DOI: 10.1111/cgf.70406
EUROGRAPHICS 2026
A. Bönsch and N. Umetani
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 45 (2026), Number 2
STAR – State of The Art Report

Establishing Shape Correspondences: A Survey

A. Heuschling[1] H. Meinhold[1] L. Kobbelt[1]

[1]RWTH Aachen University, Germany

Abstract

Shape correspondence between surfaces in 3D is a central problem in geometry processing, concerned with establishing meaningful relations between surfaces. While all correspondence problems share this goal, specific formulations can differ significantly: Downstream applications require certain properties that correspondences must satisfy, while the type of input data and computational constraints influence the choice of method. In this survey, we provide an overview of different correspondence problems, popular paradigms for generating and refining correspondences, and strategies for evaluating their quality. Further, we discuss topological aspects that are especially important for correspondences between surfaces with higher genus. By offering a structured overview and highlighting open challenges, we aim to support and guide future research in the field.

CCS Concepts

• ***Computing methodologies*** → *Shape analysis;*

1 Introduction

Shape correspondence seeks to establish an intuitive and meaningful relationship between the points of two or more similar input shapes. Here, intuitive and meaningful refer to correspondences that respect both *geometric structure* and *semantic similarity* [LI14]. The intricate interplay between geometry and semantics across multiple surfaces makes shape correspondence a fundamentally challenging problem. Nevertheless, it is a core component of shape analysis, enabling a wide range of downstream applications. The most immediate use case for correspondences is the transfer of information between shapes, such as geometric detail, textures, segmentations, and parameterizations. Correspondences further enable joint shape processing tasks including co-segmentation, cofield synthesis and compatible remeshing. Further applications include shape interpolation and morphing, shape averaging, and statistical shape modeling and analysis.

The diversity of applications imposes varying qualitative and quantitative requirements and causes significant challenges. Existing methods have achieved notable success in computing meaningful correspondences, *e.g.*, through deformation-based optimization or data-driven approaches. Some methods emphasize stricter guarantees, *e.g.*, geometric consistency, bijectivity, and smoothness of the map describing the correspondence between a source and a target surface. Despite substantial efforts by the research community and numerous contributions, critical limitations persist. Current methods often lack robustness, struggle with computational scalability, especially for larger shape collections, and rarely combine semantic meaningfulness with strong guarantees for the quality of the map. Addressing these challenges is essential for broadening the applicability and reliability of shape correspondence techniques.

In this work, we aim to provide a comprehensive overview of the field, covering the full range of shape correspondence formulations. While several surveys have previously addressed the topic [vKZHC11,LI14,Sah20], the rapid pace of progress calls for a new synthesis that consolidates recent advances and offers a structured perspective on the diverse problem settings, paradigms, and open challenges. By organizing and contextualizing the literature, we seek to make the field more accessible, thus supporting continued progress.

We begin by revisiting a unified formalization of shape correspondence (Section 2) and highlight how specific problem instances may differ. We clarify related terminology, including concepts such as geometric consistency and surface homeomorphism. Furthermore, we structure the vast body of literature, identifying key paradigms (Section 3) used to establish and refine correspondences and summarizing recent progress within each of them (Sections 4 to 7). Methods based on more niche paradigms are presented in Section 8. We also cover topological aspects of correspondences between surfaces (Section 9), as well as approaches for evaluation and visualization (Section 10). In addition, we review datasets commonly used for training and benchmarking (Section 11). Finally, we discuss open challenges and outline promising directions for future research (Section 12).

1.1 Scope

To keep this report focused, we restrict our attention to methods that compute correspondences between shapes undergoing general and

potentially large non-rigid deformations. A closely related topic is 3D registration, which typically assumes rigid or near-isometric deformations and may involve incomplete or noisy data, typically in the context of surface acquisition, where an object is scanned from different directions. For a comprehensive overview of that area, we refer the reader to the survey by Deng *et al.* [DYDZ22]. Our scope is further limited to techniques that establish surface-to-surface correspondences. Thus, we do not cover volumetric (interior) mappings. Likewise, curve-to-surface and curve-to-curve correspondences are beyond the scope of this survey. Finally, methods for (partial) symmetry detection are not covered, although some of the discussed techniques may incidentally support such tasks.

2 Problem Setting

Shape correspondence serves as an umbrella term encompassing multiple problem formulations. We begin by providing a unified overview of these problems and discuss key properties that differentiate specific instances with respect to their input and output. We then introduce topological aspects of maps between surfaces, laying the foundation for the methods surveyed later.

2.1 Shape Correspondence

While shape correspondence problems may vary in formulation, assumptions, and objectives, they all operate on a common object of interest: a relation defined between two or more input surfaces $\mathcal{S}_0, ..., \mathcal{S}_n$. For a pair of surfaces $\mathcal{S}_\mathcal{A}$ and $\mathcal{S}_\mathcal{B}$, this relation can be formally defined as

$$\mathcal{R}_{\mathcal{S}_\mathcal{A}\mathcal{S}_\mathcal{B}} = \{(s_i, s_j) \mid s_i \in \mathcal{S}_\mathcal{A},\ s_j \in \mathcal{S}_\mathcal{B},\ \text{s.t. } s_i \text{ corresponds to } s_j\},$$

which associates a point s_i from surface $\mathcal{S}_\mathcal{A}$ with a corresponding point s_j on $\mathcal{S}_\mathcal{B}$.

While this formalism captures all correspondences between surfaces, it is often more natural to interpret a correspondence as a *map* from a source shape $\mathcal{S}_\mathcal{A}$ to a target shape $\mathcal{S}_\mathcal{B}$. This view is more restrictive and inherently directional, as it excludes one-to-many and symmetric correspondence relations.

2.2 Input

Methods have different assumptions about the input data they require, *e.g.*, regarding representation, similarity, and user input.

Input Surfaces. Some methods are capable of operating on point clouds or triangle soups, *i.e.*, collections of disconnected triangles. If a richer representation is necessary, the most prevalent choice is a well-defined 2-manifold surface, typically discretized as a connected triangle mesh.

Beyond representation, the methods vary in their *geometric assumptions* about the relation between the input surfaces. Some approaches are restricted to near-isometric deformations, whereas others can accommodate highly non-isometric cases.

Further, the methods differ in their *topological assumptions* about the input surfaces. Many algorithms are constrained to surfaces with arbitrary but identical genus. Further, some algorithms solely operate on surfaces with specific topologies, such as spherical or disk-like structures, restricting their applicability.

Level of Automation. While fully automatic methods only take the surfaces as input, other methods need additional user input to produce meaningful results. Some methods can optionally incorporate guidance. User input often takes the form of user-defined constraints, such as a sparse set of point correspondences, commonly referred to as *landmarks*. These specify matches between selected surface points, and are treated as hard or soft constraints. Alternatively, the user may provide a dense but imprecise initial map. In data-driven methods, a preferred solution is often indicated by roughly aligning the shapes in ambient space or by prescribing a canonical orientation. Such setups are commonly referred to as *weakly supervised*.

2.3 Output

The properties of a method's output can be categorized in different ways, *e.g.*, dense or sparse, complete or partial, soft or sharp, and pairwise or groupwise. The structure is determined by the characteristics of the input data (in particular, the surface representation), the problem's formulation, and the paradigm used to establish the correspondence. Section 3 surveys existing paradigms in more detail. Here, we take a step back and examine the broader landscape of shape correspondence, outlining the key axes of variation.

Dense vs. Sparse. The task of establishing a *dense* correspondence, which is typically the main objective in shape correspondence research, is commonly referred to as *shape matching*. Many methods work with a discrete representation of the correspondence, where the granularity may differ: some compute *vertex-to-vertex* maps, while others compute *element-to-element* maps. A special case of a discrete correspondence occurs when both surfaces are decomposed into a set of segments, and the correspondence is established between those segments. Some approaches go beyond the discrete nature of these representations and compute dense *point-to-point* maps, assigning for every point on surface $\mathcal{S}_\mathcal{A}$ a corresponding point on surface $\mathcal{S}_\mathcal{B}$. Point-to-point maps are often realized via piecewise-linear functions, starting from a vertex-to-vertex map. For an arbitrary vertex-to-vertex map, the trivial piecewise-linear extension is generally not embedded in the surface. The challenge is to find a *common triangulation* containing the vertices of both surfaces with embedded edges, *e.g.*, overlay meshes, see Section 6.1.

Some methods compute correspondences only for a small subset of surface points. These *sparse* correspondences can often serve as landmarks for subsequent dense matching methods. Although dense correspondence is required for most applications, it is advantageous to compute a sparse correspondence first: Since methods for sparse correspondence operate on a small subset of input vertices, they can often give optimality guarantees. As numerous shape matching methods only find local optima, they benefit from such meaningful initializations.

To better handle the computational complexity of most shape

matching algorithms, a commonly applied technique is *map refinement*. The idea is to start with a sparse correspondence, which is then successively refined until a dense matching is obtained. In many cases, the refinement step is local, effectively decomposing the problem into multiple smaller subproblems. Algorithms using such multiscale approaches include [VLR*17, VLB*17].

Full vs. Partial. Correspondences can be classified based on the *coverage* of the surfaces. *Full* correspondences ensure that the complete shape is covered by the correspondence. *Partial* correspondences, by contrast, cover only a subset of points on one or both surfaces, leaving some regions unmapped. Correspondences can be categorized as *full-to-full, full-to-partial*, and *partial-to-partial*. Algorithms establishing partial correspondences include [SY14b, GTOG16, LRB*16b, APO21, WY23].

Partial correspondences are useful when some features (topological or geometrical) of one surface are absent in the other, *e.g.*, a mug and a glass, where the mug has a handle that the glass lacks, or when the input shapes have defects such as missing parts or holes. In general, partial correspondences are more challenging as they require identifying not only the correspondence itself but also the parts that are missing in one shape compared to the other.

Important to note is that coverage is *independent* of density: a correspondence can cover the whole shape in a sparse fashion, assigning only a few representative points per region.

Sharp vs. Soft. Another way correspondences can be classified concerns the *sharpness* of the correspondence. *Sharp* correspondences assign a point on one surface to exactly one point on the other, while *soft* correspondences allow a single point to correspond to multiple points, *e.g.*, [SY12, SNB*12]. They can encode uncertainty or symmetries by associating probabilities (*e.g.*, [SPKS16]) with potential matches, representing multiple plausible correspondences simultaneously.

Surface Homeomorphisms. Surface homeomorphisms offer a mathematically *rigorous and unambiguous* notion of correspondence between shapes. They are widely regarded as the gold standard due to their beneficial properties, *i.e.*, bijectivity and continuity, which guarantee consistent, invertible, and well-behaved correspondences. For certain applications, *e.g.*, texture or template transfer, these properties are desirable or even crucial for success. In literature, such correspondences are referred to as *inter-surface maps* Φ.

Note that an inter-surface map can only exist between surfaces that share the *same* Euler–Poincaré characteristic, which encodes the genus of the surface, as dictated by the *classification theorem* [GX13].

Geometric Consistency. Since vertex-to-vertex or element-to-element correspondences are not trivially extendable to inter-surface maps, a discrete analogue of homeomorphisms is required to assess the structural quality of the correspondence. Enforcing *geometric consistency* [WSSC11] provides such an analogue.

Conceptually, a geometrically consistent map preserves the local proximity structure of surface elements, ensuring that neighborhoods are mapped to neighborhoods in a coherent manner (see Figure 1). The exact constraints, however, depend on the specific problem formulation, see Section 5.3.

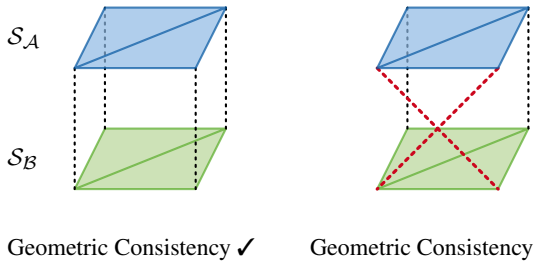

Figure 1: *Geometric consistency provides a discrete analogue for surface homeomorphisms. The map between vertices is indicated by dotted lines. A geometric consistent correspondence preserves the local neighborhood structure. Based on [RAC*24].*

Pairwise vs. Groupwise. The desire to process multiple shapes simultaneously gives rise to correspondences within *shape collections*, see Figure 2, which depicts texture transfer among multiple quadrupeds. Using a straightforward approach, correspondences between every pair of surfaces are required. Given n surfaces and a (possibly non-symmetric) correspondence relation $\mathcal{R}$, up to $n \times (n-1)$ directed correspondence relations, *i.e.*, maps, may be required, leading to quadratic complexity $\mathcal{O}(n^2)$ in the number of surfaces.

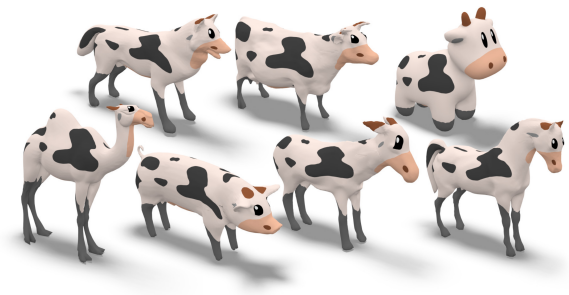

Figure 2: *Shape correspondence for a shape collection. The correspondence is visualized via texture transfer. Taken from [SPK23].*

However, it is sufficient to compute a subset of $n-1$ *invertible* maps between selected surfaces, reducing the complexity to $\mathcal{O}(n)$. To obtain maps between all surfaces in a collection, the maps can be extended to a full set of *cycle-consistent maps* via composition. Maps are cycle-consistent if for any sequence of surfaces $\mathcal{S}_1 \to \mathcal{S}_2 \to \cdots \to \mathcal{S}_k \to \mathcal{S}_1$, it holds

$$\Phi_{\mathcal{S}_k\mathcal{S}_1} \circ \cdots \circ \Phi_{\mathcal{S}_1\mathcal{S}_2} = \mathrm{id}.$$

Strictly speaking, cycle-consistency requires that maps are invertible, making inter-surface maps particularly suitable for such applications. In practice, this requirement is often relaxed and the

composition only needs to approximate the identity under a chosen error metric.

To find appropriate surfaces pairs for the subset of maps, the correspondence graph with the surfaces as nodes has to be considered. An edge in the correspondence graph indicates that a correspondence is computed for the pair of surfaces. In the correspondence graph, there needs to be a path between every surface pair. For $n-1$ maps, this graph must not contain any cycles and thus is a spanning tree. Possible choices for the graph topology are *ring topology* and *star topology*. For the latter case, a reference surface is used, which is often called *template*, and correspondences are computed between the template and every other surface. Note that distortion may accumulate along the composition path, making star topology the preferred choice.

Alternatively, a complete graph [ETLC23] can be established and optimal paths within this graph are obtained while optimizing pairwise correspondences. Cycle-consistency can also be used as regularizer with the goal to establish the full set of maps in such a way that the maps are *consistent* [SY14a], *i.e.*, all possible map compositions between any two shapes yield the same map. Further, shape collections together with an initial set of maps can be refined by replacing outlier maps with higher-quality maps generated through map composition, thereby promoting cycle-consistency [NBCW*11].

2.4 Map Topology

Beyond establishing geometric correspondences, surface homeomorphisms also encode the map's topological structure, identifying topological features, *e.g.*, handles and tunnels, as well as the twisting of the map around them.

Given a pair of surfaces and two homeomorphisms between them, the maps are said to be *homotopic* if one can be continuously deformed into the other while respecting correspondences but without introducing any discontinuities. Maps that are not homotopic belong to different *homotopy classes*, meaning they wrap around the topological features in fundamentally different ways, see Figure 3.

Additionally, fixed landmarks act as topological features by interpreting them as punctures in the surface. Thereby, the number of distinct homotopy classes for maps between the surfaces increases.

3 Paradigm Overview

The problem of determining shape correspondences can be addressed from various perspectives, resulting in different paradigms. This section provides an overview of popular paradigms and presents them in a generic form. Sections 4 to 7 examine each paradigm in more detail, highlighting recent advances. It is worth noting that specific algorithms may draw from multiple paradigms.

3.1 Shape Alignment

A natural approach to shape correspondence is to compute a *deformation* $D_i(\mathcal{S}_i)$ of the input shapes to align them. One can either deform the source shape to align it with the target, or deform both

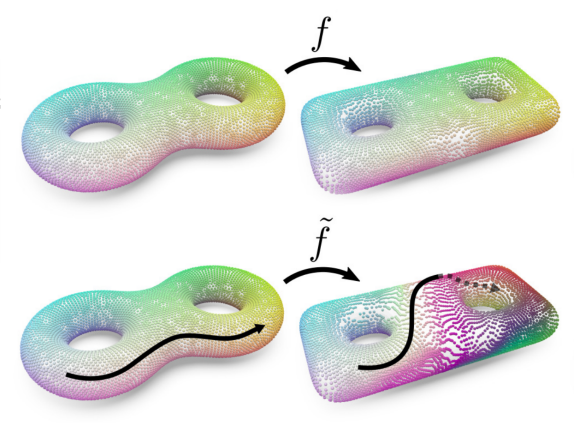

Figure 3: *Two maps between the same pair of surfaces. As the maps f and $\tilde{f}$ wrap around the handles of the surfaces differently, indicated by the black arrow, the maps belong to different homotopy classes. Note that both results are valid, although the top map corresponds to the expected map in most scenarios. Adapted from [BSCK21].*

shapes, often aligning them to a canonical pose or template, or conversely, aligning the template with the inputs. For a pair of surfaces, the correspondence task is typically formulated as an extrinsic optimization problem minimizing

$$E_{\text{align}}(D_{\mathcal{A}}(\mathcal{S}_{\mathcal{A}}), D_{\mathcal{B}}(\mathcal{S}_{\mathcal{B}})) = E_{\text{dist}}(D_{\mathcal{A}}(\mathcal{S}_{\mathcal{A}}), D_{\mathcal{B}}(\mathcal{S}_{\mathcal{B}})) + \omega_{\text{reg}} \cdot E_{\text{reg}}(D_{\mathcal{A}}(\mathcal{S}_{\mathcal{A}}), D_{\mathcal{B}}(\mathcal{S}_{\mathcal{B}})), \tag{1}$$

where $E_{\text{dist}}(D_{\mathcal{A}}(\mathcal{S}_{\mathcal{A}}), D_{\mathcal{B}}(\mathcal{S}_{\mathcal{B}}))$ measures the distance between the two shapes in ambient space and $E_{\text{reg}}(D_{\mathcal{A}}(\mathcal{S}_{\mathcal{A}}), D_{\mathcal{B}}(\mathcal{S}_{\mathcal{B}}))$ is a regularization term, weighted by ω_{reg}, that smoothes the deformation to avoid unrealistic results.

The chosen regularization term implicitly defines the underlying deformation model. Crucially, the deformation model determines the correspondence. The central challenge is to preserve local geometric features of the surfaces while finding a deformation process that yields accurate correspondences. Since such methods are extrinsic, the correspondence remains implicit and must be explicitly extracted after the deformation, typically by assigning each point to its closest point on the other surface.

3.2 Combinatorial Optimization

Shape correspondence can be approached from a *combinatorial* perspective, where the goal is to find a permutation matrix $P \in \Pi_n \subset [0,1]^{n \times n}$ that matches n vertices (or more generally, elements) of two input shapes under a specific cost function. This permutation matrix induces a map $\varphi : \mathbb{N} \to \mathbb{N}$, mapping vertex i of $\mathcal{S}_{\mathcal{A}}$ onto vertex j of $\mathcal{S}_{\mathcal{B}}$.

Such combinatorial problems are commonly cast as *assignment problems* and are well studied. For deeper insights, we refer to [BDM12].

Linear Assignment Problem. A simple formulation is the *linear assignment problem* (LAP) [BDM12]:

$$\min_{P\in\Pi_n} \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}\, p_{ij}, \tag{2}$$

where $C \in \mathbb{R}^{n\times n}$ is a cost matrix, with entry c_{ij} representing the cost of assigning vertex i of shape $\mathcal{S}_\mathcal{A}$ to vertex j of shape $\mathcal{S}_\mathcal{B}$. This cost often encodes a similarity or dissimilarity measure between vertices, see Section 10.

Standard approaches for solving LAPs include the Hungarian method [Kuh55] and the auction algorithm [Ber90].

The LAP assumes that there are n elements on $\mathcal{S}_\mathcal{A}$ that are assigned to n elements on $\mathcal{S}_\mathcal{B}$. As this is not always the case, a generalization of the LAP is sometimes considered, known as *transportation problem* [BJS11].

The continuous extension of the LAP leads to *optimal transport* formulations, where P is no longer restricted to a permutation matrix but can be a stochastic matrix with row and column sums equal to one. Then P can be viewed as a soft map from $\mathcal{S}_\mathcal{A}$ to $\mathcal{S}_\mathcal{B}$.

Quadratic Assignment Problem. The LAP considers only unary costs, *i.e.*, they are pointwise on $\mathcal{S}_\mathcal{A}$ and $\mathcal{S}_\mathcal{B}$. To incorporate pairwise relationships, shape correspondence can instead be formulated as a *quadratic assignment problem* (QAP), written as

$$\begin{aligned}&\min_{P\in\Pi_n} \|D_{\mathcal{S}_\mathcal{A}} - P D_{\mathcal{S}_\mathcal{B}} P^\top\|_F^2 \\ &= \min_{P\in\Pi_n} \sum_{i=1}^{n}\sum_{j=1}^{n} \left(D_{\mathcal{S}_\mathcal{A}}(i,j) - D_{\mathcal{S}_\mathcal{B}}(\varphi(i),\varphi(j))\right)^2,\end{aligned} \tag{3}$$

where $D_{\mathcal{S}_\mathcal{A}}, D_{\mathcal{S}_\mathcal{B}} \in \mathbb{R}^{n\times n}$ encode pairwise relations (*e.g.*, geodesic distances) of the two shapes respectively.

The QAP is a well-studied combinatorial model and is known to be NP-hard [LdAN*07, BDM12]. Practical solutions typically rely on relaxations, heuristics, or approximate solvers to make the problem tractable.

Higher Order Matchings and General Cost Functions. Cost functions involving higher-order interactions lead to higher-order assignment problems, which generalize the classical case. In the past, especially *cubic assignment problems* have been studied [ZWW*10, SHVS11]. For fully *general cost functions*, possibly non-polynomial and non-convex, the problem remains a permutation optimization, but falls outside the standard assignment problem framework. Such problems are typically approached using heuristic methods, often *stochastic*, including genetic algorithms or simulated annealing. They explore the space of possible permutations stochastically, often without guaranteeing a global optimum.

3.3 Distortion-Based Approaches

Instead of an extrinsic perspective, as in deformation-based approaches, shape correspondence can be addressed intrinsically, leading to *distortion-based* methods. Rather than deforming one shape to match the other, this paradigm conceptualizes the correspondence as "sliding" one shape over the surface of the other via a map Φ. The goal is thus to minimize the distortion introduced by Φ from source mesh $\mathcal{S}_\mathcal{A}$ to target $\mathcal{S}_\mathcal{B}$:

$$E_{\text{corres}}(\Phi(\mathcal{S}_\mathcal{A})) = E_{\text{distortion}}(\Phi(\mathcal{S}_\mathcal{A})) + \omega_{reg}(E_{reg}(\Phi(\mathcal{S}_\mathcal{A})). \tag{4}$$

The correspondence is primarily guided by the distortion measure $E_{\text{distortion}}$ (see Section 10 for a review). The regularization term E_{reg}, weighted by ω_{reg}, serves two purposes: It guides the correspondence, *e.g.*, via landmark correspondences, and ensures that the map representation remains *valid*, *e.g.*, bijective.

Since maps are inherently directional, it is common to optimize not only the map $\Phi(\mathcal{S}_\mathcal{A})$ but also, when available, its inverse $\Phi^{-1}(\mathcal{S}_\mathcal{B})$.

3.4 Functional Maps

Instead of matching surfaces directly, spectral methods establish correspondences in the spectral domain. A widely adopted spectral framework for this purpose are *functional maps* [OBS*12], with the goal of establishing correspondences between functions.

In the functional map framework, each surface $\mathcal{S}_i$ is equipped with a (truncated) orthonormal basis $\Phi_k^{\mathcal{S}_i} \in \mathbb{R}^{n_{\mathcal{S}_i}\times k}$, where n_i denotes the number of vertices of the surface. In practice, this truncated basis is typically chosen as the first k eigenvectors of the Laplace–Beltrami operator (LBO).

A functional map between two surfaces $\mathcal{S}_\mathcal{A}$ and $\mathcal{S}_\mathcal{B}$ can then be represented compactly by a matrix $C \in \mathbb{R}^{k\times k}$. This matrix describes how coefficient vectors of functions transform between the two spectral bases. Specifically, if a function on $\mathcal{S}_\mathcal{A}$ is represented in the truncated basis $\Phi_k^{\mathcal{S}_\mathcal{A}}$ by the coefficient vector $\mathbf{a} = (a_1,\dots,a_k)^\top$, then the coefficients of the corresponding function on $\mathcal{S}_\mathcal{B}$ in the truncated basis $\Phi_k^{\mathcal{S}_\mathcal{B}}$ are given by

$$\mathbf{b} = C\mathbf{a}.$$

Since generally $k \ll n_{\mathcal{S}_i}$, optimization and matching in the spectral domain are more efficient than operating directly on the surfaces. However, C encodes correspondences in the spectral domain rather than providing an explicit point-to-point map.

A central challenge in this framework is to constrain and regularize C such that it reflects desired structural properties such as bijectivity and locality. Further, functional map methods relying on the LBO basis assume (near-)isometric deformations as the operator is invariant under such deformations. In (near-)isometric scenarios, the invariance implies a one-to-one correspondence between the spectral eigenvectors. Recent work seeks to overcome this limitation and extend functional maps to non-isometric settings.

Moreover, while some applications (*e.g.*, normal transfer) can operate directly with C, other tasks (*e.g.*, shape morphing) require an *explicit* (vertex-to-vertex) correspondence. In such cases, the functional map must be converted into an explicit map representation aligning with the definition of correspondences in Section 2.1.

4 Advances in Shape Alignment

Recent years have witnessed significant progress in shape alignment. In this section, we review these advances from several complementary perspectives, including template-based versus template-free approaches, deformation representations, hierarchical strategies, and regularization techniques. These categories capture the main underlying design choices and highlight the respective advantages and limitations of different methods. A summary of key approaches and their properties is provided in Table 1.

4.1 Template-Based vs. Template-Free Approaches

Template-based methods use a canonical representation to which all shapes of a class are mapped, thereby inducing a star-shaped correspondence topology. Such templates can take the form of explicit meshes or point clouds [SMRO22], implicit fields [ZYDL21, DYT21], or latent-space templates [JHTG20, YHS*24].

Most commonly, the template is either picked from the shape collection [SMRO22] or it represents some average shape [ZYDL21]. Deng *et al.* [DYT21] learn a fusion of signed distance function (SDF) values across the collection rather than a valid SDF of an average shape. This superposition, combined with a learned correction field, enables the explicit handling of topological differences and regions where simple deformation fails. GenCorres [YHS*24] combines an implicit generator trained using the VAE-loss proposed in SALD [AL21a] with an explicit mesh template, allowing correspondences to be first established in an implicit representation and then refined using an explicit mesh.

Template-free approaches establish correspondences without relying on a canonical template. They either compute deformations directly between shapes [ENK*21, HPG*23] or learn a latent space [JHTG20, ANVL21] in which a latent code represents each shape. Note that ShapeFlow [JHTG20] uses a template for shape matching but not for establishing the latent space.

4.2 Deformation Representation

The representation of shape deformations lies at the core of shape alignment methods. The most straightforward way is to prescribe the final position or a deformation field that produces the deformed shape in one go [HPG*23]. Alternatively, a continuous deformation process [ELC19, JHTG20, ENK*21] can be used to obtain the final deformation, which also considers intermediate shapes. Instead of predicting vertex positions or a continuous deformation directly, [AGK*22] predict the Jacobians of the deformation and reconstruct the deformed surface from the predicted Jacobians by solving a Poisson equation.

More structured representations include handle-based [SHM*22] or cage-based representations [WAK*20], which manipulate shapes via a set of control points, and learned latent embeddings [JHTG20]. Reduced representations are also widely used, realized via basis representations [ELC19], patch-based deformation models [MRWB23], or mesh-free function approximation methods [SMRO22]. Sundararaman *et al.* [SMRO22] show that using a reduced representation proves beneficial in data-driven approaches, as it allows high-quality results to be obtained with fewer training samples.

4.3 Hierarchical Approaches

Deformation-based methods are prone to undesirable local optima. Hierarchical methods address these instabilities. Mesh Draping [HPG*23] gradually maps source mesh vertices using positional encodings of increasing frequency, biasing the optimization towards smooth deformations at first and progressively enabling finer geometric detail.

Merrouche *et al.* [MRWB23] adopt a coarse-to-fine strategy, beginning with patch-level matchings, which are further refined to vertex-level correspondences. This is achieved by a patch-based deformation model, which represents shape deformation as a collection of near-rigid patch transformations, reducing the number of optimization variables. Such hierarchical formulations improve robustness to noise while also reducing the dimensionality of the problem, thereby making unsupervised learning feasible.

4.4 Regularization

To ensure that estimated deformations remain geometrically plausible, regularization is commonly applied. Two complementary forms of regularization are typically employed:

- **Implicit prior**: Neural network parameterizations inherently constrain the deformation space, since the architecture limits the class of admissible transformations. For instance, Mesh Draping [HPG*23] optimizes a fully-connected network that directly predicts vertex positions, acting as an implicit prior on smooth and coherent deformations. Similarly, [ZQZ*21] use a neural network to predict the deformation of the point cloud of $\mathcal{S}_\mathcal{A}$ such that the point cloud of $\mathcal{S}_\mathcal{B}$ is well approximated and vice versa. Eisenberger *et al.* [ELC19] construct deformation fields using a coarse-to-fine basis derived via the Karhunen–Loève expansion, which inherently ensures volume preservation.
- **Explicit prior**: Hand-crafted regularization terms enforce desirable properties either of the deformed surfaces, the deformation field, or the correspondence itself. Well-known regularization terms include ARAP (as-rigid-as-possible) [ENK*21] and ACAP (as-conformal-as-possible) [YHS*24, YSC*23], which penalize distortion introduced by the deformation process. The deformation field itself can be regularized by AAAP (as-affine-as-possible) [YYN*25] and AKAP (as-killing-as-possible) [ANVL21] formulations. Further, promoting cycle consistency [MRWB23] or invertible deformations [JHTG20] encourages bijective correspondences.

The regularization terms are incorporated in frameworks differently. NeuroMorph [ENK*21] integrates correspondence and interpolation in a unified neural framework. The network computes a stochastic assignment between two input shapes and produces a continuous interpolation, expressed as a displacement field, in a single forward pass. Querying the network for intermediate times $t \in [0, 1]$ allows the application of an ARAP loss to ensure meaningful interpolation trajectories.

Methods relying on latent representations can control the construction of the latent space by enforcing that the interpolation of latent codes produces a sequence of interpolated shapes that undergo plausible deformations using regularization terms [JHTG20,

Method	Objective	Regularization	Template-based	Super-vision	Initialization	Partial	Shape Collection
[SMRO22]	GT or CD	volume preservation, ARAP	✓	✓†	✗	✗	✓
Deep Implicit Templates [ZYDL21]	reconstruction loss	smoothness	✓	weakly	✗	✗	✓
ShapeFlow [JHTG20]	CD	invertibility	✓	✗	✗	✗	✓
DIF [DYT21]	normal consistency	smoothness	✓	✗	✗	✗	✓
GenCorres [YHS*24]	VAE loss	cycle-consistency, ARAP, ACAP	✓	✗	✗	✗	✓
[AGK*22]	GT vertex positions	GT Jacobians	✗	✓	✗	✓	✗
NeuroMorph [ENK*21]	ED	ARAP, GD	✗	weakly	✗	✓*	✗
CorrNet3D [ZQZ*21]	reconstruction loss	neighborhood preservation	✗	✗	✗	✗	✗
[ANVL21]	VAE loss	AKAP	✗	✗	✗	✗	✓
Mesh Draping [HPG*23]	CD	angle distortion, triangle quality	✗	✗	landmarks	✓	✗
[MRWB23]	GD, ED	cycle-consistency, ARAP	✗	✗	✗	✗	✗
GenAnalysis [YYN*25]	reconstrucion loss	AAAP	✗	✗	✗	✓*	✓
[ELC19]	feature similarity, ED	implicit volume preservation	✗	-‡	✗	✗	✗

† Presents unsupervised training used for segmentation transfer.
* Also capable of partial-to-partial correspondences.
‡ Non data-driven method.

Table 1: *Summary of recent shape alignment methods. Note that the regularization column only contains the most important regularization terms. We use the following abbreviations: GT: Ground truth, CD: Chamfer distance, ED: Euclidean distance, GD: Geodesic distance, ARAP: As-Rigid-As-Possible, ACAP: As-Conformal-As-Possible, AKAP: As-Killing-As-Possible, AAAP: As-Affine-As-Possible.*

SMRO22, YHS*24, YYN*25]. More precisely, given the latent embedding $Z_{\mathcal{S}_\mathcal{A}}$ and $Z_{\mathcal{S}_\mathcal{B}} \in \mathbb{R}^D$ of two shapes, the linear interpolation between these latent codes produces a sequence of interpolated latent codes $Z_{\mathcal{S}_\mathcal{A}\mathcal{S}_\mathcal{B}}(\alpha) = \alpha Z_{\mathcal{S}_\mathcal{A}} + (1-\alpha) Z_{\mathcal{S}_\mathcal{B}}$. The decoded shapes corresponding to these interpolated latent codes should undergo plausible deformations, which can, for example, be achieved by promoting rigidity.

For implicit surface representations, imposing rigidity constraints directly is challenging. Therefore, in [ANVL21], the network simultaneously estimates an auxiliary piecewise-linear deformation field which is used to regularize the latent space.

5 Advances in Combinatorial Approaches

In this section, we review recent advances in combinatorial approaches, covering relaxations and heuristic approaches. Further, we report on methods ensuring geometric consistency and global optimality. The methods are summarized in Table 2.

5.1 Relaxations

Bending Graphs [SWC*22] introduces a hierarchical learning framework that integrates both local patch-level and global shape-level information. The optimal transport problem is solved via the Sinkhorn algorithm, augmented with mesh connectivity information. Specifically, confidence values are used to propagate high-confidence features to neighboring low-confidence features along the mesh graph. This encourages nearby points to be matched to nearby locations on the target region. The entire pipeline is optimized in an end-to-end manner.

In HiPPI [BTST19], a general framework for matching shape collections with theoretical convergence guarantees is proposed. At its core, the method parametrizes the pairwise matchings using a template guaranteeing cycle-consistency and enabling scalability for shape collections. This design choice leads to a fourth-order polynomial as objective function. The matching is found by using a higher-order projected power iteration method followed by a projection onto the valid set of matchings.

[BGTM20] reformulate the quadratic assignment problem with permutation matrix constraints as an unconstrained problem suit-

Method	Geometrically Consistent	Globally Optimal	Scalability	Shape Collection
Bending Graphs [SWC*22]	promoted via loss	✗	-	✗
HiPPI [BTST19]	✗	✗	few landmarks, many shapes	✓
[BGTM20]	✗	-†	logical qubits: $n = 3$ to 4^*	✗
[RSCB22]	✓	✗	order of magnitude: $\sim 10^2$ faces	✗
Simulated Annealing [HLC20]	✗	✗	$\mathcal{O}(v \log v \sqrt{v})$ v: number of vertices	✗
Q-Match [BLG*21]	✗	-†	logical qubits: $n = 16^*$	✗
CCuantuMM [BTL*23]	✗	-†	logical qubits: $n = 40^*$	✓
[Sah18]	✗	✗	$\mathcal{O}(vn^2)$ v: number of vertices n: number of landmarks	✗
ENIGMA [EEB20]	promoted via loss	✗	order of magnitude: $\sim 10^3$ faces	✗
DiscoMatch [RAC*24]	✓	✓	order of magnitude: $\sim 10^3$ faces (multi-resolution)	✗
[ERE*24]	✓	✓	order of magnitude: $\sim 10^3$ faces (multi-resolution)	✗
SpiderMatch [RB24]	(✓)‡	✓	order of magnitude: $\sim 10^3$ faces	✗
[RB25]	✓	✓	order of magnitude: $\sim 10^3$ faces	✗
MINA [BST20]	promoted via loss	✓	order of magnitude: $\sim 10^2$ control points	✗
SIGMA [GRE*23]	promoted via loss	✓	order of magnitude: $\sim 10^3$ faces	✗

† Measures optimal solution with certain probability (quantum method).
* Designed for quantum hardware.
‡ Depends on choice of cycle.

Table 2: *Summary of recent combinatorial methods. We indicate whether a method enforces geometric consistency via matching constraints ✓, promotes it via a loss term (excluding losses based on geodesic distances), or does not address it explicitly ✗. For global optimality, we mark ✓ if a method can certify a globally optimal solution. Scalability indicates the problem size a method can handle (if available from the paper). The data provides only a rough estimate, as comparisons are difficult when different hardware is used across papers. Finally, we indicate if a method was designed for shape collections ✓ or not ✗.*

able for execution on adiabatic quantum hardware. The key insight is that the permutation matrix constraints can be encoded as QUBO (Quadratic Unconstrained Binary Optimization) problem with soft permutation matrix constraints. It is worth noting that they do not project a relaxed solution to the space of permutation matrices, but that the soft constraints increase the probability to measure a valid solution, *i.e.*, a permutation matrix. They use a quantum annealer to solve the QUBO. As quantum annealers are limited, the authors can confirm the validity of their approach for $n = 3$.

5.2 Heuristics

[RSCB22] propose a combinatorial solver for non-rigid shape matching that can handle partial-to-partial correspondences. Using the geometric consistency formalism introduced by [WSSC11], they relax the resulting linear program with a Lagrange dual formulation and use a new heuristic for primal rounding which respects the geometric consistency constraints.

A common approach for finding a good matching is to start with a permutation matrix P and update the matrix iteratively, for example by applying an *r-cycle*. An r-cycle permutes disjoint indices $i_1, \dots, i_r$. Simulated Annealing [HLC20] is used as an heuristic optimization scheme, where in each iteration, the current solution and a neighboring solution, obtained by applying one 2-cycle, are evaluated in terms of energy defined by the QAP. The neighbor is accepted with a certain *probability* based on the energy change and current temperature (iteration). A generalization of this approach, in quantum computing context, is proposed in Q-Match [BLG*21]. Instead of applying only one 2-cycle per iteration, Benkner *et al.* choose k-many disjoint candidate cycles c_i (Cyclic α-Expansion) and use adiabatic quantum computing to solve the (still NP-hard, but smaller and unconstrained) subproblem of deciding whether to apply c_i or not. This allows them to explore larger solution spaces simultaneously and converge in fewer iterations. This setup was later extended to incorporate cycle-consistency [BTL*23] for the matching of three shapes allowing the computation of cycle-consistent shape correspondence for shape collections by choosing one template shape. Similar to [BGTM20], these quantum methods

rely on quantum annealers and are rather limited regarding problem sizes they can handle.

Alternatively, an appropriate permutation matrix P can be determined using a genetic algorithm. This paradigm was first used by [Sah18] in the context of isometric shape correspondence, aiming to preserve pairwise distances between sparse landmarks. [EEB20] extend this idea to non-isometric matching of landmarks. To evaluate a candidate landmark correspondence, they extend the sparse landmark correspondences to a dense map and score it by the resulting local metric distortion, favoring dense maps with low distortion. This score guides the evolutionary update: In each generation, the best candidates are selected, and crossover and mutation operators are applied to produce the next generation of landmark matchings.

5.3 Geometric Consistency

Combinatorial approaches offer the possibility to include geometric consistency constraints. In [RAC*24], learning-based features are used together with geometric consistency constraints to leverage the state-of-the-art matching performance of learning-based approaches, while also ensuring structurally consistent matches. The resulting integer linear program (ILP) is solved using a massively parallel Lagrange decomposition method. [ERE*24] extend this idea to geometrically consistent matching between partial shapes together with learning-based features. To make the method applicable to high-resolution shapes, the authors propose a coarse-to-fine scheme. [RSCB22] solve an ILP over the space of orientation-preserving diffeomorphisms, a formalism introduced in [WSSC11].

In [RB24], a path-based formalism is used, where a surface is modeled as a long self-intersecting curve. The shape matching problem is then reformulated as finding the shortest path in the product graph of the curve and the graph of the second surface. Global geometric consistency is obtained by additional constraints such as self-intersection constraints.

[RB25] represent the surface of the source shape as a collection of cyclic graphs, where each face is represented by a surface cycle. For each cycle, they build the product graph of that cycle with the target mesh. By coupling the collection of product graphs, they obtain a hyper product graph. Computing a matching in this hyper product graph together with injectivity constraints preserves neighborhoods between surface cycles and thus the matching is geometrically consistent.

5.4 Global Optimality

Instead of matching all elements of one surface to the other, it is possible to match only a sparse set of elements, *e.g.*, vertices, and deduce an approximation of the correspondence for the other vertices via a deformation model. This formulation leads to a mixed integer problem (MIP) used in MINA [BST20] and makes it possible to solve for the global optimum, making the method independent of the initialization. Similarly, SIGMA [GRE*23] use a deformation model to formulate a MIP. Although they only match a sparse set of shapes, the deformation can be approximated by reconstructing the shape from the projected LBO and the sparse correspondences. Their formulation is provably invariant to rigid transformations and global scaling and capable of handling partial shapes.

6 Advances in Distortion-Based Approaches

A class of approaches using an intrinsic representation follows at their core a *topology-first approach*, where they construct an initial valid inter-surface map Φ between the input surfaces. This initial solution resolves the topological degree of freedom. Only afterwards, they focus on optimizing the geometric aspect of the map, aiming for minimal distortion of the map between the surfaces. The focus of these methods is on *correspondence optimization* rather than *correspondence establishment*.

In the following, we will have a look at key aspects that differentiate algorithms in this class of approaches and report on recent advances. We summarize these findings in Table 3.

6.1 Overlay Meshes

A standard approach for representing inter-surface maps Φ in a piecewise linear fashion is to employ *overlay meshes*. Intuitively, an overlay mesh is the common triangulation of both surfaces $\mathcal{S}_\mathcal{A}$ and $\mathcal{S}_\mathcal{B}$ between which the inter-surface map Φ maps. In general, the map cannot be linear in the faces of $\mathcal{S}_\mathcal{A}$, but must also map the edges to piecewise linear curves on the other surface. Thus, the connectivity of $\mathcal{S}_\mathcal{A}$ must be embedded in $\mathcal{S}_\mathcal{B}$, and by subsequently triangulating the resulting polygonal mesh, the overlay mesh can be obtained. The vertex set of this mesh comprises the original vertices of both surfaces as well as additional intersection vertices generated by edge-edge intersections.

Overlay meshes serve as a foundation for several optimization-based inter-surface map approaches [SBCK19, SCBK20]. However, they are associated with significant computational and numerical challenges. First, the complexity of the overlay mesh, *i.e.*, their element count, scales with the product of the input mesh complexities. Consequently, optimization methods operating on overlay meshes tend to be computationally expensive. Further, the geometric quality of these meshes is generally poor: since vertices from $\mathcal{S}_\mathcal{A}$ may be mapped arbitrarily close to vertices or edges of $\mathcal{S}_\mathcal{B}$, the triangulation often contains *sliver triangles* (commonly referred to as caps and needles), which are known to compromise numerical robustness in subsequent computations. Even overlay meshes constructed from high-quality input surfaces, tessellated with equilateral triangles, can still contain arbitrarily ill-conditioned elements.

In [SPK23], this problem is addressed: They compute an inter-surface map represented as a piecewise linear function, while maintaining numerical stability and achieving good runtimes. The core idea is to introduce a third triangulation with high-quality elements that approximates the map. This intermediate triangulation is updated throughout the optimization, while implicitly tracking the true inter-surface map. The explicit overlay computation is deferred until the final map is extracted. Since the map resolution is decoupled from the input surfaces, this method can be used to compute shape correspondences between entire shape collections in a reasonable time. However, the method relies on spherical embeddings for the map representation, which restricts its applicability to genus-zero meshes.

Method	Surface Topology Class	Distortion Measure	Map Representation	Intermediate Domain	Shape Collection
[SAPH04]	all	to intermediate domain	overlay mesh	✓(base complex)	✗
[SBCK19]	disk	end-to-end	overlay mesh	✓(plane)	✗
[SCBK20]	all	end-to-end	overlay mesh	(✗)†	✗
[SPK23]	sphere	end-to-end	intermediate mesh	✓(unit sphere)	✓
[MAKM21]	disk	end-to-end	neural surface map	✓(plane)	✓
[MAKM24]	sphere	end-to-end	neural surface map	✓(plane)	✗
[Tak22]	all	end-to-end	compatible intrinsic triangulation	✗	✗

† Uses constant curvature metrics, and constructs an intermediate domain locally.

Table 3: *Summary of distortion-based methods.*

6.2 With Intermediate Domain

The computation of maps from a surface to a specific domain, often canonical domains such as the plane, is a well-studied topic within the geometry processing community. Thus, it comes to no surprise that many inter-surface map approaches build on this work by reformulating the problem and introducing such a canonical domain as *intermediate domain*. A map Φ from a source surface $\mathcal{S}_\mathcal{A}$ to a target surface $\mathcal{S}_\mathcal{B}$ is then expressed as the composition of two maps, $\Phi := f_{\mathcal{D}\to\mathcal{S}_\mathcal{B}} \circ f_{\mathcal{S}_\mathcal{A}\to\mathcal{D}}$, where the first map $f_{\mathcal{S}_\mathcal{A}\to\mathcal{D}}$ maps from $\mathcal{S}_\mathcal{A}$ to a well-studied intermediate domain $\mathcal{D}$, and the second map $f_{\mathcal{D}\to\mathcal{S}_\mathcal{B}}$ maps from the intermediate domain $\mathcal{D}$ to $\mathcal{S}_\mathcal{B}$.

A popular choice for the intermediate domain is the Euclidean plane [SBCK19, MAKM21, MAKM24]. This preference is largely motivated by its foundational role in surface parameterization. Alternative intermediate domains have also been considered, most notably the sphere [SPK23]. However, the use of a fixed canonical domain introduces inherent challenges due to topological constraints. For example, for a planar domain, the Uniformization Theorem states that only surfaces with disk topology are homeomorphic to it. As a result, applying such methods to closed surfaces is non-trivial and typically requires the introduction of cuts and transition functions to manage the topology. Apart from canonical domains, it is also possible to compute a base complex suited for both shapes and use this as an intermediate domain [SAPH04].

In the spirit of surface parameterization, older methods [SAPH04] aim to minimize distortion between the input surfaces and the intermediate domain. However, this strategy generally does not produce high-quality inter-surface maps with low *end-to-end* distortion. In fact, the distortion to the intermediate domain provides limited insight into the distortion of the composite map Φ, as the distortion introduced by the map to the intermediate domain can be canceled as well as amplified by the second map to the target surface.

To address this limitation, recent methods [SBCK19, SCBK20, MAKM21, Tak22, SPK23] explicitly optimize the end-to-end distortion of the full inter-surface map Φ. In these approaches, the intermediate domain is merely used to express the degrees of freedom of the map, while distortion is evaluated directly between the source surface and its image induced by Φ.

In [MAKM21], *overfitted* neural surface maps $\phi : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^3$ are used to represent the parametrization of a surface with disk-topology. An inter-surface map can be directly obtained by map composition. Since these neural surface maps represent the input surfaces, they should not be changed. Thus, to optimize the map between the surfaces, a third neural surface map $h : \Omega \to \Omega$ is introduced and optimized. The approach is easily extendable to shape collections by introducing multiple intermediate maps h_i. [MAKM24] extend this framework to genus-zero surfaces and incorporate semantic features. To accommodate for semantic features, they do not require the map to have low distortion, but instead require the Jacobian of the map to change slowly.

6.3 Without Intermediate Domain

Few methods avoid relying on an intermediate domain for their map representation. In [SCBK20], the vertices of one mesh are directly mapped to the other surface. The image of the edges is defined as a geodesic under a constant curvature metric. In this representation, the surface is locally embedded on a model manifold with constant curvature, but no global embedding is needed. Although different metrics are needed for genus 0, 1, and ≥ 2 surfaces, their approach still yields a unified formulation across all topologies.

The method of Takayama [Tak22] *entirely* avoids an intermediate domain. Instead, it leverages intrinsic triangulations combined with a heuristic edge-flipping strategy to construct two compatible triangulations, which are then optimized for low distortion.

7 Advances in the Functional Map Framework

We begin by giving an overview of the most important aspects of the functional map framework. Afterwards, we present recent advances in this framework, including both axiomatic and learning-based approaches.

7.1 Functional Map Framework

The functional map framework provides a compact representation for correspondences and has become very popular, with a large body of literature covering various aspects. We refer the reader to [OBS*12, OCB*16] for more details, while offering a brief overview here for completeness and clarity.

Recall that $\mathcal{S}_\mathcal{A}$ is equipped with a set of basis functions $\Phi^{\mathcal{S}_\mathcal{A}} = \{\phi_0^{\mathcal{S}_\mathcal{A}}, \dots, \phi_n^{\mathcal{S}_\mathcal{A}}\}$ and likewise $\mathcal{S}_\mathcal{B}$ is equipped with $\Phi^{\mathcal{S}_\mathcal{B}}$. A function f on $\mathcal{S}_\mathcal{A}$ can be expressed in terms of the basis as a linear combination: $f = \sum_i a_i \phi_i^{\mathcal{S}_\mathcal{A}}$, with scalar coefficients a_i. Therefore, f can be represented in the basis $\Phi^{\mathcal{S}_\mathcal{A}}$ as $\mathbf{a} = (a_0, \dots, a_n)^\top$. Further, assume $\mathbf{b}$ is the representation of the *corresponding* function g on $\mathcal{S}_\mathcal{B}$ in the basis $\Phi^{\mathcal{S}_\mathcal{B}}$. Then, for some $\{c_{ij}\}$, it holds:

$$g = \sum_j \sum_i a_i c_{ij} \phi_j^{\mathcal{S}_\mathcal{B}}. \tag{5}$$

The matrix C with entries $\{c_{ij}\}$ is the matrix representation of the functional map and it holds $\mathbf{b} = C\mathbf{a}$.

The representation is flexible since the basis can be chosen freely. However, using an orthonormal basis simplifies computations. The representation can further be compactified, and thus made computationally more efficient, by working with a reduced set of k basis elements. For simplicity, usually the same k is chosen for both input surfaces. An important property of the basis should therefore be its *compactness*: many natural functions on a shape should be well approximated using a small number of basis elements. Further, the basis should be *stable*, meaning the basis functions on both surfaces should not differ drastically, and the function space spanned by the reduced set of eigenvectors should be nearly identical.

The most prevalent choice for the basis are the eigenvectors of the Laplace–Beltrami Operator (LBO). Analogous in spirit to the Fourier transform, the LBO eigenvectors oscillate more if they correspond to larger eigenvalues, see Figure 4. Truncating eigenvectors with large corresponding eigenvalues removes high frequencies from a function, which can be regarded as a low-pass filter operating on the mapped functions. The eigenvectors are stable up to sign flip for near-isometries. In such scenarios, C is nearly diagonal and sparse, see Figure 5 [OBS*12]. A more in depth overview on spectral bases can be found in [CP21].

Figure 4: *The first eight eigenvectors of the LBO on two near-isometric shapes. They are invariant up to a sign flip. Further, the eigenvectors corresponding to larger eigenvalues (right) oscillate more than the eigenvectors of smaller eigenvalues (left). Taken from [BRGK23].*

Using the matrix representation, shape matching can be formulated as a system of linear equations with linear constraints. A simple approach is to solve [OCB*16]:

$$C = \arg\min_X ||XA - B||^2 + \alpha E_{\text{reg}}(X), \tag{6}$$

where A and B are *function preservation* constraints represented in a chosen basis, and α is the weighting factor for the map regularization term E_{reg}, *e.g.*, promoting commutativity with the LBO.

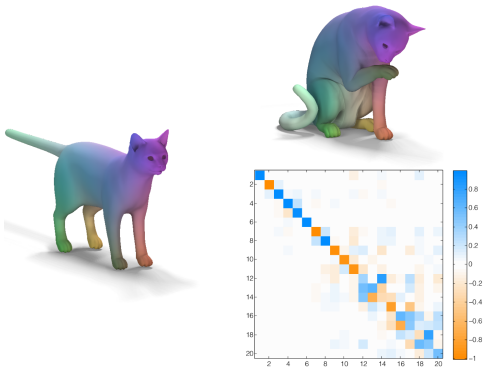

Figure 5: *Depiction of the magnitude of the matrix entries of C using the LBO basis. For a near-isometric correspondence, C is close to diagonal. Taken from [OBS*12].*

For function preservation constraints, a popular choice are descriptor functions corresponding to surface properties, such as the Gaussian curvature, or shape signature functions (see Section 10: Feature Matching).

In the following, we discuss approaches that explore various aspects of the functional map framework and its integration in machine learning methods.

7.2 Basis Functions

While the eigenvectors of the LBO are a popular choice for basis vectors, other operators have been proposed. For a summary of the methods presented in this part, see Table 4.

Using the LBO basis breaks down for partial meshes, so [KMP*21] address this problem by using Mexican hat wavelets computed by approximating the derivative of the heat kernel. This family of functions is mutli-scale with local support, enabling matching between partial meshes. The family of functions is located at a set of sample points, so they rely on an initial set of sparse correspondences.

If landmarks are incorporated via constraints using the LBO basis, they are only promoted but typically not guaranteed. To incorporate landmark constraints strictly, Panine *et al.* [PKO22] present a landmark-adapted basis using an intrinsic Dirichlet-Steklov eigenproblem. They aim for near-conformal shape correspondences that preserve landmarks exactly by representing landmarks as one-dimensional boundaries. The Dirichlet-Steklov eigenfunctions resonate near those boundaries. Using this basis, landmark preservation is achieved by only allowing a certain structure for the functional map matrix C, *i.e.*, C must be block-diagonal.

Since the truncated LBO basis acts as a low-pass filter, functional

Method	Descriptor Function	Basis	Similarity	Initialization	Partiality	Supervision
[KMP*21]	✗	Mexican hat wavelets	non-isometric	landmarks	✓	-
[PKO22]	✗	landmark-adapted	non-isometric	landmarks	✗	-
[AL21b]	WKS	data-driven using POD modes	non-isometric	landmarks	✗	-
[CBM22]	WKS	PCA of dictionary of sparse Gaussians	non-isometric	landmarks	✗	-
[HSA*23]	WKS	elastic	non-isometric	landmarks	✗	-
[BXNL24]	DiffusionNet features WKS descriptors	LBO + elastic	non-isometric	✗	✗	-
[MRMO20]	learned	high-dimensional linear embedding	non-isometric	✗	✓	✓
[SL23]	learned	high-dimensional linear embedding	non-isometric	✗	✗	✗
[YW25]	HKS	weighted LBO	near-isometric	✗	✗	✓
[JSH23]	learned	high-dimensional neural intrinsic embedding	near-isometric	✗	✓	weakly

Table 4: *Summary of recent methods with new basis functions for the functional map framework. In the supervision column, - indicates that a method is not learning-based. We use the following abbreviations: WKS: Wave Kernel Signature, HKS: Heat Kernel Signature, POD: Proper Orthogonal Decomposition, PCA: principal component analysis.*

maps computed with that basis often struggle aligning high frequency features. Therefore, [AL21b] compute the basis alongside with the functional map C. They use Proper Orthogonal Decomposition modes for dimensionality reduction, making it possible to capture high frequency data. The optimization problem is solved using Alternating Direction Method of Multipliers. Using such an adaptive basis allows to better exploit non-smooth descriptors.

The truncated LBO basis does not cover the surface uniformly, resulting in functions being approximated with varying accuracy across different regions. As a result, the quality of the correspondence may also vary over the surface. Therefore, [CBM22] propose to compute a basis from the principal component analysis (PCA) of a dictionary of sparse Gaussian functions defined on the surfaces. This basis covers the surface more evenly. In [CBM23], a generalization of this approach is introduced making it applicable to other dictionaries, *e.g.*, based on adaptive Gaussians or the WKS.

Hartwig *et al.* [HSA*23] propose a basis based on the Hessian of the elastic thin shell energy. The eigenvectors of this resonate near creases, making the basis crease aware. Since this basis is not orthogonal, they also describe how to adapt the functional map framework to enable the use of such a non-orthogonal basis. Bastian *et al.* [BXNL24] propose a crease-aware basis using a hybrid approach. They incorporate intrinsic information with the LBO as well as extrinsic information with the Hessian of the elastic thin shell energy. They use their basis, among others, within the Deep Geometric Functional Maps method [DSO20].

[MRMO20] extend and generalize the functional map framework for point cloud data and propose a fully differentiable pipeline for correspondence establishment. As such, both the basis and the descriptors are learned from data. Using the basis and the descriptor functions, the two bases are brought into correspondence by calculating the linear transformation between them, and the correspondence is estimated using nearest-neighbor search. The learning process is supervised, as it requires data with annotated ground truth. In [SL23], this method is improved by introducing a new training scheme. By using random sampling during the training, the resulting network is more robust against sampling inconsistencies. It also allows for unsupervised training via an unsupervised loss, aiming to preserve pairwise distances and the orthonormality of the basis as well as of the functional map matrix C.

In [YW25], a weighing of the LBO is learned. More precisely, three different networks are used to predict the stiffness and mass matrix used in the computation of the LBO: A network for predicting the Riemannian Metric, *i.e.* the edge weights, a network for the Voronoi cell weighting and a network for the anisotropy direction and factor.

In [JSH23], the authors propose a neural intrinsic embedding for point clouds as a replacement for the LBO basis. The embedding is trained to approximate geodesic distances between points, meaning geodesic supervision is required during training but not at inference time. Descriptors are obtained using a neural intrinsic mapping network that is trained using a cycle-consistency and isometry loss.

7.3 Vertexwise Map Recovery

For many applications, the functional map representation is insufficient as a vertex-to-vertex map is required. Therefore, the vertex-to-vertex map must be extracted from the functional map representation C.

To obtain a vertex-to-vertex map, one needs to identify for every vertex v of $\mathcal{S}_\mathcal{A}$ the corresponding vertex w of $\mathcal{S}_\mathcal{B}$ given the map C. A straightforward approach would be to use an indicator function $\chi_v(x)$, which equals 1 for vertex v and 0 otherwise, which is transported via C to $\mathcal{S}_\mathcal{B}$. The corresponding vertex of v is the maximum

of the image on $\mathcal{S}_\mathcal{B}$. The computation is in $O(n_{\mathcal{S}_\mathcal{A}} n_{\mathcal{S}_\mathcal{B}})$, as for every vertex of $\mathcal{S}_\mathcal{A}$ an indicator function needs to be transported and the maximum must be found among all $n_{\mathcal{S}_\mathcal{B}}$ vertices of $\mathcal{S}_\mathcal{B}$, making this approach inefficient [OBS*12].

An alternative is proposed in [OBS*12]. The goal is to recover a vertex-to-vertex map $T : V_{\mathcal{S}_\mathcal{B}} \to V_{\mathcal{S}_\mathcal{A}}$, mapping the vertices $V_{\mathcal{S}_\mathcal{B}}$ to the vertices of $V_{\mathcal{S}_\mathcal{A}}$. Note that T must not be bijective. This map can be extracted using nearest neighbor search in the k-dimensional embedding space:

$$T(w) = \arg\min_{v \in V_{\mathcal{S}_\mathcal{A}}} \left\| C(\Phi^{\mathcal{S}_\mathcal{A}}(v))^\top - (\Phi^{\mathcal{S}_\mathcal{B}}(w))^\top \right\|_2, \quad \forall w \in V_{\mathcal{S}_\mathcal{B}}. \tag{7}$$

Alternatives and modifications to this approach include that of [PRM*21], where the explicit map is extracted with an optimal transport approach. The method is faster and more memory efficient than previous approaches. The conversion procedure to a vertex-map is modified by introducing the adjoint map. Thereby, the existence of an inverse is assumed, which leads to improvement in bijectivity of the correspondence. Xia *et al.* [XLGM24] introduce locality preserving refinement: They compute inliers of the initial noisy vertexwise correspondence. Using this, the correspondence is refined aiming for local consistency. Hartwig *et al.* [HSA*23] propose a vertexwise map recovery method that works with non-orthonormal bases.

7.4 Refinement

A common approach within the functional map framework is the refinement of the map C, where the resolution, determined by the size of the basis, of a given functional map is increased, yielding a map of higher quality.

ZoomOut. The most prominent example for map refinement is *ZoomOut* [MRR*19]. An initial functional map with low resolution or noise is iteratively refined to a high-quality high-resolution map, as depicted in Figure 6. To this end, a k-dimensional functional map is converted to a vertexwise map, using Equation (7).

From this vertexwise map, a matrix representation $P \in \mathbb{R}^{n_{\mathcal{S}_\mathcal{B}} \times n_{\mathcal{S}_\mathcal{A}}}$ can be derived, effectively mapping vertices from $\mathcal{S}_\mathcal{A}$ to $\mathcal{S}_\mathcal{B}$. A $k+1$-dimensional map is obtained by converting the matrix representation P back to a functional map representation C with one additional eigenvector added to each basis. The conversion is computed using

$$C_{k+1} = (\Phi^{\mathcal{S}_\mathcal{B}}_{k+1})^\top A^{\mathcal{S}_\mathcal{B}} P \, \Phi^{\mathcal{S}_\mathcal{A}}_{k+1}, \tag{8}$$

where $\Phi^{\mathcal{S}_i}_{k+1}$ are the the basis vectors of increased resolution, and $A^{\mathcal{S}_\mathcal{B}}$ is the lumped area element matrix. (Note that, to remain consistent with our definition of C mapping functions from $\mathcal{S}_\mathcal{A}$ to $\mathcal{S}_\mathcal{B}$, we define T and differently from the formulation in [MRR*19].) This two-step iterative procedure ensures that each functional map C_k is induced by a vertex-to-vertex map T, while enforcing orthonormality on every principal submatrix rather than on a fixed-size functional map.

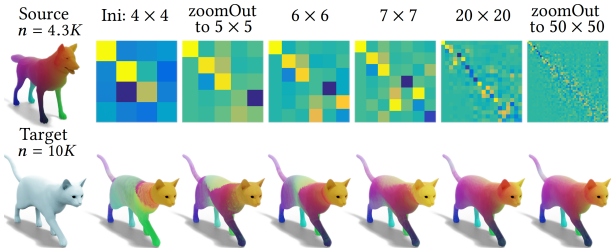

Figure 6: *ZoomOut iteratively refines an initial (low-resolution) map. Taken from [MRR*19].*

Numerous methods have adapted ZoomOut. [HRWO20] propose *consistent ZoomOut*, which promotes the consistency of the refined correspondences for map collections. *Adjoint Bijective ZoomOut* [VM23] is adapted for point cloud matching and modifies ZoomOut such that is uses the adjoint based conversion procedure used in [PRM*21]. As a result, bijectivity is promoted. This work is extended in [VM24], improving the performance and incorporating it in a novel training strategy. [VOM25] propose Neural Adjoint Maps, where they represent the action of the adjoint operator by a neural network. Therefore, the adjoint operator contains non-linear components. Using this operator, they design an iterative refinement algorithm called *Neural ZoomOut*, which produces maps with higher detail due to the non-linearity of the adjoint operator. A differentiable version of ZoomOut is proposed in [MO24]. [HSA*23] adapt ZoomOut to their setting with a non-orthonormal basis, leading to better results for non-isometries.

Maptree [RMOW20] uses an adapted version of ZoomOut with bijectivity constraints to refine multiple initial solutions, leading to different high-resolution maps corresponding to different local minima. In particular, this approach allows the exploration of the self-symmetric space of a shape and can be used for symmetry detection.

Others. Other refinement methods for the functional map framework include [LWH*22], which do not recover the vertexwise map by nearest neighbor search but instead formulate it as an optimization problem which is solved using the fast Sinkhorn filters used in [PRM*21]. [LLH*25] make use of learnable spectral filters to preserve important frequency information. They propose a refinement strategy that leverages the learned spectral filters to enhance the accuracy of the vertex-to-vertex map.

7.5 Coarsening for Spectral Methods

Spectral methods, including but not restricted to functional map methods, rely on constructing linear operators whose size scales with the number of elements in the computational domain, *i.e.*, the mesh. To reduce the computational cost of these methods, two strategies have been explored. One approach is to directly compute a coarsened version of the spectral operator while preserving its key spectral properties [LJO19, CLJL20]. In particular, Chen *et al.* [CLJL20] leverage chordal decomposition to reformulate the

coarsening problem as a convex optimization problem, enabling scalable and robust operator reduction.

An alternative strategy is to reduce complexity at the geometric level by simplifying the mesh itself. While traditional mesh simplification methods primarily aim to preserve geometric fidelity and to produce visually plausible results, they typically do not preserve the spectral properties of the underlying operators, which limits their applicability to spectral shape analysis and matching. To address this limitation, recent works explicitly aim to peserve the spectral structure during mesh simplification. [LLT*20] propose a coarsening scheme designed to maintain spectral consistency by maintaining eigenvectors, while [YS24] introduce a partition-based coarsening approach that enables parallelization and operates on eigenvalues, making the coarsening more scalable.

7.6 Axiomatic Methods

This section gives an overview of recent axiomatic methods to compute functional maps. Such methods work with the functional map representation but use different or improved approaches for computing the correspondence represented by the functional map C.

Smooth Shells [ELC20] computes correspondences in a hierarchal manner, where first coarse approximations of the shapes are aligned, followed by refinement, where higher-frequency details are added back to the shapes. The approximations are computed as spectral reconstructions, where higher frequencies are added with each iteration. Since such reconstructions may change drastically from one level of detail to the next, a shell operator is introduced to reasonably bound the change between iterations. The authors combine intrinsic and extrinsic alignment by embedding the shapes into a higher-dimensional intrinsic-extrinsic space. There, they optimize the functional map and the vertex-map for each hierarchy level. This method has also been adapted to Deep Shells [ETLC20], a learning-based approach. Instead of a strict permutation matrix, a soft matrix is used for differentiability. The used loss computes alignment tightness based on entropy regularized optimal transport.

[RMWO21] introduce a discrete optimization framework for functional maps. It is suitable for several functional map based energies such as descriptor preservation. The functional map representation and the vertexwise map are coupled. In an iterative scheme, the vertexwise map is updated while being informed about the functional map C, then C is updated based on the new vertexwise map. The restriction of the optimization space to valid vertexwise maps leads to higher-quality correspondences as the vertexwise map preserves the properties of the functional map. Magnet *et al.* [MRSO22] extend this framework by including the Dirichlet energy to promote smooth correspondences.

Complex functional maps [DCMO22] extend functional maps to tangent vector fields on surfaces, which allows for robust and efficient transport of tangent vector fields. A complex-linear operator can be used for regularization, leading to orientation-preserving maps. The approach is improved by introducing additional operator preservation constraints [LGL*24].

Functional maps are used for shape collections, targeting cycle-consistency using an artificial template shape [GLT*21]. In [CB20], the authors find correspondences between two shape collections assuming the intra-collection maps are provided. They first compute the most promising inter-collection pair, and then compute the correspondence for the chosen pair. From this, every pairwise map can be derived.

Several methods propose a scalable version of the functional map pipeline. The large computational cost typically comes from the pre-processing costs associated with the computation of the eigenfunctions of the LBO and the manipulation of the vertexwise map, *e.g.*, in ZoomOut. [MO23] therefore propose to work with a sparse set of samples and compute an approximation of the LBO while controlling the error. [MBRM24] introduce ReMatching to tackle high-resolution inputs. In this approach, the functional map is computed on low-resolution meshes, obtained via remeshing. Afterwards, the resulting functional map is transported to the high-resolution inputs.

Constraints. Several new constraints have been proposed, which can be included in functional map pipelines. [BHK20] use the eigenvectors of the scale-invariant LBO [AKR13]. Those are then aligned using a band-limited orthogonal transformation to get rid of inconsistencies, such as sign ambiguities, and used as descriptor functions. [YZZG24] improve upon the biharmonic kernel signature by including the Gaussian curvature, which is isometry invariant. As a result, the descriptor is more discriminative in the presence of noise and invariant to scaling and isometric transformations. [MO21] propose a local pointwise descriptor that reflects deformation around a point within a collection. It is computed based on shape difference operators, which capture the difference between shapes.

Instead of using many high-dimensional descriptor functions, [HLLL21] require multiscale spectral manifold wavelets to be preserved by the functional map. These preservation constraints allow to extract more information and promote the isometry of the map.

7.7 Learning-Based Methods

The popular deep functional map framework was initially proposed by Litany *et al.* [LRR*17] and has since inspired many works, summarized in Table 5.

A typical architecture starts either with a feature transformation layer, which modifies hand-crafted descriptors, or with a feature extraction layer, which extracts the features from scratch. The resulting features, computed for both shapes, typically in a Siamese fashion, are projected onto the LBO basis and used as preservation constraints to compute the functional map using a differentiable, parameter-free functional map layer. Additional regularization imposes desired structural properties on the resulting functional map.

There are supervised, weakly-supervised, and unsupervised approaches. Supervised methods leverage additional training information but require many annotated correspondences, ranging from sparse to dense. Weakly-supervised methods need less annotation, typically assuming roughly rigid alignment between inputs. Unsupervised or self-supervised methods require no annotations and instead rely on regularization terms that enforce desired properties of the resulting functional map.

Method	Supervision	Similarity	Partiality	Input Type
[DSO20, LPCH24]	supervised	near-isometric	✗	triangle meshes
[LRR*17]	supervised	near-isometric	✓	triangle meshes
[BRGK23]	supervised	near-isometric	✓*	triangle meshes
[MRMO20]	supervised	non-isometric	✓	point clouds
[ZLG25, PLDO25]	supervised	non-isometric	✗	triangle meshes
[XER*25]	supervised	non-isometric	✓*	triangle meshes
[SO20]	weakly-supervised	near-isometric	✓	point clouds
[RSO19]	unsupervised	near-isometric	✗	triangle meshes
[EGR22]	unsupervised	near-isometric	✗	triangle meshes†
[CB22]	unsupervised	near-isometric	✓	triangle meshes
[CB23]	unsupervised	near-isometric	✓	multimodal‡
[ETLC20, DCO22, LXY*22, SMJ*23] [LNS*24, MO24, LMH*24, CEA*24, LLH*25]	unsupervised	non-isometric	✗	triangle meshes
[GR20, CRB23, CRB24, CLB24]	unsupervised	non-isometric	✓	triangle meshes
[HLL*23]	unsupervised	non-isometric	✓	both
[APO21]	both	near-isometric	✓*	triangle meshes
[LDO22, AO23b]	both	non-isometric	✗	triangle meshes
[AO23a]	-◇	non-isometric	✗	triangle meshes

* Capable of partial-to-partial correspondence.
† Inference can be done on point clouds.
‡ Allows for correspondences between triangle meshes and point clouds.
◇ Is optimized per shape pair.

Table 5: *Summary of recent learning-based functional map methods.*

In the following, we review changes made to different aspects of the deep functional map pipeline.

Regularization. [GR20] introduce cyclic functional mapping. They achieve self-supervision by preserving pairwise distances in the source-to-target-to-source mapping.

To achieve unsupervised learning, SURFMNet [RSO19] promotes structural properties such as bijectivity, orthogonality, and commutativity with the LBO. In [LMH*24], commutativity between the functional map and the multiscale spectral manifold wavelet operator is enforced, promoting a map that is as isometric as possible.

[CLB24] propose using synchronous diffusion. The key idea is that diffusing a function on one shape and then transporting it to another should yield the same result as first transporting the function and then diffusing it there. The difference between the two resulting functions serves as a loss term that encourages smoothness.

Instead of hand-crafting a regularization term, [PLDO25] propose to learn functional map regularization from high-quality functional maps. Their approach employs a spectral diffusion model to capture common structural properties of functional map matrices C. Given the initial *raw* functional map recovered from features, the spectral diffusion network produces a mask, which is used to regularize the raw map. The resulting learned regularization is category-agnostic, *i.e.*, it can be applied to shape categories beyond those seen during training.

Consistency with Vertexwise Maps. Attaiki and Ovsjanikov [AO23b] show that, under mild conditions, features learned in the deep functional map pipeline can be used to construct a vertexwise map directly via feature similarity. They leverage this insight to improve training by promoting consistency between the directly computed vertexwise map and the vertexwise map obtained from the functional map. [MO24] avoid storing the soft vertex matrix P by directly computing the product with the basis $P\Phi^{\mathcal{S},A}$ without ever explicitly computing the full matrix P.

SNK [AO23a] eliminates the need for training data by optimizing the network and thus the map for a pair of input shapes directly. A feature extractor and an encoder-decoder pair are used in parallel. The extracted features are used to compute a functional map, while the encoder generates a latent representation of the source shape. The decoder then uses this representation to deform the source shape into the target shape. The loss penalizes discrepancies between the correspondence implied by the deformation and the vertexwise correspondence obtained from the functional map.

[SMJ*23] propose a method for cycle-consistent matching using a spectral latent shape as intermediate domain. They notice that spectral cycle-consistency alone does not suffice for achieving spatial cycle-consistency. They make use of a two-branch approach for training: From the alignment to the latent shape, a functional map as well as a soft vertex map is obtained. The loss enforces that the functional map derived from the soft vertex map matches the directly computed functional map.

Cao and Bernard [CB23] propose a method that allows for multimodal shape matching, allowing for the computation of corre-

spondences between point clouds and triangle meshes. They train a feature extractor based on DiffusionNet [SACO22] by using both point clouds and meshes for the two shapes. The features from the meshes are used to compute a functional map, which is used for regularization. The point cloud features are used for soft map computation based on feature similarity. A contrastive loss is used to promote similarity of mesh and point cloud features. At inference, it is decided between functional maps and direct soft maps based on the input modalities: Functional maps can only be employed for mesh-to-mesh matching.

Learning Constraints. Pipelines with hand-crafted input descriptors generalize poorly across diverse datasets, so Deep Geometric Functional maps [DSO20] make use of a feature extractor that learns the features directly from the geometry. These features are projected to the spectral bases. They use a spectral loss that avoids the conversion to a pointwise map during training. Their architecture is trainable with rather small training data size. Numerous methods have since trained feature extractors rather than feature transformers.

To achieve orientation-awareness, complex functional maps are adapted to the deep functional map framework by [DCO22]. They use an orientation-preserving loss for the training that promotes isometry.

[LLH*25] introduce a spectral filter operator preservation constraint with the goal to preserve frequency information. A set of spectral filters, based on orthonormal Jacobi polynomials, is learned and used as loss. Due to this, their method better incorporates information of different frequencies.

In [SO20], features for point clouds are learned using PointNet++ [QYSG17] as feature extractor. These features are used for the computation of spectral descriptors.

To get rid of the spectral resolution hyperparameter k determining the number of basis vectors, [LDO22] can adapt the spectral resolution to be well suited to the input.

[LPCH24] uses the boundaries of a segmentation as constraints together with a relation constrained capsule graph network. Their method heavily relies on the segmentation: The number of segments needs to be consistent making it mostly suitable for models with limbs.

Functional Map Extraction. [LXY*22] propose a map extraction layer, which, instead of taking the extracted features as constraints directly, requires the multiscale spectral manifold wavelet transform of the features to be preserved by the functional map. It allows for better incorporation of the information given by the extracted features.

[HLL*23] propose RFMNet, a robust method to compute functional maps. They use DiffusionNet [SACO22] as feature extractor and use the resulting vertexwise features to obtain an initial vertexwise map using optimal transport. The correspondence is then converted to a functional map, where additional regularization in the form of wavelet preservation, inspired by [HLLL21], is applied.

Functional maps have also been used in a diffusion based approach [ZLG25], where a diffusion network predicts the functional map C. The map C is initialized as a noise-filled matrix and the diffusion network will iteratively denoise it. In addition, an unsupervised sign correction network is used to correct the signs of the LBO eigenvectors, which restricts the solution space and therefore reduces the complexity.

Functional Maps as Regularizer. In learning-based approaches, functional maps are commonly employed as regularization during training. To combine the strengths of spatial and spectral methods, [EGR22] use a spectral teacher for a spatial student, *i.e.*, a spatial network is trained using functional maps. This improves the inference speed, as it is purely spatial, while the training can still benefit from the global and local view spectral methods offer.

In [CB22], the functional map is used for regularization during the unsupervised training by promoting cycle-consistent correspondences between multiple shapes using a virtual template.

[CRB23] propose to couple the functional map directly to the vertexwise map via a loss. They train a feature extractor to compute vertexwise features. Those are then used to compute a functional map and a vertexwise map simultaneously. The method is further refined in [CRB24] using a self-adaptive functional map solver. It can adapt to different shape correspondence scenarios such as non-isometry or partiality, where the functional map matrix C deviates from being diagonal. Two parameters control the amount of penalty on off-diagonal entries and the overall regularization strength. Those parameters are learned during training.

[CEA*24] propose a method that solves the shape correspondences problem in conjunction with shape interpolation, which allows them to combine intrinsic (spectral) information, via the functional map as regularizer, and extrinsic (spatial) information.

In [LNS*24], they compute a soft vertex-to-vertex map using an optimal transport problem regularized by a functional map loss. The cost for the optimal transport, which is based on the sliced Wasserstein distance, is computed given soft feature similarity and extracted features.

Partial Matching. A key challenge of the functional map framework is the computation of maps between partial shapes. Most partial methods are only capable of full-to-partial matching [SO20, CRB23, CB23, HLL*23, CRB24, CLB24]. In contrast, partial-to-partial matching is significantly more difficult, as the regions of overlap between shapes are unknown. Only a few approaches within the functional map framework currently address this setting [APO21, BRGK23, XER*25].

[BDK24a] argue that the functional map extraction layer used in deep functional maps is fundamentally flawed when dealing with partial shapes. If the features corresponding to regions present in only one of the shapes are nonzero, errors are introduced in the map computation. To address this issue, one would either need to predict the overlapping regions prior to computing the functional map, or account for the non-overlapping regions already during the feature extraction process, which is not possible in a Siamese fashion.

[APO21] learn features directly from data and present a novel

module that allows for communication between the features of the two different shapes via cross-attention. In addition, they train an overlap predictor.

[XER*25] introduce EchoMatch, which allows for partial-to-partial matching of two shapes by computing two partial-to-full correspondences, one for each direction, using a virtual full shape. They determine the overlapping points via correspondence reflection: If a point corresponds to its neighborhood after being mapped forward and backward, it is considered overlapping. Initial per-vertex features are obtained via DiffusionNet [SACO22]. Based on them, an overlap score is computed per vertex using a soft point map.

Rather than predicting a vertex-to-vertex map, [BRGK23] address the problem of overlap prediction for partial shape pairs. They first predict landmarks on both full and partial shapes using a pretrained feature extraction network regularized by a functional map. These landmarks are then used to learn an indicator function, represented as a neural intrinsic field, obtained by interpolating the landmark-associated values. The resulting indicator function encodes the support region of one partial shape within the other.

8 Other Approaches

Several methods cannot easily be assigned to one of the aforementioned paradigms. We present these approaches and their core ideas in the following.

Regionwise Matchings. Apart from a few exceptions, most methods establish correspondences on a point-to-point basis. For shapes belonging to the same semantic class but exhibiting substantial geometric variability, it is often more appropriate to ask for correspondences between regions rather than specific points. In this spirit, [GTOG16] exploit the observation that, although the absolute values of descriptor functions may differ for non-isometric shapes, their relative ordering remains comparable. For example, while curvature values at corresponding locations may not be identical across shapes, regions of high curvature tend to correspond. Based on this insight, they define similarity in terms of rank: Points are considered similar if they share the same or similar ranks w.r.t. multiple descriptor functions. Based on this similarity measure they match corresponding regions.

[KO19] explore the idea of finding the correspondence between coarse regions of the surfaces, making the approach less sensitive against local isometric distortion. First, the shapes are jointly decomposed based on a descriptor. As a consequence, the method inherits the properties of the descriptor, so ideally the descriptor is invariant under near-isometric deformations and robust to noise. The decomposition is followed by a graph matching step, which allows to extract the correspondence between regions.

Zero-Shot 3D Shape Correspondence [AEOW23] approaches correspondence from a purely semantic perspective. It derives a sparse, region-level alignment using pretrained vision and language models. First, the class of each input shape is predicted from multiple rendered views. Language models then infer typical parts for both shapes and establish relations between them. Finally, a segmentation model partitions the surfaces according to the provided semantic labels.

Learned Embeddings. Limp [CNH*20] is a standard VAE with PointNet [QSMG17] as its encoder. Unlike deformation-based methods, it employs an intrinsic loss that enforces distance consistency under interpolation: For a convex combination of latent codes, the loss requires that the geodesic distances on the decoded shape match the corresponding convex combination of the input distances. Similarly, [YLC*20] use a point-based network to learn pointwise features. These features will be used to extract a vertex-to-vertex map. The learning process is regularized by enforcing cycle-consistency between the maps extracted between the source shape, a rigidly transformed source shape, and the target.

Effectively, [AO22] propose a *denoising* strategy for maps of a shape collection, called Neural Correspondence Prior. The approach has two stages. In the fist stage, any unsupervised method can be used to predict correspondences of possibly poor quality containing outliers. In the second stage, a neural network is is trained with these poor correspondences with a point-to-point loss. Based on this neural network, feature embeddings are learned from which the correspondences can be extracted through nearest-neighbor search or a functional map pipeline.

Deep Coupled Embedding [ZGC25] learns a high-dimensional embedding for points of point clouds. They combine a variant of DiffusionNet [SACO22] with cross-attention. The network is trained with an unsupervised loss promoting mutual consistency between the two embedded point clouds. The vertex-to-vertex map is finally retrieved by a nearest-neighbor search of the coupled embeddings. Similarly, for partial shapes, [BDK24a] use DiffusionNet [SACO22] with shared parameters for both input shapes with a cosine similarity loss to compute an embedding.

Convolutional Operators. To enable neural networks to process surface meshes both effectively and efficiently, a number of intrinsic mesh convolution operators have been proposed. Among them, SpiralNet++ [GCBZ19] introduces a formulation that explicitly serializes the local mesh neighborhood of each vertex into a fixed-length spiral ordering. The resulting ordered feature vector is processed by a fully connected layer, allowing the network to learn discriminative local features, which in turn can be used to successfully predict vertex-to-vertex maps.

FanNet [SKS25] follows a related serialization-based paradigm, but organizes the local neighborhood around a vertex as a fan-shaped structure composed of multiple spokes. Starting from a central vertex, spokes are grown by selecting successive edges such that each spoke remains as straight as possible in the intrinsic geometry of the surface. To improve robustness w.r.t. mesh resolution and irregular connectivity, FanNet samples representative vertices along each spoke at prescribed distances from the center vertex. The resulting structured neighborhood encoding captures local geometric context while maintaining computational efficiency, making it well suited for dense correspondence estimation.

9 Topological Aspects of Correspondences

Despite the fundamental interplay between geometry and topology, only few works have explicitly considered topological aspects of correspondences. We will cover their contributions in more detail in the following.

9.1 Homeomorphism Recovery

Many common representations of surface maps, *i.e.*, element-to-element maps or functional maps, are not continuous nor bijective in a strict sense. It is possible to convert such a representation into an actual surface homeomorphism by applying a method with an *intrinsic map* representation and bijectivity guarantees using the correspondences provided by the input map as guidance. These methods typically employ a *topology-first* strategy, *i.e.*, they start from an initial map and then continuously optimize its geometric quality *within* a fixed homotopy class. Since these methods rely on continuous deformation, they cannot traverse the space of homotopy classes, making initialization and homotopy-aware modeling critical for achieving meaningful correspondences. In the top row of Figure 7, the initial map is in an unintended homotopy class, and therefore introducing unnecessary distortion. The initial map in the bottom row is in the expected homotopy class and ready for optimization.

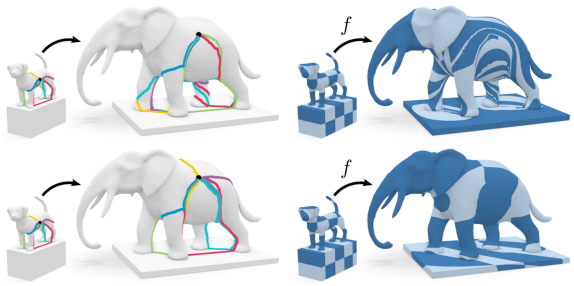

Figure 7: *If the paths of the cut graph lie in unexpected homotopy classes (top row, left), the corresponding map f (top row, right) shows artifacts. When the paths of the cut graph are in the expected homotopy class (bottom row), the resulting map f is well suited for optimization. Adapted from [BSCK21].*

To the best of our knowledge, two works have been proposed that address topology-related aspects of map recovery, identifying the *intended* homotopy class of the map.

Layout Embedding. A popular approach for obtaining the initial map involves connecting a sparse set of landmarks on both surfaces in a compatible manner, *i.e.*, if two landmarks are connected on one surface, the corresponding landmarks are also connected on the other surface, and the edges around each vertex follow the same cyclic order. The resulting graph structure, called *layout* $\mathcal{L}$, is an abstract 2-dimensional cell complex [BSK21] without geometric information.

Consider a set of layout vertices on a shape that should be connected according to a prescribed connectivity. Even if only the homotopy of the paths connecting the vertices is considered, there are infinitely many choices for routing a path around the other embedded layout vertices, as shown in Figure 8. These choices constitute the discrete degrees of freedom of the layout embedding problem. Additionally, even if the homotopy classes are fixed per path, the geometric shape of a path is a continuous degree of freedom.

To make the problem tractable, a common strategy is to embed edges sequentially as shortest paths between layout vertices. In this setting, the layout embedding is fully determined by the *insertion order* of the layout edges. It is important to note that paths are not allowed to intersect. Thus, earlier insertions may block the optimal path between other layout vertex pairs. As a result, these paths must take detours to avoid intersections with earlier insertions while preserving the cyclic ordering. As a consequence, the insertion order implicitly determines the homotopy class of the embedding.

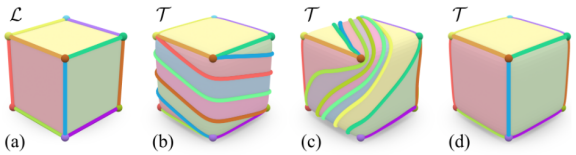

Figure 8: *For a fixed layout $\mathcal{L}$ with a fixed embedding of the layout vertices in a target mesh $\mathcal{T}$, infinitely many topologically distinct embeddings for the layout edges are available. The overall shortest embedding (d) corresponds to the embedding that is generally considered intuitive. Taken from [BSK21].*

While there is no objectively *wrong* homotopy class for the embedding, usually the overall shortest embedding corresponds to the *intended* layout embedding which aligns with human intuition. Earlier works [PSS01, SAPH04] have chosen the insertion order of edges in a greedy fashion based on heuristics, but they did not allow for backtracking. Consequently, a wrong decision early on would result in paths belonging to an *unintuitive* homotopy class, leading to long detours and twists.

To circumvent this problem, Born *et al.* [BSK21] proposed a *branch-and-bound* approach, optimizing for the insertion order leading to the overall shortest embedding. They proposed a custom branch-and-bound strategy, capable of handling the large search space. They demonstrate that various downstream applications, *e.g.*, quad-meshing and inter-surface mapping, benefit from the robustness of their method. However, their method is only applicable to genus-zero meshes because they assume that every closing cycle encloses a patch with disc topology. For higher genus objects, this is not the case, as a cycle can wrap around one or more handles.

Map Homology Inference. Most shape correspondence methods either provide only sparse correspondences or work with representations that are not trivially extendable to an embedded point-to-point map. In practice, maps may also contain defects such as noise, outliers, or overlaps, making them inherently ambiguous w.r.t. map topology. Converting such a map to a true homeomorphism requires resolving these ambiguities and defects in a globally consistent manner and committing to a specific, intended map topology.

[BSCK21] propose a method for map homology inference from

imperfect input correspondences. Specifically, they compute a *homology map*, *i.e.*, a one-to-one correspondence between the homology classes of cycles on the source and target surfaces, derived from the input correspondences. While the homology map does not fully determine the homotopy of a homeomorphism, it captures key topological features by identifying corresponding handles and tunnels.

The method takes as input two closed surfaces (triangle meshes) together with an initial, potentially low-quality correspondence relation. For each surface with genus g, a homology basis is computed, represented by $2g$ cycles of oriented edges per mesh. From this, the homology map is inferred and represented as a matrix $M \in \mathbb{Z}^{2g \times 2g}$. Computationally, this amounts to solving an integer quadratic program. The key challenge lies in computing the image of the homology basis under an unreliable map. Instead of mapping them directly, the authors [BSCK21] map the low-frequency cohomology basis, which is maximally smooth and comparable across the entire surface. They further incorporate a global topological consistency constraint, restricting the search to the subspace of homology maps compatible with homeomorphisms. These two ingredients make the inference robust, enabling recovery of the intended homology map even in the presence of noise, outliers, and sparsity.

Still, their method cannot distinguish paths that are homologous but non-homotopic, *e.g.*, those related by Dehn twists along separating cycles.

9.2 Correspondence between Surfaces with Different Topologies

Even though a homeomorphism exists only between topologically identical surfaces, correspondences between topologically different surfaces can still be of interest, *e.g.*, between chairs whose backrests contain a different number of spindles. While several approaches can operate on surfaces with differing topology, they do not explicitly account for topological differences. Park *et al.* [PCBL16] propose a method that establishes correspondences between surfaces with different topology by removing an appropriate set of points from each original surface, yielding modified surfaces that admit a homeomorphism.

At the core of their approach is a deformation process that aligns the source mesh with the target mesh. Unlike most methods, they adopt an Eulerian viewpoint: Operating on an implicit surface representation, they compute the morph from source to target by iteratively tracking the deformation of the implicit surface and applying the same deformation to the explicit mesh. A key advantage of this formulation is that it produces a velocity field defined over the entire domain, which both drives the deformation and identifies suitable locations for topological surgery, namely, the non-degenerate saddle points of the velocity field.

9.3 Adapted Objective Functions

Numerous formulations for isometric maps rely on a notion of shortest distance, *e.g.*, the geodesic distance, between pairs of surface points, typically vertices. However, this becomes problematic on meshes with topological artifacts, such as holes, accidental shortcut tunnels (*e.g.*, nearly degenerate triangles between separate features), or merged features like touching hands, as the shortest distance between nearby vertex pairs can change drastically depending on whether an artifact is present or not.

[SH23] introduce reodesics, a robust counterpart to minimal geodesics in the presence of topological noise, such as shortcut tunnels. A reodesic between two vertices is constrained to pass through a third vertex, such that they are locally shortest but not necessarily globally shortest. To make reodesics comparable across two surfaces, one anchor correspondence must already be known: This correspondence is used to reroute paths between other vertex pairs. Given a *brute-force* map, computed without accounting for topological noise, the reodesics can be used to validate candidate matches: A prediction is accepted if the corresponding reodesics agree. For additional robustness, the paths are augmented with descriptors, *i.e.*, heat vectors, for comparison.

The Wormhole loss [BDK24b] circumvents the problem of partial shapes containing holes by identifying *consistent* pairs, *i.e.*, pairs whose geodesic distance remains unchanged between the complete and partial shape. To detect such pairs, it employs a conservative heuristic: for each candidate pair, it computes both the geodesic distance between the points, and a lower bound given by the sum of their distances to the nearest boundary plus the Euclidean distance between those boundary points. If the geodesic distance is smaller than this bound, the connection is guaranteed to not be influenced by a hole, and the pair is deemed consistent. Intuitively, the heuristic investigates if a pair of vertices is intrinsically closer to each other than to the closest boundary. If this is the case, their geodesic distance is unaffected by holes. Notably, this formulation requires no access to the full shape and is thus suitable for unsupervised training.

10 Evaluation of Shape Correspondence

From a theoretical standpoint, there exist infinitely many possible correspondences between two shapes. The goal of shape correspondence establishment is therefore to identify the most meaningful correspondence within this vast space.

This naturally leads to the need for a *quantitative assessment* of correspondence quality. Beyond serving as a benchmark for algorithm performance, such assessments can also act as proxies for *shape similarity*, which is particularly relevant in downstream tasks such as shape retrieval [vKZHC11]. However, defining and evaluating correspondence quality quantitatively remains challenging. In fact, identifying a suitable evaluation metric is often as difficult as solving the correspondence problem itself, because the metric could be used as objective in optimization based approaches. Ideally, a correspondence should capture both geometric and semantic aspects, but semantics are especially hard to formalize and quantify. As a result, the community has developed a range of visualization techniques that support *qualitative analysis* and facilitate the comparison of correspondence algorithms.

In this section, we review the available evaluation tools, starting with quantitative metrics and then discussing visualization techniques for qualitative analysis.

10.1 Quality Metrics

Geodesic Error. Whenever *ground truth* correspondences are available, the most straightforward way to evaluate map quality is by computing the sum of the *geodesic distances* between the mapped points and their ground truth counterparts, typically normalized by the square root of shape area, commonly referred to as Princeton protocol [KLF11]. For this purpose, indirect map representations, *e.g.*, functional maps, must first be converted into an explicit map representation, *e.g.*, a vertex-to-vertex map. When working with point clouds, geodesic distances may not be directly accessible. The euclidean distance can serve as a reasonable approximation.

To quantitatively compare two maps, a common measure is the *cumulative geodesic error*, defined as the fraction of points (or surface area) whose geodesic error falls below a given threshold [KLF11]. By varying the threshold, one obtains a cumulative curve that concisely summarizes the map's accuracy across different error tolerances.

Feature Matching. Since a meaningful correspondence relates similar features on the surfaces, a natural quality measure is to compare and quantify the difference between features associated with corresponding points. Such features can range from simple, locally derived geometric quantities, *e.g.*, mean or Gaussian curvature, principal curvatures, or local feature size (*e.g.*, shape diameter function), to more sophisticated descriptors.

A particularly important class comprises *shape signatures*, which are constructed to be characteristic or invariant under certain transformations, such as rigid motions, scaling, and isometries. By design, these signatures aim to be unique (or nearly so) within a given object class, thereby enabling identification of corresponding structures across shapes. The choice of signature effectively defines the semantic notion of similarity. Popular examples include the Pose-Oblivious Shape Signature [GSC07], Global Point Signature [Rus07], SHOT [TSS10], Heat Kernel Signature [SOG09], Wave Kernel Signature [ASC11], and Biharmonic Kernel Signature [Rus11]. For a more exhaustive list of available spectral descriptors, we refer to the survey [LWY*25].

Finally, features can also be estimated in a data-driven manner via neural networks [QSMG17, QYSG17, SACO22].

Surface Approximation Error. For methods that work with an extrinsic map representation, the surface approximation error evaluates how well the mapped source surface aligns with the target surface. Possible candidates are the *Chamfer distance*, capturing the average deviation between two surfaces by integrating point-to-surface distances, and the *Hausdorff distance*, measuring the maximum deviation, *i.e.*, the largest pointwise mismatch between the two shapes.

Distortion. Analogously to classical surface parametrization tasks, correspondences, in particular inter-surface maps, are often assessed by measuring overall distortion based on the map Jacobian J_Φ.

For piecewise-linear map definitions, the Jacobian is constant per face: Consider a face f in the common triangulation of both input surfaces $\mathcal{S}_\mathcal{A}$ and $\mathcal{S}_\mathcal{B}$, with vertex positions $a_{\mathcal{S}_\mathcal{A}}, b_{\mathcal{S}_\mathcal{A}}, c_{\mathcal{S}_\mathcal{A}}$ on $\mathcal{S}_\mathcal{A}$, and corresponding positions $a_{\mathcal{S}_\mathcal{B}}, b_{\mathcal{S}_\mathcal{B}}, c_{\mathcal{S}_\mathcal{B}}$ on $\mathcal{S}_\mathcal{B}$. A local coordinate system is constructed by defining the edges $e_0 = b_{\mathcal{S}_\mathcal{A}} - a_{\mathcal{S}_\mathcal{A}}$ and $e_1 = c_{\mathcal{S}_\mathcal{A}} - a_{\mathcal{S}_\mathcal{A}}$ on $\mathcal{S}_\mathcal{A}$, and analogously $e_0' = b_{\mathcal{S}_\mathcal{B}} - a_{\mathcal{S}_\mathcal{B}}$ and $e_1' = c_{\mathcal{S}_\mathcal{B}} - a_{\mathcal{S}_\mathcal{B}}$ on $\mathcal{S}_\mathcal{B}$. The local Jacobian of the map Φ is then given by:

$$J_\Phi(f) = [e_0' \; e_1'][e_0 \; e_1]^{-1}.$$

The singular value decomposition of J_Φ yields the singular values σ_1 and σ_2, which characterize local stretching along the principal directions. Based on the singular values, several distortion measures are commonly used [SS15, SBCK19, SCBK20, ASGS23]. Here, we present two examples and express them as pointwise measures.

- **Symmetric Dirichlet energy:**

$$E_{\text{SDE}}(J_\Phi) = \sigma_1^2 + \sigma_2^2 + \sigma_1^{-2} + \sigma_2^{-2},$$

- **As-rigid-as-possible:**

$$E_{\text{ARAP}}(J_\Phi) = (\sigma_1 - 1)^2 + (\sigma_2 - 1)^2.$$

The overall distortion induced by a map Φ is obtained by integrating the pointwise measure $E_\star(J_\Phi)$ over the input surfaces. In the discrete setting, this corresponds to integrating the pointwise measure per face f, and summing this over all triangles (with piecewise constant Jacobian $J_\Phi(f)$) of the common triangulation:

$$E_\star(\Phi) = \sum_{f \in F} |f| \, E_\star(J_\Phi(f)),$$

where $|f|$ is the area of face f on $\mathcal{S}_\mathcal{A}$.

In the correspondence setting, neither mapping direction is favored over the other, so we are interested in symmetric objectives satisfying $E_\star(\Phi) = E_\star(\Phi^{-1})$. Not every distortion measure $E_\star$ is symmetric, but it can easily be symmetrized by computing [ASGS23]:

$$E_\star^{\text{total}}(\Phi) = E_\star^{\text{sym}}(\Phi) = \frac{1}{2} E_\star(\Phi) + \frac{1}{2} E_\star(\Phi^{-1}).$$

Expressed in terms of singular values for a pointwise measure, this leads to [ASGS23]:

$$E_\star^{\text{sym}}(J_\Phi) = \frac{1}{2} E_\star(J_\Phi) + \frac{1}{2} |\sigma_1 \sigma_2| \, E_\star(J_\Phi^{-1}).$$

For an in-depth discussion, we refer to [ASGS23]. Here, we focus on the key observation: An asymmetric distortion measure may promote isometry (*i.e.*, its minimum is at $(\sigma_1, \sigma_2) = (1, 1)$), while its symmetric version does not. In Figure 9, it can be seen that the symmetric version of ARAP promotes isometry, while the symmetric version of the SDE does not. However, this property can be restored by a slight modification of the energy: $E_{SDE'} = E_{SDE} - 4$.

In addition to reporting the global value $E_\star^{\text{total}}$, one can also consider the *cumulative distortion distribution*, defined as the percentage of surface area where the distortion does not exceed a given threshold. Varying the threshold yields a cumulative curve that highlights how distortion is distributed across the surface.

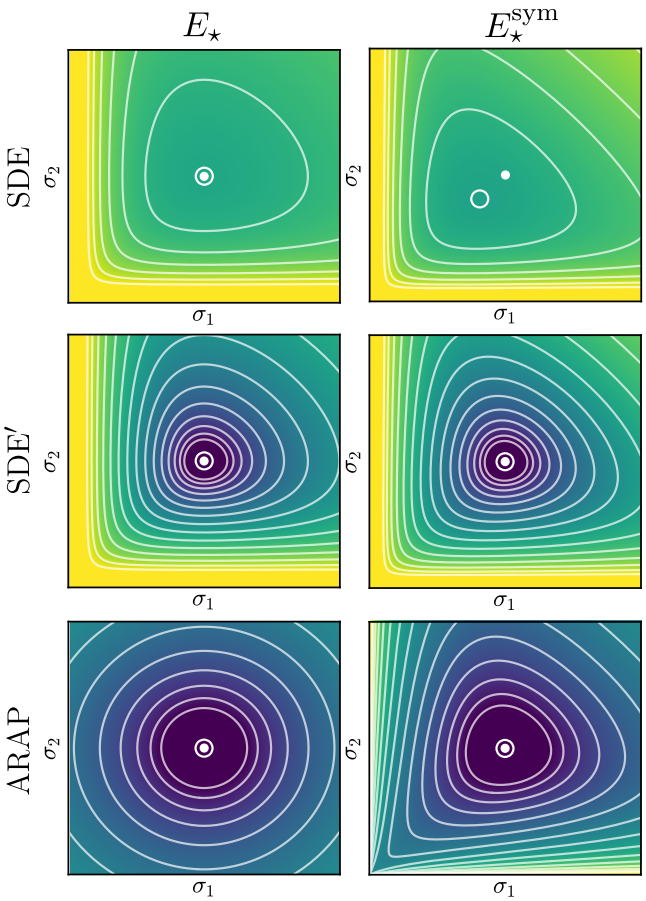

Figure 9: *Following the visualization style of [ASGS23], we plot the energy landscape (dark colors correspond to small values) for different distortion measures and their symmetrized versions on* $[0,2]^2$*. The minimum of the function is indicated by a circle, and a dot at* $(1,1)$ *marks the expected location of the minimum for an isometry-promoting function. Note that the symmetrized version of the standard SDE does not promote isometry.*

Elastic Energy. The previous formulations for measuring distortion are usually used for near-isometric shape matching. For matching shapes that are not approximately isometric, a common alternative is to use *elastic energies* [EHA*19, EEB20]:

$$E_{\mathrm{elastic}}(\mathcal{S}_\mathcal{A},\mathcal{S}_\mathcal{B}) = E_{\mathrm{mem}}(\mathcal{S}_\mathcal{A},\mathcal{S}_\mathcal{B}) + \omega_{\mathrm{bend}} E_{\mathrm{bend}}(\mathcal{S}_\mathcal{A},\mathcal{S}_\mathcal{B}),$$

where $E_{\mathrm{mem}}(\mathcal{S}_\mathcal{A},\mathcal{S}_\mathcal{B})$ is a membrane energy based on Jacobian of the map, penalizing area distortion. $E_{\mathrm{bend}}(\mathcal{S}_\mathcal{A},\mathcal{S}_\mathcal{B})$ is a bending energy penalizing misalignment of curved features, and ω_{bend} is the corresponding weighting factor balancing the contribution of the bending term.

10.2 Visualization Techniques for Shape Correspondence

Beyond quantitative analysis, visual inspection remains a crucial component in evaluating shape correspondence. It enables assessment of whether a computed map matches human intuition, which is inherently difficult to formalize numerically.

A range of visualization techniques has been developed, each emphasizing different aspects of correspondences:

- **Connecting lines between corresponding points** offers a direct and intuitive visualization of discrete point-to-point correspondences. To maintain visual clarity, this visualization is typically applied to a sparse subset of points rather than the entire set. It is particularly effective for verifying whether prominent geometric or semantic landmarks are consistently and correctly matched.
- **Segmentation transfer** consists of partitioning the source surface into meaningful regions and propagating this segmentation to the target surface via the correspondence. This approach enables assessment of whether large regions are mapped coherently. When the segmentation encodes semantic structure, it directly reveals whether the correspondence preserves semantics.
- **Texture transfer**, typically using a checkerboard pattern with annotated quadrilaterals, reveals qualities of the map such as smoothness, local distortion, and continuity. Moreover, they can expose local fold-overs or inversions that may be overlooked by purely quantitative metrics.
- **Pushforward (or Pullback) of surface normals** is an effective way to show higher-frequency details of a correspondence. The idea is to compute the normals on the source surface and transfer these normals to the target surface. The pushforward can be visualized directly on the target surface using a color map on the unit sphere [HSA*23] or they can be used to shade the target surface [MMM*19]. This approach is particularly effective in highlighting correspondence quality of high curvature regions such as creases.
- **Local distortion** per face allows the analysis of where and to what extent the map introduces distortion. While the integration of distortion provides a global measure of quality, visualizing the local distortion offers more detailed insights into the spatial distribution of mapping artifacts.
- **Triangulation transfer** from source to target allows to see if the matching is geometrically consistent. If it is not, artifacts such as large triangles spanning the space between distant features may appear.
- **Morphing animations** that interpolate between source and target shapes facilitate qualitative assessment of the overall continuity and plausibility of the correspondence by visualizing the deformation from one shape to the other.

Together, these visualization strategies complement quantitative metrics by offering qualitative insights that are crucial for practical assessment and algorithm development.

11 Datasets

For a fair comparison of different approaches, standardized evaluation datasets are essential. This need becomes even more critical as data-driven methods, which rely heavily on large, high-quality datasets, gain popularity. To provide a clear overview, we summarize the datasets relevant to shape correspondence. These datasets vary in multiple aspects, including size, availability of ground truth correspondences, presence of landmarks, and degree of isometry.

Table 6 lists the relevant datasets along with their key characteristics. Additional datasets are discussed in [DYDZ22], which can also be used for shape correspondence but have a bigger focus on registration.

- **Size** gives the number of models and classes that are present in the dataset. Typically, data-driven methods rely on a large number of training examples.
- **Ground truth** states whether dense ground truth annotations exist. They can either be computed with a method of choice or be obtained from scan data. Most datasets only give intra-class ground truth.
- **Landmarks** are the sparse counterpart to dense ground truth correspondences. Typically, a dataset has either dense ground truth or sparse landmarks. Landmarks are usually obtained from handcrafted annotations.
- **Similarity** describes how similar shape pairs are. A correspondence is *isometric* if pairwise geodesic distances are preserved. In practice, however, even for different poses of the same object, pairwise geodesic distances are not exactly preserved. Thus, correspondences between different poses of the same object and correspondences between different objects of the same class are considered *near-isometric*, while correspondences between objects of different classes are *non-isometric*. This classification is heuristic, as there is no universal threshold for distance preservation, and introduces a higher level of abstraction, requiring a decision about class membership. We consider a dataset near-isometric when only intra-class ground truth is provided, even though correspondences between different classes could in principle be computed.
- **URL** provides a link (if available) to the dataset's website or GitHub repository as of the date of publication.
- **In the description** in the lower line, we provide additional information of the dataset.
- **Difficulty** is indicated by the colored dot, ranging from green (easy) to red (hard). Difficulty depends on factors such as similarity, partiality, variations in density and connectivity, topological artifacts, and non-manifoldness. Although this metric is inherently subjective, it provides a useful initial overview.

12 Outlook and Future Work

We provide an organized view of shape correspondence problems and discuss common paradigms for establishing and refining correspondences between surfaces. We highlight topological considerations in the context of shape correspondence and summarize approaches for evaluation alongside an overview of available datasets. We hope that this organized overview facilitates future research.

In conclusion, methods based on data-driven features and learning-based approaches achieve strong performance across a range of settings, from near-isometric to more challenging non-isometric cases. Each of the discussed paradigms has distinct strengths and limitations. Combinatorial methods can offer strong guarantees of geometric consistency and global optimality, but they usually struggle with scalability, for instance when input geometries have triangle counts exceeding an order of magnitude of 10^3. Accordingly, a common use case for these methods is to serve as an initialization for subsequent correspondence refinement. Functional map methods are appealing due to their compact formulation in function space and are frequently used to extend sparse correspondences to dense ones. However, for a broad range of applications, an additional conversion step is required to obtain an element-to-element map. Deformation-based approaches enable correspondence across shape collections by embedding shapes in a latent space. Distortion-based methods relying on intrinsic representations can convert sparse or dense element-to-element maps into continuous point-to-point maps, providing strong bijectivity guarantees. Nevertheless, they face scalability challenges, with runtimes reaching up to several days, are largely restricted to well-defined 2-manifold inputs, and tend to struggle in highly non-isometric settings.

In the full-to-full setting, and in line with the benchmarks in BeCoS [EAX*25], one of the remaining challenges is *semantic* incompleteness, where corresponding parts are meaningfully different or missing despite both shapes being *geometrically complete*. Typical examples include the presence or absence of horns on animals or armrests on chairs. More comprehensive and standardized benchmarking, particularly across varying degrees of non-isometry, for example using datasets such as SHREC'20 [DLR*20] with subchallenges of increasing non-isometry, would help the community to better identify the limitations of current methods and allow for better comparison across methods. Overall, we argue that strongly non-isometric correspondence remains challenging even in the full-to-full setting. Benchmarking accounting for noise, topological artifacts, and outliers would also provide a clearer picture of current capabilities.

While the full-to-full setting is comparatively well studied, challenges become more pronounced once one moves beyond full-to-full correspondences. The performance of state-of-the-art algorithms degrades in full-to-partial and, in particular, in partial-to-partial scenarios. As demonstrated by the BeCoS benchmark [EAX*25], existing methods struggle in these settings, often producing noisy or unreliable correspondences. This underscores the need for further methodological advances tailored specifically to partial shape matching. At the same time, the dataset introduced in BeCoS provides a valuable foundation for more rigorous quantitative evaluation and for training learning-based approaches that may better generalize to these challenging scenarios.

Current research and publicly available datasets commonly focus on organic, low-genus shapes, such as humans and animals, where feature misalignment is not always visually apparent. In contrast, many industrial applications require precise alignment of hard features in man-made objects, which frequently exhibit higher genus. This regime remains underrepresented in existing datasets. Future work could explicitly account for the topological degrees of freedom in such cases, which is particularly important for methods based on intrinsic representations. These approaches, commonly used in distortion-driven frameworks, typically rely on continuous updates of the map and are therefore restricted to optimization within a fixed homotopy class. However, for man-made objects in particular, it may be necessary to optimize across different homotopy classes and consider maps that differ by topological twists in order to find optimal correspondences.

	Name	Size	Ground Truth	Landmarks	Similarity	URL
full shapes	SCAPE [ASK*05]	71 poses of one person	✓	-	isometric	Website
		poses of one person based on animation sequences				
	TOSCA [BBK09]	80 models 9 classes	✓	-	isometric	BeCoS GitHub
		synthetic meshes of animals and humans; compatible triangulations				
	FAUST [BRLB14]	300 scans of 10 people	✓	-	near-isometric	Website
		different poses				
	KIDS [RBW*14]	32 shapes 2 classes	✓	-	near-isometric	Website
		two different kids; contains self intersections; compatible triangulations				
	SURREAL SMPL [VRM*17, LMR*15, GFK*18]	230k human shapes via SMPL	✓	-	near-isometric	-
		also includes bent humans; [GFK*18] use SMPL [LMR*15] on SURREAL [VRM*17]				
	DeformingThings4D [LTT*21]	1972 sequences of 31 categories	✓	-	near-isometric	GitHub
		animation sequences; synthetic; compatible triangulations				
	TOPKIDS [LRB*16a]	25 shapes	✓	-	near-isometric	Website
		based on KIDS; with topological changes				
	SMAL [ZKJB17]	49 scans from 5 classes	-	✓	non-isometric	Website
		3D animal model; shapes are obtained by fitting the model to images				
	SHREC'19 [MMR*19]	44 human shapes 430 shape pairs	✓	-	non-isometric	Website
		not canonically aligned; connectivity artifacts, topological changes, and density variation				
	DT4D-Matching [MRSO22]	56 animals, 8 humans 15-50 poses each	✓	-	non-isometric	GitHub
		based on DeformingThings4D; independent remeshing; cross-category ground truth				
full and partial shapes	SHREC'16 CUTS/HOLES [CRB*16]	599 models 8 classes	✓	-	near-isometric	PyTorch
		based on TOSCA; partial meshes via cutting planes and hole growing; full-to-partial				
	FARM partial [KMP*21]	human shapes	✓	-	near-isometric	-
		based on SHREC'19; partiality; different connectivities				
	PFARM [APO21]	25 pairs from 26 human shapes	✓	-	near-isometric	GitHub
		extension of FARM partial; varying meshing & vertex density; partiality (one full shape)				
	PFAUST [BDK24a]	10 people 20 poses per person	✓	-	near-isometric	GitHub
		based on FAUST; introduces partiality: 10 full and 10 partial poses per person				
	SHREC'20 [DLR*20]	14 models of quadrupeds	-	✓	non-isometric	Website
		test sets with varying difficulty; topological changes (due to self-contact)				
	BeCoS Benchmark [EAX*25]	2543 shapes from 7 datasets	✓	-	non-isometric	Website GitHub
		framework to create large partial datasets; also partial-to-partial; cross-dataset and cross-category correspondences; annotated semantic missing parts in full shapes				
partial shapes	CP2P [APO21]	300 pairs from 8 classes	✓	-	near-isometric	GitHub
		based on SHREC'16 CUTS; partial-to-partial; only intra-class ground truth				
	PARTIALSMAL [EGR*24]	1274 shapes 8 classes	✓	-	non-isometric	GitHub
		based on SMAL; inter-class; only 404 g.t. maps; partial-to-partial via cutting planes				

Table 6: *Relevant datasets for shape correspondence.*

Finally, different applications impose specific requirements that go beyond those of the correspondence problem itself. For instance, tasks such as texture transfer or the transfer of physical simulation results require maps that are continuous and sufficiently smooth to avoid visual artifacts or implausible physical predictions on the target surface. Mesh processing applications, such as high-quality mesh transfer, benefit f rom c orrespondences t hat induce high-quality element maps, even if this sometimes comes at the expense of strict correspondence accuracy. Moreover, in this setting, all geometric features must be captured by the map, which can be challenging for explicit, element-based representations: If features are not properly sampled by the mesh, they may not be represented in the correspondence at all. For example, when transferring a hand mesh to another hand, vertices might be placed slightly too low on a finger such that the fingertip feature is no longer captured by any mesh element, and thus cannot be transferred reliably. Similarly, shape interpolation, morphing, and averaging require correspondences that produce meaningful intermediate shapes and, in particular, avoid self-intersections or other geometric degeneracies. These diverse and often application-specific demands call for more tailored correspondence frameworks that can adapt to the needs of the task at hand rather than relying on a one-size-fits-all formulation.

Acknowledgments

This work was partially funded by the German Research Foundation within the Gottfried-Wilhelm-Leibniz program and by the Excellence Strategy of the Federal Government and the States of Germany. Open Access funding enabled and organized by Projekt DEAL.

References

[AEOW23] ABDELREHEEM A., ELDESOKEY A., OVSJANIKOV M., WONKA P.: Zero-shot 3d shape correspondence. In *SIGGRAPH Asia* (2023), ACM, pp. 59:1–59:11. 17

[AGK*22] AIGERMAN N., GUPTA K., KIM V. G., CHAUDHURI S., SAITO J., GROUEIX T.: Neural jacobian fields: Learning intrinsic mappings of arbitrary meshes. *ACM Transactions on Graphics 41*, 4 (2022), 109:1–109:17. 6, 7

[AKR13] AFLALO Y., KIMMEL R., RAVIV D.: Scale invariant geometry for nonrigid shapes. *SIAM Journal on Imaging Sciences 6*, 3 (2013), 1579–1597. 14

[AL21a] ATZMON M., LIPMAN Y.: SALD: Sign agnostic learning with derivatives. In *International Conference on Learning Representations* (2021), OpenReview.net. 6

[AL21b] AZENCOT O., LAI R.: A data-driven approach to functional map construction and bases pursuit. *Computer Graphics Forum 40*, 5 (2021), 97–110. 12

[ANVL21] ATZMON M., NOVOTNÝ D., VEDALDI A., LIPMAN Y.: Augmenting implicit neural shape representations with explicit deformation fields. *arXiv preprint* (2021). arXiv:2108.08931. 6, 7

[AO22] ATTAIKI S., OVSJANIKOV M.: NCP: Neural correspondence prior for effective unsupervised shape matching. In *Advances in Neural Information Processing Systems* (2022). 17

[AO23a] ATTAIKI S., OVSJANIKOV M.: Shape non-rigid kinematics (SNK): A zero-shot method for non-rigid shape matching via unsupervised functional map regularized reconstruction. In *Advances in Neural Information Processing Systems* (2023). 15

[AO23b] ATTAIKI S., OVSJANIKOV M.: Understanding and improving features learned in deep functional maps. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), IEEE, pp. 1316–1326. 15

[APO21] ATTAIKI S., PAI G., OVSJANIKOV M.: DPFM: Deep partial functional maps. In *International Conference on 3D Vision* (2021), IEEE, pp. 175–185. 3, 15, 16, 23

[ASC11] AUBRY M., SCHLICKEWEI U., CREMERS D.: The wave kernel signature: A quantum mechanical approach to shape analysis. In *IEEE International Conference on Computer Vision Workshops* (2011), IEEE Computer Society, pp. 1626–1633. 20

[ASGS23] ABULNAGA S. M., STEIN O., GOLLAND P., SOLOMON J.: Symmetric volume maps: Order-invariant volumetric mesh correspondence with free boundary. *ACM Transactions on Graphics 42*, 3 (2023), 25:1–25:20. 20, 21

[ASK*05] ANGUELOV D., SRINIVASAN P., KOLLER D., THRUN S., RODGERS J., DAVIS J.: SCAPE: Shape completion and animation of people. *ACM Transactions on Graphics 24*, 3 (2005), 408–416. 23

[BBK09] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: *Numerical Geometry of Non-Rigid Shapes*. Monographs in Computer Science. Springer, 2009. 23

[BDK24a] BRACHA A., DAGÈS T., KIMMEL R.: On unsupervised partial shape correspondence. In *Asian Conference on Computer Vision* (2024), vol. 15480 of *Lecture Notes in Computer Science*, Springer, pp. 316–332. 16, 17, 23

[BDK24b] BRACHA A., DAGÈS T., KIMMEL R.: Wormhole loss for partial shape matching. In *Advances in Neural Information Processing Systems* (2024). 19

[BDM12] BURKARD R., DELL'AMICO M., MARTELLO S.: *Assignment Problems*. Society for Industrial and Applied Mathematics, 2012. 4, 5

[Ber90] BERTSEKAS D. P.: The auction algorithm for assignment and other network flow problems: A tutorial. *Interfaces 20*, 4 (August 1990), 133–149. 5

[BGTM20] BENKNER M. S., GOLYANIK V., THEOBALT C., MOELLER M.: Adiabatic quantum graph matching with permutation matrix constraints. In *International Conference on 3D Vision* (2020), IEEE, pp. 583–592. 7, 8

[BHK20] BRACHA A., HALIMI O., KIMMEL R.: Shape correspondence by aligning scale-invariant LBO eigenfunctions. In *Eurographics Workshop on 3D Object Retrieval* (2020), Eurographics Association, pp. 1–9. 14

[BJS11] BAZARAA M. S., JARVIS J. J., SHERALI H. D.: *Linear programming and network flows*. John Wiley & Sons, 2011. 5

[BLG*21] BENKNER M. S., LÄHNER Z., GOLYANIK V., WUNDERLICH C., THEOBALT C., MOELLER M.: Q-Match: Iterative shape matching via quantum annealing. In *IEEE/CVF International Conference on Computer Vision* (2021), IEEE, pp. 7566–7576. 8

[BRGK23] BENSAÏD D., ROTSTEIN N., GOLDENSTEIN N., KIMMEL R.: Partial matching of nonrigid shapes by learning piecewise smooth functions. *Computer Graphics Forum 42*, 5 (2023), i–viii. 11, 15, 16, 17

[BRLB14] BOGO F., ROMERO J., LOPER M., BLACK M. J.: FAUST: Dataset and evaluation for 3d mesh registration. In *IEEE Conference on Computer Vision and Pattern Recognition* (2014), IEEE Computer Society, pp. 3794–3801. 23

[BSCK21] BORN J., SCHMIDT P., CAMPEN M., KOBBELT L.: Surface map homology inference. *Computer Graphics Forum 40*, 5 (2021), 193–203. 4, 18, 19

[BSK21] BORN J., SCHMIDT P., KOBBELT L.: Layout embedding via combinatorial optimization. *Computer Graphics Forum 40*, 2 (2021), 277–290. 18

[BST20] BERNARD F., SURI Z. K., THEOBALT C.: MINA: Convex mixed-integer programming for non-rigid shape alignment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), Computer Vision Foundation / IEEE, pp. 13823–13832. 8, 9

[BTL*23] BHATIA H., TRETSCHK E., LÄHNER Z., BENKNER M. S., MOELLER M., THEOBALT C., GOLYANIK V.: CCuantuMM: Cycle-consistent quantum-hybrid matching of multiple shapes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), IEEE, pp. 1296–1305. 8

[BTST19] BERNARD F., THUNBERG J., SWOBODA P., THEOBALT C.: HiPPI: Higher-order projected power iterations for scalable multi-matching. In *IEEE/CVF International Conference on Computer Vision* (2019), IEEE, pp. 10283–10292. 7, 8

[BXNL24] BASTIAN L., XIE Y., NAVAB N., LÄHNER Z.: Hybrid functional maps for crease-aware non-isometric shape matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), IEEE, pp. 3313–3323. 12

[CB20] COHEN A., BEN-CHEN M.: Robust shape collection matching and correspondence from shape differences. *Computer Graphics Forum 39*, 2 (2020), 555–568. 14

[CB22] CAO D., BERNARD F.: Unsupervised deep multi-shape matching. In *European Conference on Computer Vision* (2022), vol. 13663 of *Lecture Notes in Computer Science*, Springer, pp. 55–71. 15, 16

[CB23] CAO D., BERNARD F.: Self-supervised learning for multimodal non-rigid 3d shape matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), IEEE, pp. 17735–17744. 15, 16

[CBM22] COLOMBO M., BORACCHI G., MELZI S.: PC-GAU: PCA basis of scattered gaussians for shape matching via functional maps. In *Smart Tools and Applications in Graphics* (2022), Eurographics Association, pp. 29–39. 12

[CBM23] COLOMBO M., BORACCHI G., MELZI S.: Extracting a functional representation from a dictionary for non-rigid shape matching. *Computers & Graphics 113* (2023), 43–56. 12

[CEA*24] CAO D., EISENBERGER M., AMRANI N. E., CREMERS D., BERNARD F.: Spectral meets spatial: Harmonising 3d shape matching and interpolation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), IEEE, pp. 3658–3668. 15, 16

[CLB24] CAO D., LÄHNER Z., BERNARD F.: Synchronous diffusion for unsupervised smooth non-rigid 3d shape matching. In *European Conference on Computer Vision* (2024), vol. 15063 of *Lecture Notes in Computer Science*, Springer, pp. 262–281. 15, 16

[CLJL20] CHEN H., LIU H. D., JACOBSON A., LEVIN D. I. W.: Chordal decomposition for spectral coarsening. *ACM Transactions on Graphics 39*, 6 (2020), 265:1–265:16. 13

[CNH*20] COSMO L., NORELLI A., HALIMI O., KIMMEL R., RODOLÀ E.: LIMP: Learning latent shape representations with metric preservation priors. In *European Conference on Computer Vision* (2020), vol. 12348 of *Lecture Notes in Computer Science*, Springer, pp. 19–35. 17

[CP21] CAMMARASANA S., PATANÈ G.: Localised and shape-aware functions for spectral geometry processing and shape analysis: A survey & perspectives. *Computers & Graphics 97* (2021), 1–18. 11

[CRB*16] COSMO L., RODOLÀ E., BRONSTEIN M. M., TORSELLO A., CREMERS D., SAHILLIOGLU Y.: Partial matching of deformable shapes. In *Eurographics Workshop on 3D Object Retrieval* (2016), Eurographics Association. 23

[CRB23] CAO D., ROETZER P., BERNARD F.: Unsupervised learning of robust spectral shape matching. *ACM Transactions on Graphics 42*, 4 (2023), 132:1–132:15. 15, 16

[CRB24] CAO D., ROETZER P., BERNARD F.: Revisiting map relations for unsupervised non-rigid shape matching. In *International Conference on 3D Vision* (2024), IEEE, pp. 1371–1381. 15, 16

[DCMO22] DONATI N., CORMAN E., MELZI S., OVSJANIKOV M.: Complex functional maps: A conformal link between tangent bundles. *Computer Graphics Forum 41*, 1 (2022), 317–334. 14

[DCO22] DONATI N., CORMAN E., OVSJANIKOV M.: Deep orientation-aware functional maps: Tackling symmetry issues in shape matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), IEEE, pp. 732–741. 15, 16

[DLR*20] DYKE R. M., LAI Y., ROSIN P. L., ZAPPALÀ S., DYKES S., GUO D., LI K., MARIN R., MELZI S., YANG J.: SHREC'20: Shape correspondence with non-isometric deformations. *Computers & Graphics 92* (2020), 28–43. 22, 23

[DSO20] DONATI N., SHARMA A., OVSJANIKOV M.: Deep geometric functional maps: Robust feature learning for shape correspondence. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), Computer Vision Foundation / IEEE, pp. 8589–8598. 12, 15, 16

[DYDZ22] DENG B., YAO Y., DYKE R. M., ZHANG J.: A survey of non-rigid 3d registration. *Computer Graphics Forum 41*, 2 (2022), 559–589. 2, 22

[DYT21] DENG Y., YANG J., TONG X.: Deformed implicit field: Modeling 3d shapes with learned dense correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition* (2021), Computer Vision Foundation / IEEE, pp. 10286–10296. 6, 7

[EAX*25] EHM V., AMRANI N. E., XIE Y., BASTIAN L., GAO M., WANG W., SANG L., CAO D., WEISSBERG T., LÄHNER Z., CREMERS D., BERNARD F.: Beyond complete shapes: A benchmark for quantitative evaluation of 3d shape surface matching algorithms. *Computer Graphics Forum 44*, 5 (2025). 22, 23

[EEB20] EDELSTEIN M., EZUZ D., BEN-CHEN M.: ENIGMA: Evolutionary non-isometric geometry matching. *ACM Transactions on Graphics 39*, 4 (2020), 112. 8, 9, 21

[EGR22] EFRONI O., GINZBURG D., RAVIV D.: Spectral teacher for a spatial student: Spectrum-aware real-time dense shape correspondence. In *International Conference on 3D Vision* (2022), IEEE, pp. 1–10. 15, 16

[EGR*24] EHM V., GAO M., ROETZER P., EISENBERGER M., CREMERS D., BERNARD F.: Partial-to-partial shape matching with geometric consistency. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), IEEE, pp. 27478–27487. 23

[EHA*19] EZUZ D., HEEREN B., AZENCOT O., RUMPF M., BEN-CHEN M.: Elastic correspondence between triangle meshes. *Computer Graphics Forum 38*, 2 (2019), 121–134. 21

[ELC19] EISENBERGER M., LÄHNER Z., CREMERS D.: Divergence-free shape correspondence by deformation. *Computer Graphics Forum 38*, 5 (2019), 1–12. 6, 7

[ELC20] EISENBERGER M., LÄHNER Z., CREMERS D.: Smooth shells: Multi-scale shape registration with functional maps. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), Computer Vision Foundation / IEEE, pp. 12262–12271. 14

[ENK*21] EISENBERGER M., NOVOTNÝ D., KERCHENBAUM G., LABATUT P., NEVEROVA N., CREMERS D., VEDALDI A.: NeuroMorph: Unsupervised shape interpolation and correspondence in one go. In *IEEE Conference on Computer Vision and Pattern Recognition* (2021), Computer Vision Foundation / IEEE, pp. 7473–7483. 6, 7

[ERE*24] EHM V., ROETZER P., EISENBERGER M., GAO M., BERNARD F., CREMERS D.: Geometrically consistent partial shape matching. In *International Conference on 3D Vision* (2024), IEEE, pp. 914–922. 8, 9

[ETLC20] EISENBERGER M., TOKER A., LEAL-TAIXÉ L., CREMERS D.: Deep shells: Unsupervised shape correspondence with optimal transport. In *Advances in Neural Information Processing Systems* (2020). 14, 15

[ETLC23] EISENBERGER M., TOKER A., LEAL-TAIXÉ L., CREMERS D.: G-MSM: Unsupervised multi-shape matching with graph-based affinity priors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), IEEE, pp. 22762–22772. 4

[GCBZ19] GONG S., CHEN L., BRONSTEIN M. M., ZAFEIRIOU S.: SpiralNet++: A fast and highly efficient mesh convolution operator. In *IEEE/CVF International Conference on Computer Vision Workshops* (2019), IEEE, pp. 4141–4148. 17

[GFK*18] GROUEIX T., FISHER M., KIM V. G., RUSSELL B. C., AUBRY M.: 3D-CODED: 3d correspondences by deep deformation. In *European Conference on Computer Vision* (2018), vol. 11206 of *Lecture Notes in Computer Science*, Springer, pp. 235–251. 23

[GLT*21] GAO M., LÄHNER Z., THUNBERG J., CREMERS D., BERNARD F.: Isometric multi-shape matching. In *IEEE Conference on Computer Vision and Pattern Recognition* (2021), Computer Vision Foundation / IEEE, pp. 14183–14193. 14

[GR20] GINZBURG D., RAVIV D.: Cyclic functional mapping: Self-supervised correspondence between non-isometric deformable shapes. In *European Conference on Computer Vision* (2020), vol. 12350 of *Lecture Notes in Computer Science*, Springer, pp. 36–52. 15

[GRE*23] GAO M., ROETZER P., EISENBERGER M., LÄHNER Z., MÖLLER M., CREMERS D., BERNARD F.: Σigma: Scale-invariant global sparse shape matching. In *IEEE/CVF International Conference on Computer Vision* (2023), IEEE, pp. 645–654. 8, 9

[GSC07] GAL R., SHAMIR A., COHEN-OR D.: Pose-oblivious shape signature. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (2007), 261–271. 20

[GTOG16] GANAPATHI-SUBRAMANIAN V., THIBERT B., OVSJANIKOV M., GUIBAS L. J.: Stable region correspondences between non-isometric shapes. *Computer Graphics Forum 35*, 5 (2016), 121–133. 3, 17

[GX13] GALLIER J., XU D.: *A Guide to the Classification Theorem for Compact Surfaces*, vol. 9 of *Geometry and Computing*. Springer, 2013. 3

[HLC20] HOLZSCHUH B., LÄHNER Z., CREMERS D.: Simulated annealing for 3d shape correspondence. In *International Conference on 3D Vision* (2020), IEEE, pp. 252–260. 8

[HLL*23] HU L., LI Q., LIU S., YAN D., XU H., LIU X.: RFMNet: Robust deep functional maps for unsupervised non-rigid shape correspondence. *Graphical Models 129* (2023), 101189. 15, 16

[HLLL21] HU L., LI Q., LIU S., LIU X.: Efficient deformable shape correspondence via multiscale spectral manifold wavelets preservation. In *IEEE Conference on Computer Vision and Pattern Recognition* (2021), Computer Vision Foundation / IEEE, pp. 14536–14545. 14, 16

[HPG*23] HERTZ A., PEREL O., GIRYES R., SORKINE-HORNUNG O., COHEN-OR D.: Mesh draping: Parametrization-free neural mesh transfer. *Computer Graphics Forum 42*, 1 (2023), 72–85. 6, 7

[HRWO20] HUANG R., REN J., WONKA P., OVSJANIKOV M.: Consistent zoomout: Efficient spectral map synchronization. *Computer Graphics Forum 39*, 5 (2020), 265–278. 13

[HSA*23] HARTWIG F., SASSEN J., AZENCOT O., RUMPF M., BEN-CHEN M.: An elastic basis for spectral shape correspondence. In *SIGGRAPH* (2023), ACM, pp. 58:1–58:11. 12, 13, 21

[JHTG20] JIANG C. M., HUANG J., TAGLIASACCHI A., GUIBAS L. J.: ShapeFlow: Learnable deformation flows among 3d shapes. In *Advances in Neural Information Processing Systems* (2020). 6, 7

[JSH23] JIANG P., SUN M., HUANG R.: Neural intrinsic embedding for non-rigid point cloud matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), IEEE, pp. 21835–21845. 12

[KLF11] KIM V. G., LIPMAN Y., FUNKHOUSER T. A.: Blended intrinsic maps. *ACM Transactions on Graphics 30*, 4 (2011), 79. 20

[KMP*21] KIRGO M., MELZI S., PATANÈ G., RODOLÀ E., OVSJANIKOV M.: Wavelet-based heat kernel derivatives: Towards informative localized shape analysis. *Computer Graphics Forum 40*, 1 (2021), 165–179. 11, 12, 23

[KO19] KLEIMAN Y., OVSJANIKOV M.: Robust structure-based shape correspondence. *Computer Graphics Forum 38*, 1 (2019), 7–20. 17

[Kuh55] KUHN H. W.: The hungarian method for the assignment problem. *Naval Research Logistics Quarterly 2*, 1-2 (1955), 83–97. 5

[LdAN*07] LOIOLA E. M., DE ABREU N. M. M., NETTO P. O. B., HAHN P., QUERIDO T. M.: A survey for the quadratic assignment problem. *European Journal of Operational Research 176*, 2 (2007), 657–690. 5

[LDO22] LI L., DONATI N., OVSJANIKOV M.: Learning multi-resolution functional maps with spectral attention for robust shape matching. In *Advances in Neural Information Processing Systems* (2022). 15, 16

[LGL*24] LI Q., GUO Y., LIU X., HU L., LUO F., LIU S.: Deformable shape matching with multiple complex spectral filter operator preservation. *The Visual Computer 40*, 7 (2024), 4885–4898. 14

[LI14] LI X., IYENGAR S. S.: On computing mapping of 3d objects: A survey. *ACM Computing Surveys 47*, 2 (2014), 34:1–34:45. 1

[LJO19] LIU H. D., JACOBSON A., OVSJANIKOV M.: Spectral coarsening of geometric operators. *ACM Transactions on Graphics 38*, 4 (2019), 105:1–105:13. 13

[LLH*25] LUO F., LI Q., HU L., WANG H., XU H., LIU X., LIU S., CHEN H.: Deep frequency awareness functional maps for robust shape matching. *IEEE Transactions on Visualization and Computer Graphics 31*, 10 (2025), 7781–7794. 13, 15, 16

[LLT*20] LESCOAT T., LIU H. D., THIERY J., JACOBSON A., BOUBEKEUR T., OVSJANIKOV M.: Spectral mesh simplification. *Computer Graphics Forum 39*, 2 (2020), 315–324. 14

[LMH*24] LIU S., MENG J., HU L., GUO Y., LIU X., YANG X., WANG H., LI Q.: Multiscale spectral manifold wavelet regularizer for unsupervised deep functional maps. *Computer Graphics Forum 43*, 7 (2024), i–xxii. 15

[LMR*15] LOPER M., MAHMOOD N., ROMERO J., PONS-MOLL G., BLACK M. J.: SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics 34*, 6 (2015), 248:1–248:16. 23

[LNS*24] LE T., NGUYEN K., SUN S., HO N., XIE X.: Integrating efficient optimal transport and functional maps for unsupervised shape correspondence learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), IEEE, pp. 23188–23198. 15, 16

[LPCH24] LIAN Y., PEI S., CHEN M., HUA J.: Relation constrained capsule graph neural networks for non-rigid shape correspondence. *ACM Transactions on Intelligent Systems and Technology 15*, 6 (2024), 121:1–121:26. 15, 16

[LRB*16a] LÄHNER Z., RODOLÀ E., BRONSTEIN M. M., CREMERS D., BURGHARD O., COSMO L., DIECKMANN A., KLEIN R., SAHILLIOGLU Y.: Matching of deformable shapes with topological noise. In *Eurographics Workshop on 3D Object Retrieval* (2016), Eurographics Association. 23

[LRB*16b] LITANY O., RODOLÀ E., BRONSTEIN A. M., BRONSTEIN M. M., CREMERS D.: Non-rigid puzzles. *Computer Graphics Forum 35*, 5 (2016), 135–143. 3

[LRR*17] LITANY O., REMEZ T., RODOLÀ E., BRONSTEIN A. M., BRONSTEIN M. M.: Deep functional maps: Structured prediction for dense shape correspondence. In *IEEE International Conference on Computer Vision* (2017), IEEE Computer Society, pp. 5660–5668. 14, 15

[LTT*21] LI Y., TAKEHARA H., TAKETOMI T., ZHENG B., NIESSNER M.: 4DComplete: Non-rigid motion estimation beyond the observable surface. In *IEEE/CVF International Conference on Computer Vision* (2021), IEEE, pp. 12686–12696. 23

[LWH*22] LIU S., WANG H., HU L., LI Q., LIU X.: Incremental functional maps for accurate and smooth shape correspondence. *The Visual Computer 38*, 9 (2022), 3313–3325. 13

[LWY*25] LIU S., WANG H., YAN D., LI Q., LUO F., TENG Z., LIU X.: Spectral descriptors for 3d deformable shape matching: A comparative survey. *IEEE Transactions on Visualization and Computer Graphics 31*, 3 (2025), 1677–1697. 20

[LXY*22] LIU S., XU H., YAN D., HU L., LIU X., LI Q.: WTFM layer: An effective map extractor for unsupervised shape correspondence. *Computer Graphics Forum 41*, 7 (2022), 51–61. 15, 16

[MAKM21] MORREALE L., AIGERMAN N., KIM V. G., MITRA N. J.: Neural surface maps. In *IEEE Conference on Computer Vision and Pattern Recognition* (2021), Computer Vision Foundation / IEEE, pp. 4639–4648. 10

[MAKM24] MORREALE L., AIGERMAN N., KIM V. G., MITRA N. J.: Neural semantic surface maps. *Computer Graphics Forum 43*, 2 (2024), i–iii. 10

[MBRM24] MAGGIOLI F., BAIERI D., RODOLÀ E., MELZI S.: ReMatching: Low-resolution representations for scalable shape correspondence. In *European Conference on Computer Vision* (2024), vol. 15095 of *Lecture Notes in Computer Science*, Springer, pp. 183–200. 14

[MMM*19] MELZI S., MARIN R., MUSONI P., CASTELLANI U., TARINI M.: Visual assessments of functional maps. In *Symposium on Geometry Processing - Posters* (2019), The Eurographics Association, pp. 5–6. 21

[MMR*19] MELZI S., MARIN R., RODOLÀ E., CASTELLANI U., REN J., POULENARD A., WONKA P., OVSJANIKOV M.: Matching humans with different connectivity. In *Eurographics Workshop on 3D Object Retrieval* (2019), Eurographics Association, pp. 121–128. 23

[MO21] MAGNET R., OVSJANIKOV M.: DWKS : A local descriptor of deformations between meshes and point clouds. In *IEEE/CVF International Conference on Computer Vision* (2021), IEEE, pp. 3773–3782. 14

[MO23] MAGNET R., OVSJANIKOV M.: Scalable and efficient functional map computations on dense meshes. *Computer Graphics Forum 42*, 2 (2023), 89–101. 14

[MO24] MAGNET R., OVSJANIKOV M.: Memory-scalable and simplified functional map learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), IEEE, pp. 4041–4050. 13, 15

[MRMO20] MARIN R., RAKOTOSAONA M., MELZI S., OVSJANIKOV M.: Correspondence learning via linearly-invariant embedding. In *Advances in Neural Information Processing Systems* (2020). 12, 15

[MRR*19] MELZI S., REN J., RODOLÀ E., SHARMA A., WONKA P., OVSJANIKOV M.: Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics 38*, 6 (2019), 155:1–155:14. 13

[MRSO22] MAGNET R., REN J., SORKINE-HORNUNG O., OVSJANIKOV M.: Smooth non-rigid shape matching via effective dirichlet energy optimization. In *International Conference on 3D Vision* (2022), IEEE, pp. 495–504. 14, 23

[MRWB23] MERROUCHE A., REGATEIRO J., WUHRER S., BOYER E.: Deformation-guided unsupervised non-rigid shape matching. In *British Machine Vision Conference* (2023), BMVA Press, pp. 417–428. 6, 7

[NBCW*11] NGUYEN A., BEN-CHEN M., WELNICKA K., YE Y., GUIBAS L.: An optimization approach to improving collections of shape maps. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 1481–1491. 4

[OBS*12] OVSJANIKOV M., BEN-CHEN M., SOLOMON J., BUTSCHER A., GUIBAS L. J.: Functional maps: A flexible representation of maps between shapes. *ACM Transactions on Graphics 31*, 4 (2012), 30:1–30:11. 5, 10, 11, 13

[OCB*16] OVSJANIKOV M., CORMAN E., BRONSTEIN M. M., RODOLÀ E., BEN-CHEN M., GUIBAS L. J., CHAZAL F., BRONSTEIN A. M.: Computing and processing correspondences with functional maps. In *SIGGRAPH Asia - Courses* (2016), ACM, pp. 9:1–9:60. 10, 11

[PCBL16] PARK H., CHO Y., BANG S., LEE S.: An eulerian approach for constructing a map between surfaces with different topologies. *Computer Graphics Forum 35*, 7 (2016), 11–19. 19

[PKO22] PANINE M., KIRGO M., OVSJANIKOV M.: Non-isometric shape matching via functional maps on landmark-adapted bases. *Computer Graphics Forum 41*, 6 (2022), 394–417. 11, 12

[PLDO25] PIERSON E., LI L., DAI A., OVSJANIKOV M.: DiffuMatch: Category-agnostic spectral diffusion priors for robust non-rigid shape matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2025), pp. 5745–5756. 15

[PRM*21] PAI G., REN J., MELZI S., WONKA P., OVSJANIKOV M.: Fast sinkhorn filters: Using matrix scaling for non-rigid shape correspondence with functional maps. In *IEEE Conference on Computer Vision and Pattern Recognition* (2021), Computer Vision Foundation / IEEE, pp. 384–393. 13

[PSS01] PRAUN E., SWELDENS W., SCHRÖDER P.: Consistent mesh parameterizations. In *Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH* (2001), ACM, pp. 179–184. 18

[QSMG17] QI C. R., SU H., MO K., GUIBAS L. J.: PointNet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition* (2017), IEEE Computer Society, pp. 77–85. 17, 20

[QYSG17] QI C. R., YI L., SU H., GUIBAS L. J.: PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems* (2017), pp. 5099–5108. 16, 20

[RAC*24] ROETZER P., ABBAS A., CAO D., BERNARD F., SWOBODA P.: DiscoMatch: Fast discrete optimisation for geometrically consistent 3d shape matching. In *European Conference on Computer Vision* (2024), vol. 15111 of *Lecture Notes in Computer Science*, Springer, pp. 443–460. 3, 8, 9

[RB24] ROETZER P., BERNARD F.: SpiderMatch: 3d shape matching with global optimality and geometric consistency. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024), IEEE, pp. 14543–14553. 8, 9

[RB25] ROETZER P., BERNARD F.: Fast globally optimal and geometrically consistent 3d shape matching. In *IEEE/CVF International Conference on Computer Vision* (2025). 8, 9

[RBW*14] RODOLÀ E., BULÒ S. R., WINDHEUSER T., VESTNER M., CREMERS D.: Dense non-rigid shape correspondence using random forests. In *IEEE Conference on Computer Vision and Pattern Recognition* (2014), IEEE Computer Society, pp. 4177–4184. 23

[RMOW20] REN J., MELZI S., OVSJANIKOV M., WONKA P.: MapTree: Recovering multiple solutions in the space of maps. *ACM Transactions on Graphics 39*, 6 (2020), 264:1–264:17. 13

[RMWO21] REN J., MELZI S., WONKA P., OVSJANIKOV M.: Discrete optimization for shape matching. *Computer Graphics Forum 40*, 5 (2021), 81–96. 14

[RSCB22] ROETZER P., SWOBODA P., CREMERS D., BERNARD F.: A scalable combinatorial solver for elastic geometrically consistent 3d shape matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), IEEE, pp. 428–438. 8, 9

[RSO19] ROUFOSSE J., SHARMA A., OVSJANIKOV M.: Unsupervised deep learning for structured shape matching. In *IEEE/CVF International Conference on Computer Vision* (2019), IEEE, pp. 1617–1627. 15

[Rus07] RUSTAMOV R. M.: Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Symposium on Geometry Processing* (2007), vol. 257 of *ACM International Conference Proceeding Series*, Eurographics Association, pp. 225–233. 20

[Rus11] RUSTAMOV R. M.: Multiscale biharmonic kernels. *Computer Graphics Forum 30*, 5 (2011), 1521–1531. 20

[SACO22] SHARP N., ATTAIKI S., CRANE K., OVSJANIKOV M.: DiffusionNet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics 41*, 3 (2022), 27:1–27:16. 16, 17, 20

[Sah18] SAHILLIOGLU Y.: A genetic isometric shape correspondence algorithm with adaptive sampling. *ACM Transactions on Graphics 37*, 5 (2018), 175. 8, 9

[Sah20] SAHILLIOGLU Y.: Recent advances in shape correspondence. *The Visual Computer 36*, 8 (2020), 1705–1721. 1

[SAPH04] SCHREINER J., ASIRVATHAM A., PRAUN E., HOPPE H.: Inter-surface mapping. *ACM Transactions on Graphics 23*, 3 (2004), 870–877. 10, 18

[SBCK19] SCHMIDT P., BORN J., CAMPEN M., KOBBELT L.: Distortion-minimizing injective maps between surfaces. *ACM Transactions on Graphics 38*, 6 (2019), 156:1–156:15. 9, 10, 20

[SCBK20] SCHMIDT P., CAMPEN M., BORN J., KOBBELT L.: Inter-surface maps via constant-curvature metrics. *ACM Transactions on Graphics 39*, 4 (2020), 119. 9, 10, 20

[SH23] SAHILLIOGLU Y., HORSMAN D.: Augmented paths and re-odesics for topologically-stable matching. *ACM Transactions on Graphics 42*, 2 (2023), 17:1–17:15. 19

[SHM*22] SHECHTER M., HANOCKA R., METZER G., GIRYES R., COHEN-OR D.: NeuralMLS: Geometry-aware control point deformation. In *Annual Conference of the European Association for Computer Graphics* (2022), Eurographics Association, pp. 65–68. 6

[SHVS11] SMEETS D., HERMANS J., VANDERMEULEN D., SUETENS P.: Dense shape correspondences using spectral high-order graph matching. In *IEEE Conference on Computer Vision and Pattern Recognition* (2011), IEEE Computer Society, pp. 1–8. 5

[SKS25] SUCU G., KALKAN S., SAHILLIOGLU Y.: FanNet: A mesh convolution operator for learning dense maps. *Computers & Graphics 132* (2025), 104320. 17

[SL23] SIDDIQI S., LÄHNER Z.: A network analysis for correspondence learning via linearly-embedded functions. In *German Conference on Pattern Recognition* (2023), vol. 14264 of *Lecture Notes in Computer Science*, Springer, pp. 100–114. 12

[SMJ*23] SUN M., MAO S., JIANG P., OVSJANIKOV M., HUANG R.: Spatially and spectrally consistent deep functional maps. In *IEEE/CVF International Conference on Computer Vision* (2023), IEEE, pp. 14451–14461. 15

[SMRO22] SUNDARARAMAN R., MARIN R., RODOLÀ E., OVSJANIKOV M.: Reduced representation of deformation fields for effective non-rigid shape matching. In *Advances in Neural Information Processing Systems* (2022). 6, 7

[SNB*12] SOLOMON J., NGUYEN A., BUTSCHER A., BEN-CHEN M., GUIBAS L. J.: Soft maps between surfaces. *Computer Graphics Forum 31*, 5 (2012), 1617–1626. 3

[SO20] SHARMA A., OVSJANIKOV M.: Weakly supervised deep functional maps for shape matching. In *Advances in Neural Information Processing Systems* (2020). 15, 16

[SOG09] SUN J., OVSJANIKOV M., GUIBAS L. J.: A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum 28*, 5 (2009), 1383–1392. 20

[SPK23] SCHMIDT P., PIEPER D., KOBBELT L.: Surface maps via adaptive triangulations. *Computer Graphics Forum 42*, 2 (2023), 103–117. 3, 9, 10

[SPKS16] SOLOMON J., PEYRÉ G., KIM V. G., SRA S.: Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics 35*, 4 (2016), 72:1–72:13. 3

[SS15] SMITH J., SCHAEFER S.: Bijective parameterization with free boundaries. *ACM Transactions on Graphics 34*, 4 (2015), 70:1–70:9. 20

[SWC*22] SALEH M., WU S., COSMO L., NAVAB N., BUSAM B., TOMBARI F.: Bending graphs: Hierarchical shape matching using gated optimal transport. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), IEEE, pp. 11747–11757. 7, 8

[SY12] SAHILLIOĞLU Y., YEMEZ Y.: Minimum-distortion isometric shape correspondence using em algorithm. *IEEE transactions on pattern analysis and machine intelligence 34*, 11 (2012), 2203–2215. 3

[SY14a] SAHILLIOĞLU Y., YEMEZ Y.: Multiple shape correspondence by dynamic programming. In *Computer Graphics Forum* (2014), vol. 33, pp. 121–130. 4

[SY14b] SAHILLIOGLU Y., YEMEZ Y.: Partial 3-d correspondence from shape extremities. *Computer Graphics Forum 33*, 6 (2014), 63–76. 3

[Tak22] TAKAYAMA K.: Compatible intrinsic triangulations. *ACM Transactions on Graphics 41*, 4 (2022), 57:1–57:12. 10

[TSS10] TOMBARI F., SALTI S., STEFANO L. D.: Unique signatures of histograms for local surface description. In *European Conference on Computer Vision* (2010), vol. 6313 of *Lecture Notes in Computer Science*, Springer, pp. 356–369. 20

[vKZHC11] VAN KAICK O., ZHANG H., HAMARNEH G., COHEN-OR D.: A survey on shape correspondence. *Computer Graphics Forum 30*, 6 (2011), 1681–1707. 1, 19

[VLB*17] VESTNER M., LÄHNER Z., BOYARSKI A., LITANY O., SLOSSBERG R., REMEZ T., RODOLÀ E., BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R., CREMERS D.: Efficient deformable shape correspondence via kernel matching. In *International Conference on 3D Vision* (2017), IEEE Computer Society, pp. 517–526. 3

[VLR*17] VESTNER M., LITMAN R., RODOLÀ E., BRONSTEIN A. M., CREMERS D.: Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *IEEE Conference on Computer Vision and Pattern Recognition* (2017), IEEE Computer Society, pp. 6681–6690. 3

[VM23] VIGANÒ G., MELZI S.: Adjoint Bijective ZoomOut: Efficient Upsampling for Learned Linearly-invariant Embedding. In *Smart Tools and Applications in Graphics* (2023), The Eurographics Association. 13

[VM24] VIGANÒ G., MELZI S.: Bijective upsampling and learned embedding for point clouds correspondences. *Computers & Graphics 122* (2024), 103985. 13

[VOM25] VIGANÒ G., OVSJANIKOV M., MELZI S.: NAM: Neural adjoint maps for refining shape correspondences. *ACM Transactions on Graphics 44*, 4 (2025), 60:1–60:15. 13

[VRM*17] VAROL G., ROMERO J., MARTIN X., MAHMOOD N., BLACK M. J., LAPTEV I., SCHMID C.: Learning from synthetic humans. In *IEEE Conference on Computer Vision and Pattern Recognition* (2017), IEEE Computer Society, pp. 4627–4635. 23

[WAK*20] WANG Y., AIGERMAN N., KIM V. G., CHAUDHURI S., SORKINE-HORNUNG O.: Neural cages for detail-preserving 3d deformations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), Computer Vision Foundation / IEEE, pp. 72–80. 6

[WSSC11] WINDHEUSER T., SCHLICKEWEI U., SCHMIDT F. R., CREMERS D.: Geometrically consistent elastic matching of 3d shapes: A linear programming solution. In *IEEE International Conference on Computer Vision* (2011), IEEE Computer Society, pp. 2134–2141. 3, 8, 9

[WY23] WU Y., YANG J.: Multi-part shape matching by simultaneous partial functional correspondence. *The Visual Computer 39*, 1 (2023), 393–412. 3

[XER*25] XIE Y., EHM V., ROETZER P., AMRANI N. E., GAO M., BERNARD F., CREMERS D.: EchoMatch: Partial-to-partial shape matching via correspondence reflection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2025), Computer Vision Foundation / IEEE, pp. 11665–11675. 15, 16, 17

[XLGM24] XIA Y., LU Y., GAO Y., MA J.: Locality preserving refinement for shape matching with functional maps. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence* (2024), AAAI Press, pp. 6207–6215. 13

[YHS*24] YANG H., HUANG X., SUN B., BAJAJ C. L., HUANG Q.: GenCorres: Consistent shape matching via coupled implicit-explicit shape generative models. In *International Conference on Learning Representations* (2024), OpenReview.net. 6, 7

[YLC*20] YANG L., LIU W., CUI Z., CHEN N., WANG W.: Mapping in a cycle: Sinkhorn regularized unsupervised learning for point cloud shapes. In *European Conference on Computer Vision* (2020), vol. 12355 of *Lecture Notes in Computer Science*, Springer, pp. 455–472. 17

[YS24] YAZGAN M., SAHILLIOGLU Y.: A partition based method for spectrum-preserving mesh simplification. *IEEE Transactions on Visualization and Computer Graphics 30*, 10 (2024), 6839–6850. 14

[YSC*23] YANG H., SUN B., CHEN L., PAVEL A., HUANG Q.: GeoLatent: A geometric approach to latent space design for deformable shape generators. *ACM Transactions on Graphics 42*, 6 (2023), 242:1–242:20. 6

[YW25] YIGIT O., WILSON R. C.: LBONet: Supervised spectral descriptors for shape analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 47*, 12 (2025), 12054–12068. 12

[YYN*25] YANG Y., YANG H., NAKAYAMA K., HUANG X., GUIBAS L. J., HUANG Q.: GenAnalysis: Joint shape analysis by learning man-made shape generators with deformation regularizations. *ACM Transactions on Graphics 44*, 4 (2025), 61:1–61:19. 6, 7

[YZZG24] YAN Y., ZHOU M., ZHANG D., GENG S.: Improved biharmonic kernel signature for 3d non-rigid shape matching and retrieval. *The Visual Computer 40*, 12 (2024), 8545–8571. 14

[ZGC25] ZENG H., GAO M., CREMERS D.: CoE: Deep coupled embedding for non-rigid point cloud correspondences. In *International Conference on 3D Vision* (2025), IEEE, pp. 286–295. 17

[ZKJB17] ZUFFI S., KANAZAWA A., JACOBS D. W., BLACK M. J.: 3d menagerie: Modeling the 3d shape and pose of animals. In *IEEE Conference on Computer Vision and Pattern Recognition* (2017), IEEE Computer Society, pp. 5524–5532. 23

[ZLG25] ZHURAVLEV A., LÄHNER Z., GOLYANIK V.: Denoising functional maps: Diffusion models for shape correspondence. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2025), Computer Vision Foundation / IEEE, pp. 26899–26909. 15, 16

[ZQZ*21] ZENG Y., QIAN Y., ZHU Z., HOU J., YUAN H., HE Y.: CorrNet3D: Unsupervised end-to-end learning of dense correspondence for 3d point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition* (2021), Computer Vision Foundation / IEEE, pp. 6052–6061. 6, 7

[ZWW*10] ZENG Y., WANG C., WANG Y., GU X., SAMARAS D., PARAGIOS N.: Dense non-rigid surface registration using high-order graph matching. In *IEEE Conference on Computer Vision and Pattern Recognition* (2010), IEEE Computer Society, pp. 382–389. 5

[ZYDL21] ZHENG Z., YU T., DAI Q., LIU Y.: Deep implicit templates for 3d shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition* (2021), Computer Vision Foundation / IEEE, pp. 1429–1439. 6, 7