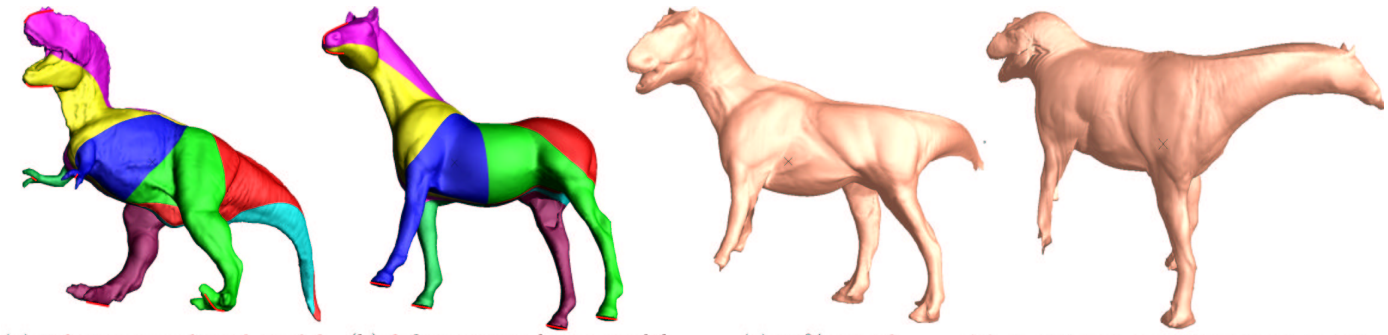
Topology-driven Surface Mappings with Robust Feature Alignment

Christopher Carner, Miao Jin, Xianfeng Gu, and Hong Qin

Stony Brook University



(a) 8 features on lizard model (2 on tail; not shown) (b) 8 features on horse model (2 on rear; not shown) (c) 50% morph (d) A different mapping: head to tail (domain rotation of π radians)

Figure 1: Surface mapping between horse and lizard. The color-coding shows the mapping of each region, guided by eight user-specified feature curves. Our topology-driven method provides mappings of different homotopy type between the two surfaces as shown in (c) and (d). We show feature curves in red.

Abstract

Topological concepts and techniques have been broadly applied in computer graphics and geometric modeling. However, the *homotopy type* of a mapping between two surfaces has not been addressed before. In this paper, we present a novel solution to the problem of computing continuous maps with different homotopy types between two arbitrary triangle meshes with the same topology. Inspired by the rich theory of topology as well as the existing body of work on surface mapping, our newly-developed mapping techniques are both fundamental and unique, offering many attractive advantages. First, our method allows the user to change the homotopy type or *global* structure of the mapping with minimal intervention. Moreover, to *locally* affect shape correspondence, we articulate a new technique that robustly satisfies hard feature constraints, without the use of heuristics to ensure validity. In addition to acting as a useful tool for computer graphics applications, our method can be used as a rigorous and practical mechanism for the visualization of abstract topological concepts such as *homotopy type* of surface mappings, *homology basis*, *fundamental domain*, and *universal covering space*. At the core of our algorithm is a procedure for computing the *canonical homology basis* and using it as a common cut graph for any surface with the same topology. We demonstrate our results by applying our algorithm to shape morphing in this paper.

1 Introduction

Surface mapping is of prime significance in many graphics applications including shape analysis, texture mapping, animation transfer, shape morphing, feature registration, and many other digital geometry processing methods. In principle, parameterization-based surface mapping methods can be classified by the topology of the parametric domain. Typically, surfaces that are homeomorphic to a disk can be easily parameterized over the plane. For topologically more complicated shapes, the parameter domain can be an arbitrary surface in $\mathbb{R}^3$, and so a parameterization will essentially be a mapping between two objects. In such a case, a continuous, meaningful mapping requires that the two surfaces share the same topological attributes, such as genus and number of boundaries. In addition to the topological factors, a desirable surface mapping also hinges upon the specific application demands. For example, texture mapping frequently requires a planar parameterization because most textures are acquired/synthesized as 2D images. However, applications such as remeshing, morphing, and medical model registration are better suited to a surface mapping between topologically equivalent manifolds in 3D.

Furthermore, topological concepts and techniques have been broadly applied in computer graphics and geometric modeling ([6]). However, the *homotopy type* of a mapping between two surfaces has not been addressed before. While other surface mapping methods focus on a single homotopy class, in this paper, we articulate a theoretically rigorous method that produces many continuous maps of different homotopy type between two arbitrary triangle meshes with the same topology (see Figures 2 and 10). The uniqueness of our methodology results from applying the rich mathematical theory of topology to surface classification, rather than relying solely on the embedded geometry. In a nutshell, we first compute a special set of curves, called the *canonical homology basis*, that will cut any two homeomorphic surfaces in the same way into a topological disk (see Figure 6). Then, we can parameterize each sliced surface over a planar domain, using a metric to reduce the distortion. Next, we can create a mapping between the planar domains of two surfaces and extract the final mapping between the original surfaces from the shared planar domain.

In addition, our novel, topology-based approach has several other advantages. First, current methods require the use of heuristics to avoid bad path-tracing while partitioning the surface into multiple regions between features. On the other hand, our technique enables an elegant feature mapping mechanism that can robustly satisfy user-specified, hard constraints without relying on ad-hoc approaches to ensure validity. Moreover, unlike some methods, we do not require a minimum number of features to be inserted. The features (including points and line segments) are always guaranteed to be parts of cutting curves mapped to hard, planar bound-

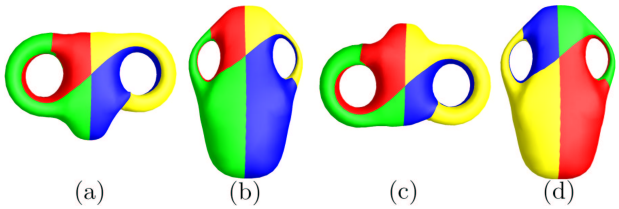


Figure 2: Visualization of two different homotopy classes of mappings between the vase and the two-hole torus by color coding. (a) and (b) show the mapping of the handles between the two-hole torus and the vase: right to right, left to left. Each mapping is produced by a different homology basis on each surface. We obtain a different mapping between the handles in (c) and (d): right to left, left to right. (a) and (c) are 25% morphs.

aries, so the exact feature alignment will be naturally enforced, while non-feature regions are automatically mapped. As a result, our method is best suited to robustly handle shapes with arbitrary genus > 0. Nonetheless, the framework that we have developed in this paper also enables us to easily handle surfaces of genus-0 type.

Aside from acting as a useful tool for computer graphics applications, our method can be used as a rigorous and practical mechanism for visualizing and gaining an intuitive understanding of abstract topological concepts such as *homotopy type* of surface mappings, *homology basis*, *fundamental domain*, and *universal covering space*. Our approach computes many different mappings between two surfaces, each of which corresponds to a single homotopy class (see Figure 10).

The main **contributions** of our method are outlined as follows:

- A novel technique for computing mappings of different homotopy type between two arbitrary surfaces with the same topology.
- A fundamental method for computing *sets of canonical homology basis curves* that slice the surface to a single disk.
- An elegant, local feature correspondence technique that robustly satisfies hard constraints, without the use of heuristics to avoid bad path-tracing.
- A new mechanism that allows the user to quickly change the homotopy type, or global structure, of the mapping with minimal user intervention.
- An approach for the visualization and intuitive understanding of abstract topological concepts such as *homotopy class* of surface mappings, *homology basis*, *fundamental domain*, and *universal covering space*.

Note that, in the interest of space, this paper focuses only on applying our mapping to shape morphing and color transfer. However, the technical scope of this work easily extends to other visual computing applications, including animation transfer, feature registration for medical applications, remeshing, and other geometry processing applications. The space limitation also compels us to assume that the reader has some knowledge of the relevant concepts and language of abstract topology. We only include a minimal explanation of these concepts in the Appendix, and we refer the reader to [16] and [14] for a more thorough treatment.

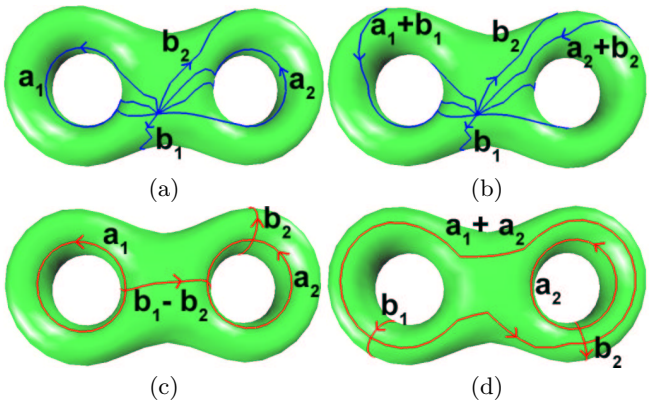


Figure 3: Visualization of several homology basis curve sets for the two-hole torus. (a) and (b) have canonical homology bases, producing a regular cut graph. (c) and (d) show homology bases that are non-canonical.

2 Previous Work

Planar parameterization. Parameterization of 3D surfaces over the plane are the most developed because of the early attention to texture mapping for enhancing the visual quality of polygonal meshes. In order to compute a one-to-one mapping to a planar domain, a surface has to be homeomorphic to a disk. The goal of a planar parameterization of a disc-like mesh M in $\mathbb{R}^3$ is to find an appropriate polygonal domain $\Omega \subset \mathbb{R}^2$ and a suitable piecewise linear mapping $\phi : P \subset \mathbb{R}^3 \rightarrow \Omega$, such that $\phi(x, y, z) = (u, v) = \mathbf{x}^{-1}$ and P is the set of vertices i in M with positions $\mathbf{x}_i$. See [7] for a survey of planar parameterization methods. including Floater's shape-preserving map and mean-value map [8].

Surface cutting and computational topology. Other planar parameterization methods handle surfaces that do not have the topology of a disk, for example, closed surfaces. This requires that the topology be modified by partitioning the surface into a set disk-like charts and mapping them individually to the plane. While some methods slice the surface into multiple charts ([24],[27],[23]), others cut the mesh into a single disk ([2],[3],[11],[31],[12]). The quality of the resulting parameterization depends on the cut placement.

For our purpose of computing an intermediate planar mapping, we must slice open the two homeomorphic surfaces using the same cut graph. In addition, mapping an atlas of multiple charts can be cumbersome, and many of these methods rely on heuristics to place the cuts; they cannot guarantee a canonical planar domain boundary for each surface. To alleviate this difficulty, [6] develop a method for cutting a surface along a canonical homology basis into a single disk. They show that finding such a cut that minimizes the length is generally NP-hard. Thus, they produce an algorithm that approximates a minimum cut graph on the surface, computing a *single* set of such curves. On the other hand, our method can compute many canonical cuts for a given surface, allowing the user to make the most desirable choice. Each cut corresponds to a mapping of different homotopy type (see Figures 2 and 10).

Non-planar parameterization. Non-planar methods parameterize the surface over a 3D domain. In the genus-0 case, the surface can be mapped directly to a sphere, but it is harder to guarantee robustness and avoid flipped triangles. Spherical parameterization methods include [1], [13], [9], [25], and [30]. Other methods use a hierarchical approach to map the surface to some topologically-equivalent, coarser domain.

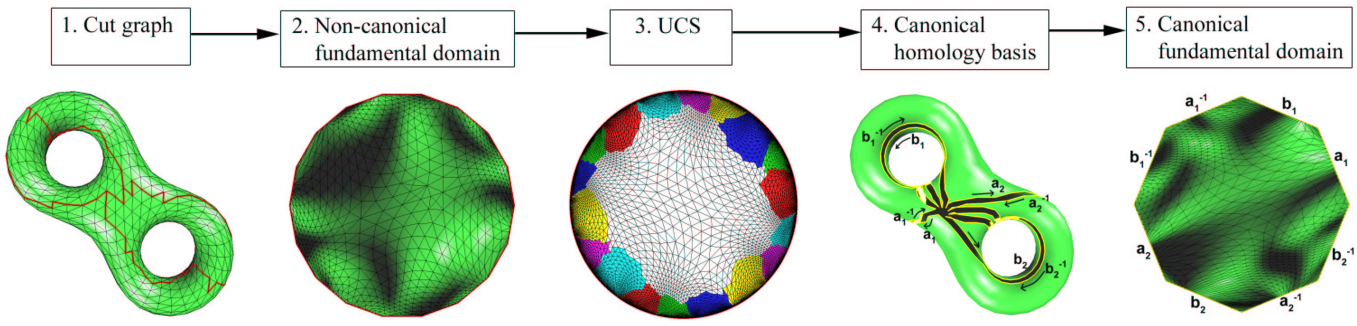


Figure 4: Procedure for computing a canonical fundamental domain, M^*. Step 1: Compute an irregular cut graph that will open the surface to a single disk. Step 2: Slice the surface open to a non-canonical fundamental domain, whose 18 edges are segments of homology basis curves. Step 3: Glue copies of the domain to itself along corresponding segments to form the UCS $\tilde{M}$. Choose a vertex $v \in M$ and a preimage of v, $\tilde{v}_0 \in \tilde{M}$. Trace paths between $\tilde{v}_0$ other preimages $\tilde{v}_k \in \tilde{M}$ of v, each of which corresponds to a single homology class of curves on M. Step 4: Find a combination of such curves that forms a canonical homology basis. Step 5: Slice the surface along this cut graph, constructing a canonical fundamental domain M^*. The ith handle is sliced open along the curve sequence $a_i, b_i, a_i^{-1}, b_i^{-1}$.

Generally, they create a mapping between two surfaces of arbitrary topology by constructing a parameterization over an intermediate base domain and establishing a map between the base meshes. These methods partition the mesh into a set of polygonal regions, each to be mapped onto a single face in the base domain ([4], [20], [21], [18], [26], [28]). However, partitioning M into regions is a problem in itself; it involves tracing consistently-arranged paths between feature vertices on the meshes. Moreover, these techniques requires a set of heuristics to avoid bad path-tracing and guarantee the correspondence of the patches. On the other hand, our method does not encounter such difficulties because we slice the mesh in a consistent way to a single disk, based on the topology of the surface.

Feature correspondence. For feature alignment, some approaches allow the user to specify a set of point constraints, others (like ours) represent features as a collection of curves (edge paths) on the mesh. Some use "soft" constraints ([22],[3]), where the system will attempt to match the features as closely as possible while minimizing a distortion metric. Others create "hard" constraints that will exactly match corresponding feature positions ([5],[19],[28]). Hard constraints can be difficult to handle because there may exist no valid mapping that satisfies all the constraints. However, our technique robustly satisfies user-defined hard constraints easily, without the use of ad-hoc approaches like the addition of so-called Steiner vertices ([5],[19]).

3 Algorithms

3.1 Overview

The intent of our method is to create a continuous, piecewise-linear map between triangulated surfaces with the same genus and number of boundaries. Thus, our goal is to obtain a common domain for the surfaces, from which to extract the final mapping. To this end, for each surface we compute a single planar disk with a regular polygonal boundary, the canonical *fundamental domain* M^* of the surface (see Figure 4).

To obtain this, we first compute a special set of curves (edge paths), a *canonical homology basis*, for the surface (see Figures 3 and 5). This set of curves defines a regular cut graph, along which we slice each surface open in a canonical way. As mentioned above, the homotopy type of the mapping is governed by the choice of a homology basis for each surface, as well as the one-to-one matching between these sets of curves. This will produce a mapping that belongs to a unique homotopy class of surface mappings. In Figures 1,2, and 10, we visualize the effect of the homology basis on the final surface mapping.

Once we have the two homology basis sets and a matching between them, the surface can be parameterized over the canonical *fundamental domain* M^*. Every surface with the same genus can be cut open to a disk in this manner. The overall alignment of mesh features is governed by the way in which the curves on two surfaces correspond. In our system, the curve mappings can be controlled by the user before and after the mapping to the plane. Once we have cut the surfaces open, we parameterize them onto the plane. Then, we align these planar domains and extract the mapping from the two sets of vertices and faces.

The outline of our approach is as follows:

1. The user selects a set of feature curves, which we use in the *Feature Handle Conversion* of the surface (see Section 3.4).
2. Compute a set of curves, the *canonical homology basis*, which becomes a cut graph for slicing each surface open to a single disk. The user can choose any computed set to map corresponding structures of surfaces as desired.
3. Slice each surface open to a disk along the cut graph formed by the set of curves.
4. Parameterize each sliced surface over the plane. The set of curves form the boundary of the canonical *fundamental domain* M^*, a polygon with $4g$ sides.
5. Extract the map between the two meshes and construct a *meta-mesh* $\hat{M}$ (see [21]), combining the vertices and tessellations of both models.

With a simple rotation that re-aligns the boundary edges of the polygonal domain regions (which translates to curve matching on the original surfaces), we can easily change the homotopy type, or global structure, of the mapping. Then, we repeat step 4.

3.2 Computing the Canonical Homology Basis

First, we informally introduce the relevant terminology. We say that two curves are *homotopic* to each other if one can be deformed to the other on the surface without any tearing (see

Figure 5). A more general curve classification theory is *homology*; we say that two curves are *homologous* if they bound any 2-dimensional patch on the surface (including handles). Given a surface with genus g, a *homology basis* is a set of $2g$ curves $\{a_1, b_1, a_2, b_2, \cdots, a_g, b_g\}$, each of which is not homotopic to any other, that spans the curve space of the surface. (see Figure 3). A *canonical homology basis* is a homology basis that meets the following criteria:

1. All the curves meet at a single base point, v.
2. Each pair of curves $\{a_i, b_i\}$ *algebraically intersect* each other exactly once.
3. No curve in another pair $\{a_j, b_j\}$ *algebraically intersects* either of $\{a_i, b_i\}$.

Here, we define *algebraic intersection* in the following way. Given two oriented loops a and b on the surface M, let x_I be their *algebraic intersection number*. At any point $p \in M$ where a and b cross, we compute the unit vectors tangent to a and b, $\mathbf{t}_a$ and $\mathbf{t}_b$, respectively. Let the unit surface normal at p be $\mathbf{n}$. If $(\mathbf{t}_a \times \mathbf{t}_b) \cdot \mathbf{n} > 0$, increment x_I; otherwise, decrement x_I. a and b *algebraically intersect* iff $x_I = +1$.

Here, we explain steps 1-4 of Figure 4. A mapping between two genus g surfaces requires that the cut graph structure of both surfaces be equivalent. To accomplish this, we 1) compute *canonical homology basis* curve sets for the surface, and 2) choose a basis for each surface. In the simplest case, each handle i on the surface will be assigned a pair of curves $\{a_i, b_i\}$: one that travels around the hole a_i, and another b_i that runs through it (see Figures 3a and 5). The surface M is sliced along the regular cut graph formed by a canonical homology basis to obtain the canonical *fundamental domain* M^*, a $4g$-gon. The boundary of this polygon has a regular pattern (see Figure 4) $a_1 b_1 a_1^{-1} b_1^{-1} a_2 b_2 a_2^{-1} b_2^{-1} \cdots a_g b_g a_g^{-1} b_g^{-1}$ (the inverse of a curve c is denoted c^{-1}, traversed in the direction opposite to c).

To compute the canonical homology basis, we first compute a *non-canonical* fundamental domain M^*_{nc}, formed by the irregular cut graph of a non-canonical homology basis (see Figure 4, steps 1-2). In this case, the boundary of M^*_{nc} is a set of homology basis curve *segments*, as opposed to the $4g$-sided *canonical* fundamental domain, whose boundary consists of whole curves formed by a regular cut graph on M. We use the method of [12] to compute the irregular cut graph and the resulting non-canonical fundamental domain.

Next (step 3 of Figure 4), we use a concept called the *universal covering space* (UCS) from Riemann surface theory (see the Appendix). Intuitively, the UCS $\tilde{M}$ of a surface M is an infinite topological disk that consists of an unbounded number of copies of the fundamental domain M^* glued together along corresponding homology basis curves or curve segments (i.e. a_i in one copy is glued to a_i^{-1} in the neighboring copy, and b_i in one copy is glued to b_i^{-1} in another copy. Let $p : \tilde{M} \to M$ be the so-called *covering map*, which is a periodic function (see Figure 5). Therefore, each vertex $v \in M$ will have a discrete set of preimages or copies $\tilde{v}_k \in p^{-1}(v)$ in the UCS $\tilde{M}$. The central idea is this: if we pick a base vertex $\tilde{v}_0$ in $\tilde{M}$, any curve γ drawn between the pair $\{\tilde{v}_0, \tilde{v}_k\}$ represents a unique homology class of curves when mapped to the surface by p. Since $\tilde{v}_0$ and $\tilde{v}_k$ each map to $v \in M$, $p(\gamma)$ will be a loop on M. Each $\tilde{v}_k$ corresponds to a curve of different homotopy type.

In essence, we construct the canonical homology basis in the following manner (see Figure 4):

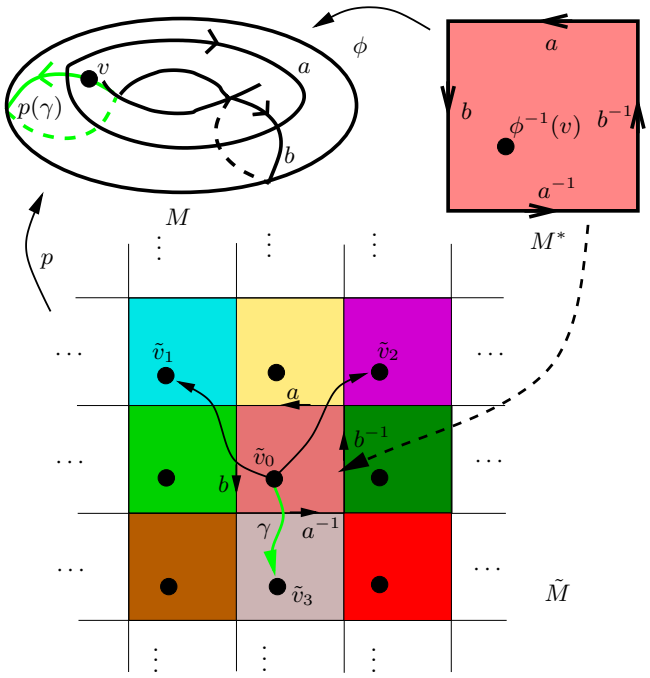


Figure 5: This *universal covering space* (UCS) $\tilde{M}$ is formed by infinite copies of the canonical *fundamental domain* M^* of a torus M, sliced open along a *canonical homology basis*. The *covering map* $p : \tilde{M} \to M$ is a periodic function. The base vertex v has a discrete set of preimages, $p^{-1}(v) = \tilde{v}_0, \tilde{v}_1, \ldots$ on $\tilde{M}$. A curve γ drawn between each pair $\{\tilde{v}_0, \tilde{v}_k\}$ represents a unique homotopy class of curves $p(\gamma)$ when mapped to the surface by p. Here, $p(\gamma)$ is homotopic to b.

1. Compute the irregular cut graph of the surface M (non-canonical homology basis) using the method of [12].
2. Slice M along the cut graph to obtain the non-canonical fundamental domain, M^*_{nc}. The new boundary of the planar mesh is a set of homology basis curve segments.
3. Construct the UCS $\tilde{M}$ by gluing copies of M^*_{nc} to itself along corresponding segments.
4. Choose a vertex $v \in M$ and fix one corresponding base vertex $\tilde{v}_0 \in p^{-1}(v)$ in the UCS $\tilde{M}$.
5. For each $\tilde{v}_k \in p^{-1}$, find the shortest path on $\tilde{M}$ between the base vertex and $\tilde{v}_k$.
6. For each $\tilde{v}_k \in p^{-1}$, project the path to M, where the path is actually a closed loop. This closed curve represents a single *homology class* of curves on the surface.
7. Choose $2g$ projected curves to form the canonical homology basis (see below).

Choosing the canonical homology basis. For any surface with handles (genus $g > 0$), there is an infinite number of canonical homology basis curve sets. For each surface, our technique allows the user to visualize a subset of all possible homology bases on the surface and choose the one most appropriate for the desired mapping. After we compute the UCS, we trace shortest paths to find representative candidates for the homology basis and sort them by length on the surface M. Then, we employ a verification procedure, based on the criteria in Section 3.2 to find a combination of curves that constitutes a valid canonical homology basis. If the curve set is unsatisfactory, we can continue searching until we find a favorable combination.

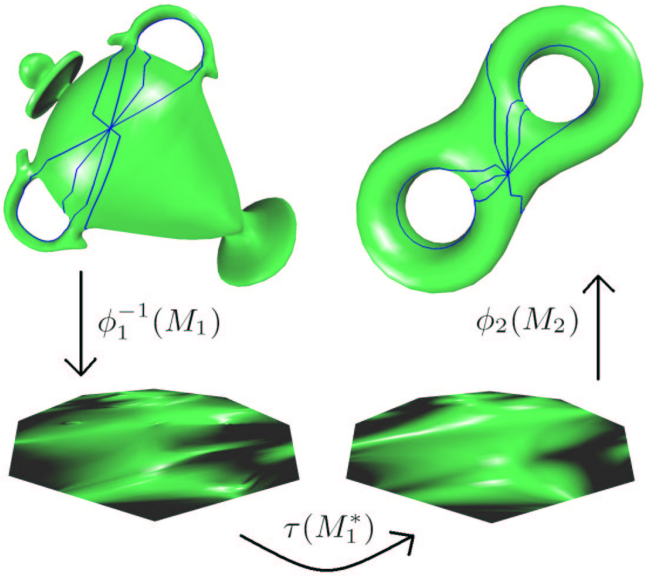


Figure 6: An overview of the surface mapping process. The vase and the two-hole torus are cut open along a common set of curves and parameterized over the canonical fundamental domain M^*. The final mapping is extracted from the overlaid meshes. τ can be used to change the homotopy type of the mapping.

Note that, if the mesh M is too sparse, some curves can share edges, especially around the base vertex v. In order to prevent this, we subdivide the mesh as needed in problem areas to allow more direct routes for shortest path tracing (step 5 of the algorithm above for computing the canonical homology basis).

3.3 Surface Mapping

Planar Parameterization. Once we have computed and chosen a canonical homology basis for each surface, we slice the mesh along these curves to obtain a disk. Each curve a is fixed on one edge of the boundary of the planar $4g$-gon domain; the inverse a^{-1} represents the opposite edge (see Figure 4). We compute a parameterization $\phi : M^* \rightarrow M$ of the vertices $v \in M$ over this domain by using a harmonic mapping $f(v) = \phi^{-1}(v)$ to obtain the planar mesh M^*, minimizing the harmonic energy:

$$E_h(f) = \frac{1}{2}\int_M |\nabla f|^2. \qquad (1)$$

This metric, popular in the computer graphics community, works quite well to reduce the distortion on the plane. In addition, this translates into a simple linear system of equations, whose weights we compute as the mean value coordinates [8].

Final Mapping. After we have parameterized both meshes over the same planar polygonal region, the goal is to compute the surface mapping $\rho : M_1 \rightarrow M_2$, where

$$\gamma(M_1) = \phi_1^{-1}(M_1) \circ \tau(M_1^*) \circ \phi_2(M_2), \qquad (2)$$

where M_1 is mapped to M_2 (see Figure 6), and τ is a transformation of the planar domain.

A mapping between two meshes M_1 and M_2 involves storing, for each vertex, its position, normal, color, and any other desired attributes for each mesh. We compute the surface mapping in the following steps:

1. For each $v \in M_1$, find the face $f \in M_2$ in which the (u, v) position $\phi^{-1}(v)$ lies. Compute its barycentric coordinates within f and interpolate the position and attributes of the vertices (v_0, v_1, v_2) of face f.

2. For each $v \in M_2$, find the corresponding face $f \in M_1$. Compute the new M_1 attributes of v as in the previous step.

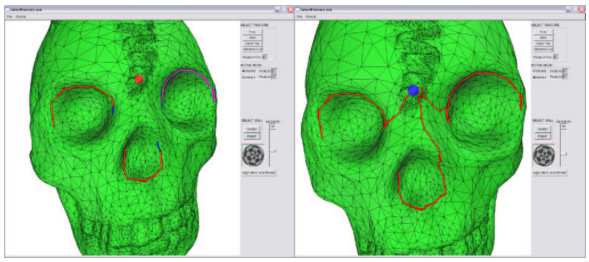
3. Combine the vertices and edges of M_1 and M_2 into the meta-mesh.

4. For each vertex formed by an intersection of an edge of M_1 and an edge of M_2, interpolate the positions and attributes from the endpoints of those edges. Enter the vertex and the newly-formed edges into the meta-mesh.

3.4 Feature Correspondence by Feature Handle Conversion

The matching of homology basis curves determines the alignment of major shape elements of both meshes; handles are mapped to handles. This defines the homotopy type of the map. However, depending on the geometric differences between the surfaces, sometimes it is necessary to adjust the mapping, or even if the surfaces are very similar, the user may want to create a highly unusual mapping of features. To this end, our system incorporates a novel feature-matching procedure, *Feature Handle Conversion*, that satisfies user-defined hard constraints robustly and elegantly. The idea is to treat each feature simply as another handle on the surface. We allow the user to input a set of corresponding open feature curves or edge paths on each mesh before it is cut into a disk (see Figure 7). Then, for each feature, we can create a new boundary in the mesh M by slicing along the curve. Next, we perform a technique known as *double-covering* (see [12]), by which a copy of M is glued to M along the newly-cut boundaries to obtain a new mesh M'. The result is a set of new *feature handles* on each surface. Specifically, for a genus g surface with n feature curves, there will be $n-1$ feature handles; thus, the surface M' will be of genus $2g+n-1$.

Then, we can simply proceed as before; compute the canonical homology basis for each doubled surface, inserting the homology basis curve that travels along each feature and back into the basis set in order to ensure that the features are exactly mapped. Then, we match each pair of corresponding feature handles as well as any other true handles of the surfaces. When the mapping procedure is completed, we will have two copies of the meta-mesh glued together. We simply cut one copy away and sew together the boundary holes created when we sliced along the feature curves. Finally, we obtain a mapping between the original meshes: a single meta-mesh with combined tessellations from M_1 and M_2. This elegant technique guarantees exact feature alignment and a robust surface mapping because the special curves computed on the surfaces are mapped to the hard boundaries on the planar domain. Aside from robustness, this procedure holds an advantage over other ad-hoc hard constraint handling techniques in that it is unnecessary to use heuristics to avoid bad path-tracing between features. Since each feature is treated as a topological handle, the curves are a homology basis, and thus, form a valid configuration on the mesh.



(a) (b)

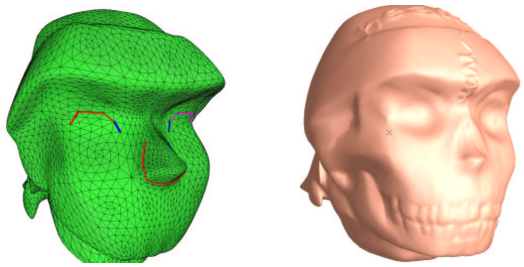
Figure 7: (a) Our interactive system allows the user to select a set of corresponding feature curves on each mesh. We treat each curve as a topological handle. (b) We double-cover the surface and compute the homology basis curves through a single vertex.

4 Implementation

Choosing a canonical homology basis. We have found that it takes less than 2-3 layers of the UCS to produce a canonical homology basis. Each matching of canonical homology bases between the two surfaces will produce a mapping of different homotopy type. Our interactive method produces curve bases one at a time, starting with shorter curves. A naive user can view a preliminary mapping (before the expensive step of meta-mesh construction) for each choice of canonical basis for the two surfaces. If it is unsatisfactory (as explained in Section 3), the system computes the next canonical basis (adding more layers to the UCS if necessary) and presents the resulting partial mapping.

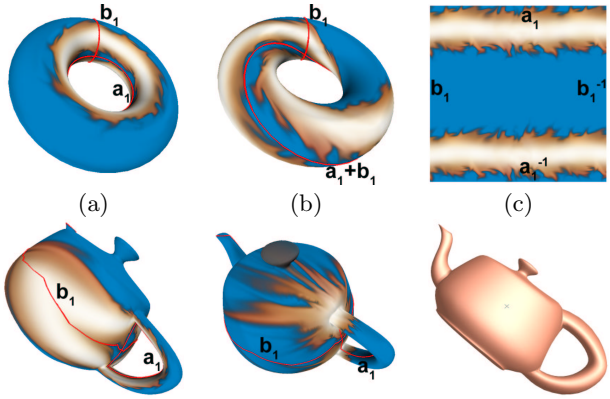
Feature mapping. We have developed an interactive system for feature mapping that allows the user to enter and match a set of feature curves on each mesh (see Figure 7). Then, the system will proceed to compute curves on the meshes along which to slice them open and parameterize them over the planar polygonal region. In this method, the pair of homology basis curves for each feature handle created at a specified feature includes one curve a_i that surrounds the handle and another curve b_i that is traced from the common vertex v, through the handle, and back to v. By using this system, we can also map surfaces with boundaries by treating each boundary as an additional feature; the user simply chooses the mapping between the boundaries on each surface. When the surface is double-covered, we get another feature handle for the boundary.

Merging meshes. We represent a map between two meshes M_1 and M_2 by storing, for each vertex, its position, normal, color, texture coordinates, and any other desired attributes on each mesh. We combine their tessellations into a *meta-mesh* $\hat{M}$ [21]. The computational effort of the final mapping depends on the desired completeness of the map. For applications like morphing between the source (M_1) and target (M_2) meshes, this combined mesh is paramount: we must retain the complete geometry of both source and target. Therefore, $\hat{M}$ combines the complete connectivity, both vertices and edges, of both meshes. The original geometry is retained by including the vertices created by the intersection of original mesh edges on the planar domain. On the other hand, the meta-mesh is much more dense than the original meshes, taking up more memory, and the edge connectivity network may be rather cumbersome. In order to produce the best trade-off between this and the geometry loss, it is possible to choose only those edges whose insertion into the meta-mesh will prevent a change in geometry from the original meshes. To this end, our mesh-merging procedure allows the user to specify the maximum dihedral face angle for determining which edges to insert into the meta-mesh.



(a) Feature matching. (b) 50% morph.

Figure 8: The skull model mapped to "Dirk's Head".



(a) (b) (c)

(d) (e) (f)

Figure 9: In (a) and (b), we show curve $a + b$ matched with a and b with b. (c) shows the texture used to create the effects. In (d), we map a to a, b to b. In (e), we map a to b, and b to a. (f) shows a mapping between the torus and teapot with a morph (10% torus, 90% teapot).

5 Results

In the surface mapping function (Equation 2), we can utilize τ to change homotopy type of the map. For example, by rotating of the domain of M_2 by $2\pi/g$ radians, we can switch the mapping of the handles. This rotation changes the mappings between the homology basis curves of each surface. We illustrate this concept in Figure 2. In the first mapping between M_1 (the two-hole torus) and M_2 (the vase), the right handles are mapped to each other, as are the left handles. In the second, the right handle is mapped to the left handle and vice-versa.

Figure 1 shows the visualization of two homotopy classes of mappings between a lizard and a horse. Here, the features are treated as topological handles. By rotating the planar fundamental polygon by π radians, we can easily change the mapping. In the first mapping, the lizard and horse heads are mapped to each other. In the next, the horse's head is mapped to the lizard's tail and vice-versa.

In Figure 8, we demonstrate the mapping between the head of "Dirk" and a skull. Facial feature curves such as the eyes and nose are specified by the user. Our system slices the mesh open along these curves and double covers it, increasing the number of handles. Then, we compute the canonical homology basis, slice the surface open to the plane, and extract the mapping. Finally, we discard one copy of the mesh and sew the cuts to obtain the surface mapping.

In Figure 9, we visualize several homotopy classes of mappings between the torus and itself, as well as between the teapot and torus. By using this concept, we can create various new coloring effects. The torus in (a) is mapped to the

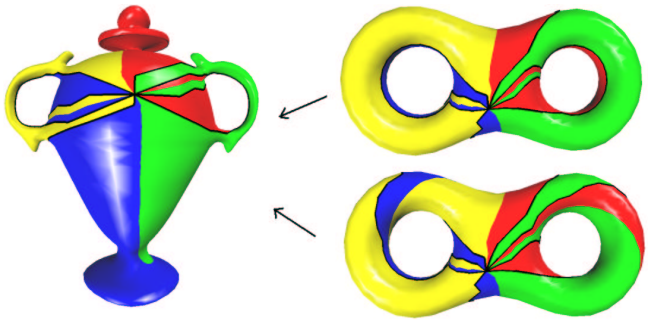


Figure 10: Visualization of two different homotopy classes of mappings between the vase and the two-hole torus. Each mapping is facilitated by the choice of a homology basis on the two-hole torus.

torus in (b); by matching the curve $a+b$ with a and b with b, we obtain a swirling effect on the flame. In (d) and (e), we create two different mappings between the teapot and itself. We obtain a flame around the handle in (d) by simply mapping a to a and b to b. Next, we switch the curve mappings in (e) (a to b, b to a) to revolve the flames through the handle. In (f), we use the mapping between the torus and teapot to enlarge the handle with a morph (10% torus, 90% teapot).

We demonstrate the results of our method on two genus-0 surfaces that each have one boundary in Figure 11. The face meshes were acquired from a real-time 3D scanning device. We use four features to guide the mapping process, and we apply our map to morphing and color transfer in (c) and (d), respectively.

Figure 12 illustrates the mapping between the Stanford bunny and a gargoyle statue. Eight feature curves were specified by the user: two for the wings, two for the ears, two at the bottom, and two on the head. Then, we convert the genus-0 mesh to a genus-7 surface by cutting along the feature curves and double-covering the surface. We color-code corresponding regions on the surfaces to demonstrate the mapping. Figure 12(c) shows the planar domain; we use one colored pie slice for each region between adjacent homology basis curves that travel from v to each feature.

6 Conclusion and Future Work

We have articulated a novel, robust approach to the problem of automatically computing continuous maps between two arbitrary triangle meshes with the same topology. Strongly inspired by concepts from topology theory, we have developed several techniques that can easily create a suite of meaningful mappings between two surfaces. We compute canonical homology basis sets to slice the surfaces open, one of which the user chooses for each surface. Moreover, we presented an elegant and robust technique for satisfying hard feature constraints; we do not require ad-hoc approaches that previous methods use to avoid bad path-tracing between features.

Furthermore, our method tackles the *homotopy type* of a mapping between two surfaces, which has never been addressed before. In addition to acting as a useful tool for computer graphics applications, our method can be used as a rigorous and practical mechanism for visualizing and gaining an intuitive understanding of abstract topological concepts. Our approach computes many different mappings between two surfaces, each of which corresponds to a single homotopy class. Our results visualize these concepts using shape morphing, color coding, and new coloring effects.

By drawing from the rigorous mathematical theory, we have developed fundamental, enabling techniques for interactive graphics applications. There are many possible avenues for future research. Given the infinite number of homology basis curve sets for a given surface, so it would be beneficial to provide a more artistic control for naive users, an intuitive way for computing the "best" set of homology basis curves for a surface based on certain metric. Thus, we aspire to extend our method by including such a metric for determining a good choice and placement of the curves based on geometry (curvature). In addition, flattening the surface can induce large distortion if regions of high curvature are not sliced apart. We are currently working on a technique that, given the matching of homology bases, relaxes the mapping on the surface to obtain the minimum distortion. Also, our mapping can be straightforwardly adapted to many other graphics applications, including remeshing, animation and detail transfer, and shape analysis, which we plan to explore.

M_1	M_2	M_1	M_2	$\hat{M}$	g	g'
Lizard	Horse	25002	19859	216821	0	7
Bunny	Gargoyle	12502	10002	109825	0	7
Dirk	Skull	4322	5002	40458	0	5
2H-Torus	Vase	770	1476	10955	2	2
Teapot	Torus	4256	2500	37569	1	1
Lock	Torus	1120	2500	16999	1	1
Face 1	Face 2	76365	59700	N/A	0	4

Table 1: Mesh sizes (number of vertices) and genus for several data sets used in our system. $\hat{M}$ is the mesh that contains the merged connectivities of M_1 and M_2. The change from genus g to $g' = 2g + n - 1$ reflects the addition of n features by the user, which are treated as new handles on the surface. Face 1 and Face 2 are from Figure 11; the resolution of Face 1 is large enough to avoid constructing $\hat{M}$.

A Appendix

The major concepts and theories in topology used in the paper are introduced in detail in this section.

Homology and Canonical Basis. Suppose M is a triangle mesh and suppose its vertices are $v_0, v_1, \cdots, v_n$. We use $[v_i, v_j]$ to denote a half-edge from v_i to v_j, and $[v_i, v_j, v_k]$ to denote a face with vertices v_i, v_j, v_k. An n-dimensional *chain* is a linear combination of n simplicies with integer coefficients. All n-chains form a linear space called an n-dimensional *chain space*. The boundary operator $\partial_n : C_n \to C_{n-1}$ is a linear operator, which maps a chain to its boundaries. For example, the boundary of a face $[v_0, v_1, v_2]$ is $[v_0, v_1]+[v_1, v_2]+[v_2, v_0]$. Closed 1-chains have zero boundaries, namely, they belong to $ker\partial_1$. The boundary of 2-chains belong to $img\partial_2$. All boundary chains are closed. The first *homology group* is the quotient group, $H_1(M, Z) = \frac{ker\partial_1}{img\partial_2}$. $H_1(M, Z)$ is a finite dimensional group; for a genus g closed mesh, $H_1(M, Z)$ is $2g$ dimensional.

Homotopy of curves. A closed curve on a surface S is a map from the unit interval to the surface, $r : [0, 1] \to S, f(0) = f(1)$. A *homotopy* between two curves r_0 and r_1 is a continuous map $R : [0, 1] \times [0, 1] \to S$, such that $R(\cdot, 0) = r_0$ and $R(\cdot, 1) = r_1$. If r_0 and r_1 can be connected by a homotopy, then r_0 is *homotopic* to r_1.

Homotopy of mappings. Given two surfaces M_1 and M_2 and two mappings $f_0 : M_1 \to M_2$ and $f_2 : M_1 \to M_2$, a homotopy between f_0 and f_1 is a continuous map $F : M_1 \times [0, 1] \to M_2$, where $M_1 \times [0, 1]$ is the direct product of M_1 and the unit interval, such that $F(\cdot, 0) = f_0$, $F(\cdot, 1) = f_1$. Two continuous mappings are homotopic to each other if there exists a homotopy between them. Suppose $f : M_1 \to M_2$ is a continuous map that induces a map $f^* : H_1(M_1, Z) \to H_2(M_2, Z)$. f^* is a linear map between

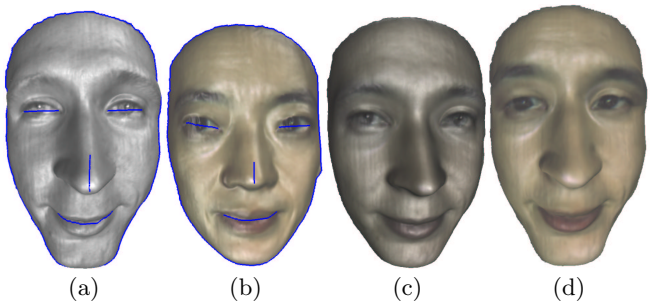


Figure 11: (a) and (b) are the original face meshes, acquired via 3D scanning. The mapping between them is guided by four features, shown in blue. (c) shows a 50% morph. A color value is associated with each vertex of (a) and (b) and is transferred from (b) to (a) using the mapping in (d).

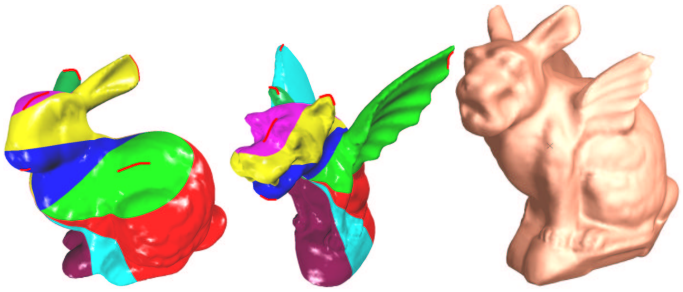


Figure 12: Mapping between Stanford bunny and a gargoyle statue, guided by eight user-specified features (shown by red curves). (a) and (b) show the mapping by color-coding corresponding regions. (c) shows a 50% morph.

two linear spaces; it can be represented as a matrix. Two maps $f_1, f_2 : M_1 \to M_2$ are homotopic if and only if $f_1^* = f_2^*$.

Universal covering space. Suppose M_1, M_2 are meshes. A map $f : M_1 \to M_2$ maps vertex to vertex, edge to edge and face to face. If, for any vertex $v \in M_1$, f maps the one-ring neighbor composed of all faces adjacent to v to a one-ring neighbor of $f(v) \in M_2$, and the mapping is one to one, then f is a *covering* map, and M_1 is a covering space of M_2. Conceptually, for any mesh M, there exists a unique mesh $\tilde{M}$ that is a covering space of M and is an infinite topological disk; $\tilde{M}$ is called the *universal covering space* of M. We denote the covering map as $p : \tilde{M} \to M$. Given a vertex $v \in M$, its preimage $p^{-1}(v)$ is a discrete set. Suppose $\tilde{v}_0, \tilde{v}_1 \in p^{-1}(v)$. If we select arbitrarily two curves $\tilde{r}_0, \tilde{r}_1$ connecting $\tilde{v}_0, \tilde{v}_1$, then $p(\tilde{r}_0)$ and $p(\tilde{r}_1)$ are closed curves on S through p. Furthermore, they are homotopic to each other.

References

[1] M. Alexa. Merging polyhedral shapes with scattered features. *The Visual Computer*, pages 26–37, 2000.

[2] C. Bennis, J.-M. Vezien, G. Iglesias, and A. Gagalowicz. Piecewise surface flattening for non-distorted texture mapping. *ACM SIGGRAPH*, pages 237–246, 1991.

[3] M. Desbrun, M. Meyer, and P. Alliez. Intrinsic parameterizations of surface meshes. *Eurographics*, pages 209–218, 2002.

[4] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. *ACM SIGGRAPH*, pages 173–182, 1995.

[5] I. Eckstein, V. Surazhsky, and C. Gotsman. Texture mapping with hard constraints. *Eurographics*, 2001.

[6] J. Erickson and S. Har-Peled. Optimally cutting a surface into a disk. *Proceedings of the 18th Annual ACM Symposium on Computational Geometry*, pages 244–253, 2002.

[7] M. Floater and K. Hormann. Recent advances in surface parameterization. *Multiresolution in Geometric Modeling Workshop*, 2003.

[8] M.S. Floater. Mean value coordinates. *CAGD*, pages 19–27, 2003.

[9] C. Gotsman, X. Gu, and A. Sheffer. Fundamentals of spherical parameterization for 3d meshes. *ACM Transactions on Graphics*, 2003.

[10] P. Griffiths and J. Harris. *Principles of Algebraic Geometry*. John Wiley and Sons, 1978.

[11] X. Gu, S.J. Gortler, and H. Hoppe. Geometry images. *ACM SIGGRAPH*, pages 355–361, 2002.

[12] X. Gu and S.-T. Yau. Global conformal surface parameterization. *Eurographics Symposium on Geometry Processing*, pages 127–137, 2003.

[13] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Shapiro, and M. Halle. Conformal surface parameterization for texture mapping. *IEEE Transactions on Visualization and Computer Graphics*, pages 181–189, 2000.

[14] M. Henle. *A Combinatorial Introduction to Topology*. Dover, 1979.

[15] K. Hormann and G. Grenier. Mips: An efficient global parameterization method. *Curve and Surface Design*, pages 153–162, 1999.

[16] J. Jost. *Compact Riemann Surfaces: An Introduction to Contemporary Mathematics*. Springer, 1991.

[17] J.R. Kent, W.E. Carlson, and R.E. Parent. Shape transformation for polyhedral objects. In *ACM SIGGRAPH*, pages 47–54, 1992.

[18] A. Khodakovsky, N. Litke, and P. Schroder. Globally smooth parameterizations with low distortion. *ACM Transactions on Graphics*, 2003.

[19] V. Kraevoy, A. Sheffer, and C. Gotsman. Matchmaker: constructing constrained texture maps. *ACM Transactions on Graphics*, 22(3):326–333, 2003.

[20] A. Lee, W. Sweldens, P. Schroder, L. Cowsar, and D. Dobkin. Maps: multiresolution adaptive parameterization. *ACM SIGGRAPH*, 1998.

[21] A.W.F. Lee, D. Dobkin, W. Sweldens, and P. Schröder. Multiresolution mesh morphing. *ACM SIGGRAPH*, pages 343–350, 1999.

[22] B. Levy. Constrained texture mapping for polygonal meshes. *ACM SIGGRAPH*, pages 417–424, 2001.

[23] B. Levy, S. Petitjean, N. Ray, , and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM SIGGRAPH*, 2002.

[24] J. Maillot, H. Yahia, and A. Verroust. Interactive texture mapping. *ACM SIGGRAPH*, pages 27–34, 1993.

[25] E. Praun and H. Hoppe. Spherical parametrization and remeshing. *ACM Transactions on Graphics*, pages 340–349, 2003.

[26] E. Praun, W. Sweldens, and P. Schröder. Consistent mesh parameterizations. *ACM SIGGRAPH*, pages 179–184, 2001.

[27] P.V. Sander, J. Snyder, S.J. Gortler, and H. Hoppe. Texture mapping progressive meshes. *ACM SIGGRAPH*, pages 409–416, 2001.

[28] J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe. Inter-surface mapping. In *ACM SIGGRAPH*, pages 870–877, 2004.

[29] A. Sheffer and E. de Sturler. Parameterization of surfaces for meshing using angle-based flattening. *Engineering with Computers*, pages 326–337, 2001.

[30] A. Sheffer, C. Gotsman, and N. Dyn. Robust spherical parameterization of triangular meshes. *5th Israel-Korea Binational Workshop on Computer Graphics and Geometric Modeling, Tel Aviv*, 2003.

[31] O. Sorkine, D. Cohen-Or, R. Goldenthal, and D. Lischinski. Bounded-distortion piecewise mesh parameterization. *IEEE Visualization*, pages 355–362, 2002.