Consistent Correspondence between Arbitrary Manifold Surfaces

Huai-Yu Wu, Chunhong Pan, Qing Yang, Songde Ma
National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
huaiyuwu@gmail.com,(chpan,qyang)@nlpr.ia.ac.cn,songde.ma@mail.ia.ac.cn

Abstract

We propose a novel framework for consistent correspondence between arbitrary manifold meshes. Different from most existing methods, our approach directly maps the connectivity of the source mesh onto the target mesh without needing to segment input meshes, thus effectively avoids dealing with unstable extreme conditions (e.g. complex boundaries or high genus). In this paper, firstly, a novel mean-value Laplacian fitting scheme is proposed, which aims at computing a shape-preserving (conformal) correspondence directly in 3D-to-3D space, efficiently avoiding local optimum caused by the nearest-point search, and achieving good results even with only a few marker points. Secondly, we introduce a vertex relocation and projection approach, which refines the initial fitting result in the way of local conformity. Each vertex of the initial result is gradually projected onto the target model's surface to ensure a complete surface match. Furthermore, we provide a fast and effective approach to automatically detect critic points in the context of consistent correspondence. By fitting these critic points that capture the important features of the target mesh, the output compatible mesh matches the target mesh's profiles quite well. Compared with previous approaches, our scheme is robust, fast, and convenient, thus suitable for common applications.

1. Introduction

Establishing consistent correspondence (or mapping) between different shapes is a fundamental task in various applications such as fitting template models to multiple 3D data sets [2], shape blending [8], statistical analysis of shape (e.g. principal component analysis), transferring texture and surface properties (BRDFs, normal maps, etc), surface classification [20] and recognition [19], video tracking [21], and performance driven facial animation (PDFA) etc. As the important first step for these applications, the consistent correspondence technique transforms 3D models (commonly represented by surface meshes) with different connectivity into compatible meshes (i.e. meshes with the same topology) while respecting surface detail features.

Due to the non-Euclidean nature (the lack of a regular Euclidean parameterization domain) of manifold surface, to build a consistent correspondence between mesh surfaces is a difficult task in itself. Existing approaches suffer from several problems: low robustness, slow speed, hard to obtain high-quality compatible mesh that preserves the shape features of original models. All these difficulties greatly reduce this method's usefulness.

In this paper, we propose a new framework to give an elegant solution for establishing high-quality consistent correspondence between arbitrary meshes. Different from most existing methods, which need to segment the input meshes and then construct intermediate parameterization domains, our approach directly uses the connectivity of one mesh to approximate the geometry of another, thus effectively avoids potential difficulties when dealing with various complex conditions such as intersections, blocking, cyclic orders, discontinuity along partition boundaries, etc.

We show that our approach fits nicely in a unified mathematical framework, where the similar type of linear operator is applied in all phases. Firstly, a new mean-value Laplacian fitting scheme is proposed, which computes a shape-preserving correspondence directly in 3D-to-3D space (without partitioning and flattening 3D surfaces onto 2D planes), efficiently avoids local optimum caused by the nearest-point search, and achieves good results with fewer marker points than previous methods.

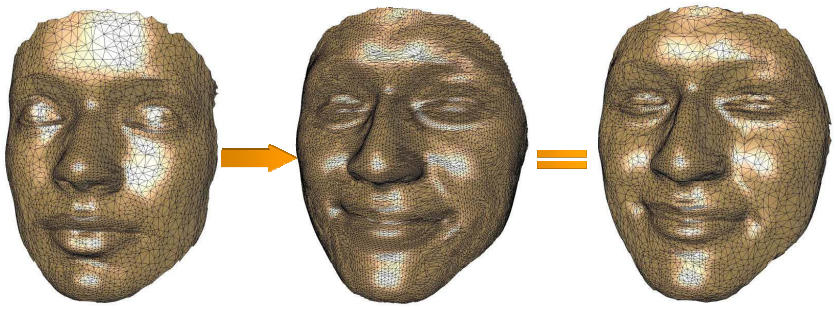
Next, we provide a novel vertex relocation and projection method, which refines the initial fitting result in the way of local conformity (angle preservation) and ensures a complete surface match.

Furthermore, we search for the key factors that will decide the quality of consistent correspondence. A fast and effective approach is introduced to detect critic points and then exploits these vertices to nicely match the target mesh's features and profiles.

1.1. Related Works

In this section, we discuss related work in the area of consistent correspondence (one-to-one mapping) between mesh surfaces [13, 1, 4, 12, 14, 15, 8, 6, 23, 3, 9, 20, 2]. Generally, existing consistent correspondence methods can

978-1-4244-1631-8/07/$25.00 ©2007 IEEE



(a) (b) (c)

Figure 1. Establishing consistent correspondence between input meshes (a) and (b). The result compatible mesh (c) has the same topology (the same number of vertices, the same number of triangles, etc.) with the source mesh (a) while having the similar geometry with the target mesh (b). Specifically, any point in (c) has a semantically one-to-one correspondence (both in geometry and topology) point on the source mesh (a). Thus mesh (c) is called the source mesh's consistent correspondence with the target mesh.

be classified into two categories: whether the consistent correspondence is constructed directly or indirectly.

Most existing consistent correspondence methods belong to indirect schemes. In these methods, an intermediate parameterization domain, which actually consists of a set of 2D convex planar sub-domains (for disk-like surfaces) or a spheral domain (limited to genus zero surfaces), is first constructed. Then, the sub-mapping between each model and the common parameterization domain is built separately. Finally, mapping between models is constructed through sub-mappings' composition. After obtaining the cross-parameterization, the consistent correspondence between meshes can be straightforwardly obtained by interpolating vertices' barycentric coordinates. The difference of various algorithms lies in the type of intermediate domain to be chosen, such as plane [4], sphere [12], cylinder [15], triangle patch [8] etc. Despite the diversity, in fact, the parameterization process of all these domains is eventually carried on the 2D convex planar sub-domains or a spheral domain, due to excellent shape-preservation [4] or conformal property [6] in mathematics. However, for indirect schemes, the difficulty to compatibly construct well-shaped patch layouts makes these methods hard to keep balance between efficiency and robustness, especially in complex-topology and long-and-narrow regions. Moreover, the error of sub-mappings can be amplified through mapping composition, so the final inter-surface mapping may have huge error somewhere. Another downside of indirect schemes is that though continuous within each patch, the mapping may be discontinuous when transiting inter-patch boundaries. As a result, some post-processes, such as *smoothing* [8], are necessary for the final results.

Except building intermediate domains, consistent correspondence can also be constructed directly, i.e. using the target mesh as the common domain, thus avoids explicit cross-parameterization. In [2], by smoothing local affine transformations, Allen et al. use the template fitting technique to directly construct consistent correspondence for a set of human models. Sumner et al. [17] propose a similar iterated closest algorithm to build correspondence map between meshes. However, direct schemes so far don't explicitly take shape preservation property into account, thus will introduce large approximation errors when the input models have significantly different geometry. Furthermore, these methods adopt Euclidean nearest distance as the iterated fitting metric, which is prone to trap into local optimum when input meshes have complex shapes.

To overcome the drawbacks above, our method constructs the shape-preserving correspondence directly in 3D-to-3D space without partitioning and flattening 3D surfaces onto 2D plane (i.e. the common 3D-2D-2D-3D procedure), thus effectively avoids error amplification, discontinuity along partition boundaries, and various tricky conditions. Instead of Euclidean nearest distance, our approach uses gradual approximating metric as the fitting metric to avoid trapping into local optimum. Furthermore, our critical points scheme significantly reduces approximation errors even when the input models have very different geometric features and sample rates.

1.2. Notations and Definitions

Definition 1.2.1 *Manifold surface:* A manifold of dimension n is a connected Hausdorff space M for which any point $p \in M$ has a neighborhood $U \subset M$ which is homeomorphic to an open subset of $\Re^n$ Euclidean space. Specifically, a triangle mesh is manifold surface if:

$\diamond$ There are exactly two faces adjacent to each edge (not including the boundaries).

$\diamond$ The vertices v_i adjacent to p can be ordered v_0,...,v_{n-1} such that the triangles $v_i, p, v_{(i+1) \ mod \ n}$ all exist.

Definition 1.2.2 *Consistent Correspondence:* As shown in Figure 1, the compatible mesh M_C has the same topology with the source mesh M_S while having the similar geometry with the target mesh M_T. Specifically, any point $p_c \in M_c$ has a semantically (both in geometry and topology) one-to-one correspondence point p_s on the source mesh M_S.

Thus mesh M_C is called the source mesh M_S's consistent correspondence with the target mesh M_T.

2. Consistent Correspondence Framework

Our consistent correspondence framework consists of three steps. Firstly, a novel mean-value Laplacian fitting scheme is proposed, which aims at computing a shape-preserving correspondence directly in 3D space, efficiently avoiding local optimum caused by the nearest-point search, and achieving good results even with only a few marker points (Section 2.1). Then, in order to more accurately match the output compatible mesh to the target mesh's important features, we introduce a fast and effective approach to detect critic points and then exploits these vertices to nicely match the target mesh's features and profiles (Section 2.2). Finally, the fitting result is further refined by our vertex relocation and projection approach in the way of local conformity. Each vertex of the initial result is gradually projected onto the target model's surface to ensure a complete surface match (Section 2.3). We show that our approach fits nicely in a unified linear framework in a least squares sense, which can be efficiently minimized by fast solving a sparse linear system.

2.1. Mean-value Laplacian Fitting Scheme

We now describe our mean-value Laplacian fitting technique that builds an initial consistent correspondence between the source mesh M_S (e.g. a template surface) and the target mesh M_T (e.g. a scanned example surface). The initial correspondence result M_C^{init} will have the identical topology (i.e. identical number of vertices and triangles) with the source mesh M_S and the similar geometry with the target mesh M_T.

Our fitting scheme is based on the energy minimization framework. Different from the framework of [2, 17], our scheme aims at obtaining a shape-preserving (conformal) correspondence, and avoiding local optimum in the case of only a few marker points.

Our energy-minimization framework contains three terms: mean-value Laplacian energy item $\mathbf{E}_l$, gradual fitting term $\mathbf{E}_g$, and global constraint item $\mathbf{E}_c$. In the following, we formulate the three items, respectively.

The first one is the mean-value Laplacian energy item $\mathbf{E}_l$. Laplacian linear operator [16] provides an effective differential representation of the mesh. Let $M = (T, G)$ be a manifold triangular mesh. $T = (V, E, F)$ is a graph where V denotes the set of vertices, E denotes the set of edges and F denotes the set of facets. G is the geometry associated with each vertex in V. The ordinary Laplacian differential coordinates ζ_i of vertex v_i are represented by the difference between v_i and the average of its neighbors (see Figure 2):

$$\zeta_i = (\zeta_i^{(x)}, \zeta_i^{(y)}, \zeta_i^{(z)}) = v_i - \frac{1}{d_i} \sum_{j \in N(i)} v_j \qquad (1)$$

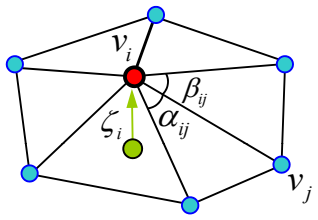


Figure 2. Laplacian differential coordinates (ζ_i), and the angles (α_{ij} and β_{ij}) used in the definition of the mean-value weights.

where $N(i) = \{j|(i, j) \in E\}$ are the edge neighbors, $d_i = |N(i)|$ is the valence of a vertex, i.e. the number of edges which emanate from this vertex.

The Laplacian coordinate is the average difference vector of its adjacent vertices to the vertex v_i. If setting the vector to zero at each vertex, we build a smoothness energy, i.e. smoothly distributing each vertex as close as possible to the barycenter of its immediate neighbors. In matrix form, the formulations for all the vertices can be rewritten as:

$$\mathbf{E}_l = \left\| \mathbf{L}\mathbf{V}^{init} \right\|^2 \qquad (2)$$

where $\mathbf{V}^{init} = [v_1^{init}, v_2^{init}, ..., v_{N(V_S)}^{init}]^T$, $\mathbf{L}$ is the Laplacian coefficient matrix of M_S, i.e. $\mathbf{L}_{mn} = -\delta_{\{m=n\}} + 1/d_i \cdot \delta_{\{(m,n)\in E_S\}}$, δ is the Dirac function.

However, as shown in Figure 3(b), this uniform representation cannot reflect geometric properties of the source mesh M_S. In the following, we try to incorporate information about the original shape, for instance encoding the information about the size, angle and orientation of local surface shape. Here, we consider mean value coordinates [4] for its excellent properties such as shape preservation and low angular distortion.

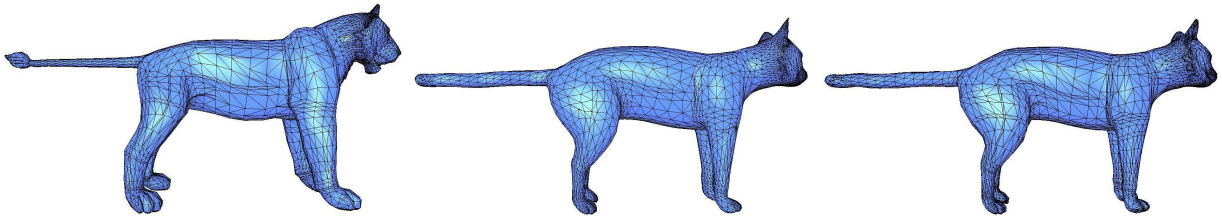
$$\zeta_i = v_i - \frac{1}{\sum_{(i,j)\in E} w_{ij}} \sum_{(i,j)\in E} w_{ij} v_j \qquad (3)$$

$$w_{ij} = \frac{\tan(\alpha_{ij}/2) + \tan(\beta_{ij}/2)}{||v_j - v_i||} \qquad (4)$$

where w_{ij} is mean-value coefficient, α_{ij} and β_{ij} are the angles shown in Figure 2.

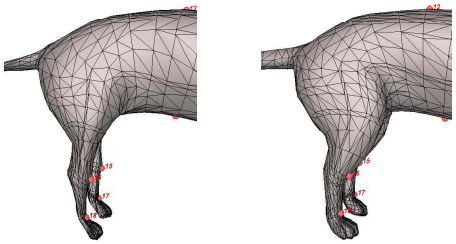
From Figure 3(c), it can be seen that our mean-value Laplacian representation successfully captures the shape information of the source mesh, such as the threadlike stripes on the legs and belly.

The second term is gradual fitting term $\mathbf{E}_g$. According to the definition of consistent correspondence, we wish that mesh M_C has the same geometry with the target mesh M_T, i.e., M_C should be as close as possible to the target surface M_T. However, if only a few of initial markers are provided, existing iterated closest point algorithms may fall into local optimum and introduce large errors (see Figure 4(a)). To overcome this drawback (see Figure 4(b)), instead



(a) (b) (c)

Figure 3. The example of mapping a lion model (a) to a cat. Compared with the uniform representation (b), the mean-value Laplacian representation can obtain a shape-preserving correspondence (c).



(a) (b)

Figure 4. (a) When only a few markers are provided, the iterated closest point algorithm may fall into local optimum. (b) Our fitting scheme effectively overcomes this drawback.

of Euclidean nearest distance, our method adopts *gradual approximating metric* as the fitting measurement.

Now, we introduce the definition of *gradual approximating metric*. A vertex v_i's *gradual approximating metric* $\Uparrow_i$ is defined as:

$$\Uparrow_i = \frac{1}{d_i^n + \alpha} \cdot \frac{area(F(v_i))}{\overline{area}(F(M_S))} = \frac{1}{d_i^n + \alpha} \cdot \frac{\sum_{k=1}^{N(v_i)} area(f_k)}{\sum_{k=1}^{N(M_S)} area(f_k)/N(M_S)} \tag{5}$$

where d_i^n is the normal distance from v_i to the intersection position with the target mesh along v_i's normal n_i (if $d_i^n \to \infty$, we reverse the normal direction); $\alpha = 0.01$ is the laxation factor to avoid ∞ when d_i^n approaches 0; $F(v_i)$ is the 1-ring facets set of v_i, $N(v_i)$ is the number of 1-ring facets; $F(M_S)$ is the facets set of M_S, $N(M_S)$ is the number of facets; M_S's average area $\overline{area}(F(M_S))$ is used for normalizing v_i's 1-ring area $area(F(v_i))$.

It can be seen that the gradual approximating metric is inversely proportional to d_i^n while directly proportional to $area(F(v_i))$. This form has the clear physical meaning: to push an "elastic" mesh onto the target mesh, first, some marker points act as initial global constraints. Then, in order to avoid large distortion, we need to keep imposing a certain tension (mean-value Laplacian quadratic energy) on the whole surface all the time. To obtain a natural mapping, we first spread the regions near the global constraints (with a small d_i^n value), and then gradually extend to distant regions. Furthermore, similar to the coarse-to-fine strategy, we also first spread the large regions (with large areas $area(F(v_i))$) and then deal with small regions, which can achieve better results in our tests.

Thus, gradual fitting term $\mathbf{E}_g$ is formulated as:

$$\mathbf{E}_g = \sum_{i \in G} \Uparrow_i \left| v_i^{init} - g_i \right|^2 \tag{6}$$

where G is the index set of vertices in M_S except global constraint vertices, $\Uparrow_i$ is v_i's gradual approximating metric, g_i is the intersection position with the target mesh along v_i's normal n_i.

The third term focuses on global constraint vertices. Using the $\mathbf{E}_l$ and $\mathbf{E}_g$ would be sufficient if the source and target meshes were initially very close to each other. However, in the common situation, two input models are not close or aligned well, the optimization can become stuck in local minima. To avoid undesirable minima, we need to identify some marker points on both input meshes, which serve as the global constraints, i.e. a priori. So we can make use of these markers to initialize the correspondence, although we show that once enough shapes have been matched, we do not require these markers any more.

The global constraint term $\mathbf{E}_c$ is defined as:

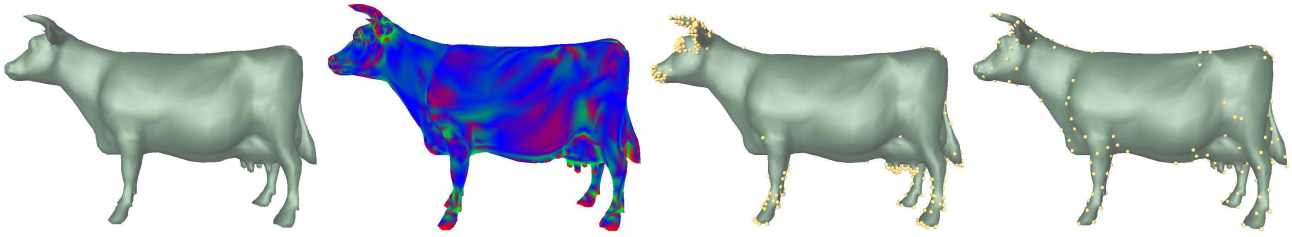
$$\mathbf{E}_c = \sum_{i \in C} \left| v_i^{init} - c_i \right|^2 \tag{7}$$

where C is the index set of global constraint vertices, c_i is the position of correspondent marker on the target mesh.

Our complete objective function $\mathbf{E}$ is the weighted sum of the three error functions:

$$\min \mathbf{E}(\mathbf{V}^{init}) = w_l E_l + w_g E_g + w_c E_c \tag{8}$$

where w_l, w_g, w_c are weights. The quadratic optimization formulation can be minimized by efficiently solving a sparse linear system. And the system is separable in the three coordinates of the vertices, thus reduces linear system's scale to 1/3. We solve the minimization problem in two phases. First, we ignore the gradual fitting term $\mathbf{E}_g$ by using weights $w_l = 1, w_c = 0.3, w_g = 0$, and we obtain an initial mapping result. Then, we increase w_g each



(a) (b) (c) (d)

Figure 5. (b) is the discrete Gaussian curvature contour of the cow model (a). The red color indicates the convex regions, the green color indicates the concave regions, and the blue color indicates the flat regions. (c) shows the salient points obtained by using the method in [5]. (d) shows the salient points obtained by our method. It can be seen that our method not only successfully captures the salient features, but also avoids producing over-dense points, thus is more suitable in the context of consistent correspondence.

time and update M_C^{init}'s vertices. In our experiments, increasing w_g step by step from 0.001 to 0.01 generated good results. Although the marker data is useful for global optimization, we found that the placement of the markers was sometimes unreliable. So we gradually reduce the weight of the global constraints from 0.3 to 0. Each time the optimization problem is solved, M_C^{init} is updated from its original position and more closely approximates M_T. Note that our optimizing system is actually designed to deform the source mesh M_S into the target mesh M_T to produce the compatible mesh M_C^{init}, thus implicitly guarantees that M_C^{init} has the identical connectivity with M_S.

2.2. Critical Vertices

After mean-value Laplacian fitting, the compatible mesh M_C^{init} can approximate M_T quite well in many cases. However, if M_T has complex features not available in M_S, M_C^{init} may miss these high-frequency features, which will cause profile's collapse, as shown in Figure 6(c). In this case, we should improve the initial correspondence result. Previous approaches, such as [8], update it using *adaptive-smoothing* procedures. This scheme has two disadvantages. (1) The smoothing procedure needs to check errors of all input meshes' vertices and edge midpoints, so it is slow and its parameter is hard to set. (2) The scheme cannot accurately locate the key vertices onto the profiles, and thus has to introduce extra vertices much more than necessary.

These disadvantages of previous methods root from the fact that they do not distinguish vertices' different importance, but treat all vertices with the same importance. In our setting, we detect vertices that are important for feature capturing and error reducing, and give them special attention. Furthermore, to reduce local distortion, these points should be proper both in the number and space distribution.

We provide a fast and effective method to detect these critical points. These points' number and space distribution can also be well controlled. In our method, we adopt Gaussian curvature, the important intrinsic property of a surface, as the critical measurement. As shown in Figure 5(b), following human visual perception, high Gaussian curvature (absolute value) obviously reflects important regions of the surface, while the smooth regions (curvature tends to zero) are relatively trivial.

2.2.1 Detect Critical Vertices

According to the Gauss-Bonnet theorem, the total Gaussian curvature of an embedded triangle is expressed in terms of the total geodesic curvature (k_g) of the boundary and the jump angles (α_i) at the corners.

$$\iint_T K dA = 2\pi - \sum \alpha_i - \int_{\partial T} k_g ds \tag{9}$$

[10] provides an approach to compute Gaussian curvature for the discrete surface (e.g. triangle meshes):

$$K(v_i) = (2\pi - \sum_{j \in F(v_i)} \theta_j)/A_{mixed} \tag{10}$$

where $F(v_i)$ are vertex v_i's facet neighbors; θ_j is the angle of neighbor triangle at the vertex v_i; A_{mixed} is the mixed area used to account for obtuse triangulations.

To compute critical vertices, we first sort M_T's vertices and build a queue according to vertices' discrete Gaussian curvatures in descending order. Then, for each vertex v_i, we grow the "local region" that can approximate its neighborhood. All the vertices included in the region are removed from the queue. The same procedure is repeated until the sorted queue is empty or the number of critical points reaches a given threshold (e.g. $0.08 * \frac{N^2(V_T)}{N(V_S)}$). To judge whether vertices nearby v_i are in the local region, a scale independent threshold, e.g. 10^{-2} bounding box diagonal length, is used. As shown in Figure 5(d), our region growing algorithm, though simple, is more effective and robust in the context of consistent correspondence.

2.2.2 Establish Critical Vertices Constraints

After the critical vertices on M_T are detected, we need to map them onto M_S to ensure exact matches. Thanks to

our unified linear framework, this procedure is also straightforward. First, for each critical vertex on M_T, we choose the closest vertex on M_S as its counterpart. Then, similar to foregoing constraint items, we formulate the critical (salient) constraint term $\mathbf{E}_s$ as a quadratic energy function:

$$\mathbf{E}_s = \sum_{i \in S} \left| v_i^{init} - s_i \right|^2 \tag{11}$$

where S is the index set of critical constraint vertices.

After several iterated optimization steps, M_C^{init} can approximate M_T exactly at the critical vertices, thus the profiles of M_C^{init} will not collapse. Figure 6(d) shows the updated compatible mesh.

2.3. Vertex Relocation and Projection Scheme

Due to the global energy optimizing, the initial fitting result cannot guarantee that all the vertices of compatible mesh just lie on the surface of the target mesh. And the critical vertex constraints may introduce shape distortion as well. Therefore, we provide a novel vertex relocation and projection scheme, which refines the initial fitting result in the way of conformity (angle preservation). Specifically, for a vertex v_i of the initial result M_C^{init}, we want to find a new location satisfying some condition, e.g. improving the shapes (angle, size, orientation) of the triangles incident on v_i. This procedure is also called *remeshing* [11].

For this task, we define a shape preserving operator. To our knowledge, this is the first time that shape preserving operator has been used for remeshing. And our scheme can be implemented quite efficiently, such that we can refine the initial compatible mesh in a few seconds.

Based on the umbrella operator [7], our shape preserving operator $\mho$ is designed to optimize the interior vertices. The umbrella operator minimizes the membrane energy $\mathbf{E}_M$ of a mesh M. It shifts a vertex v_i depending on a convex combination of its direct neighbors v_{ij}. The choice of the weights w_{ij} determines the energy to be minimized in the optimization process. Uniform weights will provide a uniform distribution of triangles in the local domains. However, some surface features of mesh M might be lost during the uniform remeshing.

We found that by choosing the weights w_{ij} of the umbrella operator to be proportional to mean value coordinates covering the 1-ring of v_i, our shape preserving operator can successfully capture mesh's local shape features. Specifically, in case of the membrane energy $\mathbf{E}_M$ our shape preserving operator leads to a local update rule:

$$\begin{aligned} \mho: \quad & v_i \rightarrow \widetilde{v_i} = v_i + \lambda \overrightarrow{\mathbf{U}}(v_i) \\ = \quad & v_i + \lambda \frac{1}{\sum_{j=0}^{n} w_{ij}} \sum\nolimits_{j=0}^{n} w_{ij}(v_{ij} - v_i),\ w_{ij} \geqslant 0 \end{aligned} \tag{12}$$

where $\lambda = \alpha / d(v_i, M_t, \overrightarrow{\mathbf{U}})$, $d(v_i, M_t, \overrightarrow{\mathbf{U}})$ is the distance from v_i to M_T along the direction $\overrightarrow{\mathbf{U}}$, α is the gradual convergence factor used to avoid oscillations, and w_{ij} is the mean-value coefficient.

With this operator, each vertex of the initial result is gradually projected onto the target model's surface to ensure a complete surface match. The shape preserving operator is applied iteratively. With the vertices moving in the local domain the size of triangles (including the norm) needs to be recomputed after every iteration of the operator $\mho$. The iterations stop when the change of the vertex positions falls below a certain threshold. Usually very few (< 10) iterations suffice to reduce the total distortion, resulting in a mesh that preserves (almost) equal shape features with the source mesh M_S.

3. Results and Discussion

We tested our algorithm on various surfaces, including human face surfaces, animal models and complex geometrical objects.

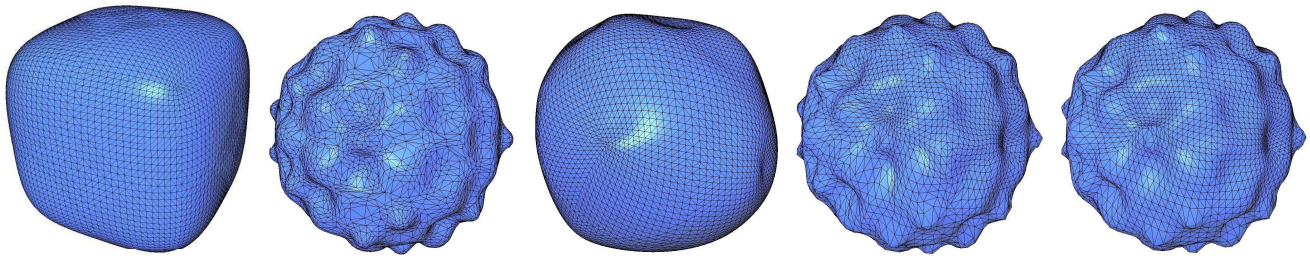
From Figure 3, it can be seen that our mean-value Laplacian fitting scheme successfully establishes a shape-preserving correspondence between the input surfaces, such as the threadlike stripes on lion's legs and belly. Moreover, instead of Euclidean nearest distance, our scheme adopts gradual approximating metric as the fitting measurement, thus effectively avoids trapping into local optimum suffered by the closest point fitting techniques when only a few of initial markers are provided (see Figure 4).

As shown in Figure 5, our critical points algorithm not only successfully captures the salient features of the surface, but also effectively avoids producing over-dense points that can cause large local distortions, thus is more suitable in the context of consistent correspondence than previous methods (such as [5]).

It is worthwhile to point out that it is rather difficult to build consistent correspondence between the two meshes shown in Figure 6 using previous methods (such as [2]) as the bumpy sphere model has lots of complex features not available in the cube model. Therefore, when only 8 markers are provided (lying on the eight corners of the cube), these features will be lost. Our critical point algorithm successfully makes M_C approximate M_T exactly at the critical vertices. Thus the profiles of M_C will not collapse.

As shown in Figure 6(e), our vertex relocation and projection method refines the initial fitting result in the way of local conformity, thus achieves more accurate correspondence results and ensures a complete surface match.

Figure 8 demonstrates a few snapshots from the morphing (shape interpolation) series of two models (a women face and a man face). Thanks to the well established consistent correspondence, it can be seen that the gradual changes are natural and visual appealing result is obtained. In another example (Figure 7), the three-sided blends of a lion,



(a) (b) (c) (d) (e)

Figure 6. Another example of our method. It also shows the flow chart of our method. The bumpy sphere model (b) has a large amount of complex features not available in the cube model (a). (c) Therefore, when only 8 markers are provided (just lying on the eight corners of the cube), the initial fitting result will miss these features. (d) To overcome this drawback, we exploit discrete Gaussian curvature to detect critical points and map them onto the cube model. (e) Finally, the result is further refined by our remeshing method to ensure a complete surface match in the way of conformity (angle preservation).

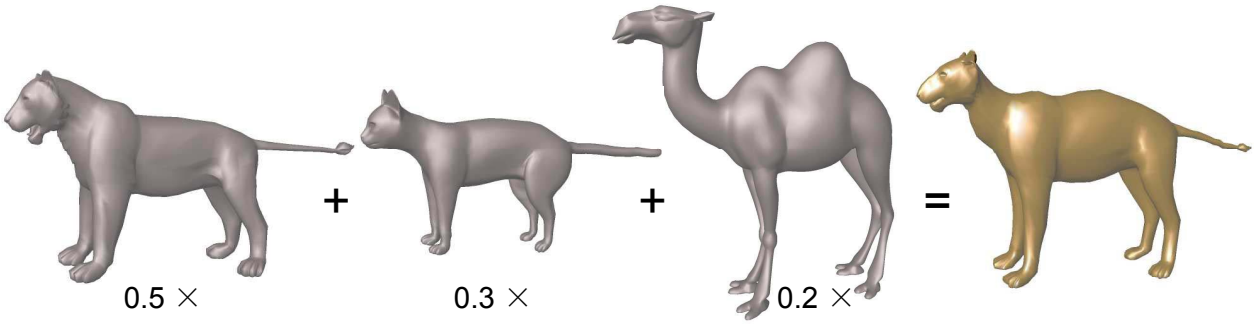


Figure 7. Having established consistent correspondence between input meshes, some originally difficult tasks (e.g. shape blending or statistical analysis) become quite easy. This example shows the blending of three models. The numbers in the blends are the affine combination weights of each model.

cat, and camel highlight our method's ability to establish consistent correspondence among multiple surfaces.

Since our method efficiently avoids falling into local optimum, fewer marker points than previous methods are needed. For the lion and cat examples in Figure 3, only 18 markers are enough for a good correspondent result, in comparison with 77 markers in [17] and 68 markers in [22] for this pair.

Our method builds the shape-preserving correspondence directly in 3D-to-3D space without needing to partition and flatten 3D surfaces onto 2D plane, i.e., directly exploits the 3D topology information of the whole source mesh. So the intrinsic difficulties suffered by indirect methods, such as preventing intersections and blocking, preserving cyclic orders, smoothing discontinuous boundaries, are effectively avoided. Therefore, our method doesn't need to pay particular attention to many extreme conditions, such as complex boundaries and high genus. In Figure 1, the man face model has one more boundary in the mouth than the woman face model. However, our method obtains satisfactory result without needing to carefully deal with this region.

Our framework is numerically efficient, as the solution to the optimization problem can be obtained by fast solving a sparse linear system. With a sparse LU solver [18], for example, 5.5K vertices require 0.6 seconds for factorization and 0.03 seconds for back-substitution on a 3.0GHz Pentium IV computer. Our critical points algorithm is also very fast. For the bumpy sphere model (11444 facets), the detection procedure only took 0.25 seconds.

4. Conclusions and Future Work

In this paper, we have introduced a novel framework for robustly computing consistent correspondence between meshes of arbitrary topology. Our mean-value Laplacian fitting method holds shape preserving property directly in 3D-to-3D space without segmenting the meshes. By detecting the critical vertices, our scheme accurately approximates the profiles and important features of the target geometry. Our vertex relocation and projection approach not only remeshes the global fitting result in the way of local conformity, but also ensures a complete surface match.

To automatically construct a high-quality consistent correspondence for two arbitrary shapes is still an open problem. One feasible way is to statistically analyze the whole surface and then compute out some candidate (i.e. not very accurate) marker pairs. In our framework, since the markers are only used for the initial correspondence (these constraints are gradually removed during optimiza-

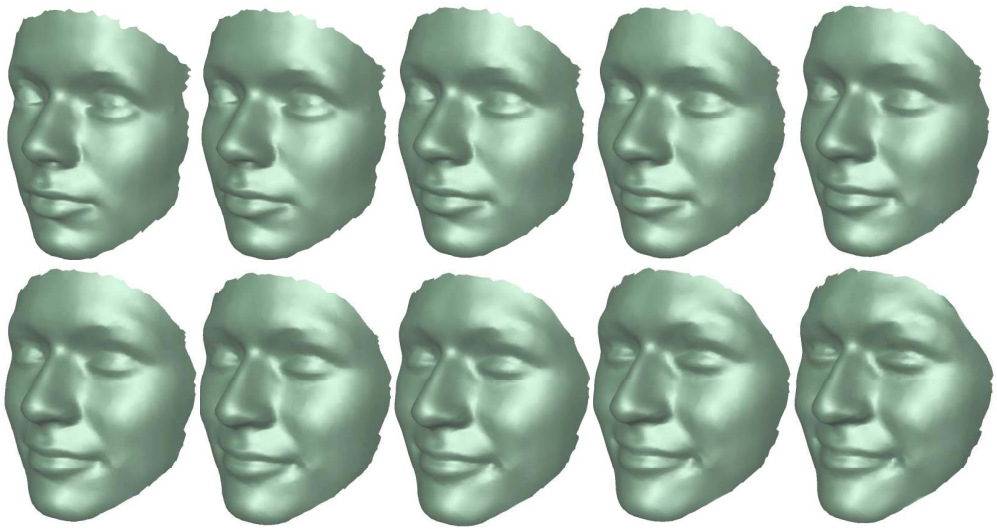


Figure 8. With a high-quality consistent correspondence between meshes, a women's face surface is being gradually transformed into a man's face surface just by linear interpolation.

tion), it is likely for these automatically detected markers to achieve desired correspondence results. We plan to carry on this work in the future.

Acknowledgements: This research work is supported by the National Natural Science Foundation of China (NSFC No. 60675012).

References

[1] M. Alexa. Recent advances in mesh morphing. *Computer Graphics Forum*, 21(2):173–196, 2002.

[2] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: Reconstruction and parameterization from range scans. *SIGGRAPH*, pages 587–594, 2003.

[3] Z. Fan, X. Jin, J. Feng, and H. Sun. 3d mesh morphing using polycube-based cross-parameterization. *Computer Animation and Virtual Worlds*, 16(3-4):499–508, 2005.

[4] M. S. Floater. Mean value coordinates. *Computer Aided Geometric Design*, 20(1):19–27, 2003.

[5] R. Gal and D. Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics*, 25(1):130–150, 2006.

[6] X. Gu and S.-T. Yau. Global conformal parameterization. In *Eurographics/SIGGRAPH SGP*, pages 127–137, 2003.

[7] L. Kobbelt, S. Campagna, J. Vorsatz, and H.-P. Seidel. Interactive multi-resolution modeling on arbitrary meshes. *ACM SIGGRAPH*, pages 105–114, 1998.

[8] V. Kraevoy and A. Sheffer. Cross-parameterization and compatible remeshing of 3D models. *SIGGRAPH*, pages 861–869, 2004.

[9] N. Litke, M. Droske, M. Rumpf, and P. Schröder. An image processing approach to surface matching. In *EG Symposium on Geometry Processing*, pages 207–216, 2005.

[10] M. Meyer, M. Desbrun, P. Schr, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. *Visualization and Mathematics III*, pages 35–57, 2003.

[11] G. Peyré and L. D. Cohen. Geodesic remeshing using front propagation. *IJCV*, 69(1):145–156, 2006.

[12] E. Praun and H. Hoppe. Spherical parametrization and remeshing. *SIGGRAPH*, pages 340–349, 2003.

[13] E. Praun, W. Sweldens, and P. Schröder. Consistent mesh parameterizations. In *Proceedings of ACM SIGGRAPH / ACM Transactions on Graphics*, pages 179–184, 2001.

[14] J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe. Inter-surface mapping. *ACM Transactions on Graphics*, 23(3):870–877, 2004.

[15] S. Shlafman, A. Tal, and S. Katz. Metamorphosis of polyhedral surfaces using decomposition. *EUROGRAPHICS / Computer Graphics Forum*, 21(3):219–228, 2002.

[16] O. Sorkine. Laplacian mesh processing. *Proceedings of Eurographics STAR*, pages 53–70, 2005.

[17] R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. *SIGGRAPH*, pages 399–405, 2004.

[18] S. TOLEDO. Taucs: A library of sparse linear solvers, version 2.2. *Tel Aviv University, Available online at http://www.tau.ac.il/ stoledo/taucs/, Sept.4*, 2003.

[19] S. Wang, Y. Wang, M. Jin, X. Gu, and D. Samaras. 3D surface matching and recognition using conformal geometry. In *CVPR*, pages 2453–2460, 2006.

[20] Y. Wang, X. Gu, K. M. Hayashi, T. F. Chan, P. M. Thompson, and S.-T. Yau. Surface parameterization using riemann surface structure. *ICCV*, pages 1061–1066, 2005.

[21] Y. Wang, M. Gupta, S. Z. 0002, S. Wang, X. Gu, D. Samaras, and P. Huang. High resolution tracking of non-rigid 3D motion of densely sampled data using harmonic maps. In *ICCV*, pages 388–395, 2005.

[22] R. Zayer, C. Rossl, Z. Karni, and H.-P. Seidel. Harmonic guidance for surface deformation. *EUROGRAPHICS / Computer Graphics Forum*, 24(3):601–609, 2005.

[23] L. Zhang, L. Liu, Z. Ji, and G. Wang. Manifold parameterization. *Computer Graphics International*, pages 160–171, 2006.