

# Searching geometry-aware pants decomposition in different isotopy classes

KANG ZHANG AND XIN LI\*

We propose an optimization framework to compute the desirable pants decomposition of surfaces. A pants decomposition partitions a surface into a set of genus-0 sub-patches with 3 boundaries. Any surface with non-trivial topology admits infinitely many pants decompositions that are isotopically inequivalent. We traverse different classes of pants decompositions to search for the optimal one with the pre-determined geometric criterion. Our proposed framework is general, and can be used to construct different suitable segmentations according to different applications. We also generalize this algorithm for consistent decomposition of multiple surfaces, which can be used to construct compatible cross-surface parameterization.

## 1. Introduction

Given a surface with complicated geometry and topology, the surface segmentation partitions it into a group of simple sub-patches. It is widely used in many divide-and-conquer methods in geometric modeling. For multiple surfaces, a *consistent decomposition* is used to segment them into sets of sub-patches with the coherent adjacency relationship, namely, their dual graphs are isomorphic. It has many important applications, including cross-surface parameterization, morphing, animation transfer/synthesis, and many others [7, 15, 12, 11, 8, 20].

Existing algorithms segment the surface by local geometry regardless of the topology of the surface itself and its resultant decomposition. They are mostly used in single surface segmentation. However, the topological consistency of the segmentation is critical when processing multiple models in a unified divide-and-conquer scheme. Hence, existing single model segmentation methods often can not be easily generalized to consistently segment multiple surfaces. For the consistent segmentation, one popular scheme is to partition multiple surfaces into sets of topological disks [8, 7, 15]; the cutting boundaries of these disks are shortest paths connecting pre-defined marker

---

\*Corresponding author.

(feature) points. However, this approach requires the same number of feature points to be coherently placed on every model. For high-genus surfaces, quite a few amount of markers are necessary to ensure the correctness of the path tracing. Corresponding markers often need to be labeled manually and carefully, making the computation laborious and potentially unreliable. An automatic algorithm for high-genus surfaces with guaranteed segmentation consistency is hence desirable.

### 1.1. Consistent segmentation

Consistent segmentation of multiple objects has been intensively studied in computer graphics literatures. It is desirable to partition multiple objects into similar salient parts. For example, part analogies [16] segments each model into parts independently and then creates a distance measure by both local shape signatures and the context of the parts within a hierarchical decomposition. A consistent segmentation is then created based on this catalog of parts with inter-part distances. Golovinskiy and Funkhouser [4] simultaneously segment models and create correspondences between segments, so that the salient segments that are shared across the set of objects can be better identified. Kraevoy et al. [18] develop a modeling tool that consistently partition two meshes into sub-patches. These sub-patches are one-to-one corresponded and hence can be transplanted from one model onto the other. Yu et al. [24] compute consistent partitioning of shapes based on compatible skeletonization and star decomposition. Unlike [4] that allows outlier segments, this algorithm rigorously partitions all models consistently. This *topological coherency* is necessary for applications like cross-surface mapping, where outliers should not be allowed.

### 1.2. Pants decomposition for consistent segmentation

In this section we will introduce the definition for the *pants decomposition*, which is a powerful topological tool for automatic consistent segmentation.

**Definitions.** A *pants decomposition* (PD) is a set of curve cycles that partitions a surface into disjoint *pants patches*. Each *pants patch* is a topological sphere with three boundaries (see Fig. 1 for examples).

Two curves  $c_1$  and  $c_2$  on a surface  $M$  are *homotopic* to each other, if one can continuously evolve to the other through a family of paths on the surface. If a cycle is homotopic (i.e. can continuously contract) to a point, it is a *trivial cycle*, otherwise, it is a *non-trivial cycle* (cycles in Fig. 1(b)).

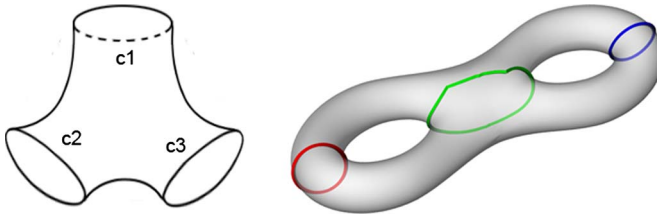


Figure 1: A Maximal Cut System (Pants Decomposition) for Two-torus. Left: a pants patch; Right: a maximal cut system (please check the electronic version of the paper for color figures).

A cycle is simple, if it is not self-intersected. Two cycles are disjoint if the intersection of them is empty.

A surface patch  $M$  is genus- $g$  if  $M$  can remain connected with at most  $g$  simple non-trivial cycles removed. A genus- $g$  surface patch with  $b$  boundaries is denoted as the type  $(g, b)$ . A *maximal cut system* of  $M$  is a set  $C$  of simple and pairwise disjoint cycles  $\{c_i\}$  that partitions  $M$  into sub-patches of type  $(0, 3)$ . Fig. 1(b) illustrates a maximal cut system computed on a two-torus model. Each sub-patch of type  $(0, 3)$  is called a *pants patch*. We also call this  $C$  a *pants decomposition* of surface  $M$ .

Any compact orientable surface, except for four trivial cases: sphere ( $b = g = 0$ ), disk ( $b = 1, g = 0$ ), torus ( $b = 0, g = 1$ ), and cylinder ( $b = 1, g = 1$ ), admits a pants decomposition [6]. If  $M$  is of genus  $g$  and has  $b$  boundaries, a pants decomposition (PD)  $C$  of  $M$  is made of  $|C| = 3g + b - 3$  cycles  $\{c_i, i = 1, \dots, |C|\}$ , and partitions  $M$  into  $2g + b - 2$  pants patches  $\{M_i, i = 1, \dots, 2g + b - 2\}$  [10].

*Topological classification of PD* A surface  $M$  admits infinitely many pants decompositions. We say two pants decompositions  $C = \{c_i\}$  and  $\hat{C} = \{\hat{c}_i\}$  are topologically equivalent, or homotopic, to each other if  $|C| = |\hat{C}|$  and there is a permutation  $\tau : \{1, \dots, |C|\} \rightarrow \{1, \dots, |\hat{C}|\}$  that  $c_i \sim \hat{c}_{\tau(i)}$  for  $\forall 1 \leq i \leq |C|$ . Intuitively, two pants decompositions are homotopic if we can continuously evolve one set of cycles on  $M$  to the other. The topological class of pants decompositions of  $M$  is that any two pants decompositions are topologically equivalent to each other. In this paper, our goal is to develop an algorithm to traverse among different topological equivalence classes to search for the optimal pants decomposition with the given geometric criteria.

*Existing work in PD* Pants decomposition algorithm is widely used in many geometric modeling applications. For example, Wang et al. [22] uses pants

decomposition to construct the T-spline. Other works including [23] and [13] also generate the volumetric T-splines by applying the pants decomposition algorithm.

Surfaces with nontrivial topology can be studied by its *homotopy types* of maps. In [12], canonical homology bases [5] and systems of cycles [3] are used to study this issue and build maps of different homotopy types. Dey et al. [1] define the terms *handle cycles* and *tunnel cycles*, which provide another intuitive way to study the topological handles on surfaces; they also introduced an efficient computation algorithm to compute tunnel and handle cycles [2]. Li et al. [10, 11] took the handles and tunnel cycles computed by [2] as inputs to compute canonical pants decomposition. Hatcher et al. [6] studied the enumerations of pants decomposition of different homotopy types. Computation of pants decompositions with shortest cutting locus has also been studied [6, 19]. Zhang et al. [25] first studied the optimization of the pants decomposition under certain geometric criteria.

A consistent pants decomposition algorithm for multiple surfaces was proposed in [11]. Due to the topological regularity of each sub-patch (genus-0, 3 boundaries), pants decomposition can be conducted on multiple high-genus surfaces for consistent decomposition. The algorithm proposed in [11] removes handles (type-(1, 1) patch) of the given genus- $g$  surface  $M$  iteratively and produces a type-(0,  $g$ ) remaining patch  $M'$ , then it further partitions  $M'$  into many type-(0, 3) pants patches iteratively. However, such a pants decomposition computed following a sequence of predetermined topological operations, may generate less geometrically-desirable partitioning on some models due to two main reasons:

- Topologically, the pants decomposition obtained via iteratively removing handles may not be geometrically natural and can lead to undesirable partitioning for some shapes (see Fig. 4(b) for an example).
- Geometrically, tracing the segmentation boundary only using shortest paths is often not the most suitable criterion for many shapes.

### 1.3. Main contribution

We develop an algorithm to traverse among different topological equivalence classes to search for the optimal pants decomposition. This topological optimization framework is ubiquitous for different applications preferring different geometric criteria. Guided by different criteria, this framework can provide locally optimal pants decomposition results. We also generalize this method to consistent pants decomposition for multiple surfaces and discuss its applications on inter-surface mapping and morphing. The three specific contributions of work are:

1. We develop a computation framework to traverse among the different topologically equivalence classes of the pants decompositions to search for the optimal pants decomposition under given geometric criteria.
2. We evaluate and integrate several useful geometric criteria in our decomposition framework.
3. Based on the above tool, we design a reliable consistent pants decomposition framework. When  $2g + b$  cycles are identified, a unique, locally-optimal, consistent set of pants decompositions across multiple objects can be obtained. We explore its applications on cross-surface mapping and morphing.

The remainder of this paper is organized as follows. Section 2 introduces the computation of PD in different homotopy classes; Section 3 integrates different geometric criteria into the optimization of PD. We discuss consistent PD computation across multiple surfaces in Section 4, show applications and experimental results in Section 6, and conclude the paper in Section 7.

## 2. Computing and enumerating PD

### 2.1. Pants decomposition graph

To help analyze homotopy classes of pants decomposition, we introduce the following *Pants Decomposition Graph* (PD-graph)  $G = (V, E)$ .

The node  $v_i \in V$  denotes the pants patch  $p_i$ .  $p_i$  and  $p_j$  are *adjacent* if two pants patches  $p_i$  and  $p_j$  share one or more boundaries and  $v_i$  and  $v_j$  are connected by the edge  $e_{ij} \in E$  for each pair of adjacent pants patches  $p_i$  and  $p_j$ . Each edge  $e_{ij}$  also corresponds to a cycle shared by two adjacent pants patches  $p_i$  and  $p_j$ . With the help of the PD-graph, we can enumerate the topologically inequivalent pants decompositions which will be elaborated in the next section. Fig. 2 shows one possible PD and its corresponding PD-graph on the genus-4 Greek model. Note that a node can be adjacent to itself if two of its boundaries share a common cycle. By gluing this cycle we will get a  $(1, 1)$ -typed patch.

### 2.2. Computing initial pants decomposition

In the first step, we will compute an initial pants decomposition on a given surface  $M$ . We can apply the existing pants decomposition method such as [11]. However, we develop a more efficient and reliable algorithm to compute the initial pants decomposition.

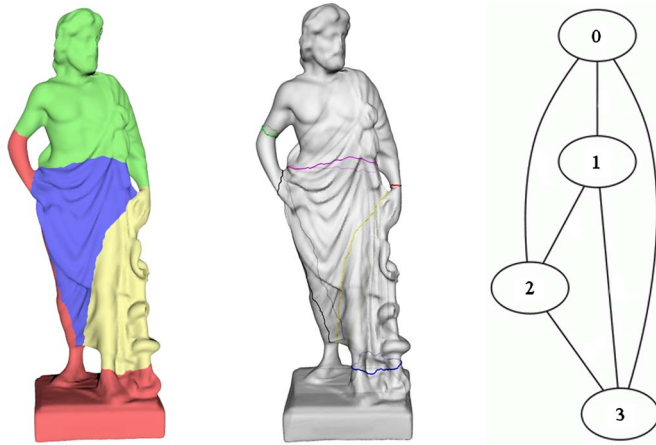


Figure 2: A Pants Decomposition Graph for Greek. Left: A decomposition computed on Greek; Middle: Its maximal cut system; Right: Its corresponding PD-Graph.

For a surface  $M$  with genus- $g$ , its homology basis consists of  $2g$  cycles. The homology basis formed by the *handle* and *tunnel* cycles [1] are used to compute the pants decomposition. In this paper, we apply the algorithms/software of [2] to obtain the homology basis. First, we pick a subset  $B$  composed of  $g$  simple and pairwise disjoint cycles  $\{b_1, b_2, \dots, b_g\}$  (e.g. all the  $g$  tunnel (handle) loops, or some disjoint  $k$  tunnels plus  $g - k$  handles, form such a subset  $B$ ). The maximal cut system is then computed based on  $B$ . First, we slice along all the cycles in  $B$ , which makes  $M$  become a type- $(0, 2g + b)$  surface. We denote this surface as  $\hat{M}$  and its  $2g + b$  boundaries as  $W = \{w_1, w_2, \dots, w_{2g+b}\}$ . Second, we iteratively pick two boundaries  $w_i$  and  $w_j$  from  $W$  and compute the cycle  $w'$  which is homotopic to  $w_i \circ w_j$ . These three cycles can form a pants patch  $M_k$ . Then we remove  $M_k$  from  $\hat{M}$  and the remaining surface  $\hat{M}' = \hat{M} \setminus M_k$  becomes type- $(0, 2g + b - 1)$ . That is, its genus is 0 but the number of its boundaries are reduced by 1 (since two cycles  $w_i$  and  $w_j$  are removed, and one new cycle  $w'$  are inserted). We repeat this step until  $|W| = 3$ . The detailed algorithm is formulated in Algorithm 1. Fig. 3 illustrates the process.

**Algorithm 1.** Initial PD computation.

In: A type- $(g, b)$  surface  $M$ ,  $B(M) = \{b_1, \dots, b_g\}$ ;

Out: Its PD  $\{M_1, M_2, \dots, M_{2g+b-2}\}$ , where  $M = \bigcup M_i$ .

1. Cut  $M$  open all cycles in  $B$  and get a type- $(0, 2g + b)$  surface  $\hat{M}$  with

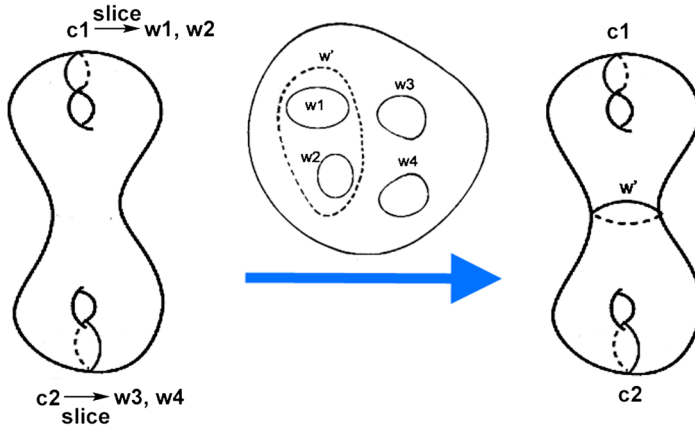


Figure 3: An illustration of initial pants decomposition.

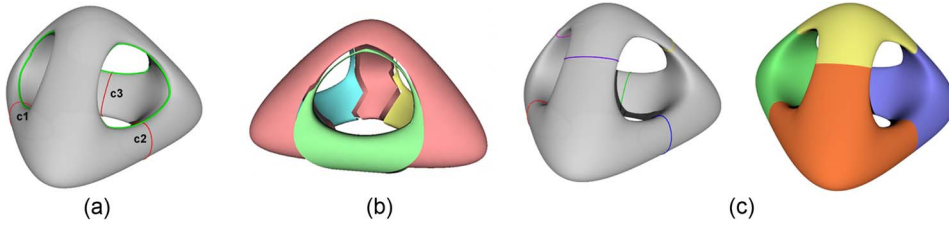


Figure 4: Pants decomposition of a genus-3 surface (a) Homotopy equivalence class (b) PD by previous method (c) PD by our method.

boundaries  $W = \{w_1, w_2, \dots, w_{2g+b}\}$ .

2. Pick two cycles in  $W$ , compute a shortest cycle  $w'$  homotopic to  $w_i \circ w_j$ .
3.  $\{w_i, w_j, w'\}$  bound a pants patch  $M_k$ ; remove  $M_k$  from  $\hat{M}$ , remove  $w_i, w_j$  from  $W$  and add  $w'$  into  $W$ .
4. If the remaining surface  $\hat{M} \leftarrow \hat{M} \setminus M_k$  is not a pants patch, go to STEP 2, otherwise STOP.

### 2.3. Homotopic cycle computation

In the above algorithm and the following sections, we frequently use an operation which computes a cycle  $w'$  homotopic to cycle  $w_i \circ w_j$ . In this section we would elaborate the details of the computation.

Given a surface  $M$  with multiple boundaries  $w_i$  ( $i = 1, 2, \dots$ ), we would like to compute the shortest cycle  $w'$  homotopic to cycle  $w_i \circ w_j$ :

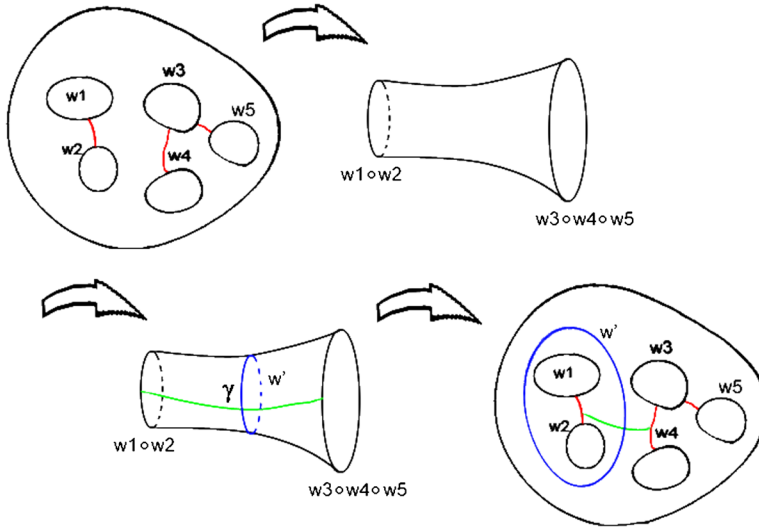


Figure 5: Computing Homotopic Cycle.

1. Connect  $w_i$  and  $w_j$  with shortest path. Connect all the boundaries of  $M$  except  $w_i$  and  $w_j$  together with shortest paths (Red curves in Fig. 5).
2. Slice all those computed shortest paths apart.  $M$  becomes a topological cylinder.
3. Connect the two boundaries of the cylinder using shortest path  $\gamma$  (Green curve in Fig. 5).
4. Slice  $\gamma$  apart. Each point  $p$  on  $\gamma$  will split to a pair  $(p, \tilde{p})$ . Find the pair of points with the minimal length of the shortest connecting path. This path is the cycle  $w'$  (Blue cycle in Fig. 5).

## 2.4. Enumerating pants decompositions

To search for a desirable pants decomposition from different homotopic classes, we need to traverse all the possible PD. Here we introduce two basic topological operations, denoted as *Associativity Move* (A-Move) and *Simple Move* (S-Move) [6]. By sequentially applying one of these two operations, any two topologically different pants decompositions can transform to each other. Therefore, all topological types of PD can be enumerated. We illustrate these operations in Fig. 6, and elaborate their definitions and computations as follows.



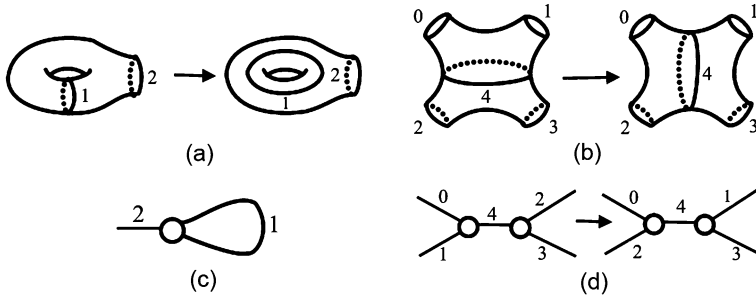


Figure 6: A-Move and S-Move. (a) An S-Move conducted on a handle patch i.e., a self-connected node (c) on the PD-graph; (b) An A-Move; which modifies the connectivity of the PD-graph (d).

- Associativity move (**A-move**). If we glue two adjacent pants patches along one of their shared boundary and get a  $(0, 4)$ -typed patch, then we can perform an A-move. There must exist a circle  $c'$  which is disjoint to the elements in  $B(M)$  other than  $c$  and intersects  $c$  with 2 distinct points. After replacing  $c$  by this circle we can get a new PD.
- Simple move (**S-move**). If we glue one pants patch along one of its shared boundary and get a  $(1, 1)$ -typed patch, then we can perform an S-move. There must exist a circle  $c'$  which is disjoint to the elements in  $C(M)$  other than  $c$  and intersects  $c$  with a single point. After replacing  $c$  by  $c'$  we can get a new PD.

Using our PD-graph, for a cycle  $c$ , we do not need to explicitly glue adjacent patches then compute the resultant topological type such as  $(0, 4)$  or  $(1, 1)$ ; we can directly check this cycle's corresponding edge  $e$  on the PD-graph. If  $e$  links two different nodes  $p \neq q$ , then gluing pants patches  $p$  and  $q$  along  $c$  will lead to a patch whose type is  $(0, 4)$ . If  $e$  links a node to itself  $p = q$ , then gluing  $p$  and  $q$  along  $c$  will lead to a patch of type  $(1, 1)$ . Therefore, on a PD-graph, we can efficiently enumerate different pants decompositions using A-moves and S-moves, and traverse all possible pants decompositions of the surface  $M$ .

**2.4.1. Computing A-move** Consider two adjacent pants patches  $p, q$ ; let  $\Gamma_1, \Gamma_2, \Gamma_3$  be 3 boundaries of  $p$  and  $\gamma_1, \gamma_2, \gamma_3$  be 3 boundaries of  $q$ , and  $\Gamma_3$  and  $\gamma_3$  be the same boundary  $\Gamma_3 = \gamma_3 \in C$ . After an A-Move,  $\Gamma_3 = \gamma_3$  will be glued and a new cycle  $\Gamma_4 = \gamma_4$  with a different homotopy type is generated; and  $C$  will be updated:  $C' \leftarrow C \setminus \{\Gamma_3\} \cup \{\Gamma_4\}$ . Fig. 7 illustrates the process of computing A-move.

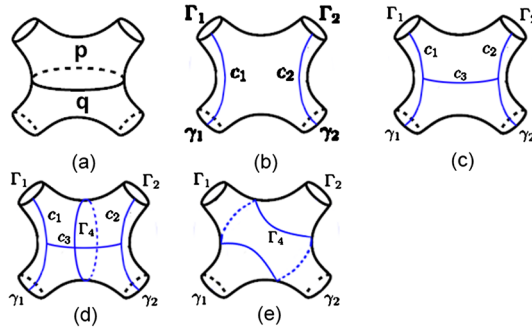


Figure 7: Computing A-Move. (a) The initial pants patches  $p, q$ . (b) Glue  $p, q$  and get a patch with type  $(0,4)$  Connect each pair of boundaries with  $c_1$  and  $c_2$ . (c) Connect  $c_1$  and  $c_2$  with  $c_3$ . (d) Find the point pair with minimal length of shortest path. (e) Different homology type of A-Move.

**Algorithm 2.** Computing an A-Move.

**In:** Adjacent patches  $p, q$ , and their ordered boundaries  $(\Gamma_1, \Gamma_2, \Gamma_3) \subset p$ ,  $(\gamma_1, \gamma_2, \gamma_3) \subset q$ ,  $\Gamma_3 = \gamma_3$ .

**Out:**  $\Gamma_4 = \gamma_4$  and updated maximal cuts.

1. Glue  $p$  and  $q$  along  $\Gamma_3(\gamma_3)$ .
2. Compute the shortest path  $c_1$  connecting  $\Gamma_1$  and  $\gamma_1$  and the shortest path  $c_2$  connecting  $\Gamma_2$  and  $\gamma_2$ .
3. Compute the shortest path  $c_3$  connecting  $c_1$  and  $c_2$ .
4. Slice  $c_3$  apart: each point  $t_k \in c_3$  splits to a pair of points  $(T_k, \tilde{T}_k)$ . Compute the shortest paths  $s_k$  connecting each pair of  $T_k$  and  $\tilde{T}_k$ , pick the shortest path that has the minimal length. This path is the corresponding cycle  $\Gamma_4$ .
5. Update the maximal cut system  $C' \leftarrow C \setminus \{\Gamma_3\} \cup \{\Gamma_4\}$ .

Here all the shortest paths between two curves/cycles are computed on triangle mesh using the Dijkstra algorithm. In the next section, we will apply different weights on triangle edges following different geometric criteria. Also, all traced shortest pathes are enforced to circumvent  $c_1, c_2$  (in step 3), boundaries, and all the cycles in  $C$ . This guarantees that the resulting  $\Gamma_4$  is simple, non-trivial, and not homotopic to any other boundaries. The computed  $c_1, c_2, c_3$  and the result  $\Gamma_4$  are shown in blue in Fig. 7(b,c,d). Note that the indices of boundary cycles matter: if the input boundaries of  $q$  becomes  $(\gamma_2, \gamma_1, \gamma_3)$ , then the result is shown in Fig. 7(e).

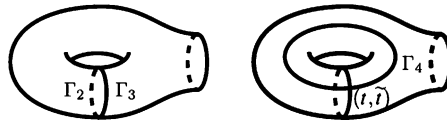


Figure 8: Computing S-Move. Left: The initial pants patch  $p$ ; Right: Find the point pair with minimal length of shortest path connecting them.

**2.4.2. Computing S-move** Consider a pants patch  $p$  with 3 boundaries  $\Gamma_1, \Gamma_2, \Gamma_3$ , suppose  $\Gamma_2$  and  $\Gamma_3$  are the same boundaries  $\Gamma_2 = \Gamma_3 \in C$ . An S-move can be computed by tracing shortest paths. Fig. 8 illustrates the process of computing S-move. For each point  $t \in \Gamma_2$ , there should be a corresponding point  $\tilde{t} \in \Gamma_3$ . Note that when we glue  $\Gamma_2$  and  $\Gamma_3$ ,  $t$  and  $\tilde{t}$  will merge, but now since the surface is sliced open, the shortest path  $Path(t, \tilde{t})$  between  $t$  and  $\tilde{t}$  will go following a longer path like a handle or tunnel cycle. We search all the point pair  $(t, \tilde{t})$  on  $\Gamma_2(\Gamma_3)$ , and pick the pair  $t, \tilde{t}$ , whose  $Path(t, \tilde{t})$  has the shortest length. This shortest path is a new cycle  $\Gamma_4$  with a different homotopy type. And  $C$  shall be updated accordingly:  $C' \leftarrow C \setminus \{\Gamma_3\} \cup \{\Gamma_4\}$ .

### 2.5. Topological surgery

In order to decompose surfaces with trivial topology (four cases discussed in Section 1.2), or more frequently, surfaces with features. We can use the *topological surgery* operation to evolve a region with genus 0 to a handle. Given a pair of surgery points (e.g. features), first punch holes on them and make their one-ring neighbors to become boundaries,  $w_1$  and  $w_2$ . Then compute a cycle  $w_3 \sim w_1 \circ w_2$  bounding them. Then this region bounded by those three boundaries becomes a type-(0, 3) patch and can be matched with a handle region, as illustrated in Fig. 9. This is very helpful in controlling

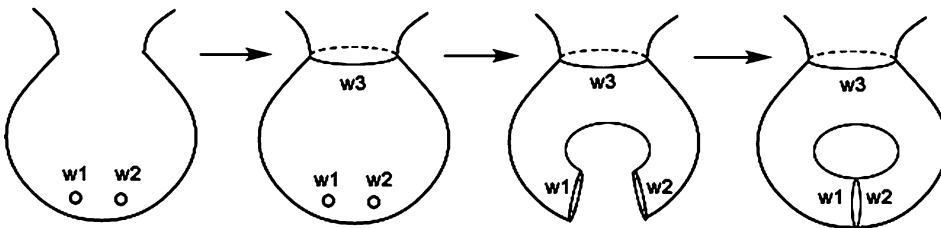


Figure 9: Topological surgery and evolution to a handle.

decomposition using feature alignment and also in matching surfaces with different topologies (details to be discussed later).

### 3. Optimizing pants decomposition

The topological operations enable us to enumerate pants decompositions of different homotopy types. With that the optimization of PD under some pre-determined geometric criterion can be performed. We adopt a few popular criteria to guide this optimization, but note that our topological framework is general and can be applied on other user-specified metrics for different applications. We adopt a greedy search over the PD-graph upon which different geometric criteria are defined. The algorithm starts from an initial decomposition, and conducts either the S-move or A-move if the total cost defined on PD-graph decreases.

#### 3.1. Defining geometric criteria on PD-graph

We adopt several useful geometric criteria to guide the PD optimization. In this framework, since all the tracing of cutting cycles are from Dijkstra's algorithm conducted on the weighted triangle meshes. These criteria can be integrated into the weight of each triangle edge. On the PD-Graph, each node indicates a pants patch and each arc corresponds to a traced cycle. We perform a breadth-first search on the PD-Graph, and pick the decomposition whose cost is minimal.

*Shortest length* Short cutting paths are usually desirable in shape decomposition, because shortened paths better approximate the geodesic curves and avoid the zigzagged boundary geometry of the sub-regions. One can minimize the total length of all the cycles in the maximal cut system. On edge  $e = (v_i, v_j)$ , we can take the Euclidean distance between two vertices as its weight,  $\sigma_l(e) = |v_i - v_j|^2$ , to be used in the Dijkstra tracing. Then the PD-graph, in its optimized state, indicates a partitioning whose boundaries have the minimal total length.

*Minima rule* Human perception often cuts the surfaces along concave regions, which is referred to as the minima rule [9]. Following the minima rule to partition an object along the concave regions, the models are likely to be decomposed into convex sub-parts. For example, for human bodies, the head, arms and legs are naturally separated by the minima rule. We integrate this into our decomposition so that it traces the cutting boundaries

along the concave regions, i.e., the regions with salient negative minimal curvature: (1) compute the minimal curvature  $\kappa(v)$  for each vertex on the surface; (2) normalize the minimal curvature by  $r(v) = (\kappa(v) - \mu)/\alpha$ , where  $\mu$  is the mean and  $\alpha$  is the standard deviation of  $\kappa(v)$  over all vertices of the surface; then (3) further normalize  $r(v)$  into range  $(0, 1)$ . The minima-rule weight of an edge  $e = (v_i, v_j)$  is then defined as:  $\sigma_m(e) = \frac{r(v_i) + r(v_j)}{2}$ .

*Symmetry* Many geometric shapes have intrinsic symmetric patterns, it is natural and desirable that they are also partitioned symmetrically. For example, in order to cut the surface along its symmetry plane, we can define a scalar field, encoding on each vertex  $d(v)$  its (Euclidean) distance to the symmetry plane. We further normalize this scalar field into range  $(0, 1)$ . The weight on each edge is then:  $\sigma_s(e) = \frac{d(v_i) + d(v_j)}{2}$ . An edge that on the symmetry plane has higher priority to be included in the tracing curves.

The final combined weight to be used in optimization is

$$(1) \quad \sigma(e) = \sigma_l(e) \cdot (\sigma_m(e))^{\alpha_m} \cdot (\sigma_s(e))^{\alpha_s},$$

where  $\alpha_m, \alpha_s$  are weights indicating importance of different terms. Note that each of the  $\sigma_m(e)$  and  $\sigma_s(e)$  is normalized into  $(0, 1)$ , bigger corresponding  $\alpha$  indicates larger importance. Other geometric criteria can be integrated similarly.

Fig. 10 and Fig. 11 illustrate PD guided by the aforementioned metrics. Fig. 10 shows an example of adopting the minima rule in PD computation.

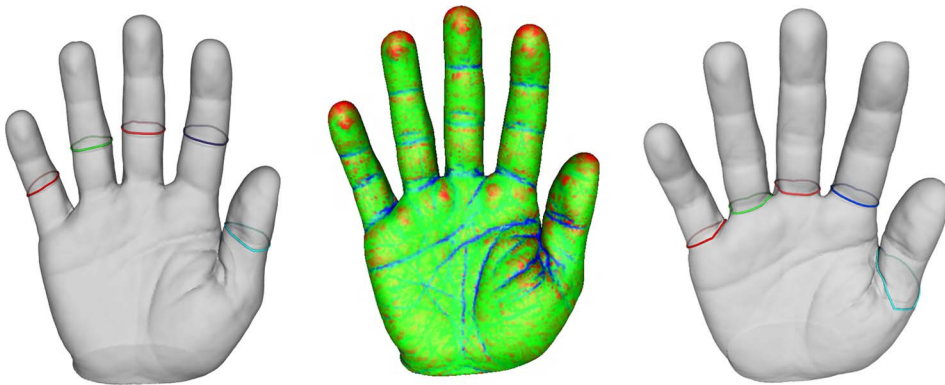


Figure 10: PD of the Hand model using different criteria. Left: PD cycles traced via shortest lengths; Middle: the minimal curvature field; Right: PD cycles traced via minima rule.

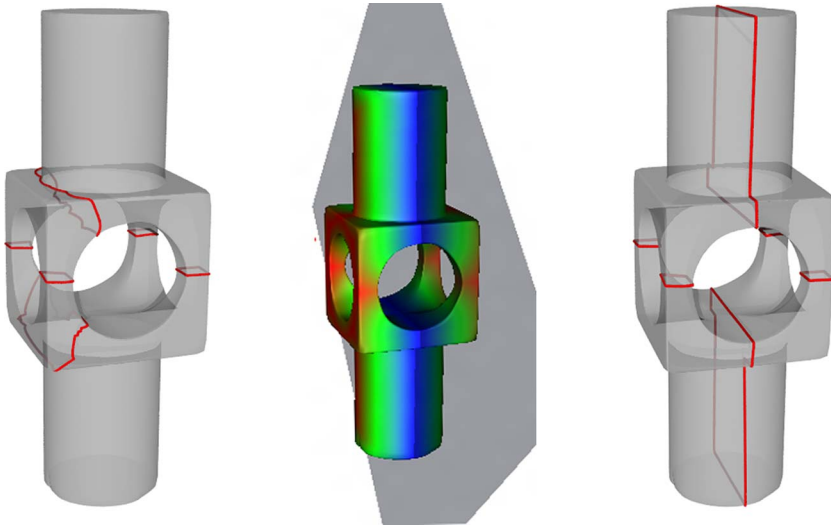


Figure 11: PD of genus-3 mechanical part using different criteria. Left: PD using shortest lengths; Middle: the distance field to the symmetry plane; Right: PD guided by symmetry.

With big  $\alpha_m$ , the cutting boundaries are more likely to go through concave regions, i.e., in (b) the blue regions with high negative minimal curvature. Fig. 11 shows an example of symmetry-emphasized PD computation.

### 3.2. PD optimization under geometric guidance

We use a greedy strategy to search for the optimal pants decomposition from topologically-inequivalent classes. Given a genus- $g$  surface and its PD-Graph  $G_0$ , for each element  $c_i$  ( $i = 1, \dots, 3g - 3$ ) in the maximal cut system of  $M$ , we do an S-moves or A-moves on  $c_i$ . Then we can generate a different pants decomposition and a new corresponding PD-Graph  $G_i$ . Enumerate all the PD-Graphs  $G_i$  ( $i = 1, \dots, 3g - 3$ ) that are evolved from  $G_0$  by one move of different edges, we pick the PD-Graph with the minimal cost, denoted as  $G'$ . If  $G' = G_0$ , the original PD is a locally optimal result, then return with  $G'$ , otherwise, let  $G_0 \leftarrow G'$  and repeat this search.

Since our searching is a greedy approach, the result is only a local optimum. However, we can increase the searching space by comparing two or more moves from the current PD-graph. On the other hand, although our greedy searching cannot guarantee to find the globally optimal solution, it

usually produces topologically simple and geometrically desirable partitioning. More importantly, this approach can be easily generalized for consistent segmentation.

#### 4. Consistent decomposition of multiple models

The PD optimization framework introduced above can be generalized from one surface to multiple surfaces. In the following, we elaborate an algorithm on consistently segmenting two surfaces  $M_1$  and  $M_2$  (generalizing this to handle multiple (more than two) surfaces is straightforward). We can start from a consistent decomposition of  $M_1$  and  $M_2$ , and simultaneously perform the same moves iteratively, when the algorithm stops, the resultant partitioning is consistent. This provides a general consistent partitioning optimization framework for graphics applications such as cross-surface parameterization, matching, morphing, and etc.

##### 4.1. Initial consistent PD computation

We can compute the consistent decomposition of  $M_1$  and  $M_2$  using the algorithm in Section 3. Denote  $C(M_1)$  as a maximal cut system for  $M_1$  and  $C(M_2)$  a maximal cut system for  $M_2$ . We shall discuss the correspondence between two maximal cut systems. The correspondence between pants decompositions of surfaces is determined by the indexing of the two maximal cut systems. Since the maximal cut system is deduced by the initial  $2g + b$  boundaries obtained in Section 2.2, we only need to specify the index of the initial  $2g + b$  boundaries. In practice, the correspondence between  $2g + b$  boundaries can be specified by user, or computed using heuristic methods such as [11] (spatially matching the mass centers of handle loops).

Now, given an arbitrary indexing of  $B(M_1)$  and  $B(M_2)$ , say,  $T_1, \dots, T_{2g+b}$  and  $T'_1, \dots, T'_{2g+b}$ , respectively. We then follow the same consistent order to compute the partitioning cycles, i.e., trace  $w' \sim w_{T'_i} \circ w_{T'_j}$  in  $M_2$  if and only if we trace cycle  $w \sim w_{T_i} \circ w_{T_j}$  in  $M_1$ , and the correspondence between these two maximal cut systems will be consistent. By this means, we can obtain a consistent pants decompositions of  $M_1$  and  $M_2$ .

##### 4.2. Consistently optimizing multiple decompositions

Using a similar aforementioned greedy searching scheme, we simultaneously optimize both decompositions on PD-Graph  $G_0(M_1)$  of surface  $M_1$  and the PD-Graph  $G_0(M_2)$  of surface  $M_2$ . Namely, the same S-Move or A-Move

should be applied on both the cycle  $c_i \in B(M_1)$  and its corresponding cycle  $c'_i \in B(M_2)$ . Then we will obtain a new PD-Graph  $G_i(M_1)$  of surface  $M_1$  and an isomorphic PD-Graph  $G_i(M_2)$  correspondingly. During this searching, we consider weights on each edge to be the sum of the weights defined on  $G_i(M_1)$  and  $G_i(M_2)$ .

## 5. Cross-surface parameterization

When two high genus surfaces  $M$  and  $\widetilde{M}$  are consistently partitioned into pants patches  $\{M_1, \dots, M_n\}$  and  $\{\widetilde{M}_1, \dots, \widetilde{M}_n\}$ , where  $M = \bigcup_{i=1}^n M_i$  and  $\widetilde{M} = \bigcup_{i=1}^n \widetilde{M}_i$ . We can get the cross parameterization  $\phi : M \rightarrow \widetilde{M}$  by individually mapping each corresponding pair of sub-regions  $\phi_i : M_i \rightarrow \widetilde{M}_i$ . Due to the standard topological structure of pants decomposition, this divide-and-conquer can be performed uniformly, using the following three-step algorithm: (1) Flatten the pants patches  $M_i$  and  $\widetilde{M}_i$  into two canonical planar hexagons  $\Omega_i^1$  and  $\Omega_i^2$ ,  $\psi_i : M_i \rightarrow (\Omega_i^1 \cup \Omega_i^2)$ ,  $\widetilde{\psi}_i : \widetilde{M}_i \rightarrow (\Omega_i^1 \cup \Omega_i^2)$ , (2) compose pants mapping  $\phi_i : M_i \rightarrow \widetilde{M}_i$  by  $\phi_i = \widetilde{\psi}_i^{-1} \circ \psi_i$ , (3) relax the map across the decomposition boundaries to reduce the mapping distortion. In our work the mapping distortion is defined as the angle distortion. Angle-preserving property is important in many applications, including texture transfer and quad meshing.

### 5.1. Flattening a pants patch

A pants patch  $M_i$  can be partitioned and parameterized onto two planar hexagons  $\Omega_i^1, \Omega_i^2$ . We adopt a combination of the partitioning methods suggested in [14] and [11]. (1) On  $M_i$  we pick a boundary with the biggest length, call it the outer boundary and denote it as  $\gamma_3$ , then map  $\gamma_3$  onto a unit circle. We then flatten the interior region using the harmonic map onto the unit disk. Then the largest boundary  $\gamma_3$  becomes the external boundary  $\Gamma_3$  of the planar disk and the other two boundaries  $\gamma_1$  and  $\gamma_2$  of  $P$  become the planar region's internal boundaries  $\Gamma_1$  and  $\Gamma_2$ . (2) We connect the three boundaries  $\Gamma_1, \Gamma_2$  and  $\Gamma_3$  with the shortest paths  $\Lambda_1, \Lambda_2$  and  $\Lambda_3$ . These three paths have three corresponding paths  $\lambda_1, \lambda_2$  and  $\lambda_3$  on  $M_i$ . (3) We slice  $M_i$  along  $\lambda_1, \lambda_2$  and  $\lambda_3$ , and get two patches  $M_i^1$  and  $M_i^2$  that are both topological disks. Here we can flatten  $M_i^1$  and  $M_i^2$  using harmonic maps  $\psi_i^1 : M_i^1 \rightarrow \Omega_i^1$  and  $\psi_i^2 : M_i^2 \rightarrow \Omega_i^2$  in order to minimize the angle distortion.

*Optimizing planar hexagon shapes* In [11], regular planar hexagons are used to parameterize  $M_i^j$ . However, using a regular hexagon to parameterize dif-



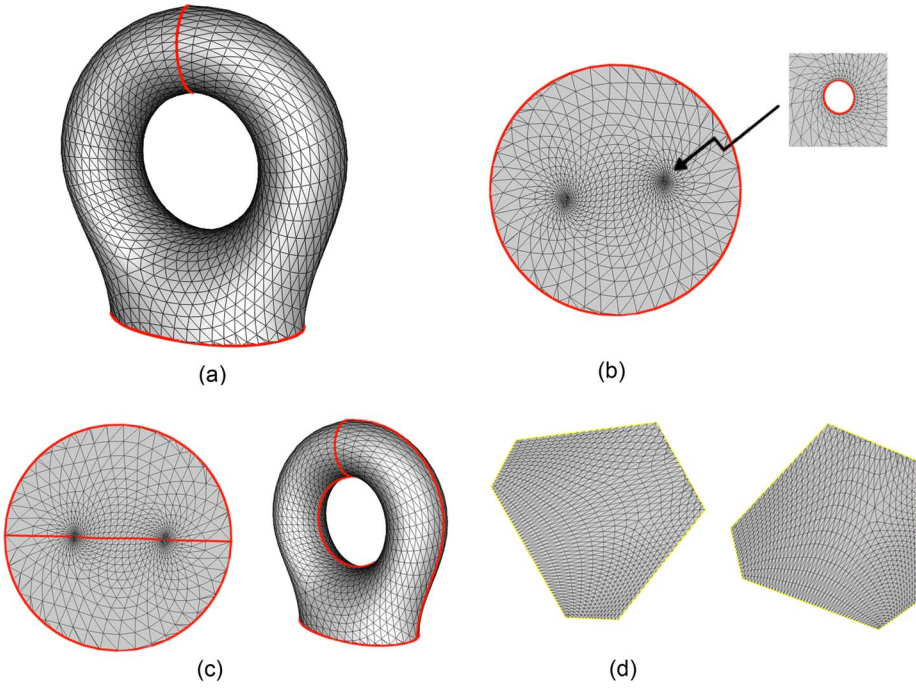


Figure 12: Parameterization of a Pants Patch. (a) A pants patch with 3 boundaries; (b) flattened to a big circle. There will be three boundaries on this planar domain; (c) Compute the shortest path between three boundaries and map the path back to the pant patch. (d) Slice the patch and map the two parts into 2 hexagons.

ferent patches with different shapes can result in a big mapping distortion. Inspired by [21], we also seek a better approximated planar domain shape for each  $M_i^j$ . We denote the lengths of six paths on partitioned patches, which will be mapped to six edges of hexagon, by  $l_i$  ( $i = 1 \dots 6$ ), and denote six corner points by  $C_i$  ( $i = 1 \dots 6$ ) (See Fig. 13). Then we solve the six corners of a hexagon on a circle (centered at the origin  $O$ ) such that each central angle is  $\angle C_i O C_{i+1} / 2\pi = l_i / \sum l_i$  due to simplicity and efficiency. Intuitively, the length of each arc on the circle should be proportional with respect to the corresponding path length. After locating corner points, other boundary points are parameterized onto the boundary of the planar hexagon using the arc-length ratio.

We can evaluate the angle distortion of these mappings by accumulating the angle changes during this flattening [17]:

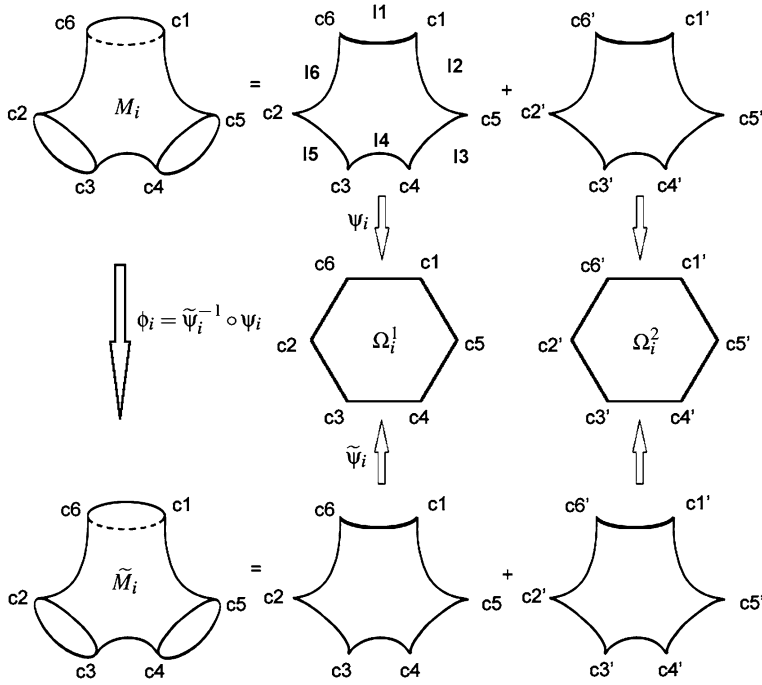


Figure 13: Mapping two pants patches. Each pants patch is mapped to two common planar hexagons. Then they are used as the common domain to compose the mapping.

$$\epsilon(\psi) = \frac{1}{3N_f} \sum_f \sum_{i=1}^3 \left( \frac{\alpha_i - \alpha'_i}{\alpha_i} \right)^2,$$

where  $N_f$  is the number of triangles; on each triangle  $f$ ,  $\{\alpha_i\}$  are the original angles while  $\{\alpha'_i\}$  are the angles of the flattened mesh. Table 1 shows the comparison result. Using optimized hexagon domain significantly reduces angle distortion.

### 5.2. Composing cross-surface mapping

Suppose  $M$  and  $\tilde{M}$  are consistently partitioned into two compatible sets of pants patches:  $\{M_1, \dots, M_n\}$  and  $\{\tilde{M}_1, \dots, \tilde{M}_n\}$ . We parameterize  $M_i$  and  $\tilde{M}_i$  onto two common planar hexagons  $\Omega_i^1, \Omega_i^2$  using the above algorithm. Here each optimal planar hexagon  $\Omega_i^j$  is computed from both  $M_i^j$  and  $\tilde{M}_i^j$  simultaneously.

Table 1: Angle Distortion of the Flattening of the 3-torus Model. Each of the four pants patches  $M_1, \dots, M_4$  is decomposed into two topological hexagons  $M_i^1, M_i^2$ . Rows 1 and 2 show the angle distortions when using the regular planar hexagon as parametric domains; rows 3 and 4 show distortions when using the optimized planar hexagon

Hexagons	Patches	$M_1$	$M_2$	$M_3$	$M_4$
Regular	$\epsilon(\psi_i^1)$	0.292	0.324	0.234	0.391
	$\epsilon(\psi_i^2)$	0.315	0.298	0.335	0.400
Optimized	$\epsilon(\psi_i^1)$	0.0635	0.0642	0.0432	0.0776
	$\epsilon(\psi_i^2)$	0.0780	0.0727	0.0596	0.0797

The composition  $\phi_i^j = (\tilde{\psi}_i^j)^{-1} \circ \psi_i^j$  can then be computed on the common domain  $\Omega_i^j$ , where the inverse of  $\psi_i^j$  is approximated by the barycentric interpolation. Fig. 13 illustrate the procedure of mapping two pants patches.

### 5.3. Global smoothing

The distortion of cross-surface mapping highly depends on the shape of patches. The cross-surface mapping will have low distortion only if the patches are well shaped, i.e., similar to regular hexagon. However, this is not always the case. Here we introduce a smoothing procedure to allow a vertex to migrate from one patch to another, which will drastically reduce the mapping distortion.

For a vertex  $v$ , denote its one ring neighborhood as  $N(v)$ . Suppose  $N(v)$  is lying on two adjacent patches  $p_i, p_j$ , i.e., patches sharing the common boundary. We use the following procedure to relax  $v$ :

1. Map  $p_i, p_j$  to local planar domain  $P$ . Compute the coordinates  $u(v_i)$  of  $v_i \in N(v)$  with respect to  $P$ .
2. Compute the new coordinates  $u(v)$  of  $v$  in  $P$  using the harmonics average of its neighbors:

$$u(v) = \frac{1}{\sum w_i} \sum w_i u(v_i),$$

where  $w_i$  is the harmonic weight of edge  $e(v, v_i)$ .

3. Check if the relocation flips triangle. If not, map  $v$  back to its patch, and update the location of  $v$ . Otherwise, discard the relocation.

Note that when the vertex's one ring neighborhood is lying on more than 2 patches, it is not moved. Fig. 14 shows the procedure. Fig. 15 shows the

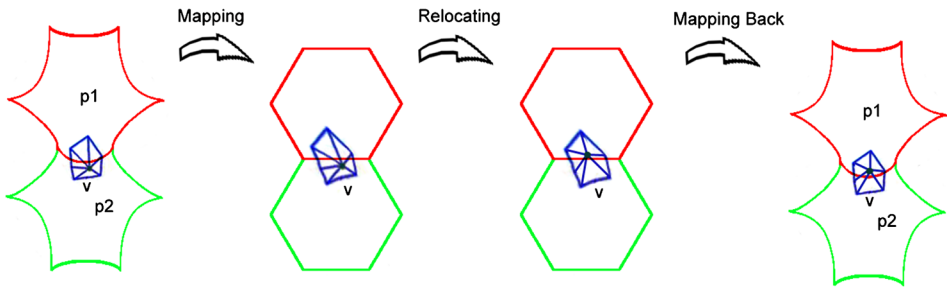


Figure 14: Relaxing Vertex.

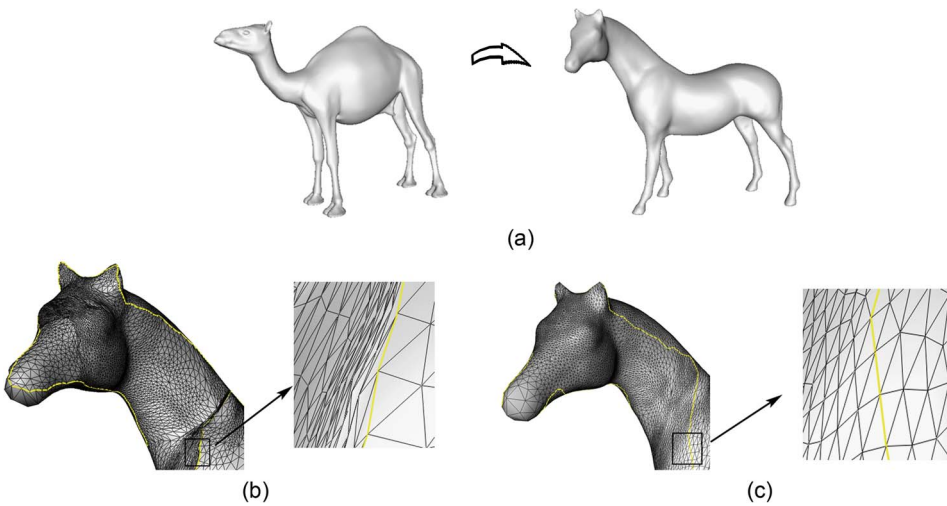


Figure 15: Mapping Relaxation. (a) A camel model is mapped onto a horse. The camel's mesh is transferred onto the horse geometry. The resultant mesh is stretched if the map has large distortion (b); after relaxation the distortion across the cutting boundary has significantly reduced (c).

relaxing result on camel to horse. From it, we can observe that the mapping distortion is greatly reduced by smoothing process.

## 6. Experimental results

*Enumerating pants decomposition and optimization* Fig. 16 shows our computation on enumerating different pants decomposition on the genus-3 Michelangelo-David model. It shows both the maximal cut system and corre-

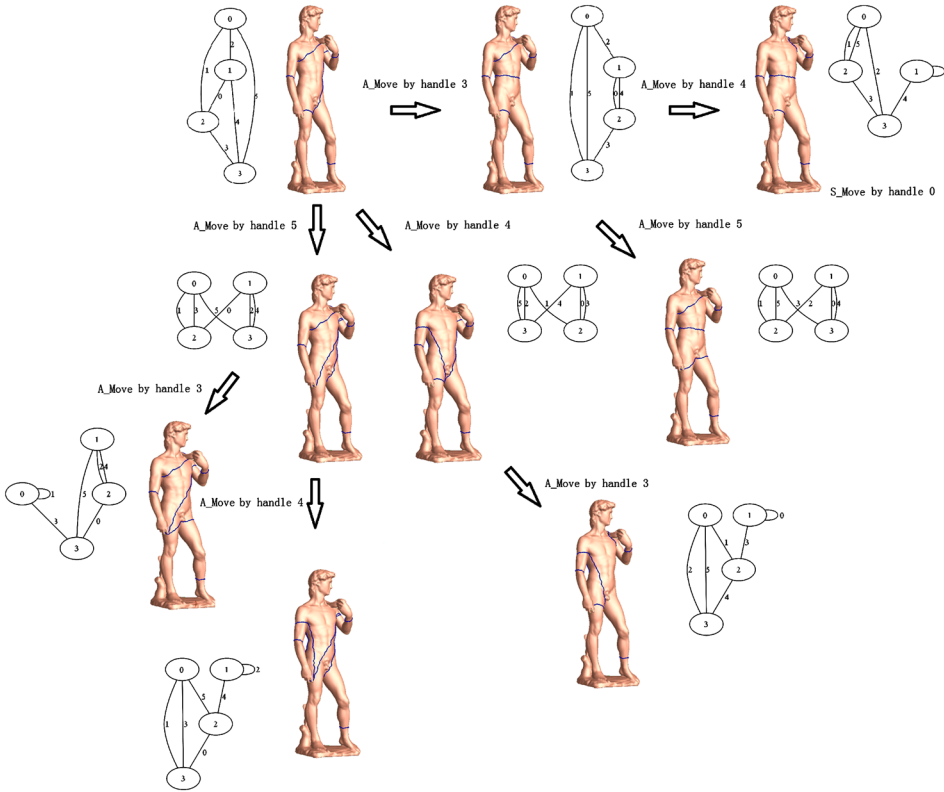


Figure 16: Enumeration of Topologically-different Pants Decomposition on Michelangelo-David Model. Both the decomposition and its corresponding PD-graph are illustrated. This figure shows part of the entire enumeration without considering any geometric criterion. The first row also indicates a path to the pants decomposition with the shortest total length.

sponding PD-graph. The first row also indicates a path to the optimal pants decomposition using shortest length as the criterion.

We show some more optimization results on pants decomposition in genus-5 Botijo and genus-6 Happy-Buddha (Fig. 17). The original pants decomposition are optimized using shortest length as the criterion. In all these examples we can see the final decomposition patches have better shapes than that before the optimization.

*Consistent pants decomposition and cross-surface mapping* Fig. 18 shows some examples of the optimal consistent pants decomposition for multiple



Figure 17: Optimization for Pants Decomposition on Botijo and Happy-Buddha. (a,e) Initial pants decomposition; (b,f) optimized pants decomposition, using shortest length as the criterion. (c,d,g,h) The front and back views of the decomposition results; patches are color-encoded and moved away from each other slightly for better visualization purpose.

objects. The cross-surface mappings are also computed and the results are visualized using the morphing sequence. In Fig. 18(a), surface mapping is generated with no user interference involved. In Fig. 18(b), two surgery points are placed on the head of each model to guarantee the correspondence between their head region.

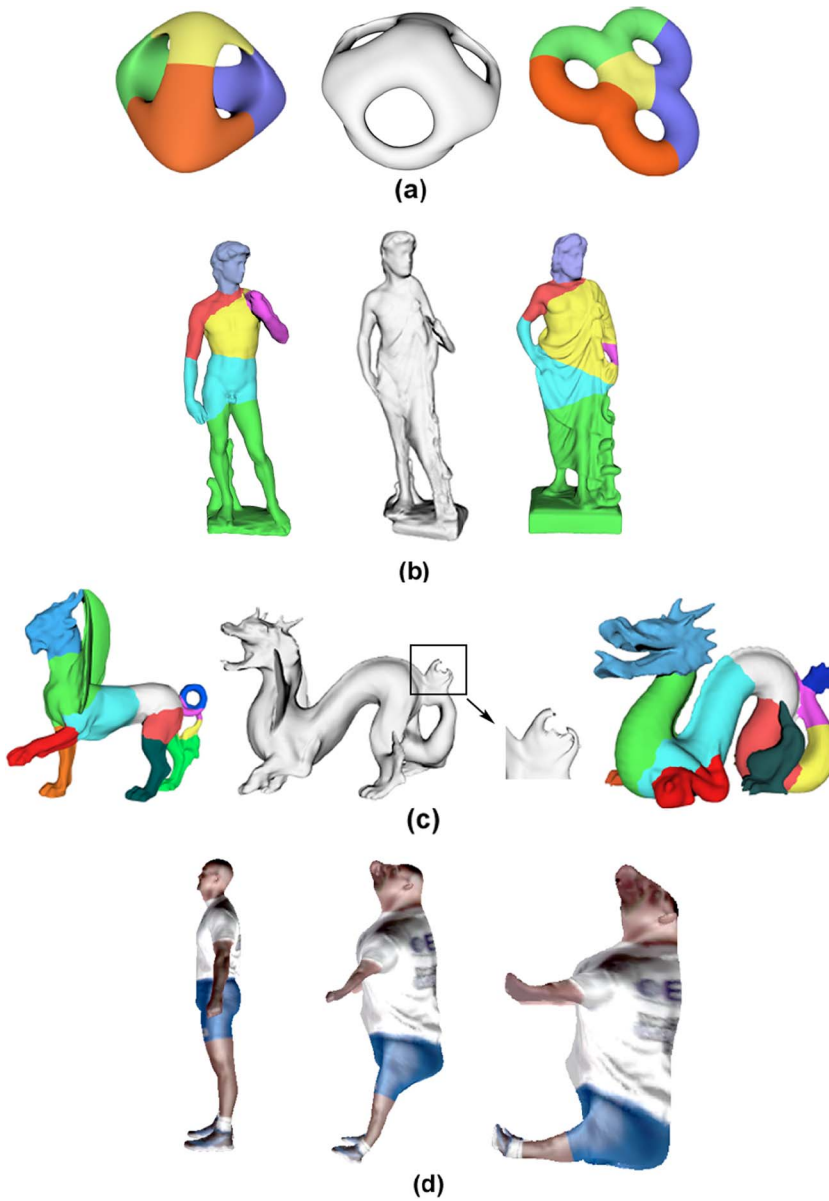


Figure 18: Several Examples for Morphing (0%, 50% and 100% morphing).

Fig. 18(c) shows the cross surface mapping between two models with different topology (Dragon with genus-1 and Feline with genus-2). To enforce the correspondence between their heads and feet, two surgery points are

Table 2: Run-time table for the cross parameterization in our experiments.  $\# V$  is the number of vertices of the two models and  $\# P$  is the number of pants generated.  $T_{A-Move}$ ,  $T_{S-Move}$ ,  $T_{PD}$  and  $T_{all}$  are the computational times for one step of A-Move, S-Move, construction of pants decomposition, and the entire mapping framework, respectively. The time unit is ‘second’

Models	$\# V$	$\# P$	$T_{A-Move}$	$T_{S-Move}$	$T_{PD}$	$T_{all}$
genus-3 To 3-torus	27719 & 29094	4	4.0	0.2	15.5	47.5
Feline To Dragon	67252 & 25065	12	12.2	0.4	36.7	218.1
David To Greek	26156 & 41057	6	4.8	0.2	18.7	52.4
Male To Cow	25002 & 30000	5	3.6	0.2	19.2	58.1

needed in each of those regions. In addition, a pair of surgery points are generated on the tail of dragon corresponding to the handle of feline. We can observe the tearing of handles and the evolution to a region with trivial topology.

Fig. 18(d) shows the cross surface mapping from man to cow with texture on the model, which demonstrates that our method can produce the cross surface mapping with low angle distortion.

*Running time* The decomposition and mapping were implemented in C++ and we performed experiments on a desktop with Xeon(R) E5507 CPU 2.27GHz and 12GB RAM. The runtime statistics are documented in Table 2.

## 7. Conclusion

We propose a general pants decomposition optimization framework that can traverse the space of pants decomposition. Given a predefined geometric criterion define on cycles or pants patches, we search for the optimal partitioning using the PD-graph. This framework is robust, automatic, and efficient. Therefore, it can facilitate cross-surface mapping computation by providing optimized consistent segmentation among multiple models. We demonstrate the effectiveness of the optimization framework by showing optimal decomposition under different geometric criteria and illustrating the morphing indicated by the mapping.

## References

- [1] T. K. Dey, K. Li and J. Sun (2007). On computing handle and tunnel loops. In *International Conf. on Cyberworlds* 357–366.



- [2] T. K. Dey, K. Li, J. Sun and D. Cohen-Steiner (2008). Computing geometry-aware handle and tunnel loops in 3D models. *ACM Trans. Graph.* **27** 45:1–45:9.
- [3] J. Erickson and K. Whittlesey (2005). Greedy optimal homotopy and homology generators. In *ACM-SIAM Symp. on Discrete Algorithms* 1038–1046. [MR2298363](#)
- [4] A. Golovinskiy and T. Funkhouser (2009). Consistent segmentation of 3D models. *Computers and Graphics (Shape Modeling International 09)* **33** 262–269.
- [5] X. Gu and S. T. Yau (2003). Global conformal surface parameterization. In *Proc. Symp. Geometry Processing* 127–137.
- [6] A. Hatcher, P. Lochak and L. Schneps (2000). On the Teichmüller tower of mapping class groups. *J. Reine Angew. Math.* **521** 1–24. [MR1752293](#)
- [7] V. Kraevoy and A. Sheffer (2004). Cross-parameterization and compatible remeshing of 3D models. *ACM Trans. Graph.* **23** 861–869.
- [8] T.-H. KwoK, Y. Zhang and C. C. L. Wang (2011). Efficient optimization of common base domains for cross-parameterization. *IEEE Transactions on Visualization and Computer Graphics* **99**.
- [9] Y. Lee, S. Lee, A. Shamir, D. Cohen-Or and H.-P. Seidel (2005). Mesh scissoring with minima rule and part salience. *Comput. Aided Geom. Des.* **22** 444–465. [MR2152096](#)
- [10] X. Li, X. Gu and H. Qin (2008). Surface matching using consistent pants decomposition. In *SPM'08: Proc. ACM Symp. on Solid and Physical Modeling* 125–136.
- [11] X. Li, X. Gu and H. Qin (2009). Surface mapping using consistent pants decomposition. *IEEE Transactions on Visualization and Computer Graphics* **15** 558–571.
- [12] X. Li, Y. Bao, X. Guo, M. Jin, X. Gu and H. Qin (2008). Globally optimal surface mapping for surfaces with arbitrary topology. *IEEE Transactions on Visualization and Computer Graphics* **14** 805–819.
- [13] B. Li, X. Li, K. Wang and H. Qin (2013). Surface mesh to volumetric spline conversion with generalized polycubes. *IEEE Transactions on Visualization and Computer Graphics* **19** 1539–1551.

- [14] G. Patane, M. Spagnuolo and B. Falcidieno (2004). Para-graph: graph-based parameterization of triangle meshes with arbitrary genus. *Comput. Graph. Forum* 783–797.
- [15] J. Schreiner, A. Asirvatham, E. Praun and H. Hoppe (2004). Inter-surface mapping. *SIGGRAPH* **23** 870–877.
- [16] S. Shalom, L. Shapira, A. Shamir and D. Cohen-Or (2008). Part analogies in sets of objects.
- [17] A. Sheffer, B. Lévy, M. Mogilnitsky and A. Bogomyakov (2005). ABF++: fast and robust angle based flattening. *ACM Trans. Graph.* **24** 311–330.
- [18] V. Kraevoy, D. Julius and A. Sheffer (2007). Shuffler: modeling with interchangeable parts. *The Visual Computer*.
- [19] E. Verdière and F. Lazarus (2007). Optimal pants decompositions and shortest homotopic cycles on an orientable surface. *J. ACM* **54** 18. [MR2351841](#)
- [20] S. Wan, Z. Yin, K. Zhang, H. Zhang and X. Li (2011). A topology-preserving optimization algorithm for polycube mapping. *Computers and Graphics* **35**(3) 639–649.
- [21] C. Wang (2008). Computing length-preserved free boundary for quasi-developable mesh segmentation. *IEEE Transactions on Visualization and Computer Graphics* **14** 25–36.
- [22] H. Wang, Y. He, X. Li, X. Gu and H. Qin (2009). Geometry-aware domain decomposition for T-spline-based manifold modeling. *Computers & Graphics* **33** 359–368. IEEE International Conference on Shape Modelling and Applications 2009.
- [23] K. Wang, X. Li, B. Li, H. Xu and H. Qin (2012). Restricted trivariate polycube splines for volumetric data modeling. *IEEE Transactions on Visualization and Computer Graphics* **18** 703–716.
- [24] W. Yu and X. Li (2011). Computing 3D shape guarding and star decomposition. *Computer Graphics Forums* **30** 2087–2096.
- [25] K. Zhang and X. Li (2012). Optimizing geometry-aware pants decomposition.

KANG ZHANG  
LOUISIANA STATE UNIVERSITY  
USA  
*E-mail address:* [kzhang4@lsu.edu](mailto:kzhang4@lsu.edu)

XIN LI  
LOUISIANA STATE UNIVERSITY  
USA  
*E-mail address:* [xinli@cct.lsu.edu](mailto:xinli@cct.lsu.edu)

RECEIVED 25 JULY 2014