LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-JRNL-405252

A Genus Oblivious Approach to Cross Parameterization

J. C. Bennett, V. Pascucci, K. I. Joy

July 9, 2008

Computer Aided Geometric Design

Disclaimer

A Genus Oblivious Approach to Cross Parameterization

Janine Bennett [a,*] Valerio Pascucci [b] Kenneth Joy [a]

[a] *University of California, Davis*

[b] *Lawrence Livermore National Laboratory*

Abstract

In this paper we present a robust approach to construct a map between two triangulated meshes, M and M' of arbitrary and possibly unequal genus. We introduce a novel initial alignment scheme that allows the user to identify "landmark tunnels" and/or a "constrained silhouette" in addition to the standard landmark vertices. To describe the evolution of non-landmark tunnels we automatically derive a continuous deformation from M to M' using a variational implicit approach. Overall, we achieve a cross parameterization scheme that is provably robust in the sense that it can map M to M' without constraints on their relative genus. We provide a number of examples to demonstrate the practical effectiveness of our scheme between meshes of different genus and shape.

Key words: Cross Parameterization, Morphing, Morse Theory, Genus Reduction

1 Introduction

Cross parameterizations are maps between two input meshes that play a key role in geometry processing algorithms such as morphing, attribute transfer and mesh blending. There are three main factors that affect the quality of the resulting map: mesh geometry, mesh topology and the placement of *landmark vertices* or hard constraints that the final mapping must maintain. This

* Corresponding author.

Email addresses: janine.bennett@gmail.com (Janine Bennett), pascucci@acm.org (Valerio Pascucci), kijoy@ucdavis.edu (Kenneth Joy).

URLs: http://www.janinebennett.org (Janine Bennett), http://www.pascucci.org (Valerio Pascucci), http://graphics.cs.ucdavis.edu/~joy (Kenneth Joy).

work focuses on minimizing the adverse affect that topology can have on a cross parameterization. Early cross parameterization schemes considered only input meshes that were topological spheres. More recent algorithms allow inputs with an arbitrary number of tunnels but require the input meshes to have equal genus, mapping tunnel to tunnel. Other schemes which allow more general inputs are not guaranteed to work and the authors do not provide a characterization of the input meshes that can be processed successfully. Moreover, the techniques have difficulty dealing with coarse meshes with many tunnels. This can pose problems as input models frequently contain small tunnels that do not reflect the topology of the original object. These tunnels are introduced during model acquisition and/or through the use of standard mesh simplification algorithms. Our approach has several advantages over these previous techniques:

- We provide a provably robust method to generate mappings between meshes of arbitrary and different genus.
- We extend Morse theory to provide a formal framework to analyze and compare cross parameterization methods.
- We provide a completely parametric system that is not affected by convoluted geometry.
- We propose a novel initial alignment scheme which allows for landmark tunnels and constrained silhouettes in addition to standard landmark vertices.

The rest of the paper is outlined as follows. Related work is introduced in section 2 and theoretical foundations are discussed in section 3. We introduce a formalization in section 4 that provides a framework to guarantee the robustness of a cross parameterization scheme and we discuss our implementation in section 5. Optimizations to the map are explored in section 6 and results and future work are discussed in section 7 and section 8 respectively.

2 Related Work

In this section we discuss the related work most relevant to this paper in the areas of cross parameterization, functional morphing schemes, genus reduction and Morse theory.

2.1 Surface Parameterization

A surface parameterization is a bijective map from a given input mesh to a standard parametric domain. Closed surface genus-0 meshes can be parameterized to spherical domains using extensions of planar techniques (Gotsman

et al., 2003; Praun and Hoppe, 2003; Sheffer et al., 2004; Friedel et al., 2005). Meshes of arbitrary genus can be mapped to the plane (Gu et al., 2002; Carner et al., 2005) or to a topologically equivalent base domain (Khodakovsky et al., 2003; Kraevoy and Sheffer, 2004; Schreiner et al., 2004; Gu and Yau, 2003). The work of (Jin et al., 2006) shows that all metric surfaces can be conformally mapped to the sphere, the plane and the hyperbolic disk using Ricci flow. Recent work by (Lee et al., 2006) maps a mesh of genus γ to a series of $\gamma + 1$ spherical domains. One of these domains represents a positive surface while the remaining γ are negative surfaces. The original mesh surface is obtained via boolean difference operations. A thorough exposition on the fundamentals of surface parameterization can be found in (Floater and Hormann, 2004).

2.2 Cross Parameterization

Cross parameterizations determine a bijective mapping between a source mesh, M, and a target mesh, M'. A typical cross parameterization algorithm computes a common base domain across which both M and M' are parameterized. The final map between the two meshes is obtained by determining for each target/source vertex, the source/target face that contains the vertex parametrically. Many of the existing approaches (Schreiner et al., 2004; Kraevoy and Sheffer, 2004; Carner et al., 2005; Lee et al., 1999; Lin and Lee, 2005; Li et al.,a, 2008) require M and M' to be of equal genus. This is due primarily to the manner in which the common base domain is computed. The work presented in (Kraevoy and Sheffer, 2004) is guaranteed to work only on genus zero meshes, while the approach of (Schreiner et al., 2004) finds a maximal non-separating cut graph on each input mesh, guaranteeing the success of their algorithm on meshes of higher genus. The work of (Carner et al., 2005) addresses the homotopy type of a mapping and provides a framework to compute many canonical cuts rather than a single cut graph. The work of (Lin and Lee, 2005) is unique in that it dynamically adds or removes vertices to gradually transform the connectivity from M to M'. (Li et al.,a, 2008) details how to use discrete Ricci flow, introduced in (Jin et al., 2007), to find a globally unique and optimal map between two surfaces of arbitrary but equal genus. Angular distortion is minimized by using a uniformization metric to perform heat diffusion globally.

The methods of (DeCarlo and Gallier, 1996; Lee et al., 2006; Li et al.,b, 2008; Bennett et al., 2007) allow for maps between meshes of different genus. (DeCarlo and Gallier, 1996) requires the user to manually specify control meshes with the same number of faces for both the source and target. The recent work of (Lee et al., 2006) parameterizes meshes of genus γ via a single "positive" spherical parameterization and γ "negative" spherical parameterizations. The approach works well in many cases, however suffers from several key issues.

During genus reduction two boundary loops are computed per tunnel and fine detail in the original mesh may be lost due to the removal of mesh faces between the boundary loops. Pseudo-negative meshes are generated for non-landmark tunnels, however the authors do not provide details on their creation. Furthermore, the algorithm fails on meshes with complicated geometry. The work of (Li et al.,b, 2008) computes a surface pants decomposition introduced by (Hatcher, 1999) and (de Verdiere and Lazarus, 2007). They require the user to specify feature tunnels as well as two surgery points in M' per non-feature tunnel of M (holes are introduced into M' at each of these surgery points). While the manual specification required for non-feature tunnels gives the user an added degree of control over the map, the approach becomes intractable when the number of non-feature tunnels is high.

This paper is an extended version of the work in (Bennett et al., 2007) which introduced the first provably robust cross parameterization algorithm between meshes of unequal genus. This work extends the formal framework in (Bennett et al., 2007) to handle a more general class of inputs. Optimizations to the map are included and additional results are provided on data sets with more complicated geometry and topology.

2.3 Functional Morphing Techniques

Functional morphing algorithms do not maintain the mapping between source and target meshes, however they naturally handle morphing between meshes of unequal genus. The approach of (Wiley et al., 2005) generates a morph sequence between a series of models by computing a distance function for each model. The surface at a given times step is an extremal surface extracted from the weighted average of the functions associated with the input models. The work of (Turk and O'Brien, 1999) computes a variational implicit function in dimension $n+1$ to solve for the morph sequence between two n-dimensional objects (time is the additional dimension).

2.4 Morse Theory

Morse theory characterizes the invariants of a manifold M in terms of the topology of a function f defined on M. The work of (Ni et al., 2004) details how to minimize the number of critical points of f to describe the genus of M. A skeleton of the shape of M is provided by the Reeb graph which is the contraction of the components of level sets of f to points. In recent years Reeb graphs have been used as a search key in shape databases (Hilaga et al., 2001), as well as to characterize complex scientific data (Edelsbrunner et al., 2004). They provide a surface based method for genus reduction (Patane et al.,

2004; Lee et al., 2006; Zhang et al., 2005) that does not require conversion of input models as in volumetric approaches (Zhou et al., 2007; Wood et al., 2004; Nooruddin and Turk, 1999). The notion of persistence was introduced in (Edelsbrunner et al., 2002) and is used to rank the importance of topological features (Gyulassy et al., 2005; Bremer et al., 2004). Jacobi sets are defined in (Edelsbrunner and Harer, 2002) and have been used to explore scientific data sets with multiple fields (Edelsbrunner et al., 2004). In this paper we introduce the notion of a Morse field to formally characterize the correctness of a parameterization scheme.

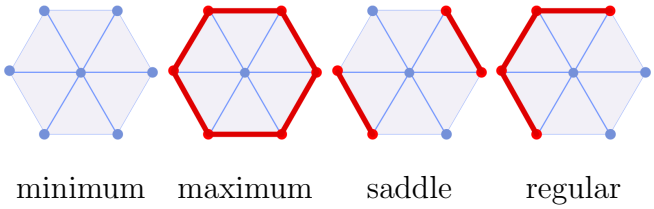


Fig. 1. A vertex is classified by its lower link. Here, simplices in the lower link are drawn in red.

3 Foundation

For completeness we briefly introduce mathematical concepts related to Morse theory and variational implicit functions that are used throughout this paper.

3.1 Morse theory

Let $\mathbf{M}$ be a 2-manifold without boundary embedded in $\mathbf{R}^3$ and let $f : \mathbf{M} \to \mathbf{R}$ denote a smooth real-valued function over $\mathbf{M}$. Assuming a local coordinate system, at a point $x \in \mathbf{M}$ the *gradient* of f at x consists of two partial derivatives and the *Hessian* is the matrix of second-order partial derivatives of f. A point x is *critical* when the gradient of f at x is zero and is *regular* otherwise. The value of f at a critical point is called a *critical value* and a level set of f containing a critical point is called a *critical level set*. When a critical point has a non-singular Hessian it is called *non-degenerate*. In the neighborhood of a non-degenerate critical point x, a local coordinate system can be constructed such that $f(x_0, x_1) = f(x) \pm x_0^2 \pm x_1^2$. Critical points are categorized by their *index* which is equal to the number of minus signs in this equation. A *maximum* has index 2, a *minimum* has index 0 and a *saddle* has index 1.

A *Morse function* is a function f that satisfies two constraints: (1) all critical points are non-degenerate and (2) for all critical points $p \neq q$, $f(p) \neq f(q)$. A

Morse function is *minimal* when f has the minimal number of critical points at each index among all Morse functions on $\mathbf{M}$. *Morse theory* provides techniques to explore the topology of a manifold $\mathbf{M}$ via functions defined on the manifold (Milnor, 1963; Matsumoto, 2002).

A *k-simplex* is the convex hull of $k+1$ affinely independent points. A *triangulation* M of a 2-manifold $\mathbf{M}$ is a set of 0, 1, and 2 simplices commonly called vertices, edges, and faces. A piecewise linear (PL) function f on M is defined by a set of scalar values at the vertices that are extended over the edges and faces of M via linear interpolation. The function f is assumed to be non-degenerate (all function values at vertices are unique) and Simulation of Simplicity (Edelsbrunner and Mucke, 1988) guarantees this through symbolic perturbation.

The *star* of a vertex v consists of all simplices in M that contain v and the *link* of a vertex v, denoted $Lk(v)$, consists of those simplices in the star of v that do not contain v. The lower link, $Lk_{-}(v)$, are those vertices $v_i \in Lk(v)$ such that $f(v_i) < f(v)$ and the edges $\overline{v_i v_j}$ such that $f(v_i) < f(v)$ and $f(v_j) < f(v)$. The upper link, $Lk_{+}(v)$ is defined in an analogous fashion.

The criticality of a vertex with respect to a PL function f is defined in terms of its link as demonstrated in Fig. 1. A vertex is regular if its lower link is a non-empty connected segment of the link. The lower link of a minimum is empty and the lower link of a maximum consists of the entire link. A $k-$fold saddle consists of $k+1$ components along the link with $k \geq 1$.

The *Euler characteristic* χ relates the genus γ of M to the critical points of f:

$$\chi = minima - saddles + maxima = 2 - 2\gamma$$

If f has a single minimum and a single maximum, then f has 2γ saddles, or 2 saddles per tunnel. The level sets of f undergo topological changes at the critical points of f. Level set components originate at minima and end at maxima. Saddles are classified as *split saddles* (where a single level set component splits into two components) and *merge saddles* (where two level set components merge into one). Split saddles and minima are referred to as *parent* critical points and merge saddles and maxima are *children*.

3.1.1 Reeb Graph

The *Reeb graph* encodes the topology of the manifold. It is the skeleton that remains when the components of the level sets of f on M are contracted to points. It consists of a series of *nodes* connected by *arcs*. The nodes correspond to critical points in the mesh, and the arcs correspond to mesh components. A sequence of arcs between a parent and child critical point is called a *component*

path.

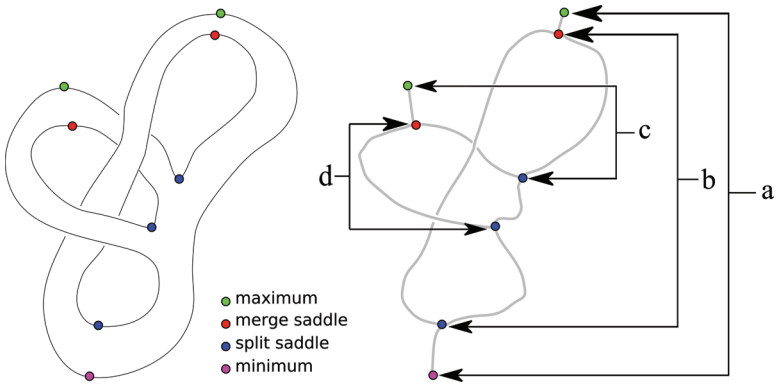


Fig. 2. A 2-manifold and a corresponding Reeb graph. The critical point pairs are labeled and were computed using extended persistence.

The sweep algorithm of (Cole-McLaughlin et al., 2003) computes the Reeb graph by processing mesh vertices in increasing order of function value. The level set associated with the current isovalue f_v is maintained as a collection of cyclic lists of mesh edges whose function values span f_v. The operations necessary to update the current level set are dependent on the classification of the vertex v being processed:

- **Minima:** Create a new component C. Add all edges in $Lk_+(v)$ to C.
- **Maxima:** All edges in $Lk_-(v)$ belong to the same component C. C is empty after removal of these edges.
- **Regular:** All edges in $Lk_-(v)$ belong to the same component C. Remove the edges in $Lk_-(v)$ from C and add edges in $Lk_+(v)$ to C.
- **Split Saddle:** All edges in $Lk_-(v)$ belong to the same component C. Remove the edges in $Lk_-(v)$ from C and add the edges in $Lk_+(v)$ to C. C now contains two components. Separate the components, labeling them D and E.
- **Merge Saddle:** The edges in $Lk_-(v)$ belong to two different components D and E. Merge the sets associated with D and E and store in C. Remove the edges in $Lk_-(v)$ from C and add edges in $Lk_+(v)$ to C.

3.1.2 Persistence

Critical points are often ranked in importance by *persistence*, defined as the difference in function value between parent and child critical point pairs. Traditional persistence (Edelsbrunner et al., 2002) pairs split saddles with maxima and merge saddles with minima. As seen in Fig. 2 extended persistence (Agarwal et al., 2006) pairs the global maximum with the global minimum in addition to the split and merge saddles associated with each tunnel of the mesh. The algorithm processes each critical point in a Reeb graph in increasing

order of function value. Critical points become active when they are first processed and become inactive after they have been successfully paired. When a child critical point (merge saddle or maximum) is processed, descending Reeb graph arcs are traversed monotonically until the first active parent (split saddle or minimum) is reached. In the case that the child critical point is a merge saddle the two descending arcs are then merged. Note that simple component path traversal does not guarantee a successful pairing of all critical points. The mesh in Fig. 2 has "interleaved" tunnels where simple component path traversal of the Reeb graph beginning at two different children terminates at the same parent.

3.1.3 Jacobi Set

The *Jacobi set* is the set of simultaneous critical points of up to k functions on a k-manifold. Given two functions f and g defined on a 2-manifold, the Jacobi set, J, is the set of all critical points of the restriction of f to the level sets of g. These critical points occur when the gradients of f and g are linearly dependent: $\nabla f + \lambda \nabla g = 0$. This implies that J consists of the critical points of the function:

$$h_\lambda = f + \lambda g$$

In the PL setting, functional extrema lie on mesh vertices and the Jacobi set is comprised of mesh edges.

The algorithm of (Edelsbrunner and Harer, 2002) computes the Jacobi set of two functions on a mesh, checking the criticality of each edge individually and returning the union of all critical edges. An edge e with function values (f_1, g_1) and (f_2, g_2), is critical if its lower link is empty or consists of both v_3 and v_4, see Fig. 3.

λ is defined such that $h_\lambda(v_1) = h_\lambda(v_2)$. Thus, for e:

$$\lambda = \frac{f_1 - f_2}{g_2 - g_1}$$

The edge e is critical if both $h_\lambda(v_2) < h_\lambda(v_3)$ and $h_\lambda(v_2) < h_\lambda(v_4)$ or if both $h_\lambda(v_2) > h_\lambda(v_3)$ and $h_\lambda(v_2) > h_\lambda(v_4)$.

3.2 Variational Implicit Functions

A *radial basis function* is a real-valued function whose value depends only on the distance from a specified origin. In this paper we use the radial basis function: $\rho(\mathbf{x}) = |\mathbf{x}|$.

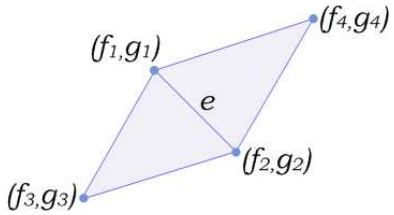


Fig. 3. The edge e with function values ranging from (f_1, g_1) to (f_2, g_2) is critical if the gradients of f and g are linearly dependent on e.

A $k+1$-dimensional *variational implicit function* ψ (Turk and O'Brien, 1999) is used to describe the evolution of k-dimensional objects over time. It does this by minimizing a given energy while satisfying a set of input constraints: $\psi(\mathbf{c_i}) = h_i$. The input constraints consist of two types: *boundary* (where ψ is zero) and *normal* (where ψ is positive or negative). The function ψ is expressed in terms of ρ:

$$\psi(\mathbf{x}) = \sum_{j=1}^{n} d_j \rho(\mathbf{x} - \mathbf{c_j}) + P(\mathbf{x})$$

Here $\mathbf{c_j}$ are constraint locations, d_j are weights and $P(\mathbf{x})$ is a degree one polynomial. To determine the d_j and $P(\mathbf{x})$ that satisfy the interpolation constraints, the linear system is solved using the constraints as input:

$$h_i = \sum_{j=1}^{n} d_j \rho(\mathbf{c_i} - \mathbf{c_j}) + P(\mathbf{c_i})$$

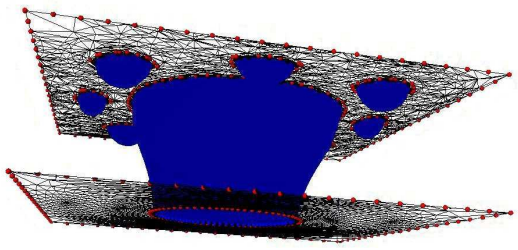


Fig. 4. A 2-dimensional variational implicit function.

4 Formalization of the problem

In this section we introduce a formal framework to analyze the correctness of a parameterization method.

Definition 1 *Let M be a $2-$manifold without boundary and consider the map $\Phi = (f, g, b) : M \to \mathbb{R}^2 \times \mathbb{R}^2 \times [0, 1]$ where f and g are Morse functions and b is a binary function. We call (f, g, b) a* Morse field *if the following conditions hold:*

(1) b is 1 on J, the Jacobi set of f and g, and b is constant on each connected component of $M \setminus J$.
(2) On any level set of g, the value of f on distinct components is distinct.
(3) For any component C of a non-critical level set of g:
  (a) $f|_C$ is a minimal Morse function.
  (b) The values of b approaching a critical point of $f|_C$ from the left and from the right are distinct.

Theorem 1 *Given two Morse fields $\Phi_1 : M_1 \to \mathbb{R}^2 \times \mathbb{R}^2 \times [0,1]$ and $\Phi_2 : M_2 \to \mathbb{R}^2 \times \mathbb{R}^2 \times [0,1]$, if Φ_1 and Φ_2 have the same image M_I then $\Phi_2^{-1} \circ \Phi_1$ is a bijection from M_1 to M_2.*

Proof 1 *$\Phi_2^{-1} \circ \Phi_1$ is a bijection from M_1 to M_2 if both $\Phi_1 : M_1 \to M_I$ and $\Phi_2 : M_2 \to M_I$ are bijections. Without loss of generality we show that $\Phi_1 : M_1 \to M_I$ is a bijection. By construction M_I is the image of Φ_1 and therefore $\Phi_1 : M_1 \to M_I$ is surjective. Assume by contradiction that there exists $x, y \in M_1, x \neq y$ such that $\Phi_1(x) = \Phi_1(y)$. This implies $f(x) = f(y)$ and $g(x) = g(y)$ and $b(x) = b(y)$. Without loss of generality assume that f is restricted to a level set of g. By definition, along this level set $g(x) = g(y)$. The ranges of f restricted to non-critical level set components of g are disjoint, therefore $f(x) = f(y)$ implies x and y lie on the same component C of a level set of g. Since $f|_C$ is a minimal Morse function there are only two critical points of $f|_C$ on C. By removing those two critical points we are left with two open intervals C_0 and C_1 where f is monotonic. This implies if $x \in C_0$ then $y \in C_1$. However, by continuity, the value of b on x and y has to be distinct since x and y belong to two components of g that approach the same critical point from the left and from the right. This is a contradiction, therefore Φ_1 is injective.* □

The definition of a Morse field applies to the PL setting where Σ is a 2-dimensional simplicial complex triangulating M and for each vertex $v \in \Sigma$ there is a triplet (f_v, g_v, b_v). f, g are piecewise linear Morse functions whose non-critical level sets are piecewise linear 1-manifolds. The restrictions of f and g to each others' non-critical level sets are piecewise linearly Morse functions.

(a) (b)

Fig. 5. (a) A minimal Morse function on a 1-manifold with boundary will have a maximum and a minimum on the boundary and will behave strictly monotonically between these two points. (b) On a closed 1-manifold the maximum and minimum of the minimal Morse function partition the 1-manifold into two topologically equivalent sections, along which the function behaves strictly monotonically.

Theorem 1 provides a formal characterization of the properties that f, g, and b must satisfy to guarantee a bijective mapping. It is useful to consider the following examples to clarify the result. In one dimension a minimal Morse function, f, on a 1-manifold M with boundary has a maximum and a minimum that lie on the boundary. When M is closed, the maximum and minimum of f partition M into two regions, along which f is strictly monotonic, requiring a front/back bit to disambiguate between these two regions, see Fig. 5.

Assuming the functions f and g form a Morse field, the 2-dimensional case is a natural extension of the 1-dimensional case. In a Morse field the generic level sets of f and g are $1-$manifolds along which the restriction of the other function is a minimal Morse function. Just as the critical points of a single function partition a closed 1-manifold in 1-dimension, the Jacobi set of a Morse field defined on a closed surface mesh M partitions M into two topologically equivalent regions. We call the Jacobi set of a Morse field a *minimal* Jacobi set. When M is of genus γ a minimal Jacobi set consists of $\gamma + 1$ closed loops, see Fig. 6. There is one loop for each tunnel in M and one loop along the perimeter or silhouette of M. In a mesh of genus $\gamma > 0$ each of these loops is *non-trivial*, meaning it cannot be contracted to a point.

On the basis of the conditions of Theorem 1, we have designed a surface mapping algorithm with the following four properties:

Feature Selection We introduce both landmark tunnels and the constrained silhouette as user-selectable features in the map. These two concepts are motivated by minimal Jacobi sets.

Computing Paths Jacobi sets are one-manifolds. Therefore, we guarantee that no loops touch when computing paths on the mesh.

Genus Reduction Theorem 1 considers Morse fields with the same image. This implies that genus reduction may be necessary during the construction of the map.

Bijective Mapping Theorem 1 specifies the requirements that f, g and b must satisfy in order to provide a bijective mapping from M to M'. However, it does not imply a method to construct such global functions. To create the final bijective mapping, we find a common base decomposition and compute maps locally on each disk.

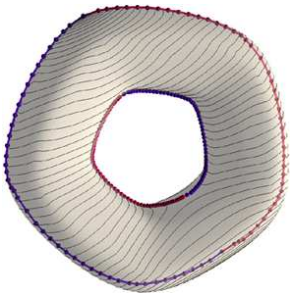


Fig. 6. A minimal Jacobi set partitions M into two topologically equivalent regions. In this image red denotes maximal values and blue denotes minimal values on a level set component.

5 Robust Cross Parameterization

5.1 Algorithm Overview

To construct a cross parameterization between two input meshes, M and M', the user first orients the two meshes on screen and the physical location of each vertex is reassigned according to this orientation. This provides an initial alignment for the map and eliminates unnecessary self intersections that can occur when the "upper" part of M is mapped to the "lower" part of M'. Next, the user selects landmark features: vertices, tunnels, or constrained silhouettes. To specify landmark tunnels and constrained silhouettes, the user selects from a series of automatically generated loops on the mesh. All specified landmark features are used to construct a common base domain B across which M and M' are each parameterized. Prior to computing B we reduce the genus of both M and M' by cutting along the non-landmark tunnel loops and patching the resulting holes with triangulated disks. A metamesh containing attribute information for M and M' is generated after embedding both input meshes in B. A base face $F \in B$ may contain one or more loops associated with non-landmark tunnels (belonging to either M or M'). We describe the evolution of non-landmark tunnels in each such face parametrically using a variational implicit function ψ, guaranteeing a robust evolution of tunnels regardless of their respective geometry.

5.2 Computing Paths

Several of the steps in our approach require paths to be computed along an input mesh between two vertices, v_p and v_c. In many cases, the paths are restricted to lie within a subset of the simplices of the original mesh. Moreover, we require that none of the computed paths touch any other paths (except at path endpoints). In coarsely triangulated regions of the mesh these restrictions

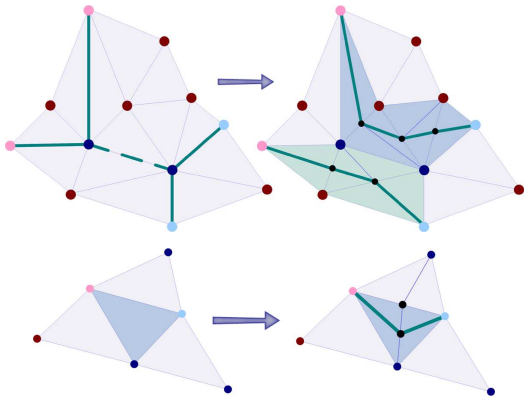


Fig. 7. Top Row: In a coarse triangulation the possible edge paths computed using Dijkstra's algorithm between two different pairs of vertices can intersect (the green dashed edge). To address this issue face paths are computed between source and target vertices, from which a new edge path is extracted. Bottom row: Splitting a face path of length one.

may over constrain the problem when using Dijkstra's algorithm along mesh edges, see the top row in Fig. 7.

To address this we compute paths between vertices by conceptually running Dijkstra's algorithm on the dual mesh. A shortest face path, F_P is first computed between v_p and v_c and then an edge path is extracted from F_P by splitting mesh edges that are incident on two faces in F_P. If a face path is of length one we split the face as illustrated in the bottom row of Fig. 7.

Remark 1 *Calculating paths in this manner guarantees that regardless of the coarseness of the triangulation we are able to compute paths that do not touch any other paths (except at path endpoints).*

5.3 Feature Selection

Landmark features can be one of three types: landmark vertices, landmark tunnels, or constrained silhouettes. Landmark vertices are a fundamental part of all prior cross parameterization algorithms and are hard constraints that the final mapping must maintain. The user specifies landmark vertices by associating a vertex from M and and a vertex from M' with a vertex in B.

We introduce the notions of landmark tunnels and the constrained silhouette which contribute both vertices and edges to B. These two concepts are motivated by the minimal Jacobi set, which for a mesh of genus γ contains $\gamma + 1$ non-trivial loops. These include the γ non-trivial loops associated with mesh tunnels and an additional non-trivial loop along the silhouette of the model.

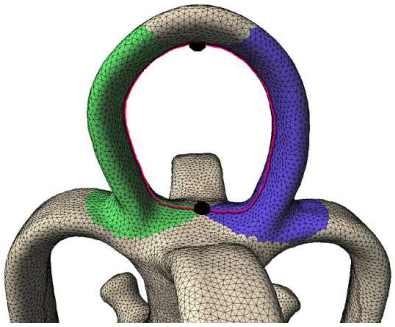


Fig. 8. Shown in red, the non-trivial loop associated with a tunnel is comprised of paths traced between the associated parent and child saddles in each of the tunnel's components (shown in blue and green).

5.3.1 Landmark Tunnels

For every tunnel in a mesh we automatically compute a single non-trivial loop using tools from Morse theory. The user provides an initial coarse alignment of M and M' via screen space orientation and a fair Morse function (Ni et al., 2004) is generated for each mesh using the screen height minimum and maximum of the models as fixed input values. We employ Simulation of Simplicity (SoS) (Edelsbrunner and Mucke, 1988) and split all saddles with degree greater than 2 as described in (Edelsbrunner et al., 2003).

We determine the parent/child critical point pair for each tunnel in M by computing the Reeb graph using the approach of (Cole-McLaughlin et al., 2003) and the extended persistence algorithm of (Agarwal et al., 2006). During Reeb graph construction we build a list for each face in M of the Reeb graph components it spans.

For each tunnel, the loop that is computed is comprised of two paths between the associated parent and child saddles; one in each of the tunnel's components. Paths are computed using the method described in section 5.2. The domain of each face path is restricted to faces that span the tunnel's associated component path, see Fig. 8. Pairs of paths are computed in increasing order of persistence and component labels are merged in M as we proceed.

The user selects α of the pre-computed loops from each mesh to identify the tunnels that will remain as features in the map. When a loop is constructed its homotopy type is dependent on the function used to compute the Reeb graph: a loop either walks "around" a tunnel or around the complementary handle, see Fig. 9. When the user prefers a loop of a different homotopy type than that which was automatically generated, they can either "toggle" the loop or can specify an alternate non-trivial loop directly on M or M' manually. To toggle a loop l, the user selects l and a vertex v_l along l. The toggled loop is the edge path that is extracted from the shortest face path connecting v_l to

v_l such that:

(1) The face path does not touch l at any point other than v_l.
(2) The face path begins and ends with faces that are on opposite sides of l.

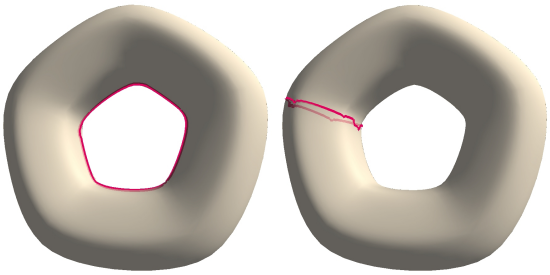


Fig. 9. Non-trivial loops can belong to one of two homotopy types. Shown in red, the loop on the left walks "around" the tunnel and the loop on the right walks around the complementary handle.

After selecting a loop the user must identify n, $n \geq 3$, vertices along the loop. These are landmark vertices that partition the loop into n distinct edge paths, each of which corresponds to an edge in B.

5.3.2 Constrained Silhouette

In addition to landmark tunnels, a *constrained silhouette* can be selected as a feature in the map. A constrained silhouette is computed using the Jacobi set of the screen space coordinate functions, x and y of the user-oriented input meshes.

Using the approach of (Edelsbrunner and Harer, 2002), we obtain a series of closed loops on the mesh, one of which contains global functional extrema. From these closed loops we construct a constrained silhouette that satisfies the following properties:

(1) It contains the global extremal vertices of each function: $x_{min}, x_{max}, y_{min}$, and y_{max}.
(2) It is a 1-manifold.
(3) It does not touch any tunnel loops.

Screen space coordinate functions rarely form a Morse field and, as a result, their Jacobi set may be non-manifold and/or may "wind" extraneously along the mesh, see Fig. 10.

To generate a constrained silhouette that satisfies the necessary conditions, we calculate four distinct shortest paths: x_{min} to y_{max}, y_{max} to x_{max}, x_{max} to y_{min}, and y_{min} to x_{min}. We label the set of all mesh edges as E, and the set of edges belonging to a tunnel loop are labeled E_T. All edges belonging to the Jacobi

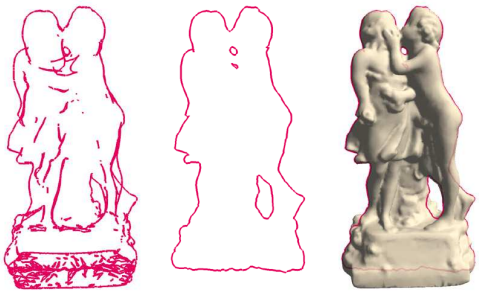


Fig. 10. The Jacobi set of a statue (left) winds and is non-manifold. The constrained silhouette (center/right) does not touch any of the tunnel loops and is a 1-manifold.

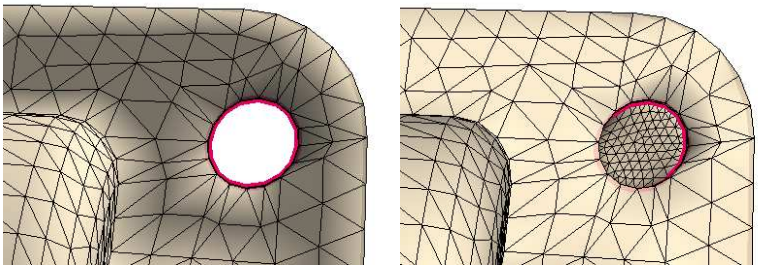
set and the neighbors of those edges are labeled E_J. To guarantee robustness against coarse triangulations we compute the constrained silhouette paths as described in section 5.2. We first attempt to find the shortest face path whose coincident edges belong to $E_J \setminus E_T$. When this is not possible we find the shortest face path with coincident edges belonging to $E \setminus E_T$ (this is always possible as all non-trivial loops are non-separating). The constrained silhouette consists of the four edge paths extracted from these face paths.

If selected as a feature, the constrained silhouette partitions the mesh into two regions (front and back) and thus the remaining landmark vertices and landmark tunnel loops are required to belong to the same region(s) in both M and M'.

5.4 Base Mesh

Prior to generating a common base mesh B, we reduce the genus of both M and M' using their respective non-landmark tunnel loops. Cutting a mesh along a non-trivial loop creates two "holes", each of which is filled with a triangulated disk that is proportional in size to the number of vertices in the loop, see Fig. 11. Combinatorially, the triangulated disk consists of a series of concentric rings of vertices that are connected by annuli of faces. The number of vertices in each ring decreases by n (a user specified parameter) and the positions of vertices in the interior of each disk are assigned by solving a linear system of equations using the positions of the loop vertices as fixed input values.

A common base mesh B, see Fig. 12, is computed by tracing consistent pairs of paths between landmark vertices in M and M' that correspond to edges in B. At this stage in the algorithm M and M' are now of equal genus, α. To guarantee that M and M' are decomposed into sets of topologically equivalent disks, the topology of the path networks in M and M' must be equivalent. It is observed in (Schreiner et al., 2004) that the path networks should contain both a minimum spanning tree of the landmark vertices and 2α non-trivial



(a) (b)

Fig. 11. (a) A mesh with a non landmark tunnel loop highlighted in red. (b) The mesh is cut along the non-landmark tunnel loop and the resulting holes are filled with triangulated disks.

loops, prior to including any paths that form trivial loops on either mesh. We note that, by cutting the mesh along each of the α landmark tunnel loops, we introduce 2α non-trivial loops on the mesh satisfying the second portion of this requirement (see Fig. 13 (a)).

Similarly to (Schreiner et al., 2004; Kraevoy and Sheffer, 2004) we initially compute all possible pairs of face paths between landmark vertices in M and M'. However, in our algorithm base mesh edges may already exist due to landmark tunnels and/or the constrained silhouette and it is unnecessary to recompute the face paths between the vertices in these features. To guarantee that each face in B is a topological disk, we iteratively introduce a new edge e in B until B is triangulated. The edges are added in a greedy fashion based on shortest combined path lengths in M and M' while ensuring that a minimum spanning tree of the landmark vertices is generated prior to introducing any trivial loops in either M or M'.

The paths are computed as described in section 5.2 and are not allowed to touch mesh edges already embedded in base edges of B nor are they allowed to touch non-landmark tunnel loop edges. As each e is added to B a sweep of the mesh faces in M and M' is performed to guarantee that the landmark vertices contained in the new partitions of M and M' are equivalent. Path pairs that do not partition M and M' in an equivalent manner are disregarded.

When adding an edge e, the face path in M associated with e may need to be recomputed in order to maintain the cyclic ordering of edge paths at landmark vertices. The edge paths incident on a landmark vertex v decompose the region around v into sectors. When the face paths associated with e in M and M' do not begin/end in the same sectors we recompute the path in M forcing the face path's beginning and ending sectors to agree with the beginning and ending sectors of the corresponding face path in M'.

After a triangulation has been computed, the landmark tunnel loops are merged within B, M, and M', see Fig. 13. Merging tunnels can cause a single

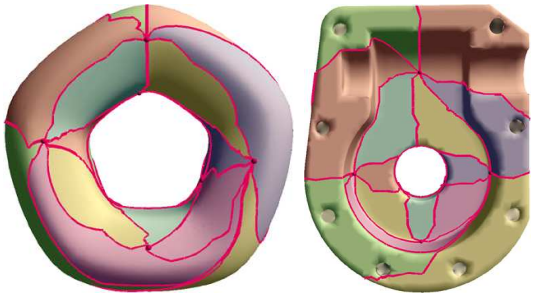
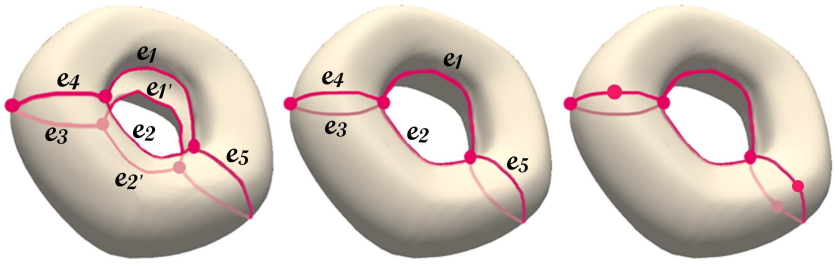


Fig. 12. Base mesh decomposition of a torus (left) and a mechanical part originally of genus 9 (right). The central tunnel in the mechanical part and the tunnel in the torus have been specified as landmark tunnels. The remaining 8 tunnels in the mechanical part have been filled with triangulated disks.

edge in B to form a closed loop. Also, multiple edges in B may now share the same endpoints. Additional landmark vertices are automatically inserted in edges that fall into either of these categories and additional edge paths are computed to complete the final base triangulation. The remaining vertices of M and M' are embedded in B using a traditional surface parameterization technique (Floater, 1997) on each base mesh face.

5.5 Metamesh

A metamesh contains the attribute information associated with both source and target meshes. Traditional metamesh construction (Lee et al., 1999) overlays M and M' creating a super-mesh with the connectivity information from both input meshes. There are two drawbacks to this approach: the output is often an order of magnitude greater in size than the input meshes and the final metamesh may contain a large number of poorly shaped triangles.



(a) (b) (c)

Fig. 13. (a) M is cut open along a landmark tunnel loop creating 2 loops on the mesh: the loop formed by edges e_1 and e_2 and the loop formed by edges $e_{1'}$ and $e_{2'}$. Edges e_3, e_4, and e_5 are computed while M is cut. (b) The landmark tunnel loop is merged by gluing edges e_1 and $e_{1'}$ and e_2 and $e_{2'}$ back together. Now e_5 forms a single loop and edges e_3 and e_4 share the same endpoints. (c) New vertices are automatically added along e_4 and e_5 to avoid problems with the base triangulation.

To achieve a higher quality metamesh we re-mesh each base face individually. For each base face $F \in B$ our re-mesh algorithm consist of three steps:

- Compute combinatorial structure of F.
- Compute parametric locations of all vertices $v \in F$.
- Compute physical locations of all vertices $v \in F$ with respect to M and M'.

Combinatorially, each face patch F is comprised of a series of concentric rings of vertices connected by strips of faces. The number of vertices in each ring increases by a user-defined parameter n until the total vertices in the patch is equal to the greater of the two interior face vertex counts associated with M and M'. To connect all face patches, edge paths are created for each edge e in B (with vertex count equal to the greater of the vertex counts associated with e in M and M'). The edge paths corresponding to edges of a face $F \in B$ form a ring of vertices R_F that can be connected to the boundary of the face patch of F via an annulus of faces. The parametric locations of the vertices in each patch F are computed by fixing the locations of the vertices in R_F and using a traditional surface parameterization technique (Floater, 1997) to solve for positions of vertices on the interior of F. The physical location for each vertex v is computed with respect to both M and M' using the barycentric coordinates of the faces in M and M' that contain v parametrically.

5.6 Variational Implicit Functions

Topological changes in the cross parameterization are described using variational implicit functions, ψ. For each face $F \in B$ containing at least one non-landmark tunnel loop, ψ_F is derived automatically using the parametric descriptions of M and M'. Each ψ_F is a 3-dimensional variational function whose zero level set describes the evolution of a 2-dimensional source curve to a 2-dimensional target curve over time. We define ψ_F parametrically to guarantee that the deformation is robust regardless of mesh geometry.

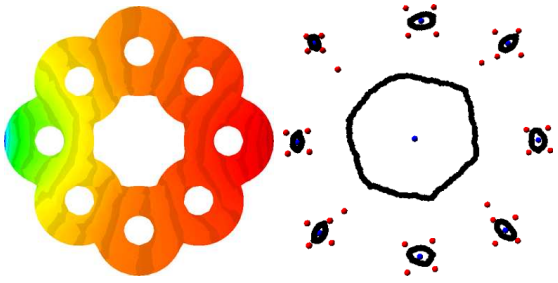


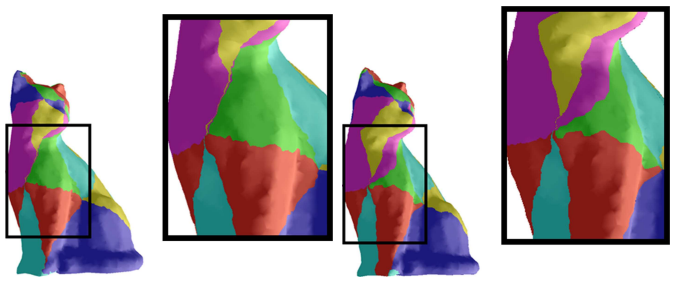
Fig. 14. Negative normal constraints are placed at the center of the triangulated disk (blue). Boundary constraints (black) are placed at each non-landmark tunnel loop vertex. Four positive normal constraints (red), are placed around each loop.

M is positioned at time 0 and M' is positioned at a distance that is equal to twice the "parametric radius" of the largest non-landmark tunnel loop in F. The parametric radius of a loop l is defined to be the longest straight line parametric distance between a vertex in l and the center of the triangulated disk associated with l. This positioning of time planes guarantees that every non-landmark tunnel has adequate time to "close" during the map. The position of the M' time plane can be adjusted to achieve various effects when F contains non-landmark tunnel loops associated with both M and M'. Specifically, the proximity of the time planes (in conjunction with the parametric location of the tunnels) determines whether tunnels from M and M' will merge together or whether tunnels in M will "close" entirely prior to the "opening" of tunnels in M'.

We place positive normal constraints at the base vertices of F in the time planes associated with both M and M'. This guarantees a valid ψ_F is computed regardless of differing mesh genus. Boundary constraints are positioned at all vertices belonging to a non-landmark tunnel loop and a negative normal constraint is placed at the centers of the triangulated disks associated with each tunnel. Four additional positive constraints are positioned around each embedded non-landmark tunnel loop (taking into consideration the position of the other loops to guarantee conflict-free placement). Fig. 14 shows the constraint locations for a 9-handle torus.

6 Optimizations

In an effort to improve the visual quality of the map, we optimize in two ways prior to finalizing the metamesh: edge path optimization and vertex optimization.



(a) (b)

Fig. 15. (a) Upon computing the initial base triangulation of a cat dataset the pink and yellow base faces are poorly shaped (i.e. portions of the triangles are nearly degenerate). (b) After edge path optimization the shapes of the pink and yellow base faces are much improved.

6.1 Edge Path Optimization

The parametric distortion associated with a base face F increases when the embedding of F is not well shaped (e.g. long skinny triangles). Therefore, after computing the base mesh B we optimize all edge paths in both M and M' in an effort to improve the shape of the embedding of B in each input mesh.

For a given base mesh edge $e \in B$ with end points v_a and v_b, there are two incident base faces F_1 and F_2. The quadrilateral region formed by F_1 and F_2 is re-mapped to the plane using a traditional surface parameterization technique (Floater, 1997). The original edge path associated with e is replaced with a new shortest path from v_a to v_b that is computed using Dijkstra's algorithm in the parametric domain. The face sets associated with F_1 and F_2 are updated and their respective embeddings in B are recomputed.

The edges are iteratively optimized in both M and M'. Due to the manner in which the paths are modified, the topology of the path networks remains constant, however, the shape of the base faces improves reducing parametric distortion, see Fig. 15.

6.2 Vertex Optimization

In addition to edge path optimization we perform iterative optimization of mesh vertices. The parametric location of the vertices are optimized in a series of passes, first relaxing the positions of M followed by a relaxation of the vertices in M'. This process allows the parameterizations to "slide" with respect to each other in an effort to improve the overall stretch efficiency of the map. A mesh vertex is allowed to move to a position within the kernel of its one ring that reduces the symmetrized stretch (Schreiner et al., 2004) between the two meshes:

$$L^2(T) = \frac{A_{T'} A_M}{A_{M'}^2} \left(\frac{1}{\gamma^2} + \frac{1}{\Gamma^2} \right) + \frac{A_T A_{M'}}{A_M^2} \left(\gamma^2 + \Gamma^2 \right)$$

Here A_T is the surface area of a face T with respect to M, $A_{T'}$ is the surface area of T with respect to M', A_M and $A_{M'}$ are the surface areas of M and M' respectively and γ and Γ are the singular values of the map between M and M' for face T.

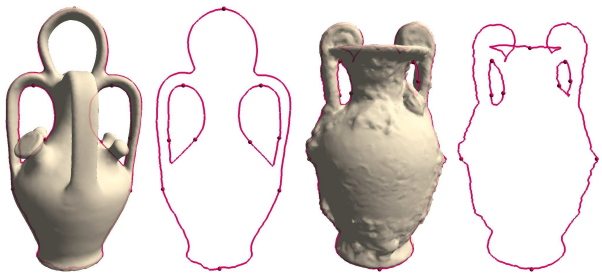
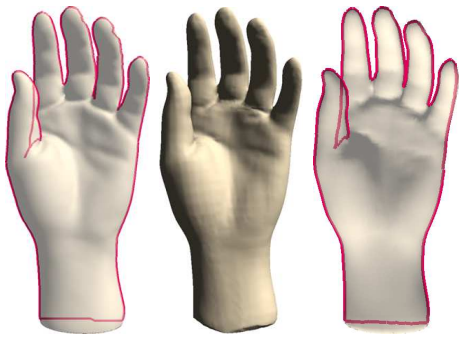


Fig. 16. Ten landmark vertices are associated with the botijo and amphora meshes: 4 along the constrained silhouette and 3 per landmark tunnel loop.

7 Results

We demonstrate the results of our cross parameterization scheme by morphing between meshes of differing genus. The embedding of the surface at each stage in the morph from M to M' is computed as a simple linear interpolation of corresponding vertex positions between M and M'. In some cases self-intersections in the morph sequence are entirely unavoidable without temporarily cutting the mesh (e.g. a morph between a knotted torus and torus). While the design of a good interpolation scheme is not the focus of this paper, one could use the approach of (Kilian et al., 2007) to minimize the number of self intersections in the morph. Topological transitions are handled by evaluating the variational implicit function value of each metamesh vertex and clipping out those vertices with negative function values.

Fig. 19 shows the results of our approach on a variety of data sets. Our initial alignment scheme provides a useful mechanism to control the visual quality of the map using landmark tunnels and the constrained silhouette. In Fig. 19(a)-(d) the constrained silhouette has been specified as a feature with 4 landmark vertices placed along each silhouette. Additionally, a single landmark tunnel



(a) (b) (c)

Fig. 17. The full resolution hand in (a) has 38218 vertices and is genus 1. In (c) a simplified version of the hand is shown with 8808 vertices and genus 0. In (b) we show a 50% morph. The constrained silhouette is shown in red in (a) and (c).

(with 3 landmark vertices/tunnel) has been specified for the meshes in Fig. 19(a) and Fig. 19(b). Whereas previous techniques are unable to generate maps between meshes with convoluted geometry, our approach is robust and successfully produces a map, see Fig. 19(d).

Fig. 16 shows the landmarks used to generate a map between two vases of unequal genus: the botijo (genus 5) and amphora (genus 2). Two landmark tunnels in each mesh are selected with three landmark vertices assigned per tunnel. The constrained silhouette is also specified as a landmark, along which 4 landmark vertices are positioned. The resulting morph sequence is shown in Fig. 19(e). Using the variational implicit approach, a smooth deformation is automatically calculated for the non-landmark tunnels.

In our experiments we have found that constrained silhouettes are a very useful tool when (1) they contain the same desired landmarks and (2) a traversal of the two silhouettes encounters the landmark pairs in the same order. This is the case in the maps in Fig. 19(a)-(e). This is also the case in Fig. 17. A hand data set at full resolution is shown in Fig. 17(a). During model acquisition a small tunnel was introduced between the pointer and middle fingers. A simplified version of the model that is genus 0 is shown in Fig. 17(c). The constrained silhouettes, shown in red, are specified as landmarks and our algorithm produces a natural map between the two meshes. Fig. 17(b) shows the morph at 50%. Fig. 18 illustrates a case where the constrained silhouettes would not be selected as a landmark in the map. The constrained silhouette of the horse contains all 4 legs, while that of the feline contains only two of the legs. The map in Fig. 19(f) was created without constrained silhouettes by specifying 15 landmark vertices on each mesh.

We show in Fig. 19 (g) that our algorithm is robust and can handle meshes of high genus by mapping between the dragon of genus 46 and the Asian dragon of genus 0.

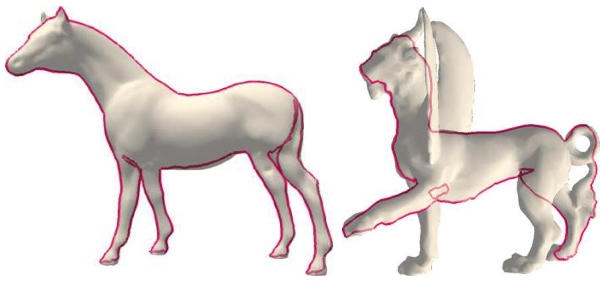


Fig. 18. The constrained silhouettes of the horse (left) and feline (right) data sets are shown in red.

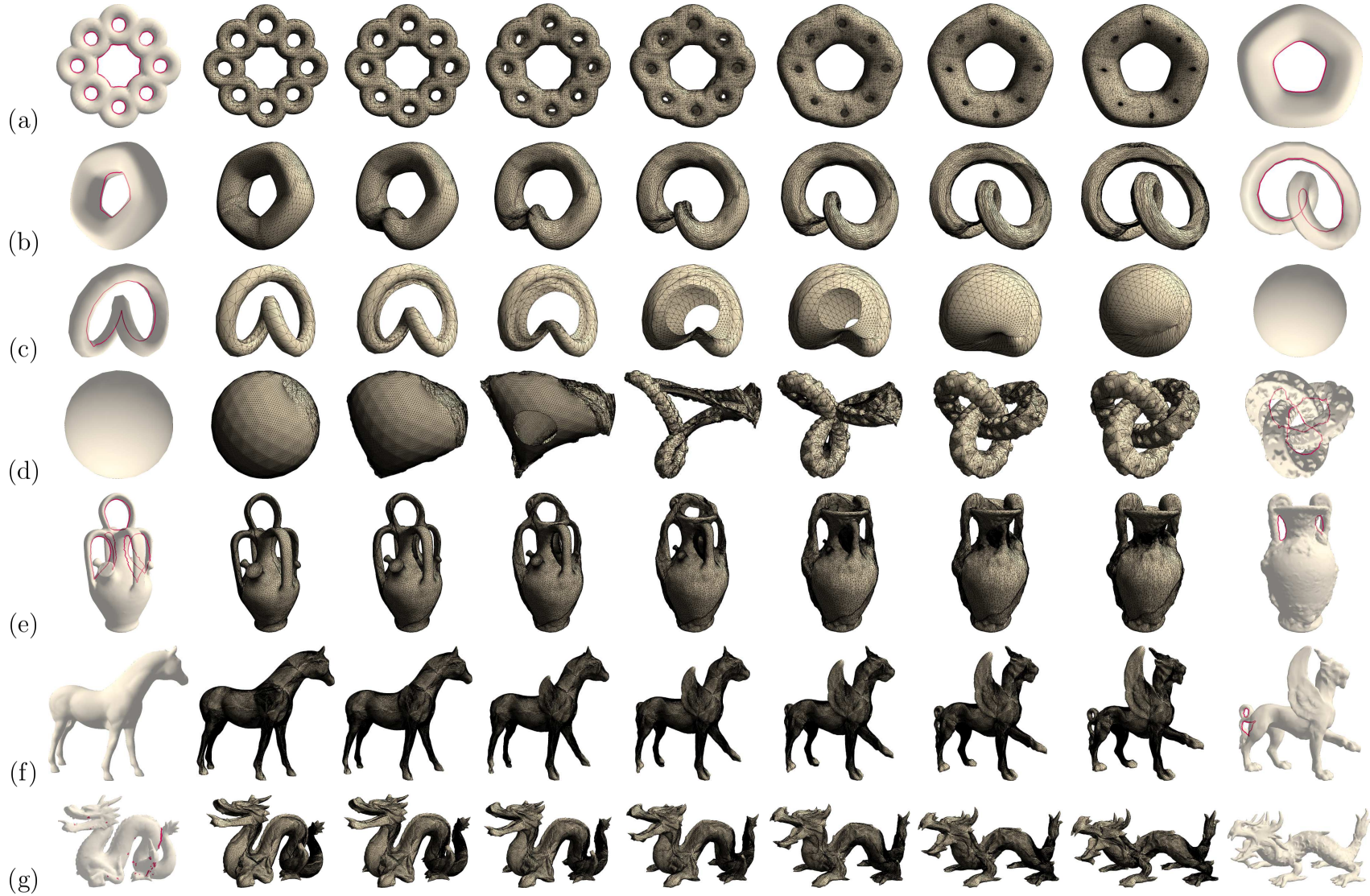


Fig. 19. A series of maps demonstrating the effectiveness of our scheme. Non-trivial tunnel loops are shown in red in the images on the far left and right of each row. (a) Genus 9 torus to genus 1 torus. (b) Genus 1 torus to genus 1 torus. (c) Genus 1 torus to genus 0 sphere. (d) Genus 0 sphere to genus 1 knot. (e) Genus 5 botijo to genus 2 amphora. (f) Genus 0 horse to genus 2 feline. (g) Genus 46 dragon to genus 0 Asian dragon.

8 Conclusion and Future Work

In this work we have used Morse theory to develop a formalism to analyze the correctness of a parameterization scheme. Furthermore, we have created a provably robust method to generate maps between meshes of arbitrary and differing genus. Our approach is motivated by the formalism and is entirely parametric, guaranteeing a successful map even when the geometry of the input is overly convoluted. We introduce a novel alignment scheme that allows the user to specify landmark tunnels and landmark constrained silhouettes in addition to standard landmark vertices. Moreover, our algorithm has no problem processing coarse meshes with high genus.

We have focused on the formal properties and generality of the procedure that constructs a map between two meshes of unequal genus. Our future plans involve optimizing the visual quality of the map. Specifically, when the geometry of M and M' differs greatly, the use of linear interpolation can cause self-intersections and unnatural shape transitions. While the optimization techniques in section 6 do improve the visual quality of the map, they are limited when the overall difference in geometry is too great. We plan on investigating alternate interpolation schemes that could provide a more intuitive looking map in these cases. Additionally, we plan to explore alternate base mesh generation algorithms that take into consideration the shapes of both M and M'.

When tracing shortest paths on the mesh, we compute a shortest face path and refine the mesh along the path to obtain a final edge path. While this approach guarantees robustness against coarse triangulations, it can increase the size of the mesh unnecessarily in triangulations that are dense enough to support traditional Dijkstra edge path computations. We plan to include a hybrid path generation approach that accounts for both dense and coarse regions in the mesh.

While our algorithm robustly handles maps between meshes of differing genus, our metamesh generation code is un-optimized and is the dominating factor in our timings. Running the algorithm from start to finish (including user interaction times) takes around 4-5 hours when generating metameshes with $\sim$790K faces, while metameshes with $\sim$55K faces take around 15-20 minutes. We aim to address performance issues in future work.

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory in part under Contract W-7405-Eng-48 and in part under Contract DE-AC52-07NA27344. The models used in this paper were provided by the AIM@SHAPE project.

References

C. Gotsman, X. Gu, A. Sheffer, Fundamentals of spherical parametrization for 3d meshes, in: ACM Transactions on Graphics, Vol. 22, SIGGRAPH 2003, 2003, pp. 358–363.

E. Praun, H. Hoppe, Spherical parametrization and remeshing, ACM Transactions on Graphics 22 (3) (2003) 340–349.

A. Sheffer, C. Gotsman, N. Dyn, Robust spherical parameterization of triangular meshes, Computing 72 (2004) 185–193.

I. Friedel, P. Schröder, M. Desbrun, Unconstrained spherical parameterization, in: SIGGRAPH '05: ACM SIGGRAPH 2005 Sketches, ACM Press, New York, NY, USA, 2005, p. 134.

X. Gu, S. J. Gortler, H. Hoppe, Geometry images, ACM Transactions on Graphics 21 (3) (2002) 355–361.

C. Carner, M. Jin, X. Gu, H. Qin, Topology-driven surface mappings with robust feature alignment, in: IEEE Visualization, 2005, p. 69.
URL http://doi.ieeecomputersociety.org/10.1109/VIS.2005.107

A. Khodakovsky, N. Litke, P. Schröder, Globally smooth parameterizations with low distortion, ACM Trans. Graph 22 (3) (2003) 350–357.
URL http://doi.acm.org/10.1145/882262.882275

V. Kraevoy, A. Sheffer, Cross-parameterization and compatible remeshing of 3D models, ACM Transactions on Graphics 23 (3) (2004) 861–869.

J. Schreiner, A. Asirvatham, E. Praun, H. Hoppe, Inter-surface mapping, ACM Trans. Graph 23 (3) (2004) 870–877.
URL http://doi.acm.org/10.1145/1015812

X. Gu, S.-T. Yau, Global conformal surface parameterization, ACM Symposium on Geometry Processing (2003) 127–137.

M. Jin, F. Luo, X. Gu, Computing surface hyperbolic structure and real projective structure, Solid and Physics Modeling (2006) 613–622.

T.-Y. Lee, C.-Y. Yao, H.-K. Chu, M.-J. Tai, C.-C. Chen, Generating genus- n -to- m mesh morphing using spherical parameterization, Journal of Visualization and Computer Animation 17 (3-4) (2006) 433–443.
URL http://dx.doi.org/10.1002/cav.146

M. S. Floater, K. Hormann, Surface parameterization: a tutorial and survey, Advances in Multiresolution for Geometric Modelling (2004) 157–186.

A. Lee, D. Dobkin, W. Sweldens, P. Schröder, Multiresolution mesh morphing, in: A. Rockwood (Ed.), Proceedings of the Conference on Computer Graphics (Siggraph99), ACM Press, N.Y., 1999, pp. 343–350.

C. Lin, T. Lee, Metamorphosis of 3d polyhedral models using progressive connectivity transformations, IEEE Transactions on Visualization and Computer Graphics 11 (1) (2005) 2–12.

X. Li, Y. Bao, X. Guo, M. Jin, X. Gu, H. Qin, Globally optimal surface mapping for surfaces with arbitrary topology, IEEE Transactions on Visualization and Computer Graphics 14 (4) (2008) 805–89.

M. Jin, J. Kim, X. Gu, Discrete surface ricci flow: Theory and applications, in: Lecture Notes in Computer Science, Vol. 4647, 2007, pp. 209–232.

D. DeCarlo, J. H. Gallier, Topological evolution of surfaces, in: Graphics Interface, 1996, pp. 194–203.

X. Li, X. Gu, H. Qin, Surface matching using consistent pants decomposition, in: ACM Solid and Physical Modeling Symposium, 2008, to appear.

J. Bennett, V. Pascucci, K. Joy, Genus oblivious cross parameterization: Robust topological management of intersurface maps, in: Proceedings of Pacific Graphics 2007, Maui, Hawaii, 2007, pp. 238–247.

A. Hatcher, Pants decompositions of surfacesComment: 12 pages.
URL `http://arxiv.org/abs/math/9906084`

C. de Verdiere, Lazarus, Optimal pants decompositions and shortest homotopic cycles on an orientable surface, JACM: Journal of the ACM 54.

D. F. Wiley, N. Amenta, D. A. Alcantara, D. Ghosh, Y. J. Kil, E. Delson, W. Harcourt-Smith, K. S. John, F. J. Rohlf, B. Hamann, Evolutionary morphing, in: IEEE Visualization, 2005, p. 55.
URL `http://doi.ieeecomputersociety.org/10.1109/VIS.2005.30`

G. Turk, J. F. O'Brien, Shape transformation using variational implicit functions, in: SIGGRAPH, 1999, pp. 335–342.
URL `http://doi.acm.org/10.1145/311535.311580`

X. Ni, M. Garland, J. Hart, Fair morse functions for extracting the topological structure of a surface mesh, ACM Transactions on Graphics 23 (3) (2004) 613–622.

M. Hilaga, Y. Shinagawa, T. Kohmura, T. L. Kunii, Topology matching for fully automatic similarity estimation of 3D shapes, in: SIGGRAPH, 2001, pp. 203–212.

Edelsbrunner, Harer, Mascarenhas, Pascucci, Time-varying reeb graphs for continuous space-time data, in: COMPGEOM: Annual ACM Symposium on Computational Geometry, 2004.

G. Patane, M. Spagnuolo, B. Falcidieno, Para-graph: Graph-based parameterization of triangle meshes with arbitrary genus, Computer Graphics Forum 23 (4) (2004) 783–797.

E. Zhang, K. Mischaikow, G. Turk, Feature-based surface parametrization and texture mapping, ACM Transactions On Graphics 24 (1) (2005) 1–27.

Q.-Y. Zhou, T. Ju, S.-M. Hu, Topology repair of solid models using skeletons, IEEE Transactions on Visualization and Computer Graphics.

Z. Wood, H. Hoppe, M. Desbrun, P. Schröder, Removing excess topology from isosurfaces, ACM Trans. Graph. 23 (2) (2004) 190–208.

F. Nooruddin, G. Turk, Simplification and repair of polygonal models using volumetric techniques (1999).
URL citeseer.ist.psu.edu/nooruddin03simplification.html

H. Edelsbrunner, D. Letscher, A. Zomorodian, Topological persistence and simplification, Discrete Computational Geometry (28) (2002) 511–533.

A. Gyulassy, V. Natarajan, V. Pascucci, P.-T. Bremer, B. Hamann, Topology-based simplification for feature extraction from 3D scalar fields, in: IEEE Visualization, IEEE Computer Society, 2005, p. 68.
URL http://doi.ieeecomputersociety.org/10.1109/VIS.2005.106

P.-T. Bremer, H. Edelsbrunner, B. Hamann, V. Pascucci, A multi-resolution data structure for two-dimensional morse functions (May 17 2004).
URL http://www.osti.gov/servlets/purl/15005010-lAYRC4/native/

H. Edelsbrunner, J. Harer, Jacobi sets of multiple morse functions, in: Foundations of Computational Mathematics, Cambridge Univ. Press, 2002, pp. 37–57.

H. Edelsbrunner, J. Harer, V. Natarajan, V. Pascucci, Local and global comparison of continuous functions, in: IEEE Visualization, IEEE Computer Society, 2004, pp. 275–280.

J. Milnor, Morse Theory, Vol. 51 of Annals of mathematics studies, Princeton University Press, Princeton, 1963.

Y. Matsumoto, An Introduction to Morse Theory, Vol. 208 of Translations of Mathematical Monographs, American Mathematical Society, Providence, RI, 2002.

H. Edelsbrunner, E. P. Mucke, Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms, in: Symposium on Computational Geometry, 1988, pp. 118–133.
URL citeseer.ist.psu.edu/edelsbrunner90simulation.html

K. Cole-McLaughlin, H. Edelsbrunner, J. Harer, V. Natarajan, V. Pascucci, Loops in reeb graphs of 2-manifolds, in: Proceedings of the 19-th ACM Symposium on Computational Geometry (SoCG), 2003, pp. 344–350.

P. K. Agarwal, H. Edelsbrunner, J. Harer, Y. Wang, Extreme elevation on a 2-manifold, Discrete Computational Geometry (36) (2006) 553–572.

H. Edelsbrunner, J. Harer, A. Zomorodian, Hierarchical morse-smale complexes for piecewise linear 2-manifolds, Discrete Computational Geometry (30) (2003) 87–107.

M. S. Floater, Parametrization and smooth approximation of surface triangulations, Computer Aided Geometric Design 14 (1997) 231–250.

M. Kilian, N. J. Mitra, H. Pottmann, Geometric modeling in shape space, in: ACM Trans. Graphics (Proc. SIGGRAPH' 07), Vol. 26, 2007.