Generalized Cylinder Decomposition

Yang Zhou[1] Kangxue Yin[1] Hui Huang[1]* Hao Zhang[2] Minglun Gong[3] Daniel Cohen-Or[4]
[1]Shenzhen VisuCA Key Lab / SIAT [2]Simon Fraser University [3]Memorial University of Newfoundland [4]Tel-Aviv University

Abstract

Decomposing a complex shape into geometrically simple primitives is a fundamental problem in geometry processing. We are interested in a shape decomposition problem where the simple primitives sought are *generalized cylinders*, which are ubiquitous in both organic forms and man-made artifacts. We introduce a quantitative measure of *cylindricity* for a shape part and develop a cylindricity-driven optimization algorithm, with a global objective function, for *generalized cylinder decomposition*. As a measure of geometric simplicity and following the minimum description length principle, cylindricity is defined as the cost of representing a cylinder through skeletal and cross-section profile curves. Our decomposition algorithm progressively builds local to non-local cylinders, which form over-complete covers of the input shape. The over-completeness of the cylinder covers ensures a conservative buildup of the cylindrical parts, leaving the final decision on decomposition to global optimization. We solve the global optimization by finding an exact cover, which optimizes the global objective function. We demonstrate results of our optimal decomposition algorithm on numerous examples and compare with other alternatives.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

Keywords: generalized cylinder, optimal shape decomposition

1 Introduction

Decomposing a complex shape into geometrically simple primitives is one of the most fundamental problems in geometry processing. The main considerations for what constitute a *simple* primitive include compactness of representation, ease of manipulation, and computational efficiency. Convexity, planarity, parameterizability, and smoothness are some of the most well-known simplicity criteria. The key challenge to shape decomposition is to improve the trade-off between simplicity of the primitives and size of the decomposition, with the ultimate goal being to obtain the smallest number of sufficiently simple components.

In this paper, we are interested in a shape decomposition problem where the simple primitives sought are *generalized cylinders* or *GCs*, for short. Cylindrical parts are ubiquitous in both organic forms (e.g., the torsos and limbs of various creatures) and man-made artifacts due to their versatile functionality and ease of manufacturing. Compared to other simplicity criteria such as convexity, generalized cylinder covers a much larger scope of shapes, leading to decompositions with fewer parts and more compact shape abstractions; see Figure 1. With their ease of control, arising from a natural tie to curve skeletons, GCs are frequently adopted for shape modeling [Zheng et al. 2011; Jacobson et al. 2014; Baerentzen et al. 2014] and interactive surface reconstruction [Li et al. 2010; Chen et al. 2013; Yin et al. 2014].

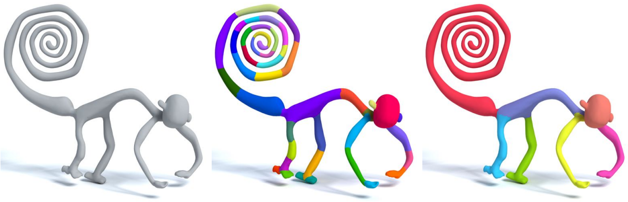


Figure 1: *Generalized cylinder decomposition obtained by our algorithm (right) produces considerably fewer parts compared to approximate convex decomposition (middle).*

Despite their ubiquity and importance in shape modeling, GCs have not been well-studied in the context of shape decomposition. We are not aware of any measure which quantifies the generality of cylindrical primitives or any work which optimizes cylindrical shape decomposition with a global objective. Typically, the extraction of cylindrical parts from a given shape is a by-product of skeletonization, e.g., [Au et al. 2008; Reniers et al. 2008], via primitive fitting, e.g., [Raab et al. 2004; Mortara et al. 2004; Attene et al. 2006], or based on locally-defined cut or clustering criteria, e.g., [Tagliasacchi et al. 2009; Goyal et al. 2012].

We introduce a quantitative measure of *cylindricity*, for a shape part and develop a cylindricity-driven optimization algorithm, with a *global* objective function, for generalized cylinder decomposition. The key premise behind our cylindricity definition is that cylindricity is a measure of *geometric simplicity*. Following the *minimum description length* (MDL) principle [Grunwald 2007], we equate simplicity to *low representation cost*.

Specifically, we model a cylindrical shape part using its central axis, i.e., a curve skeleton, and cross-sectional profile curves about the skeleton. Cylindricity is defined by combining the costs of representing the skeleton and the profiles via *recursive geometric interpolation*, which progressively adds profile curves to refine the interpolated surface so as to approximate the input shape to within a prescribed tolerance. While a perfect cylinder has the minimum representation cost since it has a straight skeleton and uniform profiles, more cross-sectional profiles are required for the interpolation to well approximate a more general cylinder. In the latter case, the simplicity of the cylinder goes down and its cylindricity increases. Geometrically, the representation costs in the cylindricity definition can be interpreted as measures of the *straightness* of the skeleton and the *variations* among the profiles respectively.

Cylindricity plays a central role in our *local-to-global* decomposition approach. Foremost, the global objective function follows the MDL principle; it combines a cylindricity term for geometric simplicity and a term for the size of the decomposition, reflecting a trade-off. Cylindricity also controls the local-to-global analysis which builds the GCs progressively. Specifically, given an input

*Corresponding author: Hui Huang (hhzhiyan@gmail.com) shape, we first construct an *over-complete* set of local cylinders accounting for all potential curve skeleton directions. The local cylinders are then merged based on the cylindricity measure to produce the set of candidate GCs in varying lengths. The candidate set forms an over-complete cover of the input shape, where any *exact* subcover is precisely a decomposition of the shape. We solve an *Exact Cover Problem* (ECP) [Vazirani 2001] by efficiently enumerating its solutions using Knuth's Algorithm X [Knuth 2000]. Finally, we pick the top ECP solution based on our objective function to obtain an optimal decomposition.

Compared to classical approaches such as clustering and recursive partitioning, the ECP solution represents a relatively unexplored alternative to solve hard shape decomposition problems [Hu et al. 2014]. Maintaining an over-complete cover until the final global optimization step avoids making potentially premature, local decisions too early. Moreover, enumerating the ECP solutions allows a larger set of decompositions to be explored, which can be beneficial in a variety of global optimization settings.

We show results of our algorithm on numerous examples and compare to those obtained by convex decomposition, primitive fitting, profile clustering, and skeleton-driven segmentation, to demonstrate the advantages of using the global cylindricity-driven search. A first example is shown in Figure 1, where we compare our method to approximate convex decomposition. Several applications which benefit from our optimal decomposition, including curve skeleton extraction, shape approximation and modeling, are also shown. In particular, we highlight the versatility and representation capability of the skeleton+profile shape approximation enabled by our decomposition, compared to quadric mesh simplification and the recently proposed sphere-mesh representations [Thiery et al. 2013].

2 Related work

Shape decomposition is a well-studied problem with extensive surveys, e.g., [Chazelle 1987; Shamir 2008]. Many methods are designed to compute meaningful segmentations based on shape semantics, e.g., the minima rule [Hoffman and Richards 1984], engineering constraints [Li et al. 2011], Reeb graphs [Biasotti et al. 2008], or those learned from training data [Kalogerakis et al. 2010]. Our work has a different focus: instead of taking a data- or knowledge-driven approach, our goal is to decompose a shape into *geometrically simple* primitives. In this section, we mainly cover methods guided by purely geometric criteria.

Planarity [Cohen-Steiner et al. 2004], parameterizability [Sander et al. 2003] and smoothness [Mangan and Whitaker 1999] are well-known simplicity criteria for surface primitives. For solid primitives, convexity [Lien and Amato 2004; Asafi et al. 2013] is most often employed; other criteria include ellipticity [Simari and Singh 2005], volumetric thickness [Shapira et al. 2008], and pyramidality [Hu et al. 2014]. While GCs are solid primitives, our algorithm operates over the surfaces of a 3D object.

Convex decomposition. Cognitive studies have revealed that the human visual system tends to divide a 3D shape into parts around concave regions [Hoffman and Richards 1984], implying that the resulting parts are likely to be nearly convex. Many methods have been developed for convex shape decompositions, e.g., [Lien and Amato 2004; Ren et al. 2011; Asafi et al. 2013; van Kaick et al. 2014]. However, one may question whether convexity is too "low-level" as a simplicity criterion, which would result in over-segmentation, as shown in Figure 1 the classical example of the monkey tail. Our work offers GCs as an alternative. The resulting decomposition algorithm would not divide a bendy tail yet it is still sensitive to constriction points, just as convexity would, due to the consideration of smoothness of profile transitions in cylindricity; see Figure 3 for an illustration.

Primitive fitting. There are decomposition methods which fit elliptical [Simari and Singh 2005], cylindrical (straight and bead-like cylinders) [Raab et al. 2004], or other simple primitives [Attene et al. 2006] to a given 3D shape. Common to most primitive fitting methods is the simplicity of the *parametric* representations of the primitives considered. GCs do not possess such simplicity due to their generality. As such, the resulting decomposition criterion is higher-level and typically results in fewer parts.

Skeleton-based segmentation. A connection between GCs and curve skeletons clearly exists. For example, one can obtain a decomposition as a by-product of curve skeleton extraction [Au et al. 2008; Reniers et al. 2008; Tagliasacchi et al. 2012]. Early work by Li et al. [2001] carries out a planar cross-section sweep along extracted curve skeletons to find locations, referred to as critical points, where the geometry and/or topology of the cross-sectional profiles changes dramatically. Reniers et al. [2008] develop a multi-scale skeleton extraction method, which is based on a global importance measure and can yield shape segmentations using the skeleton-to-boundary mapping. Similarly, Au et al. [2008] and Zheng et al. [2011] extract GCs by examining thicknesses about extracted curve skeletons and find cuts based on concavity and length of cut boundaries. Other multi-scale skeleton extraction methods include [Miklos et al. 2010] and [Jalba et al. 2015].

Cross-sectional shape profiles also play an important role in our work. However, the key difference is that we obtain the final set of GCs by solving a global optimization, rather than relying on local cut criteria as in previous works. Moreover, with over-complete GC covers, we consider multiple skeleton directions over any part of the shape. Our cylindricity measure tolerates more general cross-sectional profiles and leads to GCs with more diverse geometries.

Generalized cylinders. Some previous works had considered GCs for shape decomposition and reconstruction. Mortara et al. [2004] define a tubular region over a 3D shape as one which intersects a suitable sized sphere in two closed curves. Their tube construction is via a seeded region growing with heuristically set sphere positions and radii. Several surface reconstruction schemes have been developed to extract GCs from raw point clouds based on approximate rotational symmetry detection [Tagliasacchi et al. 2009], deformable snakes [Li et al. 2010], or interactive sweeps [Chen et al. 2013; Yin et al. 2014]. None of the works above account for cylindricity of the cylindrical parts obtained, nor do they formulate for or seek an optimal GC decomposition.

More relevant is the work of Goyal et al. [2012] for GC decomposition, which performs an affinity propagation clustering of cross-sectional slices. The key difference is that their clustering scheme is based on a pairwise profile similarity, which serves as the only measure that reflects the goal of finding cylindrical parts. Such schemes essentially group similar profiles based on a local criterion without using a quality measure for the GCs obtained.

Since GCs are intrinsic symmetric shapes, our approach is also related to methods for decomposing shapes into segments with approximate intrinsic symmetries [Solomon et al. 2011]. Nevertheless, a general shape often has more than one intrinsic rotational symmetry, which leads to multiple possible decompositions. Our key novelty is to respect all possible local symmetries and select the ones that may yield the best global decomposition through optimizing the cylindricity measure.

Optimizing decompositions. Among the large number of methods for shape decomposition, only a relatively few define explicit

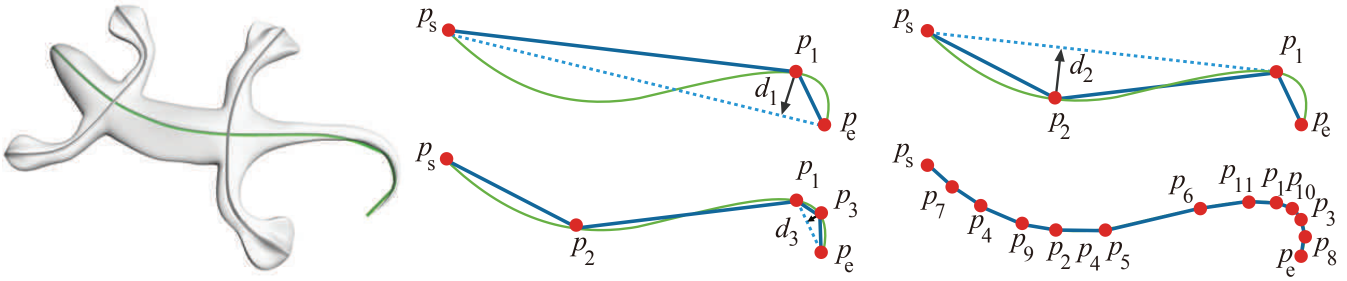


Figure 2: *Illustration of the DP algorithm used to measure the straightness score E_s in* (2)*. The skeletal curve is iteratively approximated using a polyline and E_s measures how much the polyline differs from a straight line.*

optimization objectives. Some provide a variational formulation, e.g., [Cohen-Steiner et al. 2004], some minimize part count under a tolerance, e.g., [Ren et al. 2011], while others seek a good trade-off between part count and another application-dependent cost, e.g., [Asafi et al. 2013; Hu et al. 2014].

Clustering seems to be the most popular optimization strategy. Cohen-Steiner et al. [2004] solves their variational problem via k-means clustering. Other examples include fuzzy clustering [Katz and Tal 2003] and mean-shift [Shamir et al. 2006]. Most instances of the optimal shape decomposition problems e.g., k-means, are NP-hard. Hence, enumerative solutions such as branch-and-bound, e.g., as in [Ren et al. 2011], and Algorithm X, e.g., as in [Hu et al. 2014], are sensible choices. Our ECP solution is inspired by a similar solution pipeline in Hu et al. [2014], which seeks an optimal approximate pyramidal decomposition. In their work, the cells and blocks, which are counterparts to our local GCs, form partitions, rather than over-complete covers, of the input shape.

Shape approximation. One of the main benefits of obtaining an optimal decomposition for a 3D shape is the resulting *compact* skeleton+profile shape representation. The representation not only offers the ease for shape manipulation but also provides a means for *progressive* shape approximation. From the perspective of shape approximation, our work shares the same motivation as classical mesh decimation, e.g., [Garland and Heckbert 1997], and the recent sphere-mesh representation [Thiery et al. 2013]. While the latter two approaches linearly interpolate over mesh vertices or spheres for approximation, the skeleton+profile representation relies on higher-order interpolations over profile curves to approximate cylindrical shapes, where the profile curves themselves are constructed via interpolation over point primitives.

3 Cylindricity measure

The cylindricity measure plays a central role in our method. Generally speaking, it measures the simplicity of a GC in MDL terms. Our premise is that a *right (perfect) cylinder* is a low-entropy shape due to its very compact description (a 2D profile curve plus height). With a right cylinder serving as a role model, the cylindricity aims to measure the amount of additional information needed to describe a given shape. In this regard, the development of the cylindricity measure is similar to that of circularity [Zunic et al. 2010] and convexity [Zunic and Rosin 2004], as they aim to quantify how much a closed curve departs from a circle or a convex figure.

In the following section, we develop a quantitative measure for how much a given shape departs from a right cylinder. It measures the simplicity of a GC by approximating its minimal descriptive cost. The key idea is to model or describe the GC with a curve skeleton, and a minimal set of cross-sectional profiles along its central axis. Then, intuitively, the (MDL) cost entailed with such a description of the shape is the cylindricity measure.

Our proposed cylindricity measure is conceptually simple, and consists of only two terms, straightness and profile variation:

$$cylindricity = E_s + \alpha E_v. \tag{1}$$

The straightness term E_s measures how much the central axis of the given shape differs from a straight line, and the profile variation term E_v measures how much the profile of the given shape varies along its axis. The weighting parameter α is set to 1 by default.

The straightness term. As shown in Figure 2, given an axis curve, the Douglas-Peucker algorithm [1973] is used to compute its polyline approximation. We first approximate the axis curve with a straight line that connects its start point p_s and end point p_e. The point p_1 along the curve that has the maximum distance to the line p_sp_e is then located. If this distance d_1 is larger than a given threshold ϵ, p_1 is inserted as a control point, breaking the original straight line p_sp_e into a polyline $p_sp_1p_e$; see Figure 3. The process is repeated until no point on the curve can be found with its distance to the polyline larger than the (default) threshold $\epsilon = 0.005$. Once the process stops, the straightness is computed using:

$$E_s(n) = C + \sum_{i=1}^{n} d_i, \tag{2}$$

where n is the number of control points inserted and C is a constant parameter. Larger C encourages long polyline axis, whereas smaller C encourages breaking the axis into multiple short ones. For example, if a given axis has a straightness score of E, then after breaking it into two parts at its first control point location p_1, the total straightness score of these two parts is equal to $E - d_1 + C$, which may increase or decrease by adjusting C. We can regard C as the description length needed for encoding the two endpoints, and each distance d_i as the description length needed for adding the corresponding control point.

Figure 4(a) plots how $E_s(n)$ varies as the number of control points n increases. It shows that after the first few control points are inserted ($n = 11$ in this case), increasing n either directly or indirectly through adjusting the threshold ϵ does not affect the value of $E_s(n)$ much. Hence, we consider that the $E_s(n)$ measure is robust against sampling rate requirement.

The profile variation term. This term measures how much the profiles of the given shape varies along its axis. To compute the measure, we generate a set of dense and uniformly spaced profile curves; see the gray curves in Figure 3. This is done by sampling the central axis of the shape using a set of dense points $\{s_i\}$, and

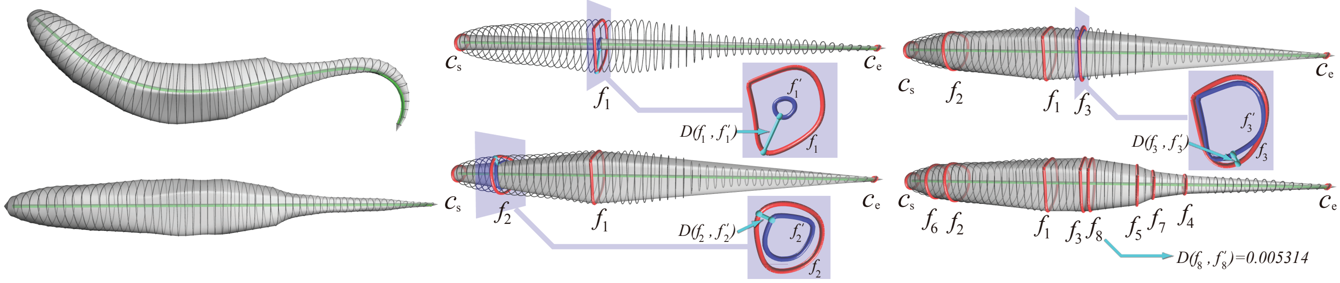
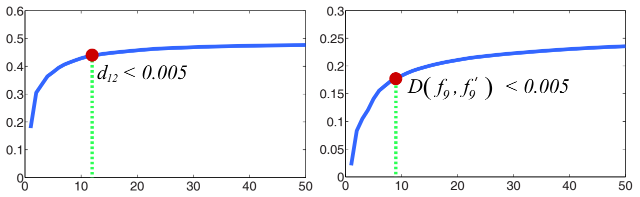


Figure 3: *Illustration of the profile insertion and intra slice error used to measure the profile variation E_v in (4). On the bottom left we show the rectification of the input GC. Through iteratively inserting control profiles, the rectified GC (bottom left) is approximated by interpolating between adjacent profile curves.*



(a) $E_s(n)$.

(b) $E_v(m)$.

Figure 4: *Straightness (a) and profile variation (b) scores w.r.t. the number of control points and control profiles, respectively.*

computing a profile curve c_i at each s_i by intersecting the shape with a plane that is perpendicular to the central axis at s_i. Next, we rectify the shape by transforming all the profile curves so that they reside on parallel planes and their centroids rest on a straight line. Equal number of 2D points $\{c_i^k\}$ are then used to uniformly sample each profile curve c_i on its plane.

With all the profiles being aligned and sampled with points, the difference between any given two profiles c_i and c_j is calculated using their Hausdorff distance as shown in the Figure 3 insets:

$$D(c_i, c_j) = \max\{\max_k(\min_h d(c_i^k, c_j^h)),\ \max_h(\min_k d(c_i^h, c_j^k))\}. \tag{3}$$

Furthermore, connecting the matching sample points on two end profiles c_s and c_e provides a 3D approximated shape that linearly interpolates the two profiles. Here, the optimal match is computed under the ordering constraint, i.e., c_s^k matches to c_e^{k+m} for all point k. The constant index offset m is computed by minimizing the distortion between the profile c_s to c_e.

Once the approximated shape is computed, the profile on the original shape that deviates from the approximated shape the most in terms of the aforementioned profile distance measure (3) is then selected; see Figure 3. Following the spirit of the Douglas-Peucker algorithm, if the distance is larger than a given threshold ϵ, a new control profile f_1 is selected at the location s_1 and a new approximated shape is obtained by linearly interpolating c_s with f_1 and f_1 with c_e. The process repeats until the distances between all profiles and their corresponding approximations are small enough, i.e., below ϵ. At this point, the profile variation term is computed using:

$$E_v(m) = D(c_s, c_e) + \sum_{i=1}^{m} D(f_i, f_i'), \tag{4}$$

where m is the number of control profiles and f_i' (blue curve in Figure 3) is the corresponding approximation curve for control profile f_i (red curve in Figure 3). Again, from an MDL point of view, $D(c_s, c_e)$ is the description length needed for encoding the profile variation between the two end profiles. Each $D(f_i, f_i')$ is the description length needed for encoding each control profile f_i.

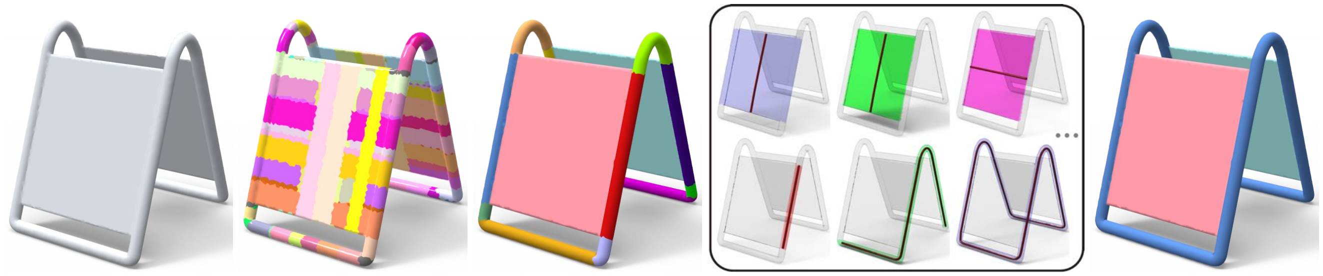
Note that the same as the straightness $E_s(n)$, the profile variation $E_v(m)$ is independent of scale and performs robustly as the number of control profiles m increases; see Figure 4(b). This makes our cylindricity measure (1) sampling independent.

4 Decomposition algorithm

As shown in Figure 5, our shape decomposition algorithm contains three major steps. Given a 3D shape S, we start with computing a large set of local GCs. We then merge local GCs into non-local ones that form an over-complete cover for S. At the last step, we solve the exact cover problem (ECP) to explore various combinations of non-local GCs and pick the globally optimal one as the decomposition of S. Throughout the algorithm, the cylindricity measure evaluates the qualities of both local and non-local GCs, as well as the final shape decomposition.

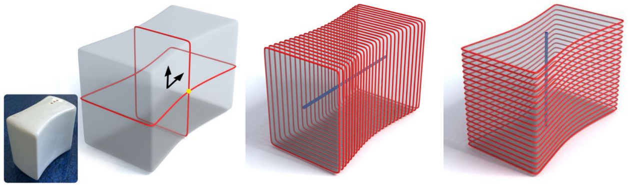
Overlapping local GCs. To compute a set of local GCs for input shape S, we uniformly sample the surface of S into a set of points Ω and construct one or more local GCs for each point sample. A local GC is defined by an axis and a small set of profile curves that are generated by intersecting S with the cross-section planes orthogonal to the axis. To compute candidate local axes for p, we use the ROSA technique [Tagliasacchi et al. 2009]. Starting with a random initial cutting plane that passes p, a profile curve can be calculated and an updated cutting plane can be obtained based on the surface normals along that profile curve. The process continues with the updated cutting plane, until it converges into a stable *ROSA plane*. The normal direction of this plane is the local rotational symmetry axis (ROSA), which is used as the axis of local GC. For a circular cylinder, different initial cutting planes converge to the same ROSA plane. However when S is a general shape, different ROSA planes may be obtained [Tagliasacchi et al. 2009]. This non-uniqueness introduces ambiguities in evaluating the cylindricity of S in the vicinity of p, since different approaches can be used to approximate S with GCs; see Figure 6.

Resolving ambiguities using only local information is likely to be premature and harmful; see Figure 7. Hence, we define a multiplicity of ROSA planes to generate a number of overlapping local GCs. To do so, 24 initial cutting planes with a fair distribution over the Gaussian sphere are used. The ROSA directions that these planes converge to form a set of potential GCs at sample p. The quality of each of these potential GCs is evaluated using the cylindricity measure (1), and only the GCs with cylindricity values smaller than



(a) Input. (b) 196 local GCs. (c) 16 non-local GCs. (d) 116 candidate GCs for ECP. (e) Final decomposition.

Figure 5: *Our decomposition algorithm overview: given a shape (a), we first compute its local GC set (b), and then merge the local GCs into fewer overlapping non-local GCs (c) that form an over-complete cover for the shape (a). The final decomposition (e) is obtained by solving the exact cover problem from various candidate GC combinations (d) of non-local GCs (b).*



(a) Two ROSA directions. (b) GC-appro 1. (c) GC-appro 2.

Figure 6: *Two approximations to a toothpick box model using GCs.*

a threshold τ (set to 0.15 by default) are kept. Hence, the local GCs may not cover the input shape completely. As shown in Figure 7(a), due to surface deformation at areas near joints (dark gray), the local GCs found there often have high cylindricity values, resulting them being pruned out.

Over-complete shape cover using non-local GCs. To create the candidate GCs for the cover, we merge local GCs into non-local ones, which form an over-complete cover of the input shape S. The merging is driven by the cylindricity measure, where two GCs are merged *iff.* the resulting GC has a lower cylindricity measure than their sum. That is:

$$cylindricity(A \oplus B) < cylindricity(A) + cylindricity(B), \tag{5}$$

where A and B can be either local GCs or non-local GCs obtained from the previous merging operation, and $A \oplus B$ represents the non-local GC obtained after merging A and B.

To calculate the cylindricity of $A \oplus B$, we need to first determine its axis curve. We use the axis end points in A and B as key points and connect them using a piece-wise Hermite curve, whose tangential directions at these key points are aligned with the corresponding GC directions. As shown in Figure 8, when the orientations of both A and B are lined up with the $A \oplus B$, the Hermite curve can closely approximate its curve skeleton. On the other hand, when the orientations of the two are not compatible, the Hermite curve contains sharp turns, e.g., in Figure 8 bottom row, the angular changes along the curve greater than 90^0. Merging incompatible curves induces a high cylindricity measure for $A \oplus B$ and resulting in the merge being rejected. Figure 5(c) illustrates that after merging we can get only a dozen non-local GCs.

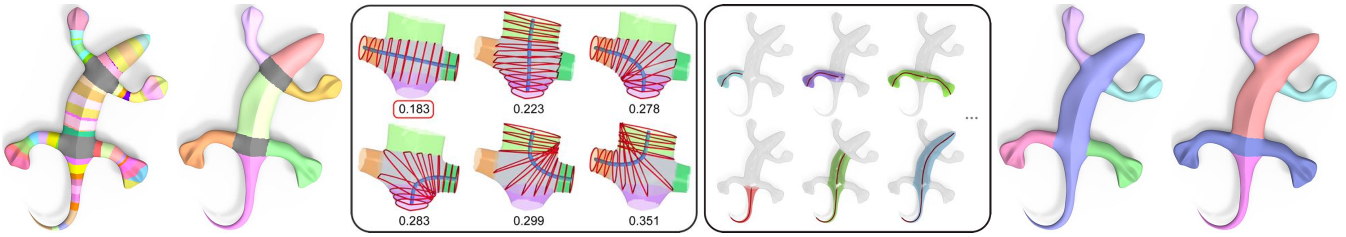
The non-local GCs may overlap or intersect each other. However, the decomposition target is to decompose parts that do not overlap. Thus, the non-local GCs are split into non-overlapping building blocks, which cover the shape S. More precisely, for any two non-local GCs A and B, we test if they overlap by computing $A \cap B$. When the area of $A \cap B$ is small enough, we simply retract both A and B to avoid overlapping. Otherwise, we split A and B into three portions: $A - B$, $B - A$ and $A \cap B$. Two new GCs are constructed for $A - B$ and $B - A$ using the axis directions of A and B, respectively. A third GC is constructed for $A \cap B$ if the axis of A and B are the same at $A \cap B$ area. Otherwise, two overlapping GCs are constructed for $A \cap B$, using the axis directions of A and B, respectively. Such intersection calculations allow regions with ambiguous GC orientations being separated from the rest. For example, the flat areas of the parking sign in Figure 5 has ambiguous GC orientations, whereas the tubular areas are not. Intersection calculation splits the two areas, which can hardly be achieved using conventional shape decomposition methods.

As shown in Figures 7(a-b), when joints are not covered in local GCs, they are still uncovered in the merged non-local GCs. To guarantee a full cover, we utilize the smoothness constraint and compute the candidate axes through interpolate Hermite curves between the axes of adjacent GCs. For example, at an X-shaped joint, there are six different ways of connecting the adjacent GCs, for which we generate six joint GCs as shown in Figure 7(c). These joint GCs cover the same (dark gray) region with different central axes, hence bridging among adjacent non-local GCs.

Candidate GCs for ECP. The non-local GCs obtained above, now consisting of a sequence of building blocks, form a set of candidates for the decomposition of S. Instead of directly solving a global optimization problem for decomposition, which is likely computationally complex and expensive, we reduce it into an Exact Cover Problem (ECP). An exact cover can be extracted from the over-complete cover $\mathscr{C}$ provided by the candidate GCs. Each exact cover is a subcover of $\mathscr{C}$ which completely covers S. That is, an exact cover is precisely a decomposition of S. In particular, we look for the subcover $\mathscr{C}_s$ that minimizes the following objective:

$$\mathscr{C}_s = \operatorname*{argmin}_{\mathscr{C}_s \subset \mathscr{C}} \sum_{g \in \mathscr{C}_s} cylindricity(g), \quad s.t. \bigcup_{g \in \mathscr{C}_s} g = S. \tag{6}$$

In theory, any random subset of the non-local GCs generated from the aforementioned merging process can be a candidate GC. However, some of these subsets may contain disconnected GCs and some others may contain adjacent but incompatible GCs. Considering these subsets when optimizing (6) will introduce redundant calculations. Hence, here we only generate candidate GCs using the following criteria: i) Each of the non-local GCs generated from the aforementioned merging process is a candidate GC; ii) a group



(a) 143 local. (b) 7 non-local. (c) 6 joint GCs for each joint (d) Candidate GCs. (e) Our decomposition. (f) Greedy search.

Figure 7: *Merge 143 local GCs (a) of a Gecko model into 7 non-local GCs (b) using the criterion defined in* (5)*. Note that the two X-shaped joints (dark gray) are not covered in both (a) and (b). To cover them, we set up six different ways of connecting adjacent GCs (c). A set of candidate GCs (d) are then computed using the 7 non-local GCs and 12 joint GCs to find the global optimal exact cover, i.e., our decomposition (e). In comparison, a greedy search would pick the first joint GC in (c) out of the 6 choices, resulting a suboptimal decomposition (f).*

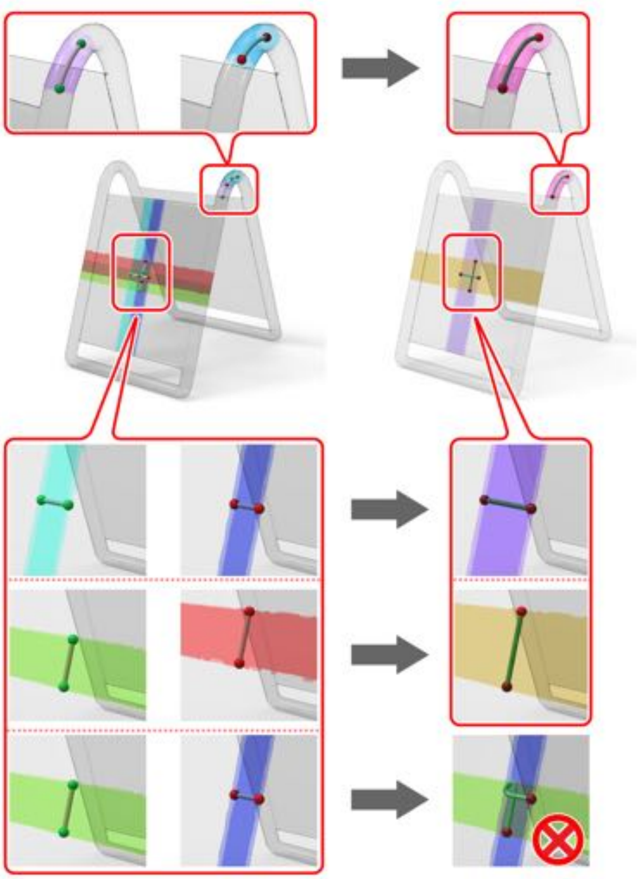


Figure 8: *Illustration of the local GC merging criteria for the shape shown in Figure 5.*

of adjacent non-local GCs and the joint GCs in between can form a single candidate GC; iii) a joint GC cannot be a candidate by itself since it needs to be attached to one or more adjacent non-local GCs.

With the candidate GC set generated, where each covers a portion of the shape S, we employ Knuths Algorithm X [Knuth 2000] to efficiently enumerate all solutions to ECP, while sorting the solutions according to their cylindricity values. Each solution corresponds to a decomposition for S and the best one is picked based on the above

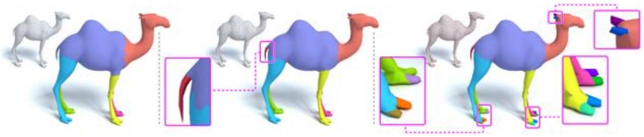


Figure 9: *Increasing,* $|\Omega|$*, the number of sample points on the input shape, allows finer-scale cylindrical structures to be extracted by the optimization algorithm. Left:* $|\Omega| = 100$*; middle:* $|\Omega| = 500$*; right:* $|\Omega| = 1,500$*.*

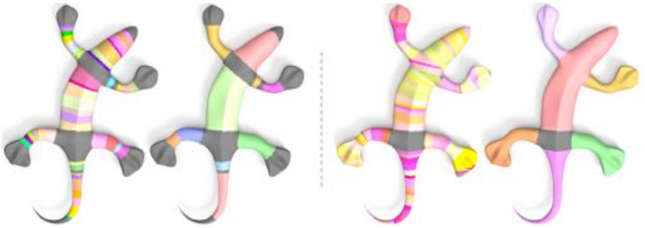


Figure 10: *Impact of parameter* τ*. Left: setting* $\tau = 0.12$ *leads to 118 local GCs, which are merged into 10 non-local GCs. Right: with* $\tau = 0.25$*, 196 local and 6 non-local GCs are formed. The results under* $\tau = 0.15$ *are shown in Figure 7. Although having different numbers of local GCs and non-local GCs, the final decompositions under all three* τ *values are the same.*

objective (6); see Figures 5(d-e) and Figures 7(d-e). In comparison, the decomposition result shown in Figure 7(f) demonstrates that when multiple overlapping GCs are covering the same region, selecting the GC with lowest cylindricity value in a local and greedy manner leads to a globally suboptimal decomposition.

5 Results

In this section, we show results of our decomposition algorithm, present applications, and evaluate and compare our approach to other alternatives. We mainly demonstrate advantages afforded by the global solutions we obtain, as well as robustness and utility offered by the cylindricity measure and the resulting decomposition.

5.1 Decomposition results and applications

Parameters. Our decomposition algorithm has four tunable parameters: the description length C, weight α between the two terms

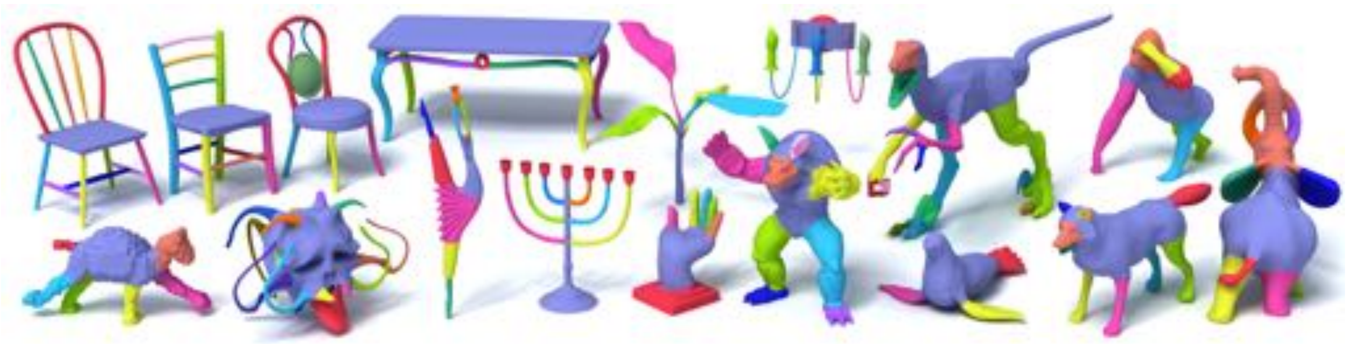
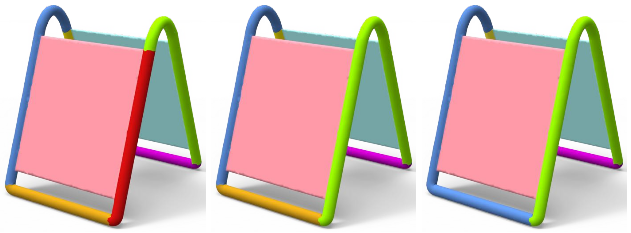
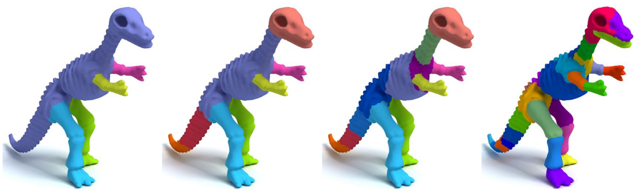


Figure 13: *A gallery of decomposition results obtained by our decomposition algorithm, all with the same parameter setting.*



(a) $C = 0.1$; 8 parts. (b) $C = 0.4$; 7 parts. (c) $C = 0.6$; 6 parts.

Figure 11: *Influence of the description length parameter C on decomposition, in particular, part counts, where α is fixed at 1. Note that with $C = 1$, we obtain three parts as shown in Figure 5(e).*



(a) $\alpha = 1$. (b) $\alpha = 6$. (c) $\alpha = 20$. (d) ACD.

Figure 12: *Influence of the weight α in cylindricity measure on decomposition results, while holding description length $C = 0.1$. Part counts for decomposition (a-c) are 5 , 8, and 12, respectively. As expected, approximate convex decomposition (ACD) produces substantially more parts (d).*

in the cylindricity measure, the size of the sample point set Ω on the input shape for initial local cylinder construction, and finally the cylindricity threshold τ for filtering out bad local cylinders. To remove the impact of object size on parameter selection, here all input shapes are normalized through uniform scaling, such that the longest diagonals of their bounding boxes equal to one. Altering the first two parameters can lead to different decomposition results, since they influence the cylindricity measure, which in turn, influence the objective function. By default, we set $C = 0.1$ and $\alpha = 1$. The next two parameters, $|\Omega|$ and τ, were chosen empirically and are fixed throughout our experiments at $|\Omega| = 500$ and $\tau = 0.15$, with only one exception discussed in Figure 9. This camel example demonstrates how the sample size $|\Omega|$ influences the scales of the extracted cylinders in the final decomposition: more samples typically lead to the identification of finer-scale structures.

Model	#lc	#nc	#cc	#G	Time
Chair (left)	220	31	151	12	9m
Chair (middle)	123	20	61	12	6m
Chair (right)	214	16	50	9	8m
Table	255	18	75	17	10m
Plant	133	15	43	5	6m
Lamp	172	24	175	11	8m
Raptor	193	30	120	16	8m
Gorilla	121	9	19	7	5m
Camel grid	49	8	10	7	4m
Sea monster	76	25	32	22	5m
Dancer	166	14	94	10	6m
Candlestick	214	18	175	14	8m
Hand	172	22	134	7	7m
Armadillo	134	27	99	10	6m
Seal	158	5	8	4	7m
Wolf	149	11	24	10	6m
Elephant	156	17	49	11	7m

Table 1: *Timing and other statistics for our algorithm, measured on the models shown in Figure 13. Number of local GCs, non-local GCs, candidate GCs, and GC part count for the optimal decomposition are denoted by #lc, #nc, #cc, and #G, respectively. Execute times, in minutes and measured on an Intel(R) Core(TM) i7-4790 3.6GHz with 16 GB RAM, account for all operations.*

Figure 10 shows the impact of parameter τ on the construction of local GCs. Generally speaking, a lower τ value leads to fewer but more reliable local GCs than a higher τ value. Since these local GCs may not fully cover the shape, they may yield more non-local GCs after the merge process. Regardless of how many local and non-local GCs were found, the final optimal decompositions obtained with different τ values are nearly identical, which suggests our approach is robust with respect to the value of this parameter.

Figure 11 shows different decomposition results by changing the description length parameter C. As can be observed, larger C values encourage the formation of long cylinders and this is reflected in the final decomposition results. Experimentally, decomposition results are relatively less sensitive to choices of the α parameter, as shown in Figure 12, where we also contrast with approximate convex decomposition [van Kaick et al. 2014].

Decomposition results. Figure 13 shows a gallery of decomposition results on a mixture of organic and man-made objects, all obtained using the default parameter setting. Table 1 provides timing and other statistics for our decomposition algorithm on these models. Computationally, the most expensive step is local GC con-

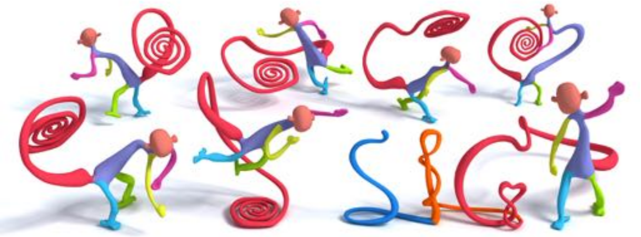


Figure 14: *A stress test: the 3D monkey model from the teaser was bent and stretched in a variety of ways to test the robustness of our cylindricity measure and decomposition algorithm.*

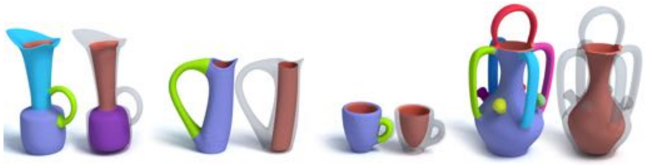


Figure 15: *Our decomposition returns both inner (right) and outer (left) GC surfaces for container-like objects with inner concavity.*

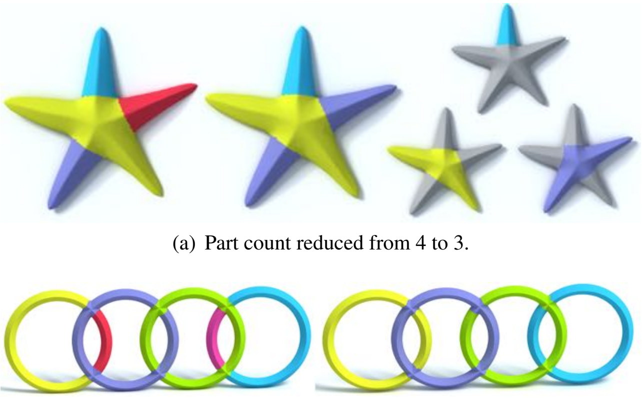
struction, though it is highly parallelizable.

Note the flat parts obtained on the chair, table, and plant models, as well as other parts, e.g, the skirt of the dancer, that are cylindrical only in a rather general sense. These parts and the corresponding skeletons are unlikely to emerge when running conventional skeletonization algorithms (as one may observe from Figure 17). On the other hand, it is evident that our decomposition is not designed to fully reflect shape semantics. For example, one finger of the dinosaur model is merged with the arm; this is not meaningful but nevertheless expected in the context of cylindrical decomposition. As well, our algorithm does not perform any boundary optimization, hence some part boundaries are unnatural or non-smooth.

In Figure 14, we show decomposition results through a “stress” (or “stretch”) test, where the 3D monkey model from Figure 1 was bent and stretched in a variety of ways. The results, obtained using the same default set of parameters, show that our cylindricity measure and cylindricity-driven decomposition scheme do exhibit robustness against bending and stretching of shape parts.

Inner+outer cylinders. As the shape processing in our decomposition is performed over the *surface* of the input model, not any enclosed volume, we do not require the volume enclosed by a surface region to belong to the interior. In fact, our algorithm does not even require the input shape to be watertight. This offers a rather unique feature to our decomposition: it would return both outer and inner GCs for container-like objects with an inner concavity; see results shown in Figure 15. Performing either surface or solid convex decompositions amid such concavities is expected to yield a large number of parts. As well, the sphere-mesh representation is not suited to handle such thin shell shapes; they represent what the authors [Thiery et al. 2013] referred to as a “worst-case” scenario.

Overlapping cover. While shape decomposition is the focus of this work, our algorithm can be easily altered to produce an overlapping cover where the obtained shape parts are allowed to overlap. Such results may be desirable if the cylindrical parts are *transient* in nature, meaning that it is more meaningful to allow them to cross each other rather than breaking them apart at the junctions. To obtain overlapping covers, only the last step needs a change from solving an ECP to solving an overlapping cover problem. Since the latter encompasses the former, from the same set of candidate GCs, the best overlapping cover according to our objective function is always at least as good as the best decomposition. We adopt the greedy algorithm [Chvatal 1979] to compute overlapping covers. Figure 16 shows that this effectively reduces part counts.



(a) Part count reduced from 4 to 3.

(b) Part count reduced from 6 (left) to 4 (right).

Figure 16: *Overlapping covers lead to reduction in part counts.*

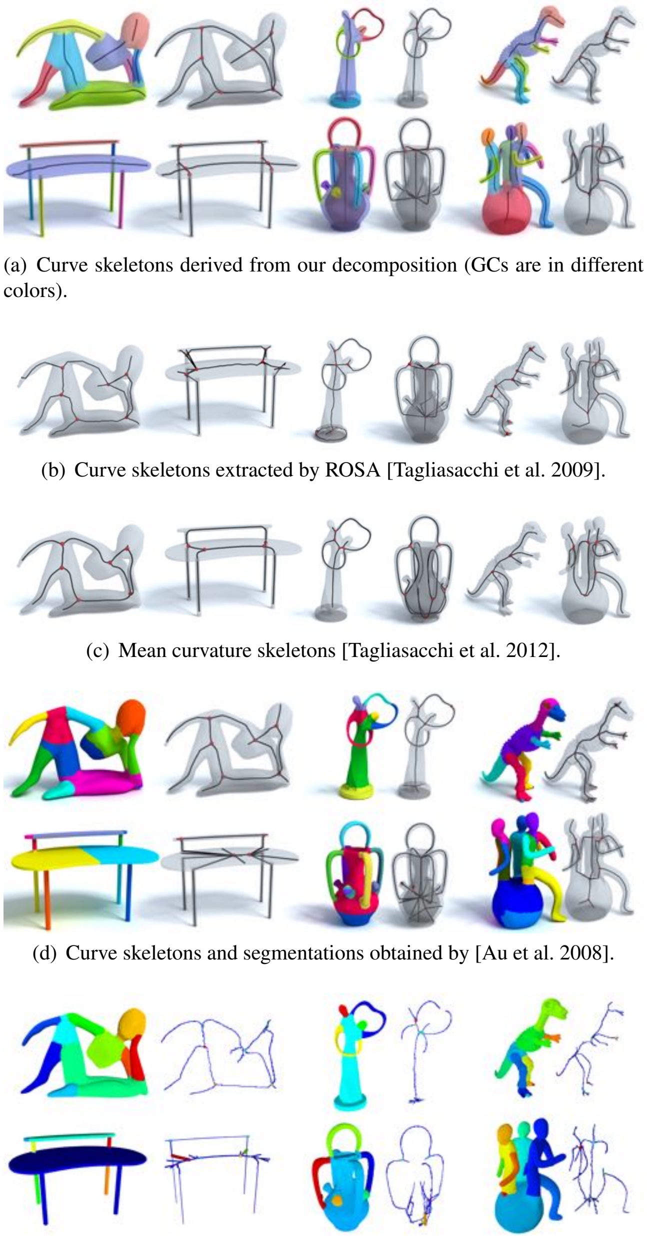
Curve skeleton extraction. While being a side product of computing the cylindricity measure for shape decomposition, the skeleton+profile representations obtained by our algorithm enable several applications. One of them is curve skeleton extraction. We can convert the axes of the set of cylinders obtained in the final decomposition into a curve skeleton representation of the input 3D shape. Specifically, we connect end points of the axes to the nearest points on the axes of adjacent GCs, and perform post-smoothing of the final set of skeletal curves. Figure 17(a) shows several results.

3D shape modeling. Curve skeletons are natural editing handles; and the skeleton+profile representations obtained also simplify the finding of shape correspondence. In the accompanying video, we show examples of shape modeling by manipulating the individual GCs through only curve controllers. This allows both deformation of the skeletal axes and editing of the profile curves. Another modeling paradigm, morphing, is also supported. The ability to obtain GCs with significant variability may allow dramatic morphing results such as the one shown in Figure 18. Correspondences between the GCs are obtained via graph matching with dissimilarity between GCs computed by a sum of distances between their cylindricity measures, axis lengths, and average profile radii.

Progressive shape approximation. The skeleton+profile representation, which is built via recursive geometric interpolation, naturally provides a *progressive approximation* of the input 3D shape. The first example is shown in Figure 22. Similar in spirit to sphere-meshes [Thiery et al. 2013], the kind of approximations obtained are based on interpolating simple primitives; in our case, profile curves are used for cylinder construction and control points are used for profile curve construction. With the MDL principle followed in our approach, our GC-based progressive approximation is expected to be efficient; we compare to sphere-meshes in Section 5.2.

5.2 Comparisons

Local merging vs. global decomposition. We conduct an experiment to compare purely local GC growing with global decom-



(a) Curve skeletons derived from our decomposition (GCs are in different colors).

(b) Curve skeletons extracted by ROSA [Tagliasacchi et al. 2009].

(c) Mean curvature skeletons [Tagliasacchi et al. 2012].

(d) Curve skeletons and segmentations obtained by [Au et al. 2008].

(e) Curve skeletons and segmentations obtained by Reniers et al. [2008].

Figure 17: *Comparing curve skeletonization and segmentation with several state-of-the-art methods.*

position. Exactly the same set of local GCs were used to initialize decomposition and a greedy merging scheme. For the latter, we iteratively and greedily select local GCs with minimal cylindricity values and merge them under the same criterion (5) as in our decomposition approach. Figure 19 (see also Figure 7) compares results obtained by the two approaches, which demonstrate the benefit of using a global optimization step that solves an ECP.

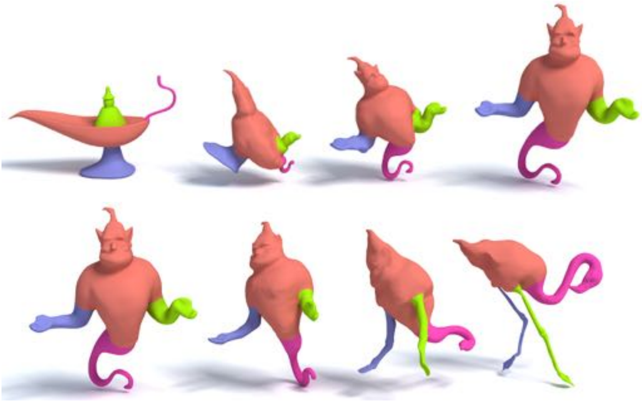


Figure 18: *Two dramatic morphing sequences that warp an Aladdin lamp to a genie then to a flamingo. After obtaining the decompositions for the source and target shapes, a cylindricity-based straightforward graph matching results in both part and skeleton correspondence, enabling morphing between corresponding profile curves.*

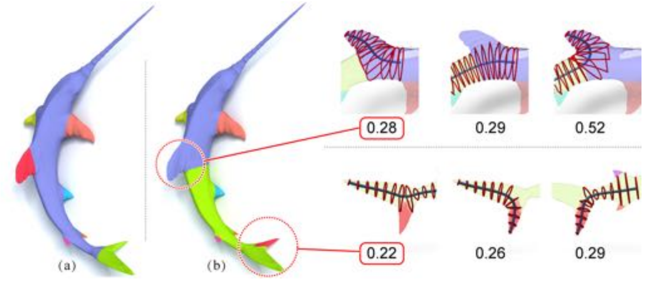


Figure 19: *Global decomposition vs. greedy local GC merging. Note that there are three different local GC merging directions respectively at the back-fin and the tail-fin of the shark. Making decisions in a local and greedy manner produces undesirable results: the body of shark is split into two parts, with each encompassing a fin (b). Our decomposition algorithm searches for the optimal solution for ECP, resulting in a more natural decomposition (a).*

Comparison on skeleton extraction. Figure 17 compares curve skeletons obtained from our decomposition results to several of the state-of-the-art methods, including [Tagliasacchi et al. 2009], [Tagliasacchi et al. 2012], [Au et al. 2008], and [Reniers et al. 2008]. As we can see, all methods perform comparably over close-to-perfect cylindrical regions. On flat parts, most methods, with the exception of mean curvature skeleton and ours, tend to produce noisier set of skeletons. Both [Au et al. 2008] and [Reniers et al. 2008] also provide means for skeleton-driven shape segmentation. Their results can be compared to ours in Figure 17. Note that none of the three methods attempt to model shape semantics; they are all purely geometry-driven. In general, [Au et al. 2008] tends to over-segment due to its reliance on local cut criteria. [Reniers et al. 2008] is multi-scale but controlling the scales is non-trivial; see results for the vase and dinosaur. With a global optimization, our solution appears to offer an attractive option.

Comparison to profile clustering. We compare our cover-based decomposition to a well-known clustering-based approach, namely, affinity propagation clustering, which was adopted by Goyal et al. [2012] in their GC analysis algorithm. Recall that affinity propagation clustering provides iterative refinement through local message passing. We report comparison results on models we created to closely resemble the models tested in their paper, since we were not able to obtain either their models or their implementations. The comparison shows that the global solution we obtain generally induces fewer parts and reflects better the structures of the shapes.

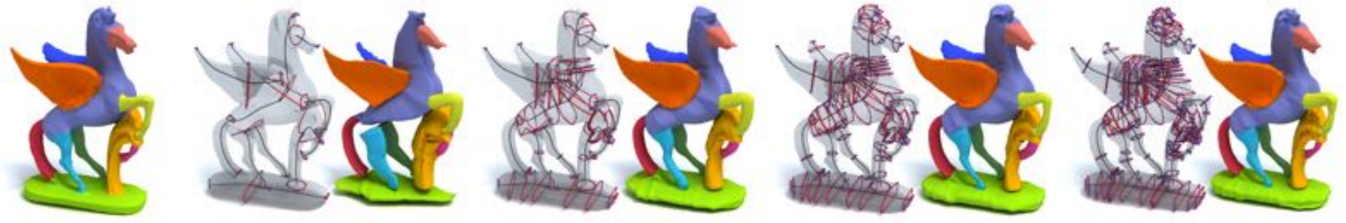
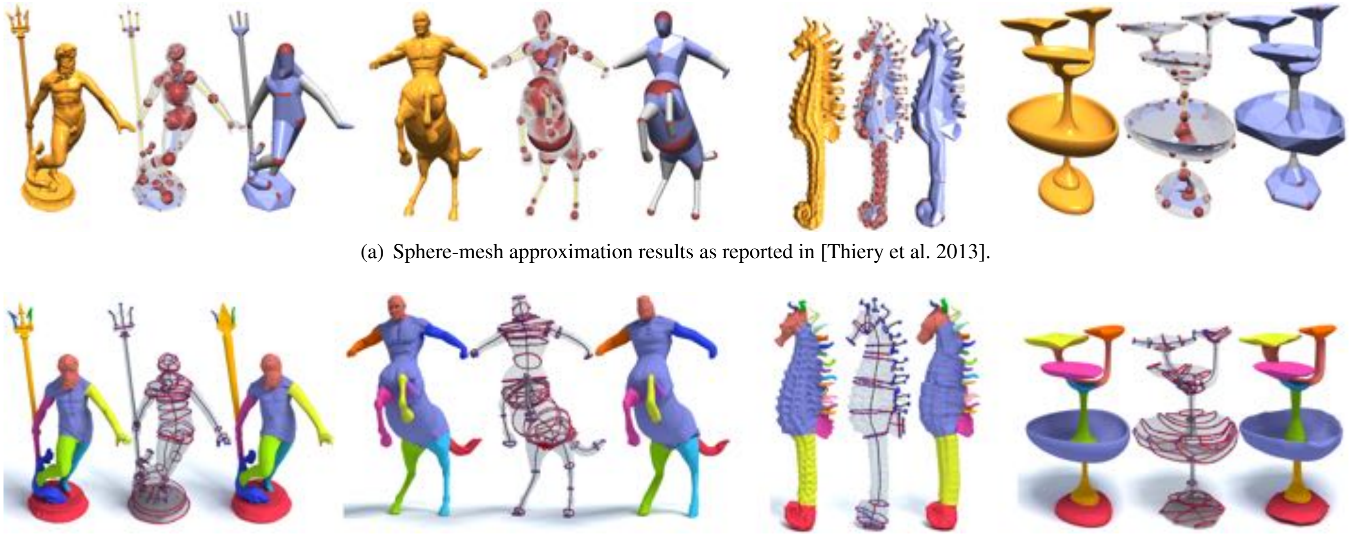


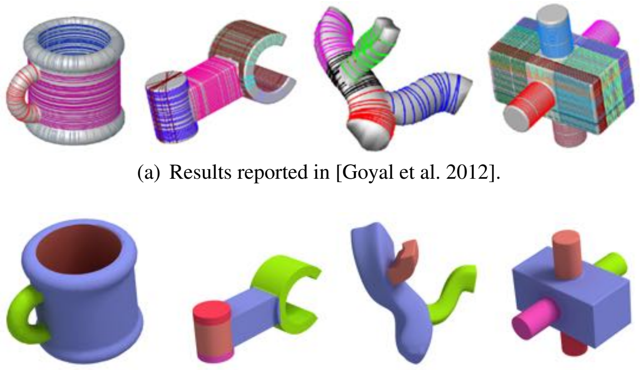
Figure 22: *Progressive shape approximation using generalized cylinders from our decomposition algorithm. The pegaso model from [Thiery et al. 2013] is decomposed (leftmost) into 11 parts. A set shape approximations are generated (from left to right) through progressively increasing the number of profile curves and control points (#P, #C) from (25, 100), (50, 300), (75, 600), to (100, 1000). The approximation errors (H, M12, M21), as defined in Table 2, decreases from (6.188, 0.813, 0.683), (2.139, 0.311, 0.32), (1.456, 0.192, 0.198), to (1.456, 0.151, 0.156).*



(a) Sphere-mesh approximation results as reported in [Thiery et al. 2013].

(b) Skeleton+profile approximations obtained from optimal decomposition on the same models.

Figure 23: *Comparison between shape approximations obtained by sphere-meshes (a) and our decomposition (b), where a comparable number of interpolated primitives are used. That is, based on the number of spheres used in sphere-meshes (neptune: 60; centaur: 50; sea horse: 200; vase: 120), we either set the same number of profile curves to be the same (neptune 60, centaur 50) or set the number of control points to be twice of the amount (sea horse 400, vase 240). Approximation errors for all models under both settings are reported in Table 2.*



(a) Results reported in [Goyal et al. 2012].

(b) Our decompositions on very similar models that we replicated.

Figure 20: *Comparison to profile clustering [Goyal et al. 2012]. On four models used in [Goyal et al. 2012], our decomposition produces 3, 5, 3, and 5 parts, respectively, whereas they obtained 4, 7, 5, and 14 parts, respectively.*

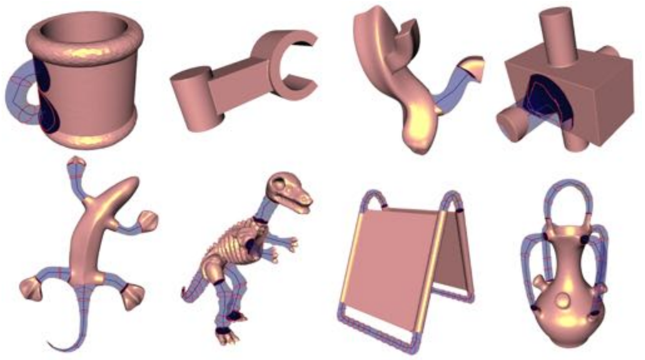
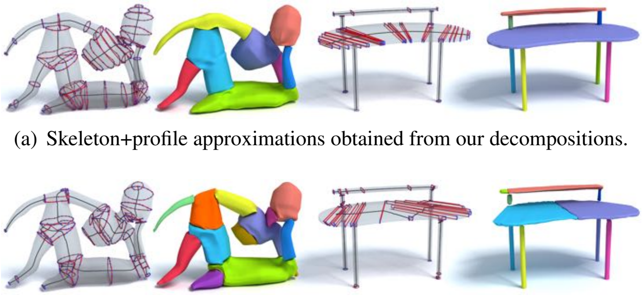


Figure 21: *Cylinder extraction by Plumber [Mortara et al. 2004] is able to extract close-to-perfect tubular parts (shown in transparent blue with embedded skeletons) but does not return a complete decomposition into* generalized *cylinders. Compare these results to ours shown in Figures 7(e), 11, and 17(a).*

INPUT	SPHERE-MESH				OUR APPROXIMATION I				OUR APPROXIMATION II			
MODEL	(#S / #E / #T)	H	M12	M21	(#G / #P / #C)	H	M12	M21	(#G / #P / #C)	H	M12	M21
Camel	50 / 19 / 34	7.201	0.465	**0.384**	7 / 50 / 500	3.453	0.17	0.326	7 / 25 / 100	**3.675**	**0.426**	0.571
Neptune	60 / 15 / 72	2.65	0.384	0.388	14 / 60 / 600	1.259	0.186	0.193	14 / 30 / 120	**2.625**	**0.319**	**0.372**
Centaur	50 / 17 / 42	3.557	**0.373**	**0.401**	10 / 50 / 500	1.234	0.172	0.173	10 / 25 /100	**2.779**	0.466	0.448
Chair	35 / 16 / 22	**1.517**	**0.313**	**0.345**	12 / 35 / 350	1.358	0.156	0.157	12 / 17 / 70	2.102	0.382	0.369
Dancer	54 / 11 / 67	**1.352**	**0.166**	**0.182**	10 / 54 / 540	0.939	0.096	0.096	10 / 27 / 108	1.615	0.294	0.274
Gorilla	36 / 5 / 37	6.109	**0.53**	**0.446**	7 / 36 / 360	2.251	0.285	0.387	7 / 18 / 72	**3.618**	0.595	0.699
Flamingo	40 / 12 / 32	2.837	0.417	0.479	4 / 40 / 400	2.454	0.112	0.126	4 / 20 / 80	**2.455**	**0.299**	**0.354**
Sea monster	150 / 60 / 161	6.081	0.367	0.364	22 / 150 / 1500	6.053	0.103	0.11	22 / 75 / 300	**5.642**	**0.307**	**0.315**
Elephant	45 / 6 / 68	**3.357**	0.564	0.62	11 / 45 / 450	1.722	0.262	0.303	11 / 22 / 90	3.521	**0.52**	**0.522**
Wolf	55 / 12 / 75	3.618	**0.38**	0.405	10 / 55 / 550	1.469	0.148	0.15	10 / 27 / 110	**2.799**	0.383	**0.365**
Fish	12 / 3 / 6	**4.272**	0.762	0.735	3 / 12 / 120	2.555	0.288	0.332	3 / 6 / 24	9.057	**0.739**	**0.624**
Sea horse	200 / 18 / 344	**1.324**	0.257	0.271	20 / 200 /2000	2.212	0.102	0.104	20 / 50 / 400	2.031	**0.184**	**0.181**
Raptor	45 / 23 / 30	4.31	0.428	0.425	17 / 45 / 450	2.682	0.179	0.185	17 / 22 / 90	**2.657**	**0.364**	**0.396**
Vase	120 / 8 / 197	2.093	0.321	0.395	9 / 120 / 1200	2.871	0.288	0.279	9 / 40 / 240	**1.487**	**0.222**	**0.217**
Camel grid	50 / 5 / 72	**4.199**	0.9	0.629	7 / 50 / 500	2.216	0.292	0.294	7 / 25 / 100	4.522	**0.563**	**0.494**
Half bear	60 / 5 / 66	14.784	0.4	2.559	7 / 60 / 600	1.297	0.147	0.31	7 / 30 / 120	**2.285**	**0.321**	**0.49**
Pegaso	100 / 11 / 170	2.554	0.381	0.393	11 / 100 / 1000	1.456	0.151	0.156	10 / 50 / 200	**2.542**	**0.374**	**0.362**

Table 2: *Quantitative comparison between sphere-meshes and skeleton+profile approximations resulting from our decompositions. We report the number of spheres ($\#S$), edges ($\#E$), and triangles ($\#T$) for sphere-meshes, and the number of GCs ($\#G$), profile curves ($\#P$), and total number of control points ($\#C$) for our approximation. For both methods, we report three approximation errors, as in [Thiery et al. 2013]: Hausdorff distance (H), mean distance from original model to approximation (M12), and mean distance from approximation to original (M21). All distances are expressed in percentages of the length of the input model's bounding box diagonal. Our approximation II uses only an half number of profile curves and one fifth number of control points of that our approximation I uses. We mark the smaller errors in bold when comparing sphere-meshes to our approximation II, with the same representation costs in their primitives (#S vs. #C). We consider that a 3D sphere (four scalars to represent) is equivalent to that of two control points (four scalars for 2D profile curves).*



(a) Skeleton+profile approximations obtained from our decompositions.

(b) Skeleton+profile approximations obtained from [Au et al. 2008].

Figure 24: *Shape approximations obtained by decompositions (a) and another skeleton+profile representation derived from [Au et al. 2008], where exact same numbers of profile curves (lady: 50; table: 40) and control points (lady: 500; table: 400) are used. The shape approximation errors (H, M12, M21) of our results are (1.241, 0.165, 0.167) for the lady model and (0.884, 0.073, 0.109) for the table model, compared with (1.98, 0.329, 0.337) and (6.823, 0.14, 0.165), respectively, obtained from [Au et al. 2008].*

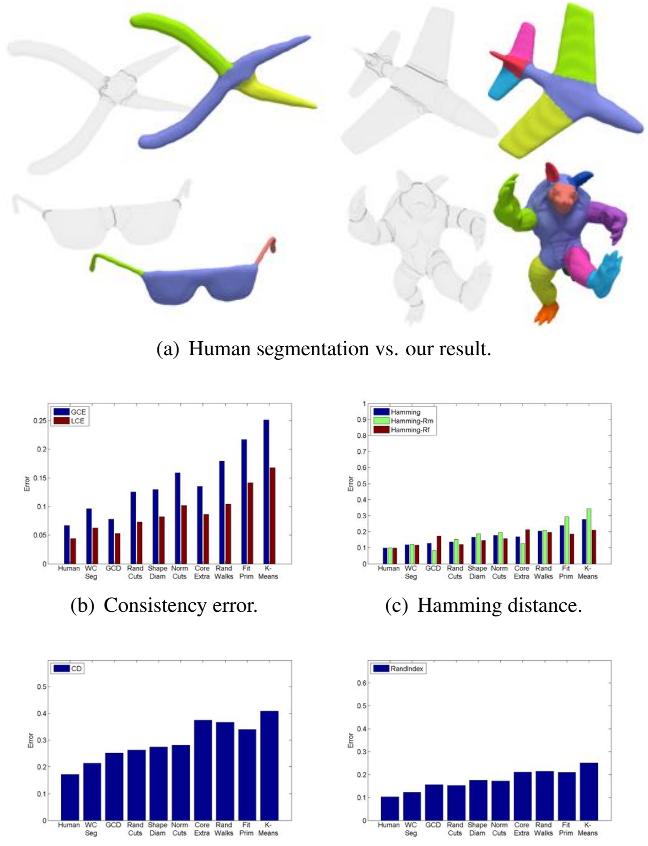
Comparison to primitive fitting. Figure 21 shows results from Plumber [Mortara et al. 2004], a primitive fitting technique aimed at recovering tubular parts from a 3D shape. Comparing these results to ours, it is evident that Plumber was not designed to process as general cylindrical shapes as those handled by our method.

Comparison to sphere-meshes. Finally, we make comparisons to sphere-meshes [Thiery et al. 2013] on shape approximation. Without access to their implementation, here we only tested our decomposition algorithm on models featured in their work, which were provided by the authors. We compare visually in Figure 23 and quantitatively in Table 2 between approximations using spheres as primitives (all numbers and figures copied from [Thiery et al. 2013]) and approximations provided by the skeleton+profile representations from our decomposition.

To evaluate the two methods in comparable settings, for each input shape, we compare approximations based on comparable number of interpolated primitives, which are spheres for sphere-meshes and profile curves or control points in our approach. Spheres and profile curves play similar roles as they are interpolated to obtain the final approximations directly. On the other hand, they are not as comparable in terms of representation costs. Considering the representation cost of a 3D sphere (four scalars to represent) is equivalent to that of two control points (four scalars for 2D profile curves), we further compare the two approximation approaches under comparable representation costs; see Figure 23 and Table 2. The results suggest that our approximations are generally superior to sphere-meshes in terms of approximation error and visual quality.

Aside from approximation quality, we believe that the utilization of skeletal and profile curves, along with the non-linear interpolations over these curves, offer a more versatile and more flexible shape modeling and manipulation paradigm compared to linear interpolations over spheres. For example, sphere-mesh representations are not particularly apt at modeling thin-shell like shapes or twisty shapes with sharp features.

Comparison on skeleton+profile approximation. In addition to comparing our skeleton+profile approximations with sphere-meshes, we also compare them to approximation results generated using an alternative skeletonization method [Au et al. 2008]; see Figure 24. Here, we apply the same profile curve selection approach used in our algorithm, but based on the skeletons extracted by Au et al. [2008]. The comparison suggests that, since our decomposition outputs fewer parts and more coherent skeletons, we obtain better approximations of the shape, under the same number of profile curves and control points, than the ones using skeletons and the skeleton-driven segmentation from [Au et al. 2008].



(a) Human segmentation vs. our result.

(b) Consistency error.

(c) Hamming distance.

(d) Cut discrepancy.

(e) Rand index.

Figure 25: *Comparison with human segmentation on Princeton Segmentation Benchmark [Chen et al. 2009].*

Comparison to human segmentation. The goal of our generalized cylinder decomposition is to maximize geometric simplicity. As a result, the decomposed parts may not agree with semantic segments labeled by humans. As shown in Figures 25(a) and (c), our decompositions generally contain fewer parts, as it often merges different segments along the same skeleton, such as forearm and hand, into one GC. Since these semantic segments labeled by humans are generally convex, the approximate convex decomposition [van Kaick et al. 2014] is more suitable for decomposing shapes semantically.

Nevertheless, we here quantitatively evaluate our approach based on human segmentation for the well-known Princeton Segmentation Benchmark. Figures 25(b-d) show the plots of four evaluation metrics on the benchmark, which reveal the corresponding properties of our method in different aspects, including good segment consistency (b), fewer decomposition parts (c), and similar (yet not very close) cut boundaries (d) to human segmentation. In Figure 25(e), the Rand index provides an overall likelihood assessment on state-of-the-art methods comparing to human segmentation. Clearly, the Rand index of our method is quite low and only outperformed by approximate convex decomposition (WCSeg) and Rand Cuts [Golovinskiy and Funkhouser 2008].

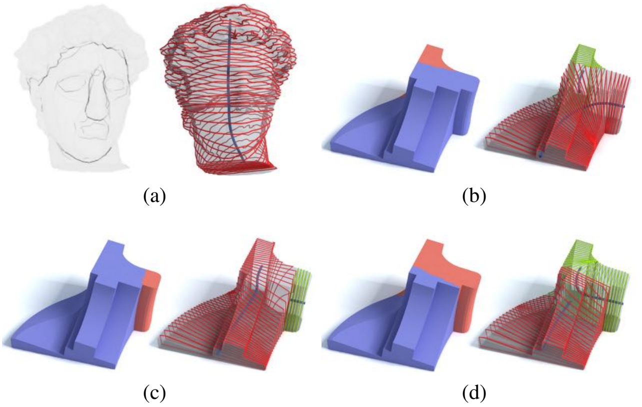


Figure 26: *For the Bust model in (a), our method (right) outputs a single GC, whereas humans may choose to separate semantically meaningful areas, such as nose or hair, as individual parts. For the Fandisk model, (b), (c), and (d) show the three solutions with the lowest costs in the final ECP, none of which are meaningful from the engineering perspective.*

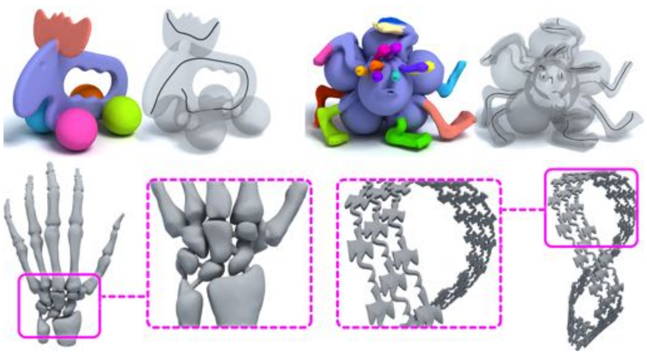


Figure 27: *Four failure cases on models from [Thiery et al. 2013]. Top: no local GCs are found for the close-to-perfect spherical parts in these two models. Bottom: the complex and delicate structures lead to too many GCs, making the final ECP unresolvable.*

6 Discussion, limitation, and future work

We have presented a method for decomposing a 3D shape into a small number of generalized cylinders (GCs), making the first attempt to compute a globally optimal solution. Furthermore, the objective function for the optimization is defined by a precise measure of cylindricity, the first of its kind. Our global approach is based on the simple observation that every non-overlapping decomposition or segmentation problem is an Exact Cover Problem (ECP). Thus, we reduce the decomposition task to an ECP and employ Knuth's Algorithm X to efficiently enumerate many exact covers. To alleviate the high cost of a large enumeration, we generate only a modest number of non-local cylinders as candidates for the optimal cover.

We have developed the cylindricity measure to estimate the geometric simplicity of a given GC in MDL terms. We have shown that the measure is robust, independent of the sampling rate, and scale-invariant. The cylindricity measure plays a double role in our work: first, it encourages the growth of local GCs into longer ones; second, it is central to our global objective function.

Limitations. Our current algorithm has several technical limitations. First, the construction of a local GC is not sampling-independent, as we apply plane detection by ROSA only at a finite number of sample points. Our current implementation uses a uniform sampling, not adaptive to features, hence it may cause some local GC orientations or small features to be missed when sampling

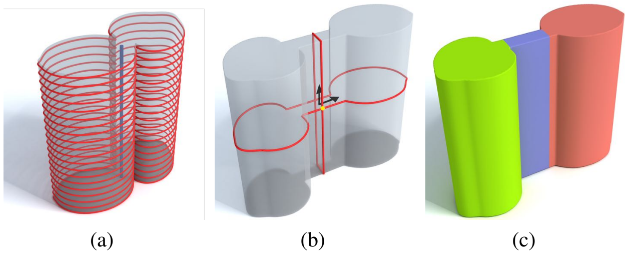


Figure 28: *The profile curve of a GC does not need to be convex and hence the decomposition does not split the shape in (a) into two convex cylinders. However, when the shape contains parts that has ambiguous GC orientations (b), splitting is performed to separate the parts with ambiguity from the ones without (c).*

rate is not sufficiently high; the latter situation is shown in Figure 9. In these cases, the small protrusions may be merged into local GCs; see, for example, the fingers of the armadillo model merged with its arm in our gallery Figure 13.

Figure 26 shows two non-cylindrical shapes, for which our method fails to generate meaningful decompositions from the semantic or engineering perspectives. Nevertheless, the skeletons found by our approach can be still useful for other applications.

Figure 27 shows four models from the sphere-mesh work [Thiery et al. 2013] which our decomposition failed to process properly. For models containing close-to-perfect spherical parts (top of Figure 27), any axis passing the sphere centers could be a rotational symmetry axis, hence no consistent local axis could be formed for a local GC. In our implementation, these spherical structures are segmented and treated as parts in the final decomposition. Also, if the models contain highly complex and delicate structures (bottom of Figure 27), our current scheme may have trouble merging a large set of small yet non-coherent local cylinders. As a result, the final exact cover problem may become intractable.

Aiming at decomposing shapes into *generalized* (rather than *circular*) cylinders, our approach does not require the profile curve of a GC to be circular or even convex. For example, the shape shown in Figure 28(a) is considered as a perfect GC and our approach does not attempt to split it into two convex parts. However, splitting will occur when the input shape contains parts that has ambiguous GC orientations; see Figures 28(b) and (c).

Future work. Conceptually, our decomposition approach does not directly consider any shape semantics or perform any boundary optimization (see results in Figure 13 for examples). Our problem belongs to the type of decompositions which seek geometrically simple parts, such as convex [Lien and Amato 2004], monotone, or pyramidal [Hu et al. 2014] decompositions. Semantic segmentation, in general, follows different criteria. Nevertheless, one can consider modifying the cylindricity measure to discourage growing GCs over local minima, following the spirit of the minima rule [Hoffman and Richards 1984]. We leave this for future work.

We would also like to look into consistent segmentation through the generalized cylinder decomposition. The key question is what other weak knowledge, beyond fixing a part count over the input set of shapes, can be beneficial. Last but not the least, we believe that the power of the ECP needs to be explored for other decomposition or segmentation problems, whose solutions are typically built only on local measures without adhering to any global objective function. ECP solves a global problem and leads to globally optimal solutions. This, however, requires developing quantitative measures for the parts or segments involved in the partition, which is often ill-posed. In the future, we would explore this avenue further for various decomposition and primitive fitting problems.

Acknowledgments

We thank the reviewers for their valuable feedback. This work was supported in part by NSFC (61522213, 61232011, 61528208, 61379090), 973 Program (2014CB360503), 863 Program (2015AA016401), Guangdong Science and Technology Program (2015A030312015, 2014B050502009, 2014TX01X033), Shenzhen VisuCA Key Lab (CXB201104220029A), SIAT Innovation Program for Excellent Young Researchers (201305), NSERC (611370 and 293127) and the Israel Science Foundation.

References

ASAFI, S., GOREN, A., AND COHEN-OR, D. 2013. Weak convex decomposition by lines-of-sight. *Computer Graphics Forum (Proc. of SGP) 32*, 5, 23–31.

ATTENE, M., FALCIDIENO, B., AND SPAGNUOLO, M. 2006. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer 22*, 3, 181–193.

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 Graphics (Proc. of SIGGRAPH) 27*, 3, 44:1–44:10.

BAERENTZEN, J. A., ABDRASHITOV, R., AND SINGH, K. 2014. Interactive shape modeling using a skeleton-mesh co-representation. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 33*, 4, 132:1–132:10.

BIASOTTI, S., GIORGI, D., SPAGNUOLO, M., AND FALCIDIENO, B. 2008. Reeb graphs for shape analysis and applications. *Theoretical Computer Science 392*, 1-3, 5–22.

CHAZELLE, B. 1987. Approximation and decomposition of shapes. In *Advances in Robotics 1: Algorithmic and Geometric Aspects of Robotics*, Lawrence Erlbaum Associates, 145–185.

CHEN, X., GOLOVINSKIY, A., AND FUNKHOUSER, T. 2009. A benchmark for 3D mesh segmentation. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 28*, 3, 73:1–73:12.

CHEN, T., ZHU, Z., SHAMIR, A., HU, S.-M., AND COHEN-OR, D. 2013. 3-sweep: extracting editable objects from a single photo. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia) 32*, 6, 195:1–195:10.

CHVATAL, V. 1979. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research 4*, 3, 233–235.

COHEN-STEINER, D., ALLIEZ, P., AND DESBRUN, M. 2004. Variational shape approximation. In *Proc. of SIGGRAPH*, ACM, SIGGRAPH '04, 905–914.

DOUGLAS, D. H., AND PEUCKER, T. K. 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Int. J. Geographic Information and Geovisualization 10*, 2, 112–122.

GARLAND, M., AND HECKBERT, P. S. 1997. Surface simplification using quadric error metrics. In *Proc. of SIGGRAPH*, 209–216.

GOLOVINSKIY, A., AND FUNKHOUSER, T. 2008. Randomized cuts for 3D mesh analysis. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia) 27*, 5, 145:1–145:12.

GOYAL, M., MURUGAPPAN, S., PIYA, C., BENJAMIN, W., FANG, Y., LIU, M., AND RAMANI, K. 2012. Towards locally and globally shape-aware reverse 3d modeling. *Computer-Aided Design 44*, 6, 537–553.

GRUNWALD, P. D. 2007. *The Minimum Description Length Principle*. Adaptive Computation and Machine Learning series. The MIT press.

HOFFMAN, D. D., AND RICHARDS, W. A. 1984. Parts of recognition. *Cognition*, 65–96.

HU, R., LI, H., ZHANG, H., AND COHEN-OR, D. 2014. Approximate pyramidal shape decomposition. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia) 33*, 6, 213:1–213:12.

JACOBSON, A., DENG, Z., KAVAN, L., AND LEWIS, J. P. 2014. Skinning: Real-time shape deformation. In *ACM SIGGRAPH Courses*.

JALBA, A., SOBIECKI, A., AND TELEA, A. 2015. An unified multiscale framework for planar, surface, and curve skeletonization. *IEEE Trans. Pattern Analysis & Machine Intelligence*.

KALOGERAKIS, E., HERTZMANN, A., AND SINGH, K. 2010. Learning 3d mesh segmentation and labeling. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 29*, 4, 102:1–102:12.

KATZ, S., AND TAL, A. 2003. Hierarchical mesh decomposition using fuzzy clustering and cuts. In *Proc. of SIGGRAPH*, ACM, SIGGRAPH '03, 954–961.

KNUTH, D. 2000. Dancing links. *Millenial Perspectives in Computer Science*, 159–187.

LI, X., TOON, T., TAN, T., AND HUANG, Z. 2001. Decomposing polygon meshes for interactive applications. In *Proc. of Symp. on Interactive 3D Graphics*, 35–42.

LI, G., LIU, L., ZHENG, H., AND MITRA, N. J. 2010. Analysis, reconstruction and manipulation using arterial snakes. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia) 29*, 6, 152:1–152:10.

LI, Y., WU, X., CHRYSATHOU, Y., SHARF, A., COHEN-OR, D., AND MITRA, N. J. 2011. Globfit: Consistently fitting primitives by discovering global relations. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 30*, 4, 52:1–52:12.

LIEN, J. M., AND AMATO, N. M. 2004. Approximate convex decomposition of polygons. In *Proc. of ACM Symp. on Computat. Geom.*, 17–26.

MANGAN, A. P., AND WHITAKER, R. T. 1999. Partitioning 3D surface meshes using watershed segmentation. *IEEE Trans. Visualization & Computer Graphics 5*, 4, 308–321.

MIKLOS, B., GIESEN, J., AND PAULY, M. 2010. Discrete scale axis representations for 3d geometry. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 29*, 4, 101:1–101:10.

MORTARA, M., PATANÉ, G., SPAGNUOLO, M., FALCIDIENO, B., AND ROSSIGNAC, J. 2004. Plumber: a method for a multi-scale decomposition of 3d shapes into tubular primitives and bodies. In *Proc. of Shape Modeling International*, 339–344.

RAAB, R., GOTSMAN, C., AND SHEFFER, A. 2004. Virtual woodwork: Making toys from geometric models. *International Journal of Shape Modeling 10*, 1, 1–30.

REN, Z., YUAN, J., LI, C., AND LIU, W. 2011. Minimum near-convex decomposition for robust shape representation. In *Proc. Int. Conf. on Computer Vision*, 303–310.

RENIERS, D., VAN WIJK, J. J., AND TELEA, A. 2008. Computing multiscale curve and surface skeletons of genus-0 shapes using a global importance measure. *IEEE Trans. Visualization & Computer Graphics 14*, 2, 355–368.

SANDER, P. V., WOOD, Z. J., GORTLER, S. J., SNYDER, J., AND HOPPE, H. 2003. Multi-chart geometry images. In *Proc. Eurographics Symp. on Geometry Processing*, 146–155.

SHAMIR, A., SHAPIRA, L., AND COHEN-OR, D. 2006. Mesh analysis using geodesic mean-shift. *The Visual Computer 22*, 2, 99–108.

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

SHAPIRA, L., SHAMIR, A., AND COHEN-OR, D. 2008. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer 24*, 4, 249–259.

SIMARI, P. D., AND SINGH, K. 2005. Extraction and remeshing of ellipsoidal representations from mesh data. In *Proc. of Graphics Interface*, 161–168.

SOLOMON, J., BEN-CHEN, M., BUTSCHER, A., AND GUIBAS, L. 2011. Discovery of intrinsic primitives on triangle meshes. *Computer Graphics Forum (Proc. of Eurographics) 30*, 2, 365–374.

TAGLIASACCHI, A., ZHANG, H., AND COHEN-OR, D. 2009. Curve skeleton extraction from incomplete point cloud. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 28*, 3, 71:1–71:9.

TAGLIASACCHI, A., ALHASHIM, I., OLSON, M., AND ZHANG, H. 2012. Mean curvature skeletons. *Computer Graphics Forum (Proc. of SGP) 31*, 5, 1735–1744.

THIERY, J.-M., GUY, E., AND BOUBEKEUR, T. 2013. Sphere-meshes: Shape approximation using spherical quadric error metrics. *ACM Trans. on Graphics (Proc. of SIGGRAPH) 32*, 6, 178:1–178:12.

VAN KAICK, O., FISH, N., KLEIMAN, Y., ASAFI, S., AND COHEN-OR, D. 2014. Shape segmentation by approximate convexity analysis. *ACM Trans. on Graphics 34*, 1, 4:1–4:11.

VAZIRANI, V. V. 2001. *Approximation Algorithms*. Springer.

YIN, K., HUANG, H., ZHANG, H., GONG, M., COHEN-OR, D., AND CHEN, B. 2014. Morfit: Interactive surface reconstruction from incomplete point clouds with curve-driven topology and geometry control. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia) 33*, 6, 41:1–41:12.

ZHENG, Y., FU, H., COHEN-OR, D., AU, O. K.-C., AND TAI, C.-L. 2011. Component-wise controllers for structure-preserving shape manipulation. *Computer Graphics Forum (Proc. of Eurographics) 30*, 2, 563–572.

ZUNIC, J., AND ROSIN, P. L. 2004. A new convexity measure for polygons. *IEEE Trans. Pattern Analysis & Machine Intelligence 26*, 7, 923–934.

ZUNIC, J., HIROTA, K., AND ROSIN, P. L. 2010. A Hu moment invariant as a shape circularity measure. *Pattern Recognition 43*, 1, 47–57.