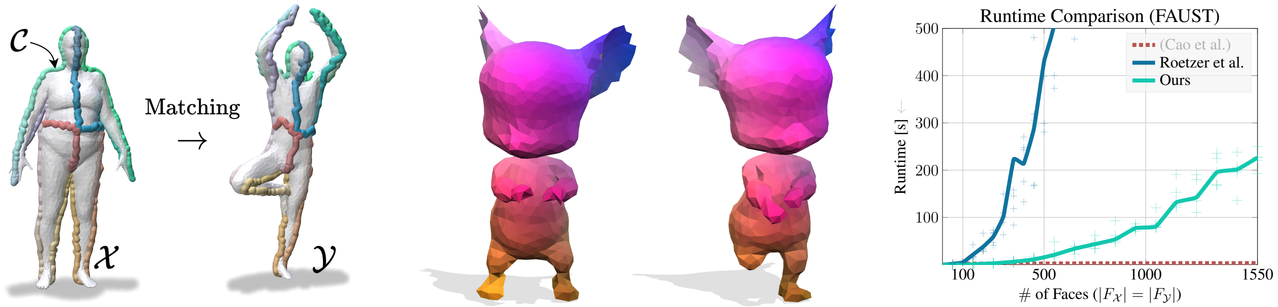
SpiderMatch: 3D Shape Matching with Global Optimality and Geometric Consistency

Paul Roetzer[1] Florian Bernard[1]
University of Bonn[1]



Idea: Match Cyclic Path $\mathcal{C}$ on $\mathcal{X}$ to $\mathcal{Y}$ Dense Geometrically Consistent Shape Matching Scalability Analysis

Figure 1. (**Left**) Schematic illustration of our method: we extract a cyclic path $\mathcal{C}$ on the surface of 3D shape $\mathcal{X}$, and match $\mathcal{C}$ to the 3D target shape $\mathcal{Y}$. (**Middle**) If $\mathcal{C}$ covers the whole source shape $\mathcal{X}$ (not shown for clarity, see Sec. 4.1 for details), our method can be used to compute globally optimal results for dense non-rigid and geometrically consistent 3D shape matching. (**Right**) We show runtimes w.r.t. shape resolution and compare to Roetzer et al. [55] (which is the only existing method that achieves geometric consistency while having a global flavour and being initialization-free) and Cao et al. [14] (which does not provide any geometric consistency guarantees). Ours solves all instances to global optimality (with geometric consistency), while in addition being much faster than Roetzer et al.

Abstract

Finding shortest paths on product spaces is a popular approach to tackle numerous variants of matching problems, including the dynamic time warping method for matching signals, the matching of curves, or the matching of a curve to a 3D shape. While these approaches admit the computation of globally optimal solutions in polynomial time, their natural generalisation to 3D shape matching is widely known to be intractable. In this work we address this issue by proposing a novel path-based formalism for 3D shape matching. More specifically, we consider an alternative shape discretisation in which one of the 3D shapes (the source shape) is represented as a SpiderCurve, *i.e. a long self-intersecting curve that traces the 3D shape surface. We then tackle the 3D shape matching problem as finding a shortest path in the product graph of the SpiderCurve and the target 3D shape. Our approach introduces a set of novel constraints that ensure a globally geometrically consistent matching. Overall, our formalism leads to an integer linear programming problem for which we experimentally show that it can efficiently be solved to global optimality. We demonstrate that our approach is competitive with recent state-of-the-art shape matching methods, while in addition guaranteeing geometric consistency.*[1]

1. Introduction

The 3D shape matching problem refers to finding correspondences between given 3D shapes and has a high relevance across the broad field of visual computing. Examples of applications include texture transfer in graphics, statistical shape analysis in medical imaging, or 3D reconstruction or shape completion in computer vision. While the state of the art in 3D shape matching relies on data-driven deep learning methods [14], respective methods have the strong shortcoming that they lack guarantees about structural properties of obtained matchings. One crucial example, which is the main focus of our work, is geometric consistency: a matching between a given pair of 3D shapes is geometrically consistent if the neighbourhood between shape elements (e.g. triangles) is preserved under the matching.

[1] https://github.com/paul0noah/spider-match

Loosely, *geometric consistency* (in the discrete domain) can be seen analogously to diffeomorphisms in the continuous domain [77].

Achieving geometric consistency in learning-based shape matching solutions is extremely challenging, since it requires to impose complex structural constraints on the prediction. While there are some optimisation-based works that explicitly aim for geometric consistency, they require initial pre-matchings to handle the severe non-convexity of respective optimisation problems [23, 24, 43, 61, 62, 64, 66, 69, 75], or they employ heuristics and only find approximate solutions [55, 78]. To date, there do not exist any 3D shape matching methods that are able to find globally optimal solutions while guaranteeing geometric consistency.

In this work we make a first step towards filling this gap. More specifically, motivated by recent advances in (polynomial-time solvable) matching problems between a curve and a 3D shape [37, 56], we propose to represent the source 3D shape as a *SpiderCurve*, i.e. a long self-intersecting curve that traces the 3D surface, loosely analogous to a spider web covering the shape surface. We then find a shortest path in the product graph of the SpiderCurve and the target 3D shape, while respecting geometric consistency constraints. We cast this as an integer linear program, for which we experimentally demonstrate that it can efficiently be solved to global optimality in all considered problem settings. Overall, we summarise our main contributions as follows (also cf. Tab. 1):

- For the first time we enable to find globally optimal solutions for geometrically consistent 3D shape matching.
- To achieve this, we introduce a novel integer linear programme (ILP) that consistently matches our SpiderCurve 3D shape representation to the target 3D shape.
- Although ILP solvers employ branch and bound procedures that have exponential worst-case runtime in general, we experimentally demonstrate that our formalism allows to efficiently find solutions for problems with practically relevant sizes (e.g. we use the off-the-shelf ILP solver Gurobi to match shapes with 1000 triangles in ≈100s.).

	Geometrically	**Globally**	**Keypoint**	**Pruning**
Method	**Consistent**	**Optimal**	**Free**	**Free**
SIGMA [26]	✗	(✓)	✗	✗
MINA [10]	✗	(✓)	✗	✗
PMSDP [47]	✗	✗	✓	✗
Roetzer et al. [55]	(✓)	✗	✓	✓
Ours	✓	✓	✓	✓

Table 1. Comparison of properties of **shape matching methods**. Ours is the only one that finds geometrically consistent and globally optimal solutions. Moreover, our method does not require the definition of keypoints, nor does it exploit any search space pruning scheme to solve the resulting integer linear program.

2. Related Work

In the following we discuss works that are most relevant to our method. For a broad overview on 3D shape matching and registration we refer interested readers to the survey papers [17, 70, 73].

Deep Shape Matching. Data-driven deep learning methods constitute the state of the art in 3D shape matching, as they can predict high-quality correspondences in short time. Most of these methods are based on the popular functional map framework [49], as it offers a convenient way to regularise the high-dimensional shape matching problem via a low-dimensional representation. Respective methods were trained both in a supervised manner [28, 40, 42, 72, 76], and in an unsupervised manner [5, 6, 12–14, 18, 19, 30, 33, 39, 46, 51, 57, 65, 68, 71]. A major leap forward was achieved with the introduction of DiffusionNet [68], which builds upon a simple diffusion process to perform learning on 3D surfaces. The method [14] builds on top of DiffusionNet while coupling functional maps with point maps, and thereby achieves state-of-the-art shape matching performance in very challenging scenarios (e.g. partial shapes and topological noisy shapes). Despite the astonishing performance of recent learning-based shape matching methods, they have the major shortcoming that the produced matchings are not geometrically consistent, see e.g. [21] for a recent demonstration.

Geometrically Consistent Shape Matching. We argue that geometric consistency is an essential property of 3D shape matching. Geometric consistency is generally hard to achieve since it leads to challenging non-convex optimisation problems, see e.g. [77]. Thus, there exists many local methods that require given correspondences between a sparse set of points [64] (e.g. often user-defined). Among them is the work [66] based on heat diffusion, and the work [75] in which given matchings are smoothed with the so-called product manifold filter. Furthermore, the works [23, 24] also consider the refinement of an initial set of matchings. The works [61, 62, 69] solve for geometrically consistent intrinsic triangulations [67] via local optimisation. Many of these methods are parameterisation-based, i.e. they rely on mappings of given 3D shapes to simple parametric surfaces, such as planes [1, 2, 34, 43, 44, 48, 79] or spheres [1, 3, 4, 50, 61]. One shortcoming of parameterisation-based methods is that they are only applicable for matching shapes with a fixed genus. Furthermore, respective methods involve highly non-convex optimisation problems that depend on good initialisations (e.g. user-provided sparse matchings).

Globally Optimal Shape Matching. While the polynomial-time solvable linear assignment problem (LAP) could in principle be used for 3D shape matching, it completely disregards geometric relations, so that neighbouring points may be matched to arbitrarily far away points. While the LAP matches vertices, the quadratic assignment problem (QAP) matches pairs of vertices (edges), so that neighbourhood relations can be better preserved. Yet, the QAP is NP-hard [54] so that shape matching instances with practically relevant sizes cannot efficiently be solved. Several attempts to tackle the QAP have been presented, including convex relaxations [9, 20, 36, 59], and heuristics based on local optimisation [31, 38, 74].

Other attempts towards global shape matching are often based on some low-dimensional (or low-rank) representation, e.g. functional maps in the continuous domain [22, 27, 47, 49], functional maps coupled to point maps [53], or sparse matching representations in the discrete domain [10, 26]. Yet, functional map-based methods do not take geometric consistency into account, and sparse discrete models are typically slow and heavily rely on an aggressive pruning of the search space [10, 26, 47].

A different class of global shape matching approaches is based on finding shortest paths in product graphs. These include the matching of signals in the popular dynamic time warping algorithm [58], the matching of 2D shapes to images to address shape-based image segmentation [16, 25, 63], or the matching of 2D shapes to each other [60]. Recently, a similar framework has been presented to match a 2D shape (a cyclic curve) to a 3D shape [37, 56]. All of these approaches have in common that solutions can efficiently be computed in polynomial time based on a shortest path algorithm (or some variant thereof). While product space formalisms have been generalised for the matching of 3D shapes [55, 77, 78], the respective problem is substantially harder and does not admit a solution via finding a shortest path – this is because in this case the matching cannot be represented as a one-dimensional path, but forms a two-dimensional surface in a four-dimensional product space. In this work we consider an alternative view that builds upon a different shape discretisation and thus allows to tackle 3D shape matching from a path-based perspective.

3. Background on 2D-3D Shape Matching

In the following section, we summarise the important product graph formalism introduced in [37] for 2D-3D shape matching. It forms the foundation of our SpiderCurve-based 3D shape matching approach.

Lähner et al. [37] aim to find correspondences between a 2D shape $\mathcal{C} = (V^{\mathcal{C}}, E^{\mathcal{C}})$ (a cyclic curve discretised as a cyclic chain graph, i.e. a graph with circular topology.) and a 3D shape $\mathcal{Y} = (V^{\mathcal{Y}}, E^{\mathcal{Y}})$ (interpreted as the undirected graph of a surface mesh). The matching between $\mathcal{C}$ and $\mathcal{Y}$ is found as a shortest path in the product graph

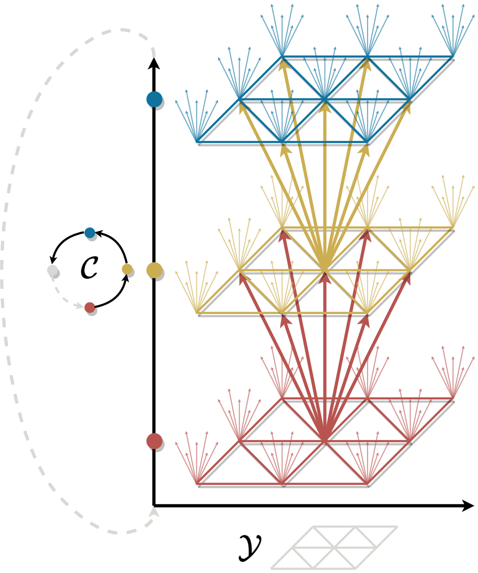


Figure 2. The **product graph** $\mathcal{P}$ is structured into layers (individual colours), where each layer corresponds to the product of a specific vertex $v^{\mathcal{C}} \in V^{\mathcal{C}}$ of the 2D shape $\mathcal{C}$ and the entire 3D shape $\mathcal{Y}$. $\mathcal{P}$ contains *intra-layer* edges (edges within a layer) and *inter-layer* edges (edges connecting neighbouring layers). To avoid clutter, we only show the inter-layer connections for one vertex in the red and yellow layers. Like $\mathcal{C}$, the product graph $\mathcal{P}$ is also cyclic.

$\mathcal{P} = (V^{\mathcal{P}}, E^{\mathcal{P}})$, where

$$\begin{aligned} V^{\mathcal{P}} &= V^{\mathcal{C}} \times V^{\mathcal{Y}}, \\ E^{\mathcal{P}} &= \{ (e^{\mathcal{C}}, e^{\mathcal{Y}}) \in V^{\mathcal{P}} \times V^{\mathcal{P}} \mid e^{\mathcal{C}} \in V^{\mathcal{C}} \times V^{\mathcal{C}}, \\ &\qquad e^{\mathcal{Y}} \in V^{\mathcal{Y}} \times V^{\mathcal{Y}},\ e^{\mathcal{C}} \text{ or } e^{\mathcal{Y}} \text{ non-deg.} \}. \end{aligned} \tag{1}$$

In Eq. (1), degenerate edges mean that they consist of the same vertex twice. This is necessary to handle different discretisations of the shapes. In Fig. 2 we show the structure of the product graph: $\mathcal{P}$ is oragnised into layers, where each layer corresponds to the product of a specific vertex of the 2D shape $\mathcal{C}$ and the entire 3D shape $\mathcal{Y}$. Thus, every vertex $v = (v^{\mathcal{C}}, v^{\mathcal{Y}}) \in V^{\mathcal{P}}$ consists of a vertex $v^{\mathcal{C}} \in V^{\mathcal{C}}$ from the 2D shape $\mathcal{C}$, and a vertex $v^{\mathcal{Y}} \in V^{\mathcal{Y}}$ from the 3D shape $\mathcal{Y}$. Any cyclic path in the product graph $\mathcal{P}$ that goes through all layers represents a (geometrically consistent) matching between $\mathcal{C}$ and $\mathcal{P}$. By defining appropriate edge costs (e.g. based on shape feature similarity, or by penalising deformations), one can find a globally optimal and geometrically consistent matching between the 2D shape $\mathcal{C}$ and the 3D shape $\mathcal{Y}$ via a (cyclic) shortest paths algorithm. The works [37, 56] propose custom algorithms that exploit the special structure of the product graph in order to efficiently find such matchings. Overall, our main idea is to generalise such 2D-3D shape matching formalisms towards 3D-3D shape matching with the help of a SpiderCurve representation.

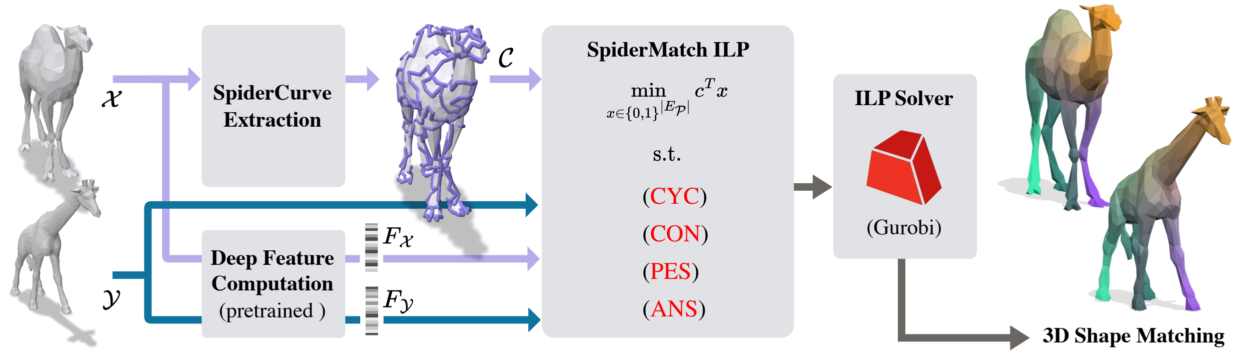


Figure 3. Overview of the **pipeline of our method.** We consider two 3D surface meshes $\mathcal{X}$ and $\mathcal{Y}$ for which we compute vertex-wise features with a pretrained feature extractor. After discretising $\mathcal{X}$ into a SpiderCurve representation $\mathcal{C}$, we construct our cyclic shortest path ILP using constraints that ensure geometric and global 3D consistency of the matching. Solving the ILP (e.g. using Gurobi [29]) results in a globally optimal 3D shape matching represent as a shortest cyclic path in the product graph.

4. SpiderCurve-based 3D Shape Matching

We aim to find geometrically consistent correspondences between two non-rigidly deformed 3D shapes $\mathcal{X}$ and $\mathcal{Y}$ (with same genus and potentially having boundaries) represented as tuples of vertices and edges $\mathcal{X} = (V^{\mathcal{X}}, E^{\mathcal{X}})$ and $\mathcal{Y} = (V^{\mathcal{Y}}, E^{\mathcal{Y}})$, respectively. Our notation is summarised in Tab. 2. Our main idea is to discretise the 3D shape $\mathcal{X}$ using a long self-intersecting curve $\mathcal{C}$ that traces the 3D surface $\mathcal{X}$ – due to the resemblance to a spider web we call this representation *SpiderCurve*, cf. Fig. 4. With that, we match $\mathcal{C}$ to the 3D shape $\mathcal{Y}$ while ensuring that (i) existing self-intersections of $\mathcal{C}$ are maintained, and that (ii) no additional self-intersections are introduces by the matching. We visualise our overall pipeline in Fig. 3.

In the following subsections we first discuss how to extract our SpiderCurve on $\mathcal{X}$, followed by the presentation of our integer linear programming (ILP) formulation.

4.1. SpiderCurve Extraction

To ensure that our SpiderCurve $\mathcal{C} = (V^{\mathcal{C}}, E^{\mathcal{C}})$ is a faithful discretisation of the 3D shape $V^{\mathcal{X}}$, we impose that $\mathcal{C}$ covers all vertices of $V^{\mathcal{X}}$. Moreover, the start and end points of $\mathcal{C}$ are equal, so that it forms a cycle. We explicitly strive for self-intersections in $\mathcal{C}$ in order to consider geometric consistency in multiple directions along the shape surface (this is achieved by imposing that intersections are maintained by the matching, see Sec. 4.2). We have found that using an approximation algorithm for the (metric) travelling salesperson problem (TSP), i.e. the Christofides algorithm [15], leads to reasonable SpiderCurves for our purposes. We note that for technical reasons it is convenient to duplicate vertices in the SpiderCurve $\mathcal{C}$ at the intersection points (so that $|V^{\mathcal{C}}| \geq |V^{\mathcal{X}}|$). In Fig. 4, we visualise extracted SpiderCurves on different shapes.

Symbol	Description
$\mathcal{X} = (V^{\mathcal{X}}, E^{\mathcal{X}})$	3D surface shape $\mathcal{X}$
$\lvert V^{\mathcal{X}}\rvert, \lvert E^{\mathcal{X}}\rvert$	Number of vertices and edges
$V^{\mathcal{X}} \in \mathbb{R}^{\lvert V^{\mathcal{X}}\rvert\times 3}$	Vertices of shape $\mathcal{X}$
$E^{\mathcal{X}} \in \mathbb{N}^{\lvert E^{\mathcal{X}}\rvert\times 2}$	Edges of shape $\mathcal{X}$
$F^{\mathcal{X}} \in \mathbb{R}^{\lvert V^{\mathcal{X}}\rvert\times 128}$	Features of shape $\mathcal{X}$
$\mathcal{Y} = (V^{\mathcal{Y}}, E^{\mathcal{Y}})$	3D surface shape $\mathcal{Y}$
$\dots$	(analogous to $\mathcal{X}$)
$\mathcal{C} = (V^{\mathcal{C}}, E^{\mathcal{C}})$	Cyclic path on $\mathcal{X}$ (Spider Curve)
$\mathcal{P} = (V^{\mathcal{P}}, E^{\mathcal{P}})$	Product graph
$x \in \{0,1\}^{\lvert E^{\mathcal{P}}\rvert}$	Indicator representation of $E^{\mathcal{P}}$
$c \in \mathbb{R}_{+}^{\lvert E^{\mathcal{P}}\rvert}$	Cost vector
$x^{\star}$	Cyclic shortest path in $\mathcal{P}$
(CYC)	Cyclic path constraints
(CON)	Path continuity constraints
(PES)	Preservation of exist. self-inters. constraints
(ANS)	Avoidance of new intersections constraints

Table 2. Summary of the **notation** used in this paper.

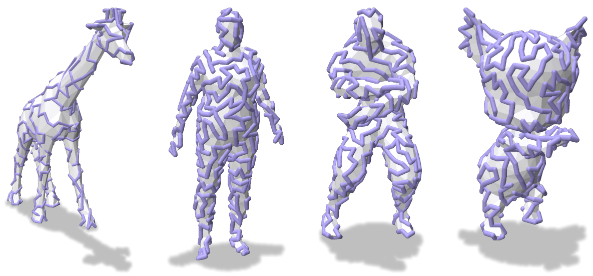


Figure 4. Visualisation of **SpiderCurves** on various types of 3D shapes including humanoid shapes and animals. We can see that our SpiderCurves cover the whole surface and thus pose a faithful discretisation of the respective 3D shape.

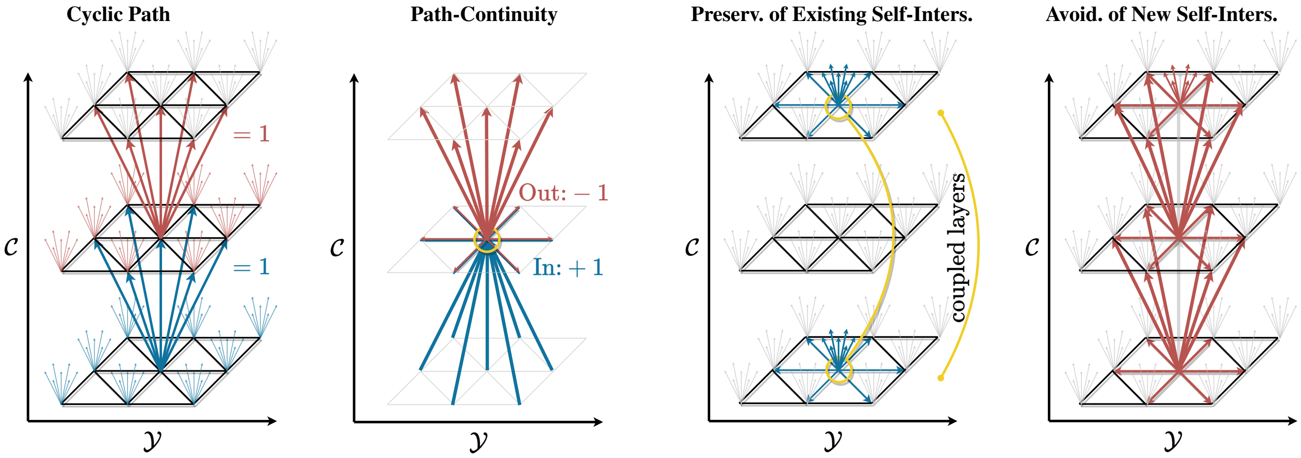


Figure 5. Illustration of our four **constraints**. **(Left)** The sum of all x_i that involve edges connecting two layers (e.g. the set of red or blue edges) should be equal to one, so that the cyclic path $x^\star$ goes through all layers. **(Middle-Left)** The sum of all incoming edges (blue) of a product vertex (yellow) must be equal to the sum of all outgoing edges (red). This ensures that the path is continuous. **(Middle-Right)** Coupling of layers ensures that self-intersections of $\mathcal{C}$ are preserved on $\mathcal{Y}$ under the matching. Coupling can be implemented by ensuring that the sum of outgoing edges (blue) of a product vertex on the bottom layer is equal to the sum of outgoing edges for a product vertex on the top layer (for corresponding vertices across layers, see yellow line). **(Right)** Self-intersections (other than the ones present in $\mathcal{C}$, see Middle-Right) can be avoided by allowing the cyclic path to only visit each vertex on $\mathcal{Y}$ once, i.e. the sum of outgoing product edges (red) must be less than or equal 1 (this does not include vertical inter-layer product edges, i.e. product edges that do not leave the vertex on $\mathcal{Y}$).

4.2. Our SpiderMatch Integer Linear Program

For finding a globally optimal matching, we construct the product graph $\mathcal{P}$ of the SpiderCurve $\mathcal{C}$ (a cyclic chain graph) and the 3D target shape $\mathcal{Y}$. Then, we find a cyclic shortest path in $\mathcal{P}$ in such way that the path goes through each layer of $\mathcal{P}$ (cf. Fig. 2). In addition, we impose additional constraints that ensure geometric consistency in multiple directions along the shape surface.

Contrary to [37, 56] that find shortest paths based on (a variant of) Dijkstra's algorithm, we formulate our optimisation problem as an ILP, so that we can directly integrate additional constraints. Most importantly, we impose that (i) self-intersections in $\mathcal{C}$ are preserved under the matching, and that (ii) additional self-intersections are avoided. To formulate our optimisation problem, we introduce for each edge $e_i \in E^{\mathcal{P}}$ in the product graph $\mathcal{P}$ a binary indicator variable $x_i \in \{0, 1\}$. The value $x_i = 1$ means that product edge e_i is part of our matching path, and $x_i = 0$ means that e_i is not contained in the matching. Our overall objective is to find the optimal matching $x^\star$ in the space of geometrically consistent matchings (to be defined in the following paragraphs) that minimises matching costs. We note that a product edge $e_i = (v_k, v_l) \in E^{\mathcal{P}}$ consists of two product vertices $v_k, v_l \in V^{\mathcal{P}}$, where each product vertex consists of a vertex $v^{\mathcal{C}} \in V^{\mathcal{C}}$ of the cyclic path $\mathcal{C}$ and a vertex $v^{\mathcal{Y}} \in V^{\mathcal{Y}}$ of 3D surface shape $\mathcal{Y}$.

In the following we introduce the constraints of our ILP that define the space of geometrically consistent matchings. We visualise all types of constraints in Fig. 5.

Cyclic Path. We want to ensure that every vertex of $\mathcal{C}$ is matched at least once, i.e. the optimal cyclic path $x^\star$ goes through all layers. This can be achieved by ensuring that the sum over all x_i between any pair of adjacent layers (cf. Fig. 5 left) is equal to one. Formally, this is expressed as

$$\forall\, (v_1^{\mathcal{C}}, v_2^{\mathcal{C}}) \in E^{\mathcal{C}} : \sum_{i: e_i = \big((v_1^{\mathcal{C}}, \bullet), (v_2^{\mathcal{C}}, \bullet)\big) \in E^{\mathcal{P}}} x_i = 1. \tag{CYC}$$

Path Continuity. Furthermore, we have to ensure that every product vertex $v_l \in V^{\mathcal{P}}$ with an active ($x_i = 1$) incoming product edge $e_i = (v_k, v_l)$ also has an active ($x_j = 1$) outgoing product edge $e_j = (v_l, v_m)$, i.e. the path must be continuous (cf. Fig. 5 middle-left). This property can be formalised as

$$\forall\, v_l \in V^{\mathcal{P}} : \sum_{i: e_i = (\bullet, v_l) \in E^{\mathcal{P}}} x_i = \sum_{j: e_j = (v_l, \bullet) \in E^{\mathcal{P}}} x_j. \tag{CON}$$

Preservation of Existing Self-Intersections. To ensure geometric consistency, i.e. neighbourhoods between shape elements[2] are preserved by the matching, every self-intersection of the SpiderCurve $\mathcal{C}$ must result in a self-intersection that involves the same vertices when $\mathcal{C}$ is matched to $\mathcal{Y}$. We can ensure this by incorporating *coupling* constraints between product vertices that involve self-

[2] In our formalism we can interpret polygons that are formed by our SpiderCurve as shape elements.

intersections. To be precise, we couple those product vertices $v_p^{\mathcal{C}}$ and $v_q^{\mathcal{C}}$ on (different) layers p and q that belong to the same vertex $v^{\mathcal{Y}}$ on $\mathcal{Y}$ (cf. Fig. 5 middle-right). Formally, this can be written as

$$\forall\, v_p^{\mathcal{C}} \in V^{\mathcal{C}}, v_q^{\mathcal{C}} \in V^{\mathcal{C}}, v^{\mathcal{Y}} \in V^{\mathcal{Y}},\ p \neq q,\ v_p^{\mathcal{C}} = v_q^{\mathcal{C}} : \\ \sum_{i:e_i=((v_p^{\mathcal{C}},v^{\mathcal{Y}}),\bullet)\in E^{\mathcal{P}}} x_i = \sum_{j:e_j=((v_q^{\mathcal{C}},v^{\mathcal{Y}}),\bullet)\in E^{\mathcal{P}}} x_j. \tag{PES}$$

Avoidance of New Self-Intersections. Lastly, we forbid that new self-intersections are introduced by the matching. This can be ensured by constraining the matching path in such a way that every vertex $v^{\mathcal{Y}} \in V^{\mathcal{Y}}$ can only be visited at most once. In other words, the sum (across all layers) over all outgoing edges of product vertices that contain a given $v^{\mathcal{Y}} \in V^{\mathcal{Y}}$ must be smaller than or equal to one. In this constraint, we do not sum over inter-layer edges whose source and target product vertex contain the same vertex $v^{\mathcal{Y}}$ on $\mathcal{Y}$ (since these edges can be interpreted as 'staying' in the same vertex $v^{\mathcal{Y}}$ on $\mathcal{Y}$). The respective constraints are explained in Fig. 5 right. Mathematically, they read

$$\forall\, v_k^{\mathcal{Y}}, v_l^{\mathcal{Y}} \in V^{\mathcal{Y}}, v_k^{\mathcal{Y}} \neq v_l^{\mathcal{Y}} : \sum_{i:e_i=((v_k^{\mathcal{C}},v_k^{\mathcal{Y}}),(v_l^{\mathcal{C}},v_l^{\mathcal{Y}}))\in E^{\mathcal{P}}} x_i \leq 1 \tag{ANS}$$

with $v_k^{\mathcal{C}}, v_l^{\mathcal{C}} \in V^{\mathcal{C}}$. Note that the sum will not include product vertices on layers which belong to coupling constraints, i.e. whenever $v_k^{\mathcal{C}}$ or $v_l^{\mathcal{C}}$ belong to a self-intersection of $\mathcal{C}$ we do not include respective product edge e_i in the sum.

Matching Costs. We consider a cost vector $c \in \mathbb{R}^{|E^{\mathcal{P}}|}$ based on vertex-wise 3D shape features $F^{\mathcal{X}}$ and $F^{\mathcal{Y}}$. The entry at the i-th position is given by

$$c_i = c(e_i) = \Psi(F_p^{\mathcal{X}} - F_r^{\mathcal{Y}}) + \Psi(F_q^{\mathcal{X}} - F_s^{\mathcal{Y}}), \tag{2}$$

where $e_i = ((v_p^{\mathcal{X}}, v_r^{\mathcal{Y}}), (v_q^{\mathcal{X}}, v_s^{\mathcal{Y}}))$, $v_p^{\mathcal{X}}, v_q^{\mathcal{X}} \in V^{\mathcal{C}}$ and $v_r^{\mathcal{Y}}, v_s^{\mathcal{Y}} \in V^{\mathcal{Y}}$. Furthermore, $\Psi(\cdot)$ is the robust loss function presented in [8].

Our SpiderMatch ILP. Putting our cost function and constraints together yields our final 3D shape matching formalism:

$$\min_{x\in\{0,1\}^{|E^{\mathcal{P}}|}} c^T x \quad \text{s.t (CYC), (CON), (PES), (ANS)}. \tag{3}$$

Essentially, Problem (3) can be seen as a cyclic shortest path problem (over product edges encoded in x, cf. Sec. 4.2) that involves the additional constraints (PES), (ANS).

5. Experiments

In this section we evaluate the performance of our method. First, we explain the considered evaluation metric and the competing methods. Subsequently, we evaluate the non-rigid shape matching performance of our method and compare it to other approaches. We conclude this section with ablation studies.

We solve Problem (3) with the off-the-shelf ILP solver Gurobi [29] (Version 10.0.3) on an Intel Core i9 12900K with 64 GB DDR5 RAM. We decimate all shapes to 1000 triangles using the mesh decimation algorithm provided in [32]. We also compute matchings for methods Cao et al. [14], Ren et al. [53], and Eisenberger et al. [22] on 1000 triangles. Only for Roetzer et al. [55] we decimate shapes to 450 triangles so that runtimes are acceptable (cf. Fig. 1 right). For our robust loss function $\Psi(\cdot)$ [8] we choose parameters $\alpha = 2$ and $c = 0.3$ and instead of a quadratic bowl we use a cubic bowl so that small differences in features are less relevant. We empirically observe that our method can solve *all* instances to *global optimality* in less than 200s.

5.1. Evaluation Metric

We evaluate all shape matching methods using geodesic errors. Geodesic errors allow us to measure the correctness of the resulting matching w.r.t. to a ground-truth matching, i.e. how far is the resulting matching from the ground-truth matching. We follow the Princeton protocol [35] to compute geodesic errors which are normalised by the square-root of the area of a shape (see [35, Sec. 8.2]).

5.2. Methods

We compare our method to four other methods:

Cao et al. [14] is a recent functional map-based deep learning method which achieves state of the art performance by coupling functional maps with point maps during learning. We use this method as our feature extractor and thus consider it as our baseline. Moreover, the extensive evaluation in [14] confirms that it is indeed the state-of-the-art learning-based approach, so that we consider it as a representative for the broad body of learning-based shape matching methods.

Ren et al. [53] is a functional map-based method which associates a functional map with a point wise map as a hard constraint.

Eisenberger et al. [22] is a functional map-based iterative alignment scheme of shapes, which incrementally refines the geometrical information used to perform the matching.

Roetzer et al. [55] implements a combinatorial solver to solve a triangle-matching based ILP presented in [77]. This is the only competing approach that guarantees geometric consistency.

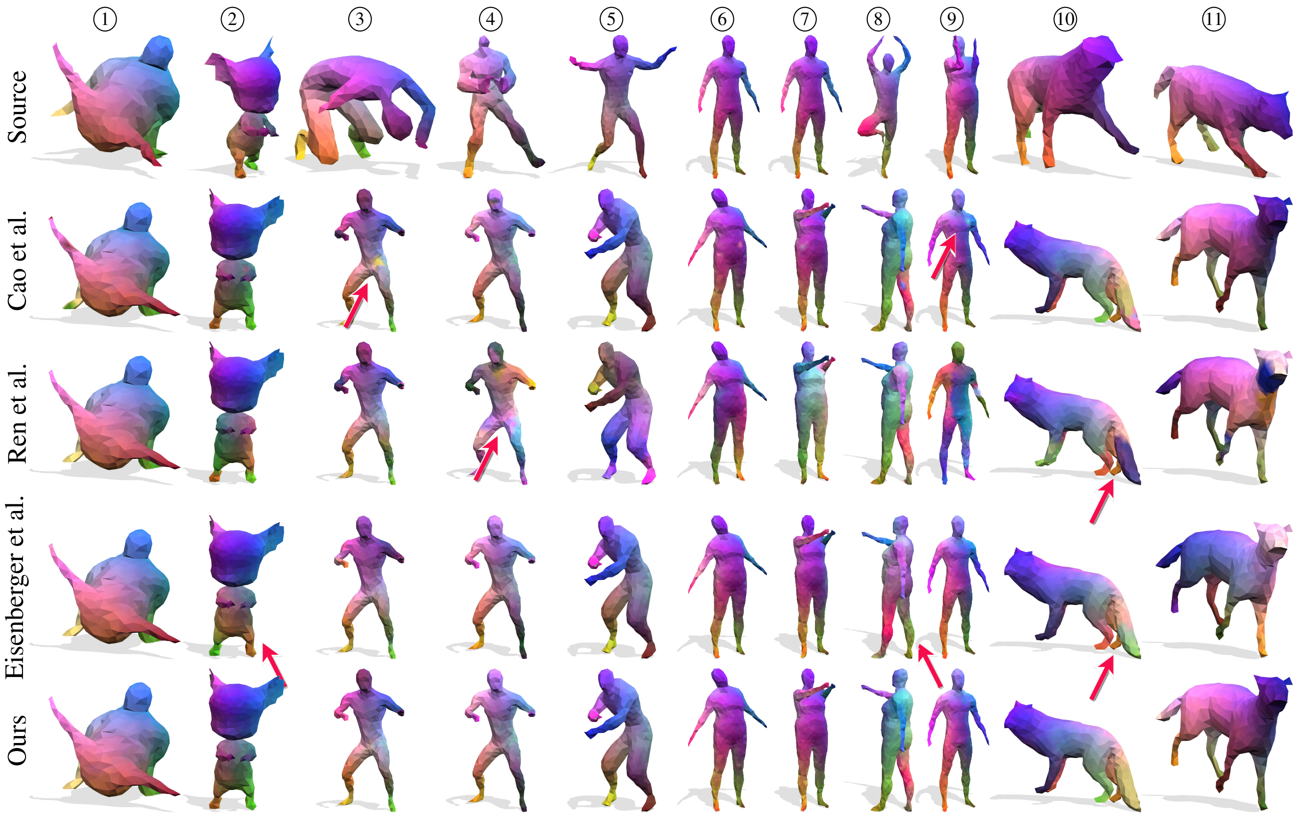


Figure 6. **Qualitative Results** on the datasets FAUST, SMAL, DT4D-H (intra class and inter class). The state-of-the-art learning based method by Cao et al. produces matches with local geometric inconsistencies (cf. columns ③, ⑨). Ren et al. yields visually the worst results due to large mismatches (cf. columns ④, ⑩). Obvious large mismatches are not produced by Eisenberger et al., which however produces left-right flips (cf. columns ②, ⑧) as well as local mismatches (cf. column ⑩). Due to the geometric consistency of our method we produce smooth matchings for all shapes. Note that we show the lower-resolution matchings of Roetzer et al. in the appendix.

5.3. Non-Rigid Shape Matching

In this section we evaluate our method on three different datasets: FAUST [11, 18, 52], SMAL [80], DT4D-H [45] (in intraclass and interclass settings). We show qualitative results in Fig. 6. We can see that our geometrically consistent formalism yields better matchings than Cao et al., since our approach does not produce local mismatches. Furthermore, we observe severe mismatches and geometric inconsistencies in the matchings by Eisenberger et al. and Ren et al.

FAUST. We test on the (more challenging) remeshed version [18, 52] of the FAUST dataset [11]. The dataset consists of 100 near-isometric human shapes (we randomly sample 100 pairs from the test set). In Fig. 7 left, we show geodesic errors on FAUST. Our geometrically consistent approach helps to resolve local missmatches (cf. Fig. 6) which yields improved geodesic error scores compared to Cao et al. The method by Ren et al. results in left-right flips due to intrinsic symmetries of human shapes of FAUST, which increases geodesic errors drastically.

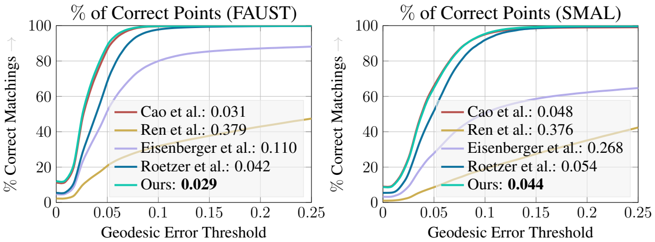


Figure 7. **Geodesic errors** on FAUST and SMAL datasets. Numbers in the legends are mean geodesic errors (↓). Our method yields best results w.r.t. to mean geodesic errors on both datasets.

SMAL. Next, we evaluate on the remeshed version of the SMAL dataset [80]. This dataset consists of 49 non-isometric deformed animal shapes of eight species (we randomly sample 100 pairs from the test set). In Fig. 7 right, we show geodesic errors on SMAL. Again, ours is the best among all methods. For this dataset, shapes are unaligned, which harms the performance of Eisenberger et al. since it has to heuristically find a pre-alignment necessary for the method.

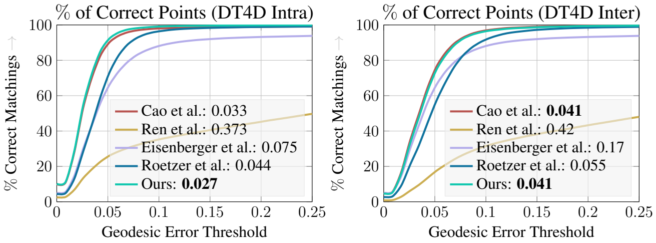


Figure 8. **Geodesic errors** on DT4D-H dataset in the intra-class and inter-class setting. Numbers in the legends are mean geodesic errors (↓). In the intra-class settings our method produces best results among all competitors. In the inter-class setting ours is on par with Cao et al.

DT4D-H. Finally, we test on DT4D-H [45], which consist of 9 different classes of humanoid/game character shapes in different poses. These shapes are taken from DeformingThings 4D [41]. We sample 100 random intra-class pairs, i.e. near-isometric deformed shape pairs. We also sample 100 random inter-class pairs, i.e. non-isometric deformed shape pairs. In Fig. 8, we show geodesic errors for both settings. In the intra-class setting our method outperforms all methods. For inter-class shapes our method is on par with Cao et al. The method by Eisenberger et al. suffers from the unaligned shapes in both settings, while Ren et al. produces a significant amount of left-right flips. In Sec. 8 in the supp. material we provide additional geometric consistency comparisons among the methods.

5.4. Ablations

In Tab. 3, we summarise results of our ablation study. For that, we sample 25 random pairs of the FAUST dataset, downsample them to 500 triangles and compute matchings in the different settings reported in Tab. 3. We can see that constraints (PES), (ANS), and the robust loss function $\Psi(\cdot)$ contribute to the performance of our method. We also replace $F^{\mathcal{X}}$ and $F^{\mathcal{Y}}$ with wave kernel signatures (WKS) features [7] and compare this setting to our method where we extract $F^{\mathcal{X}}$ and $F^{\mathcal{Y}}$ with [14]. In Sec. 11 in the supp. material we study our method under different resolutions.

Method	**Mean Geo. Err.**
Ours w/o (PES), (ANS)	0.036
Ours w/o (ANS)	0.035
Ours w/o (PES)	0.035
Ours with $\Psi = \|\cdot\|_2$	0.036
Ours with WKS features	0.213
Ours	**0.034**

Table 3. **Ablation studies** of our method.

6. Discussion and Limitations

Time complexity. We have introduced the first geometrically consistent and globally optimal method for 3D shape matching. Our problem instances involve up to $2{\cdot}10^6$ binary variables. Nevertheless, we experimentally demonstrate that we can find globally optimal solutions in all considered matching instances within about 100s, which indicates that our search space has a benign structure (likely stemming from the underlying shortest path-based motivation) that can effectively be exploited by off-the-shelf ILP solvers. Still, our proposed formalism belongs to the (generally difficult) class of integer programming problems. We observe that our constraints do not lead to a totally unimodular matrix, so that in general we may have exponential worst-case runtime. We leave a theoretical analysis of the time complexity of our proposed approach, and the question whether there may exist polynomial time algorithms that solves our problem (or related variants) for future work.

SpiderCurve extraction. The focus of this work was on presenting a novel framework that generalises path-based shape matching formalisms for 3D shape matching. Although our TSP-based SpiderCurve extraction leads to promising matchings in all conducted experiments, we believe that there may be potential using alternative curve-based shape discretisations. An in-depth analysis and comparison of different approaches was not our main focus and is an interesting direction for follow-up works.

Feasibility. In theory, our SpiderMatch ILP (3) might be infeasible, which may for example stem from discretisation artefacts in low-resolution settings. Yet, our method is able to operate on (relatively) high resolution shapes so that we expect these cases to be rare (we found all of the 400 instances considered in our experiments to be feasible).

7. Conclusion

We present a fresh view on geometrically consistent 3D shape matching. Our key insight is to discretise one of the shapes using a SpiderCurve, i.e. a long self-intersecting curve that traces the 3D shape surface. This allows to utilise ideas and concepts from shortest path-based formalisms that have so far not been applicable to the matching of 3D shapes. More specifically, our work is motivated by the recent shortest path-based curve to 3D shape matching formalism by Lähner et al. [37], which we augment with additional self-intersection constraints that allows to ensure geometric consistency in multiple directions along the shape surface. Experimentally we showcase the great potential of our novel approach, which on the one hand leads to state-of-the-art shape matching performance. Furthermore, at the same time we achieve both geometric consistency and global optimality – the latter two properties are highly desirable but have so far not been reached by any previous 3D shape matching method. Overall, we believe that our approach may be insightful and may inspire follow-up works in geometrically consistent shape matching and beyond.

References

[1] Noam Aigerman and Yaron Lipman. Orbifold tutte embeddings. *ACM Trans. Graph.*, 34(6):190–1, 2015. 2

[2] Noam Aigerman, Roi Poranne, and Yaron Lipman. Lifted bijections for low distortion surface mappings. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014. 2

[3] Marc Alexa. Merging polyhedral shapes with scattered features. In *Proceedings Shape Modeling International'99. International Conference on Shape Modeling and Applications*, pages 202–210. IEEE, 1999. 2

[4] Arul Asirvatham, Emil Praun, and Hugues Hoppe. Consistent spherical parameterization. In *Computational Science–ICCS 2005: 5th International Conference, Atlanta, GA, USA, May 22-25, 2005. Proceedings, Part II 5*, pages 265–272. Springer, 2005. 2

[5] Souhaib Attaiki and Maks Ovsjanikov. Understanding and improving features learned in deep functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1316–1326, 2023. 2

[6] Souhaib Attaiki, Gautam Pai, and Maks Ovsjanikov. Dpfm: Deep partial functional maps. In *2021 International Conference on 3D Vision (3DV)*, pages 175–185. IEEE, 2021. 2

[7] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. Pose-consistent 3d shape segmentation based on a quantum mechanical feature descriptor. In *Pattern Recognition: 33rd DAGM Symposium, Frankfurt/Main, Germany, August 31–September 2, 2011. Proceedings 33*, pages 122–131. Springer, 2011. 8

[8] Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4331–4339, 2019. 6

[9] Florian Bernard, Christian Theobalt, and Michael Moeller. DS*: Tighter Lifting-Free Convex Relaxations for Quadratic Matching Problems. In *CVPR*, 2018. 3

[10] Florian Bernard, Zeeshan Khan Suri, and Christian Theobalt. Mina: Convex mixed-integer programming for non-rigid shape alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13826–13835, 2020. 2, 3

[11] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *CVPR*, 2014. 7

[12] Dongliang Cao and Florian Bernard. Unsupervised deep multi-shape matching. In *ECCV*, 2022. 2

[13] Dongliang Cao and Florian Bernard. Self-supervised learning for multimodal non-rigid 3d shape matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17735–17744, 2023.

[14] Dongliang Cao, Paul Roetzer, and Florian Bernard. Unsupervised learning of robust spectral shape matching. *ACM Transactions on Graphics (TOG)*, 2023. 1, 2, 6, 8, 3

[15] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. 1976. 4

[16] 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

[17] Bailin Deng, Yuxin Yao, Roberto M Dyke, and Juyong Zhang. A survey of non-rigid 3d registration. In *Computer Graphics Forum*, pages 559–589. Wiley Online Library, 2022. 2

[18] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8592–8601, 2020. 2, 7

[19] Nicolas Donati, Etienne Corman, and Maks Ovsjanikov. Deep orientation-aware functional maps: Tackling symmetry issues in shape matching. In *CVPR*, 2022. 2

[20] 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. 3

[21] Viktoria Ehm, Paul Roetzer, Marvin Eisenberger, Maolin Gao, Florian Bernard, and Daniel Cremers. Geometrically consistent partial shape matching. In *3DV*, 2024. 2, 1, 3

[22] Marvin Eisenberger, Zorah Lahner, and Daniel Cremers. Smooth shells: Multi-scale shape registration with functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12265–12274, 2020. 3, 6, 1

[23] Danielle Ezuz and Mirela Ben-Chen. Deblurring and denoising of maps between shapes. In *Computer Graphics Forum*, pages 165–174. Wiley Online Library, 2017. 2

[24] Danielle Ezuz, Behrend Heeren, Omri Azencot, Martin Rumpf, and Mirela Ben-Chen. Elastic correspondence between triangle meshes. In *Computer Graphics Forum*, pages 121–134. Wiley Online Library, 2019. 2, 1

[25] Pedro F Felzenszwalb. Representation and detection of deformable shapes. *TPAMI*, 27(2):208–220, 2005. 3

[26] Maolin Gao, Paul Roetzer, Marvin Eisenberger, Zorah Lähner, Michael Moeller, Daniel Cremers, and Florian Bernard. Sigma: Scale-invariant global sparse shape matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 645–654, 2023. 2, 3

[27] Andrea Gasparetto, Luca Cosmo, Emanuele Rodola, Michael Bronstein, and Andrea Torsello. Spatial maps: From low rank spectral to sparse spatial functional representations. In *2017 International Conference on 3D Vision (3DV)*, pages 477–485. IEEE, 2017. 3

[28] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *European Conference on Computer Vision*, 2018. 2

[29] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023. 4, 6

[30] Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *CVPR*, 2019. 2

[31] Stefan Haller, Lorenz Feineis, Lisa Hutschenreiter, Florian Bernard, Carsten Rother, Dagmar Kainmüller, Paul Swoboda, and Bogdan Savchynskyy. A comparative study of

graph matching algorithms in computer vision. In *European Conference on Computer Vision*, pages 636–653. Springer, 2022. 3

[32] Alec Jacobson, Daniele Panozzo, et al. libigl: A simple C++ geometry processing library, 2018. https://libigl.github.io/. 6

[33] Puhua Jiang, Mingze Sun, and Ruqi Huang. Neural intrinsic embedding for non-rigid point cloud matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[34] Takashi Kanai, Hiromasa Suzuki, and Fumihiko Kimura. 3d geometric metamorphosis based on harmonic map. In *Proceedings The Fifth Pacific Conference on Computer Graphics and Applications*, pages 97–104. IEEE, 1997. 2

[35] Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. *ACM transactions on graphics (TOG)*, 30(4):1–12, 2011. 6

[36] 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. 3

[37] Zorah Lähner, Emanuele Rodola, Frank R Schmidt, Michael M Bronstein, and Daniel Cremers. Efficient globally optimal 2d-to-3d deformable shape matching. In *CVPR*, 2016. 2, 3, 5, 8

[38] D. Khuê Lê-Huu and Nikos Paragios. Alternating Direction Graph Matching. In *CVPR*, 2017. 3

[39] Lei Li, Nicolas Donati, and Maks Ovsjanikov. Learning multi-resolution functional maps with spectral attention for robust shape matching. *NIPS*, 2022. 2

[40] Qinsong Li, Shengjun Liu, Ling Hu, and Xinru Liu. Shape correspondence using anisotropic chebyshev spectral cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[41] Yang Li, Hikari Takehara, Takafumi Taketomi, Bo Zheng, and Matthias Nießner. 4dcomplete: Non-rigid motion estimation beyond the observable surface. In *ICCV*, 2021. 8

[42] Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *ICCV*, 2017. 2

[43] Nathan Litke, Marc Droske, Martin Rumpf, and Peter Schröder. An image processing approach to surface matching. In *Symposium on Geometry Processing*, pages 207–216, 2005. 2

[44] Marco Livesu. Scalable mesh refinement for canonical polygonal schemas of extremely high genus shapes. *IEEE transactions on visualization and computer graphics*, 27(1): 254–260, 2020. 2

[45] Robin Magnet, Jing Ren, Olga Sorkine-Hornung, and Maks Ovsjanikov. Smooth non-rigid shape matching via effective dirichlet energy optimization. In *International Conference on 3D Vision (3DV)*, 2022. 7, 8, 1

[46] Riccardo Marin, Marie-Julie Rakotosaona, Simone Melzi, and Maks Ovsjanikov. Correspondence learning via linearly-invariant embedding. In *NeurIPS*, 2020. 2

[47] 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, 3

[48] Alexander Naitsat, Yufeng Zhu, and Yehoshua Y Zeevi. Adaptive block coordinate descent for distortion optimization. In *Computer Graphics Forum*, pages 360–376. Wiley Online Library, 2020. 2

[49] 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, 3

[50] Chao Peng and Sabin Timalsena. Fast mapping and morphing for genus-zero meshes with cross spherical parameterization. *Computers & Graphics*, 59:107–118, 2016. 2

[51] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NIPS*, 2017. 2

[52] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (ToG)*, 37:1–16, 2018. 7

[53] Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. Discrete optimization for shape matching. In *Computer Graphics Forum*, pages 81–96. Wiley Online Library, 2021. 3, 6

[54] F Rendl, P Pardalos, and H Wolkowicz. The Quadratic Assignment Problem: A Survey and Recent Developments. In *DIMACS workshop*, 1994. 3

[55] Paul Roetzer, Paul Swoboda, Daniel Cremers, and Florian Bernard. A scalable combinatorial solver for elastic geometrically consistent 3d shape matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 428–438, 2022. 1, 2, 3, 6

[56] Paul Roetzer, Zorah Lähner, and Florian Bernard. Conjugate product graphs for globally optimal 2d-3d shape matching. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3, 5

[57] Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *ICCV*, 2019. 2

[58] David Sankoff and Joseph B Kruskal. *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison*. Addison-Wesley Publishing Company, 1983. 3

[59] Christian Schellewald and Christoph Schnörr. Probabilistic Subgraph Matching Based on Convex Relaxation. In *EMMCVPR*, 2005. 3

[60] Frank R. Schmidt, Eno Töppe, and Daniel Cremers. Efficient planar graph cuts with applications in computer vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009. 3

[61] Patrick Schmidt, Marcel Campen, Janis Born, and Leif Kobbelt. Inter-surface maps via constant-curvature metrics. *ACM Transactions on Graphics (TOG)*, 39(4):119–1, 2020. 2

[62] Patrick Schmidt, Dörte Pieper, and Leif Kobbelt. Surface maps via adaptive triangulations. *Computer Graphics Forum*, 42(2), 2023. 2

[63] Thomas Schoenemann and Daniel Cremers. A combinatorial solution for model-based image segmentation and real-time tracking. *TPAMI*, 32(7):1153–1164, 2009. 3

[64] John Schreiner, Arul Asirvatham, Emil Praun, and Hugues Hoppe. Inter-surface mapping. In *ACM SIGGRAPH 2004 Papers*, pages 870–877. 2004. 2

[65] Abhishek Sharma and Maks Ovsjanikov. Weakly supervised deep functional maps for shape matching. *NIPS*, 2020. 2

[66] Avinash Sharma, Radu Horaud, Jan Cech, and Edmond Boyer. Topologically-robust 3d shape matching based on diffusion geometry and seed growing. In *CVPR 2011*, pages 2481–2488. IEEE, 2011. 2

[67] Nicholas Sharp, Yousuf Soliman, and Keenan Crane. Navigating intrinsic triangulations. *ACM Transactions on Graphics (TOG)*, 38(4):1–16, 2019. 2, 3

[68] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces. *arXiv preprint arXiv:2012.00888*, 2020. 2

[69] Kenshi Takayama. Compatible intrinsic triangulations. *ACM Transactions on Graphics (TOG)*, 41(4):1–12, 2022. 2

[70] Gary KL Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C Langbein, Yonghuai Liu, David Marshall, Ralph R Martin, Xian-Fang Sun, and Paul L Rosin. Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE transactions on visualization and computer graphics*, 19(7):1199–1217, 2012. 2

[71] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *ICCV*, 2019. 2

[72] Giovanni Trappolini, Luca Cosmo, Luca Moschella, Riccardo Marin, Simone Melzi, and Emanuele Rodolà. Shape registration in the time of transformers. *Advances in Neural Information Processing Systems*, 2021. 2

[73] Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. In *Computer graphics forum*, pages 1681–1707. Wiley Online Library, 2011. 2

[74] 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

[75] Matthias Vestner, Roee Litman, Emanuele Rodola, Alex Bronstein, and Daniel Cremers. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3327–3336, 2017. 2

[76] Ruben Wiersma, Elmar Eisemann, and Klaus Hildebrandt. Cnns on surfaces using rotation-equivariant features. *ACM Transactions on Graphics (ToG)*, 39(4):92–1, 2020. 2

[77] Thomas Windheuser, Ulrich Schlickewei, Frank R Schmidt, and Daniel Cremers. Geometrically consistent elastic matching of 3d shapes: A linear programming solution. In *ICCV*, 2011. 2, 3, 6

[78] Thomas Windheuser, Ulrich Schlickwei, Frank R Schimdt, and Daniel Cremers. Large-scale integer linear programming for orientation preserving 3d shape matching. In *Computer Graphics Forum*, pages 1471–1480. Wiley Online Library, 2011. 2, 3

[79] Yang Yang, Wen-Xiang Zhang, Yuan Liu, Ligang Liu, and Xiao-Ming Fu. Error-bounded compatible remeshing. *ACM Transactions on Graphics (TOG)*, 39(4):113–1, 2020. 2

[80] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *CVPR*, 2017. 7