world images and aim at problems where no a priori knowledge of the shape is present [11], the availability of information on the shape primitives has favored the proliferation of other extraction methods. However, humans have strong recognition capabilities even when a lot of data lack. Transferring these capabilities in the field of computational geometry should improve the reliability and robustness of the algorithms.

In an attempt to describe human vision, Marr [3] suggested that 2D contours are interpreted as projections of *generalized cones*. In his work, he demonstrates that, in the absence of overlap along the boundary of the projection, the axis of symmetry of a projected shape is the **actual projection** of the axis of symmetry of the 3D shape [3, Theorems 3, 4, and 5]. Even if the absence of overlap is a strong constraint, and Marr's work states that this property is valid for most of the real-world cases (see Fig. 3), the ambiguity given by superimpositions on the image plane can be solved by looking at the object from many different points of view. It has been shown that it improves the recognition capabilities in human observers [12], suggesting that multiple rotated projections of the object should be sufficient to detect the features of the 3D counterparts.

Numerous works facing the curve-skeleton extraction problem were published before this one. Some of them working over voxelizations of the shapes; some others working over triangle meshes. On one hand, voxel-based methods usually tend to be computationally expensive,

• *The authors are with the Dipartimento di Matematica e Informatica, Università degli Studi di Cagliari, via Ospedale, 72, Cagliari 09124, Italy. E-mail: {marco.livesu, guggeri, riccardo}@unica.it.*

Manuscript received 26 July 2011; revised 9 Nov. 2011; accepted 31 Jan. 2012; published online 29 Feb. 2012.
Recommended for acceptance by W. Wang.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2011-07-0171.
Digital Object Identifier no. 10.1109/TVCG.2012.71.

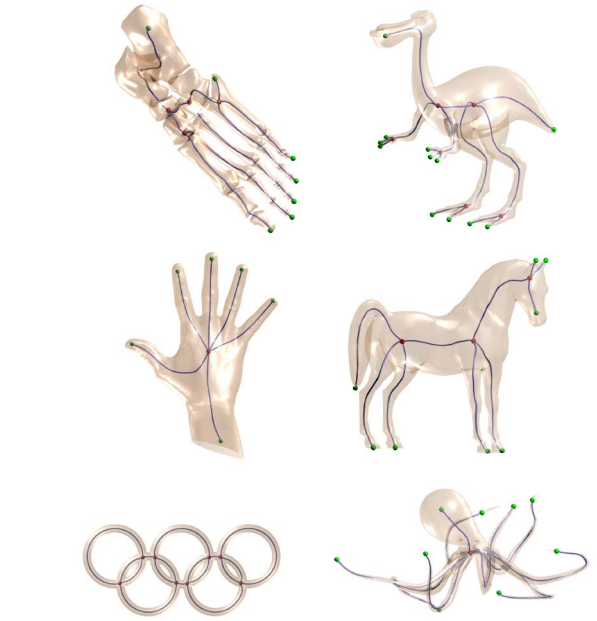


Fig. 1. Curve-skeletons extracted (top left to bottom right) from the Foot, Dinopet, Hand, Horse, Olympics, and Octopus.

sometimes fail to preserve the topology of the object, and are strongly dependent on the model resolution. On the other hand, mesh-based methods are all dependent on the resolution of the mesh, both in terms of running time and output quality, raising the problem to find a tradeoff between efficiency and quality.

Our **main claims** are that: our method is able to work over any kind of shape representation (e.g., triangle meshes, quad meshes, point clouds, polygon soups, etc.), and it overcomes the drawbacks of both voxel- and mesh-based methods since it achieves good results with coarse meshes and extracts the curve-skeleton from fine meshes quickly enough.

We present a summary of the previous works in curve-skeleton extraction in Section 2, followed by a short introduction of the Computer Vision principles useful to our work in Section 3. We then describe in detail our approach in Section 4, with implementation details that help in improving the efficiency of the computation. Section 5 shows the results obtained with our approach along with limitations and comparisons between the proposed method and other techniques, showing that our method is faster than the current state-of-the-art algorithms. We finally summarize our work in Section 6.

2 Related Works

Previous work on skeleton extraction consists of a large number of methods and approaches due to their importance and usefulness in many fields. The heterogeneity of such fields, however, makes it difficult to expose the previous works under a common point of view. It is however reasonable to subdivide the methods in two main families according to the object representation used. In the volumetric category, the works are based on a discretization of the surface for extraction, while geometric methods perform the skeletonization directly on the primitives that define the surface. For an extensive survey on skeleton

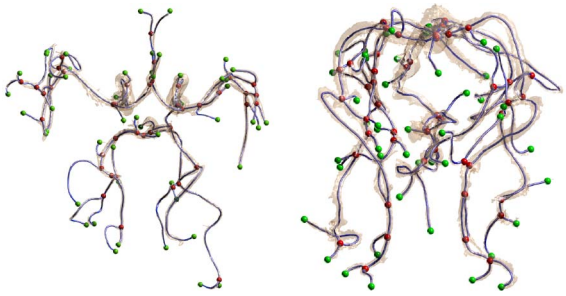


Fig. 2. Two examples of curve-skeletons extracted from data sets with complex morphology: the ramifications of, respectively, normal and aneurysmatic vessels.

extraction methods, along with a discussion on the common characteristics of such techniques, one may refer to [13]. In the rest of this section, we focus our attention only on some methods most relevant to our work and mainly published after the survey was written.

2.1 Volumetric Methods

Most voxel-based methods take advantage of the discretized space and known topological constraints. Thinning-based methods tend to iteratively shrink the shape while maintaining the topological coherency in different fashions. In [14], a hierarchical decomposition of the volume is performed, thinning each simple subvolume to extract the segments that form the final skeleton. In [15] and [16], the goal is to parallelize the thinning process while trying to

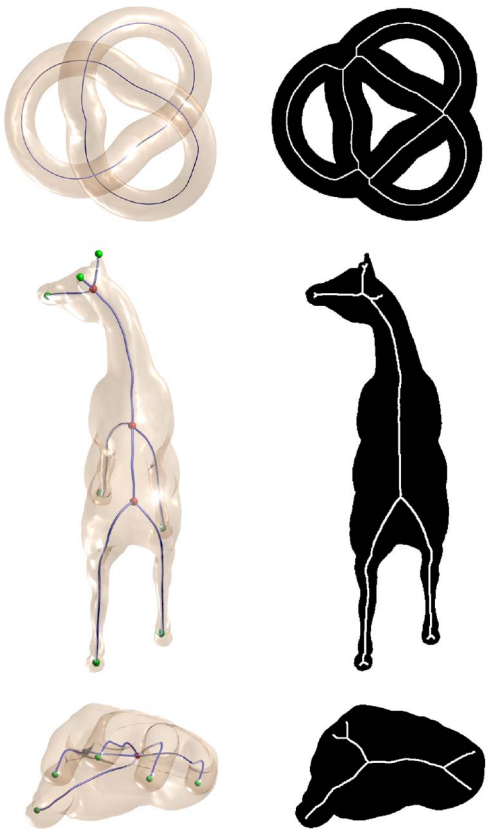


Fig. 3. Most medial axes reflect the actual 3D shape when the overlap is reduced (upper, Knot model). According to the direction of view, some features may be missing (middle, Horse model), or completely unrelated (lower, Hand model).

maintain the topology. Methods based on the discretization of a function detect critical points in such functions in order to extract the skeleton. In [17], a repulsive force function is computed from the border of the object to the interior, detecting ridges as skeletal points. In [18], thinning is guided by the distance transform (DT), similarly to [19] where the distance from the border is used to propagate a front with different speeds. In [20], the thinning process is guided by a measure called *medial persistence* to increase the robustness.

2.2 Mesh-Based Methods

Mesh-based algorithms are a highly heterogeneous family. Since skeletons may be used for several purposes, the methods for extraction vary strongly according to the goal. In [2], Dey and Sun provide a formal definition of curve-skeletons as a subset of the medial axis, introducing a function called Medial Geodesic Function (MGF) based on the geodesic distances between the contact points of the maximal balls. The skeleton is extracted as the singularities of the MGF. In [21], a Laplacian contraction is applied to the object with topological constraint for skeleton extraction and segmentation, in a manner similar to [22] where the Laplacian contraction can be also applied to point clouds in order to extract the skeleton. In [23], a deformable model is grown into the object to detect the branches from both meshes and point clouds, while in [24] the mesh is iteratively decomposed into hierarchical segments, computing a center-line compression error until a threshold is reached.

3 Theoretical Foundations

Our approach borrows some tools and definitions from the field of *computer vision*, where the bonds with human perception are stronger. As some notions can be unfamiliar to a shape processing audience, we briefly summarize the definitions in this section in order to make the presentation of our approach more self-consistent.

3.1 The Visual Hull ($\mathcal{VH}$)

Let $C \subseteq \mathbb{R}^3$ be a set of points of view; the *visual hull* of an object O relative to C, $\mathcal{VH}(O, C)$, is defined as the subspace of $\mathbb{R}^3$ such that, for each point $p \in \mathcal{VH}(O, C)$, and each point $c \in C$, the projective ray starting at c and passing through p contains at least a point of O [25]

$$\mathcal{VH}(O, C) = \{p \mid \overline{cp} \cap O \neq \emptyset, \forall c \in C\}.$$

In other words, the visual hull of an object is the maximal object that gives the same silhouettes of the original one from any considered viewpoint. $\mathcal{VH}(O, C)$ is also the **closest approximation** of O that we can obtain using its silhouettes [25, Proposition 2], that is, the best representation of what a human viewer would appreciate of the object in the absence of other information.

3.2 Epipolar Geometry and Rectification

Epipolar geometry provides us a quite useful constraint while trying to couple planar images of a point lying in the 3D space in order to find its spatial coordinates. Let's suppose to have a point $M \in \mathbb{R}_3$ and two projective cameras $\mathrm{P}_1, \mathrm{P}_2$ (with $\mathrm{P}_1 \neq \mathrm{P}_2$) which, respectively, project into the planes $\Pi_{\mathrm{P}_1}, \Pi_{\mathrm{P}_2}$. Given the image point $m_1 = \mathrm{P}_1(M)$, the point projected by the secondary camera, $m_2 = \mathrm{P}_2(M)$, must lie in the line defined by the intersection between the plane Π_{P_2} and the *epipolar plane*. Namely, the plane defined by the point M and the centers of projection of P_1 and P_2, meaning that for each m_1 the corresponding point m_2 must be searched in a monodimensional space instead of a bidimensional one.

The epipolar constraint can be enforced with *image rectification* in order to further reduce the complexity of this search: when the epipolar planes are parallel, *epipolar lines* become horizontal and the search is restricted to a *scanline matching*, meaning that is possible to search for a correspondence between the pixels along the same scanline in the two images. While image transformations are employed for rectification in real-world cases with projective cameras, this effect is also obtained using **affine projections**: it can be approximated by using a very long focal length camera. In synthetic environment, like the one we present in Section 4.2, the possibility to directly employ parallel projections can increase the efficiency of the matching.

4 Our Proposal

The algorithm is, on a high-level, extremely simple and intuitive. We first implicitly compute an approximation of the Visual Hull of the object using a set of stereoscopic projections from different points of view. A medial axis is extracted for each silhouette and the spatial position given by the stereoscopic match is used to vote the corresponding voxel in a regular grid. Spurious votes are filtered out in the grid if they fall outside the $\mathcal{VH}$, and a maximized spanning tree of the grid is computed. The tree is pruned and processed according to an estimation of the radii of the inscribed balls obtained by the 2D information so that no artifacts remain in the final skeleton and the $\mathcal{VH}$ topology is preserved. Finally, the skeleton is smoothed to improve its visual appearance.

The next sections describe in detail each step of the algorithm.

4.1 Camera Positioning and Medial Axis Extraction

The choice of the viewpoints is the core factor in the construction of the approximated model of the object. Even though it could be possible to specify a mesh-dependent set of views [26], there is no way to understand whether the obtained $\mathcal{VH}$ is a satisfactory approximation of the shape [25].

We thus choose to evenly cover the space around the object, employing a regular grid of cameras centered in the vertexes of a discrete 21-point hemisphere. Covering just half of the visible horizon is enough since the silhouettes projection is symmetric. Both the shape and the hemisphere are centered in the coordinate reference system, while the cameras point toward the origin (see Fig. 4). Intuitively, the more the viewpoints the more accurate the $\mathcal{VH}$. However, in our experiments, we found that finer resolution hemispheres do not increase the $\mathcal{VH}$ accuracy significantly.

To project points in the 3D space, we then build up a set of 21 stereo acquisition systems, pairing each camera in the hemisphere with a second one, having direction of projection perpendicular to it. This direction is also parallel to the less principal component of its projection (which is given by the smallest eigenvector of its projection's Principal

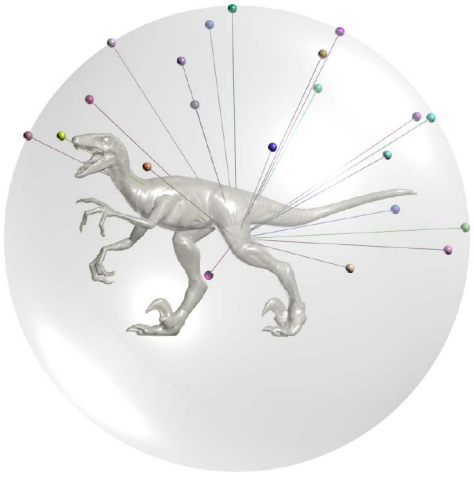


Fig. 4. To evenly cover the space around the shape, we place cameras in the vertexes of a discrete 21-points hemisphere. The shape is centered in the origin of the frame, and each camera looks toward the origin.

Component Analysis). This way of choosing the two directions of projection minimizes depth overlapping, and should, thus, be the best possible ones (an example of stereo pairs is in Fig. 5).

For each projected binary silhouette, we extract a Distance Transform-based medial axis [27]. The medial axis is stable and reliable, and the DT values of each pixel give the thickness of the $\mathcal{VH}$ along the image plane, adding volumetric information to each stereo projection. The usefulness of this information will be evident in the following tree processing step.

4.2 Matching and Radii Estimation

As pointed out in Section 3, a way to simplify the matching is to employ parallel projections. Two affine cameras P_z and P_x are positioned, respectively, along the z and the x axis, whereas the shape is centered in the coordinate system reference. Such cameras are defined by the homogeneous matrices

$$\mathrm{P}_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathrm{P}_x = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

In such system, epipolar planes are parallel and their normal direction is the y-axis. As the epipolar constraint coincides with the scanlines and projection rays are both

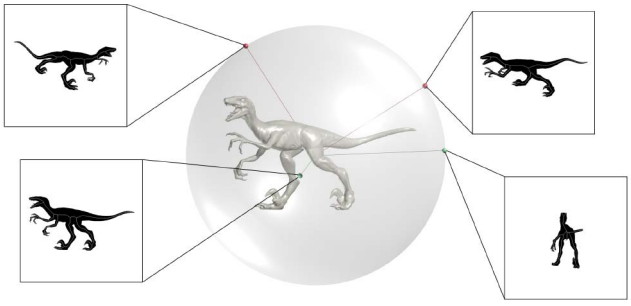


Fig. 5. Two stereo pairs, color-coded, that include the point of view from which they are taken and the resulting silhouettes.

orthogonal and axis aligned, back-projection becomes trivial: each pair of rays has the form

$$r \begin{cases} x = p \\ y = q \end{cases} \qquad r' \begin{cases} z = k \\ y = q, \end{cases}$$

where $y = q$ is provided by the epipolar constraint and the complete separation between x and z coordinates is provided by the orthogonal directions of projection. The back-projected point is then

$$r \times r' = [p\ q\ k\ 1]^T.$$

To keep the back-projection so simple, we move the shapes instead of the cameras. Let's consider a shape S centered in the coordinate system $\mathcal{F}(O, X\ Y\ Z)$: given a set of stereo points of view v_1, v_2, we define a new coordinate system reference $\mathcal{F}'(O, X'\ Y'\ Z')$ where the Z' and X' axes correspond, respectively, to the lines joining v_1 and v_2 with the origin O. To get the projection, we then bring S in $\mathcal{F}'$ applying the transformation $t^{-1}(S)$, where t is the rotation matrix defined such that $\mathcal{F}' \equiv t(\mathcal{F})$.

The method is based on a discretized voting space that can result in different vote accumulations depending on the coordinate system. In order to improve the robustness to rotational variance, we perform a Principal Component Analysis over the mesh vertices in preprocessing. We then rotate the object so that the first camera points toward the eigenvector corresponding to the smallest eigenvalue. This minimizes the information loss in projection. The *up-vector* is set parallel to the greatest eigenvector, thus maximizing data distribution on the y direction used in scanline matching. In this way, we are able to define a pair of views where the xy image is supposedly the best representative of

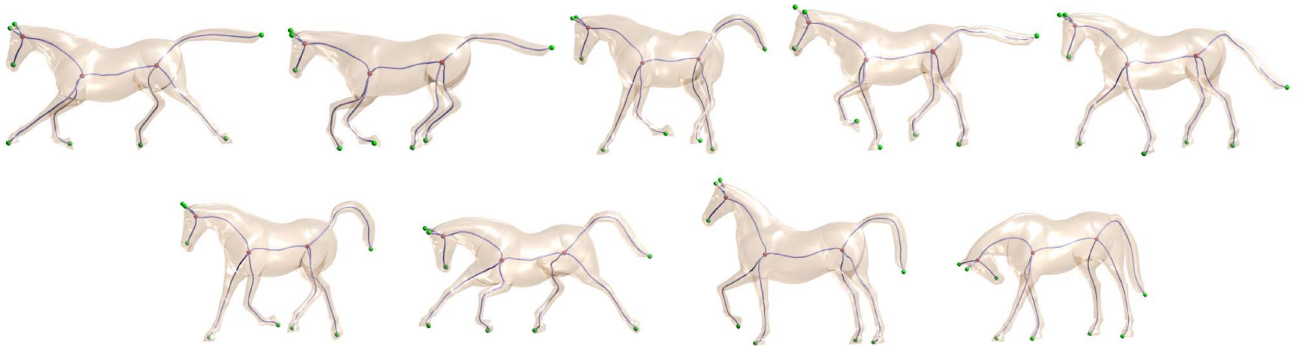


Fig. 6. We show that extracting the skeletons of different meshes of the same object in different poses, we always obtain the same result.

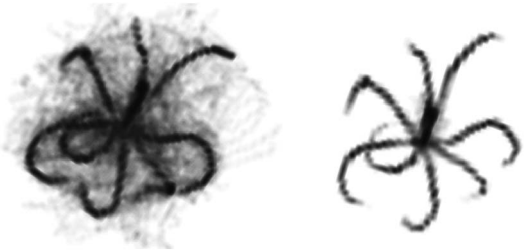


Fig. 7. The cleaning process drastically improves the quality of the voxel grid.

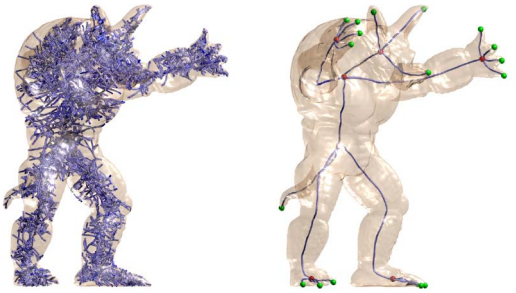


Fig. 8. The ORG spanning tree (left) is an unorganized set of connected voxels. Only perceptually significant branches are extracted as part of the skeleton (right).

the shape, and tends to uniform the set of views used for similar objects even under rotation.

The stereographic view system can accurately reconstruct the original positions and radii of the medial axes balls when matching is one-to-one. When the matching is many-to-many, the direct xy to z matching results in multiple points and the level of confidence of both ball position and radius rapidly decreases as the number of points increases. We employ a multiview voting system in order to give higher weights to those branches that remain consistent through a higher number of views.

The regularity of the grid makes the voting a simple operation. Let suppose we have a match, falling in the voxel i, j, k of the grid $\mathcal{G}$. The update procedure consists just in incrementing the correspondent cell

$$\mathcal{G}[i, j, k] = \mathcal{G}[i, j, k] + 1.$$

Each voxel stores also an estimated radius of the inscribed ball thanks to Distance Transform information on the generating images. Since the DT is an approximation of the triple of the euclidean Distance, for each pixel we can compute an estimated distance of the skeleton to the border of the hull along the image plane. Let $\mathcal{R}[i, j, k]$ be the radius associated with the voxel i, j, k. For each new vote in that voxel, we update the radius as

$$\mathcal{R}[i, j, k] = \min(\mathcal{R}[i, j, k], \min(DT_{\text{front}}, DT_{\text{side}})),$$

where DT_{front} and DT_{side} are the DT values in the generating pixels, taking the lowest radius estimate along all the views that contribute to that pixel. In this way, we obtain a good approximation of the distance of each voxel from the border of the hull.

Each voxel in the grid $\mathcal{G}$ has a starting value of 0 and each estimate of the radius has a starting value of ∞. All the nonvoted voxels will not be taken into account by the further steps of the algorithm.

4.3 Grid Processing

The low reliability of many-to-many matches may result in situations where spurious external branches stand out in the grid, especially in meshes with complex topologies or a high number of skeletal pixels in each scanline. Since we do not store an explicit $\mathcal{VH}$, but we define it implicitly using the silhouettes and the direction of projection, we perform a grid cleaning step where each cell is reprojected onto the images. If it falls outside a silhouette (i.e., if the voxel is outside the $\mathcal{VH}$ of the object), it is set to zero. In this way, we get rid of the votes that are certainly spurious and do not have to be processed as skeletal candidates. This technique results in a strong improvement of the grid quality and, due to the low number of views and the simplicity of the operations, is computationally cheap and adds a little overhead to the whole procedure. In Fig. 7, we show a comparison between a cleaned grid on the Octopus versus its uncleaned counterpart.

Once we have the *voting grid*, we can proceed with the extraction of the final curve-skeleton. The most voted voxels have the highest reliability in term of both position and radius, thanks to the higher directions of radius estimation and a higher centeredness in most views. We, then, choose to give higher weights to these voxels when computing the curve-skeleton. We extract a maximized spanning tree from the grid adopting a technique loosely based on the Ordered Region Growing (ORG) algorithm, described in [28], with several adaptations in order to fit the different kinds of structures we want to extract.

The ORG algorithm builds a tree-like representation of a 3D image (see Fig. 8), where each voxel is a node and the edges between nodes form the path between two voxels. Such paths satisfy the least minimum intensity constraint, that is, the intensity in a path between two voxels is the maximum intensity achievable. Let $min(p_{ij})$ be the minimum intensity of each voxel in the path p between voxels i and j, and let g_{ij} be the path obtained by the graph traversal from node i to node j: it is guaranteed that $min(g_{ij}) \geq min(p_{ij})$ for every other p_{ij}. This feature is highly desirable in skeleton extraction, as the voxels with higher values in the grid are the best representative of the shape, due to a higher number of voting views, while low-valued ones should be used only for connecting different high-valued regions due to their expected inaccuracy or spuriousness.

The ORG tree is built as follows: starting from a seed point G_0 (the maximum valued voxel in the grid), the region G_1 is constructed from its 26-neighborhood, and edges between the seed point and each neighbor are added to the graph. Let G_i and B_i be, respectively, the region and its boundary at the ith iteration, and s_i the maximum valued voxel in B_i. G_{i+1} and B_{i+1} are constructed from s_i by adding its unvisited neighbors, that is, those neighbors that are not already contained in G_i. New edges are created between s_i and each voxel in $(B_{i+1} \setminus G_i)$ and the process is iterated until every voxel has been included in the region.

4.4 Topological Operations

After building the ORG spanning tree, we process it to obtain the final skeleton. Let $\mathbf{N}$ be the set of all the nodes of the

spanning tree, with $\mathbf{J}$ the subset of nodes with at least three incident arcs, $\mathbf{L}$ the subset of nodes with only one incident arc, and $\mathbf{A}$ the set of all the arcs. Let $A_i \vdash N_j$ define that the arc A_i is incident on the node N_j. We define three topological operations that, applied to the spanning tree, give the final curve-skeleton. Such operations employ the definition of *zone of influence* (ZI) [29] of a node, that is, the volume defined by the maximal ball centered in it.

4.4.1 Perceptual Core Extraction

The skeleton consists of a very small subset of the spanning tree (see Fig. 8), where the majority of the voxels have been voted as a result of spurious matches. The skeleton is extracted as the set of those nodes that are *perceptually relevant*, as human interpretation does not suppose the presence of dents or bumps in a shape without evidence (as shown in [6]). We, thus, discard the tree branches not projecting medial axes onto the images. In order to be perceptually significant, a branch endpoint must *stand out* in at least one view, that is, if and only if there is no intersection between its ZI and the ZI of the joint node its branch generates from

$$\forall A_k \in \mathbf{A} : \; A_k \vdash L_i \wedge A_k \vdash J_j \\ \Rightarrow ZI(L_i) \cap ZI(J_j) = \emptyset, \qquad L_i \in \mathbf{L}, \; J_j \in \mathbf{J}.$$

See an example of this operation in Fig. 9a.

4.4.2 Branch Collapsing

Often segments of the skeleton that are supposed to converge into a single junction point meet in different joints linked each other by short arcs. We, thus, apply branch collapsing, until convergence, for internal branches as long as there is intersection between the ZI's of each junction point

$$\bigcap_{J_i \in \mathbf{J}} ZI(J_i) = \emptyset.$$

Merged joints will have as coordinates the barycenter of the junction points involved in the merging, and as radius, the minimum radius of the junction points involved in the merging. See an example of this operation in Fig. 9b.

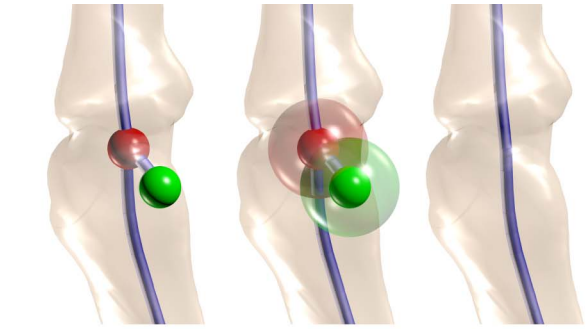
4.4.3 Loops Recovery

If a shape has genus greater than zero, a subgraph of the spanning tree cannot represent its topology. In order to recover the proper topology, we check the zone of influence of each leaf, closing a loop between all the endpoints whose zone of influence have nonempty intersection (see an example of this operation in Fig. 9c)

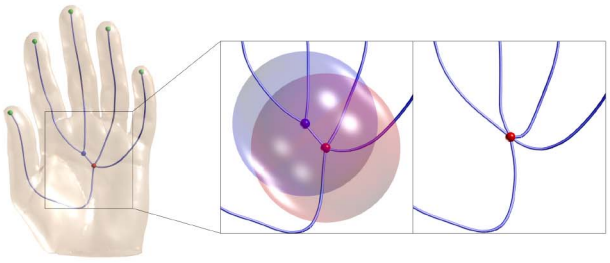
$$\forall L_i \in \mathbf{L}, \forall N_i \in \mathbf{N} \setminus \{L_i\} \quad ZI(L_i) \cap ZI(N_i) = \emptyset.$$

Note that the immediate neighbors of a leaf always satisfy the condition above, thus, to be sure that we really need to close a loop, an additional condition must be satisfied. Let p be the skeleton point whose zone of influence intersects the zone of influence of a leaf l. In the path joining p to l, there must be at least one point having empty intersection with $ZI(l)$. This condition must hold for all the leaves involved in the loop closure.

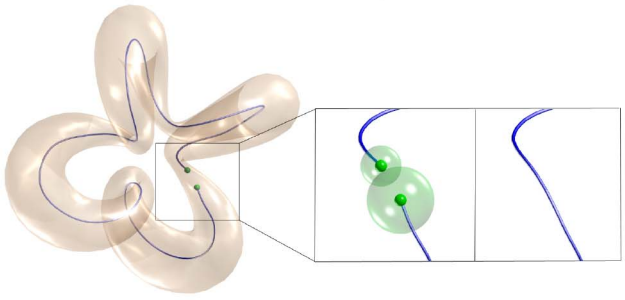
To avoid the creation of fake loops, the topological operations described above must be applied in the order we presented them.



(a) We prune a branch each time there is intersection between the ZI of its leaf and the ZI of the joint node its branch generates from.



(b) We collapse an internal branch each time there is intersection between the ZI's of its two endpoints.



(c) We close a loop each time there is intersection between the ZI of two leaves.

Fig. 9. The three topological operations.

In order to increase the visual appearance of the extracted skeletons, we end the processing applying Laplacian smoothing to the branches, averaging the position of each branch point respect to the position of its two neighbors. However, to prevent branch shortening, such operation does not involve skeleton leafs and joints.

5 Results and Comparisons

In this section, we show the results of our approach in terms of skeleton quality, robustness and timings, along with some comparisons between our method and other skeleton extraction techniques. All the timings are obtained by single-thread implementations on an iMac with Intel Core 2 Duo, 2.66 GHz, 4 GB RAM, and Ati Radeon 2600 Pro GPU.

Our approach is capable of extracting correct skeletons that accurately reflect the topology of most kinds of meshes of any genus (see Olympics in Figs. 1, 3, and 9c), even with complex morphologies like the Angiography and the Aneurysm models (see Fig. 2), where the multiview system helps in solving the ambiguities caused by projection. The results are visually appealing and satisfy most of the

TABLE 1
Comparisons among Skeletons Extracted from the Raptor Model (2M Faces) at Various Resolutions

Faces	50%	25%	5%	2.5%	0.5%	0.25%	0.1%
Max error	2.19%	0.67%	1.53%	1.32%	1.48%	1.65%	1.87%
Avg error	0.16%	0.14%	0.17%	0.19%	0.31%	0.36%	0.57%

In the first row, there are the decimation percentages. Each cell shows the Hausdorff and average error (in percentage of the length of the bounding box diagonal), showing that our output is mostly insensitive to strong decimation.

expected criteria of curve-skeletons listed in [13]. **Homotopy** is achieved through the loop recovery operation described in Section 4.4. While there is no strict guarantee of correctness, the algorithm produces good results as long as the topology of the $\mathcal{VH}$ equals the topology of the shape and the estimated radii differ slightly from the actual inscribed balls. The tree extracted from the grid satisfies **thinness** and **connectedness**, while the algorithm is **robust**, **efficient** and guarantees **invariantness to isometric transformations**. Some properties are not fully satisfied. **Centeredness** is observed, but not strictly guaranteed. **Componentwise differentiation** and **hierarchy** are obtained in most of the objects, but the view-based approach has some limitations in capturing secondary junctions in some meshes, this behavior is discussed later. **Reconstruction** cannot be fully guaranteed by monodimensional descriptors like curve-skeletons, however, the union of the maximal balls centered in each skeleton point can produce a coarse-shape approximation. Finally, our skeletons cannot satisfy the **reliability** criteria: no effort has been made toward direct shape-object visibility.

Our experiments show that the three resolution parameters (mesh, projection, and grid) have little influence on the overall results both in terms of timing and output quality. While the overhead coming from bigger silhouettes is negligible and the grid resolution affects only slightly the ORG construction, the main computational bound is given by the shape projections: the mesh resolution influences the timings as more time is needed by the GPU to rasterize the primitives of the object. Quality wise, however, our method extracts coherent skeletons from simplified meshes, so it is possible to reduce the running times by decimating high-resolution meshes with nearly no information loss (see Table 1). The method is also unaffected by changes in grid and image resolution; Fig. 10 shows how the different parameters affect the final computation; it can be noted that the skeletal structure is consistent and stable and, thus, is worth choosing low resolutions for both parameters plotted.

The projection approach makes the method very robust when used on noisy data and even on incomplete ones. It is capable of extracting skeletons from nonwatertight meshes, as soon as their visual aspect is reasonable, since the holes are not influencing the production of the silhouettes. An example of these features can be found in Fig. 12.

5.1 Extraction from Raw Point Clouds

An appealing feature of our approach is its capability of extracting curve-skeletons from raw point sets (see some examples in Fig. 14), as it needs no information about normals, thus differing from the majority of previous works in the field [30] which specifically need point clouds with normals. By performing a *morphological closing* of the projected image, it is possible to reconstruct a silhouette that allows to proceed with the skeleton extraction. To obtain an accurate silhouette, the size of the structural element must be chosen as a function of the density of the cloud, even if, for sparser point sets, narrow regions may be merged due to its higher size. However, the experimental results remain more than acceptable. The general benefits of the approach apply also to the point set case: the skeleton is noise insensitive and robust.

5.2 Comparisons

In this paragraph, we compare our approach with four techniques cited in Section 2; in the volumetric category, we compare to the Force Following algorithm [17] and Cell Complex Thinning [20], while we chose to compare to

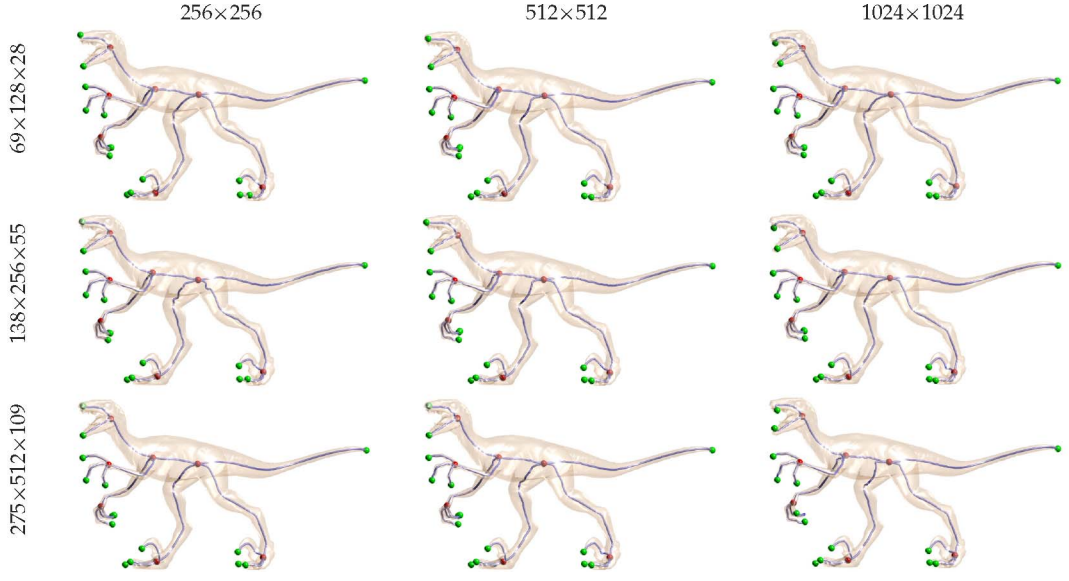


Fig. 10. Results obtained at different silhouette (column-wise) and voxel grid resolutions (row-wise): the difference, in time, between the fastest (upper left, 0.31 secs) and the slowest (lower right, 5.02 secs) is one order of magnitude.

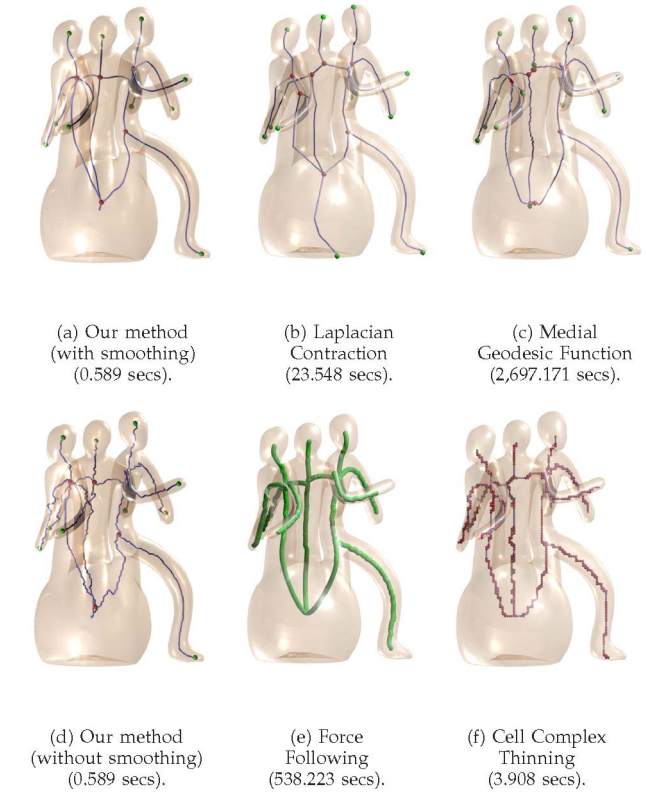


Fig. 11. Visual comparison of different curve-skeletons extracted from the Memento model (52,550 faces).

Laplacian Contraction [21] and the Medial Geodesic Function [2] in the mesh-based category (see Fig. 11). The timings, as listed in Table 2, show that our algorithm is noticeably faster than the state-of-the-art counterparts.

As for the volumetric methods, the main factor influencing timing is the thickness of each branch. The Hand mesh, for instance, even though has higher resolution than meshes like the Horse or the Octopus, is processed faster, due to the thinness of its palm and fingers. A finer voxelization, needed for higher accuracy, then results in a strong increase in computational time, while our method is insensitive to the grid dimension. Parameters are a key factor also in terms of topological coherence. The Force Following algorithm, for example, may result in great gaps between the skeletal points, and a shape-dependent parameter tuning has to be found in order to obtain a coherent skeleton. The same can be said for the Cell Complex Thinning algorithm, which can produce both 1D and 2D skeletons, depending on the parameters setting. Our skeleton is unaffected by parameter changes, and failures in the topological reconstruction can be ascribed to a low-quality estimation of the radii or to the $\mathcal{VH}$.

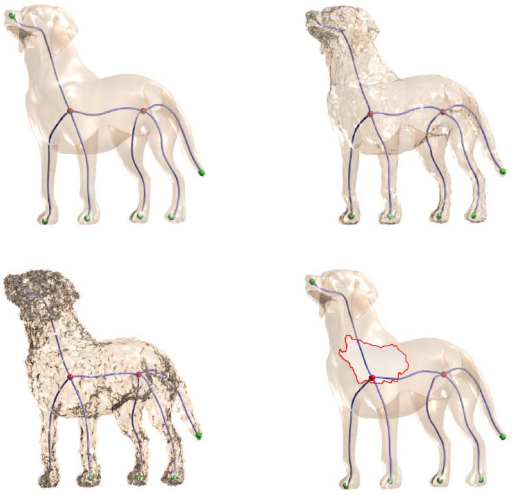


Fig. 12. The projection approach leads to robustness under noise and incompleteness. We introduced increasing artificial noise in the upper right and lower left meshes, while in the lower right one we drilled a hole, highlighted in red.

Comparisons with mesh-based methods show that our method is faster, especially for high-resolution meshes because of their dependence on the vertices. This dependence also affects the output quality when the resolution of the object is lower than a certain threshold. A coarse mesh with few vertices results in too few nodes for the skeleton or in an information loss (Fig. 15b and 15c, respectively), while our method can accurately reconstruct the descriptor as long as its visual appearance is coherent.

5.3 Limitations

Our method is intended to work on character-like meshes as animals, human figures, or cylindrical, and articulated

TABLE 2
Time Comparison (in Milliseconds) among the Different Methods Tested

Model (#Faces)	Our Method	Laplacian Contraction	Medial Geodesic Function	Force Following	Cell Complexes
Torus (1,536)	190	454	1,987	322,438	4,334
Octopus (15,054)	300	2,828	217,707	10,406	559
Dog (36,224)	341	10,675	554,937	51,500	2,123
Dino (47,955)	349	13,390	1,024,007	27,875	1,277
Hand2 (49,586)	450	18,172	1,434,891	86,109	2,282
Gargoyle (50,000)	522	19,328	897,839	180,938	4,677
Armadillo (69,184)	637	30,630	1,596,273	118,390	4,712
Horse (96,966)	585	41,765	3,294,194	49,516	2,024
Hand1 (273,060)	1,340	281,469	33,775,316	25,547	1,073

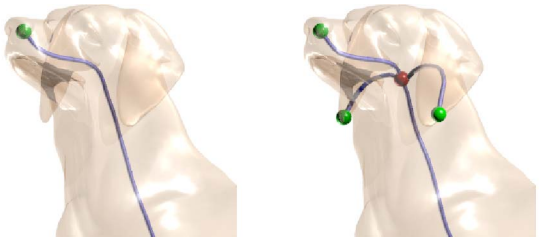


Fig. 13. Two different skeletons for the dog's head, computed, respectively, with 256×256 (left) and 512×512 (right) pixel silhouette images. It is our opinion that the loss of fine details in low-resolution projections reflects the behavior of human vision where the saliencies in an object are relative to the scale of observation [32].

objects as tools, that is, the class of shapes where the Visual Hull is a good approximation of the actual shape. The multiview system computes a skeleton of the $\mathcal{VH}$ of the shape rather than the object itself, making our algorithm unreliable for objects with no $\mathcal{VH}$ features (e.g., the cup in Fig. 16). However, a curve-skeleton may not be the best descriptor for such kind of objects in first place, where a surface skeleton like [31] would better describe the shape when no protruding cylindrical regions are found. As long as it makes sense to choose a curve skeleton for the shape (e.g., for purposes of segmentation or animation), our algorithm is able to perform well.

Being based on the shape approximation given by the visual hull, the algorithm cannot extract all the features which are overlapped in **every** projection (e.g., the buckyball molecule in Fig. 16) or which are much smaller with respect to the projection resolution. In Fig. 13, the ears of the dog remain visually close to its head in every silhouette, being ignored by our algorithm while detected in approaches like Laplacian mesh contraction. However, higher resolution projections are able to isolate small features in at least one view, solving the problem. In our opinion, this reflects the behavior of human vision where the saliencies in an object are relative to the scale of observation [32]: a distant observer will notice less features in an object than a near one, while detecting anyway the most important parts (see Fig. 10).

6 Conclusions and Future Work

In this paper, we proposed a novel approach to skeleton extraction that takes advantage of the principles of human perception and stereoscopic vision to reconstruct a 3D curve-skeleton of the visual hull of the shape starting from the 2D medial axes of the projections of the object into the image plane. The method is robust, fast, parameter free, and capable of yielding high-quality skeletons even from low-resolution meshes and raw point sets. The skeletons reflect the topology of the shape and capture the main features of the object, achieving rotational and pose invariance (see Fig. 6). The graph-like structure makes this descriptor suitable for shape matching or retrieval algorithms (such as [33] or [34]), while the correctness in branch detection suggests a possible use of the skeletons as a guidance in segmentation algorithms. The insensitivity to holes suggests also a use in skeleton driven hole filling, taking into account the average vertex-skeleton distances to perform a shape-aware filling.

Future works will include skeleton-driven reconstruction of raw point sets, as our method helps in understanding the topology of, possibly incomplete, raw point clouds and may prove itself useful for guiding the remeshing of such point sets (as opposed to other techniques like [35] or [36]). The scope of the method has been restricted to an external Visual Hull processing, but an extension to a user-specified set of viewpoints in order to capture the topology of

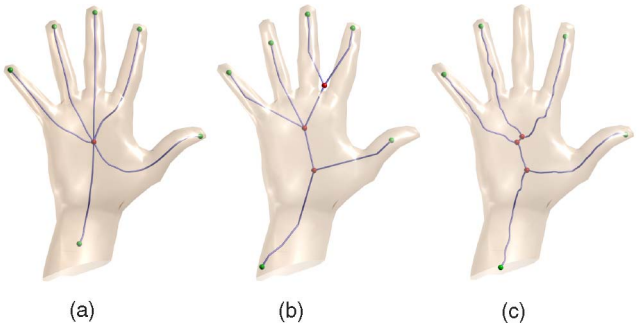


Fig. 15. Skeletons extracted from a very coarse model (1,000 faces). Primitive-based approaches cannot recover the underlying shape when too little information is available.

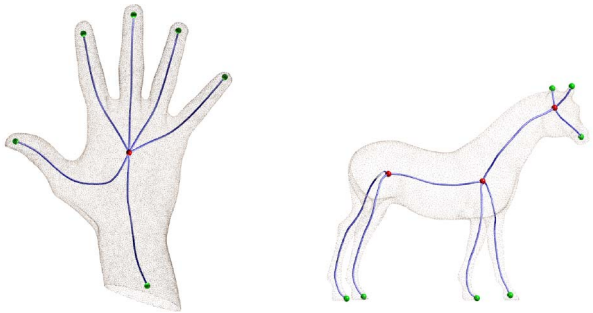


Fig. 14. With our method, we can extract skeletons even from unoriented point clouds. By performing a *morphological closing* of the cloud projections, we reconstruct a set of filled silhouettes fed to the skeleton extraction for further stages.

Fig. 16. Two examples of data sets where our method fails: a model of a mug (top), and an isosurface of the buckyball molecule (bottom).

complex meshes, or to detect features that cannot be found by an external Visual Hull, can make an interesting possibility for future improvements. Moreover, the algorithm is extremely parallelizable, from the silhouette processing [37] to matching and voting, as the viewpoints act independently from each other; future improvements could be achieved by a parallel implementation of the voting section using GPUs or cloud computing.

ACKNOWLEDGMENTS

Grid snapshots are obtained thanks to the Gorgon software (http://gorgon.wustl.edu) which has been extremely useful in 3D gray-scale images visualization. The authors used the image processing algorithms of the QGar Library (http://www.qgar.org). They would like to thank the Aim@shape project (http://www.aimatshape.net) for providing most of the shapes used for testing. Finally, they gratefully thanks the authors of [17], [21], [2], and [20] for providing them the executables of their algorithms. They also like to thank the anonymous reviewers for the interesting comment which helped them a lot in writing the final and current version of this work. Last, but non less important, they want to warmly thank all the colleagues of their laboratory, especially Stefano Marras, for the constant exchange of ideas on the subject of this work.

REFERENCES

[1] H. Blum, "A Transformation for Extracting New Descriptions of Shape," *Models for the Perception of Speech and Visual Form,* pp. 362-380, MIT Press, 1967.

[2] T.K. Dey and J. Sun, "Defining and Computing Curve-Skeletons with Medial Geodesic Function," *Proc. Eurographics Symp. Geometry Processing,* pp. 143-152, 2006.

[3] D. Marr, "Analysis of Occluding Contour," *Proc. Royal Soc. London. Series B, Biological Sciences,* vol. 197, no. 1129, pp. 441-475, 1977.

[4] D. Hoffman and W. Richards, "Parts of Recognition," *Cognition,* vol. 18, nos. 1-3, pp. 65-96, 1984.

[5] D. Marr and H.K. Nishihara, "Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes," *Proc. Royal Soc. London. Series B, Biological Sciences,* vol. 200, no. 1140, pp. 269-294, 1978.

[6] W.A. Richards, J.J. Koenderink, and D.D. Hoffman, "Inferring Three-Dimensional Shapes from Two-Dimensional Silhouettes," *J. Optical Soc. Am.,* vol. A4, no. 7, pp. 1168-1175, July 1987.

[7] J.M.H. Beusmans, D.D. Hoffman, and B.M. Bennett, "Description of Solid Shape and Its Inference from Occluding Contours," *J. Optical Soc. Am.,* vol. A4, no. 7, pp. 1155-1167, July 1987.

[8] E. Bullitt, A. Liu, and S. Pizer, "Three-Dimensional Reconstruction of Curves from Pairs of Projection Views in the Presence of Error (I. Algorithms)," *Medical Physics,* vol. 24, no. 24, pp. 1671-1678, 1997.

[9] S.M. Yoon, C. Malerczyk, and H. Graf, "3D Skeleton Extraction from Volume Data Based on Normalized Gradient Vector Flow," *Proc. 17th Int'l Conf. Central Europe on Computer Graphics, Visualization and Computer Vision,* pp. 177-182, 2009.

[10] S.M. Seitz and C.R. Dyer, "Photorealistic Scene Reconstruction by Voxel Coloring," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition (CVPR '97),* pp. 1067-1073, 1997.

[11] R.D. Murch and B.P. McGregor, "Reconstituting Object Shape and Orientation from Silhouettes," *J. Optical Soc. Am.,* vol. A9, no. 9, pp. 1491-1497, Sept. 1992.

[12] E. Leek, "Effects of Stimulus Orientation on the Identification of Common Polyoriented Objects," *Psychonomic Bull. and Rev.,* vol. 5, no. 4, pp. 650-658, 1998.

[13] N.D. Cornea, D. Silver, and P. Min, "Curve-Skeleton Applications," *Proc. IEEE Visualization,* pp. 95-102, Oct. 2005.

[14] X. Zhang, J. Liu, Z. Li, and M. Jaeger, "Volume Decomposition and Hierarchical Skeletonization," *Proc. ACM SIGGRAPH Int'l Conf. Virtual-Reality Continuum and Its Applications in Industry (VRCAI '08),* pp. 1-6, 2008.

[15] C.M. Ma and M. Sonka, "A Fully Parallel 3D Thinning Algorithm and Its Applications," *Computer Vision Image Understanding,* vol. 64, no. 3, pp. 420-433, 1996.

[16] T. Wang and A. Basu, "A Note on 'A Fully Parallel 3D Thinning Algorithm and Its Applications'," *Pattern Recognition Letters,* vol. 28, no. 4, pp. 501-506, 2007.

[17] N. Cornea, D. Silver, X. Yuan, and R. Balasubramanian, "Computing Hierarchical Curve-Skeletons of 3D Objects," *The Visual Computer,* vol. 21, no. 11, pp. 945-955, Oct. 2005.

[18] N. Gagvani and D. Silver, "Parameter-Controlled Volume Thinning," *Graphical Models and Image Processing,* vol. 61, no. 3, pp. 149-164, 1999.

[19] M.S. Hassouna and A.A. Farag, "Robust Centerline Extraction Framework Using Level Sets," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition (CVPR '05),* vol. 1, pp. 458-465, 2005.

[20] L. Liu, E.W. Chambers, D. Letscher, and T. Ju, "A Simple and Robust Thinning Algorithm on Cell Complexes," *Computer Graphics Forum,* vol. 29, no. 7, pp. 2253-2260, 2010.

[21] O.K.-C. Au, C.-L. Tai, H.-K. Chu, D. Cohen-Or, and T.-Y. Lee, "Skeleton Extraction by Mesh Contraction," *Proc. SIGGRAPH '08,* pp. 1-10, Aug. 2008.

[22] J. Cao, A. Tagliasacchi, M. Olson, H. Zhang, and Z. Su, "Point Cloud Skeletons via Laplacian Based Contraction," *Proc. Shape Modeling Int'l Conf. (SMI '10),* pp. 187-197, June 2010.

[23] A. Sharf, T. Lewiner, A. Shamir, and L. Kobbelt, "On-the-Fly Curve-Skeleton Computation for 3D Shapes," *Computer Graphics Forum,* vol. 26, no. 3, pp. 323-328, Oct. 2007.

[24] J.-M. Lien, J. Keyser, and N.M. Amato, "Simultaneous Shape Decomposition and Skeletonization," *Proc. ACM Symp. Solid and Physical Modeling (SPM '06),* pp. 219-228, 2006.

[25] A. Laurentini, "The Visual Hull Concept for Silhouette-Based Image Understanding," *IEEE Trans. Pattern Analysis Machine Intelligence,* vol. 16, no. 2, pp. 150-162, Feb. 1994.

[26] S. Petitjean, "A Computational Geometric Approach to Visual Hulls," *Int'l J. Computational Geometry & Applications,* vol. 8, pp. 407-436, 1998.

[27] G. Sanniti di Baja, "Well-Shaped, Stable, and Reversible Skeletons from the (3,4)-Distance Transform," *J. Visual Comm. and Image Representation,* vol. 5, pp. 107-115, 1994.

[28] P. Yim, P. Choyke, and R. Summers, "Gray-Scale Skeletonization of Small Vessels in Magnetic Resonance Angiography," *IEEE Trans. Medical Imaging,* vol. 19, no. 6, pp. 568-576, June 2000.

[29] L. Serino, G.S. di Baja, and C. Arcelli, "Object Decomposition via Curvilinear Skeleton Partition," *Proc. Int'l Conf. Pattern Recognition (ICPR),* pp. 4081-4084, 2010.

[30] A. Tagliasacchi, H. Zhang, and D. Cohen-Or, "Curve Skeleton Extraction from Incomplete Point Cloud," *ACM Trans. Graphics,* vol. 28, no. 3, pp. 1-9, 2009.

[31] B. Miklos, J. Giesen, and M. Pauly, "Discrete Scale Axis Representations for 3D Geometry," *ACM Trans. Graphics,* vol. 29, pp. 101:1-101:10, July 2010.

[32] A.P. Witkin, "Scale-Space Filtering," *Proc. Eighth Int'l Joint Conf. Artificial Intelligence,* vol. 2, pp. 1019-1022, 1983.

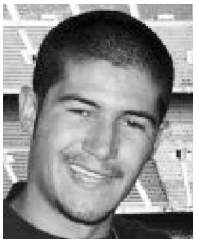
[33] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson, "Skeleton Based Shape Matching and Retrieval," *Proc. Shape Modeling Int'l Conf. (SMI '03),* pp. 130-142, 2003.

[34] O.K.-C. Au, C.-L. Tai, D. Cohen-Or, Y. Zheng, and H. Fu, "Electors Voting for Fast Automatic Shape Correspondence," *Computer Graphics Forum,* vol. 29, no. 2, pp. 645-654, May 2010.

[35] P. Alliez, D. Cohen-Steiner, Y. Tong, and M. Desbrun, "Voronoi-Based Variational Reconstruction of Unoriented Point Sets," *Proc. Fifth Eurographics Symp. Geometry Processing (SGP '07),* pp. 39-48, July 2007.

[36] H. Sheung and C.C.L. Wang, "Robust Mesh Reconstruction from Unoriented Noisy Points," *Proc. SIAM/ACM Joint Conf. Geometric and Physical Modeling (SPM '09),* pp. 13-24, 2009.

[37] S.J. Shyu, T. Chou, and T.L. Chia, "Distance Transformation in Parallel," *Proc. Workshop Combinatorial Math. and Computation Theory,* pp. 298-304, 2006.



Marco Livesu received the master's degree magna cum laude from the University of Cagliari, Italy, in September 2010. Currently, he is working toward the PhD degree in the Department of Computer Science, University of Cagliari. His research interests include computer graphics, shape description, shape deformation and computer vision.

Fabio Guggeri received the MS degree magna cum laude from the University of Cagliari where he has been working toward the PhD degree since 2009. The subjects of his studies are the topological aspects of the triangle meshes and investigating the usage of stable methods from image processing in the solution of 3D problems.

Riccardo Scateni is an associate professor at the Department of Mathematics and Computer Science, University of Cagliari. He joined the University after a long period spent at CRS4 in Cagliari. He has worked in Scientific Visualization for a long period and his current research interests are in the field of Computer Graphics, especially Geometry Processing, and Interaction Design.