



## Joint planar parameterization of segmented parts and cage deformation for dense correspondence

Srinivasan Ramachandran<sup>a</sup>, Donya Ghafourzadeh<sup>a</sup>, Martin de Lasa<sup>b</sup>, Tiberiu Popa<sup>c</sup>, Eric Paquette<sup>a</sup>

<sup>a</sup>École de technologie supérieure

<sup>b</sup>Autodesk

<sup>c</sup>Concordia University

### ARTICLE INFO

#### Article history:

Received May 16, 2018

**Keywords:** Mesh Matching, Mapping Evaluation, Planar Parametrization, Attributes Transfer

### ABSTRACT

In this paper, we present a robust and efficient approach for computing a dense registration between two surface meshes. The proposed approach exploits a user-provided sparse set of landmarks, positioned at semantic locations, along with closed paths connecting sequences of landmarks. The approach segments the mesh and then flattens the segmented parts using angle-based flattening and low distortion boundary constraints. It adjusts the segmented parts with a cage deformation to align the interior landmarks. As a last step, our approach extracts the dense registration from the flattened and deformed segmented parts. The approach is capable of handling a wide range of surfaces, and is not limited to genus-zero surfaces. It handles small features, such as fingers and facial attributes, as well as non-isometric pairs and pairs in different poses. The results show that the proposed approach is superior to current state-of-the-art methods.

© 2018 Elsevier B.V. All rights reserved.

### 1. Introduction

A dense, non-rigid registration of two meshes consists of a mapping between the two surfaces. Non-rigid registration is a fundamental problem with applications in attribute transfer [1, 2], morphing [3], shape database analysis [4], and even deep learning on geometries [5]. The goal of the registration is to align the corresponding features of the meshes as closely as possible, while at the same time minimizing the geometric distortion of the mapping between the two surfaces. Current state-of-the-art methods are able to handle a wide range of surfaces, but impose genus-based limitations. Often, handling small features and non-isometric surfaces in different poses continues to be a challenge.

Among dense registration methods, we identify two ways to handle the problem of finding a mapping. The first type of methods consists in finding the dense registration by spectral analysis or by automatically detecting the sparse correspondences [6, 7, 8, 9]. This type of methods is fully automatic, but does not allow control over the final mapping, which can

sometimes exhibit mismatches at certain semantic areas. The second type of methods relies on the user to define sparse correspondences, and thus control the final mapping to some extent [1, 10, 11]. The approach presented here adopts this latter type, with the user retaining control over the final mapping. Further, reuse of a selected source mesh mapped with different targets can significantly reduce the user input, while sufficient control is retained. Shapes can be categorized according to their morphological classes; for each class, a source mesh is chosen, on which landmarks are created only once. The presented approach generates a mapping with low semantic and isometric distortion errors. It works by segmenting the two meshes into multiple parts to perform a part-wise matching. User-specified landmarks and closed paths drive the segmentation process. The pairs of segmented parts are then flattened, and their boundaries are aligned. Aligning only the boundaries of the patches does not guarantee the alignment of the interior landmarks. To address this problem, we apply a cage deformation step, which is a novel approach to aligning interior landmarks. The final step consists in constructing the mapping from the flattened and

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19

20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39

aligned parts. Our robust dense registration approach makes four novel contributions. Firstly, we demonstrate a process for constructing small patches based on closed paths. Secondly, we propose a dual-flattening approach using the mesh with the least distortion to align mesh boundaries. Thirdly, we present a novel cage deformation method that aligns interior mesh features, also ensuring no fold-overs are introduced in the flattened meshes. Finally, we propose a quantitative evaluation measure using isopoints to compare different dense registrations. It should be noted that in our paper, we refer to a non-rigid dense registration as a *surface mapping* or simply as a *mapping*.

## 2. Previous Work

Surface mapping methods, also referred to in the literature as correspondence or registration methods, relate semantically similar surface components to one another. We are interested in mapping methods that can handle a wide range of surfaces. Accordingly, we favor methods that are not limited to genus-zero surfaces, or that can handle surfaces with small features, such as fingers and facial attributes. We are also interested in methods that do not impose too many constraints on the surfaces. In that regard, we want to handle non-isometric surface pairs, pairs with different genera, and pairs set in different poses. In this section, we present the mapping methods most relevant to the proposed approach. We classify them in terms of the space within which they establish the mapping: 3D Euclidean, Möbius, functional, spherical, and planar. The reader is referred to the survey of van Kaick et al. [12] for a more exhaustive list of geometric correspondence methods.

### 2.1. Deformation in 3D Euclidean Space

Non-rigid registration methods deform the given surfaces until they match [4, 13, 14]; however, most such methods are limited to near-isometric objects. Generally, few methods try to extend the range of objects to handle non-isometric pairs. Sumner et al. [10] propose an iterated closest point method with regularization based on input landmarks to deform one surface into another, and allowing the extraction of mapping through the deformed surface. Zell and Botsch [11] combine the concepts of deformation-based registration and transformation of surfaces into smoother shapes. While their method works relatively well for character heads, it has a strong tendency to collapse protruding extremities, such as legs and arms, which causes artifacts in the resulting mapping. Methods that deform surfaces in 3D Euclidean space are prone to fail if the surfaces have different poses; accordingly, their resulting mapping depends greatly on how well the surfaces are initially aligned. Moreover, most of these methods only handle near-isometric objects or small non-isometric deformations [13], which in turn highly restricts their application domain.

### 2.2. Möbius and Functional Spaces

Möbius methods [7, 15] rely on the hypothesis that isometries are a subspace of conformal maps, which could be explored based on Möbius transformations. These methods are limited to isometric and near-isometric surfaces. Kim et al. [8] present

Blended Intrinsic Maps (BIM) to handle non-isometric surfaces by using weighted combinations of low-dimensional intrinsic maps to generate a blended map. The BIM method provides an efficient search procedure to find smooth maps between surfaces in a fully automatic fashion. The method handles surfaces with different poses, but it fails for examples containing small features such as facial details and fingers.

The functional space of the Laplace-Beltrami decomposition is also used to express mappings based on real valued functions instead of the regular point-to-point maps [16, 17]. This provides a flexible representation of the maps between the shapes, but as with the Möbius methods, it struggles in handling mappings between non-isometric surface pairs.

### 2.3. Spherical Parameterization

Reliance on parameterization works by the transformation of the surfaces into a space where detecting a correspondence is facilitated. The spherical domain allows a seamless and continuous parameterization of genus-zero surfaces [18, 19]. Athanasiadis et al. [20] drive a geometrically-constrained optimization technique to map 3D genus-zero surfaces on a sphere. Then, they apply a feature-based method to morph between surfaces with structural similarities. Mocanu et al. [21] present a spherical parameterization method relying on Gaussian curvature in order to align feature correspondences of the input surfaces. Then, they apply a morphing step by establishing correspondences. Their method generates artifacts on hands, feet, and facial feature examples due to triangle degeneration and fold-over problems. Accordingly, in spherical parameterization methods, surfaces with higher degrees of complexity and features increase the distortions. Moreover, spherical parameterization methods provide a seamless parameterization, but are restricted to genus-zero surfaces.

### 2.4. Planar parameterization

To deal with higher genera and objects with finer details, a widely used approach is to cut the surfaces prior to extracting the mapping. Bronstein et al. [22] propose a framework allowing such matching of partial shapes through an optimization that computes a minimum-distortion mapping. The presented approach is the most similar to methods that cut the surfaces and that also flatten the resulting pieces. Some methods [23, 24] can be applied to objects of arbitrary genus, but they require a very carefully chosen and, in some cases, large set of corresponding landmarks.

Aigerman et al. [25] provide a bijective mapping between the surfaces based on user landmarks and a cut-graph. It cuts the surfaces and flattens them based on a minimization of isometric distortion. Their method generates artifacts and jumps, with both of these problems occurring across the cuts. In a follow-up work [26], Aigerman et al. overcome the cut-graph problem. The method applies a surface flattening (G-flattening), which is optimized with an energy functional. G-flattening denotes a group of affine transformations, which are optimized to minimize distortion energy. All these planar parameterization methods present the advantage of being able to handle non-isometric surfaces and different poses, but the mappings resulting from

1 them depend greatly on the user inputs, and distortion occurs  
2 near the landmarks.

3 In addition to spherical and planar domains, other domains  
4 such as hyperbolic orbifolds have been proposed [27, 28]. These  
5 methods are also constrained to genus-zero models.

6 The presented approach aims to handle non-isometric surfaces  
7 with fine features, and that can have different genera and poses.  
8 Planar parameterization is thus the best option, given our con-  
9 cerns; the only problem is the amount of user input it requires. It  
10 should nevertheless be noted that this user input is proportional  
11 to the complexity of the surfaces to be matched.

### 12 3. Registration with Closed Paths and Cages

13 Fig. 1 shows an overview of our approach. In it, the meshes are  
14 cut into smaller segmented parts, making them homeomorphic to  
15 a disk. These smaller segmented parts are then flattened, while  
16 avoiding distortions, fold-overs, and overlapping parts problems.  
17 The latter is achieved with our selection of boundary conditions  
18 and our cage deformation, which are both used to align the  
19 boundaries and interiors of pairs of segmented parts. Finally, the  
20 mapping is extracted based on how the pairs overlap each other.  
21 The approach establishes the mapping between two surfaces:  
22  $S$ , referred to as the *source*, and  $T$ , referred to as the *target*.  $S$   
23 and  $T$  are discrete surfaces provided as triangular meshes. In  
24 addition to the meshes, the user provides a sparse set of semantic  
25 landmarks  $L_s = \{l_{s_1}, l_{s_2}, \dots, l_{s_k}\}$  and  $L_t = \{l_{t_1}, l_{t_2}, \dots, l_{t_k}\}$  for  $S$   
26 and  $T$  (red and green spheres in Fig. 1), together with closed  
27 paths for  $S$  or  $T$  (yellow outlines in Fig. 1). The entries in  $L_s$   
28 and  $L_t$  are spatial locations on the surface in  $\mathbb{R}^3$ , expressed as  
29 barycentric coordinates based on a face in  $S$  and  $T$ , respectively.  
30 Each landmark on  $S$  has a corresponding landmark on  $T$  placed  
31 at the corresponding semantic location. In order to extract the  
32 segmented parts, the user also provides closed paths connecting  
33 landmarks. The mesh is cut along these paths, creating the  
34 segmented parts ( $\hat{S}_1, \hat{S}_2, \hat{T}_1$ , and  $\hat{T}_2$  in Fig. 1(b) of the surfaces.  
35 The closed paths are defined either on  $S$  or  $T$ , and replicated on  
36 the other using the corresponding landmarks.

37 The bijective mapping between the sparse sets of semantic  
38 landmarks  $\{(l_{s_1}, l_{t_1}), \dots, (l_{s_k}, l_{t_k})\}$  and the related bijective map-  
39 ping between closed paths are used as constraints to find a sur-  
40 face mapping  $f : S \rightleftharpoons T$ . The steps of our approach are:

- 41 1. Segmentation using closed paths (Sec. 3.1): With the user-  
42 supplied closed paths travelling along the meshes,  $S$  and  $T$   
43 are cut into segmented parts (see Fig. 1(a)).
- 44 2. Planar parameterization of segmented parts (Sec. 3.2): A  
45 planar parameterization is applied for each pair of seg-  
46 mented parts. The left and right sides of Fig. 1(b) represent  
47 this step.
- 48 3. Cage deformation (Sec. 3.3): This step aligns the interior  
49 landmarks of the segmented parts. The middle of Fig. 1(b)  
50 shows this step.
- 51 4. Mapping extraction (Sec. 3.4): The mapping is extracted  
52 from the overlapping segmented parts and transferred back  
53 to the original meshes.

54 The following sections elaborate on each step in detail.

#### 3.1. Segmentation Using Closed Paths

55 Closed paths  $G$  on a mesh are defined by connecting a se-  
56 quence of semantic landmarks from that mesh. In our imple-  
57 mentation we choose geodesic paths to connect the sequence of  
58 landmarks. Given a sequence of semantic landmarks, geodesic  
59 paths connecting them are found on each mesh respectively. To  
60 find geodesic paths between the landmarks, a vertex is added  
61 to the mesh for each landmark if it is not already on a vertex.  
62 The sequence of landmarks are then connected by a geodesic  
63 path local to each mesh. Each path travels along the surface  
64 from landmark to landmark, and will create a closed loop by  
65 connecting back from the last landmark to the first landmark of  
66 the sequence (see Fig. 2a). A landmark can be used only once  
67 in the sequence, and by definition, we impose the condition that  
68 the paths on the surface not intersect each other (see Fig. 2b) and  
69 do not self-intersect (see Fig. 2c, 2d and 2e). Corresponding  
70 landmarks are identified on the second mesh and connected in a  
71 similar fashion, yielding closed paths  $G_s = \{G_1^s, G_2^s, \dots, G_n^s\}$  on  
72  $S$ , and  $G_t = \{G_1^t, G_2^t, \dots, G_n^t\}$  on  $T$ .

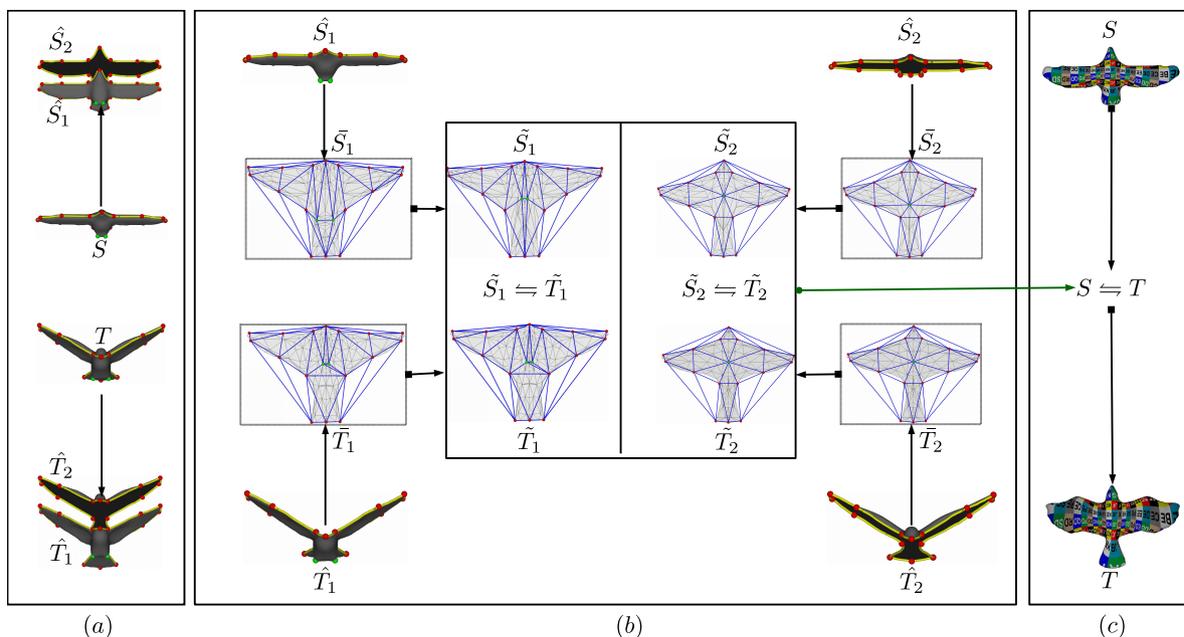
74 The objective of the closed paths in  $G_s$  and  $G_t$  is to define  
75 regions on the meshes that will become segmented parts. Tri-  
76 angles crossed by closed paths are split with additional vertices  
77 and edges such that two sub-triangles can later be linked to the  
78 correct segmented part on each side of the path. Closed paths  
79 are processed one by one to cut the mesh. Each closed path  
80 cuts the mesh by disconnecting faces sharing an edge traveled  
81 by the closed path. This process duplicates vertices and edges  
82 where needed, and will link the landmarks to both segmented  
83 parts. The resulting disconnected components (the intermediate  
84 segmented parts) are iteratively processed in the same way with  
85 each closed path. By definition, the user has to create the paths  
86 such that each resulting segmented part is homeomorphic to a  
87 disk.

88 The segmented parts on  $S$  and  $T$  need to be linked to each  
89 other, yielding  $P = \{(\hat{S}_1, \hat{T}_1), (\hat{S}_2, \hat{T}_2), \dots, (\hat{S}_m, \hat{T}_m)\}$ . The ap-  
90 proach identifies  $(\hat{S}_i, \hat{T}_i)$  based on the landmarks they contain.  
91 Fig. 3a shows a case where the two resulting segmented parts  
92 have the exact same landmarks. In such cases, the user should  
93 set an additional *correspondence* landmark on one of the parts,  
94 such that the correspondence can be established without any  
95 ambiguities (see Fig. 3b). Only the two examples of Fig. 19  
96 required a few correspondence landmarks.

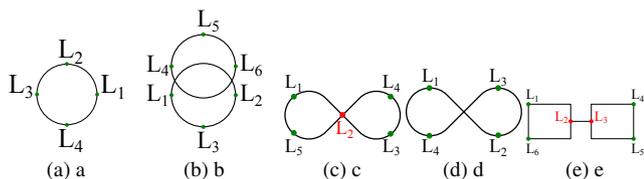
97 Several techniques can be used to find the paths. During the  
98 interactive creation of the paths, we often first used Dijkstra on  
99 the edge graph of the mesh as it is fast to compute. When it  
100 did not provide an adequate path, we switched to more time-  
101 consuming techniques [29, 30] that improved the smoothness  
102 of the paths. The closed paths need to be specified once for  
103 each source in a class, and can be reused across multiple targets.  
104 From our experiments, it was observed that one closed path was  
105 sufficient for genus-zero meshes, while higher-genus meshes  
106 required more closed paths.

#### 3.2. Planar Parameterization of Segmented Parts

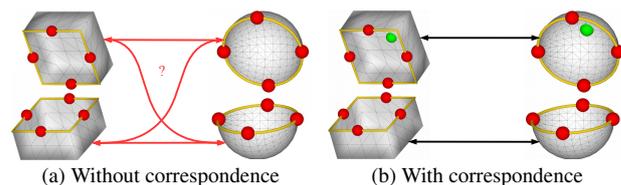
107 The goal of this step is to find a single planar parameteriza-  
108 tion where each pair of segmented parts  $\hat{S}_i$  and  $\hat{T}_i$  overlap (see  
109



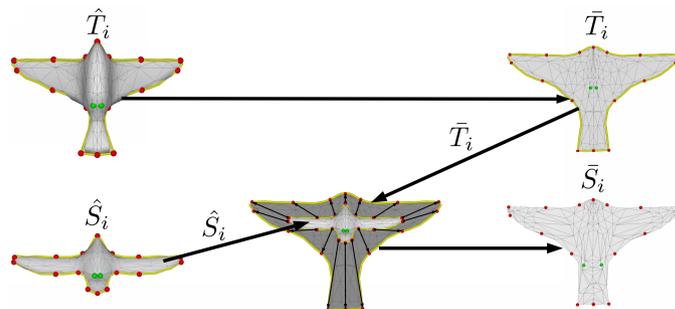
**Fig. 1.** Overview of the approach: (a) The source and target meshes and their respective segmented parts ( $\hat{S}_1, \hat{S}_2, \hat{T}_1, \hat{T}_2$ ). (b) Planar parameterization applied on the source (top) and target (bottom). The pairs  $\tilde{S}_1, \tilde{T}_1$  and  $\tilde{S}_2, \tilde{T}_2$  are aligned with their boundaries. Internal landmarks of the target (colored in green) are aligned to the source internal landmarks using our deformation based on cages (blue lines), resulting in new pairs of fully aligned mesh pairs  $\tilde{S}_1, \tilde{T}_1$  and  $\tilde{S}_2, \tilde{T}_2$ . The mappings are extracted between each pairs and transferred to the original source and target. (c) Visualization of the resulting mapping between the source and target.



**Fig. 2.** (a) A valid closed path from the landmark sequence  $\{L_1, L_2, L_3, L_4\}$ . (b) Invalid: two paths cannot intersect each other. (c) Closed path  $\{L_1, L_2, L_3, L_4, L_2, L_5\}$ , but intersects (crosses itself) at  $L_2$  colored in red. (d) Closed path  $\{L_1, L_2, L_3, L_4\}$ , but intersects (crosses itself) between landmarks. (e) Intersecting closed path  $\{L_1, L_2, L_3, L_4, L_5, L_3, L_2, L_6\}$  with two repeating landmarks colored in red.



**Fig. 3.** (a) Two segments sharing the same landmarks (colored in red). This will lead to an ambiguity in matching segmented parts. (b) With a single correspondence landmark (colored in green), the segmented parts can be matched without ambiguity.



**Fig. 4.** Boundary constraints. The top row shows the planar parameterization of a target segment  $\hat{T}_i$  using ABF++ resulting in  $\tilde{T}_i$ . The bottom row shows the parameterization of a source segment  $\hat{S}_i$  by constraining boundary vertices to positions along the boundary of  $\tilde{T}_i$ , thus resulting in  $\tilde{S}_i$ .

1 Fig. 1(b)). The mapping is later extracted from this overlapping  
2 parameterization.

3 To avoid issues in the mapping, a good choice of planar param-  
4 eterization is vital. Our objective is to keep the parameterization  
5 distortion to a minimum, as well as ensure a bijective mapping  
6 by avoiding fold-overs and overlapping parts in the resulting planar  
7 parameterizations. We use the ABF++ method by Sheffer et  
8 al. [31] to generate the planar maps, as it produces low conformal  
9 distortion and uses free boundaries. As in the work of Bradley  
10 et al. [32], we use the assumption that low-distortion planar param-  
11 eterizations (in our case, low conformal distortion as we use  
12 ABF++) allow to extract low distortion mappings. Furthermore,  
13 the free boundaries of ABF++ avoid unnatural distortions that  
14 occur on triangles connected to the boundaries. Highly distorted  
15 triangles lead to sudden jumps in correspondence, preventing  
16 the extraction of a smooth mapping.

17 Each segmented part consists of a boundary (defined by the

18 closed paths) passing through semantic landmarks. For each pair  
19 in  $P$ , a planar parameterization is applied to segmented parts  $\hat{S}_i$   
20 and  $\hat{T}_i$ , resulting in  $\tilde{S}'_i$  and  $\tilde{T}'_i$ . Either  $\tilde{S}'_i$  or  $\tilde{T}'_i$  is used  
21 as a basis to define the boundary conditions. We select the one with lower

distortion values  $L_2$  and  $L_\infty$  using the stretch metric of Sander et al. [33]. For clarity, going forward, explanations will assume that  $\tilde{T}_i'$  is the lower distortion mesh and  $\tilde{T}_i'$  becomes the final  $\tilde{T}_i$ .

The boundary of  $\tilde{T}_i$  is used as the boundary conditions to find the planar map of  $\tilde{S}_i$ . The boundary for  $\tilde{S}_i$  is found by first positioning the boundary landmarks of the source to match the position of the boundary landmarks of  $\tilde{T}_i$  (see Fig. 4, middle bottom). Between each pair of landmarks along the boundary, the vertices of the boundary of the source are positioned along the boundary of  $\tilde{T}_i$  at the same ratio of distance between the two landmarks as they were along the boundary of  $\tilde{S}_i$  (see Fig. 4, bottom right).

### 3.3. Cage Deformation

Planar maps  $\tilde{S}_i$  and  $\tilde{T}_i$  are similar because of the constrained boundary conditions used in obtaining  $\tilde{S}_i$ . The semantic landmarks chosen to represent each closed path are termed boundary landmarks, while those not used for any closed path are interior landmarks. Not all pairs  $\tilde{S}_i$  and  $\tilde{T}_i$  have interior landmarks. For segmented parts without interior landmarks, the cage deformation step is skipped. For the pairs  $\tilde{S}_i$  and  $\tilde{T}_i$  that do have interior landmarks, those landmarks are unlikely to align with each other (see Fig. 1 (b)).

The interior landmarks have to be aligned to extract a good quality mapping. To this end, a triangular cage mesh  $C_i^t$  is constructed for  $\tilde{T}_i$ , connecting the semantic landmarks using a Delaunay triangulation (colored blue in Fig. 1(b)). The connectivity information for  $C_i^t$  is replicated on  $C_i^s$  as they share the same set of semantic landmarks.

The cage  $C_i^s$  is transformed to align with  $C_i^t$  by moving the landmarks of  $C_i^s$ . This will affect the triangles of the cage mesh  $C_i^s$ . The change in shape of the triangles of cage mesh  $C_i^s$  will move the interior vertices of  $\tilde{S}_i$ . The deformed segmented parts are referred to as  $\tilde{S}_i$ . Given this improved semantic alignment, a mapping can be found, as is explained in the next section.

In order to obtain a bijective mapping, it is sufficient (as it will be shown in Section 5.2) to construct a cage triangulation that does not have foldovers simultaneously in both the source and the target mesh embedding. Unfortunately, a simple Delaunay triangulation may not have this property as shown in Fig. 5(a)-(b). Kraevoy et al. [23] show an algorithm that can construct such a triangulation that is simultaneously fold-over free in two different planar embeddings using Steiner vertices if necessary. Adding Steiner vertices does not affect our algorithm as they correspond to additional constraints added automatically. Fig. 5(c)-(d) shows a triangulation that is foldover free in both source and target cages.

### 3.4. Mapping Extraction

The mapping between  $\tilde{S}_i$  and  $\tilde{T}_i$  is expressed as a barycentric location for every vertex in  $\tilde{S}_i$  on the closest face in  $\tilde{T}_i$ , and *vice versa*. The presented approach employs a KD-tree to efficiently find the closest locations. The mapping for  $S$  and  $T$  are found by transferring and aggregating the individual mappings between each pair of  $\tilde{S}_i$  and  $\tilde{T}_i$ .

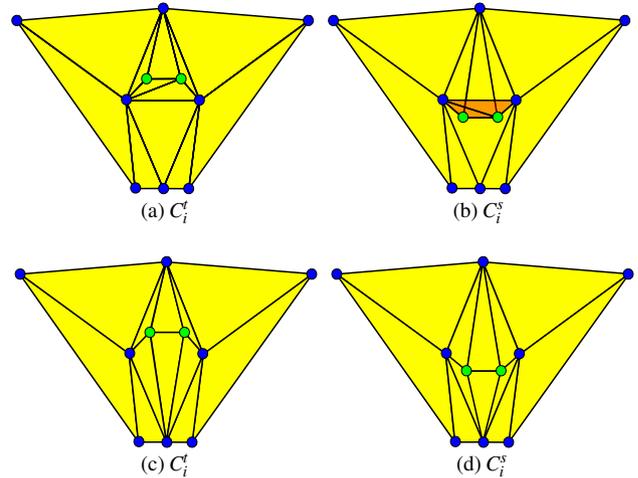


Fig. 5. Top row: (a) and (b) show that the original cage triangulation cannot guarantee the absence of foldovers. This issue is fixed in the triangulation of (c) and (d).

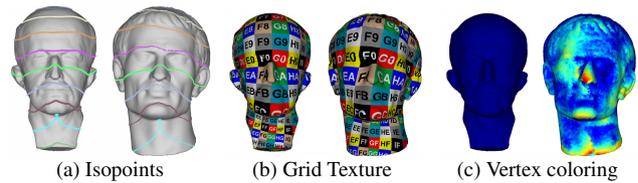


Fig. 6. The visualization techniques used for evaluation purposes. (a): Isopoints, (b): Grid texture, (c) Vertex Coloring

## 4. Results

This section starts with a detailed explanation of evaluation techniques used, and how they are applied to the results. Then, it proceeds to compare the presented approach with other methods.

### 4.1. Qualitative Evaluation

A good visualization allows the evaluation of two key elements, namely, smoothness and distortion. To address these requirements, we used three visualization techniques: using isopoints, grid texture, and vertex coloring (see Fig. 6). For the isopoints visualization, a scalar function representing the geodesic distance is calculated on the source mesh. The geodesic distance is calculated between each vertex and a seed point that we manually selected on the source mesh for each class. Using this geodesic distance, 10 isocurves are found. A different color is assigned to the isopoints of consecutive isocurves to facilitate the identification of anomalies. Isopoints are the locations where an isocurve crosses an edge or a vertex of the source mesh. With a mapping  $f : S \Rightarrow T$  each isopoint is transferred to the target. Isopoints lying on a vertex are directly transferred to the target face using the mapping. For an isopoint on an edge, we first get the locations on the target corresponding to the vertices of the edge. Then, we linearly interpolate between the two locations to get the position at the same ratio of distance as the isopoint was on the source. Finally, that position is projected to the closest face of the target.

The erroneous areas can be identified at places where the isopoints are too cluttered or too distant from each other, or as well as where we observe zigzagging along the sequence of points. Moreover, erroneous locations in the mapping are identified by isopoints that are missing at locations where they are expected. This technique is also useful for visualizing the smoothness of a mapping.

Our second qualitative evaluation technique works by applying a  $uv$  map with a checkerboard texture on the source mesh. The mapping  $f : S \rightleftharpoons T$  expresses a location for every source vertex  $v_s$  and target vertex  $v_t$  as a barycentric coordinate on a face of the other mesh,  $f_t$  and  $f_s$  respectively. To transfer the grid texture, every vertex  $v_t$  is assigned a texture coordinate by interpolating the  $uv$  coordinates of vertices on the mapped face  $f_s$ . This technique is useful for finding problems such as distortions and semantic mismatches.

The third qualitative evaluation technique morphs the source to the target and is our *deformed source*. Then for each vertex of the target its corresponding location is found on the *deformed source*. If a mapping is good, the distance between the vertex location of the target and its corresponding location on the *deformed source* should be small. The displacement for each vertex of the target is colored based on the distance errors of the mappings. This visualization readily shows regions of meshes with higher errors.

#### 4.2. Quantitative Evaluation

The qualitative evaluation techniques convey the problems of mapping from a visual sense. For their part, quantitative measurements are useful for comparisons between different mappings from a numerical perspective. Hence, a quantitative measure is proposed for comparison between mappings that is highly motivated towards finding semantic discrepancies. Our measure derives from the following observation: if we deform the target mesh (morphing) into the source mesh using a good mapping, corresponding locations of the source vertices on the deformed target should be close to each other. If the mapping maps together places that do not relate to each other, corresponding locations will be further from each other. By measuring that distance and adding up for several locations, we can make comparisons between mappings. Before adding up the distances, two operations must be completed: (i) deformation of the target into the shape of the source and (ii) identification of corresponding locations.

For the first operation, our *deformed target* is the mesh of the target morphed to the shape of the source using a mapping. As explained earlier, the mapping of the target expresses a location for each target vertex  $v_t$  using barycentric coordinates on a source face  $f_s$ . This is used to find a spatial location for  $v_t$  along the surface of the source, thus morphing to the shape of the source.

For our second operation, we will identify the corresponding locations by transferring a set of points from the source to the target using the mapping. To this end, we will be using the isopoints. Although we use the same isopoints as in Sec. 4.1, it should be noted that the transfer of isopoints for the visualization and the transfer of isopoints for a quantitative measure are two different processes.

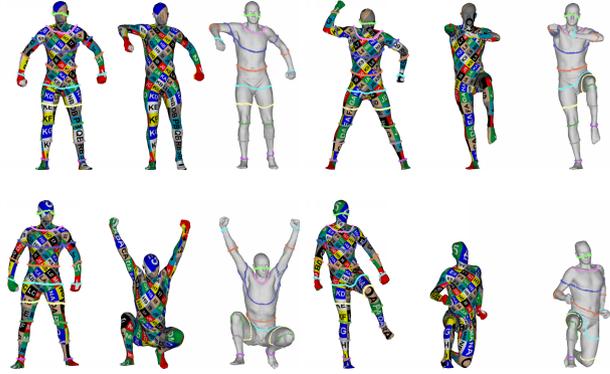


Fig. 7. Isopoints and grid visualizations for the mappings of four different SCAPE pairs. The mappings were created using the benchmark landmarks of the SCAPE database.

The distance between corresponding isopoints, the source isopoint and deformed target isopoint, is seen as an isopoint error. This error measures two things. First, it measures the bijectivity of the mapping as we use the mapping twice: for isopoints mapped from the source to the target, and target vertices mapped from the target to the source. Secondly, it measures how well the mapping links related regions together. As an example, we often saw mappings from BIM incorrectly linking small features, such as the toes to coarser features such as the ankle. Even though such a mapping could be bijective, the number of vertices on the ankle is insufficient to recreate the small features of the toes, resulting in an odd looking deformed target and in an increase in the distances we measure.

#### 4.3. Near Isometric Objects (SCAPE Dataset)

We first experimented with mappings between near-isometric meshes. To this end, we used the SCAPE [34] database that contains 71 meshes representing one model in different poses with their groundtruth correspondence between the vertices. Also the database provides a set of 36 landmarks that was used for our experiments. A single closed path was specified on one of the meshes passing through 25 of the 36 landmarks. As these pairs have ground truth correspondence between their vertices the same landmarks and closed path were used to establish a mapping between four pairs of SCAPE meshes. As it can be seen in Fig. 7 the mappings are smooth with low distortion everywhere.

In Fig. 8 we show the comparison between the groundtruth and our mapping. The groundtruth is globally accurate, but



Fig. 8. Comparison between groundtruth mapping with our mapping for the example shown in Fig. 7a. The blown-up view of the chest region is shown here. It can be seen that groundtruth mapping is accurate, but exhibits the high-frequency distortion typical of vertex to vertex mappings. Note how our mapping is both accurate and smooth.

lacks smoothness. This is a typical problem of vertex to vertex correspondences, but our mapping is devoid of those problems, similar to the results of Aigerman et al. [25]. We also ran the evaluation tool provided by Kim et al. [8] on random pairs of SCAPE models to compare with other mappings. Fig. 9 shows the average error recorded by the benchmark tool for SCAPE class of meshes. The average error recorded for our method is the lowest.

#### 4.4. Comparisons with other approaches

We compared our approach to BIM [7], Symmetry Axis Curves (SYMAXIS) [9], Elastiface [11], Deformation Transfer (DT) [10], Lifted Bijections (LB) for low distortion surface mappings [25], seamless surface mappings [26] and Orbifold Tutte Embeddings (OBTE) [35]. The first two methods BIM and SYMAXIS are based on blending maps to achieve the final mapping. These methods do not require any manual correspondences as inputs, but such manual correspondences can still be specified during the pipeline. If manual correspondences are not provided then an initial set of sparse feature correspondences are found by extracting points at maxima of average geodesic distance function.

The following two methods Elastiface, and DT, work by deforming in the 3D space and requires manual correspondences to be specified. These approaches employ energy minimization and solve linear systems of equations to deform the source or target or both meshes to a similar looking shape. The mapping is then established by closest point correspondences on the deformed shapes.

The final approach OBTE is a state of art approach and it works with fixed numbers of user supplied landmarks (either three or four). The approach starts by connecting the landmarks to cut through the edge graph of the mesh and make it homeomorphic to a disk. It optimizes the mesh boundary, beginning from the shape of a convex polygon with the number of polygon sides based on the input orbifold and then progressively solves for a boundary with least distortion.

Not all the above mentioned methods are capable of successfully generating a mapping for all of the test cases. The characteristics of some of the inputs, such as genus differences and manifoldness, render some methods unusable, but are not a limitation in the presented approach.

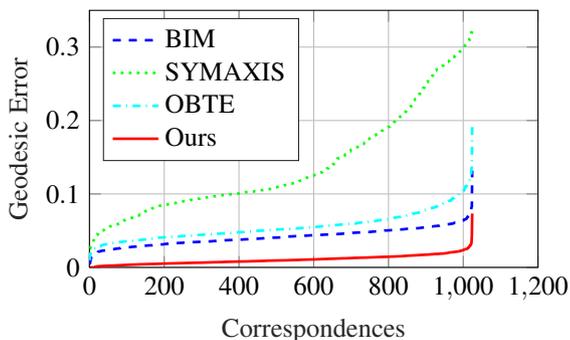


Fig. 9. Comparison of the average errors of SCAPE meshes with different mapping methods using the benchmark evaluation provided by the Kim et al. [8].

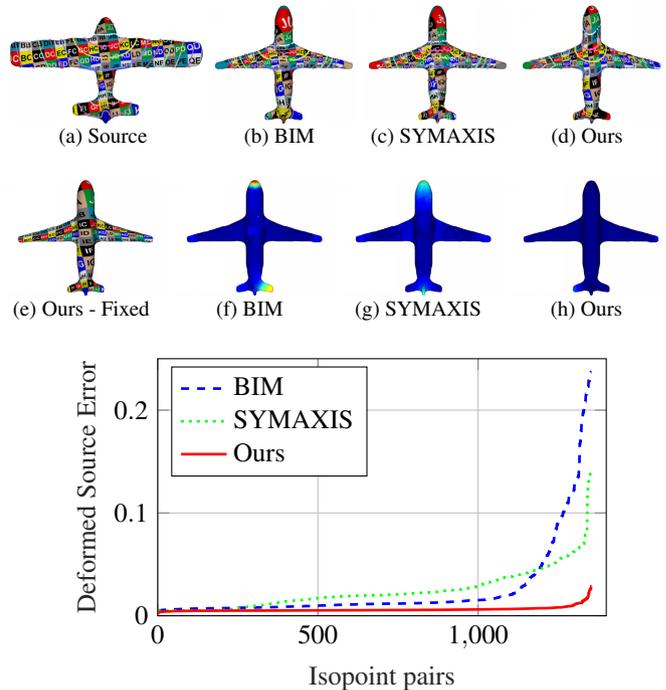


Fig. 10. Isopoints, grid, and deformed source error visualization show mappings of BIM, SYMAXIS, and ours compared to each other. Top: Grid texture visualization. Middle: deformed source error visualization. Bottom: Graph showing for the isopoint pairs deformed source error, sorted from lowest to highest

#### 4.4.1. Comparison to BIM and SYMAXIS

We first compared our method to BIM and SYMAXIS as both approaches work on a similar foundation of employing blended maps during the step of correspondence extrapolation. BIM relies on the maxima of average geodesic distance to automatically identify feature point correspondences. We used the set of five feature points and their inter-correspondence from BIM to generate a mapping. A closed path was created on the source connecting these five feature points and was then transferred to the target mesh based on the correspondence between the landmarks. The comparison can be seen in Fig. 10. Firstly, the mappings are upside for all the mappings. This is because of a symmetry flip in the feature point correspondences found by BIM. SYMAXIS produces a lot of noise as seen in Fig. 10c and 10g because of their vertex to vertex correspondences instead of barycentric locations. BIM produced less noise but still results in a distorted mapping (see Fig. 10b and 10f). Our map is superior to both and exhibits a smooth mapping. By adding one extra landmark that is not part of the closed path on both source and target, and also by manually fixing the feature point correspondence, the mapping was fixed removing the left-right, top-bottom flip (see Fig. 10e).

#### 4.4.2. Comparison to DT and Elastiface

Secondly, we compared our approach to methods relying on deformation in 3D Euclidean space: DT and Elastiface. Fig. 11 shows a representative example for the animal class. Both source and target were chosen from the SHREC dataset (389 and 397

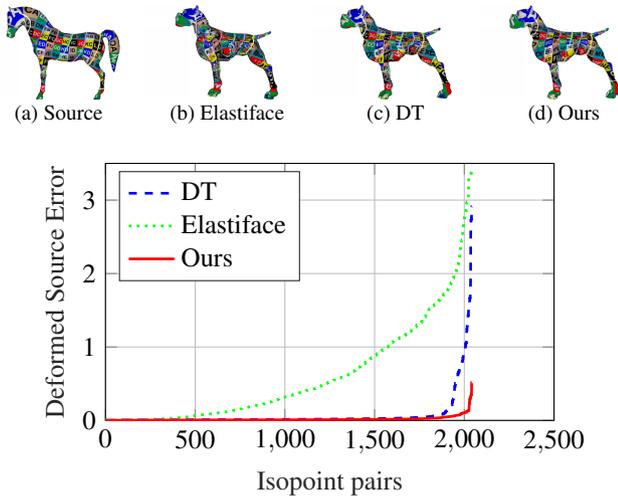


Fig. 11. Isopoints, grid, and graph show mappings of Elastiface, DT, and our approach compared to each other. It is clear from the graph that our approach provides the best mapping, even compared to DT.

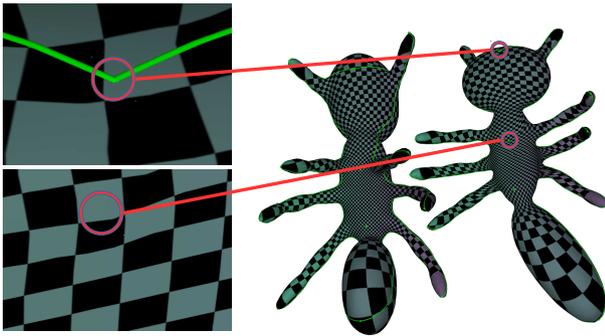


Fig. 12. Results based on our approach with similar objects, viewpoints, landmarks, and texture visualization as those from Fig. 12 of the paper by Aigerman et al. [26]. The distortion for the interior landmarks in our approach is less than the distortion found in Fig. 12 from the paper by Aigerman et al. [26].

1 respectively). We provided the same landmarks and compared  
 2 the mappings. DT performs better relative to Elastiface, but it is  
 3 still outperformed by our approach as seen with the quantitative  
 4 measure in Fig. 11e.

#### 4.4.3. State of the Art Cut-Graph Methods [25, 26]

5 As in our approach, cut-graph methods [25, 26] cut the source  
 6 and target meshes, and establish the correspondence between the  
 7 flattened parts. The main limitation of cut graph methods [25,  
 8 26] is having all of the landmarks lie along the cut and along the  
 9 resulting boundary of the segment. This can create a cluttered  
 10 configuration if a large number of landmarks are used that may  
 11 create visual artifacts. Compared to these methods, our  
 12 approach is more scalable and flexible allowing arbitrary interior  
 13 landmarks. These landmarks act as internal point constraints,  
 14 without the need to increase the complexity of the boundary. We  
 15 ran an experiment on a pair from the SHREC dataset (meshes  
 16 #82 and #95) placing our landmarks at the same place as in the  
 17 paper by Aigerman et al. [26]. When comparing the distortion  
 18 of our approach (Fig. 12) to that shown in Fig. 12 of the paper  
 19

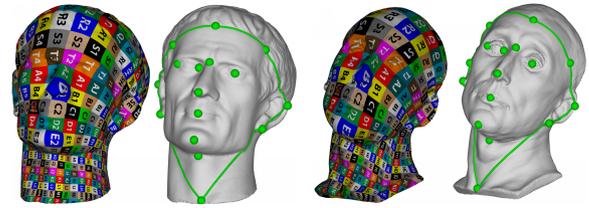


Fig. 13. Results based on our approach with landmark positions and number matching those used in Fig. 1 of the paper by Aigerman et al. [25]. Note how our mapping is smooth across the seams.

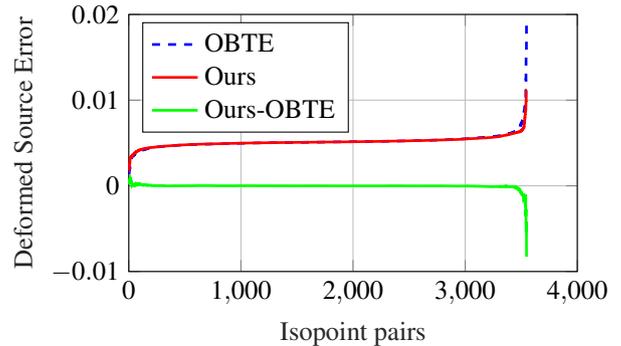


Fig. 14. Comparison between our approach and OBTE. The mappings are almost similar except for few points on OBTE for which the error is higher than ours. This can be seen with the plotting of difference between ours and OBTE (green line). The difference plot shows our errors are lesser than OBTE.

by Aigerman et al. [26], we first see that the mapping is smooth  
 with our approach around the internal landmark in the chest  
 region (Fig. 12) as it is not part of the boundary. Furthermore,  
 even for the landmark on the head, which is part of the closed  
 path, the mapping remains smooth.

In Fig. 13 we show our results for the same experiment found  
 in Fig. 1 of the manuscript of the LB method [25] for meshes  
 from the SHREC dataset (meshes #315 and #318).

#### 4.4.4. Comparison to OBTE

Finally, we compared our approach to OBTE. The same four  
 landmarks were used with OBTE and our approach to ensure a  
 fair comparison. Two things can be seen with the isopoints visu-  
 alization (Fig. 14b and 14c). Firstly, the isopoints are smoother  
 with our approach, and the neck region shows that our mapping  
 is more faithful to source. In terms of mapping accuracy OBTE  
 was able to generate a precise mapping unlike the previous meth-  
 ods compared so far. The deformed source error shows that the  
 error of transferred isopoints is less with our approach. This can

**Table 1. Information about the classes, meshes, data source, number of landmarks, number of boundary landmarks, and closed paths for the results shown.**

Class	# Meshes	# Vert.	Data Source	# Boundary \ # Landmarks \ # Closed paths
Quadrupeds	7	3 - 9k	SHREC07	40\46\1
Aircrafts	6	5 - 6k	SHREC07	15\17\1
Insects	7	6 - 8k	SHREC07	35\36\1
Fish	6	4 - 10k	SHREC07	15\19\1
Birds	6	3 - 11k	SHREC07	16\18\1
Coarse Humanoids	6	8 - 15k	SHREC07	21\24\1
Busts	5	5 - 27k	SHREC07	6\16\1
Detailed Humanoids	4	4 - 13k	MakeHuman, artists	70\81\1
Pots	5	6 - 14k	SHREC07	8\9\1
Genus one	2	6 - 21k	Modelling	8\12\4
Genus two	2	6 - 21k	software primitives	18\22\4

be seen with the graph in Fig. 14d, where the green line plot shows the difference between our mapping and OBTE. OBTE is restricted to use three or four landmarks. This is limiting in scenarios where multiple detailed features need to be aligned in the mapping. In contrast, our approach supports any number of landmarks, which provides a much improved control to the user. Our method was tested on 56 objects from 10 different morphological classes. The presented approach performs robustly with a wide range of surfaces, is not limited to genus-zero surfaces, and handles surfaces with small features. Also, it works with non-isometric pairs and pairs in different poses.

## 5. Discussion

Our method was tested on 56 objects from 10 different morphological classes. Experiments are presented and compared to other methods, and our results show lower distortion and show that our approach is more robust in handling a wider range of morphologies. The presented approach performs robustly with a wide range of surfaces, is not limited to genus-zero surfaces, and handles surfaces with small features. Also, it works with non-isometric pairs and pairs in different poses.

The experiments were conducted on various meshes (see Fig. 15) from different sources: SHREC07 [36], a MakeHuman character, and artist contributions (see Table 1).

The presented approach works between any pair of meshes; there are no particular requirements, such as mesh resolution, morphology, smoothness, and pose difference, for choosing one mesh as the source. Meshes are classified based on their morphological classes such as humanoids, animals, and aircrafts.

The number of landmarks is proportional to the complexity of the features of the object class. The detailed humanoids class used the greatest number, with 81 landmarks. Table 1 shows the number of landmarks, boundary landmarks, and closed paths used for each class.

The input to the proposed approach are the user defined landmarks and a sequence to connect the landmarks for the cut-graph. The intuition behind retaining these inputs is that it gives the user sufficient control over the final mapping. Adding the user in the loop and allowing an arbitrary number of landmarks enables for

mappings that are globally good, but that can also be accurate for small features (See Fig. 16a).

The choice of landmarks and closed paths will always have an effect on the mappings. As long as the landmarks are located and connected with a closed path in meaningful fashions, the resulting mapping is reasonable. For example, Fig. 17a uses 16 landmarks while Fig. 17b has 12 landmarks. The closed paths are different, but the mappings are reasonably equivalent to each other.

Other methods such as the ones of Aigerman et al. [25, 26] also use landmarks, but create the cut-graphs automatically. Adding the user in the loop has the advantage of creating natural seams. In our experiments, we noted that when deriving the cut-graph from a minimal spanning tree, as in the work of Aigerman et al. [25, 26], the seams can lie in unnatural places, and the resulting geodesic path could loop around the meshes in conflicting directions.

Given the way we map the boundaries of the segmented parts, our mappings are  $C^0$  continuous across cuts. Furthermore, even though it is only  $C^0$  continuous, the mapping is quite smooth across the cuts (See Fig. 16b).

### 5.1. Robustness

Other experiments were conducted to test the robustness of the proposed approach. Apart from applying our approach on pairs of meshes from the same class, a biped vs. quadruped (boy and donkey) experiment was conducted. The resulting mapping did not exhibit any artifacts, except for the texture distortion near the seam lines (see Fig. 18).

Fig. 19 shows that our approach works for higher genera. The main drawback when handling higher genus meshes is the increase in the number of required landmarks and closed paths. The genus one meshes of Fig. 19(a) required 12 landmarks and 4 closed paths, while the genus two meshes of Fig. 19(b) required 22 landmarks and 4 closed paths.

An interesting property of the presented approach is that even if a segmented part is not homeomorphic to a disk (it could contain handles or holes), it will still be flattened during the planar parameterization step. The mapping will no longer be bijective, but if the areas where the discrepancies appear are relatively small, then they will lead to an overall good mapping as the errors in the mapping do not propagate throughout the surface. This can be seen in the severe test cases of Fig. 20. Fig. 21 presents a very typical example in which such robustness is important. While the meshes generally appear to have the same genus, Fig. 21(b) shows that the camel mesh has small handles. It is therefore a case of mapping between meshes of different genera, and still the mapping produced by the presented approach is of considerable quality.

### 5.2. Bijectivity

Our mapping is bijective if the initial 2D parameterizations have no fold-overs. While it is well known that ABF is not guaranteed to produce a fold-over free parameterization, in most cases it does, and even in the few cases when it does not, the fold-overs are usually small and fixable by small local relaxation steps.

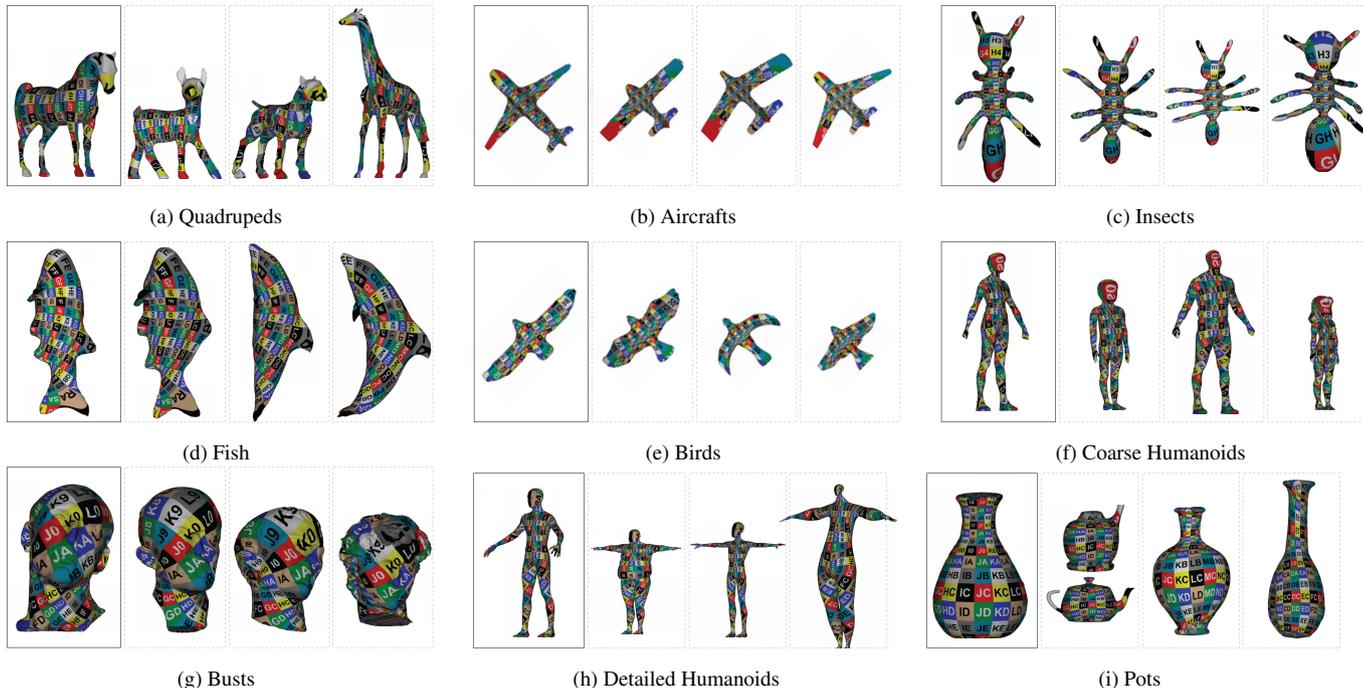


Fig. 15. Meshes used for the experiments, grouped in classes of objects. The leftmost object in each class is the source mesh (outlined with a black box) used to find a mapping with multiple targets of that class.

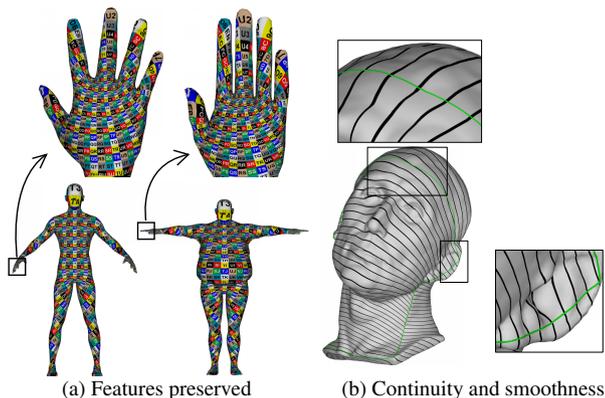


Fig. 16. (a) The figure shows how the features are preserved in a mapping. (b) The mapping is  $C^0$  continuous but still maintains smoothness without producing any odd discontinuities.

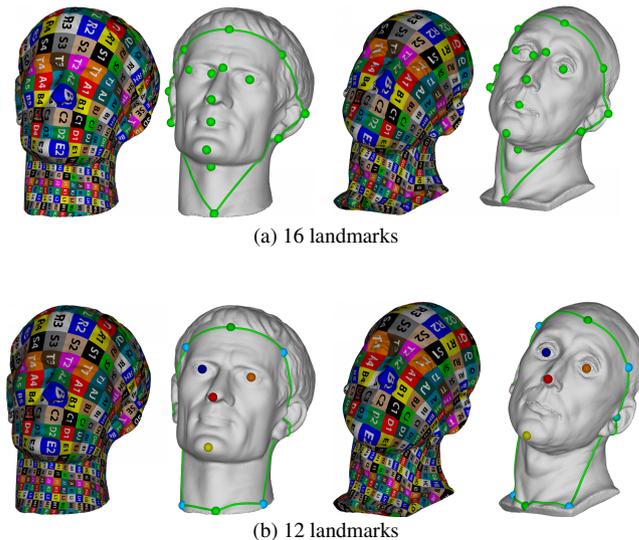


Fig. 17. Different number of landmarks and different closed paths both yield mappings that are reasonably equivalent. This demonstrates that when using our approach, it is not necessary to heavily tune the number of landmarks and their position, as well as the closed paths.

1 To show that our map is bijective we consider two observa-  
 2 tions. First, by construction, our cage mesh has no folders in both  
 3 the source and target embedding (Section 3.3). Second, since we  
 4 are using barycentric coordinates on triangular domains, trian-  
 5 gles completely contained within one cage triangle will not have  
 6 flips. However, when the vertices of a mesh triangle belong to  
 7 different cage triangles, flips can occur in rare cases. To address  
 8 this we can add these intersection vertices to the original mesh  
 9 and perform a local triangulation of the mesh. By doing this  
 10 we effectively decompose the offending triangle in a number of  
 11 smaller triangles whose vertices belong to the same cage triangle  
 12 and thus the resulting triangulation will have no flips.

### 5.3. Run-Time and Implementation Details

The experimentations were executed on a 2.2 GHz Intel Core-i7 computer with 12 GB of memory. The presented approach computes any of the mappings presented in this paper within half a minute. The time to find a mapping mostly depends on the mesh resolution (see Fig. 22).

The irregularities in the curve are mostly due to the triangulation quality. The most time consuming step is the planar

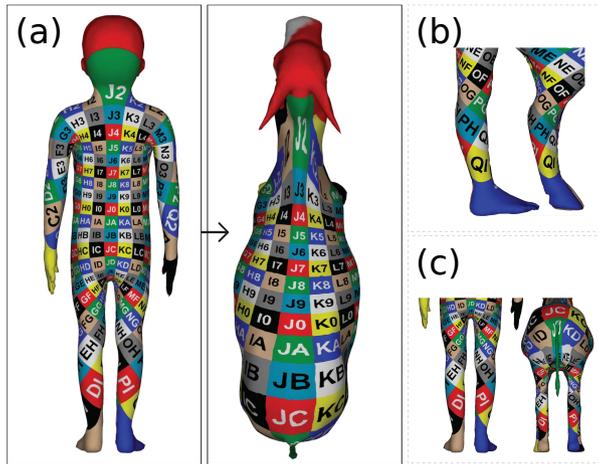


Fig. 18. Mapping with grid texture applied between two different morphological classes. The source is the boy and the target is the donkey. The mapping uses one closed path and 34 landmark pairs. Different views of the same mapping: (a) back, (b) right, and (c) rear.

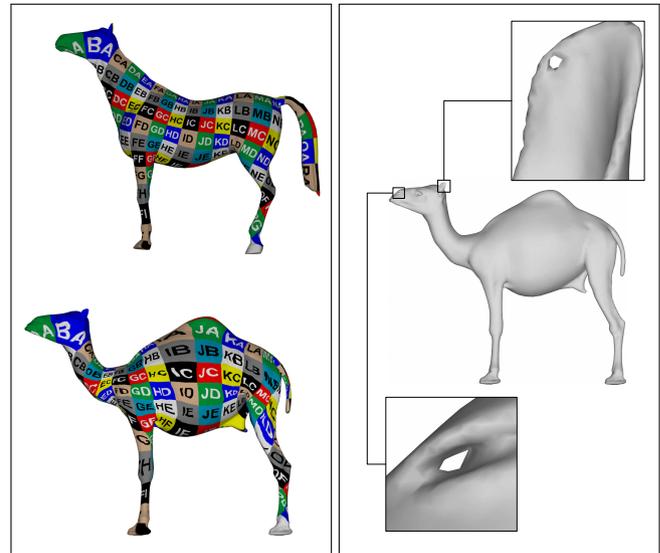


Fig. 21. Example of genus discrepancies

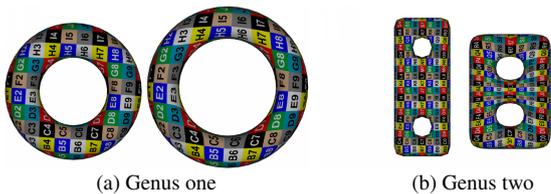


Fig. 19. Mapping of (a) genus one meshes and (b) genus two meshes.

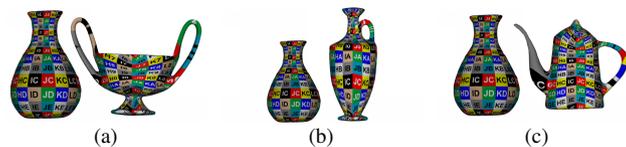


Fig. 20. Cases where a mapping is established between two objects of different genera (pots from SHREC07). These examples use nine landmarks and one closed path.

1 parameterization, followed by the segmentation (see Fig. 23).  
 2 The cage deformation is the fastest of the four steps. The pre-  
 3 sented approach was implemented in Python, within the Blender  
 4 modeling software, and relies on the Scipy and Numpy python  
 5 packages to solve the linear systems.

## 6. Conclusion

7 An approach to find a dense registration between surfaces  
 8 was presented. The approach is practical to implement, works  
 9 robustly and outperforms state of the art methods. The results  
 10 show that, given a sparse set of landmarks and closed paths  
 11 connecting these landmarks, the presented approach provides a  
 12 well behaved mapping, free of high isometric distortions.

13 Our approach introduces a process for constructing small  
 14 patches based on the closed paths. We present a dual-flattening  
 15 process that relies on the boundary of the least distorted mesh.

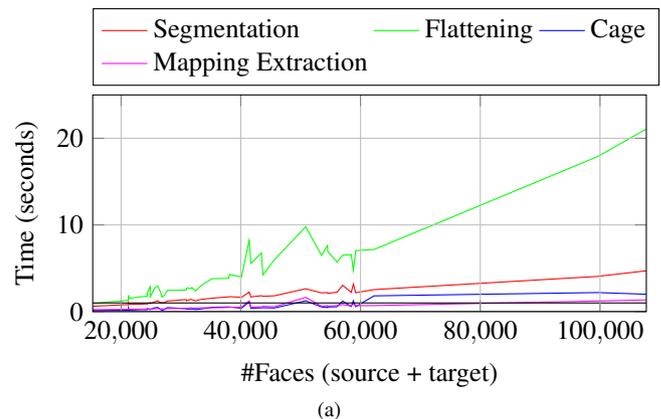


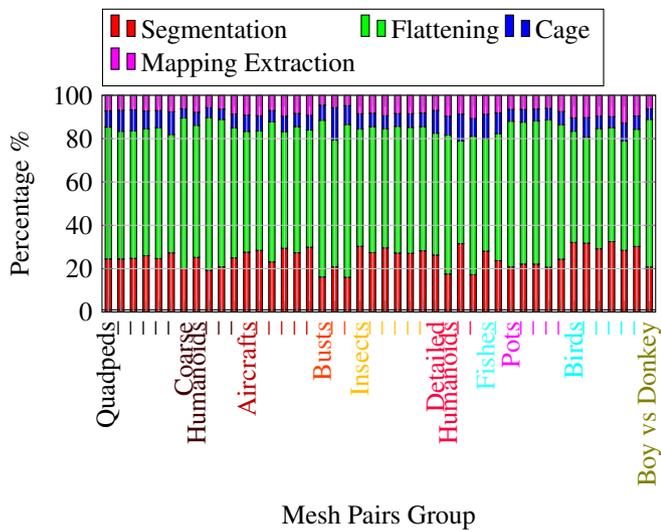
Fig. 22. Graph showing the total computation time as a function of the number of faces in the source and target meshes.

The planar map is solved using exterior boundary constraints. The cage deformation step is an interesting contribution here.

Results were evaluated with a quantitative measure we proposed using isopoints. The approach works in a robust fashion over a wide range of surfaces. It is not limited to genus-zero surfaces, and can handle even surfaces with small features (fingers and facial attributes). Furthermore, it does not impose too many constraints on the choice of surfaces: it can handle non-isometric surface pairs, pairs with different genera, and pairs set in different poses.

There are limitations in the presented approach, the first limitation being the choice of landmarks itself. For our approach and several other mapping methods, such as deformation transfer, the landmarks must be chosen with care. An automatic algorithm for finding sparse correspondences by a voting mechanism [7] or by a combinatorial tree traversal [37] could be exploited to that end, and could represent an interesting direction for future work. In our approach, the landmarks drive the construction of closed paths. Improving the closed paths with other geodesic path methods [38] is an interesting avenue for future work. Other improvements could be made to the planar parameterization step

16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36



**Fig. 23.** Graph showing the percentage of time spent on each step of the presented approach. The x-axis represents the mesh pairs arranged by their morphological class

by exploring other parameterization methods choices [39, 40].

The cage deformation approach aligns interior vertices more naturally but there is also a cost to it. In scenarios where the landmark correspondences are flipped, the cage mesh triangles will be flipped even after resolving ambiguous cages. This will cause the interior mesh triangles to overlap. This is an interesting avenue for future work.

### Acknowledgements

This work was supported by Autodesk Inc., Prompt Inc., and NSERC. We are thankful to Olivier Dionne for his support in the initial phase of the collaboration with Autodesk. We want to thank the anonymous reviewers, Joël Morency for providing 3D models, Renée Bourassa and Pierre Pellerin for their help in rigging and Shi-Qing Xin for sharing his geodesics code.

### References

- [1] Kraevoy, V, Sheffer, A. Cross-parameterization and compatible remeshing of 3D models. *ACM Trans Graph* 2004;23(3):861–869.
- [2] Avril, Q, Ghafourzadeh, D, Ramachandran, S, Fallahdoust, S, Ribet, S, Dionne, O, et al. Animation setup transfer for 3D characters. *Computer Graphics Forum* 2016;35(2):115–126.
- [3] Alexa, M. Recent advances in mesh morphing. In: *Computer graphics forum*; vol. 21. Wiley Online Library; 2002, p. 173–198.
- [4] Allen, B, Curless, B, Popović, Z. The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Trans Graph* 2003;22(3):587–594.
- [5] Bronstein, MM, Bruna, J, LeCun, Y, Szlam, A, Vandergheynst, P. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine* 2017;34(4):18–42.
- [6] Jain, V, Zhang, H, Van Kaick, O. Non-rigid spectral correspondence of triangle meshes. *Intl Journal of Shape Modeling* 2007;13:101–124.
- [7] Kim, VG, Lipman, Y, Chen, X, Funkhouser, T. Möbius transformations for global intrinsic symmetry analysis. *Computer Graphics Forum* 2010;29(5):1689–1700.
- [8] Kim, VG, Lipman, Y, Funkhouser, T. Blended intrinsic maps. *ACM Trans Graph* 2011;30(4):79:1–79:12.
- [9] Liu, T, Kim, VG, Funkhouser, T. Finding surface correspondences using symmetry axis curves. *Computer Graphics Forum* 2012;31(5):1607–1616.
- [10] Sumner, RW, Popović, J. Deformation transfer for triangle meshes. *ACM Trans Graph* 2004;23(3):399–405.

- [11] Zell, E, Botsch, M. Elastiface: Matching and blending textured faces. In: *Proc. of the Symposium on Non-Photorealistic Animation and Rendering, NPAR '13*; ACM. ISBN 978-1-4503-2198-3; 2013, p. 15–24.
- [12] Van Kaick, O, Zhang, H, Hamarneh, G, Cohen-Or, D. A survey on shape correspondence. *Computer Graphics Forum* 2011;30(6):1681–1707.
- [13] Huang, QX, Adams, B, Wicke, M, Guibas, LJ. Non-rigid registration under isometric deformations. *Comp Graph Forum* 2008;27(5):1449–1457.
- [14] Li, H, Sumner, RW, Pauly, M. Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum* 2008;27(5):1421–1430.
- [15] Lipman, Y, Funkhouser, T. Möbius voting for surface correspondence. *ACM Trans Graph* 2009;28(3):72:1–72:12.
- [16] Ovsjanikov, M, Ben-Chen, M, Solomon, J, Butscher, A, Guibas, L. Functional maps: A flexible representation of maps between shapes. *ACM Trans Graph* 2012;31(4):30:1–30:11.
- [17] Ovsjanikov, M, Corman, E, Bronstein, M, Rodolà, E, Ben-Chen, M, Guibas, L, et al. Computing and processing correspondences with functional maps. In: *SIGGRAPH ASIA 2016 Courses*. ACM; 2016, p. 9.
- [18] Alexa, M. Merging polyhedral shapes with scattered features. In: *Intl. Conf. on Shape Modeling and Applications*. IEEE; 1999, p. 202–210.
- [19] Praun, E, Hoppe, H. Spherical parametrization and remeshing. *ACM Trans Graph* 2003;22(3):340–349.
- [20] Athanasiadis, T, Fudos, I, Nikou, C, Stamati, V. Feature-based 3D morphing based on geometrically constrained spherical parameterization. *Computer Aided Geometric Design* 2012;29:2–17.
- [21] Mocanu, B, Zaharia, T. A complete framework for 3D mesh morphing. In: *Proc. of the 11th ACM SIGGRAPH Int. Conf. on Virtual-Reality Continuum and its Applications in Industry, VRCAI '12*; ACM; 2012, p. 161–170.
- [22] Bronstein, AM, Bronstein, MM, Kimmel, R. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proc of the National Academy of Sciences* 2006;103(5):1168–1172.
- [23] Kraevoy, V, Sheffer, A, Gotsman, C. Matchmaker: constructing constrained texture maps; vol. 22. ACM; 2003.
- [24] Sheffer, A, Praun, E, Rose, K. Mesh parameterization methods and their applications. *Found Trends Comput Graph Vis* 2006;2(2):105–171.
- [25] Aigerman, N, Poranne, R, Lipman, Y. Lifted bijections for low distortion surface mappings. *ACM Trans Graph* 2014;33(4):69:1–69:12.
- [26] Aigerman, N, Poranne, R, Lipman, Y. Seamless surface mappings. *ACM Trans Graph* 2015;34(4):72:1–72:13.
- [27] Aigerman, N, Kovalsky, SZ, Lipman, Y. Spherical orbifold tutte embeddings. *ACM Trans Graph* 2017;36(4):90.
- [28] Aigerman, N, Lipman, Y. Hyperbolic orbifold tutte embeddings. *ACM Trans Graph* 2016;35(6):217:1–217:14.
- [29] Kimmel, R, Sethian, JA. Computing geodesic paths on manifolds. *Proc of the National Academy of Sciences* 1998;95(15):8431–8435.
- [30] Xin, SQ, Wang, GJ. Improving chen and han's algorithm on the discrete geodesic problem. *ACM Trans Graph* 2009;28(4):104:1–104:8.
- [31] Sheffer, A, Lévy, B, Mogilnitsky, M, Bogomyakov, A. Abf++: Fast and robust angle based flattening. *ACM Trans Graph* 2005;24(2):311–330.
- [32] Bradley, D, Popa, T, Sheffer, A, Heidrich, W, Boubekur, T. Markerless garment capture. In: *ACM Trans. Graph.*; vol. 27. ACM; 2008, p. 99.
- [33] Sander, PV, Snyder, J, Gortler, SJ, Hoppe, H. Texture mapping progressive meshes. In: *Proc. of SIGGRAPH 01. Annual Conf. Series*; ACM; 2001, p. 409–416.
- [34] Anguelov, D, Srinivasan, P, Koller, D, Thrun, S, Rodgers, J, Davis, J. Scape: shape completion and animation of people. In: *ACM Trans. Graph.*; vol. 24. ACM; 2005, p. 408–416.
- [35] Aigerman, N, Lipman, Y. Orbifold tutte embeddings. *ACM Trans Graph* 2015;34(6):190–1.
- [36] Veltkamp, R, ter Haar, F. Shrec 2007-shape retrieval contest. 2007.
- [37] Zhang, H, Sheffer, A, Cohen-Or, D, Zhou, Q, Van Kaick, O, Tagliasacchi, A. Deformation-driven shape correspondence. *Computer Graphics Forum* 2008;27(5):1431–1439.
- [38] Zhuang, Y, Zou, M, Carr, N, Ju, T. Anisotropic geodesics for live-wire mesh segmentation. *Computer Graphics Forum* 2014;33(7):111–120.
- [39] Rabinovich, M, Poranne, R, Panozzo, D, Sorkine-Hornung, O. Scalable locally injective mappings. *ACM Trans Graph* 2017;36(2):16.
- [40] Schüller, C, Kavan, L, Panozzo, D, Sorkine-Hornung, O. Locally injective mappings. In: *Computer Graphics Forum*; vol. 32. Wiley Online Library; 2013, p. 125–135.