Pacific Graphics 2012
C. Bregler, P. Sander, and M. Wimmer
(Guest Editors)

Volume 27 (2008), Number 1

Analytic Curve Skeletons for 3D Surface Modeling and Processing

Jean-Marc Thiery Bert Buchholz Julien Tierny Tamy Boubekeur

Telecom Paristech - CNRS/LTCI, Paris, France

Abstract

We present a new curve skeleton model designed for surface modeling and processing. This skeleton is defined as the geometrical integration of a piecewise harmonic parameterization defined over a disk-cylinder surface decomposition. This decomposition is computed using a progressive Region Graph reduction based on both geometric and topological criteria which can be iteratively optimized to improve region boundaries. The skeleton has an analytical form with regularity inherited from the surface one. Such a form offers well-defined surface-skeleton and skeleton-surface projections. The resulting skeleton satisfies quality criteria which are relevant for skeleton-based modeling and processing. We propose applications that benefit from our skeleton model, including local thickness editing, inset surface creation for shell mapping, as well as a new mid-scale feature preserving smoothing.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

1. Introduction

Curve skeletons are 1-manifold networks capturing – to a certain extent – the topology and the geometry of a wide class of 3D objects. Their abstraction power has been exploited in various applications requiring analyzing 3D shapes, such as shape matching in visual search [TS04], shape registration or animation [YSZ06]. Basically, they are defined as graphs, with *curved* edges linking 3D nodes and a spatial embedding mimicking the input shape.

Over the many algorithms which have been introduced to compute such skeletons, the primary goal has always been to provide a skeleton with very smooth curves while avoiding any excess of nodes. This stems from the fact that most applications making use of skeletons exploit their coarseness. Unfortunately, in many cases, existing solutions do not model explicitely the relationship existing between a piece of surface and a bone of the skeleton.

We introduce a curve skeleton model, together with its construction algorithm, which embeds such a relationship by the means of a piecewise smooth cylindrical parameterization. Our basic observation is that, up to the desired level of accuracy, a given curve bone should map to a *topological cylinder* portion of the surface, while extremal nodes should map to *topological disks* of the surface.

This has immediate consequences on the potential applications which can benefit from such a meta-structure. In particular, we demonstrate how shape editing and processing can make use of a *parameteric* skeleton model enriching the 3D surface: for such applications, our skeleton acts as a 1-dimensional domain onto which the surface is expressed. This layout facilitates interactive editing methods such as local shape thickness control or *inset* creation emulating a "geometric peeling" process (e. g. for *Shell Mapping* [PBFJ05]). Moreover, as the skeleton's embedding reflects faithfully the shape's one, we can express the object's geometry as a signal which is parameterized over the skeleton and can redefine surface processing w.r.t. a "skeletal" basis. In particular, we introduce a new feature preserving mesh filter extending the *bilateral* one [JDD03] to better preserve mid-scale surface structures.

Overview. While previous approaches are based on Laplacian Contractions or Medial Axis (Section 2), our method (Section 3) decomposes the surface in regions which can be parameterized smoothly onto single curves, i. e. *topological cylinders* (Section 4). In order to guarantee curve bones with regularity inherited from the surface itself (Section 5.1), we generate the actual bones' geometry by integrating the surface one along harmonic maps (Section 5.2) automati-

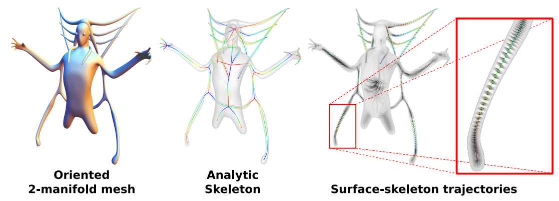

Figure 1: ***Analytic Curve Skeleton*** *of a polygonal surface mesh. Smoothly varying surface-skeleton trajectories offer a versatile layout for shape editing and processing.*

cally constructed from the regions boundaries,with remaining (disk-) regions projecting to skeleton extremities. We also introduce an iterative optimization process inspired by Centroidal Voronoï Tessellation (Section 6) which improves the surface decomposition. Compared to other curve skeletons (Section 7), the regularity and the parametric nature of the skeleton model makes it suitable for interactive modeling and processing of shapes (Section 8): we demonstrate its use for modeling applications with shape editing and *inset surface* design tools, as well as the benefit it brings to geometry processing with a *skeletal* feature-sensitive mesh smoothing operator.

Contributions. The main contributions of this paper are:

- a curve skeleton model with an analytical form and the following properties:

  Smoothness: each bone is a smooth 1D-curve;
  Harmonicity: each bone embeds a harmonic map with a topological cylinder on the surface;
  Regularity: each bone's geometry corresponds to its surface region, where *trajectories* (i. e. , surface-skeleton vector field) have good differential properties;

- an efficient algorithm to construct it from a surface mesh;
- a disk-cylinder surface decomposition structuring harmonic maps over the surface which encodes the surface-skeleton correspondances;
- a closed-form geometric embedding of the skeleton, with regularity derived from the surface.

We also illustrate skeletal modeling and processing with three practical contributions:

- an interactive shape modeling tool allowing to edit shape thickness locally;
- an inset surface modeling tool exploiting the surface-skeleton inter space with application to Shell Mapping;
- a feature-sensitive mesh filter based on skeleton-aware shape thickness.

2. Related Work

Extracting a curve skeleton from a given shape is a well studied problem. For a recent overview, we refer the reader to the survey of Cornea et al. [CSM07] and focus on previous methods which are relevant in our context.

Reeb Graphs. A Reeb Graph [Ree46, PSBM07, TVD08, FK97, LCT11] is essentially a data structure for understanding and representing the topology of scalar functions on shapes. Used with euclidean space related functions (e. g. the Z function), it allows to compute a skeleton with a meaningful geometrical embedding by contracting isocontours of such a function into a single point.

Contraction Skeletons. Curve skeletons have also been defined as the result of surface contraction processes. Au et al. [ATC*08] use the Laplacian operator on the mesh coordinates to shrink the shape progressively until they obtain a (almost) zero-volume mesh. This contracted mesh is understood as a skeleton and is very smooth. At this point, it is naturally attached to the surface as their is a one-to-one correspondance between the vertices on the surface and the vertices on the skeleton mesh. To get a 1 dimensional skeleton, they collapse edges until they obtain the wanted structure. One vertex of the skeleton then corresponds to a set of vertices of the mesh, which exhibit most of the time a cylindrical shape. However, no parametric relationship is directly available at the end of the process and the pointwise vertex-skeleton relationship lacks control on smoothness and regularity.

Similar approaches [TZCO09] have been developed to handle polygon soups and pointsets, allowing to use the skeleton for surface reconstruction and hole filling.

Alternatively, Sharf et al. [SLSK07] cast the problem of curve skeleton generation as a deforming model one, using the midpoint of advancing fronts on the surface to draw the skeleton. Although applicable to non uniform 3D point clouds, this method is sensitive to variations in sampling.

Geometric Cylinders Decomposition. As mentioned earlier, curve bones can be interpreted as central structures of *topological* cylinders. Finding *geometric* cylinders inside 3D data [RvdH, BF81, CG01, BF06] has been studied to discover semantic shape information from raw geometry such as 3D point clouds. In this context, geometry-based segmentation techniques, [RBG*09, KLT05, KT03, WK05] are very efficient at finding cylinder-like clusters. Unfortunately, as we are interested in **topological** cylinders – with geometry often differing significantly from conventional cylinders – we have to define a new algorithm with cluster topology as the major constraint.

Topology-based Decomposition. Topological disk surface decomposition is a classical process in surface parameterization [SWG*03, SSGH01, SCOGL02, LPRM02, ACSD*03]. The idea is to segment the surface into several disks, with the geometry of the disk controlled by an error function. The topology of the decomposition can be assessed by computing the Euler characteristic and inspecting the adjacency configuration of the region boundaries [ES94, DZM08]. In our context, we seek for regions which are homotopy equivalent to cylinders, which is a different problem than finding geometric cylinders to approximate the shape. While

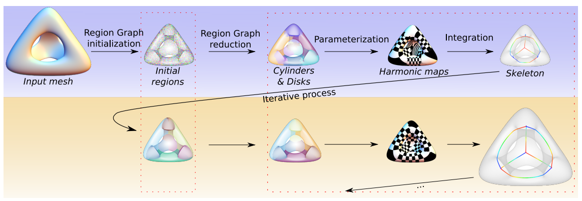

Figure 2: ***Algorithm overview:*** *the skeleton creation process (top), followed by optional optimization (bottom).*

the latter can effectively be addressed with variational methods [CSAD04, YLW06, WK05], we propose a new approach, suitable to find topological cylinders and which takes inspiration from both adaptive surface optimization methods [Hop96] and hierarchical segmentation [Sha08].

3. Our Algorithm at a Glance

Given a triangulated oriented 2-manifold mesh, our algorithm outputs a skeleton composed of *articulations* (3D points) and curve bones linking them, together with a disk-cylinder surface decomposition and harmonic maps defining the relationship between both:

$$\text{Input mesh} \longmapsto \begin{cases} \text{Analytic Skeleton} \\ \text{Cylinders and Disks Decomposition} \\ (b,u,\theta)\text{-Parameterization} \end{cases}$$

Our algorithm starts by defining an initial segmentation made up of topological disks and applies 3 processes (Fig. 2):

1. the dual graph of the segmentation is reduced so that the different regions merge together and create "well-shaped" cylinders;
2. an harmonic parameterization is computed for each region;
3. a bone is generated from each harmonic map.

4. Cylinder-Disk Segmentation

Any 2-manifold can be tessellated with an infinite number of topological disks. In order to find a reduced set of large cylinders and disks, we adopt a bottom-up strategy where we start from many regions and progressively aggregate them to form cylinders and larger disks.

4.1. Bottom-Up Strategy

The segmentation is initialized by defining a large number of small regions on the surface. These regions are connected sets of triangles and the so-defined clustering must satisfy the *closed-ball property* [ES94], i. e. each region is homotopy equivalent to a disk and the (non-empty) intersection between 2 partitions is a closed 1-ball, i. e. a single open curve [DZM08]. For the sake of simplicity, the algorithm can be safely initialized with one triangle per region to guarantee this property. Alternatively, a space subdivision structure (e. g. octree) can be adaptively refined until meeting such a condition in each leaf, with leaf triangle sets becoming initial regions.

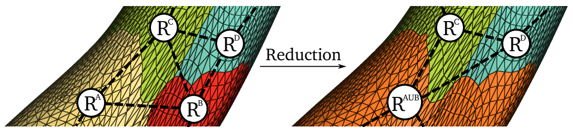

Figure 3: ***Region graph reduction****: collapsing an edge (black dot lines) is equivalent to merging two regions.*

The region adjacency graph (or *Region Graph*) is defined as a set of nodes (one for each initial region) and a set of edges (one for each pair of regions sharing at least one triangle edge). The region graph provides a single *reduction* operator by the means of successive edge collapse merging 2 nodes/regions and removing one edge of this graph (see Fig. 3).

Once the *region graph* is initialized, all possible edge collapses are pushed into a priority queue, with the priority defined for the collapse of edge $\{A,B\}$ as a 2-term value $\{e_T(A,B), e_G(A,B)\}$:

- e_T is defined on a discrete scale of *topological classes* and models the topological change induced by the collapse,
- e_G is defined on a continuous scale and models the geometric cost of the collapse.

Edge collapses are ordered w.r.t. increasing topological cost e_T, and then w.r.t. increasing geometrical cost e_G when two candidates have equal e_T.

When processing an edge-collapse, the two nodes of the edge are merged and their neighborhood in the graph is updated. Last, the two nodes are flagged as *merged* so that they cannot be merged later in the queue.

At this step, new candidate edge collapses can be pushed into the queue if they pass a *discarding test*, which depends on both, topology and geometry (see Section 4.4).

4.2. Topological Priority Term

The topological priority term e_T of a candidate collapse of edge $\{A,B\}$ depends only on the topology of the region sets $\{R^A, R^B, R^A \cup R^B\}$. We summarize the different cases in Table 1.

4.3. Geometric Priority Term

In order to compute the geometric cost of a collapse, we use *proxies* [CSAD04] to capture the shape of regions. A proxy P^A is a triplet $(\mathbf{p}^A, \mathbf{n}^A, s^A)$ representing a surface region R^A, with $\mathbf{p}^A$ its barycenter, $\mathbf{n}^A$ its average normal and s^A its area. The collapse operator requires to build a new proxy for the emerging region. When $R^X = \bigcup_i t^i$ is the union of triangles t^i with barycenters p_i, normals n_i and areas s_i, then $s^X = \sum_i s_i$, $p^X = \sum_i s_i \cdot p_i / s^X$, and $n^X = \sum_i s_i \cdot n_i / s^X$. The expressions of p^X and n^X correspond to the natural discretization of the averaging of the functions p and n defined by surfacic integration on X. Therefore, if $R^X = \bigcup_i R^i$, we obtain P^X by additivity on $\{P^i\}$.

Let $\langle \cdot | \cdot \rangle$ (resp. $\langle \cdot \otimes \cdot \rangle$) be the dot (resp. cross) product between two vectors. For each topological priority term k, we define a specific geometric cost function E_k ordering candidate collapses which are topologically equivalent and affect $e_G(A,B) = E_{e_T(A,B)}(A,B)$. Let $R^X = R^A \cup R^B$. The definition of E_0 must favor hole filling while penalizing emerging flat "caps" at cylinder's extremities:

$$E_0(A,B) = 1 - \|\mathbf{n}^X\| \qquad (1)$$

E_1 must favor the creation of cylinders while E_4 should detect the possible union of cylinders leading to valid ones, we model both errors using a low mean normal:

$$E_1(A,B) = E_4(A,B) = \|\mathbf{n}^X\| \qquad (2)$$

At this point, we do not aim at defining an orientation for cylindrical regions. Instead we only exploit the fact that for ideal cylinders, the average of all normals is the null vector (which is equivalent to minimizing E_1).

Geometric errors E_2 (*Grow Disk*) and E_3 (*Grow Cylinder*) are defined as regions' standard deviation of which the minimum favors sphere-like shapes:

$$E_2(A,B) = E_3(A,B) = \sum_i s_i \cdot \|\mathbf{p}_i - \mathbf{p}^X\|^2 / s^X \qquad (3)$$

However, without normals, this error is blind to orientation and trades concavity for convexity. As a consequence, it cannot be used alone to drive the algorithm, but gives satisfying results in conjunction with the validity predicate, in particular the *Quasi-Star* estimator (see below).

4.4. Validity Predicate

When an edge is collapsed, two regions merge and new potential collapses occur. However, a large number of them lead to worse configurations with complicated topology and/or bad geometry. We prevent such degeneration using a *Validity Predicate* $\mathcal{V}$ indicating if an edge $\{A,B\}$ can be inserted in the priority queue. The topological part of this predicate has been mentioned before (see Table 1): any edge corresponding to a collapse with an infinite topological error e_T is ignored.

$$\text{(i)}\ \ e_T(A,B) \notin [0,4] \Rightarrow \mathcal{V}(A,B) = \mathit{false} \qquad (4)$$

The geometrical part of $\mathcal{V}$ checks e_G to prevent segmentation discrepancy. In particular, "ideal caps" should be preserved at cylinders' extremity. These cases being equivalent to a semi-sphere in $\mathbb{R}^3$, the mean normal over them should equal $1/2$ when a "Fill Hole" edge-collapse is considered:

$$\text{(ii)}\ \ \begin{cases} e_T(A,B) = 0 \\ E_0(A,B) < 1/2 \end{cases} \Rightarrow \mathcal{V}(A,B) = \mathit{false} \qquad (5)$$

Note that $1/2$ is **exactly** the length of the averaged outward normal of any semi-sphere (i.e. any positive radius).

Classical *meaningful decomposition* metrics [KS06] are too restrictive for generalized cylinders, which may not be convex. Instead, we take into account orientation and penalize tunnels to be interpreted as cylinders by defining a *Quasi-Star* estimator $\mathcal{Q}$ on the union of two regions:

$$\mathcal{Q}(R^X) = \sum_i s_i \cdot \langle \mathbf{p}_i - \mathbf{p}^X | \mathbf{n}_i \rangle \qquad (6)$$

and add a third test to our predicate :

$$\text{(iii)}\ \ \begin{cases} e_T(A,B) \notin \{3,4\} \\ \mathcal{Q}(R^A \cup R^B) < 0 \end{cases} \Rightarrow \mathcal{V}(A,B) = \mathit{false} \qquad (7)$$

Since we seek for *generalized* cylinders, we do not use the Quasi-Star estimator when a "Grow Cylinder" or a "Merge Cylinders" edge-collapse occurs.

If the three tests fail, then $\mathcal{V}$ is true and $\{A,B\}$ is inserted into the priority queue.

Topology			Name	e_T
R^A	R^B	$R^A \cup R^B$		
Cyl.	Disk	Disk	*Fill Hole*	**0**
Disk	Disk	Cyl.	*Create Cylinder*	**1**
Disk	Disk	Disk	*Grow Disk*	**2**
Cyl.	Disk	Cyl.	*Grow Cylinder*	**3**
Cyl.	Cyl.	Cyl.	*Merge Cylinders*	**4**
Other cases			*Skip*	∞

Table 1: *Topological priority term as a function of the topological change. Candidate edge collapses are ordered in the priority queue by increasing topological priority term.*

4.5. Segmentation Results

In practice the entire process can be sped up when starting from a coarse initial segmentation (e.g., octree partition). Fig. 4 shows several examples of surface decomposition with our algorithm.

5. Skeletonization

With a disk-cylinder surface decomposition in hand, we can now generate the curve bones. Each cylinder is contracted to

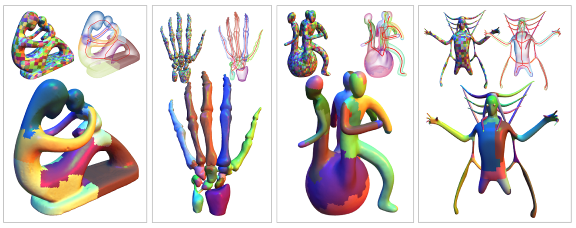
Figure 4: Region Graph *construction results.*

a smooth 1-curve and each disk (which ideally corresponds to a cylinder "cap") is contracted to a single point, located at the extremity of a bone. We propose to conduct this process using an harmonic map on the regions.

5.1. Harmonic Parameterization

We note $\triangle$ the Laplace operator onto the manifold surface. We use the approximation of [PJP93] to solve the linear systems 8, 9 and 11 in a least-squares sense onto the mesh vertices. We use the CHOLMOD library for that purpose.

(u,θ) **Cylinder Parameterization.** Considering Fig. 5, the u-parameterization of a cylinder is obtained by solving the linear system:

$$\begin{cases} u(\mathbf{v}) = 0 & \forall \mathbf{v} \in U_0 \\ u(\mathbf{v}) = 1 & \forall \mathbf{v} \in U_1 \\ \triangle u(\mathbf{v}) = 0 & \forall \mathbf{v} \notin U_0 \cup U_1 \end{cases} \tag{8}$$

The θ-parameterization is obtained by first cutting the cylinder with a path that follows the best $\vec{\nabla} u$, and then solving a similar linear system to Eq. 8:

$$\begin{cases} \theta(\mathbf{v}) = 0 & \forall \mathbf{v} \in \theta_0 \\ \theta(\mathbf{v}) = 2\pi & \forall \mathbf{v} \in \theta_{2\pi} \\ \triangle \theta(\mathbf{v}) = 0 & \forall \mathbf{v} \notin \theta_0 \cup \theta_{2\pi} \end{cases} \tag{9}$$

To obtain the cut, a cost function is attributed to each edge $\mathbf{e}$ of the cylinder that calculates the deviation between the orientation of the edge and the normalized gradient of u $\vec{\nabla}_1 u = \vec{\nabla} u / \|\vec{\nabla} u\|$:

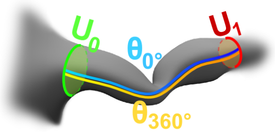

Figure 5: *Cylinder*

$$c(\mathbf{e}) = \|\langle \vec{\nabla}_1 u(\mathbf{v}_0) + \vec{\nabla}_1 u(\mathbf{v}_1) \bigotimes \vec{\mathbf{e}} \rangle\| \tag{10}$$

$\mathbf{v}_0$ and $\mathbf{v}_1$ being the two extremities of $\mathbf{e}$. We find the path – between the two borders of the cylinder – which minimizes this functional using Dijkstra's algorithm.

(u,v) **Disk Parameterization.** For (topological) disk regions, we directly compute a planar parameterization. Let B be the border of such a given disk. We start by defining a θ-parameterization of B before computing (u,v)-coordinates by solving the linear system:

$$\begin{cases} u(\mathbf{v}) = \cos(\theta(\mathbf{v})) & \forall \mathbf{v} \in B \\ v(\mathbf{v}) = \sin(\theta(\mathbf{v})) & \forall \mathbf{v} \in B \\ \triangle u(\mathbf{v}) = 0 & \forall \mathbf{v} \notin B \\ \triangle v(\mathbf{v}) = 0 & \forall \mathbf{v} \notin B \end{cases} \tag{11}$$

5.2. Bones' Geometry

Since one of the main applications we target is skeletal signal processing, we need to make sure that the skeleton's geometry we obtain is not only visually smooth, but has mathematical guarantees about its smoothness. We present our construction below, and discuss it in Section 5.3.

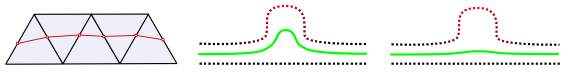
(a) Iso-Curve (b) Spatial Weight (c) θ-Dist. Weight

Figure 6: *A cylinder (dot line), its θ-distortion (dot line color) together with its bone (green line). The integration of the u-iso-value (right, Eq. 12) yields almost no noise.*

Starting from the cylinder parameterization, we derive its bone's analytical form $b : [0;1] \to \mathbb{R}^3$ as the mean of u-iso-values. We compute it by contouring a given u-iso-value as a polygonal iso-curve made of iso-edges crossing the triangle edges (see Fig. 6(a)). We then consider a weighted combination of these iso-edges' centers. Our experiments show that accounting for distortion is the foremost concern. Therefore, the analytical form of b boils down to:

$$b(u) = \frac{\int_0^{2\pi} p(u,\theta) d\theta}{\int_0^{2\pi} d\theta} \tag{12}$$

This simple strategy is insensitive to noise and tends to define the most "natural" bone. See Fig. 6 for a comparison with spatial weighting. Note that the spatial weighting is used in almost every Reeb Graph embedding technique. In practice, we represent bones using interpolating B-Splines with control points sampled from Eq. 12.

5.3. Bones' Regularity

Fig. 7 (left) recalls the definition of a 2-manifold and the regularity of a function defined over it: μ is defined over M and locally maps each open set of M to an open set of $\mathbb{R}^2$. These sets recover M in a way that the transition map $\mu_{|U} \circ \mu_{|V}^{-1}$ from $U \cap V \subset \mathbb{R}^2$ to $\mathbb{R}^2$ is regular, for each open set U and V such that $U \cap V \neq \emptyset$. The regularity of M as a 2-manifold is defined as the one of the transition maps. Therefore the function f can be seen through local coordinates given by μ, and $\hat{f}$ is defined by $f = \hat{f} \circ \mu$ everywhere. The regularity of the function f is defined as the one of $\hat{f}$.

In our case, the regularity of the function b depends on the regularity of its related cylinder surface mesh and on the regularity of the (u,θ)-parameterization. In fact, b has the same regularity as the position function on the mesh seen through the (u,θ)-coordinates $\tilde{p} = \hat{p} \circ \hat{\lambda}^{-1}$ since $b(u) \sim \int_{\theta=0}^{2\pi} \tilde{p}(u,\theta) d\theta$ (see Fig. 7).

$$\begin{tikzcd} M \arrow[rr, "f"] \arrow[dr, bend right, "\mu"] & & Y \\ & \mathbb{R}^2 \arrow[ur, bend right, "\hat{f}"] & \end{tikzcd} \qquad \begin{tikzcd} M \arrow[rr, "p"] \arrow[dr, bend right, "\mu"] \arrow[ddr, bend right, "\lambda"] & & \mathbb{R}^3 \\ & \mathbb{R}^2 \arrow[ur, bend right, "\hat{p}"] \arrow[d, "\hat{\lambda}"] & \\ & \mathbb{R}^2 \arrow[uur, bend right, "\tilde{p}"] & \end{tikzcd}$$

Figure 7: ***Left:*** *The regularity of a function f on a manifold is defined as the one of $\hat{f}$.* ***Right:*** *The position function p is defined over the 2-manifold M. λ is the (u,θ)-coordinates function defined over M.*

5.4. Connecting Bones

Once each bone is created, we obtain a *disconnected* skeleton (see Fig. 8). Although one can enforce additional constraints on the (u,θ)-parameterization to artificially join their extremities by construction, this usually leads to lower skeleton quality. Instead, we explicitly connect the independently constructed bones by joining the extremities of the ones related to adjacent regions on the triangular mesh.

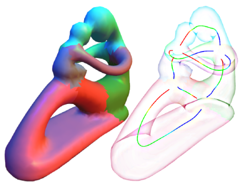
Figure 8: *Skeleton*

Given such an *articulation*, the adjacency graph $\mathbb{AG}$ of the corresponding cylinders is built. All related bones' extremities are merged, introducing a minor deviation from the original bones: the splines defining them with border U_0 (resp. U_1) are updated with their first (resp. last) control point being relocated at the barycenter of the previous extremities. The induced skeleton deformation can be controlled by setting the domain of the original spline to be unchanged.

6. Optimization

In the previous sections, we presented an algorithm to find topological cylinders in shapes and a method to construct a curve skeleton from them. For some applications, it is also important to provide a surface decomposition with well shaped, smooth boundaries We now present the third (optional) component of our approach to iterate over the process, in order to obtain such smooth boundaries.

6.1. Surface-restricted skeletal Voronoï diagram

Inspired by the Lloyd relaxation algorithm, we propose to optimize the smoothness of our segmentations in an iterative process. Given the curve skeleton, we compute a new surface decomposition by associating the index of the closest bone of the skeleton to each vertex of the mesh. This corresponds to euclidean space subdivision based on individual bones, therefore defining a *Surface-Restricted Skeletal Voronoï Diagram* (*SRSVD*). This *SRSVD* segmentation is interleaved with our skeleton construction technique in an iterative global relaxation (see Fig. 9). On the contrary to the initial step, the segmentation of the following relaxation steps is not necessary made of topological disks only, but contains topological cylinders as well. Since the *SRSVD* segmentation does not guarantee to preserve all the cylinders generated at previous steps, nor prevent from introducing regions with different topology, we adopt a parametric approach to propose a topology-preserving solution.

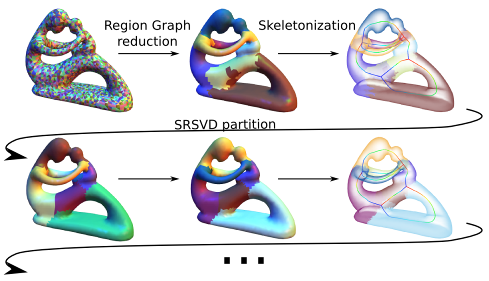

Figure 9: *Each row corresponds to one step of our relaxation algorithm. The input for the next step is defined using* SRSVD *segmentation.*

6.2. Parametric skeletal Voronoï diagram

To guarantee a valid segmentation from the *SRSVD*, we parameterize the projection process and consider cylinders/bones at previous step. More precisely, we intersect the Skeletal Voronoï Diagram with the model's surface parameterized by sliding the surface towards its projection onto the previously defined bone: $\forall \mathbf{v} : \hat{\mathbf{v}}_\alpha = \alpha \cdot b(\mathbf{v}) + (1-\alpha)\cdot \mathbf{v}$. By taking $\alpha \in]0,1[$, the process of smoothing the boundaries is slower, but we have guarantees over the topology of the resulting segmentation. Defining an optimal value for α at each iteration of the algorithm remains an open problem. In practice, we sample $[0,1]$ uniformly, and take the smallest value that preserves the topology of the resulting segmentation.

7. Results and Discussion

7.1. Performances

Results are presented in Fig. 10 for various shapes. The skeletons were obtained with 2 iterations in average. Our curve skeletons faithfully reproduce shape topology and geometry (through regularity). We have implemented the entire algorithm in C++ on an Intel Core2 Duo running at 2.5 GHz with 4 GB of main memory. We report detailed timing for various meshes in Table 2.

7.2. Properties

Our curve skeleton model satisfies major curve skeleton quality criteria [CSM07]:

Thin our skeleton is analytically defined as a 1-dimensional object

Component-wise this property can be assessed visually on the different illustrations of this paper

Model		**Timing (in sec.)**		
Name (Triangles)	Patches	Reg. Graph (Edge Coll.)	Skel.	Total
Armadillo (345 944)	9 370	11.0 (66k)	9.9	21.8
Neptune (56 112)	56 112	31.0 (295k)	1.2	32.7
	2 730	2.6 (20k)	1.1	4.0
Memento (52 550)	52 550	30.2 (290k)	1.2	32.9
	3 897	3.6 (29k)	1.1	5.0
Hand (23 464)	23 464	11.7 (132k)	0.4	12.9

Table 2: *Analytic Curve Skeleton generation time.*

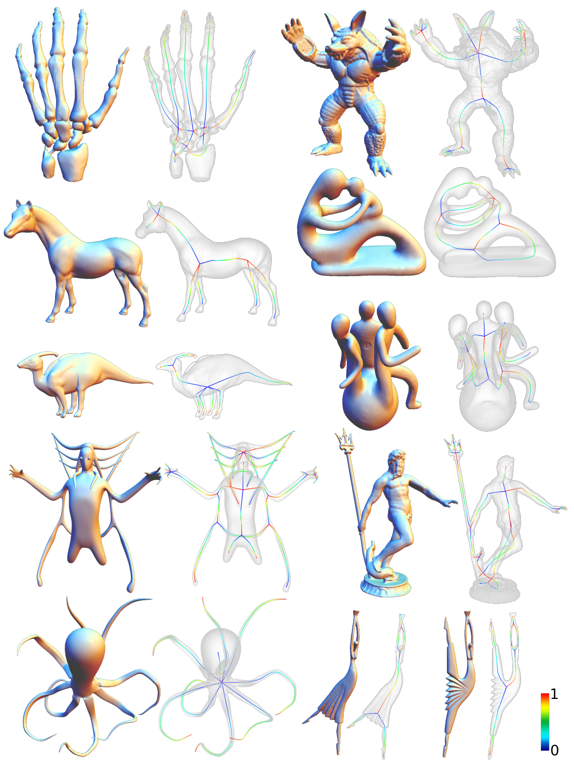

Figure 10: *Analytic Curve Skeletons of various shapes. The rainbow color code display the* ***parameterization*** *values from 0 (red) to 1 (blue).*

Efficiency the bottleneck of the algorithm is the region graph reduction, which can be significantly sped-up using prepartitioning (see Table 2)

Nonetheless, we stress the fact that *smoothness* and *regularity reproduction*, although not mentioned in Cornea et al. 's list [CSM07], are fundamental properties for skeletal geometry editing and processing. We refer to Section 5 regarding good properties of our skeleton in this context.

7.3. Comparison

We compare our skeleton model to Laplacian skeletons [ATC*08] (results obtained with the authors' implementation, using default values) and Reeb Graphs in Fig. 11, as they use the same input, and are commonly used in different applications.

Laplacian skeletons are defined as a contraction of the input mesh, contraction made by successive edge-collapse. As a result, one vertex of the skeleton corresponds to a collection of mesh vertices, usually describing a cylinder as the authors underline in their paper (Fig. 4 in [ATC*08]). But this property is not enforced at the core of the algorithm, and we observe that a **lot** of bones in the obtained skeletons do not represent any cylindrical shape of the input geometry (see Fig.11, second row). This makes Laplacian skeletons not suitable for the applications we target (e.g., one dimensional model representation for skeletal signal processing, see Section 8).

Reeb Graphs [Ree46, PSBM07] are defined as the contraction of scalar function iso-contours over a surface. By definition, Reeb Graphs shares with our skeleton the property that one bone of the reconstructed skeleton corresponds to one generalized cylinder in the input geometry. However, Reeb Graphs' geometry depends heavily on the chosen scalar function, which in general does not reproduce a faithful component-wise, surface-related geometry. See Fig. 11, third and fourth rows, for examples of skeletons extracted as the Reeb Graph of different functions. The existence of at least one maximum and one minimum of the function implies directly that the extracted skeleton contains at least two extremities, which limits the space of possible structures Reeb Graph can model (see "Fertility"-2ndcolumn in Fig. 11 for an example).

Criterion	AS	LS
Analytical form	+	-
Attachement to the surface	+	-
Post smooth editing of the skeleton geometry	+	-
Robustness on non-organic/noisy shapes	-	+

Table 3: *Our model (AS) compared to [ATC*08] (LS).*

7.4. Limitations

First, one limitation of our algorithm is its restriction to manifold meshes: extension to point clouds and polygon soups is

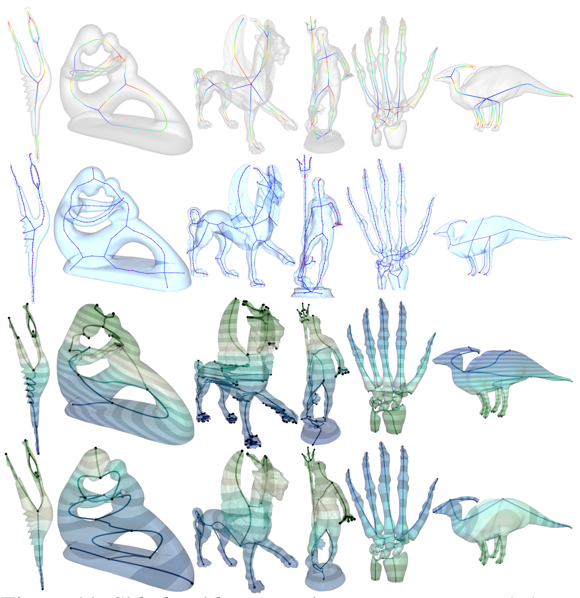

Figure 11: ***Side-by-side comparison****.* ***Top row:*** *Our skeleton model.* ***Second row:*** *Laplacian Skeleton [ATC*08].* ***Third row:*** *Reeb Graph on 'Z' function.* ***Bottom row:*** *Reeb Graph on harmonic function with user prescribed constraints.*

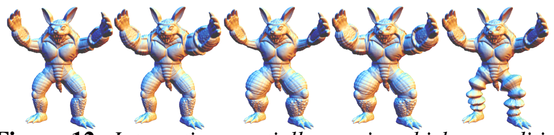
Figure 12: *Interactive, spatially varying thickness editing using parametric displacement mapping from our skeleton .*

an interesting direction for future work. Second, our technique can suffer from poorly sampled surfaces with high genus during the region graph *local* reduction. Nevertheless, remeshing or *global* reduction can overcome this issue. Third, *medialness* is not enforced in our algorithm, and some parts of the skeleton may still intersect in some places the mesh. This can have an impact on the presented applications. Fourth, although having too many branches is usually not suitable for the typical applications we target (e.g., requiring a *smooth* base domain), it could be interesting to allow the user to refine the skeleton in a multiresolution fashion, to favor particular locations, while keeping coarse the others. Last, as with all curve skeletons, there are shapes which are not suitable for skeletal analysis or processing and which would benefit more from alternative internal structures (e.g., medial surfaces).

8. Applications

Beyond animation, recognition and visualization, geometric modeling and processing have strongly motivated our work. We illustrate the effectiveness of the Analytic Curve Skeletons with 2 examples for such applications.

8.1. Skeletal Shape Modeling

Many applications can benefit from a smooth definition of the "near inside" and "near outside" of a shape, information usually given by the normal of the surface. In Fig. 12 we present the Armadillo model and several variations we created simply by defining a displacement function $d_b(u)$ on each bone b and sliding each vertex $\mathbf{v}$ of the mesh along its trajectory by the corresponding value $d_{b_\mathbf{v}}(u(\mathbf{v}))$.

The trajectories from the vertices to their projection on the skeleton offer a more consistent way to use displacement maps than the normals of the mesh which, in case of large displacements values, often lead to (local) self intersections of the surface. The user can also intuitively control the induced deformation "along" components.

8.2. Inset Surfaces Modeling

Inset surfaces allow to generate internal structures starting at the boundary surface of a shape and our Analytic Skeleton helps to design them intuitively.

First, the skeleton's geometry and the surface-skeleton trajectories can be edited in a simple way: the user defines a smooth bijective function $I_b(\cdot)$ to change the definition of the u-coordinate onto the bone b. The new coordinate is defined as: $\tilde{u}(\mathbf{v}) = I_b(u(\mathbf{v}))$ for any vertex $\mathbf{v}$. This $\tilde{u}$-coordinate is not harmonic anymore, but gives the user the ability to edit the trajectories intuitively. For instance, in Fig. 13, the user "slides" the trajectories along the bones, to smooth them or to create singularities – in this particular example, trajectories from the head of the mother point to the same location after editing.

Second, the user can edit smooth inset surfaces with local control of its shape, while preserving the natural layout of the vertices on the bone, this layout being induced by the initial harmonic u-parameterization. The inset surface is then defined as a linear interpolation between the input surface and its skeleton, with $\alpha \in [0;1]$ an interpolation parameter that can vary spatially (i. e. for each vertex):

$$IS_\alpha(\mathbf{v}) = (1-\alpha)\cdot \mathbf{p}_v + \alpha \cdot b(\tilde{u}(\mathbf{v})) \qquad (13)$$

Examples of interactively edited inset surfaces are shown in red in Fig. 13 (top).

Due to the parametric nature of the Analytic skeleton, painting offers a simple means to edit the geometry of the skeleton as well as the shape of inset surfaces. The user draws using a brush equipped with smooth functions $W_i : M \Rightarrow \mathbb{R}^i$ that act as a displacement modulation from the surface to the skeleton or vice versa. This function is inserted in Eq. 12 to obtain $b(u) = \frac{\int_0^{2\pi} W_1(v(u,\theta))\cdot p(u,\theta)d\theta}{\int_0^{2\pi} W_1(v(u,\theta))d\theta}$, affecting smoothly the skeleton's geometry.

User-supervised weighting can also be integrated in Eq. 13 to obtain $IS_t(v) = (1 - Spl(\mathbf{v})(\alpha)) \cdot \mathbf{p}_v + Spl(v)(t) \cdot b(\tilde{u}(v))$ and locally deform the inset surface by locally editing its depth. The weighting function is then in the form $Spl : M \Rightarrow \{F_n : [0;1] \rightarrow [0;1]\}$, F_n being a bijective cubic spline of order n.

As an application, we propose an interactive Shell Mapping [PBFJ05] tool, with which the user can "peal" surfaces using one of its specific inset surface and then map arbitrary geometry in the so-defined in-between shell space (see Fig. 13 bottom).

8.3. Skeletal Mesh Filtering

Mesh smoothing is a fundamental tool in geometry processing. Beyond low-pass (Laplacian) filtering, the Bilateral Filter has recently emerged as an efficient feature-preserving operator. Originally designed for images [TM98], it has been adapted to meshes and higher dimensional data. Here, the basic idea is to replace the standard spatially weighted local combination of samples by bilateral weights accounting for both sample proximity in the domain and in the range of the sampled function. Although domain and range spaces are clearly defined for images (i. e. , pixel positions and pixel colors), this notion is ill-posed for geometry, as 3D vertex positions are usually not expressed relatively to a global parametric domain. Jones et al. [JDD03] overcome this problem by deducing a range value from the distance to

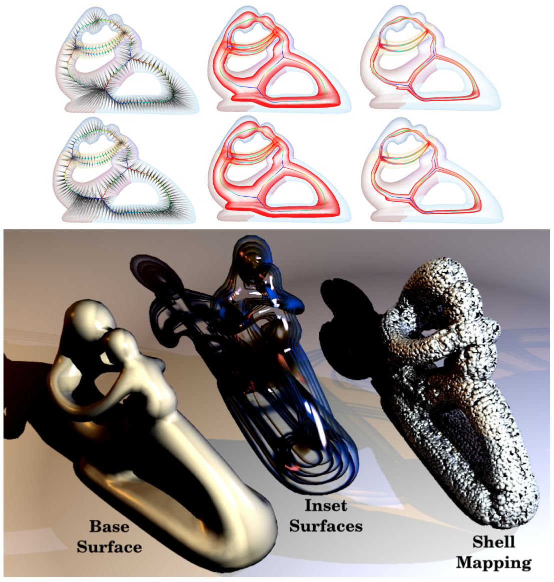

Figure 13: ***Top:*** *Smooth inset surfaces extraction and interactive editing.* ***Bottom:*** *Interactive design of in-between shell space for mapping of arbitrary, even procedural geometry (right) without inflating the global volume of the shape.*

neighbors tangent planes:

$$BL(\mathbf{p}_v) = \frac{\sum_{\mathbf{u}\in N(\mathbf{v})} \Pi_u(\mathbf{p}_v)\omega_s(||\mathbf{p}_u - \mathbf{p}_v||)\omega_r(||\Pi_u(\mathbf{p}_v) - \mathbf{p}_v||)a_u}{\sum_{\mathbf{u}\in N(\mathbf{v})} \omega_s(||\mathbf{p}_u - \mathbf{p}_v||)\omega_r(||\Pi_u(\mathbf{p}_v) - \mathbf{p}_v||)a_u}$$

with a_u an area weight, ω a Gaussian kernel or one of its polynomial approximations and the orthogonal projection $\Pi_u(\mathbf{p}_x) = \mathbf{p}_x - \langle \mathbf{p}_x - \mathbf{p}_u | \mathbf{n}_u \rangle \mathbf{n}_u$ modeling *coplanar similarity* as a range space.

Although this definition works perfectly well for small scale high frequency features, it quickly loses its detail preserving behavior for mid-size structures. Basically, the coplanar similarity assumption does not hold for larger structures and we propose to replace it by a local *thickness* similarity measure. To do so, we make use of our curve skeleton to define a new filtering operator, more suitable for larger structure preservation: *Skeletal* (Bilateral) Filtering.

Let $\Psi(\mathbf{v}) = b(u(\mathbf{v}))$ be the projection of $\mathbf{p}_v$ onto its related bone. Basically, we model the shape thickness similarity r as the norm of the surface-skeleton vector $\Psi(\mathbf{v}) - \mathbf{p}_v$, establishing the skeleton as the domain and Ψ^{-1} as the geometric signal (range). We can therefore express skeletal filtering as:

$$SBL(\mathbf{p}_v) = \frac{\sum_{\mathbf{u}\in N(\mathbf{v})} \Pi_u(\mathbf{p}_v)\omega_s(||\mathbf{p}_u - \mathbf{p}_v||)\omega_r(|r(\mathbf{u}) - r(\mathbf{v})|)a_u}{\sum_{\mathbf{u}\in N(\mathbf{v})} \omega_s(||\mathbf{p}_u - \mathbf{p}_v||)\omega_r(|r(\mathbf{u}) - r(\mathbf{v})|)a_u}$$

Fig. 14 illustrates the differences to standard bilateral filtering: medium size structures are preserved even if small structures are strongly smoothed out. This behavior is often desirable when processing shapes with a wide range of frequencies. The skeletal filter can be used in conjunction with laplacian and bilateral filters in a multi-lateral filter, where planarity and thickness act as complementary range spaces.

8.4. Discussion

By nature, curve-skeletons are not differentiable at the joint points where more than two bones connect (it is not even locally 1-manifold). This property has to be taken into account when designing skeleton-based applications. The skeletal mesh filtering application (see Sec. 8.3) is not affected by this, as demonstrated in the results where the filtering has been performed *globally* on the mesh, and not only onto a bone. The inset surface modelling application (see Sec. 8.2) is not affected either, as the result of this application is used to construct a shell-space embedding, which does not require necessarily smooth boundaries everywhere. On the contrary, the displacement application (see Sec. 8.1) requires to set a displacement that is *regular* in order to avoid strong normal discontinuities at the boundaries of the cylinders. This is achieved in our framework by adjusting the displacement function derivatives at the joints of the bones. These considerations are, of course, application-wise.

9. Conclusion

Curve skeletons offer an intuitive and efficient way to track global shape information such as topology, thickness and global alignment of structure on a large variety of shapes. Our Analytic Skeleton model is generated efficiently by means of a disk-cylinder surface decomposition followed by the geometrical integration of per-region harmonic maps. The related surface decomposition can then be improved at boundaries using a specific relaxation method. We illustrated the utility of such a parametric skeleton on geometry modeling and processing examples. We believe that a large variety of new applications can benefit from it. In particular, automatic processes such as shape analysis, compression and visualization can make use of its surface-inherited regularity; while animation, interactive shape modeling and processing can exploit its parametric definition and explicit relationship to the surface.

Acknowledgments
Data-sets are courtesy of AIM@SHAPE. This work has been partially funded by the REVERIE European project, the 3DLife European Network of Excellence and the ANR KidPocket & FETUS projects.

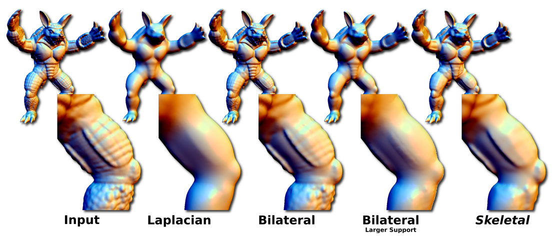

Figure 14: *Surface filtering comparison. All pictures but the fourth use identical support sizes.*

References

[ACSD*03] Alliez P., Cohen-Steiner D., Devillers O., Levy B., Desbrun M.: Anisotropic polygonal remeshing. *ACM Transactions on Graphics. Special issue for SIGGRAPH conference* (2003), 485–493. 2

[ATC*08] Au O. K.-C., Tai C.-L., Chu H.-K., Cohen-Or D., Lee T.-Y.: Skeleton extraction by mesh contraction. In *ACM SIGGRAPH 2008 papers* (New York, NY, USA, 2008), SIGGRAPH '08, ACM, pp. 44:1–44:10. 2, 7

[BF81] Bolles R. C., Fischler M. A.: A ransac-based approach to model fitting and its application to finding cylinders in range data. In *IJCAI'81: Proceedings of the 7th international joint conference on Artificial intelligence* (1981), pp. 637–643. 2

[BF06] Beder C., Förstner W.: Direct solutions for computing cylinders from minimal sets of 3d points. In *Proceedings of the 9th European conference on Computer Vision - Volume Part I* (Berlin, Heidelberg, 2006), ECCV'06, Springer-Verlag, pp. 135–146. 2

[CG01] Chaperon T., Goulette F.: Extracting cylinders in full 3d data using a random sampling method and the gaussian image. In *VMV '01: Proceedings of the Vision Modeling and Visualization Conference 2001* (2001), pp. 35–42. 2

[CSAD04] Cohen-Steiner D., Alliez P., Desbrun M.: Variational shape approximation. In *ACM SIGGRAPH* (2004), pp. 905–914. 3

[CSM07] Cornea N. D., Silver D., Min P.: Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics 13*, 3 (2007), 530–548. 2, 6, 7

[DZM08] Dyer R., Zhang H., Möller T.: Surface sampling and the intrinsic Voronoi diagram. In *ACM Symposium on Geometry Processing* (2008), pp. 1393–1402. 2, 3

[ES94] Edelsbrunner H., Shah N. R.: Triangulating topological spaces. In *Proc. of the 10th Symposium on Computational Geometry* (1994), pp. 285–292. 2, 3

[FK97] Fomenko A., Kunii T.: Topological modeling for visualization, 1997. 2

[Hop96] Hoppe H.: Progressive meshes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1996), SIGGRAPH '96, ACM, pp. 99–108. 3

[JDD03] Jones T. R., Durand F., Desbrun M.: Non-iterative, feature-preserving mesh smoothing. In *ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), SIGGRAPH '03, ACM, pp. 943–949. 1, 8

[KLT05] Katz S., Leifman G., Tal A.: Mesh segmentation using feature point and core extraction. *The Visual Computer 21*, 8-10 (2005), 649–658. 2

[KS06] Krayevoy V., Sheffer A.: Variational, meaningful shape decomposition. In *ACM SIGGRAPH 2006 Sketches* (New York, NY, USA, 2006), SIGGRAPH '06, ACM. 4

[KT03] Katz S., Tal A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers* (2003), pp. 954–961. 2

[LCT11] Liu Y.-J., Chen Z., Tang K.: Construction of iso-contours, bisectors, and voronoi diagrams on triangulated surfaces. *IEEE Trans. Pattern Anal. Mach. Intell. 33*, 8 (Aug. 2011), 1502–1517. 2

[LPRM02] Lévy B., Petitjean S., Ray N., Maillot J.: Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph. 21*, 3 (2002), 362–371. 2

[PBFJ05] Porumbescu S. D., Budge B. C., Feng Z., Joy K.: Shell maps. *ACM SIGGRAPH 2005, ACM Transactions on Graphics 24*, 3 (2005), 626–633. 1, 8

[PJP93] Pinkall U., Juni S. D., Polthier K.: Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics 2* (1993), 15–36. 5

[PSBM07] Pascucci V., Scorzelli G., Bremer P.-T., Mascarenhas A.: Robust on-line computation of reeb graphs: simplicity and speed. In *ACM SIGGRAPH 2007 papers* (New York, NY, USA, 2007), SIGGRAPH '07, ACM. 2, 7

[RBG*09] Reuter M., Biasotti S., Giorgi D., Patanè G., Spagnuolo M.: Discrete laplace-beltrami operators for shape analysis and segmentation. *Computers & Graphics 33* (2009), 381–390. 2

[Ree46] Reeb G.: Sur les points singuliers d'une forme de pfaff completement intergrable ou d'une fonction numerique on the singular points of a complete integral pfaff form or of a numerical function. *Comptes Rendus Acad. Science Paris. v222.* (1946), 847–849. 2, 7

[RvdH] Rabbani T., van den Heuvel F.: Efficient hough transform for automatic detection of cylinders in point clouds. 2

[SCOGL02] Sorkine O., Cohen-Or D., Goldenthal R., Lischinski D.: Bounded-distortion piecewise mesh parameterization. In *VIS '02: Proceedings of the conference on Visualization '02* (2002), pp. 355–362. 2

[Sha08] Shamir A.: A survey on mesh segmentation techniques. *Computer Graphics Forum 27*, 6 (2008), 1539–1556. 3

[SLSK07] Sharf A., Lewiner T., Shamir A., Kobbelt L.: On-the-fly curve-skeleton computation for 3d shapes. *Computer Graphics Forum 26*, 3 (Sept. 2007), 323–328. 2

[SSGH01] Sander P. V., Snyder J., Gortler S. J., Hoppe H.: Texture mapping progressive meshes. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (2001), pp. 409–416. 2

[SWG*03] Sander P. V., Wood Z. J., Gortler S. J., Snyder J., Hoppe H.: Multi-chart geometry images. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (2003), pp. 146–155. 2

[TM98] Tomasi C., Manduchi R.: Bilateral filtering for gray and color images. In *ICCV* (1998), pp. 836–846. 8

[TS04] Tung T., Schmitt F.: Augmented reeb graphs for content-based retrieval of 3d mesh models. In *SMI '04: Proceedings of the Shape Modeling International 2004* (2004), pp. 157–166. 1

[TVD08] Tierny J., Vandeborre J.-P., Daoudi M.: Enhancing 3d mesh topological skeletons with discrete contour constrictions. *The Visual Computer 24* (2008), 155–172. 2

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

[WK05] Wu J., Kobbelt L.: Structure recovery via hybrid variational surface approximation. *Computer Graphics Forum 24*, 3 (2005), 277–284. 2, 3

[YLW06] Yan D.-M., Liu Y., Wang W.: Quadric surface extraction by variational shape approximation. In *Proceedings of the 4th international conference on Geometric Modeling and Processing* (Berlin, Heidelberg, 2006), GMP'06, Springer-Verlag, pp. 73–86. 3

[YSZ06] Yang X., Somasekharan A., Zhang J. J.: Curve skeleton skinning for human and creature characters: Research articles. *Comput. Animat. Virtual Worlds 17*, 3-4 (2006), 281–292. 1