

# MINA: Convex Mixed-Integer Programming for Non-Rigid Shape Alignment

Florian Bernard

Zeeshan Khan Suri

Christian Theobalt

MPI Informatics

Saarland Informatics Campus

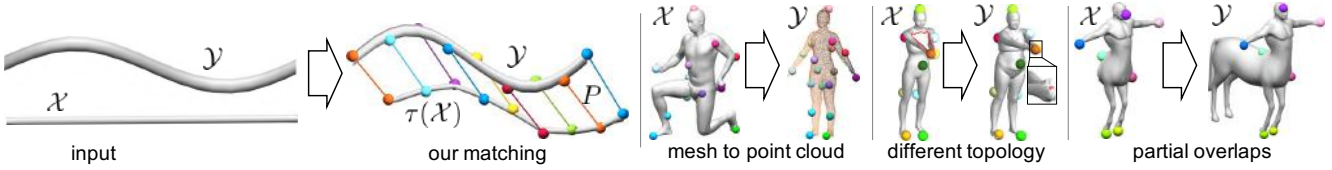


Figure 1. Left: MINA matches the straight cylinder  $\mathcal{X}$  non-rigidly to the curved cylinder  $\mathcal{Y}$  (their orientations are arbitrary, but we visualise them with same orientation). It finds the *globally optimal* correspondences  $P \in \mathbb{P}$  between a sparse set of points (coloured dots), as well as the non-rigid deformation  $\tau$  of  $\mathcal{X}$  so that  $\tau(\mathcal{X})$  aligns with  $\mathcal{Y}$ . MINA is highly flexible, e.g. it can match a mesh to a point cloud (middle left), match shapes with different topologies (middle right, where the hands in  $\mathcal{X}$  are *not* touching, while the hands in  $\mathcal{Y}$  are, see the geodesic paths between both hands shown as red lines), or deal with partial overlaps (right). (Best viewed on screen when zoomed in)

## Abstract

We present a convex mixed-integer programming formulation for non-rigid shape matching. To this end, we propose a novel shape deformation model based on an efficient low-dimensional discrete model, so that finding a globally optimal solution is tractable in (most) practical cases. Our approach combines several favourable properties: it is independent of the initialisation, it is much more efficient to solve to global optimality compared to analogous quadratic assignment problem formulations, and it is highly flexible in terms of the variants of matching problems it can handle. Experimentally we demonstrate that our approach outperforms existing methods for sparse shape matching, that it can be used for initialising dense shape matching methods, and we showcase its flexibility on several examples.

belongs to a particular class  $\Omega$  of transformations) such that the transformed shape  $\tau(\mathcal{X})$  best aligns with  $\mathcal{Y}$ . Formally, this can be written as the optimisation problem

$$\min_{\tau \in \Omega} d(\tau(\mathcal{X}), \mathcal{Y}), \quad (1)$$

where  $d(\cdot, \cdot)$  is a suitable metric that quantifies the discrepancy between both shapes. The particular shape matching setting depends on the choice of the metric  $d(\cdot, \cdot)$  and the class of transformations  $\Omega$ . For example, *rigid shape matching* refers to  $\Omega = \text{SE}(d)$ , where  $\text{SE}(d)$  is the special Euclidean group in dimension  $d$ . In this work we study the *non-rigid shape matching* problem, where  $\Omega$  comprises non-rigid deformations (to be defined in Sec. 3).

Although many previous works have addressed non-rigid shape matching, there are several open challenges: (i) due to the non-convex nature of Problem (1) for virtually all relevant choices of  $d(\cdot, \cdot)$  and  $\Omega$ , existing methods cannot guarantee to find global optima. Hence, these methods heavily depend on the initial choice of  $\tau$ . (ii) Oftentimes, non-rigid shape matching methods require that both shapes have the same representation (e.g. meshes). (iii) Existing approaches have a limited flexibility in terms of the matching formulation that can be handled, e.g. they can only handle bijective matchings, or they cannot guarantee injectivity, they do not allow for additional constraints (e.g. bounding the maximum distortion of a matching), or they cannot deal with shapes that have different topologies. (iv) Moreover, existing formulations that purely aim for preserving pairwise distances when finding a matching (see *quadratic assignment problem* in Sec. 2) are not guaranteed to maintain the

## 1. Introduction

Finding correspondences in geometric data is a long-standing problem in vision, graphics, and beyond. The applications range from the creation of statistical shape models, 3D reconstruction, object tracking, or recognition, to more recent settings such as the alignment of geometric data to enable the training of deep learning models. In this work we consider the problem of finding correspondences between two given shapes, known as *shape matching*.

We assume that one shape is a geometrically transformed version of the other shape. With that, matching shape  $\mathcal{X}$  to shape  $\mathcal{Y}$  can be phrased as finding a transformation  $\tau$  (which

orientation of the surface.

**Our contribution.** Our main idea is to formulate non-rigid shape matching in terms of a convex mixed-integer programming (MIP) problem, while addressing (i)-(iv). We summarise our main contributions as follows:

- We propose a low-dimensional discrete model for non-rigid shape matching that is highly flexible as it allows to tackle a wide range of matching formulations.
- Although solving MIP problems to global optimality has worst-case time complexity that is exponential in the number of integer variables, our proposed formulation only requires a small number of integer variables that is *independent* of the shape resolution.
- Our formulation does not require an initialisation and it is oftentimes possible to (certifiably) find a globally optimal solution in practice.

## 2. Related Work & Background

Due to the vast amount of literature related to shape matching and correspondence problems, it is beyond the scope of this paper to provide an exhaustive background of related work. A broad overview of the topic is for example presented in [47]. In the following we summarise works that we consider most relevant.

**Rigid shape matching.** Finding a rotation and translation that aligns two shapes is known as *rigid* shape matching. The Procrustes problem [42] considers the setting when the correspondences between points on both shapes are known, which admits an efficient closed-form solution. However, rigid shape matching becomes significantly harder if the correspondences are unknown. Most commonly, this is addressed via local optimisation. A popular approach is the Iterative Closest Point (ICP) algorithm [7], which also comes in various variants, such as a probabilistic formulation [30]. These methods have in common that they do not guarantee to find a globally optimal solution and therefore their outcome is highly dependent on a good initialisation. Contrary to these local methods, for the rigid shape matching problem there are also global approaches, e.g. based on a semidefinite programming relaxation [26], or on branch and bound algorithms [31, 52].

A downside of these shape matching approaches is that they have the strong assumption that both shapes can be aligned based on a *rigid-body* transformation. However, in practice this assumption is oftentimes violated, so that non-rigid shape matching approaches are more appropriate, which we will discuss next.

**Functional maps.** A popular paradigm for *isometric* shape matching are functional maps (FM) [32, 18, 34],

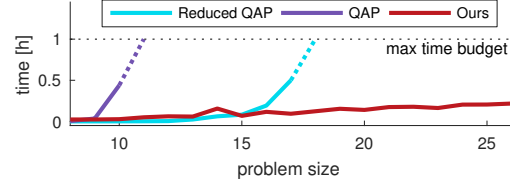


Figure 2. Runtime comparison between our formulation, a QAP and a reduced QAP (cf. search space reduction in the Supp. Mat.) when solved with MOSEK [2].

which define a framework for transferring a function from a source to a target shape. Although FM were shown to be a powerful tool for isometric shape matching, they also have some shortcomings: they are sensitive to noise and suffer from symmetries, point-to-point maps obtained from FM are neither guaranteed to be smooth nor injective, and they are not suitable for severe non-isometries. Rigid shape matching methods applied to spectral embeddings (obtained via FM) can also be used for isometric matching, such as done in PM-SDP [26]. However, in this case the mentioned shortcomings also apply.

**Quadratic assignment problem.** Another popular approach for non-rigid shape matching are formulations based on the quadratic assignment problem (QAP) (or graph matching) [25], which aim for a non-rigid deformation with small distortion. In a discrete setting this can be phrased as matching vertices between two shapes in such a way that pairwise geodesic distances (or similar quantities) are (approximately) preserved by the vertex-to-vertex correspondences. The QAP is known to be NP-hard [35], so that most solution approaches are based on heuristic approaches without formal guarantees, such as e.g. [22]. There are also more principled methods based on convex relaxations, including lifting-free [53, 13, 6] and lifting-based relaxations [40, 46, 20]. However, they do not guarantee to find a globally optimal solution of the *original* non-convex problem as they rely on some kind of rounding procedure to obtain a binary solution. Globally optimal QAP solvers are based on combinatorial search, e.g. via branch and bound [4], and these methods scale exponentially in the number of variables. Similarly as in the QAP, our method also takes the spatial context of matchings into account, but we demonstrate that in practice our proposed formulation is significantly faster to solve, cf. Fig. 2. We believe that this is because our formulation has a special structure (based on our sparse deformation model) that can more efficiently be leveraged by combinatorial solvers.

**Global non-rigid matching.** It was shown that certain matching problems can be solved globally optimal by finding shortest paths in a graph, or based on dynamic programming. These include matching 2D shapes (contours) to a

2D image [11, 14, 41], or matching a 2D contour to a 3D shape [21]. As for example pointed out in [5], non-rigidly matching two objects in 3D is a significantly more difficult problem as it does not allow such a formulation. In [51] the elastic matching of two 3D meshes is addressed based on a linear programming formulation. However, the formulation is sensitive to the mesh triangulation, requires a large number of binary variables, and due to the non-tightness of relaxation it relies on sophisticated rounding techniques after which global optimality cannot be guaranteed anymore. In [10] the authors propose a convex formulation for non-rigid registration that is solved via message passing. This approach requires an extrinsic term in order to disambiguate intrinsic symmetries, which in practice means that an initial alignment between both shapes is indispensable, thereby mitigating the advantages of a convex formulation.

**Local non-rigid matching.** In a similar spirit, local refinement techniques also rely on a good matching initialisation. Such methods include [49, 48] and [27], where a given initial matching is gradually refined. While [49] relies on a QAP formulation, in [27] a spectral method based on FM is used for a hierarchical upsampling. In Sec. 4.3 we show that our method can be used as initialisation for such methods.

In [44] the authors propose a non-rigid deformation model based on per-triangle affine transformations. Within this framework they also pose a correspondence problem, which, however, requires a good initial alignment between both shapes in order to make the optimisation problem well-posed. Moreover, since the problem is non-convex, in general one only finds local optima. In our work we leverage a similar deformation model, but (i) we phrase the problem in a well-posed way without requiring an initial shape alignment, and (ii) we perform a global optimisation.

**Learning-based matching.** Shape matching has also been tackled with machine learning techniques, e.g. with random forests [38], supervised deep functional maps [23], deep functional maps trained in self- or unsupervised settings [17, 39], or using PointNet [33] for learning point cloud correspondences [16]. Undeniably, machine learning has the potential to address many open challenges in shape matching, e.g. for learning appropriate shape representations. In the past it was demonstrated that combinatorial shape matching benefits from learned deep features [5], and reversely, that embedding combinatorial optimisation solvers into neural networks (“differentiable programming”) opens up new possibilities for tackling a range of interesting matching problems [28]. We believe that in the future our method may also be amenable to utilise such synergies, and therefore consider it to be orthogonal to learning-based methods.

**Convex mixed-integer programming.** Mixed-integer programming refers to optimisation problems that involve both continuous and discrete variables. Their advantage is that they are extremely flexible and allow to model a wide range of complex problems. For example, they can be used to discretise difficult non-convex problems, such as formulations that impose rotation matrix constraints, or for phrasing matching problems with binary variables. However, the downside is that MIP problems have a search space that has exponential size in the number of discrete variables, so that in general it is very hard to solve large problems to global optimality. *Convex mixed-integer programming* refers to a subclass of MIP problems that are *convex* for fixed integer variables. A major advantage is that for this class of problems there exist efficient branch and bound solvers that globally optimise such problems. Albeit the fact that these solvers have a worst-case runtime that is exponential in the number of integer variables, in this work we demonstrate that solving non-rigid shape matching using a convex MIP reformulation is tractable in (most) practical scenarios.

### 3. Non-Rigid Shape Matching

First, we summarise our notation. For an integer  $i \in \mathbb{N}$  we define  $[i] := \{1, \dots, i\}$ . For a matrix  $X \in \mathbb{R}^{p \times q}$  and the index set  $\mathcal{I} \subseteq [p]$  we use  $X_{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}| \times q}$  to denote the  $|\mathcal{I}|$  rows of  $X$  selected by  $\mathcal{I}$ .  $\mathbf{1}_n$  and  $\mathbf{I}_n$  denote the  $n$ -dimensional vector of all ones and the  $n$ -dimensional identity matrix,  $\|\cdot\|$  denotes the Frobenius norm, and matrix and vector inequalities are understood element-wise.

Let  $\mathcal{X}$  and  $\mathcal{Y}$  be triangular surface meshes that are discretisations of Riemannian 2-manifolds embedded in 3D space. Note that later in Sec. 4.4 we will also address the case when  $\mathcal{Y}$  is a point cloud. Our aim is to find a non-rigid deformation  $\tau$  that transforms shape  $\mathcal{X}$  to  $\tau(\mathcal{X})$ , so that it aligns well with shape  $\mathcal{Y}$ , cf. Problem (1). For notational convenience we use  $\mathcal{X} \in \mathbb{R}^{n_{\mathcal{X}} \times 3}$  and  $\mathcal{Y} \in \mathbb{R}^{n_{\mathcal{Y}} \times 3}$  to refer to the matrices containing the  $n_{\mathcal{X}}$  and  $n_{\mathcal{Y}}$  3D vertex positions of shapes  $\mathcal{X}$  and  $\mathcal{Y}$ , respectively. Moreover, let  $\mathcal{F}^{\mathcal{X}} \in [n_{\mathcal{X}}]^{f_{\mathcal{X}} \times 3}$  be a matrix that encodes the triangular faces of  $\mathcal{X}$ , where  $f_{\mathcal{X}}$  is the number of triangles.

#### 3.1. Non-Rigid Deformation Model

We model the non-rigid deformation of  $\mathcal{X}$  by applying an affine transformation to each triangle. In conjunction with suitable mesh consistency constraints, the individual per-triangle affine deformations globally constitute a non-rigid deformation. Although related deformation models have been introduced before [43, 44, 5, 45], they have not been used for a global optimisation of non-rigid shape matching.

**Affine per-triangle transformations.** For the  $i$ -th vertex  $\mathcal{X}_i$  we define the non-rigid deformation  $\tau_p$  in terms of its

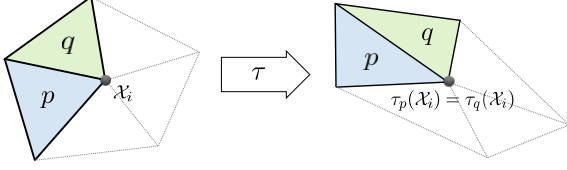


Figure 3. Illustration of the deformation  $\tau$ , where each triangle undergoes an affine transformation. The consistency constraint imposes that the transformed  $\mathcal{X}_i$  is the same, no matter whether it is transformed by  $\tau_p$  or  $\tau_q$  for  $p \in \mathcal{N}_i$  and  $q \in \mathcal{N}_i$ .

adjacent triangle  $p$  as

$$\tau_p(\mathcal{X}_i) := (\mathcal{X}_i - c_p)Q_p + c_p + t_p, \quad (2)$$

where  $c_p \in \mathbb{R}^{1 \times 3}$  is the centroid of the  $p$ -th triangle in the undeformed shape  $\mathcal{X}$ ,  $Q_p \in \mathbb{R}^{3 \times 3}$  is a linear transformation, and  $t_p \in \mathbb{R}^{1 \times 3}$  is a translation. As such, we first centre a given vertex, apply a linear transformation, undo the centring, and eventually translate it to its global position.

**Mesh consistency constraints.** In order to ensure a consistent mesh deformation, we impose the constraints

$$\tau_p(\mathcal{X}_i) = \tau_q(\mathcal{X}_i) \quad \forall i \in [n_{\mathcal{X}}], p \in \mathcal{N}_i, q \in \mathcal{N}_i, \quad (3)$$

where  $\mathcal{N}_i \subset [f_{\mathcal{X}}]$  is the set of all triangles in  $\mathcal{X}$  that are adjacent to vertex  $i$ , cf. Fig. 3. The purpose of the constraints (3) is to enforce that a given vertex  $\mathcal{X}_i$  is transformed to the same place, no matter which transformation of its adjacent triangles is applied, and thereby ensuring that the triangle topology is preserved by the deformation.

### 3.2. Mixed-Integer Non-Rigid Shape Alignment

**Low-dimensional correspondence model.** Our non-rigid deformation  $\tau$  is indirectly defined by a low-dimensional discrete model. To this end, subsets of the shape vertices are used as *control points*, which are represented by the matrices  $\mathcal{X}_{\mathcal{I}} \in \mathbb{R}^{u \times 3}$  and  $\mathcal{Y}_{\mathcal{J}} \in \mathbb{R}^{v \times 3}$ . Here,  $u := |\mathcal{I}|$  and  $v := |\mathcal{J}|$  denote the total number of control points for each shape, and  $\mathcal{I} \subseteq [n_{\mathcal{X}}]$  and  $\mathcal{J} \subseteq [n_{\mathcal{Y}}]$  denote the index sets that select the control points from the original shapes. A similar approach has been pursued in [45] for the interactive *manipulation* of shapes, where, however it is assumed that for each control point of  $\mathcal{X}_{\mathcal{I}}$  the corresponding control point of  $\mathcal{Y}_{\mathcal{J}}$  is already known. In contrast, we are interested in the much more difficult shape *matching* problem, where the correspondence between control points is *unknown*, and, moreover, there may not even exist an *exact* counterpart in  $\mathcal{Y}_{\mathcal{J}}$  for each point in  $\mathcal{X}_{\mathcal{I}}$ .

**Convex polyhedral surface approximation.** We propose to address the issue that there may not exist exact counterparts between control points as follows: rather than matching control points of  $\mathcal{X}_{\mathcal{I}}$  directly to control points of  $\mathcal{Y}_{\mathcal{J}}$ ,

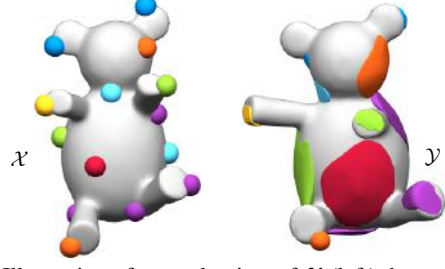


Figure 4. Illustration of control points of  $\mathcal{X}$  (left) that are matched to convex polyhedra of  $\mathcal{Y}$  (right). Colours indicate correspondences between control points and convex polyhedra.

we match the points of  $\mathcal{X}_{\mathcal{I}}$  to *convex polyhedra* that locally approximate the surface of  $\mathcal{Y}$ , see Fig. 4. To this end, we associate a convex polyhedron with each control point of  $\mathcal{Y}_{\mathcal{J}}$ , which we represent using the matrix  $Z_j \in \mathbb{R}^{d_j \times 3}$  for  $j \in [v]$ . Here, each row of  $Z_j$  contains one of the  $d_j \in \mathbb{N}$  vertices (corner points) of the  $j$ -th polyhedron on  $\mathcal{Y}$ . As such, any point that lies inside the convex polyhedron can be specified as a convex combination of rows of  $Z_j$ , i.e.  $\theta^T Z_j$ , where the  $d_j$ -dimensional vector  $\theta$  satisfies the *convex combination constraints*  $\theta \geq \mathbf{0}_{d_j}$  and  $\theta^T \mathbf{1}_{d_j} = 1$ . We note that this point-to-polyhedron matching is a strict generalisation of point-to-point matching, since the latter is achieved for  $d_j=1$ . Using this formulation allows to find a matching between  $\mathcal{X}_{\mathcal{I}}$  and  $\mathcal{Y}_{\mathcal{J}}$  even when there exists only an approximate counterpart between the control points on both shapes. For details how we obtain the polyhedra see the Supp. Mat.

**Correspondence term.** We tackle the non-rigid shape matching problem by establishing correspondences between the control points  $\mathcal{X}_{\mathcal{I}}$  and the convex polyhedra of  $\mathcal{Y}$ , while at the same time ensuring that the resulting non-rigid deformation  $\tau$  is “regular”. For now, let us assume that  $u \leq v$  and that each control point of  $\mathcal{X}_{\mathcal{I}}$  is matched to one of the convex polyhedra of  $\mathcal{Y}$ . Moreover, we also allow that more than one control point of  $\mathcal{X}_{\mathcal{I}}$  can be matched to the same convex polyhedron of  $\mathcal{Y}$ . We model these *matching constraints* using the matrix  $P \in \mathbb{P}_{uv}$ , where we define

$$\mathbb{P}_{uv} := \{P \in \{0, 1\}^{u \times v} : P\mathbf{1}_v = \mathbf{1}_u\}. \quad (4)$$

An element  $P_{ij}=1$  means that the  $i$ -th control point of  $\mathcal{X}_{\mathcal{I}}$  is matched to the  $j$ -th convex polyhedron of  $\mathcal{Y}$ . Later, in Sec. 4.4, we will also present more general formulations that allow to also match shapes when some control points on  $\mathcal{X}$  do not have a counterpart on  $\mathcal{Y}$ .

Since there are  $u$  control points of  $\mathcal{X}_{\mathcal{I}}$ , where each of them is matched to one of the  $v$  convex polyhedra on  $\mathcal{Y}$ , for each  $i \in [u], j \in [v]$  we introduce a convex combination weight vector  $\alpha_{ij} \in \mathbb{R}^{1 \times d_j}$ . Here,  $i$  is the index of the control point of  $\mathcal{X}_{\mathcal{I}}$  and  $j$  is the index of the convex polyhedron of  $\mathcal{Y}$ . By defining  $d := \sum_{j=1}^v d_j$ ,  $Z := [Z_1^T, \dots, Z_v^T]^T \in \mathbb{R}^{d \times 3}$ ,



as well as the matrix of convex combination weights

$$\alpha := [\alpha_{ij}]_{i \in [u], j \in [v]} \in \mathbb{R}^{u \times d}, \quad (5)$$

we model our *correspondence term* as

$$f_{\text{corr}}(\tau, P) := \lambda_c \|\tau(\mathcal{X}_I) - \alpha Z\|. \quad (6)$$

In addition we impose  $\alpha \geq 0$ ,  $\alpha \mathbf{1}_d = \mathbf{1}_u$ , and  $\alpha_{ij} \mathbf{1}_{d_j} \leq P_{ij}$  for  $i \in [u], j \in [v]$ . As such, we can effectively enforce the convex combination and matching constraints using *linear equalities*. With that, the  $(u \times 3)$ -dimensional matrix  $\alpha Z$  contains points that lie inside the convex polyhedra, where each of its  $u$  rows correspond to the respective row of the transformed control point matrix  $\tau(\mathcal{X}_I) \in \mathbb{R}^{u \times 3}$ .

Moreover, to avoid that multiple control points are assigned to the same *vertex* of a convex polyhedron, we impose the “soft-injectivity” constraints  $\mathbf{1}_u^T \alpha \leq \mathbf{1}_d^T$ . The soft-injectivity constraint enforces that the sum of weights in each column of  $\alpha$  is at most one. As such, if a (single) element in a column is exactly one, only this control point is assigned to the respective vertex of the convex polyhedron. If elements in a column of  $\alpha$  are strictly smaller than one, all respective control points are assigned to *non-extreme points* of the polyhedron, thereby preventing that multiple control points are matched to the same *vertex* of a polyhedron.

We use the notation  $(\alpha, P) \in \Gamma$  to refer to the four constraints introduced in this paragraph. In overall, the correspondence term has the purpose to minimise the discrepancy between the control points of the transformed shape  $\tau(\mathcal{X}_I)$  and their corresponding convex polyhedra of  $\mathcal{Y}$ .

**Deformation regularisers.** For regularising the deformation  $\tau$  we decompose each linear transformation  $Q_p$  in (2) into the sum of a rotation matrix  $R \in \text{SO}(3)$  and a (small) general linear part  $T_p \in \mathbb{R}^{3 \times 3}$ , so that  $\tau_p$  in (2) now becomes

$$\tau_p(\mathcal{X}_i) := (\mathcal{X}_i - c_p)(R + T_p) + c_p + t_p. \quad (7)$$

The purpose of using the additive factorisation  $Q_p = R + T_p$  (with  $\|T_p\|$  small) is to ensure that the global shape deformation  $\tau$  (approximately) preserves the morphology of  $\mathcal{X}$ . This has a similar effect as the as-rigid-as-possible (ARAP) model [43], but requires only a single rotation matrix compared to  $f_{\mathcal{X}}$  rotation matrices as used in ARAP. In order to keep the linear part  $T_p$  small, we impose the *rigidity loss* as

$$f_{\text{rigid}}(\tau) := \lambda_r \|[T_1, \dots, T_{f_{\mathcal{X}}}] \|. \quad (8)$$

Moreover, for achieving a locally smooth deformation, we introduce the *smoothness loss*

$$f_{\text{smth}}(\tau) := \lambda_s \|\omega_1 \Delta_1, \dots, \omega_{|\mathcal{E}|} \Delta_{|\mathcal{E}|}\|, \quad (9)$$

where  $\mathcal{E} \subset [f_{\mathcal{X}}]^2$  denotes the set of all neighbouring triangle pairs in  $\mathcal{X}$ , and  $\omega_e$  is a scalar weight. For  $e = (p, q) \in \mathcal{E}$ , so that triangles  $p$  and  $q$  are neighbours, we define the  $e$ -th smoothness residual as

$$\Delta_e = \tau_p(c_q) - \tau_q(c_q) = \tau_p(c_q) - (c_q + t_q). \quad (10)$$

The purpose of the residual  $\Delta_e \in \mathbb{R}^{1 \times 3}$  is to quantify the difference between transforming the triangle centroid  $c_q$  using the transformation  $\tau_q$  of the same triangle, and using the transformation  $\tau_p$  defined for its neighbour triangle  $p$ .

**Optimisation problem.** Based on the introduced terms and constraints our mixed-integer non-rigid alignment (MINA) formulation reads

$$\begin{aligned} \arg \min_{P, \alpha, R, \{T_p\}, \{t_p\}} \quad & f_{\text{corr}}(\tau, P) + f_{\text{rigid}}(\tau) + f_{\text{smth}}(\tau) \quad (11) \\ \text{s.t.} \quad & \tau_p(\mathcal{X}_i) = \tau_q(\mathcal{X}_i), \\ & P \in \mathbb{P}_{uv}, \\ & (\alpha, P) \in \Gamma, \\ & R \in \text{SO}(3). \end{aligned}$$

We assume that all weights  $\lambda_{\bullet} \geq 0$ . The mesh consistency and the  $\Gamma$  constraints are affine in the variables  $P, R, \alpha, T_p$  and  $t_p$ , and all objective function terms  $f_{\bullet}$  are compositions of affine transformations with the Frobenius norm, so that they are convex. However, due to the binary constraints imposed upon  $P$ , and the non-convex quadratic equality constraints  $R \in \text{SO}(3)$ , the overall problem is non-convex.

**Convex mixed-integer formulation.** To transform Problem (11) into a convex MIP problem, we use a piece-wise linear approximation of the  $\text{SO}(3)$  constraint based on binary variables, see the Supp. Mat. and [12]. To keep the number of binary variables small, we use an efficient Gray encoding for the piece-wise linear approximation, cf. [50], so that the number of binary variables is *logarithmic* in the number of discretisation bins  $b$ . The main idea here is to utilise a more efficient representation that requires fewer binary variables and thus admits a more efficient optimisation. In particular, this results in  $6 \cdot \lceil \log_2(b) \rceil$  binary variables, in contrast to  $6 \cdot b$  binary variables for a naive linear encoding. In Fig. 5 we compare our used logarithmic encoding with a linear one, where it can be seen that the logarithmic one requires less computation time, and that the determinant of the resulting matrix is already very close to 1 for  $b=4$ .

## 4. Experiments

In this section we present an experimental evaluation of our proposed MINA approach. To this end, we compare it to other sparse correspondence methods, we analyse the gaps to global optimality, we demonstrate that MINA can be used

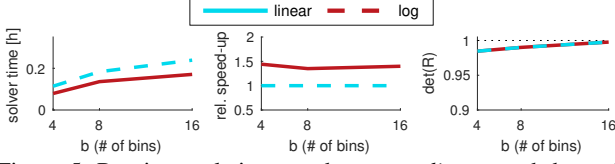


Figure 5. Runtime, relative speed-up w.r.t. *linear*, and determinants of  $R$  for *linear* and *logarithmic* encodings of the  $SO(3)$  discretisation.

as initialisation for dense shape matching, and we showcase its flexibility on several exemplary settings. We provide additional implementation details in the Supp. Mat.

#### 4.1. Sparse Shape Matching

In this section we compare our method with other approaches that perform a sparse matching between a pair of shapes. In particular, we consider the convex matching method PM-SDP [26] in a rigid setting, the sparse game-theoretic approach by Rodola et al. [37], the coherent point drift (CPD) algorithm [30] (randomly initialised), and the convex relaxation by Chen & Koltun [10]. As such, we cover a wide range of shape matching paradigms, including convex relaxations for rigid (PM-SDP) and non-rigid (Chen & Koltun) shape matching, a local non-rigid method (CPD), and a sparse method that considers a quadratic assignment problem formulation (Rodola et al.). In this set of experiments we use the sparse points from [19] for matching pairs of shapes from the TOSCA dataset [9]. Hence, we directly match control points on  $\mathcal{X}$  to control points on  $\mathcal{Y}$  when using our MINA method (i.e.  $d_j=1$  for  $j \in [v]$ ).

In Fig. 6 we show correspondences obtained from our method for various shape matching pairs. In Fig. 7 we show quantitative results, where we summarise the percentage of correct matches (relative to the number of given control points) for each shape class in the TOSCA dataset. It can be seen that our MINA method generally outperforms the other sparse matching approaches. The lower scores for smaller geodesic thresholds arise due to our sparse modelling, since matchings can only be as accurate as the sparse control points allow for. Since the method by Rodola et al. [37] does not match all of the given points, the respective curves do not reach 100%. Moreover, the performance of PM-SDP indicates that a rigid matching setting is too restrictive. Additional results can be found in the Supp. Mat.

#### 4.2. Global Optimality Analysis

Here, we analyse the gaps to global optimality dependent on the processing time  $t$  for the TOSCA shape matching instances in Sec. 4.1. To this end, we define

$$g(t) = \frac{1}{N} \sum_{i: t_i \leq t} (1 - \sigma_i^{\text{rel}}), \quad (12)$$

where  $N$  is the total number of shape matching pairs ( $N=71$  for TOSCA),  $t_i$  denotes the total solver time for the  $i$ -th shape matching problem, and  $\sigma_i^{\text{rel}}$  is the *relative gap* of the  $i$ -th problem that is defined as  $\sigma_i^{\text{rel}} = \frac{|\bar{f} - \underline{f}|}{\max(\delta, |\bar{f}|)}$  (see [2]).

Here,  $\bar{f}$  and  $\underline{f}$  are the upper and lower bounds of the objective value of the MIP formulation of Problem (11), respectively, and  $\delta$  is a small number. In Fig. 8 (left) it can be seen that after 1h (our time budget for the MIP solver) the value of  $g$  reaches 0.98, i.e. on average the solutions are close to being globally optimal (a value of 1 means that all instances are solved to global optimality). After 1h, for 82% of the cases we certify global optimality, see Fig. 8 (right).

#### 4.3. Dense Shape Matching

Next, we demonstrate that our method can be used to obtain a suitable initialisation for dense correspondence methods. Since dense non-rigid matching approaches are highly initialisation-dependent (even the convex approach [10] requires a good initial alignment, cf. Sec. 2), it is crucial that they are provided with a good initialisation.

For these experiments we use the product manifold filter (PMF) [49] for obtaining a dense matching from a given sparse matching. To obtain the initial sparse matching, in addition to our MINA approach, we consider a random matching, a rigid alignment obtained via PM-SDP [26], and the approach by Rodola et al. [37]. Unlike in the previous section, here we extract the sparse points that we want to match based on geodesic farthest point sampling (FPS), which obtains an (approximately) uniform sampling of control points on the shapes.

In Fig. 9 we show results for the PMF-based densification for shapes from the TOSCA dataset [9] (cat, dog, wolf, human), the SHREC watertight dataset [15] (glasses, teddy, pigs), the FAUST dataset [8] (human) and the SCAPE dataset [1] (human). Using a random initialisation (second row) fails in all cases and therefore confirms the dependence of PMF to its initialisation. Although PM-SDP finds a global optimum (of a convex relaxation), the rigid deformation model is too restricted and therefore does not produce reliable dense correspondences for *non-rigid* shape matching (third row). The method by Rodola et al. [37] works well for several cases (fourth row), but due to its initialisation-dependence and potential orientation flips it also leads to several wrong matchings. We find that for various types of matching problems, including strong non-rigid deformations (cat in the first column), or inter-object matching (wolf-dog in the second column), our MINA method provides the most reliable initialisation (last row). Although in many cases MINA is able to properly handle self-symmetries, such symmetries form a particular difficulty for all considered methods and therefore may lead to wrong matchings (last two columns). Another difficulty are drastic

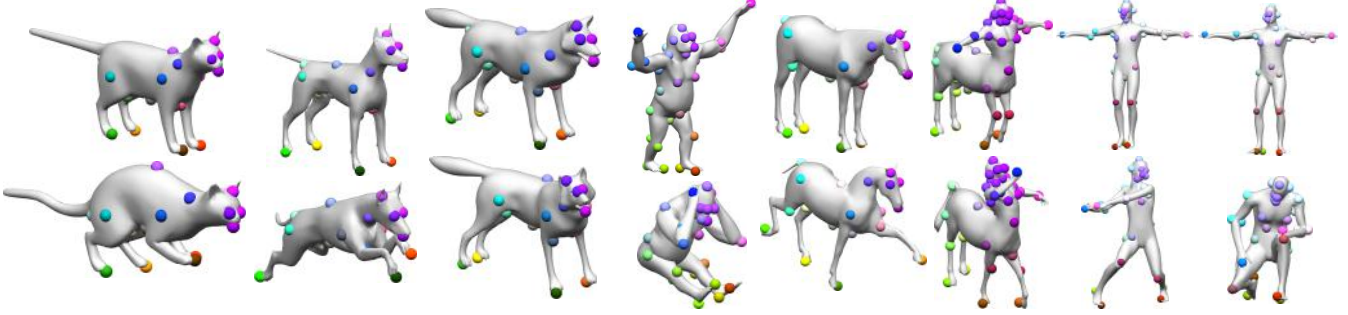


Figure 6. Correspondences obtained from our method for several shape matching instances from the TOSCA dataset [9]. Correspondences between  $\mathcal{X}$  in the top row and  $\mathcal{Y}$  in the bottom row are indicated by dots with corresponding colours.

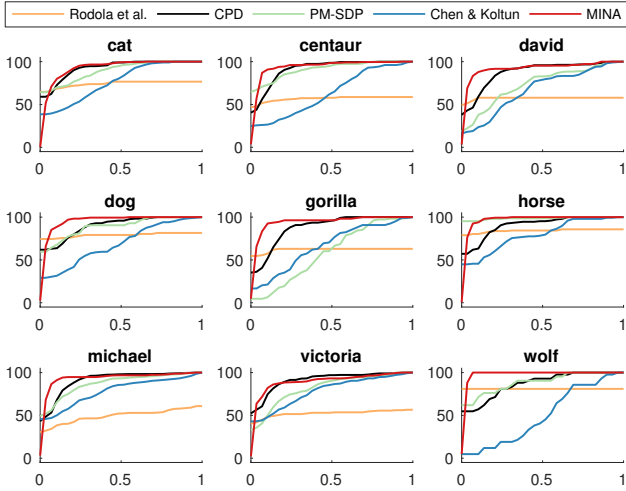


Figure 7. Each plot summarises the percentage of correct matchings for different TOSCA shape classes. The horizontal axis shows the geodesic error threshold, and the vertical axis shows the percentage of matches that are smaller than or equal to this error.

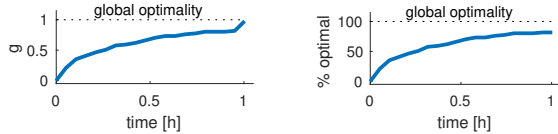


Figure 8. Optimality gaps (left) and proportion of instances solved to global optimality (right) dependent on the solver runtime.

non-rigid deformations (dog in the fourth last column).

#### 4.4. Flexibility of MINA Formulation

Next, we demonstrate the flexibility of our MINA model by addressing several variants of shape matching formulations in a proof-of-concept manner.

**Outlier rejection.** So far, we assumed that there exists a corresponding convex polyhedron on  $\mathcal{Y}$  for each control point  $\mathcal{X}_I$ . In order to allow that some control points of  $\mathcal{X}_I$  are not matched to a convex polyhedron on  $\mathcal{Y}$ , we propose to use an outlier rejection mechanism where up to  $n_{\text{out}}$  of

the control points can remain unmatched. To this end, we replace the correspondence term  $f_{\text{corr}}$  in (6) with

$$\tilde{f}_{\text{corr}} := \lambda_c \|\tau(\mathcal{X}_I) - \alpha Z + \epsilon\|, \quad (13)$$

where  $\epsilon \in \mathbb{R}^{u \times 3}$  is a sparse *error variable* with  $\|\epsilon\|_0 \leq n_{\text{out}}$ . Here, we use  $\|\cdot\|_0$  to denote the row-wise  $\ell_0$ -norm that counts the total number of non-zero rows. To model the  $\ell_0$ -norm as MIP, we introduce the *outlier indicator* variable  $\delta \in \{0, 1\}^u$ , where we impose  $\mathbf{1}_u^T \delta \leq n_{\text{out}}$ . Moreover, we make use of the fact that both shapes are spatially bounded, which implies a bounded correspondence error. With that, we can enforce sparsity of  $\epsilon$  with the linear constraint  $-\delta_i M \mathbf{1}_3^T \leq \epsilon_i \leq \delta_i M \mathbf{1}_3^T$  for a sufficiently large (positive) number  $M$ . As such, whenever  $\delta_i = 1$ , the  $i$ -th control point does not contribute any error towards the term  $\tilde{f}_{\text{corr}}$  since  $\epsilon_i$  will compensate for the discrepancy between  $(\tau(\mathcal{X}_I))_i$  and  $(\alpha Z)_i$ . In Fig. 10 we compare our original formulation with the outlier rejection mechanism, which makes it possible to match pairs of shapes even when the control points are inconsistent between both shapes.

**Shape to point cloud matching.** We used MINA for matching a human body mesh (from [1]) to a real point cloud that we acquired using a *TreedyScan Full Body Scanner*. The raw point cloud was cropped using a manually specified bounding box, downsampled to about 10k points, and denoised. The control points were sampled using geodesic FPS, where we used a nearest neighbour graph for computing geodesics (and estimating normals) on the point cloud. For this experiment we enforce that  $P$  is an injective matching, i.e. we impose  $\mathbf{1}_u^T P \leq \mathbf{1}_v^T$ . In Fig. 1 (middle left) we show the resulting matching, which confirms that our method works well in this setting.

**Different topologies.** We used MINA for matching two human shapes with different topologies, as shown in Fig. 1 (middle right), where the hands in  $\mathcal{X}$  are not touching, whereas the hands in  $\mathcal{Y}$  are touching, as indicated by the geodesic paths between both hands shown as red lines. Here, we used geodesic FPS to sample the control points.





Figure 9. Comparison of different sparse matchings (random, PM-SDP [26], Rodola et al. [37], MINA) that are used as initialisation for PMF [49] to obtain a dense matching. The first row shows the *reference* shape and the other rows show the colour-coded dense correspondences for the respective methods.

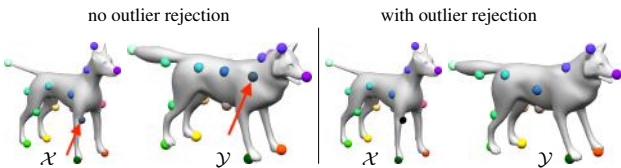


Figure 10. Left: without outlier rejection the dog’s upper thigh is matched to the wolf’s neck (red arrows). Right: our outlier rejection variant (13) effectively disregards this control point inconsistency (the unmatched point on  $\mathcal{X}$  is shown in black).

**Partial shape matching:** We also match a partial shape to a full shape, which we show in Fig. 1 (right). Here, we used geodesic FPS to sample the control points and we enforce that  $P$  is an injective matching, as above.

**Bounded distortion matching.** Our formulation also allows to bound the maximum distortion of a matching. This can be implemented by imposing linear constraints  $P_{st} + P_{pq} \leq 1$  for those  $s, p \in u$  and  $t, q \in v$  where the geodesic distance between points  $s, p$  on  $\mathcal{X}$  and points  $t, q$  on  $\mathcal{Y}$  exceed the maximum allowed distortion. With that, at

most one of the matchings  $P_{st}$  or  $P_{pq}$  is allowed.

## 5. Discussion & Limitations

Although our proposed MINA method has a range of desirable properties, including its high flexibility, its tractability (in practice) due to a low-dimensional matching representation, or its initialisation independence, there are also open points that we aim to address in the future. In Sec. 4.4 we demonstrated that MINA enables matching a mesh to real-world point cloud data. Considering severely cluttered data, cf. [36], or matching shapes with other data representations (e.g. polygon soups) are interesting next steps. A prominent strength of our formulation is that solely using geometric properties already achieves good results. However, additionally incorporating feature descriptors, as commonly done for shape matching, is straightforward and may be useful for further boosting the matching performance.

**Scalability.** Our MINA formulation allows to solve non-rigid shape matching problems with  $u, v$  being of order  $10^2$ . Ideally one would be able to address matching problems



with a much denser sampling of control points, so that more severe non-rigid deformations can be modelled accurately. Although we gained a significant scalability improvement compared to a QAP formulation, cf. Fig. 2, a further reduction of the computational time would be beneficial.

**Multi matching.** The presented MINA formulation is phrased for matching *pairs* of shapes. We believe that multi matching problems would also benefit from related formulations. One potential way for achieving this is to consider all pairwise matching problems (in a symmetric fashion), and coupling these using cycle-consistency constraints.

## 6. Conclusion

We have presented a convex mixed-integer programming formulation for non-rigid shape matching problems, and we have demonstrated that finding the global optimum is tractable in (most) practical scenarios (see Fig. 8). In overall, our formulation comes with a range of benefits: (i) it is more efficient to solve to global optimality compared to the frequently used QAP formulation (Fig. 2), (ii) it is initialisation independent, (iii) it is able to obtain suitable initialisations for dense shape matching methods (Sec. 4.3), and (iv) it is highly flexible in the type of non-rigid shape matching problems it can handle (Sec. 4.4). Although MIP formulations are oftentimes evaded for matching problems in computer vision (due to their high computational complexity), in this work we have shown that a suitable problem-specific modelling indeed allows to solve non-rigid shape matching problems as MIP.

**Acknowledgement:** This work was funded by the ERC Consolidator Grant 4DRePLY.

## References

- [1] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: Shape completion and animation of people. In *SIGGRAPH*, 2005. 6, 7
- [2] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 9.0.*, 2019. 2, 6, 12
- [3] MOSEK ApS. *MOSEK Modeling Cookbook. Release 3.2.1*, 2020. 11
- [4] Mokhtar S Bazaraa and Alwalid N Elshafei. An exact branch-and-bound procedure for the quadratic-assignment problem. *Naval Research Logistics Quarterly*, 26(1):109–121, 1979. 2
- [5] Florian Bernard, Frank R Schmidt, Johan Thunberg, and Daniel Cremers. A combinatorial solution to non-rigid 3d shape-to-image matching. In *CVPR*, 2017. 3
- [6] Florian Bernard, Christian Theobalt, and Michael Moeller. DS\*: Tighter Lifting-Free Convex Relaxations for Quadratic Matching Problems. In *CVPR*, 2018. 2
- [7] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992. 2
- [8] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *CVPR*, 2014. 6
- [9] Alexander Bronstein, Michael Bronstein, and Ron Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer Publishing Company, Incorporated, 1 edition, 2008. 6, 7, 14
- [10] Qifeng Chen and Vladlen Koltun. Robust nonrigid registration by convex optimization. In *CVPR*, 2015. 3, 6
- [11] James Coughlan, Alan Yuille, Camper English, and Dan Snow. Efficient deformable template detection and localization without user initialization. *Computer Vision and Image Understanding*, 78(3):303–319, 2000. 3
- [12] Hongkai Dai, Gregory Izatt, and Russ Tedrake. Global inverse kinematics via mixed-integer convex optimization. *The International Journal of Robotics Research*, May 2017. 5, 11
- [13] Nadav Dym, Haggai Maron, and Yaron Lipman. DS++ - A flexible, scalable and provably tight relaxation for matching problems. *ACM Transactions on Graphics (TOG)*, 36(6), 2017. 2
- [14] Pedro F Felzenszwalb. Representation and detection of deformable shapes. *TPAMI*, 27(2):208–220, 2005. 3
- [15] Daniela Giorgi, Silvia Biasotti, and Laura Paraboschi. Shape retrieval contest 2007: Watertight models track, 2007. 6
- [16] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Unsupervised cycle-consistent deformation for shape matching. In *Computer Graphics Forum*, volume 38, pages 123–133, 2019. 3
- [17] Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *CVPR*, 2019. 3
- [18] Ruqi Huang and Maks Ovsjanikov. Adjoint map representation for shape analysis and matching. In *Computer Graphics Forum*, volume 36, pages 151–163, 2017. 2
- [19] Vladimir G. Kim, Yaron Lipman, and Thomas Funkhouser. Blended Intrinsic Maps. *ACM Transactions on Graphics (TOG)*, 30(4), 2011. 6
- [20] Yam Kushinsky, Haggai Maron, Nadav Dym, and Yaron Lipman. Sinkhorn algorithm for lifted assignment problems. *SIAM Journal on Imaging Sciences*, 12(2):716–735, 2019. 2
- [21] Zorah Löhner, Emanuele Rodola, Frank R Schmidt, Michael M Bronstein, and Daniel Cremers. Efficient globally optimal 2d-to-3d deformable shape matching. In *CVPR*, 2016. 3
- [22] D Khuê Lê-Huu and Nikos Paragios. Alternating direction graph matching. In *CVPR*, 2017. 2
- [23] Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *CVPR*, 2017. 3
- [24] Johan Lofberg. Yalmip: A toolbox for modeling and optimization in matlab. In *ICRA*, 2004. 11

- [25] Eliane Maria Loiola, Nair Maria Maia de Abreu, Paulo Oswaldo Boaventura Netto, Peter Hahn, and Tania Maia Querido. A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176(2):657–690, 2007. 2
- [26] Haggai Maron, Nadav Dym, Itay Kezurer, Shahar Kovalsky, and Yaron Lipman. Point registration via efficient convex relaxation. *ACM Transactions on Graphics (TOG)*, 35(4):73, 2016. 2, 6, 8, 11, 12
- [27] Simone Melzi, Jing Ren, Emanuele Rodolà, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics (TOG)*, 38(6), 2019. 3
- [28] Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with gumbel-sinkhorn networks. In *ICLR*, 2018. 3
- [29] James Munkres. Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, Mar. 1957. 11
- [30] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *TPAMI*, 32(12):2262–2275, 2010. 2, 6
- [31] Carl Olsson, Fredrik Kahl, and Magnus Oskarsson. Branch-and-bound methods for euclidean registration problems. *TPAMI*, 31(5):783–794, 2008. 2
- [32] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30, 2012. 2
- [33] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017. 3
- [34] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. In *SIGGRAPH Asia*, 2018. 2
- [35] F Rendl, P Pardalos, and H Wolkowicz. The quadratic assignment problem: A survey and recent developments. In *Proceedings of the DIMACS workshop on quadratic assignment problems*, volume 16, pages 1–42, 1994. 2
- [36] Emanuele Rodolà, Andrea Albarelli, Filippo Bergamasco, and Andrea Torsello. A scale independent selection process for 3d object recognition in cluttered scenes. *IJCV*, 102(1-3):129–145, 2013. 8
- [37] Emanuele Rodolà, Alex M Bronstein, Andrea Albarelli, Filippo Bergamasco, and Andrea Torsello. A game-theoretic approach to deformable shape matching. In *CVPR*, 2012. 6, 8
- [38] Emanuele Rodolà, Samuel Rota Buló, Thomas Windheuser, Matthias Vestner, and Daniel Cremers. Dense non-rigid shape correspondence using random forests. In *CVPR*, 2014. 3
- [39] Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *ICCV*, 2019. 3
- [40] Christian Schellewald and Christoph Schnörr. Probabilistic subgraph matching based on convex relaxation. In *EMM-CVPR*, 2005. 2
- [41] Thomas Schoenemann and Daniel Cremers. A combinatorial solution for model-based image segmentation and real-time tracking. *TPAMI*, 32(7):1153–1164, 2009. 3
- [42] Peter H Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. 2
- [43] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, 2007. 3, 5
- [44] Robert W Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Transactions on graphics (TOG)*, 23(3):399–405, 2004. 3
- [45] Robert W Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. *ACM Transactions on Graphics (TOG)*, 26(3):80, 2007. 3, 4, 12
- [46] Paul Swoboda, Carsten Rother, Hassan Abu Alhaija, Dagmar Kainmüller, and Bogdan Savchynskyy. Study of lagrangean decomposition and dual ascent solvers for graph matching. In *CVPR*, 2017. 2
- [47] Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. In *Computer Graphics Forum*, volume 30, pages 1681–1707. Wiley Online Library, 2011. 2
- [48] Matthias Vestner, Zorah Lähner, Amit Boyarski, Or Litany, Ron Slossberg, Tal Remez, Emanuele Rodola, Alex Bronstein, Michael Bronstein, Ron Kimmel, and Daniel Cremers. Efficient deformable shape correspondence via kernel matching. In *3DV*, 2017. 3
- [49] Matthias Vestner, Roei Litman, Emanuele Rodolà, Alex Bronstein, and Daniel Cremers. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *CVPR*, 2017. 3, 6, 8
- [50] Juan Pablo Vielma and George L Nemhauser. Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. *Mathematical Programming*, 128(1-2):49–72, 2011. 5, 11
- [51] Thomas Windheuser, Ulrich Schlickewei, Frank R Schmidt, and Daniel Cremers. Geometrically consistent elastic matching of 3d shapes: A linear programming solution. In *ICCV*, 2011. 3
- [52] Jiaolong Yang, Hongdong Li, and Yunde Jia. Go-icp: Solving 3d registration efficiently and globally optimally. In *CVPR*, 2013. 2
- [53] Feng Zhou and Fernando De la Torre. Factorized Graph Matching. *TPAMI*, 38(9):1774–1789, 2016. 2

# Supplementary Material

## A. Obtaining Convex Polyhedra on $\mathcal{Y}$

Given the  $j$ -th control point of  $\mathcal{Y}_{\mathcal{T}}$ , we obtain its associated convex polyhedron using a neighbourhood propagation strategy. To this end, we define a *planarity criterion* using the maximum of the *mean absolute deviation* (MAD) of the surface normals at the points in  $Z_j$ . For a given matrix  $N \in \mathbb{R}^{n \times 3}$  and its column mean  $\bar{N} \in \mathbb{R}^{1 \times 3}$ , the MAD is defined as  $\text{mad}(N) := \frac{1}{n} \sum_i |N_i - \bar{N}|$ . As such, starting with  $t=0$ , we consider the vertices of the  $t$ -ring of the  $j$ -th vertex as  $Z_j$ , where we increase  $t$  as long as  $\max(\text{mad}(N_j^t)) \leq \eta$ . Here,  $N_j^t$  denotes the matrix of the normals of the  $t$ -ring of the  $j$ -th vertex and  $\eta$  specifies a user-defined threshold. Once we have determined the largest  $t$  such that the  $t$ -ring neighborhood is sufficiently planar (below the threshold  $\eta$ ), we discard all points in the rows of  $Z_j$  that are interior vertices of the convex polygon defined by  $Z_j$  (as they are redundant). In Fig. 4 (right) we show so-obtained convex polyhedra.

## B. Piece-wise Linear Approximation of $\text{SO}(3)$ Constraints

The constraint  $R \in \text{SO}(3)$  can be expressed as the orthogonality constraint  $R^T R = I_3$  in combination with  $R_1 \times R_2 = R_3$ . Hence, the constraint  $R \in \text{SO}(3)$  comprises exactly 6 quadratic equality constraints, which form a non-convex set. In order to define a piece-wise linear approximation we use *specially-ordered set of type 2* (sos2) variables. An sos2 variable is a non-negative vector where at most two consecutive element can be non-zero. With that, such a variable allows to encode a non-convex quadratic function in terms of a piece-wise linear one, so that in the end all quadratic constraints become linear, and the sos2 constraints are imposed based on (few) binary variables.

For illustrative purposes, we will now provide a simple example for a piece-wise linear approximation of a quadratic function. Let us consider the function  $h(x) = x^2$  on the interval  $[-1, 1]$ . First, we split the domain into  $b$  bins, so that we evaluate  $x^2$  at these  $b$  discrete positions, and then compute all values that fall in-between the sampled points as linear approximation between its two neighbour sample points. Let  $b = 4$ , and let  $\phi = [-1, -0.5, 0, 0.5, 1]^T$  be a vector that contains the discretised domain, so that  $\phi^2 = [1, 0.25, 0, 0.25, 1]^T$  defines the elementwise square of  $\phi$ . Moreover, let  $\lambda \in \mathbb{R}^{b+1}$  be a non-negative sos2 variable that sums to one (as mentioned, sos2 means that only two consecutive elements can be non-zero). Then, we can approximate

$$h(x) \approx \lambda^T \phi^2 \quad \text{for} \quad x = \lambda^T \phi. \quad (14)$$

For example, for  $x = 0.75$ , we obtain the sos2 variable  $\lambda = [0, 0, 0, 0.5, 0.5]^T$  (since  $x = 0.75 = \lambda^T \phi$ ). With that, we obtain  $h(0.75) = 0.5625 \approx 0.625 = \lambda^T \phi^2$ . The important property is that (14) allows to approximate the quadratic function  $h(\cdot)$  based on a representation that is *linear* in the variables  $x$  and  $\lambda$ . In addition to [12] and [50], we refer the interested reader to [3, Ch. 9.1.11]<sup>1</sup>, where sos2 constraints as well as the idea of using a logarithmic Gray encoding are explained.

## C. Search Space Reduction

In addition to using a logarithmic encoding of the  $\text{SO}(3)$  discretisation variables, we also impose further constraints upon the matching matrix  $P$ , so that the size of the overall search space can be reduced. A similar idea has also been pursued in [26], where a scalar criterion based on the average geodesic distance (ADG) was used. In contrast, rather than using a single scalar value for each vertex, we propose to leverage a more powerful approach that considers more descriptive statistics of geodesic distances, see Fig. 11. To this end, for each control point we compute  $n_{\text{prctile}}$  evenly spaced percentiles from 0 to 100% of the geodesic distance from this control point to all other points. Let  $\gamma^{\mathcal{X}} \in \mathbb{R}^{u \times n_{\text{prctile}}}$  and  $\gamma^{\mathcal{Y}} \in \mathbb{R}^{v \times n_{\text{prctile}}}$  denote the so-obtained percentile matrices, where the columns are the ordered percentiles from 0 to 100%. As such, the matrices  $\gamma^{\mathcal{X}}$  and  $\gamma^{\mathcal{Y}}$  can be seen as features of the respective shapes extracted at the control points. Whenever two control points  $i \in [u], j \in [v]$  correspond to each other, the features  $\gamma_i^{\mathcal{X}}$  and  $\gamma_j^{\mathcal{Y}}$  should be similar, so that  $d_{ij} := \|\gamma_i^{\mathcal{X}} - \gamma_j^{\mathcal{Y}}\|$  is small. Based on this observation, we use the feature distances  $[d_{ij}]_{i,j}$  and sequentially solve  $n_{\text{LAP}}$  linear assignment problems (LAP) [29] to match features. The idea of solving a sequence of LAPs is to not only find the single best matching  $P_1$  of features, but rather finding multiple solutions  $P_1, \dots, P_{n_{\text{LAP}}}$ , so that the nonzero elements in  $P_{\text{all}} = \sum_{\ell=0}^{n_{\text{LAP}}} P_{\ell}$  define the allowed matchings in  $P$ . Here, the matrix  $P_{\ell}$  is obtained by performing a feature matching using  $[d_{ij}]_{i,j}$  when forbidding all previous matchings  $P_1, \dots, P_{\ell-1}$ . As such, when optimising MINA, we constrain all elements of  $P$  to be zero for those elements where  $P_{\text{all}}$  is zero. Using this procedure is advantageous over simple thresholding of  $[d_{ij}]_{i,j}$ , since on the one hand feasibility is guaranteed, and on the other hand the number of allowed matchings is equal for all control points.

## D. Further Implementation Details

We have implemented MINA in the optimisation modelling toolbox Yalmip [24], which uses the conic mixed-

<sup>1</sup>also available online at

<https://docs.mosek.com/modeling-cookbook/mio.html#continuous-piecewise-linear-functions>

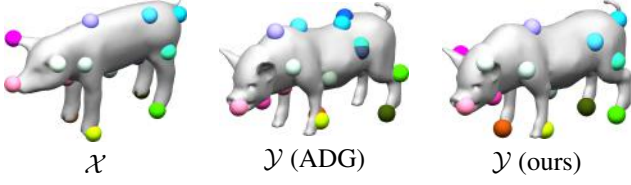


Figure 11. Shape  $\mathcal{X}$  (left) is matched to  $\mathcal{Y}$ , where a search space reduction using ADG [26] leads to wrong matchings (middle), whereas ours produces correct correspondences (right).

integer branch and bound solver MOSEK [2] as backend (with default parameters). In all experiments we used  $\lambda_c=4$ ,  $\lambda_r=1$  and  $\lambda_s=0.5$ , where we account for different problem sizes by multiplying each  $\lambda_\bullet$  with  $\frac{1}{\sqrt{\#}}$ , where  $\#$  denotes the total number of elements that the norm is applied to. We set the weights  $\omega_e$  for the smoothness term to  $\omega_e = \frac{d_e}{\sum_{e \in \mathcal{E}} d_e}$ , where for  $e = (p, q)$  by  $d_e$  we denote the length of the common edge of triangles  $p, q$ . With that, we achieve that the deformation of two adjacent triangles  $p, q$  is more flexible when their common edge is small. We set the planarity threshold to  $\eta = \frac{1}{2}$ . For keeping the number of variables small, for each convex polyhedron  $Z_j$  we only keep the respective control point as well as four additional points obtained via farthest point sampling (FPS) using geodesic distances as metric. Note that this results in convex polyhedra that are either a single point (if none of the  $t$ -rings of the  $j$ -th control point satisfies the planarity criterion), or  $Z_j$  is a  $5 \times 3$  matrix. Since the non-rigid deformation induced by a sparse set of matched control points is relatively coarse, rather than modelling  $\tau$  with the original mesh resolution we use downsampled meshes with about 300 faces, similarly as in [45]. We set  $n_{\text{LAP}}=5$ ,  $n_{\text{prtile}} = \min(n_{\mathcal{X}}, n_{\mathcal{Y}})$ ,  $M=0.2$  and use  $b=4$  bins for the  $\text{SO}(3)$  discretisation.

Next, we provide additional details on shape to point cloud matching and the relation between partial shape matching and outlier rejection.

**Shape to point cloud matching.** The main difference when  $\mathcal{Y}$  is represented as a point cloud rather than a mesh is that we need to use a different approach for computing geodesic distances and normals (required for sampling control points, for the definition of the convex polyhedra as described in Sec. A, and for the search space reduction described in Sec. C). In our case we compute geodesic distances and normals based on a nearest neighbour graph, where we use the 3 nearest neighbours. After this information is obtained, the overall optimisation problem is equivalent to the one when  $\mathcal{Y}$  is a mesh, since the only information of  $\mathcal{Y}$  that is explicitly used in our optimisation problem formulation are the convex polyhedra.

**Relation between partial shape matching and outlier rejection.** In our considered *partial shape matching* setting we match *all* control points of the partial shape  $\mathcal{X}$  to the full shape  $\mathcal{Y}$ . This is in contrast to our *outlier rejection* setting, where we allow that some control points of  $\mathcal{X}$  are not matched to  $\mathcal{Y}$ . However, although for partial shape matching we do not use outlier rejection, we mention that principally it could be used for matching a full shape to a partial one.



## E. Additional TOSCA Results

In Fig. 12 we report runtime statistics over all 71 shape matching instances from the TOSCA datasets for all considered methods. On this dataset, the median processing time of our method is  $\approx 15\text{min}$ , whereas the other methods require less than one minute.

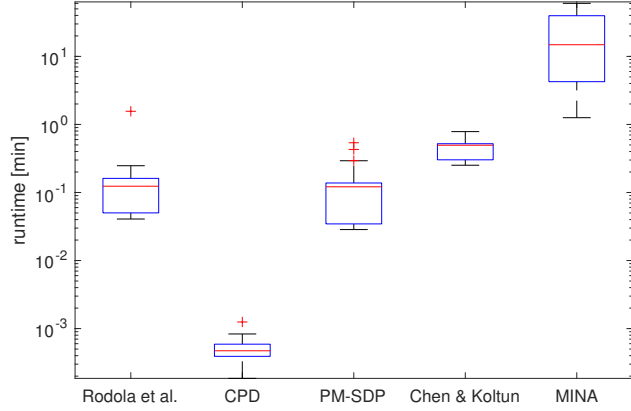


Figure 12. Runtime statistics for the TOSCA dataset. Note that the vertical axis is shown in log-scale.

In Fig. 13 we present further results where also the deformed shape  $\tau(\mathcal{X})$  is shown.

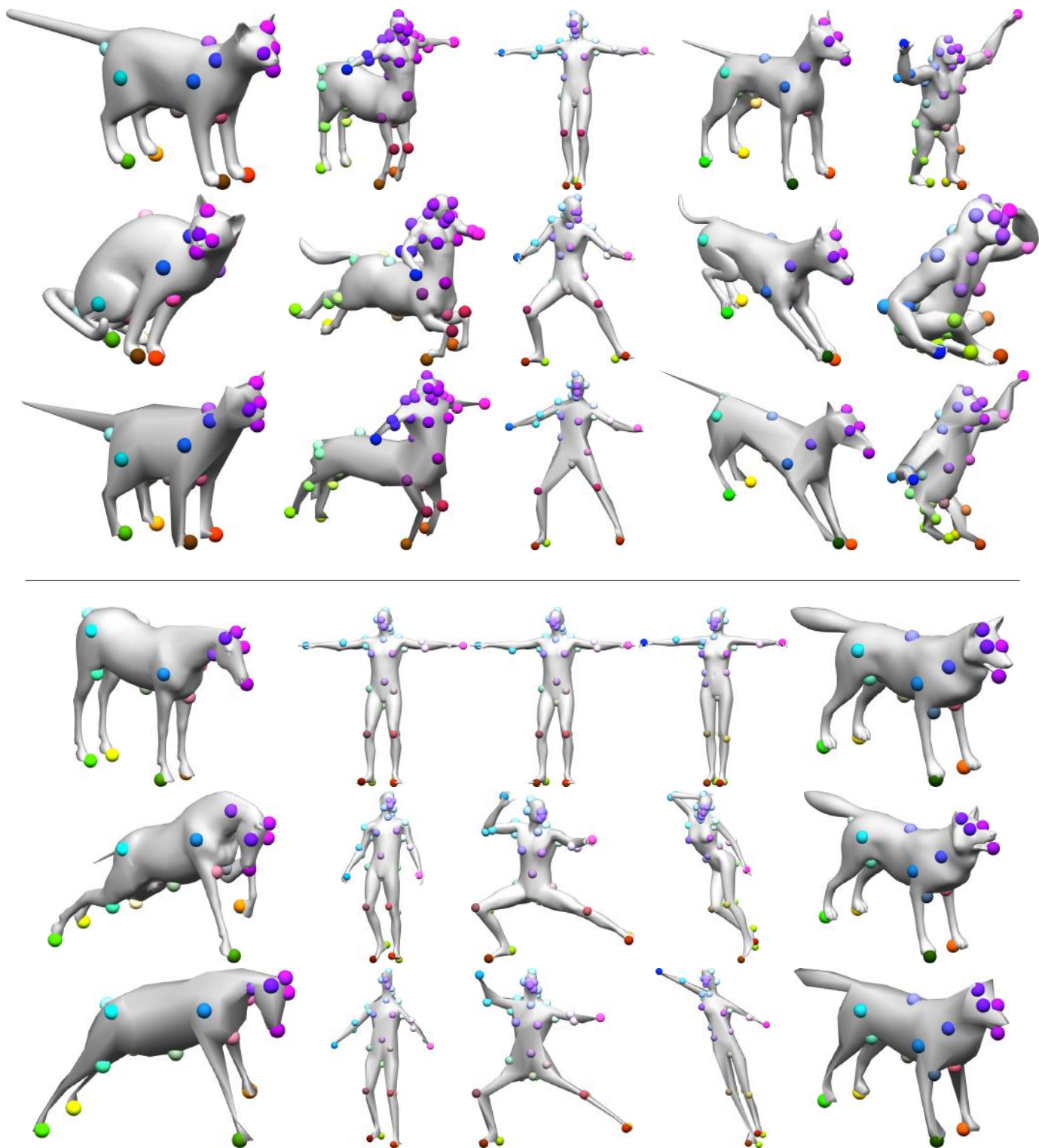


Figure 13. Correspondences obtained from our method for several shape matching instances from the TOSCA dataset [9]. Correspondences are indicated by dots with corresponding colours. In each triplet of rows we show  $\mathcal{X}$ ,  $\mathcal{Y}$ , and the deformed shape  $\tau(\mathcal{X})$  from top to bottom. Note that the deformation  $\tau$  is not always able to obtain a good alignment (particularly for severe non-rigid transformations, e.g. the cat or the gorilla), but the correspondences are still reasonable in many cases.