

Dense Elastic 3D Shape Matching

Frank R. Schmidt¹, Thomas Windheuser²,
Ulrich Schlickewei², and Daniel Cremers²

¹ BIOS Center of Biological Signalling Studies
University Freiburg, Germany

² Technische Universität München, Germany

Abstract. We propose a novel method for computing a geometrically consistent and spatially dense matching between two 3D shapes X and Y by means of a convex relaxation. Rather than mapping points to points we match infinitesimal surface patches while preserving the geometric structures. In this spirit, we consider matchings between objects' surfaces as diffeomorphisms which are by definition geometrically consistent. Since such diffeomorphisms can be represented as closed surfaces in the product space $X \times Y$, we are led to a minimal surface problem in a four-dimensional space. The proposed discrete formulation describes the search space with linear constraints which leads to a binary linear program. We propose an approximation approach to this potentially NP-hard problem. To overcome memory limitations, we also propose a multi-scale approach that refines a coarse matching until it reaches the finest level. As cost function for matching, we consider a thin shell energy, measuring the physical energy necessary to deform one shape into the other. Experimental results demonstrate that the proposed LP relaxation allows to compute high-quality matchings which reliably put into correspondence articulated 3D shapes. To our knowledge, this is the first solution to dense elastic surface matching which does not require an initialization and provides solutions of bounded optimality.

1 Introduction

1.1 Shape Similarity and Elastic Matching

Computing the similarity of 3D objects is among the central challenges both for humans and computers in order to structure the world around them. How can one determine that two hands are similar, that two faces are similar? A closer analysis of this problem reveals that the estimation of shape similarity is tightly coupled to the estimation of correspondence: Two hands in different articulation, for example, turn out to be similar because respective fingers all match a corresponding finger. The similarity of two given shapes can therefore be determined by finding the minimal elastic deformation which matches one shape into the other. While there exist algorithms for dense elastic surface matching, these algorithms typically only determine a locally optimal solution. They require appropriate initialization and can therefore not be employed in a fully



Fig. 1. Geometrically Consistent Elastic Matching. We propose to cast the dense elastic matching of surfaces in $3D$ as a codimension-two minimal surface problem which aims at minimizing the distortion when transforming one shape into the other. We show that a consistent discretization of this minimal surface problem gives rise to an integer linear program. By means of LP relaxation we can compute near-optimal matchings such as the one shown above. These matchings are dense triangle-wise matchings. (For visualization we combined triangles to patches and colored them consistently with their corresponding patch.)

unsupervised manner. In particular, the accurate and unsupervised comparison of more sophisticated shapes remains an important challenge.

In this work, we propose a novel framework for finding an optimal geometrically consistent matching between two surfaces. We formulate shape matching as a minimal surface problem which allows for a linear programming discretization. This model comes with a sound physical interpretation and allows to compute high-quality matching without need for initialization. In parts, this work has been presented at vision conferences [1, 2]. The goal of this paper is to combine both, the linear programming approach and the related multi-scaling approach into one comprehensive paper.

1.2 Related Work

While the matching of two different $3D$ shapes constitutes a very difficult problem, the problem becomes much easier in one less dimension. It is interesting to note that the matching of mere *planar* shapes can be solved by means of dynamic programming in runtimes which are subcubic in the number of points on each shape [3]. This is because the matching of planar shapes can be cast as a problem of finding a shortest closed curve of certain homotopy in a planar graph.

Unfortunately, the concepts of dynamic programming and variants of Dijkstra’s algorithm do not extend to the third dimension where the solution is

no longer a shortest path but a minimal closed surface in a higher-dimensional space. Therefore, existing approaches for three-dimensional shape matching typically rely on *local* optimization techniques. Our approach tries to overcome this limitation by describing these closed surfaces with linear constraints involving the boundary operator. Inspired by Sullivan [4], the boundary operator was previously introduced in the context of image segmentation by Grady [5] and Schoenemann et al. [6].

The paradigm of the Gromov–Hausdorff framework, proposed by Mémoli and Sapiro in [7], is to find the correspondence which minimizes the geodesic distortion. Bronstein et al. [8] proposed an efficient method for computing such correspondences in a coarse-to-fine strategy much akin to optical flow algorithms. In [9] the same Gromov–Hausdorff framework was merged with the idea of diffusion distances. Other approaches to shape matching employ techniques from conformal geometry [10, 11] or Riemannian geometry [12]. The physically motivated energy model we use in this work is related to the works of Litke et al. [13] and of Rumpf and Wirth [14].

All the above-mentioned methods have in common that they use a local optimization technique to minimize a non-convex energy. As a consequence, the quality of solutions depends heavily on a good initialization and an appropriately designed coarse-to-fine strategy. In addition, solutions do not come with any optimality guarantees, which implies that in principle they can be arbitrarily bad. To overcome these problems, methods with a more *global* flavor have been recently proposed.

On the one hand, Zeng and coworkers [15] formulate shape matching as a graph matching problem of third order and apply the QPBO algorithm [16]. Although the overall approach does not guarantee globally optimal solutions, it is able to detect when a proposed matching pair is globally optimal. Two major drawbacks of this approach are that firstly it suffers from a very high computational complexity, considering all triples of possible matchings. In practice it allows only the matching of a few feature points which is then post-processed with a local method. Secondly, this approach lacks a continuous counterpart, as it merely matches discrete points rather than surface elements.

On the other hand, Lipman and Daubechies [17] recently proposed to compare surfaces of genus zero and open surfaces using optimal mass transport and conformal geometry. Computationally, this amounts to solving a linear program in n^2 variables where n is the number of vertices used in the discretization of the surfaces. The problem with this approach is that no spatial regularity is imposed on the matchings.

1.3 Contribution

We propose a novel formulation for the shape matching problem based on finding an optimal surface of codimension 2 in the product of the two shape surfaces. This surface minimizes the physical deformation energy needed for deforming one shape into the other. We derive a consistent discretization of the continuous framework and show that the discrete minimal surface problem amounts to a

linear program. Compared to existing approaches the proposed framework has the following advantages:

- The LP formulation is a global approach allowing to compute matchings which are independent of initialization with no post-processing.
- The proposed method guarantees a geometrically consistent matching in the sense that the surfaces are mapped into one another in a continuous and orientation preserving manner.
- We provide a discretization of the set of diffeomorphisms by means of linear constraints. This is remarkable because in previous formulations the diffeomorphism constraint is non-linear and computationally very difficult [18].
- The algorithmic formulation is independent of the particular choice of deformation energy and can be applied universally. As an example, we show that one can also incorporate local feature similarity in order to improve performance.
- In order to be independent of potential memory limitations, we propose a multiscale-approach that starts with a coarse matching that is then refined in every iteration.
- Experiments demonstrate that reliable and dense matchings are obtained even for larger problem instances with no need for post-processing.

The paper is organized as follows. In Section 2 we present the relationship between 3D shape matching and the computation of a minimal surface in a 4D manifold. In Section 3 we present the discretization that we use in order to model arbitrary surfaces in the induced 4D space and in Section 4 we address the problem of finding an approximation of the involved integer linear program. After introducing in Section 5 a multi-resolution approach in order to also compute dense shape matching, we provide shape matching results in Section 6. Section 7 concludes this work.

2 From Continuous Shape Matching to Minimal Surfaces

One of our goal is to cast the shape matching problem as the computation of a minimal surface in a four-dimensional space. In Section 2.1 we formulate the overall energy that we want to minimize. It combines a physically motivated membrane energy with a bending energy. Subsequently, we show in Section 2.2 how this problem can be translated into an equivalent problem of finding a minimal codimension-two surface in the product space of the two involved shapes.

2.1 Shape Matching based on Minimizing Deformation Energies

In the following, we assume that the two shapes $X, Y \subset \mathbb{R}^3$ are differentiable, oriented, closed surfaces. While most 3D shape matching approaches like to interpret a matching just as a bijective mapping between the surface points of these shapes, we pursue a fundamentally different approach. The main reason is

that in general, bijections do not respect the underlying two-dimensional structure of surfaces. In fact, there are even continuous bijections between a line and a two-dimensional patch like the continuous *space-filling curve* as shown by Hilbert [19]. Therefore, we propose to search for diffeomorphisms instead of bijections.

Diffeomorphisms $f : X \rightarrow Y$ are bijections for which both, f and f^{-1} are differentiable. This does not only cope with the dimensionality problem presented above, but it also helps us to propose an energy function that is symmetric in f and f^{-1} . Since both f and its inverse exist, the optimal matching f between X and Y also gives rise to the optimal matching between Y and X , namely $f^{-1} : Y \rightarrow X$. The set of diffeomorphisms $\text{Diff}(X, Y)$ can be separated in two different classes, into the class of orientation preserving diffeomorphisms $\text{Diff}^+(X, Y)$ and the class of orientation reversing diffeomorphisms $\text{Diff}^-(X, Y)$.

In the following, we formulate the shape matching problem as an optimization problem over $\text{Diff}^+(X, Y)$:

$$\min_{f \in \text{Diff}^+(X, Y)} E(f) + E(f^{-1}) \quad (1)$$

where E is a suitable energy on the class of all diffeomorphisms between surfaces. Note that we choose a symmetric problem formulation, penalizing at the same time deformation energy of X into Y and of Y into X . This is necessary because usually E takes different values on f and on f^{-1} .

The energy functional we use is borrowed from elasticity theory in physics [20]. Here, we interpret the shapes X and Y as surfaces or “thin shells”. If we now try to find the deformation of X into Y , it requires a certain amount of stretching and bending. This results in an energy that usually combines a membrane energy E_{mem} and a bending energy E_{bend} penalizing deformations in the first and in the second fundamental forms of the surfaces. In this work we use the following formulation:

$$E(f) = \underbrace{\int_X (\text{tr}_{g_X} \mathbf{E}) + \mu \text{tr}_{g_X} (\mathbf{E}^2)}_{E_{\text{mem}}} + \lambda \underbrace{\int_X (H_X(x) - H_Y(f(x)))^2}_{E_{\text{bend}}} \quad (2)$$

where $\mathbf{E} = f^*g_Y - g_X$ is the difference between the metric tensors of X and Y , typically called the *Lagrange strain tensor*, $\text{tr}_{g_X}(\mathbf{E})$ is the norm of this tensor (see [21]), H_X and H_Y denote the mean curvatures and μ and λ are parameters which determine the elasticity and the bending property of the material. This energy is a slightly simplified version of Koiter’s thin shell energy [22].

After presenting the overall energy $E(f) + E(f^{-1})$ that we want to minimize, we will reformulate this problem in the next section into a problem of finding a minimal surface in a four-dimensional space.

2.2 Shape Matchings and their Graph Surfaces

As we mentioned in Section 2.1, every matching function $f : X \rightarrow Y$ between two shapes X and Y is an orientation preserving diffeomorphism. Given such a

matching, its graph

$$\Gamma = \{(x, f(x)) \mid x \in X\} \quad (3)$$

becomes a surface in the four-dimensional³ product space $X \times Y$. This surface Γ comes with the two natural projections:

$$\begin{array}{ll} \pi_X : \Gamma \rightarrow X & \pi_Y : \Gamma \rightarrow Y \\ (x, y) \mapsto x & (x, y) \mapsto y \end{array}$$

that will help us to characterize a diffeomorphism completely by its graph:

Proposition 1 (Graph Surfaces). *Let Γ be the graph of a diffeomorphism $f : X \rightarrow Y$. Then*

1. Γ is a differentiable, connected, closed surface in the product space $X \times Y$.
2. The projections π_X and π_Y are both diffeomorphisms.
3. The two orientations which Γ naturally inherits from X and Y coincide.

*Vice versa, any surface $\Gamma \subset X \times Y$ which satisfies Conditions 1–3 is the graph of an orientation-preserving diffeomorphism between X and Y . We call such surfaces **graph surfaces**.*

The energy $E(f) + E(f^{-1})$ can be expressed as $\tilde{E}(\Gamma)$ via

$$E(f) + E(f^{-1}) = E(\pi_Y \circ (\pi_X)^{-1}) + E(\pi_X \circ (\pi_Y)^{-1}) =: \tilde{E}(\Gamma). \quad (4)$$

Concluding the above discussion, we have transformed the optimization problem (1) into an optimization problem over the set of all *graph surfaces* in $X \times Y$, namely

$$\begin{array}{ll} \min & \tilde{E}(\Gamma) \\ \text{subject to} & \Gamma \subset X \times Y \text{ is a graph surface} \end{array} \quad (5)$$

We remark that the idea of casting optimal diffeomorphism problems as minimal surface problems has been applied previously in the theory of nonlinear elasticity [23]. In the setup of shape matching, it is related to the approach that Tagare [24] proposed for the matching of 2D shapes. Its connection to orientation preserving diffeomorphisms was made in [3].

3 The Discrete Setting

In this section we develop a discrete representation of graph surfaces introduced in Section 2.2. We start in Section 3.1 with the definition of discrete surface patches in $X \times Y$. These patches are derived from a given discrete triangulation of X and Y itself. The surface patches in the product space $X \times Y$ are the building blocks for discrete graph surfaces that we introduce in Section 3.2. Finally in Section 3.3 we give a discrete version of the energy minimization problem (5).

³ $X \times Y$ is a 4-manifold embedded in \mathbb{R}^6 .

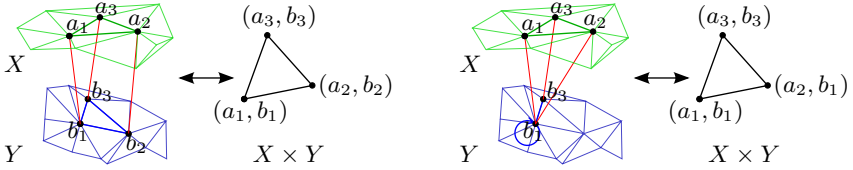


Fig. 2. Product Triangles. To assure a geometrically consistent, elastic matching from mesh X to mesh Y , we define a space of feasible solutions which is spanned by a set of 45 basic matchings among triangles, edges and vertices on either mesh. Two representative matchings and their corresponding representation in the product space $X \times Y$ are shown. **Left image:** The triangle $(a_1, a_2, a_3)^T$ on surface X is matched to triangle $(b_1, b_2, b_3)^T$ on Y by assigning vertex a_i to vertex b_i . This directly corresponds to the triangle with vertices (a_i, b_i) in the product graph. **Right image:** The triangle $(a_1, a_2, a_3)^T$ is matched to the edge $(b_1, b_3)^T$, represented here as degenerate triangle $(b_1, b_1, b_3)^T$.

3.1 Discrete Surface Patches

In the following, we assume that a shape X is given as a triangulated oriented surface mesh $G_X = (V_X, E_X, F_X)$, consisting of a set of vertices V_X , a set of directed edges E_X and a set of oriented triangles F_X .

While the orientation of X defines a natural orientation of the faces in F_X , such a natural orientation does not exist for the edge in E_X . Moreover, two faces $f_1, f_2 \in F_X$ that touch each other along an edge $e \in E_X$ induce opposite orientations onto this edge e . Since edges on X do not have a preferable orientation, we fix an orientation for each edge on X . Thus, whenever two vertices a_1 and a_2 of X are connected by an edge, either

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \in E_X \quad \text{or} \quad \begin{pmatrix} a_2 \\ a_1 \end{pmatrix} = - \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \in E_X.$$

For simplicity, we extend the set of edges by *degenerate edges*

$$\bar{E}_X = E_X \cup \left\{ \begin{pmatrix} a \\ a \end{pmatrix} \mid a \in V_X \right\}. \quad (6)$$

By assumption, the triangular faces of X are oriented. If the vertices a_1, a_2, a_3 build an oriented triangle on X , then

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} a_2 \\ a_3 \\ a_1 \end{pmatrix} = \begin{pmatrix} a_3 \\ a_1 \\ a_2 \end{pmatrix} \in F_X$$

and analogously to the edges, we extend the set of triangles by *degenerate triangles*

$$\bar{F}_X = F_X \cup \left\{ \begin{pmatrix} a_1 \\ a_2 \\ a_2 \end{pmatrix} \mid a_1, a_2 \in V_X, \pm \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \in \bar{E}_X \right\}. \quad (7)$$

Due to the definition of \overline{E}_X , degenerate triangles can consist of two vertices or even of only one vertex. The existence of these degenerate triangles will allow stretching or compression of parts of the shapes as we will see below (cf. right image of Figure 2 and Figure 6).

Next, we define triangles F , edges E and vertices V that operate as building blocks for the graph surfaces introduced in Section 2.2. To this end, let us assume that two shapes X and Y are given as triangulated oriented surface meshes $G_X = (V_X, E_X, F_X)$ resp. $G_Y = (V_Y, E_Y, F_Y)$ and that $\overline{E}_X, \overline{F}_X, \overline{E}_Y, \overline{F}_Y$ are defined as above. Then define the graph $G = (V, E, F)$ of the product space $X \times Y$ by

$$\begin{aligned} V &:= V_X \times V_Y \\ E &:= \overline{E}_X \times \overline{E}_Y \\ F &:= (\overline{F}_X \times F_Y) \cup (F_X \times \overline{F}_Y) \end{aligned}$$

The product triangles in F are the pieces which are later glued together in order to obtain discrete graph surfaces. For shape matching, a product triangle

$$\begin{pmatrix} (a_1, b_1) \\ (a_2, b_2) \\ (a_3, b_3) \end{pmatrix} \in F \quad (8)$$

is interpreted as setting vertex $a_i \in V_X$ in correspondence with vertex $b_i \in V_Y$. While a triangle provides us with such a point-to-point matching, it also takes care of the geometric structure within the two shapes X and Y . In that sense it is more powerful than a mere point matching.

Given two non-degenerate triangles $a \in F_X$ and $b \in F_Y$, we allow for 45 different matchings between them:

- 3 orientation-preserving bijective matchings,
- 36 triangle-to-edge matchings and
- 6 triangle-to-vertex matchings.

The degenerate triangle-to-edge and triangle-to-vertex matchings allow us to handle infinitesimal stretching and compression in the proposed framework. Visualizations for two of the 45 possibilities is given in Figure 2.

3.2 Discrete Surfaces

Following Section 2, a diffeomorphism can be represented as a surface $\Gamma \subset X \times Y$ satisfying conditions 1–3 of Proposition 1. In this section we derive discrete versions of these three properties. First, we define a surface in the product space:

Definition 1. *A discrete surface in $G = (V, E, F)$ is a subset $\Gamma \subset F$. The set of all discrete surfaces is denoted by $\text{surf}(G)$.*

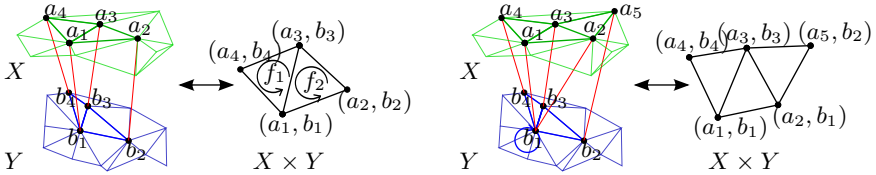


Fig. 3. Geometric Consistency. To ensure that neighboring triangles on X are matched with neighboring triangles on Y , we impose the closedness condition. **Left image (general case):** The triangles $(a_1, a_2, a_3)^T$ and $(b_1, b_2, b_3)^T$ are matched, thereby activating f_2 . The boundary condition $\partial\Gamma = 0$ ensures that the matching continues with a correspondence whose triangles in X and Y are positively incident to $(a_1, a_3)^T$ and $(b_1, b_3)^T$ respectively. This constraint is satisfied for example by triangle f_1 which is visualized here. **Right image (stretching):** The stretching is achieved by matching triangle $(a_1, a_2, a_3)^T$ to edge $(b_3, b_1)^T$. Again, the geometric consistency is granted by the boundary condition evaluated on the product edges $((a_2, b_1), (a_3, b_3))^T$ and $((a_3, b_3), (a_1, b_1))^T$.

As we have seen in Section 3.1, any triangle in F can be interpreted as matching a (possibly degenerated) triangle of \overline{F}_X to a (possibly degenerated) triangle of \overline{F}_Y . Thus, the intuitive meaning of a discrete surface $\Gamma \subset F$ is a set of point correspondences between the shapes X and Y . Imposing the discrete counterparts of 1–3 on such a discrete surface will result in a geometrically consistent matching that approximates a diffeomorphism between the continuous counter-parts of X and Y .

Discrete Version of Condition 1 In the following we will find a condition which guarantees the continuity of our matching. Recall that the boundary operator for triangle meshes [25] maps triangles to their oriented boundary. We extend this definition to the product graph G .

As for the sets E_X and E_Y we choose arbitrary orientations for each product edge $e \in E$. By means of these orientations we define for any edge $(v_1, v_2)^T \in E$ connecting two vertices $v_1, v_2 \in V$ a vector $O((v_1, v_2)^T) \in \mathbb{Z}^{|E|}$ whose e -th entry is given by

$$O((v_1, v_2)^T)_e = \begin{cases} 1 & \text{if } e = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \\ -1 & \text{if } e = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \\ 0 & \text{else.} \end{cases} \quad (9)$$

The triangles in F naturally inherit orientations from the triangles in F_X and F_Y . This allows us to define the boundary operator as follows.

Definition 2. The boundary operator $\partial : F \rightarrow \mathbb{Z}^{|E|}$ is defined by

$$\partial \begin{pmatrix} (a_1, b_1) \\ (a_2, b_2) \\ (a_3, b_3) \end{pmatrix} := O \begin{pmatrix} (a_1, b_1) \\ (a_2, b_2) \end{pmatrix} + O \begin{pmatrix} (a_2, b_2) \\ (a_3, b_3) \end{pmatrix} + O \begin{pmatrix} (a_3, b_3) \\ (a_1, b_1) \end{pmatrix}, \quad (10)$$

where the $a_i \in V_X$ and $b_i \in V_Y$ form triangles on X resp. on Y and $\binom{(a_i, b_i)}{(a_j, b_j)} \in E$ connects the vertices $(a_i, b_i) \in V$ with $(a_j, b_j) \in V$. The boundary operator is linearly extended to a map

$$\partial : \text{surf}(G) \rightarrow \mathbb{Z}^{|E|}. \quad (11)$$

and a discrete surface Γ in G is **closed** if $\partial\Gamma = 0$.

The closedness condition ensures that adjacent triangles on X are in correspondence with adjacent triangles on Y and therefore guarantees the geometric consistency (see Figure 3). The natural discrete version of Condition 1 is a closed, connected discrete surface in G .

Discrete Version of Condition 2 Analogously to the continuous case, we can project product triangles of F to triangles of the surfaces X and Y .

Definition 3. The projection $\pi_X : F \rightarrow \mathbb{Z}^{|F_X|}$ is defined by

$$\pi_X \begin{pmatrix} (a_1, b_1) \\ (a_2, b_2) \\ (a_3, b_3) \end{pmatrix} := \begin{cases} e_a & \text{if } a = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \in F_X \\ (0, \dots, 0) & \text{else} \end{cases}. \quad (12)$$

Here, e_a is the base vector with 1 in the a -entry and 0 in all other entries. We extend the projection π_X linearly to a map $\pi_X : \text{surf}(G) \rightarrow \mathbb{Z}^{|F_X|}$. The projection $\pi_Y : F \rightarrow \mathbb{Z}^{|F_Y|}$ and its linear extension $\pi_Y : \text{surf}(G) \rightarrow \mathbb{Z}^{|F_Y|}$ are defined analogously.

Let now Γ be a discrete surface in G . Then we say that the projections of Γ to X and Y are discrete diffeomorphisms if and only if

$$\pi_X(\Gamma) = (1, \dots, 1) \in \mathbb{Z}^{|F_X|} \quad \text{and} \quad \pi_Y(\Gamma) = (1, \dots, 1) \in \mathbb{Z}^{|F_Y|}. \quad (13)$$

This gives a discrete version of Condition 2.

Note that in this definition we do not ask for injectivity on the vertices set. This is necessary for modelling discretely strong compressions. However, conditions (13) ensure a global bijectivity property which is sufficient in our context.

Discrete Version of Condition 3 By definition, the set of surfaces in G only contains surface patches which are consistently oriented. Therefore any surface in $\text{surf}(G)$ satisfies Condition 3.

Definition 4. Let $\Gamma \in \{0, 1\}^{|F|}$ be a discrete surface in G , represented by its indicator vector. Then Γ is a **discrete graph surface** in G if

$$\begin{pmatrix} \partial \\ \pi_X \\ \pi_Y \end{pmatrix} \cdot \Gamma = \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{1} \end{pmatrix}. \quad (14)$$

3.3 Discrete Surface Energy

Now we introduce a discrete energy on the set of product triangles in G . For the membrane energy in (2) we adopt the term proposed by Delingette [26]. Given two triangles $T_1, T_2 \subset \mathbb{R}^3$, Delingette computes the stretch energy $E_{\text{mem}}(T_1 \rightarrow T_2)$ necessary for deforming T_1 in T_2 . In our framework we associate with each product triangle $(a, b) \in F$ consisting of $a = (a_1, a_2, a_3)^\top \in \overline{F}_X$ and $b = (b_1, b_2, b_3)^\top \in \overline{F}_Y$ the membrane cost

$$E_{\text{mem}}(a, b) := E_{\text{mem}} \left(\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \rightarrow \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \right) + E_{\text{mem}} \left(\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \rightarrow \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \right). \quad (15)$$

For the bending term we proceed similarly associating with each product triangle (a, b) the cost

$$E_{\text{bend}}(a, b) = \int_a (H_X - H_Y)^2 + \int_b (H_Y - H_X)^2. \quad (16)$$

In practice we discretize the mean curvature following [27].

Next, we extend the energy linearly from discrete surface patches to discrete surfaces in G . Identify a discrete surface with its indicator vector $\Gamma \in \{0, 1\}^{|F|}$. Define the vector $E \in \mathbb{R}^{|F|}$ whose f -th entry is

$$E_f = E_{\text{mem}}(f) + E_{\text{bend}}(f). \quad (17)$$

Then the **discrete energy** of Γ is given by the vector product

$$E^t \cdot \Gamma. \quad (18)$$

4 Linear Programming Solution

In the previous section we have introduced a discrete notion of graph surfaces (14) and a discrete deformation energy (18) for such graph surfaces. This enables us to state the discrete version of (5) in the form of a binary linear program:

$$\begin{aligned} & \min_{\Gamma \in \{0, 1\}^{|F|}} E^t \cdot \Gamma \\ & \text{subject to} \quad \begin{pmatrix} \partial \\ \pi_X \\ \pi_Y \end{pmatrix} \cdot \Gamma = \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{1} \end{pmatrix}. \end{aligned} \quad (19)$$

For solving (19), we relax the binary constraints to $\Gamma \in [0, 1]^{|F|}$. This relaxed version can be solved globally optimally in polynomial time. We employed an alternating direction method developed by Eckstein et al. [29]. This algorithm is parallelizable which allowed us an efficient implementation on the GPU.

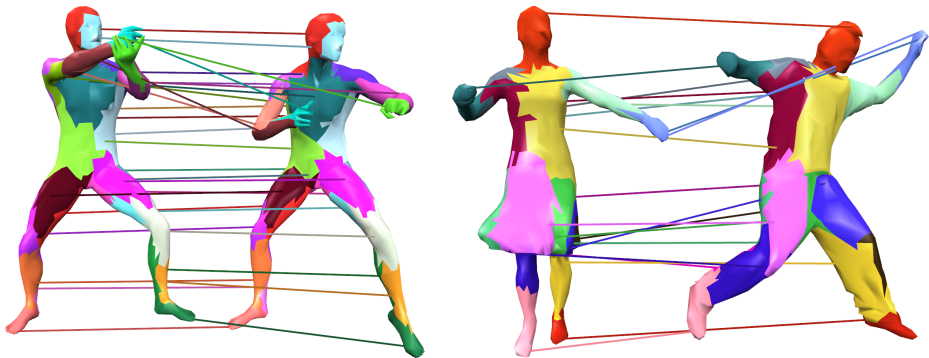


Fig. 4. SHREC 2011 benchmark and matching different objects. **Left:** The image illustrates the performance of the proposed method on the SHREC 2011 [28] dataset. **Right:** Matching of two different objects. While it is not well defined what a good matching between a skirt and trousers is, it is really remarkable how well the proposed algorithm finds a matching that apparently minimizes the deformation energy.

Since the constraint matrix of the relaxed problem is not totally unimodular, we are not guaranteed an integral solution. A simple thresholding scheme would destroy the geometric consistency of the solution. Instead, we propose an iterative scheme: solve the relaxed version of (19), fix the variables with values above a threshold > 0.5 to 1 and solve the relaxed version of (19) with these additional constraints. If there is no variable with value above the threshold fix only one variable with the highest value. In our experiments, this scheme typically converged to a binary solution after less than 10 iterations, in no experiment it took more than 20 iterations.

5 Multiresolution Framework

Because the number of product triangles grows quadratically with the number of triangles in both shapes the resulting Integer Linear Program (ILP) has a very high number of variables and constraints. Even the minimization of the relaxed Linear Program (LP) becomes impractical for state-of-the-art LP solvers, if the shapes have more than 250 triangles. In this section we present a multiresolution approach that overcomes this limitation and allows to match shapes of more than 2000 triangles.

The basic idea of the multiresolution approach is to solve the problem at a very coarse scale with the methods described in Section 4 and to recursively use the found solution to narrow the search space at the next finer level. To reduce the size of the search space we impose that a possible solution at a finer level must lie "near" an already found solution at the coarser scale.

For the definition of "near" we use a hierarchy of triangles across the resolution levels. Suppose that we obtain a triangle mesh X_i from a finer trian-

gle mesh X_{i+1} by repeatedly merging triangles. In practice we use the quadric edge decimation algorithm [30] in its OpenMesh implementation [31]. Denote by $\chi_i : F(X_{i+1}) \rightarrow F(X_i)$ the child-parent relation, mapping each triangle in $F_{X_{i+1}}$ to the triangle it is merged to on the coarser mesh X_i . These maps are extended to maps between the extended sets of triangles $\chi_i : \overline{F}_{X_{i+1}} \rightarrow \overline{F}_{X_i}$ (see Section 3.2).

Let now X and Y be two high-resolution meshes and let $X = X_n, X_{n-1}, \dots, X_0$ and $Y = Y_n, Y_{n-1}, \dots, Y_0$ be successive coarsenings with corresponding child-parent relations $\{\chi_i\}_{0 \leq i \leq n-1}$ and $\{\psi_i\}_{0 \leq i \leq n-1}$. We proceed as follows:

1. We compute a discrete graph surface Γ_0 (cf. Definition 4) inducing a matching of the coarsest meshes X_0 and Y_0 . We use the methods described in Section 4 for this task.
2. Assuming inductively that we have found a discrete graph surface Γ_i which induces a matching of X_i and Y_i , we search for a discrete graph surface Γ_{i+1} . This surface has to lie in a search space which is reduced using the input of the already computed surface Γ_i . Rather than allowing Γ_{i+1} to be built of all product triangles F_{i+1} between X_{i+1} and Y_{i+1} , we only allow for product triangles whose parents or whose parents' neighbors are set in correspondence by Γ_i . Thus, Γ_{i+1} is searched as a subset of the reduced set of product triangles

$$F_{i+1}^{\text{red}} = \left\{ (f_a, f_b) \in F_{i+1} \left| \begin{array}{l} \exists (f'_a, f'_b) \in \Gamma_i \subset F_i \text{ s.t.} \\ \chi_i(f_a) \in \mathcal{N}(f'_a) \text{ and} \\ \psi_i(f_b) \in \mathcal{N}(f'_b) \end{array} \right. \right\}. \quad (20)$$

Here, for a triangle f on a triangle mesh we used the set of triangles in the one-ring or the two-ring of its vertices as neighborhood $\mathcal{N}(f)$.

Then we compute Γ_{i+1} by solving problem (19) over the reduced search space, that is $\Gamma_{i+1} \in \{0, 1\}^{|F_{i+1}^{\text{red}}|}$.

3. We repeat Step 2 until a discrete graph surface Γ_n has been computed which induces a matching between X and Y .

6 Experimental Results

We have introduced a framework for computing geometrically consistent elastic matchings between 3D shapes using LP relaxation. We evaluated the proposed method on several shapes taken from the SHREC 2011 benchmark [28] and a dataset by Vlastic et al. [32].

6.1 Matching of Articulated Shapes

A common problem in shape matching is that the same shape may undergo substantial deformation and articulation. Nevertheless, one would like to reliably identify corresponding structures. Figures 1 and 7 show the matchings



Fig. 5. Multiresolution Framework. The images show matchings between two shapes at different resolutions. As described in Section 5 the computational cost is drastically reduced by first solving the matching problem at a lower resolution and then using the obtained solution to restrict the search space at the next higher resolution.

computed for models of different articulations. Although the movement of arms and legs deform the shapes drastically the proposed method identifies the correct matchings. Since the proposed framework enforces geometric consistency matching errors occur only on a small spatial scale. In contrast to methods without spatial regularization strong outliers such as single points matched to the wrong leg do not arise.

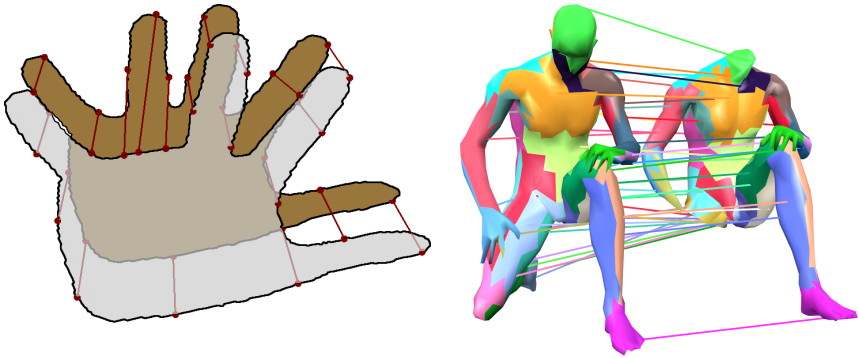
6.2 Partial Matching

The ability of the proposed method to model stretching and shrinking also allows to match shapes where large parts of the geometry are missing. The right image of Figure 6 demonstrates this ability experimentally. The proposed algorithm matches the remaining parts of a human body missing a hand, a leg and the head to its original shape.

6.3 Quantitative Evaluation

We quantitatively evaluated the proposed method on 30 pairs of models from Vlasic et al. [32] by computing the mean geodesic error. One of the matchings is visualized in Figure 1. Computing each of the matchings took about 2 hours. The results were compared to matchings generated by the GMDS method of Bronstein et al. [8] using their code.

Given two meshes X, Y and the available ground truth correspondences (x_i, y_i) we defined the mean geodesic error of a matching $\varphi : X \rightarrow Y$ by $\frac{1}{N} \sum_i d(\varphi(x_i), y_i)$, where d is the normalized geodesic distance on the manifold of mesh Y . The mean geodesic error produced by GMDS (using their code) was 0.079 while the proposed method had a mean geodesic error of 0.03.



Elastic matching of planar shapes [3] Proposed elastic matching of 3D shapes

Fig. 6. 2D and 3D Shape Matching. While the elastic matching of planar shapes can be solved in polynomial time as a minimal cyclic path on a torus [3], the framework developed in this paper allows to compute an elastic matching of 3D shapes via linear programming relaxation. In both cases, missing parts can be accounted for due to the elasticity.



Fig. 7. Linear Interpolation. The images show a matching between the leftmost and rightmost models taken from the SHREC 2011 benchmark [28] by linearly interpolating between the triangle correspondences. This transition illustrates the geometric consistency property of the proposed method: At any time during the interpolation the model is a closed triangle mesh.

Of course, this experiment does not pretend to be an exhaustive comparison against all methods in the literature. Nonetheless it shows, that the proposed method can compete with state-of-art matching algorithms in terms of accuracy while guaranteeing geometrically consistent solutions.

7 Conclusion

We proposed a new framework for finding a geometrically consistent matching of 3D shapes which minimizes an elastic deformation energy. The approach is based on finding discrete minimal surfaces which set infinitesimal surface patches on both shapes into correspondence. In that sense the framework manages to generalize the planar shape matching to the more complex 3D shape matching.

While the planar shape matching finds correspondences between infinitesimal line elements of two contours, the 3D shape matching sets infinitesimal surface elements in correspondence. We showed that a consistent discretization leads to an integer linear program. As a consequence, we can compute high-quality solutions to the matching problem which are independent of initialization by means of LP relaxation.

To improve the runtime and overcome possible memory limitations, we also introduced a multi-scale approach that improves iteratively matchings from a coarse level to its finest level. Experimental results confirm that the proposed method generates reliable dense correspondences for a variety of articulated real-world shapes.

Acknowledgments

The 3D shape data in Figures 1, 4 (right) and 5 is courtesy of Vlasic et al. [32]. The 3D shape data in Figures 4 (left), 6 and 7 is taken from the SHREC 2011 benchmark [28].

References

1. Windheuser, T., Schlickewei, U., Schmidt, F.R., Cremers, D.: Geometrically consistent elastic matching of 3D shapes: A linear programming solution. In: Proc. International Conference on Computer Vision. (2011)
2. Windheuser, T., Schlickewei, U., Schmidt, F.R., Cremers, D.: Large-scale integer linear programming for orientation-preserving 3D shape matching. In: Computer Graphics Forum (Proceedings Symposium Geometry Processing), Lausanne, Switzerland, Eurographics (2011)
3. Schmidt, F.R., Farin, D., Cremers, D.: Fast matching of planar shapes in sub-cubic runtime. In: Proc. International Conference on Computer Vision, Rio de Janeiro (October 2007)
4. Sullivan, J.M.: A Crystalline Approximation Theorem for Hypersurfaces. PhD thesis, Princeton University (October 1990)
5. Grady, L.: Minimal surfaces extend shortest path segmentation methods to 3d. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(2) (2010) 321–334
6. Schoenemann, T., Kahl, F., Cremers, D.: Curvature regularity for region-based image segmentation and inpainting: A linear programming relaxation. In: Proc. International Conference on Computer Vision, Kyoto (2009)
7. Mémoli, F., Sapiro, G.: A theoretical and computational framework for isometry invariant recognition of point cloud data. *Foundations of Computational Mathematics* **5**(3) (2005) 313–347
8. Bronstein, A., Bronstein, M., Kimmel, R.: Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Sci. Comput.* **28**(5) (2006) 1812–1836 (electronic)
9. Bronstein, A., Bronstein, M.M., Kimmel, R., Mahmoudi, M., Sapiro, G.: A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *International Journal of Computer Vision* **89**(2-3) (2010) 266–286

10. Wang, Y., Gu, X., Hayashi, K., Chan, T., Thompson, P., Yau, S.: Brain surface parameterization using Riemann surface structure. In: MICCAI (2). (2005) 657–665
11. Lui, L., Wong, T., Thompson, P., Chan, T., Gu, X., Yau, S.: Shape-based diffeomorphic registration on hippocampal surfaces using Beltrami holomorphic flow. In: MICCAI (2). (2010) 323–330
12. Kurtek, S., Klassen, E., Ding, Z., Srivastava, A.: A novel Riemannian framework for shape analysis of 3D objects. In: CVPR. (2010) 1625–1632
13. Litke, N., Droske, M., Rumpf, M., Schröder, P.: An image processing approach to surface matching. In: Symposium on Geometry Processing. (2005) 207–216
14. Wirth, B., Bar, L., Rumpf, M., Sapiro, G.: Geodesics in shape space via variational time discretization. In: EMMCVPR'09. Volume 5681 of LNCS. (2009) 288–302
15. Zeng, Y., Wang, C., Wang, Y., Gu, X., Samaras, D., Paragios, N.: Dense non-rigid surface registration using high-order graph matching. In: CVPR. (2010) 382–389
16. Torresani, L., Kolmogorov, V., Rother, C.: Feature correspondence via graph matching: Models and global optimization. In: ECCV (2). (2008) 596–609
17. Lipman, Y., Daubechies, I.: Surface comparison with mass transportation (2009) Preprint, arXiv:0912.3488.
18. Younes, L.: Shapes and diffeomorphisms. Volume 171 of Applied Mathematical Sciences. Springer-Verlag, Berlin (2010)
19. Hilbert, D.: Über die stetige Abbildung einer Linie auf ein Flächenstück. *Math. Ann.* **38** (1891) 459–460
20. Ciarlet, P.: An introduction to differential geometry with applications to elasticity. Springer, Dordrecht (2005) Reprinted from *J. Elasticity* **78/79** (2005), no. 1-3.
21. Do Carmo, M.: Riemannian geometry. Birkhäuser (1992)
22. Koiter, W.: On the nonlinear theory of thin elastic shells. I, II, III. *Nederl. Akad. Wetensch. Proc. Ser. B* **69** (1966) 1–17, 18–32, 33–54
23. Giaquinta, M., Modica, G., Souček, J.: Cartesian currents in the calculus of variations. II. Volume 38 of *Ergebnisse der Mathematik und ihrer Grenzgebiete*. Springer-Verlag, Berlin (1998) Variational integrals.
24. Tagare, H.: Shape-based nonrigid correspondence with application to heart motion analysis. *IEEE Transactions on Medical Imaging* **18**(7) (1999) 570–579
25. Desbrun, M., Hirani, A.N., Leok, M., Marsden, J.E.: Discrete exterior calculus (2005) Preprint, arXiv:0508341v2.
26. Delingette, H.: Triangular springs for modeling nonlinear membranes. *IEEE Transactions on Visualisation and Computer Graphics* **14**(2) (March/April 2008)
27. Meyer, M., Desbrun, M., Schröder, P., Barr, A.: Discrete differential-geometry operators for triangulated 2-manifolds. In: *Proc. VisMath.* (2002) 35–57
28. Boyer, E., Bronstein, A.M., Bronstein, M.M., Bustos, B., Darom, T., Horaud, R., Hotz, I., Keller, Y., Keustermans, J., Kovnatsky, A., Litman, R., Reininghaus, J., Sipiran, I., Smeets, D., Suetens, P., Vandermeulen, D., Zaharescu, A., Zobel, V.: SHREC 2011: robust feature detection and description benchmark (February 2011) Preprint, arXiv:1102.4258.
29. Eckstein, J., Bertsekas, D.: An alternating direction method for linear programming. Technical report, Harvard Business School (1990)
30. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: *SIGGRAPH.* (1997) 209–216
31. Botsch, M., Steinberg, S., Bischoff, S., Kobbelt, L.: Openmesh - a generic and efficient polygon mesh data structure. In: *OpenSG Symposium 2002.* (2002)
32. Vlasic, D., Baran, I., Matusik, W., Popović, J.: Articulated mesh animation from multi-view silhouettes. In: *ACM SIGGRAPH 2008 papers*, ACM (2008) 1–9