Analysis, Reconstruction and Manipulation using Arterial Snakes

Guo Li[†] Ligang Liu[†*] Hanlin Zheng[†] Niloy J. Mitra[‡*]

[†]Zhejiang University [‡]KAUST / IIT Delhi

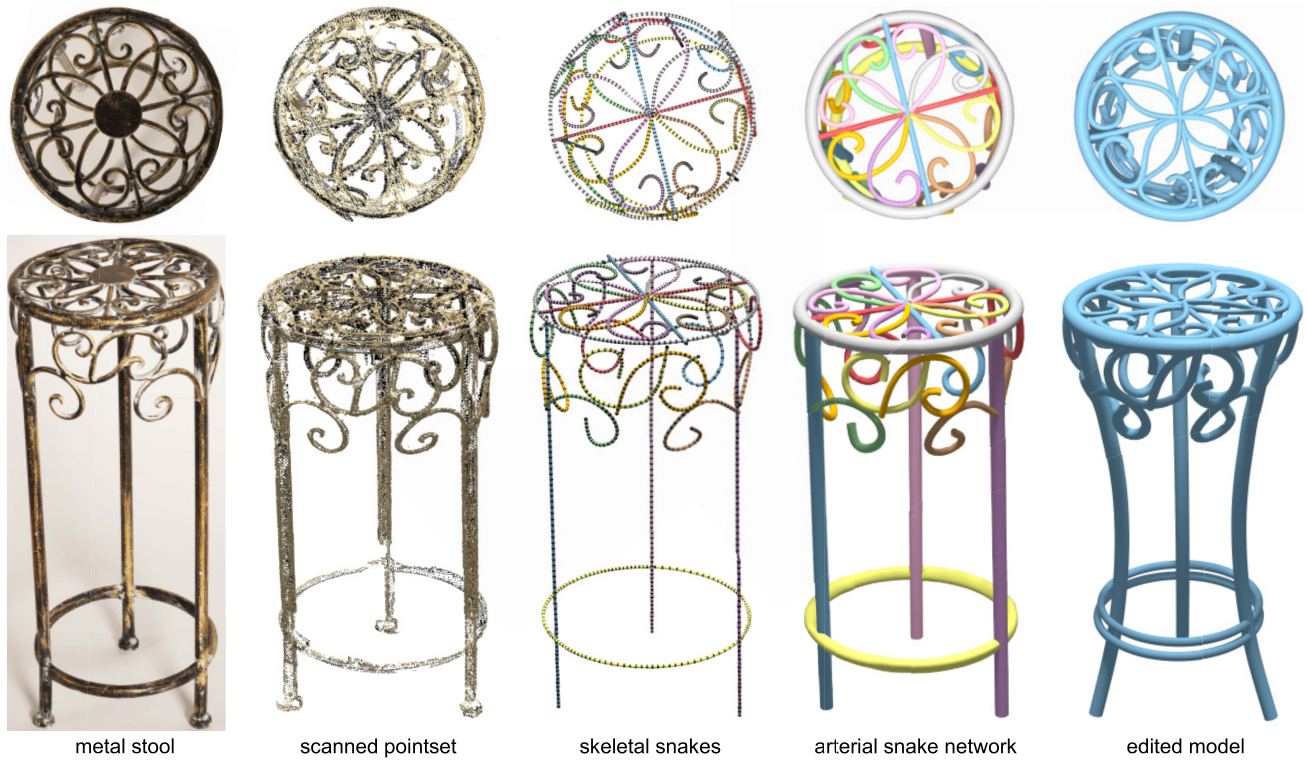

Figure 1: *Starting from a noisy raw scan with large parts missing our algorithm analyzes and extracts a curve network with associated cross-sectional profiles providing a reconstructed model. The extracted high-level shape representation enables easy, intuitive, yet powerful geometry editing. Note that our algorithm is targeted towards delicate 1D features and fails to detect the small disc at the top of the stool.*

Abstract

Man-made objects often consist of detailed and interleaving structures, which are created using cane, coils, metal wires, rods, etc. The delicate structures, although manufactured using simple procedures, are challenging to scan and reconstruct. We observe that such structures are inherently 1D, and hence are naturally represented using an arrangement of generating curves. We refer to the resultant surfaces as *arterial surfaces*. In this paper we approach for analyzing, reconstructing, and manipulating such arterial surfaces. The core of the algorithm is a novel deformable model, called *arterial snake*, that simultaneously captures the topology and geometry of the arterial objects. The recovered snakes produce a natural decomposition of the raw scans, with the decomposed parts often capturing meaningful object sections. We demonstrate the robustness of our algorithm on a variety of arterial objects corrupted with noise, outliers, and with large parts missing. We present a range of applications including reconstruction, topology repairing, and manipulation of arterial surfaces by directly controlling the underlying curve network and the associated sectional profiles, which are otherwise challenging to perform.

Keywords: Shape analysis, arterial snake, point cloud, surface reconstruction, relief analysis

1 Introduction

Man-made objects made out of tubular components, like metal rods, wood, cane, coils, are quite common in furniture design, metal sculpture, and wire art (see Figure 2). Despite recent advances in acquisition techniques, capturing such delicate structures remains challenging. The resulting scans typically suffer from significant self-occlusion resulting in imperfect point cloud data with missing

[*]Corresponding authors: ligangliu@zju.edu.cn, niloy@cse.iitd.ernet.in

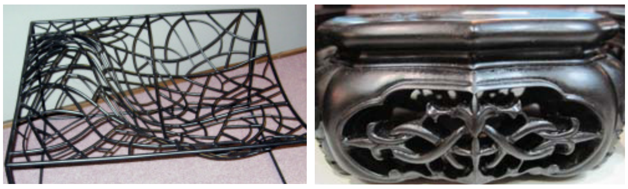

Figure 2: *Two artworks that can be well represented as an interwoven network of* arterial snakes*. Reliable reconstruction of such delicate structures from scanned data is challenging.*

parts. Moreover, the raw scans are low-level representations based on points and (unreliable) local connectivity derived based on spatial proximity or scanning order. As a result recovering a meaningful global topology from the partial scanned data is a challenging problem.

We observe that such surfaces can be explained using 1D generating curves, which can be used to generate sweeping surfaces called *arterial surfaces* (see Figure 1). More precisely, we define a *layered arterial surface* as interleaving tubular shapes, represented as a collection of curves that gives rise to a series of sweeping surfaces. This characterization suggests a natural representation that specifically encodes the underlying global topology, allows easy decomposition into meaningful layers, and subsequently enables easy and intuitive manipulations. Additionally, understanding the structure of the data allows us to improve the acquisition process and to produce a globally consistent plausible surface.

Most skeleton extraction procedures, simple or state-of-the-art, robustly recover underlying curve segments that roughly explain the geometric details around the non-intersection regions. However, the results are unreliable around the overlap regions. Also the methods are not targeted to capture the global continuity and regularity present in the data (see Figure 11).

We propose a novel algorithm that alternate between learning the topology of the curve family and the geometric attributes of the generating curve profiles to better explain the scanned data. The algorithm is based on a globally-coupled optimization, where starting from the confident small arterial segments, we grow the curves to form long smooth coherent ones. The core of the optimization is based on a novel surface representation, called *arterial snake*, which is a deformable arterial surface automatically evolved by growing reliable curve segments to fit the scanned data. The extracted high level shape representation subsequently allows easy and intuitive manipulation of the geometry layers. By separating the curves into multiple layers and learning their *local-layering* structure, we facilitate intuitive geometric edits and manipulations (see Figure 1). We have tested our algorithm on a variety of inputs, both synthetic and acquired, under varying sampling density and noise conditions with large chunks of missing data.

To the best of our knowledge, we are the first to propose an automatic approach to separate interleaving arterial patterns into multiple geometric layers. One of the key insights is that instead of extracting the curve skeleton directly, the topology of the arterial surface can be traced by growing short snake segments starting from the reliable tubular regions.

Contributions. Our key contributions are:

- *arterial snake*, a deformable primitive, to decompose a detail surface into a multi-layer collection of 1D generative curves,
- a purely geometric approach to extract the arterial patterns and define their layering relationship leading to an enhanced reconstruction of the (incomplete) input scans, and
- a simple, intuitive, yet powerful shape manipulation framework that allows easy and natural editing and creation of plausible and realistic variants of the original geometry.

2 Related Work

Skeleton extraction. Curve-skeletons are 1D structures that are often used to abstract geometry and topology of 3D shapes [Cornea et al. 2007] or medical imaging [Kirbas and Quek 2004]. A classical approach for extraction of characteristic features is *medial axis*, which consists of points that are equidistant from two or more points of the underlying surface. Variants have been proposed towards robustly defining medial axis for generating curve skeleton of a shape [Siddiqi and Pizer 2008].

Topology-driven methods track the critical points of functions defined on manifolds, and then capture their topological structure using Reeb graphs [Hilaga et al. 2001; Patané et al. 2008]. Dey and Sun [2006] adopt a medial geodesic function over the medial axis and sheet to convert them into 1D structures. Au et al. [2008] extract the skeleton by carefully shrinking a shape using constrained Laplacian smoothing. Recently, Miklos et al. [2009; 2010] introduce a family of skeletons that captures the important features of a shape in a scale-adaptive way. This yields a hierarchy of successively simplified skeletons leading to an alternate curve representation, called *scale axis* (see Figure 11).

Shape analysis. Extraction and analysis of curve skeletons from point clouds have been explored in a few recent attempts. Tagliasacchi et al. [2009] build upon the notion of generalized rotational symmetry axis for robust curve skeleton extraction. Cao et al. [2010] extend the Laplacian-based contraction approach, originally introduced by Au et al. [2008], to extract curve skeleton from point cloud data. Recently, Livny et al. [2010] use a series of global optimizations to fit skeletal structures to scans of trees. Unlike these approaches, our method attempts to recover a *functional* set of curves to capture the structure of the shape.

Mesh segmentation [Attene et al. 2006; Shamir 2008; Chen et al. 2009] decomposes a surface into a collection of non-overlapping parts, based on some notion of equivalence. In certain cases, specially for articulated bodies, segmented models are effectively used to create a model skeleton for manipulating the underlying shape [Simari et al. 2006]. These approaches, however, are ill-suited for peeling the geometric details into individual layers. For the family of shapes under investigation, we propose an arterial network representation that attempts to capture the (procedural) manufacturing ideology of the underlying objects.

Detail extraction and representation. Extracting geometric features and details from a surface is essential for various detail-preserving mesh manipulation operations [Kobbelt et al. 1998; Botsch and Kobbelt 2003]. Alternatively, meshes can be treated as signals with multiple frequency-bands, differential surface representations [Sorkine et al. 2004; Lipman et al. 2005] for efficient encoding of high-frequency content and for detail-preserving shape manipulation. On the synthesis side, researchers have focused on adding 3D geometric textures on surfaces, and creating interwoven structures [Zhou et al. 2006].

Relief structures are popular sculptured artworks typically containing surface details that can expressed as height-fields over smooth base surfaces [Weyrich et al. 2007]. Liu et al. [2006; 2007b] propose algorithms to extract isolated or periodic reliefs on a smooth background or on a textured background. Recently Zatzarinni et al. [2009] propose a novel formulation for segmenting multiple

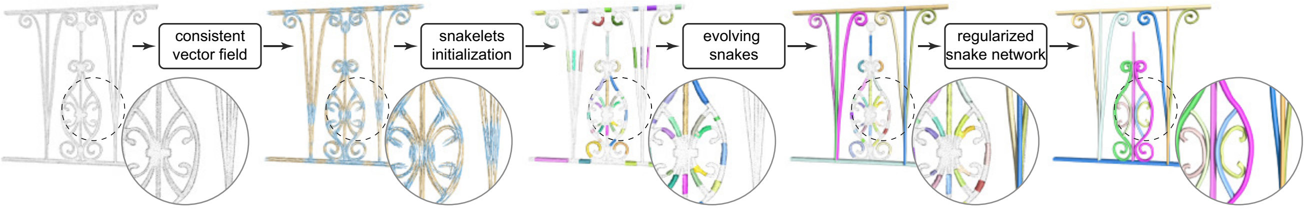

Figure 3: *System overview. (Left to right) Starting from an input point cloud $\mathcal{R}$, we estimate local directional vectors (in yellow) at each point and use the reliable ones to generate a globally consistent vector field (in blue) in the remaining points. We then initialize a set of arterial elements called* snakelets*, which then compete, evolve, and grow guided by the point samples and the computed vector field. The snakes obtained from the iterative topology recovery and geometry fitting phases are finally regularized to yield an arterial snake network.*

hight field reliefs on general surfaces. For arterial-like reliefs extracted using such methods, we use our algorithm to understand the interwoven structures of the reliefs, which can then be used to improve and manipulate the extracted structures (see Figures 1 and 14).

Deformable model. In a highly influential paper, Kass et al. [1988] introduce *snake* as a model based technique for robust segmentation of images. The definition has been subsequently extended to manifold surfaces [Lee and Lee 2002; Lee et al. 2005] and to implicit functions [Esteban and Schmitt 2003; Esteve et al. 2005]. Duan and Qin [2004] propose *intelligent balloon*, a parameterized subdivision surface governed by a data driven energy minimization, to model complex shape geometry of arbitrary topology. Sharf et al. [2007] grow a smooth deformable surface from inside of a point cloud and traces the fronts using an explicit mesh representation. In the context of man-made objects that are characterized by sharp features, 1D curves have been shown to be powerful priors for abstracting the shapes [Mehra et al. 2009] and also for intuitive manipulations [Gal et al. 2009]. In this work, we employ arterial surfaces as a shape prior for improved surface reconstructions. Further, the analyzed structure of the curve network can be directly used for high level shape manipulations as recently proposed by Igarashi et al. [2010].

3 Overview

Given an unorganized set of points $\mathcal{R}$, scanned or sampled from a surface, our goal is to recover the underlying geometry. The input point sets are noisy, contain outlier, and are incomplete. Our algorithm is designed for objects that primarily consist of arterial surfaces, i.e., can be represented using a collection of (smooth) curves and an associated set of additional attributes. The recovered arterial surfaces not only help produce a better reconstruction, but also help capture high level characteristics of the shapes to facilitate easy, intuitive, and powerful edits.

We represent each tubular section as a central curve skeleton and an associated set of cross sections. We call this deformable model an *arterial snake*, and parameterize it as a mesh $\mathcal{S}$, consisting of a curve and a set of cross section curves (see Figure 4). Automatically recovering a network of interwoven arterial snakes is challenging, even ill-posed, since we aim to recover the underlying topology from scan-quality, poorly sampled, and incomplete inputs. We observe that reliable extraction of underlying curves is possible in parts away from intersecting and overlapping regions. Local curve topology can then be systematically propagated, merged, and consolidated to yield a plausible curve network representation of the underlying geometry. We alternate between topology recovery and geometry reconstruction as we progressively fit to the input data (see Figure 3).

Starting from the set of points $\mathcal{R}$, we first estimate a local tangential direction vector at each point and identify regions with consistent vector directions. We fit small *arterial snakelets* $\mathcal{S}^j$ to the points and their directional estimates from the marked regions using an energy optimization. Next, we grow these arterial snakelets based on the input data. For this purpose, we use a diffusion operator based on a tensor field formulation to globally and smoothly propagate reliable vector directions to the unreliable regions. Guided by the point samples and the globally consistent tensor fields, we grow the snakelets $\mathcal{S}^j$ in an anisotropic fashion, favoring growth direction along the curve skeleton directions. Simultaneously, in the topology correction step, we merge snake pairs whose end points are spatially close and whose direction vectors agree. We perform this merging by mapping $\mathcal{S}^j$ to a higher dimensional similarity space (positional and tangential information), and identifying proximal pairs. Note that an analogous growth guided solely by the input data points is unreliable and leads to poor topology reconstruction. We use the generated vector field, which integrates *global* information, as a better scaffold to recover consistent topological information, while we use the sample points to help estimate geometric attributes. We alternate between the topological and geometric phases to extract a collection of arterial snakes representing the input samples.

The curve network, recovered from the noisy incomplete data, often contains artifacts, noise, and irregularities. We correct these noise irregularities learned from the corrupted data in a *regularization phase*. In this step, we project curves to smooth polynomial approximations, enforce planarity for near-planar curves, snap to tangent continuity, and enhance symmetries between curve elements. The resultant network of arterial snakes not only explains the input data, but also encodes the characteristics of the shape learned during the analysis phase.

4 Algorithm

We first describe how we explicitly represent the arterial snakes and evolve them to recover the curve network topology of the input point cloud data $\mathcal{R}$. We then describe the objective function we designed to grow and fit the snakes guided by $\mathcal{R}$. Note that the algorithm extracts topological connectivity and estimates geometric attributes for the snakes, while also estimating the quality of the underlying data.

Arterial snake representation. Similar to traditional swept surfaces we represent an arterial snake $\mathcal{S}$ by a generating cross sectional curve $\mathcal{G}$ along a skeletal curve $\mathcal{C}$. We discretize skeletal curve $\mathcal{C}$ as a polyline of nodes $\{C_1, C_2, \cdots, C_n\}$, with $C_1 = C_n$ for closed curves. The cross section G_i at each key node C_i is represented as a planar (closed) polygon with m points $G_i = \{G_i^1, G_i^2, \cdots, G_i^m\}$. The corresponding arterial snake surface $\mathcal{S}$ is represented as a mesh, called *skin mesh*, with vertices $\{G_i^j\}$. The

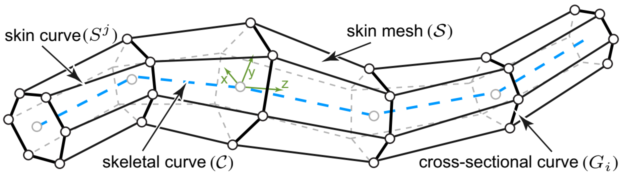
skin curve (S^j)
skin mesh ($\mathcal{S}$)
x
y
z
skeletal curve ($\mathcal{C}$)
cross-sectional curve (G_i)

Figure 4: *We represent an arterial snake surface to be generated by sweeping cross sectional curves G_i along a skeletal curve $\mathcal{C}$. Both $\mathcal{C}$ and G_i are represented as polylines. The mesh surface $\mathcal{S}$ is called skin mesh and the surface polylines S^j are called skin curves. A rotation minimizing frame is maintained along the skeletal curve.*

surface polylines $S^j = \{G_i^j\}$ are called *skin curves* (see Figure 4). To position the cross sectional curves for a snake, we use the tangents to its skeleton curve $\mathcal{C}$ as the local z-axis. We arbitrarily assign a vector in the plane containing G_1 as the local x-axis at C_1, and then propagate the local coordinate frame to other cross sectional curve G_i along the skeletal curve $\mathcal{C}$ using a rotation minimizing formulation [Wang et al. 2008].

4.1 Initialization

Let the given scanning data be $\mathcal{R} = \{\mathbf{p}_i, \mathbf{n}_i\}$. The normal vectors $\mathbf{n}_i$ are either obtained during the acquisition process, or locally estimated using a least squares local plane fitting [Mitra et al. 2004].

Cross section planes. We first detect sections of the input that can be locally approximated by tubular sections. Motivated by the work of Tagliasacchi et al. [2009], we characterize such parts as a subset of points $\mathcal{A}$ lying close to a plane with unit normal direction $\mathbf{v}$ such that the normals associated with the data points are orthogonal to $\mathbf{v}$ in a least squares sense. Thus we solve the optimization $\min_{\mathbf{v}} \sum_{\mathbf{p}_i \in \mathcal{A}} (\mathbf{v}^T \mathbf{n}_i)^2$ subject to $\|\mathbf{v}\| = 1$, which reduces to finding the smallest eigenvector of the covariance matrix $\sum_{\mathbf{p}_i \in \mathcal{A}} \mathbf{n}_i \mathbf{n}_i^T$. Note the sign of $\mathbf{v}$ is not important. For any point $\mathbf{p}_i \in \mathcal{R}$, we initialize its cutting plane π_i as a plane passing through $\mathbf{p}_i$ and containing the normal $\mathbf{n}_i$. Then we identify the subset $\mathcal{A}$ of point set $\mathcal{R}$ that is near to the plane π_i and belongs to the same connected component as $\mathbf{p}_i$ (point pairs are considered connected if their Euclidean distance is less than 0.1% of the bounding box diagonal length of the object, e.g., green points in Figure 5-left). Using the subset $\mathcal{A}$, we re-estimate the cutting plane normal $\mathbf{v}_i$, and then re-estimate points close to the current cutting plane, and alternate between the two steps. In our experiments, for each point we run the process for a maximum of four iterations. In the end, we obtain a local directional estimate $\mathbf{v}_i$ for each point $\mathbf{p}_i$ as described above (see Figure 5-right). In spirit, this is similar to the first step of the method proposed by Tagliasacchi et al. [2009] to compute the cutting plane for each data point and assign the normals of the cutting planes as the vector field over the data points. We, however, do not compute the skeleton. Instead we generate a vector field over the data points that is a key component to guide the evolution of the snakes.

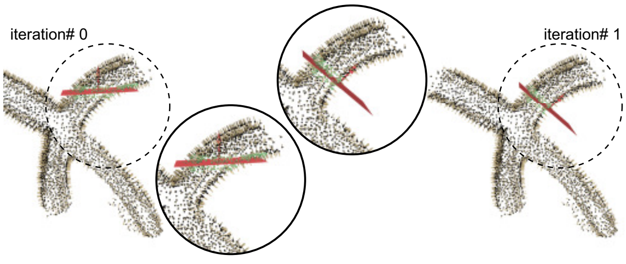

Figure 5: *Computation of cross-section plane for each sample point. Starting from an initial cutting plane (left), we iteratively re-estimate the plane (right) using a local optimization.*

Filtered vector field. The orientation vectors in tubular sections are consistent and well-behaved, while they are unstable and noisy near joints and crossings of multiple snakes. For each point $\mathbf{p}_i$, we compute the average vector $\bar{\mathbf{n}}_i$ of the point vectors associated with the cross section plane π_i. If the angle between $\bar{\mathbf{n}}_i$ and the normal $\mathbf{v}_i$ of π_i is larger than a threshold δ_a (10 degrees), we filter out such directional estimates. We also remove directional estimates from points with high variance δ_v (> 5) among the directional vectors of the points associated with the cross-sectional plane π_i. After filtering out inconsistent directional estimates, we are left with regions of the input data that can be well approximated by small arterial segments, or snakelets (see Figure 6-left). In contrast, in the unreliable parts, typically around junctions, we have only points $\mathbf{p}_i$ without any associated direction estimate $\mathbf{v}_i$. As the directions of the vectors affect subsequent interpolation, we first convert the vector field into a tensor field [Zhang et al. 2007]. At each reliable point $\mathbf{p}_i$ we create a tensor $M_{\mathbf{p}_i} = \mathbf{v}_{\mathbf{p}_i} \mathbf{v}_{\mathbf{p}_i}^T$, and then smooth and spread the tensor field to the other points without associated directional estimates by solving a global system. Specifically, we define a local Laplacian operator at each point and its neighbors, and thus obtain a system of global Laplacian equations [Liu et al. 2007a] over the points. We use an orientation-aware distance to build the graph-relationship on the input point set as presented by Tagliasacchi et al. [2009]. We set high weights on constraints derived using points with reliable directional vectors. By solving the resultant sparse linear system we obtain tensors at all the unassigned points, and thus their associated vectors. This information is later used for evolving snakes over curved parts.

Snakelets. Points from $\mathcal{R}$ with reliable directional estimates are first grouped into clusters based on how well their directionals agree and based on their spatial proximity (see Figure 6-left). We initialize each such group $\mathcal{S}$ with an initial snakelet. In order to robustly handle noise and missing data, we first restrict the snakelets to have circular cross-sectional profiles, similar to traditional canal surfaces. For each point $\mathbf{p}_i \in \mathcal{S}$, we perform a circle fit through $\mathbf{p}_i$ in its cross-sectional plane π_i, to get the fitted circle center as $\mathbf{c}_i$. The point set $\{\mathbf{c}_i\}$ gives a dense sampling of the skeletal curve of $\mathcal{S}$. We resample the initial skeletal curve using a distance spacing of σ, 0.2% of the bounding box diagonal length of the object, to get a polyline skeletal curve $\mathcal{C}$. The corresponding cross-sectional curves $\mathcal{G}_i$ are initialized using circular profiles computed in the circle fitting stage. Thus in the unambiguous regions of the dataset $\mathcal{R}$ we have initializations of snakelets (see Figure 7-leftmost), which are subsequently evolved, grown, and merged to form the snake network.

4.2 Topology Recovery

In this section, we use the initial snakelets to extract layered arterial snakes using the incomplete point cloud data $\mathcal{R}$ for guidance. In absence of any segmentation of the input point cloud, direct application of state-of-the-art skeleton extraction methods [Au et al. 2008; Tagliasacchi et al. 2009; Cao et al. 2010] produce unsatisfactory results (see Figure 11-bottom). Typically resultant curve networks are patchy, only explain the structure of the arterial objects at the non-intersection regions, and fail to capture the continuity and characteristics regularity of the data. Post-processing such curves to fix their deficiencies is likely to involve a variety of specialized

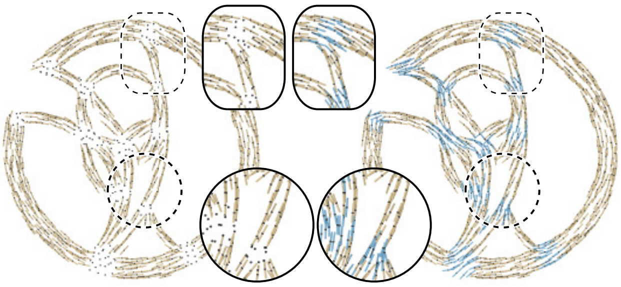

Figure 6: *Reliable directional vectors (in yellow) are propagated by solving a global Laplacian equation as directional estimates (in blue) for other points.*

rules which seem to be unreliable and infeasible. Instead we take a systematic approach of starting from a valid collection of snakelets and evolving them while maintaining their validity.

For each snake, we call the sample points that drive its evolution its *matching points*. In our setting, a key challenge is to simultaneously partition the data into different matching point sets, while fitting suitable snakes to the partitions. While it is relatively easy to decompose two arterial objects if they intersect transversely, it is harder to reliably detect tangential intersections (see Figures 6 and 7). Our snakelet growing scheme is designed to robustly handle such situations.

Our snake evolution algorithm has the following properties: (i) unlike classical deformable models [Duan and Qin 2004; Sharf et al. 2006], our deformable arterial snakes preserve the arterial shapes during their evolution; (ii) each arterial snake decides whether to grow along the center curve or stops to handle various kinds of ambiguity; (iii) the method robustly handles input data containing noise, outliers, cracks, and missing pieces.

Matching points. We iteratively grow each snake continuing along its skeletal curve. Each vertex of a snake is assigned to a matching point in the input data set. For each skin vertex, we shoot a line along its normal and find the nearest data point within a threshold (0.5%). Some vertices may not have any matching point. If a matching point has a normal that is inconsistent with the skin vertex normal (within a threshold 10 degrees), we discard the match. At the beginning, we find the matching points for the vertices of snakelets. Say a cross-sectional ring of points G_n has matching points M_n. When we grow the end ring of the snake, we use M_n and the underlying directional field to seed a set $\mathcal{M}$ of reliable matching points for newly growing end of the snake.

Snake growth. Suppose the current snake is given by a skeletal curve $\{C_i\}$ with corresponding cross-sectional profiles $\{G_i\}$, and we use rotation minimizing frames to propagate the local coordinate system. At one end we introduce a new skeletal node $C_{n+1} := C_n + \alpha T_n$ with the corresponding cross-sectional curve $G_{n+1} := G_n$, where T_n is the growth unit vector at C_n and α is the growth step. The growth step varies inversely with curvature radius κ of its skeletal curve near its evolving fronts. More specifically, we use $\alpha := \gamma \max(1, \sqrt{l/\kappa})$ where γ is the growth rate and l the current length of $\mathcal{C}$.

Intuitively, the growth direction T_n can be estimated by the tangent vector T_n^1 of a cubic Bézier curve that is obtained by fitting the four successive nodes at C_n. If a snake has small curvature at its end, this estimation process is reliable. But for a high curvature end node, we use the underlying directional vectors to adjust the growth direction, as follows: We detect the directional estimates of the matching points of nodes of G_n. If they are consistent, we compute an average directional vector T_n^2. If they are inconsistent, then the data points are unreliable, and we choose to ignore their effects, i.e., we set $T_n^2 = \mathbf{0}$. We set $T_n \leftarrow (T_n^1 + T_n^2)/2$. We then find the matching points for the vertices on G_{n+1}. If the matching points belongs to $\mathcal{M}$, we give them higher weights. Otherwise, we set smaller weights. Note that both the matching points and the vector fields are used to drive the snake evolve during the geometric optimization. The vector field is a key component to guide the snake to evolve, especially at the joint regions and incomplete regions.

Similarly, we extend the snake at its other end. We fit the extended snake to the input data using a geometric optimization (see Section 4.3). We continue extending each snake and performing geometric optimization, as long as the data fitting validates the extension. For validation we give higher weights to the point positions from $\mathcal{R}$, and rely on the directional estimates to a lesser extent. While optimizing for vertex positions of the evolving snakes, we only locally refine the positions of the mesh vertices at the evolving fonts, and freeze the inner vertex locations of the snakes that have been previously optimized.

Competing snakes. The success of our algorithm largely depends on critical topological events related to how the snakes evolve, grow, and merge to reliably extract an inter-woven snake network. We allow multiple snakes to grow simultaneously and resolve their interactions only when a collection of snakes come close (see Figure 7). Note that the input data is typically worse near intersections and overlap regions, making any local scheme unreliable.

Initially we start growing all the snakelets simultaneously by activating them. We consider both the curvature and the length of the snake to determine its growing speed. Generally, the longer and the flatter the snake, the faster it grows. In each iteration, snakes decide to grow, freeze or even shrink to handle various kinds of ambiguity based on the following strategies: (i) When a snake has little supporting data to justify growing, we mark it as inactive and freeze its growth. Thus, when a snake arrives at the data boundary or an incomplete region and cannot grow any further, we mark it as inactive. (ii) When two snakes meet, we mark both of them to be inactive, and freeze their growth. (iii) When two snakes S_1 and S_2 meet with the same snake S^*, then we reactivate S^* and grow it *backward*, i.e., shrink S^*. Intuitively, this makes way for S_1 and S_2 allowing them to continue to grow. Note that a snakelet affects a neighboring one only if both of them have open ends that are close by. Thus by shrinking S^* we allow S_1 and S_2 to grow, and subsequently S^* can grow back without being affected by S_1 and S_2

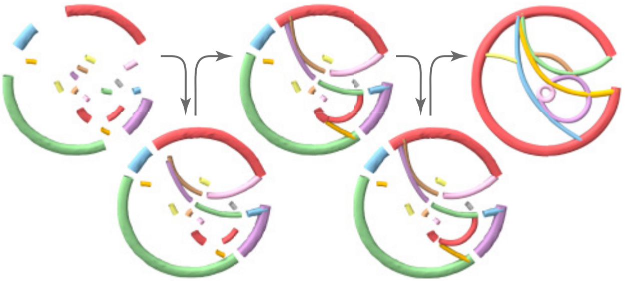

Figure 7: *Topology recovering of curve network. Starting from reliable snakelets, the arterial snakes grow simultaneously while competing with each other. Longer and flatter curves grow faster. Snakes may retract to make way for others to grow first, and adjacent snakes with suitable tangent conditions merge together (see Section 4.2).*

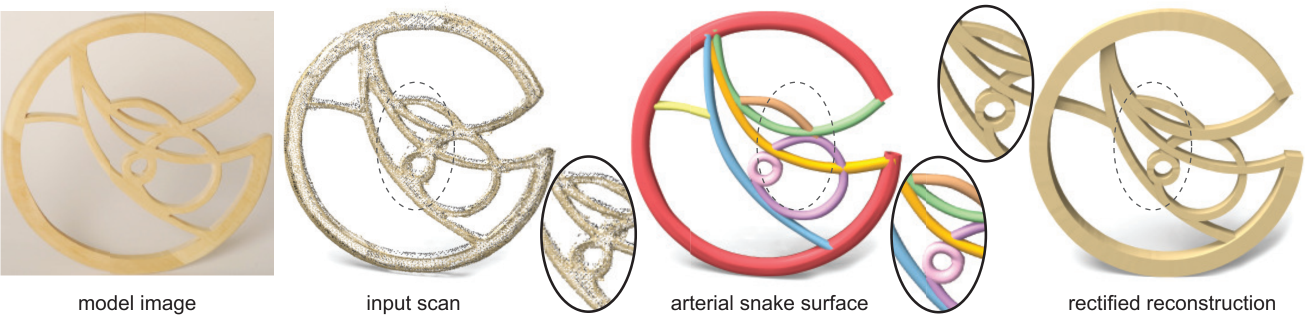

Figure 8: *Our system can analyze and extract the various arterial structures from a scan input with noise, outliers, and missing parts. The topology of the curves is recovered by our algorithm and the the final reconstructed model with the set of arterial surfaces is obtained by geometry regularization.*

that no longer have neighboring open ends (see Figure 7 and the accompanying video).

Snake merging. When all snakes are marked inactive, we look for potential snake pairs to merge. We connect two snakes if their end points are close (within 0.1 of the bounding box diagonal length of the data set) and the respective tangents to the skeletal curves near their end points agree (within 15 degrees in our implementation). This allows the snakes to bridge small gaps without sufficient data points to enable growing. Note that if two snakes overlap at a sharp angle, e.g., like in a cross, at this stage we may get two snakes that interpenetrate and explain the underlying data. We do not explicitly resolve such penetrations since in certain cases, e.g., welded metal (see Figure 1) such behavior is expected, while for others like cane penetrations are not realistic (see Figure 9-top).

4.3 Geometry Fitting

In our iterative scheme, during the evolution and growth of a snake $\mathcal{C}$, we employ geometric optimization to refine the snake parameters to better explain the input $\mathcal{R}$. While the connectivity remains fixed, we fine tune the vertices G_i^j of the skin mesh $\mathcal{S}$ by associating to each vertex an affine matrix $\mathbf{A}_i^j$ (with 12 degree of freedom per vertex). We introduce three terms to capture the conflicting requirements of the problem, and solve the resultant optimization.

Data error. We approximate the distance of the snake $\mathcal{S}$ from the pointset $\mathcal{R}$ as:

$$E_d := \sum_{i,j}^{n,m} w_i^j \left\| \mathbf{A}_i^j G_i^j - R_i^j \right\|^2, \qquad (1)$$

where R_i^j is the matching point in $\mathcal{R}$ of G_i^j. The weights w_i^j indicate confidence in data, and is proportional to the confidence in local directional estimates. In the tubular region, it is easier for skin vertices to get reliable corresponding matching points. In contrast, around a joint region, only a few skin vertices get assigned to reliable matching points, while others being in the interior of the joint region, do not get any matching points, and thus do not affect the data term (see Section 4.2).

Smoothness error. We expect coherence during evolution of the arterial snakes, and enforce spatial coherence between transformations at neighboring snake vertices using a smoothness term:

$$E_s := \sum_{(i,j)} \|\mathbf{A}_i - \mathbf{A}_j\|_F^2, \qquad (2)$$

where $\|\cdot\|_F$ is the Frobenius norm (see also [Allen et al. 2003]), and adjacency is based on connectivity of the underlying snake mesh.

Fairness error. Our arterial surfaces robustly handle thin features in the input data. This is made possible by our prior that the input model is dominated by essentially 1D smooth curves, thus allowing us to automatically reason about intersections, tangential meetings, proximal curves, etc. To prevent snakes from undesirable wiggling, we add a fairness term as:

$$\begin{aligned} E_f \;\; = \;\; & \textstyle\sum_i \|C_i - \frac{1}{2}(C_{i-1} + C_{i+1})\|^2 + \\ & \textstyle\frac{1}{m}\sum_{i,j} \|G_i^j - \frac{1}{2}(G_{i-1}^j + G_{i+1}^j)\|^2 + \\ & \textstyle\frac{1}{m}\sum_{i,j}((G_i^j - C_i)\cdot \mathbf{t}_i)^2 + \sum_i \|C_i - \frac{1}{m}\sum_j G_i^j\|^2, \end{aligned} \qquad (3)$$

where the terms enforce center curve smoothness, skin curve smoothness, planarity of cross sections, and centricity of skeletal curve with regard to the skin curves, respectively. The tangent vector $\mathbf{t}_i$ of $\mathcal{C}$ at C_i is estimated analytically using coefficients of a least squares cubic Bézier curve locally fitted to five adjacent points on the current skeletal curve.

Total error. The final optimization takes the form:

$$\min_{\{\mathbf{A}_i^j\}} (\omega_d E_d + \omega_s E_s + \omega_f E_f), \qquad (4)$$

where the weights ω_d, ω_s, and ω_f are tuned to guide the snake optimization. The data error term drives the snakes to evolve to the target point data. Even if there are some vertices on $\mathcal{S}$ that do not have any matching points on $\mathcal{R}$, minimization of the fairness error helps keep their cylindrical structure. In a final fitting stage, we geometrically refine the network of arterial snakes with inferred topology to better fit the input data $\mathcal{R}$. During the intermediate stages of snake evolution, we use the terms pertaining to individual snake terms for fine tuning positional estimates.

As the evolution of the arterial snake finds the correct topology of the lying arterial object, we perform the optimization using Equation 4 with weights $\omega_d = 1, \omega_s = 10$, and $\omega_f = 0.5$ while evolving the snakes, and set larger value of $\omega_f = 5$ to obtain a regularized mesh in the final fitting stage.

Regularizing snakes. We extract the network of arterial snakes based on the input data $\mathcal{R}$, which is often noisy. Although the fairness energy prevents the snakes from wiggling under the influence of noise, global relations present in the original models may not be well represented in the recovered snakes. In order to regularize the individual arterial snakes, we project their skeletal curves to least squares fitting planes, if the fitting errors are small. Subsequently,

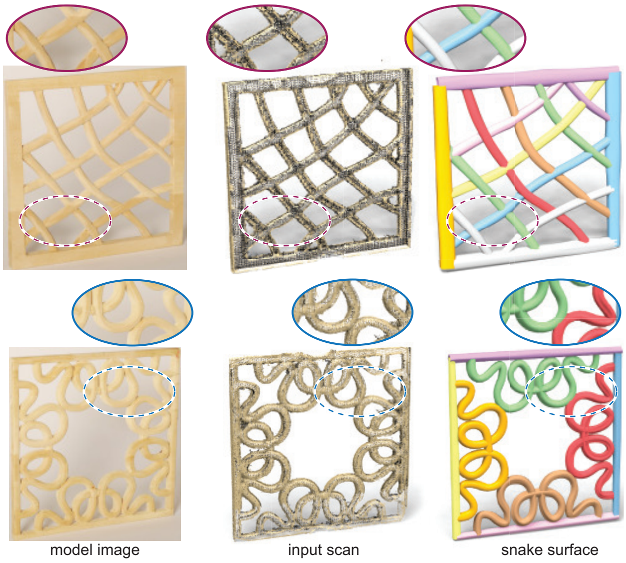

Figure 9: *The reconstructed results produced by our algorithm from two scanning data of real models.*

the planarized skeletal curves are projected to lines, or circles if the corresponding fitting error are negligible. Otherwise we smoothen the skeletal curves using cubic spline fitting. If snake pairs are nearly tangent, i.e., they mutually intersect at close by points, we locally move the snakes to enforce tangency relation (see Figure 8). Finally, we detect symmetry relations between the extracted snakes, and make such global relations exact using a simple symmetrization based deformation [Mitra et al. 2007] (see also [Gal et al. 2009]).

Implementation details. The positions of vertices can be represented as the linear function of the elements of the affine matrices as the initial vertex positions are known. In our optimization we optimize all the affine transformations instead of the vertex positions. The minimization of the total energy amounts to solving a large linear system with $12nm$ variables. We use an iterative Gauss-Newton algorithm to solve our unconstrained least squares problem. Further, we optimize for affine transformations associated mainly for the vertices at the front of the evolving snakes leading to an optimization with $O(m)$ variables. Figure 7 illustrates evolution of an arterial snake network with successive iterations.

Our algorithm alternates between topology and geometry optimization. While the skeletal curve connectivity captures the network topology, we use the cross-sectional profiles to represent geometric details. We use a coarse-to-fine evolution process where, in the topology optimization phase, we sample points from the input $\mathcal{R}$ and coarsely represent the cross sectional profiles using fewer nodes. During the geometry optimization phase, we upsample the cross-sectional profiles, and optimize for the vertex positions guided by the input data $\mathcal{R}$.

5 Results

We have tested our algorithm on a number of models, both real-captured models and synthetic data sets, under varying amounts of noise, outliers, and missing parts.

Our system can analyze and extract the various arterial structures from a scan input with noise, outliers, and missing parts, as shown in Figure 8. From the input scanning point cloud, the algorithm successfully recovers the layered arterial surfaces by the competing growing scheme. After regularizing the arterial network we obtain the final reconstructed model. In this example, the different arterial snakes intersect and meet with varying complexity. Generally, it is easy to handle the simple transversal intersection cases, even using local schemes. However, it is challenging to resolve multiple tangential intersections with large penetrations. Our algorithm successfully handles such cases leading to near-perfect reconstruction. In this example, we use input prior, and fit a rectangular cross-sectional profile in the final fitting stage.

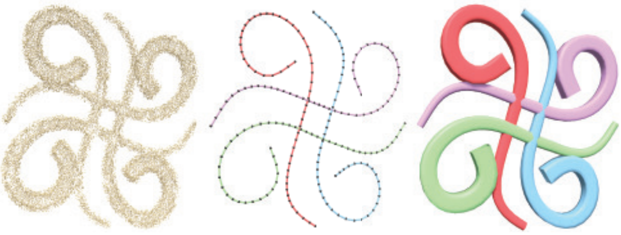

Figure 10: *Since we represent cross-sectional profiles of arterial snakes using closed polyline-segments, our framework can naturally handle varying cross-sectional profiles. In this synthetically scanned dataset the gradually varying cross-sections (evolving from rectangle to circular profiles) are correctly identified.*

Figure 9 shows the reconstructed results produced by our algorithm from two scanning data of real models. In the first example, two arterial snakes overlap partially and our algorithm correctly reconstructed the layer relationship. In the second example, our algorithm easily detects the curve network and the symmetries between the curves. Figure 1 shows a much more challenging example where a bunch of intersection cases occur in the model. Our algorithm can also recover the correct curve network and reconstruct the arterial surfaces.

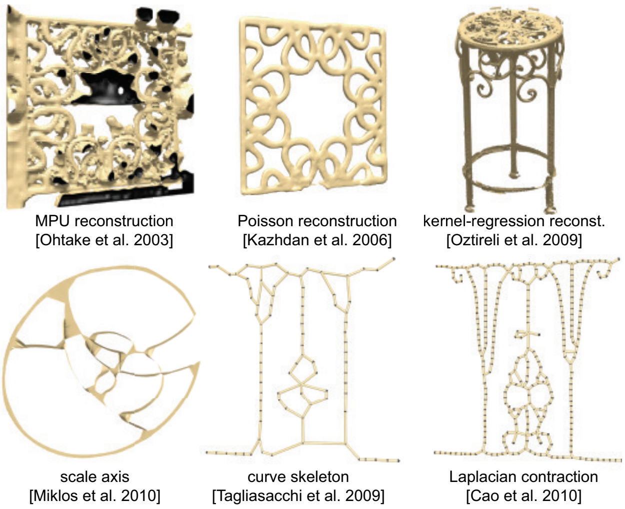

Figure 11: *Comparison with (top) state-of-the-art surface reconstruction algorithms and (bottom) curve skeleton definitions on noisy incomplete point sets (using respective author implementation). For scale-axis comparison we used [Oztireli et al. 2009] to obtain an initial surface. Unlike general skeleton extraction methods, we exploit the prior of input models being 1D structures to obtain structured skeletal backbone. Corresponding results using our algorithm are presented in Figures 9, 1, 8 and 3, respectively.*

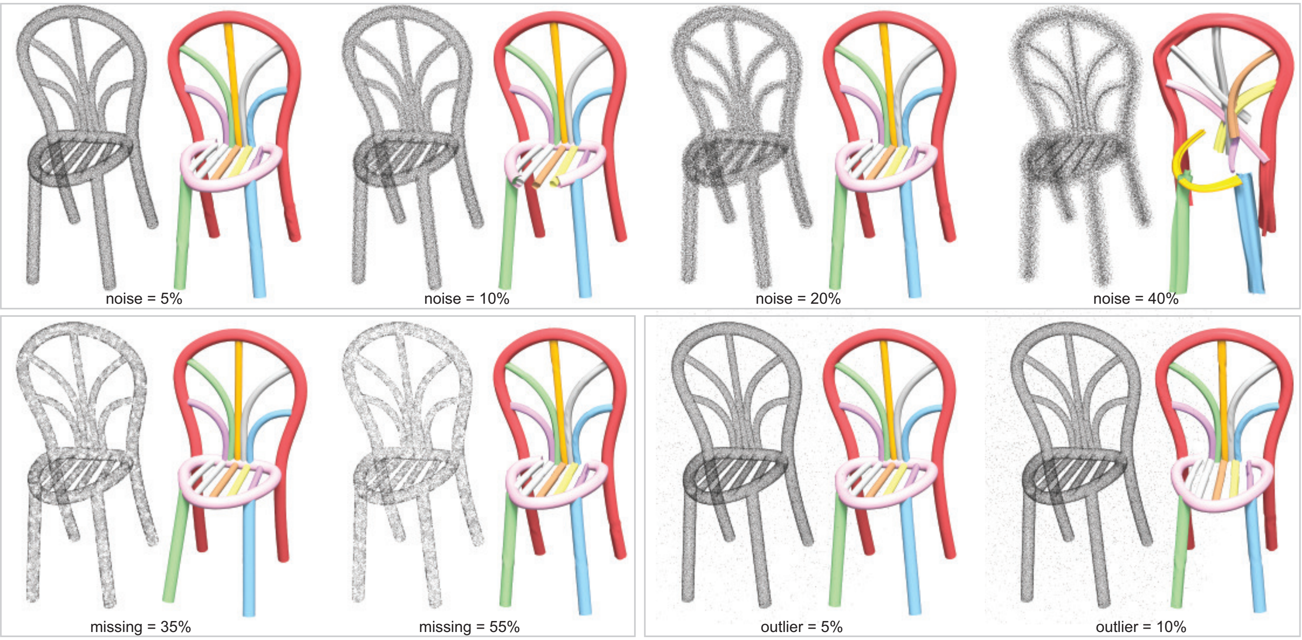

Figure 12: *Evaluation the accuracy and robustness of our algorithm. We adopt our algorithm on various quality of scanning data in the presence of noise, missing data, and outliers. First row: adding different amount of synthetic Gaussian noise to the data. Second row: (left) different amount of missing parts and (right) different amount of outliers.*

To evaluate the accuracy and robustness of our algorithm, we use various quality of scanning data in the presence of moderate noise, moderate amount of missing data, and moderate outliers, as shown in Figure 12. In the first test, we progressively add synthetic Gaussian noise to perturb the point positions and then apply our algorithm to compute the arterial snakes using exactly the same parameter settings. The upper row in Figure 12 shows the results of scanning data with various amount of noise, 5%, 10%, 20%, 40%, respectively. We can see that our algorithm is quite robust against the medium amount added noise (less than 20%), and as expected the failures are near regions with complicated junctions. It is interesting to see that one of the snakes is broken in the case of 10% noise. This is due to the failure of detection of snakelet at this part for this data. The lower row of Figure 12 shows several reconstructed results computed on the same scanning data with artificially parts removed and outliers inserted in different amounts, respectively.

The arterial snake can represent any profiles including non-circular profiles with varying tube width. In our implementation, for estimating the initial tubular parts, we first extract the snake-skeleton with circular profiles as this was found to be more robust on raw scans. Subsequently, the arterial snakes are optimized for arbitrary profiles by geometrically fitting the underlying cross-sectional profiles to the input data (see Figure 10).

The arterial snake based shape priors help us to robustly detect and extract curve networks from poor quality scanned data. In contrast it is challenging for existing methods to handle such data. Figure 11 shows examples of reconstructing surfaces from the data points in Figure 9 using MPU method [Ohtake et al. 2003] and Poisson surface reconstruction [Kazhdan et al. 2006]. With significant data parts missing, the methods fail to fill the surface at the incomplete portions and produce many bulges in the results. Although kernel-regression based reconstruction [Oztireli et al. 2009] produces significantly superior results (using $\sigma_r = 1, \sigma_n = 0.75$), but the method does not attempt to recover the underlying structure of the data. Note for scale-axis network [Miklos et al. 2010], we set $\delta = 0.005$.

Our approach treats the geometric details as multiple layers of arterial objects, and provides an extended ability for high quality geometric editing and semantic structure reusing. After covering the network of arterial objects, we can easily edit the center curves to composite new surfaces, see Figure 13. Specifically, we can manipulate and edit the shapes of the curves, change layering relationships, and the shapes of cross sections (see also the accompanying video).

Our algorithm can also handle detailed geometric structures like extracted reliefs (see Figure 14). Given a relief surface, we use the method of Zatzarinni et al. [2009] to extract the relief from the base surface. The curve structure of the relief is analyzed and discovered by our algorithm, enabling curve-level detail manipulation.

The running time of our algorithm depends on the number of data samples $\mathcal{R}$, the number of polygonal vertices m on profiles of snakes, and the growth rate γ of the snakes. In our implementation, we set $m = 16$ vertices on the profiles and the growth rate γ is set as 5% of the bounding box size. We implemented our system on an Intel 2.8GHz CPU with 2G RAM. For input data with around 100K points, the system typically takes around 5 minutes. However, our method being output sensitive can take longer on models with more complicated pattern network, e.g., the metal-stool example, Figure 1, took about 10 minutes, which is small compared to about 40 minutes of efforts to scan the stool from six views.

Limitations. Our algorithm is targeted towards input models that can be well represented by 1D curves. In parts where 2D surfaces exist, the algorithm fails to get started as we cannot detect any initial snakelets, e.g., in Figure 1 the algorithm remains oblivious to the thin disc like structure on the top of the stool. Also in presence of large missing parts and significant amount of noise, the algorithm breaks down as it cannot create a useful directional field to guide

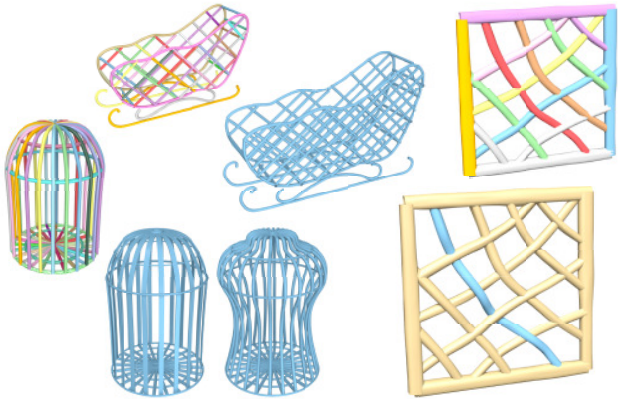

Figure 13: *Extracted arterial snake networks provide an easy, intuitive, and powerful editing metaphor involving snake-level geometrical edits, addition/deletion of snakes, and local layer ordering swapping. Such manipulations are challenging and tedious to perform using most shape deformation tools.*

the evolving snakes (see Figure 12).

In many objects (see Figure 2-right), the 1D ornamental structures are only part of the object. In such cases, we expect a segmentation to classify the data into parts that are suited for arterial-snake reconstruction and parts where standard surface reconstruction algorithms can be used. Further work is required to automate this phase.

Finally, although in principle our algorithm can be used to reconstruct very thin wire like structures, e.g., blood veins and hair strands, in most datasets our approach fails due to insufficient data quality. Understanding suitable sampling conditions for our algorithm is an interesting future direction.

6 Conclusion and Future Work

We have presented a novel approach for analyzing and reconstructing the arterial surfaces and decomposing them into a set of semantically useful arterial components. The key to the decomposition is to use the deformable snake model, which is inflated into the interior of the arterial component as much as possible. The structures of the individual arterial components are automatically determined using a simultaneous evolution scheme. The curve network representing the underlying surface is used for various applications including shape completion, reconstruction, topology repairing, and intuitive manipulation.

Our current algorithm has several limitations that we wish to address in our future work. We suppose the two arterial components intersect with each other at the overlapped region. It has difficulties handling surfaces with complex features and details which are caused by fuzzy surface boundaries, interleaved geometric details and perception-driven ambiguities. A possible way to handle such arterial surfaces may be by considering more complex composition operations at the overlapped regions such as linear blending or matting. Furthermore, we also would like to study other types of evolving surfaces with more complicated structures.

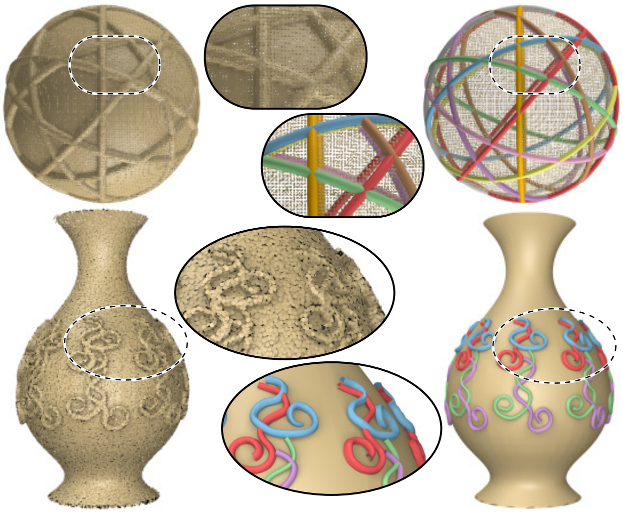

Figure 14: *Relief details, extracted from an input point set using [Zatzarinni et al. 2009], are analyzed and represented using arterial snakes. (Bottom) Additionally, in this case, the extracted base point set is fitted with a surface of revolution.*

Acknowledgements

We thank the anonymous reviewers for their constructive comments. Several people helped in generating comparison results for Figure 11 namely Junjie Cao and Oscar Au for Laplacian contraction, Misha Kazhdan for Poisson surface reconstruction, Balint Miklos for scale axis computation, Cengiz Oztireli for kernel-regression reconstruction, Andrea Tagliasacchi and Richard Hao Zhang for curve skeleton computation, and Guanghua Tan for MPU reconstruction. We thank Min Yue for her help in obtaining the physical models scanned for this paper, Martin Peternell and Johannes Wallner for their help with scanning the models, and Jonathan Balzer for video narration. Ligang Liu is supported by the 973 National Key Basic Research Foundation of China (No. 2009CB320801) and the joint grant of the National Natural Science Foundation of China and Microsoft Research Asia (No. 60776799). Niloy Mitra was partially supported by a Microsoft outstanding young faculty fellowship.

References

ALLEN, B., CURLESS, B., AND POPOVIĆ, Z. 2003. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Trans. on Graph 22*, 3, 587–594.

ATTENE, M., KATZ, S., MORTARA, M., PATANE, G., SPAGNUOLO, M., AND TAL, A. 2006. Mesh segmentation - a comparative study. In *Proc. Conf. on Shape Modeling and Appl.*, 14–25.

AU, O. K.-C., TAI, C.-L., CHU, H.-K., COHEN-OR, D., AND LEE, T.-Y. 2008. Skeleton extraction by mesh contraction. *ACM Trans. on Graph 27*, 3, 44–52.

BOTSCH, M., AND KOBBELT, L. 2003. Multiresolution surface representation based on displacement volumes. In *Eurographics*, 483–491.

CAO, J., TAGLIASACCHI, A., OLSON, M., ZHANG, H., AND SU, Z. 2010. Point cloud skeletons via laplacian-based contraction. In *Proc. Conf. on Shape Modeling and Appl.*

CHEN, X., GOLOVINSKIY, A., AND FUNKHOUSER, T. 2009. A benchmark for 3D mesh segmentation. *ACM Trans. on Graph 28*, 3, #73.

CORNEA, N. D., SILVER, D., AND MIN, P. 2007. Curve-skeleton properties, applications, and algorithms. *IEEE Trans. Vis. & Comp. Graphics 13*, 3, 530–548.

DEY, T. K., AND SUN, J. 2006. Defining and computing curve-skeletons with medial geodesic function. In *Proc. Symp. on Geometry Processing*, 143–152.

DUAN, Y., AND QIN, H. 2004. Intelligent balloon: A subdivision based deformable model for surface reconstruction of unknown topology. *Graphical Models 66*, 4, 181–202.

ESTEBAN, C. H., AND SCHMITT, F. 2003. A snake approach for high quality image-based 3D object modeling. In *Variational, Geometric and Level Set Methods in Comp. Vision*, 241–248.

ESTEVE, J., BRUNET, P., AND VINACUA, A. 2005. Approximation of a variable density cloud of points by shrinking a discrete membrane. *Computer Graphics Forum 24*, 2, 791–807.

GAL, R., SORKINE, O., MITRA, N. J., AND COHEN-OR, D. 2009. iWIRES: an analyze-and-edit approach to shape manipulation. *ACM Trans. on Graph 28*, 3, #33.

GIESEN, J., MIKLOS, B., PAULY, M., AND WORMSER, C. 2009. The scale axis transform. In *Proc. of Symp. on Computational Geometry*, 106–115.

HILAGA, M., SHINAGAWA, Y., KOHMURA, T., AND KUNII, T. L. 2001. Topology matching for fully automatic similarity estimation of 3D shapes. In *Proc. ACM SIGGRAPH*, 203–212.

IGARASHI, T., AND MITANI, J. 2010. Apparent layer operations for the manipulation of deformable objects. *ACM Trans. on Graph 29*, 4, #110.

KASS, M., WITKIN, A., AND TERZOPOULOS, D. 1988. Snakes: Active contour models. *International journal of computer vision 1*, 4, 321–331.

KAZHDAN, M., BOLITHO, M., AND HOPPE, H. 2006. Poisson surface reconstruction. In *Proc. Symp. on Geometry Processing*, 61–70.

KIRBAS, C., AND QUEK, F. 2004. A review of vessel extraction techniques and algorithms. *ACM Computing Surveys 36*, 2, 81–121.

KOBBELT, L., CAMPAGNA, S., VORSATZ, J., AND SEIDEL, H.-P. 1998. Interactive multi-resolution modeling on arbitrary meshes. In *Proc. ACM SIGGRAPH*, 105–114.

LEE, Y., AND LEE, S. 2002. Geometric snakes for triangular meshes. *Computer Graphics Forum 21*, 3, 229–238.

LEE, Y., LEE, S., SHAMIR, A., COHEN-OR, D., AND SEIDEL, H.-P. 2005. Mesh scissoring with minima rule and part salience. *Computer Aided Geometric Design 22*, 5, 444–465.

LIPMAN, Y., SORKINE, O., LEVIN, D., AND COHEN-OR, D. 2005. Linear rotation-invariant coordinates for meshes. *ACM Trans. on Graph 24*, 479–487.

LIU, S., MARTIN, R. R., LANGBEIN, F. C., AND ROSIN, P. L. 2006. Segmenting reliefs on triangle meshes. In *Proc. ACM Symp. Solid and Physical Modeling*.

LIU, L., TAI, C.-L., JI, Z., AND WANG, G. 2007. Non-iterative approach for global mesh optimization. *Computer-Aided Design 39*, 9, 772–782.

LIU, S., MARTIN, R. R., LANGBEIN, F. C., AND ROSIN, P. L. 2007. Segmenting periodic reliefs on triangle meshes. In *Mathematics of Surfaces XII*.

LIVNY, Y., YAN, F., OLSON, M., CHEN, B., ZHANG, H., AND EL-SANA, J. 2010. Automatic reconstruction of tree skeletal structures from point clouds. *ACM Trans. on Graph 29*, 5, to appear.

MEHRA, R., ZHOU, Q., LONG, J., SHEFFER, A., GOOCH, A., AND MITRA, N. J. 2009. Abstraction of man-made shapes. *ACM Trans. on Graph 28*, 5, #137.

MIKLOS, B., GIESEN, J., AND PAULY, M. 2010. Discrete scale axis representations for 3D geometry. *ACM Trans. on Graph 29*, 3.

MITRA, N. J., NGUYEN, A., AND GUIBAS, L. 2004. Estimating surface normals in noisy point cloud data. *International Journal of Computational Geometry and Applications 14*, 4-5, 261–276.

MITRA, N. J., GUIBAS, L., AND PAULY, M. 2007. Symmetrization. *ACM Trans. on Graph 26*, 3, #63.

OHTAKE, Y., BELYAEV, A., ALEXA, M., TURK, G., AND SEIDEL, H.-P. 2003. Multi-level partition of unity implicits. *ACM Trans. on Graph 22*, 3, 463–470.

OZTIRELI, C., GUENNEBAUD, G., AND GROSS, M. 2009. Feature preserving point set surfaces based on non-linear kernel regression. *Computer Graphics Forum 28*, 2, 493–501.

PATANÉ, G., SPAGNUOLO, M., AND FALCIDIENO, B. 2008. Reeb graph computation based on a minimal contouring. In *Proc. Shape Modeling and Applications*, 73–82.

SHAMIR, A. 2008. A survey on mesh segmentation techniques. *Computer Graphics Forum 27*, 6, 1539–1556.

SHARF, A., LEWINER, T., SHAMIR, A., KOBBELT, L., AND COHEN-OR, D. 2006. Competing fronts for coarse-to-fine surface reconstruction. *Computer Graphics Forum 25*, 3, 389–398.

SHARF, A., LEWINER, T., AND SHAMIR, A. 2007. On-the-fly curve-skeleton computation for 3D shapes. *Computer Graphics Forum 26*, 3, 323–328.

SIDDIQI, K., AND PIZER, S. 2008. *Medial Representations: Mathematics, Algorithms and Applications*. Springer.

SIMARI, P., KALOGERAKIS, E., AND SINGH, K. 2006. Folding meshes: hierarchical mesh segmentation based on planar symmetry. In *Proc. Symp. on Geometry Processing*, 111–119.

SORKINE, O., LIPMAN, Y., COHEN-OR, D., ALEXA, M., RÖSSL, C., AND SEIDEL, H.-P. 2004. Laplacian surface editing. In *Proc. Symp. on Geometry Processing*, 179–188.

TAGLIASACCHI, A., ZHANG, H., AND COHEN-OR, D. 2009. Curve skeleton extraction from incomplete point cloud. *ACM Trans. on Graph 28*, 3, #71.

WANG, W., JÜTTLER, B., ZHENG, D., AND LIU, Y. 2008. Computation of rotation minimizing frames. *ACM Trans. on Graph 27*, 1, 1–18.

WEYRICH, T., DENG, J., BARNES, C., RUSINKIEWICZ, S., AND FINKELSTEIN, A. 2007. Digital bas-relief from 3D scenes. *ACM Trans. on Graph 26*, 3, 32.

ZATZARINNI, R., TAL, A., AND SHAMIR, A. 2009. Relief analysis and extraction. *ACM Trans. on Graph 28*, 5, #136.

ZHANG, E., HAYS, J., AND TURK, G. 2007. Interactive tensor field design and visualization on surfaces. *IEEE Transactions on Visualization and Computer Graphics 13*, 1, 94–107.

ZHOU, K., HUANG, X., WANG, X., TONG, Y., DESBRUN, M., GUO, B., AND SHUM, H.-Y. 2006. Mesh quilting for geometric texture synthesis. *ACM Trans. on Graph 25*, 3, 690–697.