

Large-Scale Integer Linear Programming for Orientation Preserving 3D Shape Matching

Thomas Windheuser¹, Ulrich Schlickwei¹, Frank R. Schimdt², Daniel Cremers¹¹ Technische Universität München, Germany, ² University of Western Ontario, Canada

Abstract

We study an algorithmic framework for computing an elastic orientation-preserving matching of non-rigid 3D shapes. We outline an Integer Linear Programming formulation whose relaxed version can be minimized globally in polynomial time. Because of the high number of optimization variables, the key algorithmic challenge lies in efficiently solving the linear program. We present a performance analysis of several Linear Programming algorithms on our problem. Furthermore, we introduce a multiresolution strategy which allows the matching of higher resolution models.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—

1. Introduction

The registration of three-dimensional shapes is a fundamental problem in geometry processing. The abundance of such shapes – obtained from Laser scanners, image-based reconstruction methods or RGB-D cameras – gives rise to an increasing demand for efficient and unsupervised methods to compute optimal matchings – in order to relate shapes and their parts, in order to fuse different partial scans of a single 3D object, in order to transfer semantics from one shape to another, in order to quantify similarity of shapes and in order to automatically interpolate two different shapes.

In the last two decades, 3D shape matching has received considerable attention. The most classical problem studied in this context is the alignment of rigid shapes where one seeks for the Euclidean motion which moves one shape as close as possible towards the other one in the sense of the Hausdorff distance. Typically, this problem is handled by variants of the Iterative Closest Point (ICP) algorithm [BM92, GMGP05].

In *non-rigid shape matching*, the goal is to detect meaningful correspondences between two different poses of a shape or even between two different objects (as in Figure 1). Usually, this is formulated as an optimization problem over the manifold of surface mappings or diffeomorphisms subject to appropriate constraints. Because in general, such a

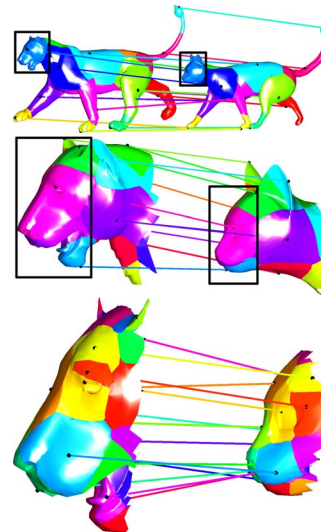


Figure 1: We compute orientation-preserving matchings of three-dimensional shapes by minimizing the elastic thin-shell energy required to deform one shape into the other one. The ILP formulation leads to solutions which are independent of the initialization. A multiresolution approach allows for matchings at reasonably high resolution, visualized here by Voronoi cells in successive zoom-ins.

manifold of surface mappings is infinite-dimensional, non-rigid shape matching is a much more challenging problem than rigid shape matching which deals with the six-dimensional group of Euclidean motions.

By assuming that the deformation preserves geodesic distances, which is approximately true for pose changes, the search space for non-rigid matchings can be made much smaller [OMMG10]. Without this restriction, Mé-moli and Sapiro [MS05] proposed the use of the Gromov–Hausdorff framework. The paradigm is to search for the correspondence with the least distortion of geodesic distances. An efficient algorithm for computing such a minimum-distortion correspondence was proposed by Bronstein et al. [BBK06]. More recently, the geodesic distance in the Gromov–Hausdorff framework was replaced by the diffusion distance [BBK*10].

While in the minimum-distortion approach only stretching and strain are penalized, other more physics-oriented approaches also take into account bending – thereby allowing for example to energetically distinguish a straight finger from a bent finger. The resulting deformation energy is a thin-shell energy [GHDS03]. Such an approach has been applied to the matching of disk-type surfaces by Litke et al. [LDRS05]. Wirth et al. [WBR09] use a similar idea, applying continuum mechanics to volumetric shapes.

Due to its mathematical beauty and simplicity, conformal geometry has been studied excessively in geometry processing. The works [WGH*05, ZZW*08, LWT*10] propose shape matching approaches which exploit this theory.

Inspired by the work of Lafon [Laf04] on data analysis based on diffusion distances, recently the eigenfunctions of the Laplace–Beltrami operator have gained a lot of attention in shape analysis [Lév06]. Shape matching based on these techniques has been proposed by Mateus et al. [MHK*08] and by Jain et al. [JzVK07].

All methods for non-rigid shape registration mentioned so far have in common that they use local optimization methods for a non-convex optimization problem. As a result, the detected correspondences depend heavily on the initialization (possibly improved with a coarse-to-fine strategy) and they could in principle be arbitrarily bad.

Recently, two methods have been proposed which incorporate methods from global optimization. Zeng et al. [ZWW*10] propose a graph matching formulation with third order energy terms. The resulting energy is approximately minimized using the dual decomposition relaxation scheme proposed by Torresani et al. [TKR08]. This approach has the drawback that all triples of possible assignments are taken into account. This results in a very high complexity and allows only the matching of a few feature points which is post-processed with a local method.

On the other hand, in their remarkable paper [LD09], Lipman and Daubechies introduce a surface distance for disk-

type surfaces which builds on the framework of Optimal Mass Transport. The transportation cost is constructed using conformal geometry. The authors are able to prove that their similarity measure satisfies all axioms of a distance function. The computation of this distance comes down to solving a linear assignment problem which can be done globally optimally in polynomial time. However, no spatial regularization is included in this framework so that the resulting assignments cannot be used as reliable shape matchings.

In this paper, we study a framework for non-rigid three-dimensional shape matching which was recently introduced in [WSSC11]. Rather than matching points to points as in all previous works, this approach brings small surface patches in correspondence. This allows to compute matchings which are geometrically consistent in the sense that a discrete continuity property is satisfied and orientations are preserved. This property appears indispensable for a meaningful comparison of the boundary surfaces of three-dimensional objects.

Computationally, our approach is formulated as an Integer Linear Program (ILP) whose relaxed version can be solved globally optimally in polynomial time. Since the number of variables grows quadratically with the number of vertices of the shapes, algorithmic efficiency is an important issue. For this reason, we investigate the performance of several Linear Programming (LP) algorithms on our problem. Moreover, we propose a multiscale framework in order to achieve shape matchings in higher resolutions.

As for the energy model, we follow the physical approach as described above, using an elastic, non-linear thin-shell energy. However, the proposed framework allows for any energy of first order consisting of a data term of order zero and a first-order regularizer involving the Jacobian of the surface map. We show that the performance of our algorithm can be boosted with the incorporation of a feature descriptor.

Compared to [WSSC11], our main contributions are summarized as follows:

- We discuss theoretical aspects of the shape matching framework and we prove that it computes discretely continuous, orientation preserving correspondences.
- We evaluate the performance of several LP algorithms for efficiently determining the globally optimal solution of the relaxed ILP.
- We introduce a multiresolution framework which allows to efficiently process shapes in higher resolution.
- We show that the incorporation of a feature descriptor further enhances the performance of our algorithm.

2. Integer Linear Program Formulation

In this section we present our ILP formulation for geometrically consistent shape matching. The key idea is to represent

a surface map by its graph surface. This idea appeared previously in the context of 2D shape matching in the work of Schmidt et al. [SFC07].

In Section 2.1 we will quickly review the continuous theory of diffeomorphisms and their graph surfaces and we will give the continuous version of the thin-shell energy used in this work. The discrete version of graph surfaces will be derived in the sequel. In Section 2.2 we will introduce product triangles which are the building blocks of discrete graph surfaces. Product triangles are glued to discrete graph surfaces in Section 2.3. Finally, we elaborate on the ILP formulation for shape matching in Section 2.4.

2.1. Diffeomorphisms and their Graphs

Let $X, Y \subset \mathbb{R}^3$ be two closed, oriented differentiable surfaces. We formulate shape matching between X and Y as an optimization problem in the group of orientation preserving diffeomorphisms $\text{Diff}^+(X, Y)$. Recall that a diffeomorphism between X and Y is an invertible, differentiable map $\varphi : X \rightarrow Y$ whose inverse $\varphi^{-1} : Y \rightarrow X$ is again differentiable. We argue that orientation preserving diffeomorphisms are the correct search space for registrations of boundary surfaces of volumetric three-dimensional shapes. Indeed, a deformation of one volumetric three-dimensional shape into another one induces an orientation-preserving diffeomorphism of their boundary surfaces. Vice versa, each such diffeomorphism can be obtained as the boundary map induced by a volumetric shape deformation.

The energy we impose on diffeomorphisms is a variant of Koiter's thin-shell energy [Koi66]. It consists of a membrane term penalizing stretching and compression and of a bending term. For a diffeomorphism $\varphi : X \rightarrow Y$ we set

$$E(\varphi) = \underbrace{\int_X (\text{tr}_{g_X} \mathbf{E}) + \mu \text{tr}_{g_X} (\mathbf{E}^2)}_{E_{\text{mem}}} + \lambda \underbrace{\int_X (H_X(x) - H_Y(\varphi(x)))^2}_{E_{\text{bend}}} \quad (1)$$

where H_X and H_Y denote the mean curvature of the two shapes X and Y resp., λ and μ are parameters that model the material's bending and stretching properties and the Lagrange strain tensor \mathbf{E} measures the difference between the local metrics on both shapes. This tensor is defined as $\mathbf{E} = \varphi^* g_Y - g_X$ where g_X is the Riemannian metric on X and $\varphi^* g_Y$ is the pullback of the metric g_Y on Y (cf. [DC92]).

Since in shape matching there is no reason to prefer one shape to the other one, we use a symmetric problem formulation. The resulting optimization problem we aim to solve is

$$\varphi^* = \text{argmin}_{\varphi \in \text{Diff}^+(X, Y)} E(\varphi) + E(\varphi^{-1}). \quad (2)$$

In our computational approach, rather than searching for

the diffeomorphism φ itself, we search for its graph surface $\Gamma_\varphi = \{(x, \varphi(x)) \in X \times Y \mid x \in X\} \subset X \times Y$. This graph surface has the following properties

- (i) Γ_φ is a differentiable, connected, closed surface in the product space $X \times Y$.
- (ii) The natural projections $\pi_X : \Gamma_\varphi \rightarrow X$ and $\pi_Y : \Gamma_\varphi \rightarrow Y$ are both diffeomorphisms.
- (iii) The two orientations which Γ_φ naturally inherits from X and Y coincide.

Vice versa, any surface $\Gamma \subset X \times Y$ satisfying conditions (i),(ii) and (iii) is the graph surface of a diffeomorphism between X and Y . In the following we work out how to define discrete versions of (i),(ii) and (iii).

2.2. Product Triangles

In this subsection we will introduce the set of product triangles which are the building blocks of discrete graph surfaces. Each such product triangle can be interpreted as setting into correspondence a small surface patch on one shape with a small surface patch on the other shape. Before defining the product triangles, we have to discuss degenerate edges and triangles of a mesh. These will enable our matching framework to cope with stretching and compression.

Let X be an oriented, connected triangle mesh. Denote by V_X the set of vertices of X . We fix orientations for edges and for triangles. Since edges do not have a natural orientation, we arbitrarily choose one. The set of oriented edges is denoted by E_X . Thus, if two vertices $v_1, v_2 \in V_X$ are connected by an edge, then either $\binom{v_1}{v_2} \in E_X$ or $\binom{v_2}{v_1} = -\binom{v_1}{v_2} \in E_X$. By assumption, triangles do have a natural orientation. Let F_X be the set of oriented triangles. If vertices a_1, a_2, a_3 build an oriented triangle, then $\binom{a_1}{a_2}{a_3} = \binom{a_2}{a_3}{a_1} = \binom{a_3}{a_1}{a_2} \in F_X$.

We extend the set of edges E_X by *degenerate edges*

$$\bar{E}_X = E_X \cup \left\{ \binom{a}{a} \mid a \in V_X \right\}. \quad (3)$$

Similarly, the set of triangles F_X is extended by *degenerate triangles*

$$\bar{F}_X = F_X \cup \left\{ \binom{a_1}{a_2}{a_2} \mid a_1, a_2 \in V_X, \pm \binom{a_1}{a_2} \in \bar{E}_X \right\}. \quad (4)$$

Note that degenerate triangles can consist of two vertices or of only one vertex.

Let now $Y = (V_Y, E_Y, F_Y)$ be a second oriented, connected triangle mesh and define \bar{E}_Y, \bar{F}_Y analogously. Then the set of product edges is defined as

$$E = \bar{E}_X \times \bar{E}_Y. \quad (5)$$

The set of oriented product triangles is defined as

$$F = \left\{ \binom{a_1 \ b_1}{a_2 \ b_2}{a_3 \ b_3} \mid \begin{array}{l} f_a = \binom{a_1}{a_2}{a_3} \in \bar{F}_X, \\ f_b = \binom{b_1}{b_2}{b_3} \in \bar{F}_Y, \\ f_a \text{ or } f_b \text{ non-degenerate} \end{array} \right\} / \sim. \quad (6)$$

where \sim is the equivalence relation generated by $\begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{pmatrix} \sim \begin{pmatrix} a_2 & b_2 \\ a_3 & b_3 \\ a_1 & b_1 \end{pmatrix} \sim \begin{pmatrix} a_3 & b_3 \\ a_1 & b_1 \\ a_2 & b_2 \end{pmatrix}$. For shape matching, an activated product triangle $\begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{pmatrix} \in F$ is interpreted as setting into correspondence triangle f_a on X with triangle f_b on Y in such a way that vertex $a_i \in V_X$ is set in correspondence with vertex $b_i \in V_Y$.

2.3. Discrete Graph Surfaces

In the previous subsection we introduced product triangles as the building blocks of discrete graph surfaces. In this section we will see how to glue product triangles to discrete graph surfaces. First we define the set of discrete product surfaces which will contain the set of discrete graph surfaces.

Definition 1 A **discrete product surface** Γ is a subset $\Gamma \subset F$.

We now formulate discrete versions of conditions (i), (ii) and (iii).

Discrete Version of (i). Here we present the discrete version of the closedness constraint. For shape matching, this constraint has the interpretation that matchings of triangular patches are continued consistently across edges. To formulate this we need to introduce the boundary operator mapping product triangles in F to product edges in E . This operator is reminiscent of the boundary operator for triangle meshes [DHLM05].

As done in Section 2.2 for edges on X and Y , we fix arbitrary orientations of the product edges E . Given a product edge $e = \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix}$, we denote by $v(e) \in \{-1, 0, 1\}^{|E|}$ the vector with entry 1 in the e -th component if e is positively oriented and -1 if it is negatively oriented and with all other entries equal to 0.

Definition 2 We define the **boundary operator** mapping product triangles to their boundaries

$$\partial : F \rightarrow \{-1, 0, 1\}^{|E|} \quad (7)$$

$$\begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{pmatrix} \mapsto v \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \end{pmatrix} + v \begin{pmatrix} a_2 & b_2 \\ a_3 & b_3 \end{pmatrix} + v \begin{pmatrix} a_3 & b_3 \\ a_1 & b_1 \end{pmatrix}.$$

The boundary operator is extended linearly to a map $\partial : [0, 1]^{|F|} \rightarrow \mathbb{Z}^{|E|}$.

A discrete product surface $\Gamma \subset F$ is called **closed** if it satisfies $\partial(\Gamma) = \mathbf{0}_{|E|}$, where $\mathbf{0}_{|E|}$ is the all-zeros vector of length $|E|$.

Discrete Version of (ii). Now we formulate the discrete analog of bijective projections. We require that each non-degenerate triangle on X or Y has a unique (possibly degenerate) correspondence partner on the other mesh. To formalize this, define for $f_a \in F_X$ the vector $v_X(f_a) \in \{0, 1\}^{|F_X|}$

as the indicator vector of f_a which has all entries equal to 0 but the f_a -th one which is 1. For $f_b \in F_Y$, the vector $v(f_b) \in \{0, 1\}^{|F_Y|}$ is defined analogously.

Definition 3 The **projection** $\pi_X : F \rightarrow \{0, 1\}^{|F_X|}$ is defined by

$$\pi_X(f_a, f_b) = \begin{cases} v(f_a) & \text{if } f_a \text{ is non-deg.} \\ \mathbf{0}_{|F_X|} & \text{else.} \end{cases} \quad (8)$$

The projection $\pi_Y : F \rightarrow \{0, 1\}^{|F_Y|}$ is defined analogously. A discrete product surface $\Gamma \subset F$ satisfies the **projection constraint** if $\pi_X(\Gamma) = \mathbf{1}_{|F_X|}$ and $\pi_Y(\Gamma) = \mathbf{1}_{|F_Y|}$ where $\mathbf{1}_n$ is the vector of length n with all entries equal to one.

Discrete Version of (iii). There is no need for imposing a further constraint in order to guarantee orientation preserving matchings because by definition F only contains patches which induce such correspondences.

For computations, discrete product surfaces are represented by their indicator vectors $\Gamma \in \{0, 1\}^{|F|}$. Then, the boundary operator can be expressed by a (sparse) matrix of size $|E| \times |F|$ and the projection operators by (sparse) matrices of size $|F_X| \times |F|$ resp. $|F_Y| \times |F|$.

Definition 4 A **discrete graph surface** is a closed discrete product surface $\Gamma \in \{0, 1\}^{|F|}$ which satisfies the projection constraint, that is

$$\begin{pmatrix} \partial \\ \pi_X \\ \pi_Y \end{pmatrix} \cdot \Gamma = \begin{pmatrix} \mathbf{0}_{|E|} \\ \mathbf{1}_{|F_X|} \\ \mathbf{1}_{|F_Y|} \end{pmatrix}. \quad (9)$$

2.4. Integer Linear Program

In Section 2.3 we have identified the correct search space for our discrete diffeomorphism optimization problem. Before stating the ILP formulation, we need to discretize the energy given in (1) (or rather its symmetrized version which appears in (2)).

Recall that each product triangle $f = \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{pmatrix}$ has the meaning to set $f_a = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$ in correspondence with $f_b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$ in such a way that a_i corresponds to b_i . The cost of such a correspondence is split in

$$E(f) = E_{\text{mem}}(f) + \lambda E_{\text{bend}}(f). \quad (10)$$

For the membrane term we adopt the discretization proposed by Delingette. We refer the reader to [Del08] for the somewhat lengthy formula of the membrane cost $E_{\text{Del}}(f_a, f_b)$ for deforming f_a in f_b . We use this discretization symmetrically

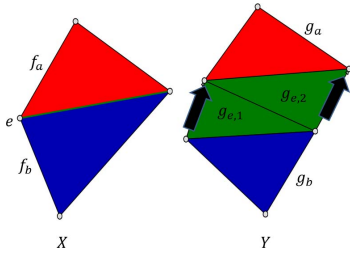


Figure 2: Discrete graph surfaces induce continuous matchings (cf. Theorem 1 (b)). The figure shows two adjacent triangles f_a (red) and f_b (blue) on X (on the left) which are set in correspondence with triangles g_a (red) and g_b (blue) on Y (on the right). The triangles $g_{e,1}$ and $g_{e,2}$ (green) on Y are contracted to the edge e between f_a and f_b as indicated by the black arrows on the right. Thus, edge e has to be stretched to the contractible patch marked in green.

and define

$$E_{\text{mem}}(f) = \begin{cases} E_{\text{Del}}(f_a, f_b) + E_{\text{Del}}(f_b, f_a) & \text{if } f_a, f_b \text{ non-deg.} \\ 2E_{\text{Del}}(f_a, f_b) & \text{if } f_b \text{ deg.} \\ 2E_{\text{Del}}(f_b, f_a) & \text{if } f_a \text{ deg.} \end{cases} \quad (11)$$

For the bending term, we use the vertex-wise squared difference of mean curvatures, weighted with the mixed surface areas. Mean curvatures are discretized using the scheme proposed by Meyer et al. [MDSB02]. The resulting bending energy is

$$E_{\text{bend}}(f) = \sum_{i=1}^3 A_i^X (H_X(i) - H_Y(i))^2 + \sum_{i=1}^3 A_i^Y (H_X(i) - H_Y(i))^2. \quad (12)$$

Here, A_i^X resp. A_i^Y refer to the area of the Voronoi cell of vertex i in triangle f_a resp. in triangle f_b .

Let now E be the vector of length $|F|$ whose f -th entry is the cost $E(f)$. We formulate the geometrically consistent shape matching problem as the Integer Linear Program (ILP)

$$\begin{aligned} & \min_{\Gamma \in \{0,1\}^{|F|}} E^t \cdot \Gamma \\ & \text{subject to} \quad \begin{pmatrix} \partial \\ \pi_X \\ \pi_Y \end{pmatrix} \cdot \Gamma = \begin{pmatrix} \mathbf{0}_{|E|} \\ \mathbf{1}_{|F_X|} \\ \mathbf{1}_{|F_Y|} \end{pmatrix}. \end{aligned} \quad (13)$$

3. Geometric Consistency

In this section we show that discrete graph surfaces induce matchings with built-in geometric consistency. Loosely speaking, Theorem 1 below says that discrete graph surfaces as defined in Definition 4 induce orientation preserving, continuous and bijective matchings. To formulate this result in a mathematically precise way we need some further notation.

A discrete product surface Γ induces correspondences between surface patches on X and on Y . To define these maps,

we introduce the projection

$$p_X : F \rightarrow \bar{F}_X, \quad \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{pmatrix} \mapsto \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \quad (14)$$

Define $p_Y : F \rightarrow \bar{F}_Y$ in a similar way. For a set A , denote by $\mathcal{P}(A)$ its power set consisting of all subsets of A .

Definition 5 We define the **correspondence map induced by Γ** as

$$\begin{aligned} \Gamma_X : \mathcal{P}(\bar{F}_X) &\rightarrow \mathcal{P}(\bar{F}_Y) \\ A &\mapsto p_Y(p_X^{-1}(A) \cap \Gamma) \end{aligned} \quad (15)$$

for all $A \subset \bar{F}_X$. The correspondence map $\Gamma_Y : \mathcal{P}(\bar{F}_Y) \rightarrow \mathcal{P}(\bar{F}_X)$ is defined analogously.

For an edge $e = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \in E_X$ on X we define the image patch of e as

$$\Gamma(e) := \Gamma_X \left(\left\{ \begin{pmatrix} a_1 \\ a_1 \\ a_2 \end{pmatrix}, \begin{pmatrix} a_1 \\ a_2 \\ a_2 \end{pmatrix} \right\} \right). \quad (16)$$

Theorem 1 (Geometric Consistency) Let $\Gamma \subset F$ be a discrete graph surface. Then

- (a) Γ induces *orientation preserving* correspondences of triangles.
- (b) Γ is *continuous* in the following sense: for adjacent triangles f_1, f_2 sharing an edge e , the image patch $\Gamma(e)$ is a contractible path on the dual mesh of Y which connects $\Gamma(f_1)$ and $\Gamma(f_2)$. This statement is visualized in Figure 2.
- (c) Statement (b) is true with the roles of X and Y being interchanged. Thus, Γ induces a *bijective* matching.

The proof of this theorem is elementary but somewhat technical. It can be found in Appendix A. The continuity statement in part (b) can be understood in the following sense: The triangle maps from f_i to $\Gamma(f_i)$ can be “continuously” extended over an edge e by thickening e along the contractible path $\Gamma(e)$.

Note that the theorem does *not* say that Γ induces a bijection between \bar{F}_X and \bar{F}_Y . In fact, this is not even desirable, since degenerate triangles should only appear at places where stretchings occur. However, the theorem assures that by means of a discrete graph surface Γ , mesh X induces a complete, geometrically consistent covering of mesh Y and vice versa. Thereby, discrete graph surfaces are the correct search space in order to find a discretized representation of the optimal shape diffeomorphism, i.e., the optimal shape matching.

4. Evaluation of LP Solution Strategies

In this section we explain how to get approximate solutions to (13) using Linear Programming (LP) relaxation and we evaluate the performance of several LP algorithms on our problem.

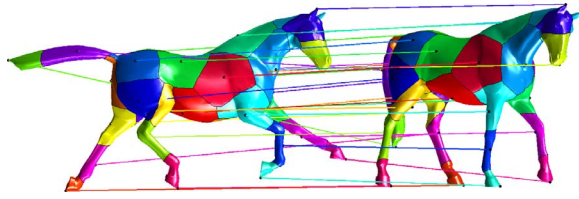


Figure 3: Correspondence of a running horse with a standing horse. Correspondences are visualized by colored patches and by connecting lines.

4.1. LP relaxation

The optimization problem (13) is a binary linear program and as such an NP-hard problem. In order to get approximate solutions we first use LP relaxation. This means that we relax the binary constraints $\Gamma \in \{0, 1\}^{|F|}$ to $\Gamma \in [0, 1]^{|F|}$. This relaxed problem can be minimized globally in polynomial time. Unfortunately, the constraint matrix in (13) is not totally unimodular. Therefore, in general solutions to the relaxed problem take values in the interior of $[0, 1]^{|F|}$.

In order to construct a binary solution out of a solution of the relaxed problem, we cannot apply a simple thresholding scheme since this would destroy the geometric consistency of the solution. Instead, we propose an iterative scheme: solve the relaxed version of (13), fix the variables with values above a threshold > 0.5 to 1 and solve the relaxed version of (13) 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.

One of the benefits of LP relaxation is that the minimum of the relaxed problem is a lower bound on the integer problem. This allows to specify the worst case energetic distance between the approximate solution obtained by the iterative scheme outlined above and the actual optimum of the ILP. Experimentally we observed that this gap lies usually between 0 and 3 percent.

4.2. Evaluation of LP algorithms

We evaluated three different LP algorithms for solving the relaxed version of (13):

- the simplex algorithm in the state-of-the-art implementation provided by the CPLEX package,
- the interior point algorithm in the CPLEX implementation,
- the parallelizable primal-dual algorithm proposed by Eckstein and Bertsekas [EB90] in our own GPU-based implementation.

The simplex algorithm could only handle very small-sized toy problems of about 10 triangles per mesh. On problems of

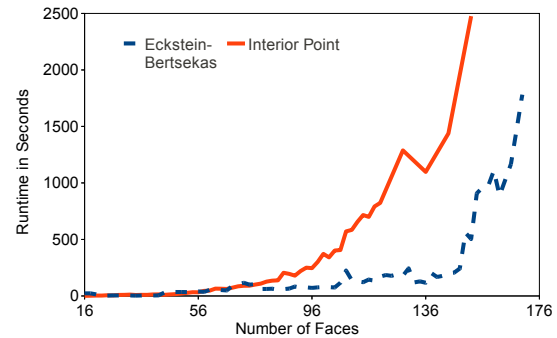


Figure 4: Comparison of the runtimes (in seconds) for solving the relaxed LP with the interior point method and with the algorithm of Eckstein–Bertsekas depending on the problem size (number of triangles per shape). Visibly, for larger problems the advantages of the GPU implementation of Eckstein–Bertsekas become important.

this size the other two solvers were faster by a factor of about 10. For bigger problems the solver crashed.

A comparison of the performance of the *interior-point method* and of the algorithm of *Eckstein–Bertsekas* is shown in Figure 4. Visibly, for larger problem size the algorithm of Eckstein–Bertsekas benefits from its parallel implementation and is much faster than the interior-point method.

Also we observed that for very large problems the interior-point method in the CPLEX implementation run out of memory. This is in contrast with the algorithm of Eckstein–Bertsekas which only needs to save the primal and the dual variables and therefore has linear memory consumption.

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-resolution 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 triangle mesh X_{i+1} by repeatedly merging triangles. In practice we use the quadric

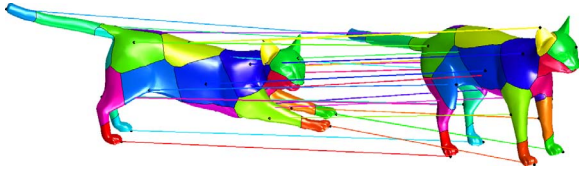


Figure 5: The matching between a stretching cat on the left and a standing cat on the right.

edge decimation algorithm [GH97] in its OpenMesh implementation [BSBK02]. 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 $\chi_i : \bar{F}_{X_{i+1}} \rightarrow \bar{F}_{X_i}$ (see Section 2.2).

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

$$F_{i+1}^{\text{red}} = \left\{ (f_a, f_b) \in F_{i+1} \left| \begin{array}{l} \exists (f'_a, f'_b) \in \Gamma_i \subset F_i \text{ s.t.} \\ \chi_i(f_a) \in \mathcal{N}(f'_a) \text{ and} \\ \psi_i(f_b) \in \mathcal{N}(f'_b) \end{array} \right. \right\}. \quad (17)$$

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 (13) 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

To experimentally validate the proposed algorithm we did an extensive evaluation on shapes from several datasets. For the shown models we computed dense correspondences at a resolution of about 1000 vertices. Since dense high-resolution matchings do not allow a direct visualization we chose a sparse set of vertices on one shape by means of farthest point

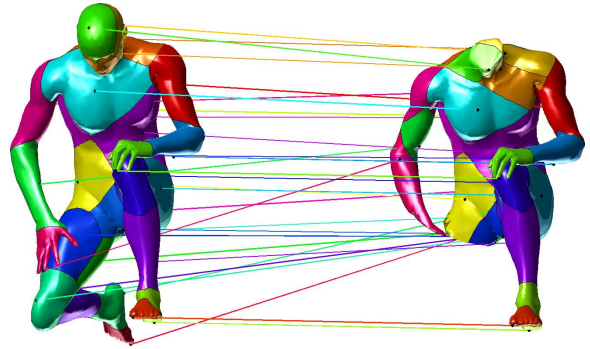


Figure 6: The proposed method is robust to cases where several parts of a model are missing. As expected from the elasticity property of the energy function, the missing parts are shrunked correctly from the entire model to incomplete model.

sampling and we rendered its Voronoi cell and the Voronoi cell of its correspondence partner on the other mesh with the same color.

6.1. 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.

6.2. Influence of Feature Descriptors

One of the benefits of the framework described in Section 2 is its flexibility regarding the diffeomorphism energy. Indeed, any first order energy with a data term of order zero and a first-order regularizer involving the diffeomorphism's Jacobian can be handled.

As an example we extended the thin-shell energy by a data term comparing the Wave Kernel Signatures (WKS) [ASC11] of matched vertices. This data term E_{WKS} was computed in the same way as the bending term in (12). The resulting matching energy of a discrete graph surface Γ was then

$$E(\Gamma) = E_{\text{mem}}(\Gamma) + \lambda E_{\text{bend}}(\Gamma) + \mu E_{\text{WKS}}(\Gamma). \quad (18)$$

While the results of the matchings visually do not change a lot when setting μ to 1 (see Figure 7), interestingly the computation time decreases dramatically. The reason for the decreasing runtimes is that with the use of a feature descriptor the solution to the LP relaxation explained in Section 4.1 often happens to be binary. This was the case in 56% of our experiments with the signature term compared to 13% without the signature term. Overall the average number of fixing steps decreased from 8 to 2.3 by using the signature term.

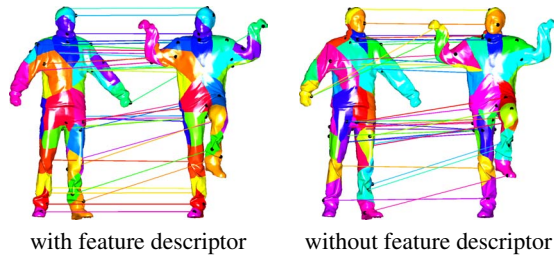


Figure 7: Influence of a feature descriptor on the computed matching. On the left we visualize the matching computed with the energy extended by a feature descriptor. The correspondences on the right are obtained by minimizing only the thin-shell energy.

6.3. Quantitative Evaluation

For quantitatively evaluating the proposed method we used pairs of models from the dataset of Vlastic et al. [VBMP08]. Namely, we took 10 pairs of models from each of the sets "crane", "samba" and "bouncing" at a resolution of 500 vertices. We computed the matchings between each of the 30 pairs by using the proposed algorithm and by using the GMDS method by Bronstein et al. [BBK06]. One of the models used in the evaluation is shown in Figure 8.

Since the exact ground truth correspondences (x_i, y_i) are available for these models X, Y , we measured the quality of a matching $\varphi: X \rightarrow Y$ by the mean geodesic error $\frac{1}{N} \sum_i d(\varphi(x_i), y_i)$ where N is the number of vertices being matched and d the normalized geodesic distance on the manifold of mesh Y .

Because GMDS cannot guarantee to deliver an orientation-preserving matching, out of the 30 pairs of models it produced 12 matchings that inverted the orientation as a whole. As there is no ground truth available for this case, we ignored these matchings in the evaluation. On the other hand, on 8 of the 30 models, the GMDS algorithm computed matchings with partially intertwined orientations. Figure 8 shows such an example where the left hand is matched to the left hand but right foot is matched to left foot. In contrast to this, our method preserved the orientations in all examples.

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.

7. Conclusion

We present a framework for efficiently computing high-resolution orientation-preserving matchings among three-dimensional shapes. Because surface patches rather than points are set into correspondence we were able to prove that registrations computed by this method are geometrically consistent in the sense that they are continuous and orientation-preserving. We derived an ILP formulation

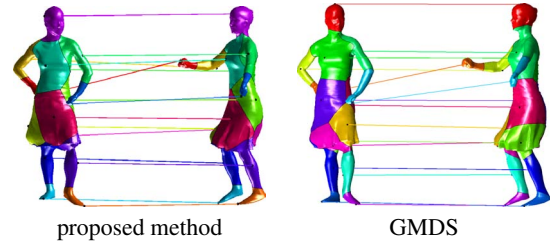


Figure 8: Comparison of a matching computed with the proposed method (on the left) and with GMDS proposed by Bronstein et al. [BBK06] (on the right). The proposed matching method guarantees to preserve the orientation. In contrast, GMDS may preserve the orientation on the upper part of the body while it inverts the orientation of the legs – in our evaluation this happened 8 times out of 30. (Visualization with MATLAB code by Bronstein et al.)

whose relaxed version can be minimized globally in polynomial time. We discussed and quantitatively compared appropriate algorithms for efficiently solving this ILP. In particular, we introduced a multiresolution framework which allows to handle meshes of higher resolutions. Numerous experiments confirm that the proposed method computes reliable dense matchings of real-world three-dimensional shapes.

Acknowledgment

We used mesh data from Vlastic et al. [VBMP08] (Figures 7 and 8), from the SHREC 2011 benchmark [BBB*11] (Figure 6) and from Sumner and Popović [SP04] (Figures 1, 3 and 5). For the visualization we used Matlab code by Bronstein et al. [BBK06].

References

- [ASC11] AUBRY M., SCHLICKWEI U., CREMERS D.: Pose-Consistent 3D Shape Segmentation Based on a Quantum Mechanical Feature Descriptor. In *Pattern Recognition (Proc. DAGM)* (to appear 2011). 7
- [BBB*11] 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. *ArXiv e-prints* (February 2011). 8
- [BBK06] BRONSTEIN A., BRONSTEIN M., KIMMEL R.: Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Sci. Comput.* 28, 5 (2006), 1812–1836 (electronic). 2, 8
- [BBK*10] 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. 2
- [BM92] BESL P., MCKAY N.: A method for registration of 3-d shapes. *IEEE PAMI* 14, 2 (1992), 239–256. 1

- [BSBK02] BOTSCH M., STEINBERG S., BISCHOFF S., KOBELT L.: Openmesh - a generic and efficient polygon mesh data structure. In *OpenSG Symposium 2002* (2002). 7
- [DC92] DO CARMO M.: *Riemannian geometry*. Birkhauser, 1992. 3
- [Del08] DELINGETTE H.: Triangular springs for modeling non-linear membranes. *IEEE Transactions on Visualisation and Computer Graphics* 14, 2 (March/April 2008). 4
- [DHLM05] DESBRUN M., HIRANI A. N., LEOK M., MARSDEN J. E.: Discrete exterior calculus, 2005. 4
- [EB90] ECKSTEIN J., BERTSEKAS D.: *An Alternating Direction Method for Linear Programming*. Technical report, Harvard Business School, 1990. 6
- [GH97] GARLAND M., HECKBERT P. S.: Surface simplification using quadric error metrics. In *SIGGRAPH* (1997), pp. 209–216. 7
- [GHDS03] GRINSPUN E., HIRANI A., DESBRUN M., SCHRÖDER P.: Discrete Shells. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (Aug 2003), pp. 62–67. 2
- [GMGP05] GELFAND N., MITRA N., GUIBAS L., POTTMANN H.: Robust global registration. In *Symposium on Geometry Processing* (2005), pp. 197–206. 1
- [JZvK07] JAIN V., ZHANG H., VAN KAICK O.: Non-rigid spectral correspondence of triangle meshes. *International Journal of Shape Modeling* 13, 1 (2007), 101–124. 2
- [Koi66] 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. 3
- [Laf04] LAFON S.: *Diffusion maps and geometric harmonics*. PhD thesis, Yale University, 2004. 2
- [LD09] LIPMAN Y., DAUBECHIES I.: Surface comparison with mass transportation, 2009. 2
- [LDRS05] LITKE N., DROSKE M., RUMPF M., SCHRÖDER P.: An image processing approach to surface matching. In *Symposium on Geometry Processing* (2005), pp. 207–216. 2
- [Lév06] LÉVY B.: Laplace-Beltrami Eigenfunctions Towards an Algorithm That Understands Geometry. In *Proc. Int. Conf. on Shape Modeling and Applications* (2006), IEEE, p. 13. 2
- [LWT*10] 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), pp. 323–330. 2
- [MDSB02] MEYER M., DESBRUN M., SCHRÖDER P., BARR A.: Discrete differential-geometry operators for triangulated 2-manifolds, 2002. 5
- [MHK*08] MATEUS D., HORAUD R., KNOSSOW D., CUZZOLIN F., BOYER E.: Articulated shape matching using laplacian eigenfunctions and unsupervised point registration. In *CVPR* (2008). 2
- [MS05] 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. 2
- [OMMG10] OVSJANIKOV M., MÉRIGOT Q., MÉMOLI F., GUIBAS L. J.: One point isometric matching with the heat kernel. *Comput. Graph. Forum* 29, 5 (2010), 1555–1564. 2
- [SFC07] 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). 3
- [SP04] SUMNER R., POPOVIĆ J.: Deformation transfer for triangle meshes. *ACM Transactions on Graphics (TOG)* 23, 3 (2004), 399–405. 8
- [TKR08] TORRESANI L., KOLMOGOROV V., ROTHER C.: Feature correspondence via graph matching: Models and global optimization. In *ECCV* (2) (2008), pp. 596–609. 2
- [VBMP08] VLASIC D., BARAN I., MATUSIK W., POPOVIĆ J.: Articulated mesh animation from multi-view silhouettes. In *ACM SIGGRAPH 2008 papers* (2008), ACM, pp. 1–9. 8
- [WBR09] WIRTH B., BAR L., RUMPF M., SAPIRO G.: Geodesics in shape space via variational time discretization. In *EMMCVPR'09* (2009), vol. 5681 of *LNCS*, pp. 288–302.
- [WGH*05] WANG Y., GU X., HAYASHI K., CHAN T., THOMPSON P., YAU S.: Brain surface parameterization using riemann surface structure. In *MICCAI* (2) (2005), pp. 657–665. 2
- [WSSC11] WINDHEUSER T., SCHLICKWEI U., SCHMIDT F. R., CREMERS D.: Geometrically consistent elastic matching of 3d shapes: A linear programming solution. In *Proc. International Conference on Computer Vision* (to appear 2011). 2
- [ZWW*10] 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), pp. 382–389. 2
- [ZZW*08] ZENG W., ZENG Y., WANG Y., YIN X., GU X., SAMARAS D.: 3d non-rigid surface matching and registration based on holomorphic differentials. In *ECCV* (3) (2008), pp. 1–14. 2

Appendix A: Proof of Theorem 1

The plan of this Appendix is as follows: First we analyze the map $\Gamma \rightarrow X$. In Proposition 1 we will prove that it is simply a sequence of edge contractions. This will be one of the key points in the proof of Theorem 1 which we will develop afterwards.

In order to formulate the result on the map $\Gamma \rightarrow X$ we need the notion of edge contractions in simplicial complexes. An *abstract simplicial complex* \mathcal{K} consists of a set K of subsets of a vertex set V such that for each $a \in V$ the set $\{a\}$ is contained in K and such that for all $\sigma \in K$ and for all subsets $\tau \subset \sigma$ we have $\tau \in K$. The sets in K are called faces, the dimension of a face σ is $\text{card}(\sigma) - 1$. Two-dimensional faces are called triangles, one-dimensional faces are edges and zero-dimensional faces are vertices. The dimension of a simplicial complex is the maximum of the dimensions of its faces.

A simplicial map between two simplicial complexes $\mathcal{K} = (K, V)$ and $\mathcal{K}' = (K', V')$ is a map $p: V \rightarrow V'$ which satisfies $p(\sigma) \in K'$ for all $\sigma \in K$. A simplicial map is surjective if so is p and it is an isomorphism if p is bijective. By definition, a simplicial map induces a map on the set of faces which is again denoted by $p: K \rightarrow K'$. Explicitly, this map is defined by $p(\{v_1, \dots, v_n\}) = \{p(v_1), \dots, p(v_n)\}$.

Let now $e = \binom{a_1}{a_2} \in K$ be an edge of \mathcal{K} . The *contraction of \mathcal{K} along e* is a simplicial complex \mathcal{K}' together with a surjective simplicial map p between \mathcal{K} and \mathcal{K}' which we now define. The vertex set V' of \mathcal{K}' is defined by replacing a_1 and

a_2 in V by a new vertex c , i.e. $V' = (V \setminus \{a_1, a_2\}) \cup \{c\}$. We introduce the map

$$p : V \rightarrow V', \quad a \mapsto \begin{cases} a & \text{if } a \in V \setminus \{a_1, a_2\} \\ c & \text{else.} \end{cases} \quad (19)$$

Then the set of faces of K' is defined as $K' = \{p(\sigma) \mid \sigma \in K\}$. Note that during this operation faces can be deleted because K' is a set and therefore each face appears only once. For example, when contracting an edge on a triangle mesh, typically one deletes two faces, three edges and one vertex.

In our context, X naturally carries the structure of a simplicial complex.

The discrete product surface $\Gamma \subset X \times Y$ induces a simplicial complex $\mathcal{K}_\Gamma = (K_\Gamma, V_\Gamma)$ in the following way: the set of vertices V_Γ of \mathcal{K}_Γ consists of all product vertices in $V_X \times V_Y$ which are adjacent to a product triangle in Γ . The set of faces K_Γ is defined as the set containing all subsets of sets of the form $\{(a_1, b_1), (a_2, b_2), (a_3, b_3)\}$ where $\begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{pmatrix} \in \Gamma$.

Proposition 1 The natural projection $V_X \times V_Y \supset V_\Gamma \rightarrow V_X$, $(a_1, b_1) \mapsto a_1$ induces a simplicial map p_X^{simp} between \mathcal{K}_Γ and X which is a sequence of edge contractions followed by a simplicial isomorphism.

Idea of Proof (Proposition). Starting with the simplicial surjection $\mathcal{K}_\Gamma \rightarrow X$ we construct a series of simplicial surjections $\mathcal{K}_{\Gamma_i} \rightarrow X$ by iteratively contracting all edges of \mathcal{K}_{Γ_i} whose projections on X are single vertices until we reach a simplicial isomorphism $\mathcal{K}_{\Gamma_n} \rightarrow X$. \square

Before proving the theorem we will formulate a technical lemma which will be used in the proof of the theorem.

Lemma 1 Let $e = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$ be an edge incident to two oriented triangles $f_1 = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}, f_2 = \begin{pmatrix} a_1 \\ a_2 \\ a_4 \end{pmatrix} \in F_X$ on X .

- (i) Define $p_X^{-1}(e) := p_X^{-1}\left(\left\{\begin{pmatrix} a_1 \\ a_1 \\ a_2 \end{pmatrix}, \begin{pmatrix} a_1 \\ a_2 \\ a_2 \end{pmatrix}\right\}\right)$ where p_X was introduced in (14). Then any two triangles in $p_X^{-1}(e) \cap \Gamma$ may be connected by a path of product triangles g_1, \dots, g_n with $g_i \in p_X^{-1}(e) \cap \Gamma$ such that for each i the triangles g_i and g_{i+1} share a product edge which lies over e .
- (ii) The set of all triangles in $(p_X^{-1}(\{f_1, f_2\}) \cup p_X^{-1}(e)) \cap \Gamma$ is connected.

The proof of the lemma is elementary but tedious. It relies on studying the preimages of edges and faces under edge contractions in simplicial complexes and on inductive application of Proposition 1.

Proof of Theorem 1. The orientation preservation property (a) is immediate since by construction F only contains orientation preserving triangle correspondences.

To prove (b), let $f_1 = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$ and $f_2 = \begin{pmatrix} a_1 \\ a_2 \\ a_4 \end{pmatrix}$ be two triangles

on X which share the edge $e = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$. Let $\bar{f}_1 = \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \end{pmatrix}$ and $\bar{f}_2 = \begin{pmatrix} a_2 & b_4 \\ a_1 & b_5 \\ a_4 & b_6 \end{pmatrix}$ be the uniquely defined product triangles in Γ lying over f_1 and f_2 .

If $b_1 = b_2$, corresponding to the case that the edge e is contracted by Γ , then the neighbor of \bar{f}_1 over the edge $\begin{pmatrix} a_1 & b_1 \\ a_2 & b_1 \end{pmatrix}$ is necessarily \bar{f}_2 because this neighbor projects to a non-degenerate triangle on X which has the edge $\begin{pmatrix} a_2 \\ a_1 \end{pmatrix}$. On the other hand, no triangle in $p_X^{-1}(e) \cap \Gamma = p_X^{-1}\left(\left\{\begin{pmatrix} a_1 \\ a_1 \\ a_2 \end{pmatrix}, \begin{pmatrix} a_1 \\ a_2 \\ a_2 \end{pmatrix}\right\}\right) \cap \Gamma$ has an edge in common with \bar{f}_1 or with \bar{f}_2 because such a triangle would be degenerate in both projections. Using Lemma 1 (ii), this implies that $\Gamma(e)$ is empty and that $\Gamma(f_1)$ and $\Gamma(f_2)$ are adjacent on Y .

If on the other hand $b_1 \neq b_2$, then $b_4 \neq b_5$. If a product triangle in $p_X^{-1}(e) \cap \Gamma$ is adjacent to \bar{f}_1 , it is necessarily of the form $\begin{pmatrix} a_2 & b_2 \\ a_1 & b_1 \\ a_1 & b_7 \end{pmatrix}$ or $\begin{pmatrix} a_2 & b_2 \\ a_1 & b_1 \\ a_2 & b_7 \end{pmatrix}$ where b_7 is the uniquely defined third vertex of the triangle $\begin{pmatrix} b_2 \\ b_1 \\ b_7 \end{pmatrix}$ on Y . Similarly, if a product triangle in $p_X^{-1}(e) \cap \Gamma$ is adjacent to \bar{f}_2 , it is necessarily of the form $\begin{pmatrix} a_1 & b_5 \\ a_2 & b_4 \\ a_1 & b_8 \end{pmatrix}$ or $\begin{pmatrix} a_1 & b_5 \\ a_2 & b_4 \\ a_2 & b_8 \end{pmatrix}$ for some b_8 . Thus, if $\Gamma(e)$ is non-empty, using Lemma 1 (ii), one of these four product triangles has to be activated. Since there is nothing to prove for $\Gamma(e)$ empty, we assume w.l.o.g. that $g_1 = \begin{pmatrix} a_2 & b_2 \\ a_1 & b_1 \\ a_1 & b_7 \end{pmatrix}$ is contained in Γ . Using that Γ is closed, there exists a product triangle $g_2 = \begin{pmatrix} a_2 & b_2 \\ a_1 & b_7 \\ a_5 & b_9 \end{pmatrix} \in \Gamma$. If $a_5 = a_4$, by the projection constraint we have $g_2 = \bar{f}_2$. Otherwise we have $a_5 = a_1$ or $a_5 = a_2$ and $g_2 \in p_X^{-1}(e) \cap \Gamma$. In this case g_2 has a product edge which lies above $\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$ and with the same argument we find a product triangle g_3 . Continuing this process inductively we obtain a triangle path $g_0 = \bar{f}_1, g_1, g_2, \dots$ with $g_i \in p_X^{-1}(e) \cap \Gamma$ for $i > 0$. This path cannot have loops because the remaining third edge of each of the g_i is contracted to a_1 or to a_2 . Since there are only finitely many product triangles, the path stops after say n steps with $g_{n+1} = \bar{f}_2$.

Finally, no triangle other than g_1, \dots, g_n in $p_X^{-1}(e) \cap \Gamma$ may be connected to the g_i over an edge which lies above e . Therefore, using Lemma 1 (i), we get $p_X^{-1}(e) = \{g_1, \dots, g_n\}$.

The proof of statement (c) is analogous. \square