Efficient Computation of Isometry-Invariant Distances Between Surfaces

Alexander M. Bronstein∗  Michael M. Bronstein∗  Ron Kimmel∗

Abstract

We present an efficient computational framework for isometry-invariant comparison of smooth surfaces. We formulate the Gromov-Hausdorff distance as a multidimensional scaling (MDS)-like continuous optimization problem. In order to construct an efficient optimization scheme, we develop a numerical tool for interpolating geodesic distances on a sampled surface from precomputed geodesic distances between the samples. For isometry-invariant comparison of surfaces in the case of partially missing data, we present the partial embedding distance, which is computed using a similar scheme. The main idea is finding a minimum-distortion mapping from one surface to another, while considering only relevant geodesic distances. We discuss numerical implementation issues and present experimental results that demonstrate its accuracy and efficiency.

∗Department of Computer Science, Technion – Israel Institute of Technology, Haifa 32000, Israel.
Table 1

Notation and symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{R} : \mathbb{R}^+$</td>
<td>Real numbers; non-negative real numbers</td>
</tr>
<tr>
<td>$\mathbb{R}^m$</td>
<td>$m$-dimensional Euclidean space</td>
</tr>
<tr>
<td>$\mathbb{R}^{M \times N}$</td>
<td>Space of $M \times N$ matrices</td>
</tr>
<tr>
<td>$\mathbb{M}$</td>
<td>Set of compact metric spaces</td>
</tr>
<tr>
<td>$(\mathcal{S}, d_\mathcal{S}) ; \mathcal{S}$</td>
<td>Metric space $\mathcal{S}$ with metric $d_\mathcal{S}$ arising from a Riemannian surface</td>
</tr>
<tr>
<td>$s ; \mathbf{s}$</td>
<td>a point on $\mathcal{S}$; extrinsic coordinates of $s$</td>
</tr>
<tr>
<td>$d_\mathcal{S}</td>
<td>_{\mathcal{S}'}$</td>
</tr>
<tr>
<td>$\mathcal{S}_N$</td>
<td>Finite sampling of $\mathcal{S}$ consisting of $N$ points</td>
</tr>
<tr>
<td>$\mathbf{D}_\mathcal{S}$</td>
<td>Matrix of geodesic distances on $\mathcal{S}_N$</td>
</tr>
<tr>
<td>$T_\mathcal{S} ; t^k$</td>
<td>Triangulation of $\mathcal{S}<em>N$; $k$-th triangle in $T</em>\mathcal{S}$</td>
</tr>
<tr>
<td>$\mathcal{S}^r ; \mathcal{S}^r_N$</td>
<td>$r$-covering; finite $r$-covering of $\mathcal{S}$</td>
</tr>
<tr>
<td>$B_{\mathcal{S}}(s_0,r)$</td>
<td>Open ball of radius $r$ in the metric space $\mathcal{S}$ around $s_0$</td>
</tr>
<tr>
<td>$\text{diam } \mathcal{S}$</td>
<td>Diameter of metric space $\mathcal{S}$</td>
</tr>
<tr>
<td>$\text{dis } \psi$</td>
<td>Distortion of the mapping $\psi$</td>
</tr>
<tr>
<td>$d_\mathcal{H}^\mathcal{Z}$</td>
<td>Hausdorff metric in the metric space $\mathcal{Z}$</td>
</tr>
<tr>
<td>$d_{GH}$</td>
<td>Gromov-Hausdorff metric between metric spaces</td>
</tr>
<tr>
<td>$d_P$</td>
<td>Permutation distance</td>
</tr>
<tr>
<td>$d_{PE}$</td>
<td>Partial embedding distance</td>
</tr>
<tr>
<td>$\Pi_N ; \pi$</td>
<td>Permutations of the set ${1, \ldots, N}$; a permutation in $\Pi_N$</td>
</tr>
<tr>
<td>$\mathbf{1}_{N \times N}$</td>
<td>$N \times N$ matrix of ones</td>
</tr>
</tbody>
</table>

1 Introduction

Many objects in nature can be described as non-rigid deformable (or bendable) surfaces. Often, the metric and the topological properties of the surface are preserved during bending. For example, most human organs bend without significantly stretching or tearing their surface. Study of living organs done through medical imaging often puts forth the challenge of comparing two deformable surfaces, which in turn requires identifying surface properties invariant to deformations. This paper addresses the problem of bending-invariant comparison between surfaces. More formally, we distinguish between the intrinsic and the extrinsic geometry of a surface. Extrinsic properties describe the way a surface is immersed into the ambient space. These properties are subject to changes while the surface bends. Intrinsic properties, on the other hand, are associated with the internal metric structure of the surface. Most natural deformations preserve, at least approximately, the intrinsic properties. Such deformations are called isometries. Our goal is therefore to define an efficiently-computable isometry-invariant measure of similarity between surfaces.

Going back to the human organs example, an interesting application for such measure of similarity is expression-invariant three-dimensional face recognition. In [6], we showed that facial expressions can be modeled as near-isometric deformations of the facial surface. Thus, an isometry-invariant similarity measure between faces would be a good candidate for face recognition insensitive to facial expressions.

1.1 Prior work

The case of Euclidean-isometry matching of surfaces was intensively explored. A classical result for rigid surface matching is the iterative closest point (ICP) method [27], finding a
Euclidean transformation between two discrete surfaces represented as point clouds.

A near-isometry-invariant comparison of surfaces based on Euclidean embeddings was presented by Elad and Kimmel in [12] as a generalization of [23]. The main idea is to map the intrinsic (metric) structure of the surfaces to a low-dimensional space and compare the resulting images in this space. The numerical core of the Elad-Kimmel approach is based on a multidimensional scaling (MDS) algorithm [3], whose numerical solutions can be computed efficiently. It is also related to dimensionality reduction [22, 11, 25] and to texture mapping [28, 16]. Similar problems are analyzed in graph theory in the context of representation of discrete metric spaces [19]. Yet, when dealing with abstract metric spaces like graphs, or even points clouds with local distances, smoothness of the underlying geometry can not be assumed, and thus, embedding is a much harder problem.

One of the caveats of the Euclidean embedding approach is the fact that usually a Riemannian surface cannot be perfectly represented in a finite-dimensional Euclidean space, thus inevitably introducing a metric distortion. It was argued and proved empirically in [5, 4] that using a two-dimensional or a three-dimensional sphere instead of the Euclidean space results in smaller embedding error for certain types of objects and consequently the recognition results get better. In [26], embedding into a space with hyperbolic geometry was shown.

In [15], Gromov introduced the Gromov-Hausdorff distance, which in a sense measures the metric distortion between metric spaces. Such a distance is invariant to isometries. In their recent fundamental paper [21], Mémoli and Sapiro adopted the Gromov-Hausdorff distance to the problem of deformable shape recognition. They considered a discrete setting in which surfaces are given as point clouds. In this case the computation of the Gromov-Hausdorff distance requires evaluating all the permutations between the samples. Such a combinatorial optimization problem appears to be computationally prohibitive. Mémoli and Sapiro showed a theoretical probabilistic framework and a practical procedure, which allowed them to formulate an approximation of the Gromov-Hausdorff distance between two point clouds without computing the permutations.

1.2 Main contribution

In many applications, like computer graphics, surfaces are represented as triangulated meshes that capture the local smoothness properties of the underlying surface. One contribution of our paper is the derivation of an MDS-like formulation for the Gromov-Hausdorff distance between surfaces. This results in a continuous non-convex optimization problem, that lends itself to a numerically exact computation of the Gromov-Hausdorff distance between surfaces.

In many practical cases, where surfaces are given only partially, the Gromov-Hausdorff distance cannot be used. For this purpose we introduced in [7] the partial embedding distance that can handle isometry-invariant partial surface matching. The key idea is to embed one surface into another and measure the minimal possible distortion of such a mapping. A numerical procedure devised in [7] for this purpose is similar in its nature to MDS and therefore has been named generalized MDS or GMDS. GMDS can be thought as a natural extension of previous works on isometric embedding into non-Euclidean spaces [26, 5, 4], as visualized in Figure 1. Here we provide the formal support for the framework presented in [7] and show the relationship between our partial embedding distance and the Gromov-Hausdorff distance.

Our paper comprises five sections. Section 2 provides the theoretical foundations of
FACIAL SURFACE 3D EUCLIDEAN EMBEDDING 3D SPHERICAL EMBEDDING PARTIAL EMBEDDING

Fig. 1. Evolution of isometry-invariant object recognition based on isometric embedding, exemplified on the problem of face recognition. Left to right: a facial surface; embedding into $\mathbb{R}^3$ [6]; embedding into $S^3$ (shown is the maximum-variance projection onto $\mathbb{R}^3$) [5]; embedding part of the face into another facial surface [7].

isometry-invariant surface matching. There, we define the Gromov-Hausdorff distance and review previous attempts for its approximation. We also describe the partial embedding distance and summarize its properties. In Section 3, we introduce an efficient algorithm for our isometry invariant distance computation. We address the problem of geodesic distance interpolation, a vital component of our numerical algorithm. We also discuss the question of GMDS convergence and show a multiresolution optimization approach as a means to overcome convergence to local minima. Section 4 presents an experimental validation of the proposed framework. Finally, Section 5 concludes the paper. Proofs and technical details can be found in the Appendices.

2 Theoretical background

The objects we deal with are surfaces. We use $\mathcal{S}$ to denote a surface, and assume it to be modeled as a smooth connected complete Riemannian 2-manifold. We denote by $d_\mathcal{S} : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}^+$, the geodesic distance function induced by the Riemannian metric. For simplicity, hereinafter we say “surface $\mathcal{S}$”, implying the underlying metric space $(\mathcal{S}, d_\mathcal{S})$. We assume, without loss of generality, that $\mathcal{S}$ is immersed into $\mathbb{R}^3$ and denote the extrinsic coordinates of a point $s \in \mathcal{S}$ by $s = (s^1, s^2, s^3)$.

In practice, we work with sampled surfaces represented by finite metric spaces. To this end, we use the following definition, a subset $\mathcal{S}^r \subset \mathcal{S}$ is called an $r$-covering of $\mathcal{S}$ (or an $r$-net in $\mathcal{S}$) if

$$\mathcal{S} = \bigcup_{s \in \mathcal{S}^r} B_\mathcal{S}(s, r).$$

Here $B_\mathcal{S}(s_0, r) = \{s \in \mathcal{S} : d_\mathcal{S}(s, s_0) < r\}$ is an open ball of radius $r$ around $s$ in $\mathcal{S}$. A finite $r$-covering of $\mathcal{S}$ consisting of $N$ points is denoted by $\mathcal{S}^r_N$. The space $(\mathcal{S}^r_N, d_\mathcal{S}|_{\mathcal{S}^r_N})$ constitutes a finite metric space, where the metric $d_\mathcal{S}|_{\mathcal{S}^r_N}$ is obtained by restricting the original metric $d_\mathcal{S}$ to the subset $\mathcal{S}^r_N$. An arbitrary finite sampling of $\mathcal{S}$ consisting of $N$ points (a point cloud) is denoted by $\mathcal{S}_N = \{s_1, \ldots, s_N\}$. In order to make a distinction between the original and the sampled surfaces, we refer to $\mathcal{S}$ as to a continuous surface and to $\mathcal{S}_N$ as to a discrete surface.
A sampled cloud can be represented as a cloud of points \( S_N \) and an \( N \times N \) matrix \( D_S = (d_{S_{ij}}(s_i, s_j)) \) of the corresponding geodesic distances. In addition, we assume that a triangulation \( T_S \) capturing the local geometry of the surface is given (here \( T_S = \{(t_{11}, t_{12}, t_{13}), \ldots, (t_{K_1}, t_{K_2}, t_{K_3})\} \) is a list of \( K \) triangles, where each triangle is a triplet of sample indices, \( t_{ki} \in \{1, \ldots, N\} \)). The resulting triangulated mesh can be thought of as a polyhedral approximation of the smooth surface \( S \).

Surface \( S \) is called bound if \( \text{diam} S = \text{sup}_{s, s' \in S} d_S(s, s') \) is finite. \( S \) is said to be compact if all its open covers have a finite subcover. If \( S \) is compact, it is necessarily bounded. This will be our tacit assumption, as surfaces encountered in practical applications are compact.

We use \( \mathbb{M} \) to denote the set of all such surfaces.

Given two surfaces \( S \) and \( Q \) in \( \mathbb{M} \), a transformation \( \varphi : S \mapsto Q \) is said to have a distortion or distort the metric by

\[
(2) \quad \text{dis} \varphi = \sup_{s, s' \in S} |d_S(s, s') - d_Q(\varphi(s), \varphi(s'))|.
\]

If in addition for all \( q \in Q \) \( d_Q(q, \varphi(S)) \leq \epsilon \) (we call this property \( \epsilon \)-surjectivity), \( \varphi \) is called an \( \epsilon \)-isometry and the spaces \( S \) and \( Q \) are termed as \( \epsilon \)-isometric. The image of \( S \), \( \varphi(S) \) is called an \( \epsilon \)-Hausdorff approximation of \( S \) in \( Q \). A transformation \( \varphi : S \mapsto Q \) with \( \text{dis} \varphi = 0 \) is called an isometry, and \( S \) and \( Q \) admitting such a transformation are called isometric. Unlike \( \epsilon \)-isometries, an isometry is always bijective and bi-Lipschitz continuous.

In this paper, we address the question of matching between two surfaces, insensitively to their isometries. More formally, we study a distance function \( d : \mathbb{M} \times \mathbb{M} \mapsto \mathbb{R}^+ \), which assigns a non-negative number to every pair of surfaces, measuring the degree of their similarity. The basic property we require is that for all isometries \( f \) of \( S \) and \( g \) of \( Q \),

\[
d(f(S), g(Q)) = d(S, Q).
\]

### 2.1 Gromov-Hausdorff distance

Mémoli and Sapiro in [21], introduced the Gromov-Hausdorff distance into the deformable surface matching arena. Here we use this powerful theory and present numerical methods for its consistent computation subject to the disclaimer that we can not guarantee to completely avoid the intrinsic local minima of the associated measure. Our starting point is the classical Hausdorff distance. Assume that the surfaces \( S \) and \( Q \) are subsets of some larger metric space \( (\mathcal{Z}, d_{\mathcal{Z}}) \). Then, the Hausdorff distance can be defined in the following way

\[
(3) \quad d_H^2(S, Q) = \max \left\{ \sup_{s \in S} d_{\mathcal{Z}}(s, Q), \sup_{q \in Q} d_{\mathcal{Z}}(q, S) \right\}.
\]

Here for notation simplicity we use \( d_{\mathcal{Z}}(s, Q) = \inf_{q \in Q} d_{\mathcal{Z}}(s, q) \), implying a point-to-set distance. It can be shown that \( d_H^2 \) is a metric on all the subsets of \( \mathcal{Z} \). Versions of the Hausdorff distance are often used to measure similarity between rigid surfaces in \( \mathbb{R}^3 \), which give birth to a family of algorithms known as iterative closest point or ICP [2, 27]. Yet, while being a good match for comparison of extrinsic geometries, the Hausdorff distance is not isometry-invariant.

In order to overcome this problem, Mémoli and Sapiro [21] adopted an extension of the Hausdorff distance introduced by Gromov [15]. The Gromov-Hausdorff distance is defined
by

$$d_{GH}(S, Q) \equiv \inf_{\rho: S \rightarrow Z, \sigma: Q \rightarrow Z} \frac{d_H^2(\rho(S), \sigma(Q))}{\rho(S)} ,$$

where $\rho : S \mapsto Z$ and $\sigma : Q \mapsto Z$ are isometric embeddings into the metric space $Z$. The Gromov-Hausdorff distance is invariant to isometries and thus allows to compare between two surfaces in a truly intrinsically-geometric way. In addition, $d_{GH}$ has several nice properties, which we summarize here.

1. **Metric properties:** $d_{GH}$ is a finite metric on the space of compact surfaces $M$, in which a point is defined as an equivalence class of a surface and its isometries [14, 10]. In other words, $d_{GH}$ satisfies the following axioms: (i) $d_{GH}(S, Q) \geq 0$; (ii) $d_{GH}(S, Q) = 0$ if and only if $S$ and $Q$ are isometric; (iii) $d_{GH}(S, Q) = d_{GH}(Q, S)$; and (iv) $d_{GH}(R, Q) \leq d_{GH}(R, S) + d_{GH}(S, Q)$, also known as the triangle inequality.

2. **Near-isometric similarity:** (i) If $d_{GH}(S, Q) \leq \epsilon$, then there exists a $2\epsilon$-Hausdorff approximation of $S$ in $Q$; (ii) if there exists an $\epsilon$-Hausdorff approximation of $S$ in $Q$, then $d_{GH}(S, Q) \leq 2\epsilon$. This property can be thought of as an extension of the second metric axiom to $\epsilon$-isometries (see [10] for details).

3. **Sampling consistency:** Given $S^r$ an $r$-covering of $S$ and $Q^{r'}$ and $r'$-covering of $Q$, $\left| d_{GH}(S, Q) - d_{GH}(S^r, Q^{r'}) \right| \leq r + r'$. In other words, the Gromov-Hausdorff distance between sampled versions of two surfaces is close to the Gromov-Hausdorff distance between the original ones, and the discrepancy vanishes as the sample radius decreases.

The main disadvantage of $d_{GH}(S, Q)$ according to definition (4) is its computational intractability.

Fortunately, for bounded metric spaces, the Gromov-Hausdorff distance can be translated to comparison of distances within $S$ and $Q$ to each other, thus avoiding the cumbersome construction of the metric space $Z$. Formally, the Gromov-Hausdorff distance can be redefined as follows [17, 21]

$$d_{GH}(S, Q) = \frac{1}{2} \inf_{\varphi: S \rightarrow Q, \psi: Q \rightarrow S} \max \{ \text{dis} \varphi, \text{dis} \psi, \text{dis}(\varphi, \psi) \} ,$$

where $\text{dis} \varphi$ and $\text{dis} \psi$ defined in (2) measure the metric distortion of the mappings $\varphi$ and $\psi$, respectively, and

$$\text{dis}(\varphi, \psi) \equiv \sup_{s \in S} \left| d_S(s, \psi(q)) - d_Q(q, \varphi(s)) \right|$$

measures how far are $\varphi$ and $\psi$ from being one the inverse of the other. This reformulation, as we will show, allows us an efficient computation of the Gromov-Hausdorff distance between surfaces.
2.2 Méromi-Sapiro approximation of $d_{GH}$

In [21], Méromi and Sapiro considered the approximation of the Gromov-Hausdorff distance between point clouds by dropping the term $\text{dis}(\varphi, \psi)$ in the definition of $d_{GH}$ in (5) and bounding its contribution probabilistically. Given two bounded surfaces $S$ and $Q$, both sampled at $N$ points ($S_N$ and $Q_N$, respectively), they defined the permutation distance

$$d_P(S_N, Q_N) \equiv \frac{1}{2} \min_{\Pi_N} \max_{1 \leq i, j \leq N} |d_S(s_i, s_j) - d_Q(q_{\pi_i}, q_{\pi_j})|,$$

where $\Pi_N$ denotes all the permutations of $\{1, \ldots, N\}$. $d_P$ is not the exact Gromov-Hausdorff distance but a different measure of surface similarity related to $d_{GH}$. Obviously, $d_{GH}(S_N, Q_N) \leq d_P(S_N, Q_N)$, since $d_P$ can be considered a particular discrete case of $d_{GH}$, in which the correspondences between the two metric spaces are restricted only to the bijective ones. The upper bound is much more complicated and is derived in a probabilistic setting [21]. Based on these results, Méromi and Sapiro described an algorithm for the approximation of $d_{GH}$. The key idea is to select subsamplings of the given point clouds, the distance between which can be related to the Gromov-Hausdorff distance. Alternative, more practical, estimation methods of the distance between point-clouds, motivated by the Gromov-Hausdorff distance, were also presented in [21].

Méromi and Sapiro also noted that restricting the $Z$ space in definition (4) of $d_{GH}$, the metric actually boils down to the canonical form distance proposed in [12].

2.3 Partial embedding distance

The main drawback of the Gromov-Hausdorff distance and its approximate version is the fact that it compares two metric spaces as a whole, which is inappropriate for partial matching. One of the most significant differences of partial matching compared to its “full” counterpart is the lack of symmetry. We therefore address the problem of finding a match between a model surface $S$ and a probe $Q$, which is a deformed version of the model. In practice, one is often required to find whether a patch $Q' \subset Q$ is similar to $S$.

In [7] we defined the non-symmetric partial embedding distance as

$$d_{PE}(S, Q') \equiv \frac{1}{2} \inf_{\psi: S \to Q'} \sup_{(q, q') \in P} |d_Q(q, q') - d_S(\psi(q), \psi(q'))|,$$

where $P = \{(q, q') : d_{Q'}(q, q') = d_Q(q, q')\} \subseteq Q \times Q$. In other words, we define the distortion only on pairs of points for which the metric on the patch $Q'$ is consistent with that on $Q$. Such a restriction is necessary because the assumption $d_{Q'} = d_Q|_{Q'}$ (i.e. that $d_{Q'}(q, q') = d_Q(q, q')$ for all $q, q' \in Q'$) holds only in particular cases, e.g. when $Q'$ is a convex subset of the surface $Q$. In general, minimal geodesics on $Q$ that cross the boundary $\partial Q'$ change in $Q'$ and therefore, the geodesic distances between the corresponding points in $Q$ and $Q'$ may differ. Note that when $P = Q \times Q$, we have $\sup_{(q, q') \in P} |d_Q(q, q') - d_S(\psi(q), \psi(q'))| = \text{dis} \psi$ and the partial embedding distance becomes

$$d_{PE}(S, Q) = \frac{1}{2} \inf_{\psi: S \to Q} \text{dis} \psi.$$

A comment about the potential usefulness of (8) for partial matching was made in [21].

The properties of $d_{PE}$ resemble those of $d_{GH}$. We summarize them here; for proof, see Appendix A.
1. Some metric properties: \( d_{PE} \) satisfies some of the metric axioms: (i) \( d_{PE}(S, Q) \geq 0 \); (ii) \( d_{PE}(S, Q) = 0 \) if and only if \( Q \) is isometrically embeddable into \( S \); (iii) \( d_{PE}(S, Q) \leq d_{PE}(S, R) + d_{PE}(R, Q) \), which is a non-symmetric version of the triangle inequality.

2. Near-isometric similarity: (i) If there exists an \( \epsilon \)-Hausdorff approximation of \( S \) in \( Q \), then \( d_{PE}(S, Q) \leq 2\epsilon \). (ii) The converse appears to be true under some more restrictive assumptions (the proof is to be published elsewhere).

3. Sampling consistency: Given \( S' \) an \( r \)-covering of \( S \) and \( Q' \) an \( r' \)-covering of \( Q \),
\[
d_{PE}(S, Q) - r' \leq d_{PE}(S', Q') \leq d_{PE}(S, Q) + r.
\]

4. Partial matching: Given \( Q' \) an \( r' \)-covering of \( Q \), \( d_{PE}(Q', Q) = 0 \).

First, we see that though not a metric in the strict sense, \( d_{PE} \) satisfies some important properties of a metric: isometry invariance and an asymmetric version of the triangle inequality. Secondly, by virtue of Property 4 (unlike to the corresponding property \( d_{GH}(Q, Q') \leq r \) of the Gromov-Hausdorff distance, as mentioned above) \( d_{PE} \) can be used for partial matching. Thirdly, note that a particular case of Property 3 is \( d_{PE}(S, Q) - r \leq d_{PE}(S, Q') \leq d_{PE}(S, Q) \). When \( r \) is too large, the lower bound makes little sense, and one can obtain \( d_{PE}(S, Q') \approx 0 \) while \( d_{PE}(S, Q) \) is large. This reveals a potential danger in partial matching: when the patch \( Q' \) is too small, such a comparison is liable to be meaningless, since a sufficiently small patch can be embedded with low metric distortion into every surface.

3 Efficient computation of the partial embedding and the Gromov-Hausdorff distances

We now address a practical question of efficiently computing the Gromov-Hausdorff and the partial embedding distances between two polyhedral surfaces. For clarity, we start from computation of the simpler \( d_{PE}(S, Q) \), of which the computation of \( d_{GH}(S, Q) \) will be a straightforward generalization. As an input, let us be given

1. A discretization of the continuous surface \( Q \), defined by the point cloud \( Q^r_M = \{q_i\}^M_{i=1} \).

2. A polyhedral approximation of the continuous surface \( S \) defined by the point cloud \( S^r_N = \{s_i\}^N_{i=1} \) and a triangulation \( T_S = \{(m_{11}, m_{12}, m_{13}), \ldots, (m_{N_T,1}, m_{N_T,2}, m_{N_T,3})\} \).

   In our notation, the polyhedron \( S'_N \) consists of \( N_T \) triangular faces, where the \( t \)-th face is described by the vertices \( s_{m_{11}}, s_{m_{12}}, s_{m_{13}} \).

3. Two distance matrices \( D_S = (d_S(s_i, s_j)) \) and \( D_Q = (d_Q(q_i, q_j)) \), representing the geodesic distances between the samples of \( S \) and \( Q \), respectively. In practice, only an approximation of these distances can be computed numerically, e.g. using the Fast Marching method [24].

Using the triangulated surface \( S'_N \) as a polyhedral \( r \)-approximation of \( S \), we parameterize the points on \( S \) in local barycentric coordinates derived from the triangulation. Let \( s \) be a point on \( S'_N \) belonging to some triangle \( t \in \{1, \ldots, N_T\} \); the Euclidean coordinates \( s \) of \( s \) can be represented as a convex combination of the triangle vertices \( s_{m_{11}}, s_{m_{12}}, s_{m_{13}} \),

\[
s = \sum_{i=1}^{3} s_{m_{1i}} u_i.
\]
Hence, every point on $S_N^r$ can be identified with a vector of barycentric coordinates $(t, u) = (t, u_1, u_2, u_3)$, where $u_i \geq 0$, $u_1 + u_2 + u_3 = 1$. We will freely switch between $s$ and its barycentric representation.

The partial embedding distance between $S_N^r$ and $Q_M^r$ can be formulated as the following constrained minimization problem

$$\min_{s', \epsilon \geq 0} \epsilon \quad \text{s.t.} \quad |d_S(s'_i, s'_j) - d_Q(q_i, q_j)| \leq \epsilon$$

where $(i, j) \in P \subseteq \{1, ..., M\} \times \{1, ..., M\}$ (for the time being we assume $P$ to be given; a way to find it is shown in Section 3.4). Once the minimal $\epsilon$ is found, our approximation to the partial embedding distance is $d_{PE}(S_N^r, Q_M^r) = \epsilon/2$.

Note that we use the images $s'_i = \psi(q_i)$ (in their barycentric coordinates) directly as the optimization variables, which frees us from the need to optimize over all possible permutations of $s_i$ and $q_j$. In fact, the problem above is a modest-sized constrained minimization problem with $M + 1$ variables and at most $\frac{1}{2}M(M - 1)$ inequality constraints. Also note that the points $s_i$ are given, and so are the distances $d_Q(q_i, q_j)$, which are the elements of the matrix $D_Q$. On the other hand, the distances $d_S(s'_i, s'_j)$ have to be approximated numerically (see Section 3.3); in the meantime, we assume them to be computable exactly. The reason is that the images $s'_i$ do not necessarily coincide with the vertices of $S_N^r$, respectively, and can fall inside any triangle on the latter polyhedron.

Theoretically, our computation is accurate up to the sampling radius, i.e. $|d_{PE}(S_N^r, Q_M^r) - d_{PE}(S, Q)| \leq 2r$. If in addition we take into account that the distances $d_S$ and $d_Q$ are computed with accuracy $\delta$, we have $|d_{PE}(S_N^r, Q_M^r) - d_{PE}(S, Q)| \leq 2(r + \delta)$. In practice, a finer grid is used to approximate the distances, therefore, $\delta \ll r$. Sampling the surfaces sufficiently densely ($r$ sufficiently small), we can compute $d_{PE}$ with any desired accuracy. In other words, our approach is numerically consistent.\(^1\)

The problem (10) can be viewed as a minimization of maximum distortion, or a min-max ($\ell_\infty$-norm) problem. Practical considerations may favor the choice of norms other than $\ell_\infty$. In order to handle them numerically, we abandon the constrained minimization formulation with the artificial variable $\epsilon$, and minimize

$$F(s'_1, \ldots, s'_M) = \sum_{i,j} w_{ij} \eta \left( d_S(s'_i, s'_j) - d_Q(q_i, q_j) \right)$$

with respect to $s'_i$, where $w_{ij} = 1$ for $(i, j) \in P$ and zero otherwise are weights representing the set of “consistent pairs” $P$, and $\eta(x)$ is some non-linear function (e.g., the choices $\eta(x) = x^2$ and $\eta(x) = |x|$ yield the $\ell_2$ and the $\ell_1$ norms, respectively). Note the resemblance of $F$ to the stress function used in the MDS problems.

Though the properties of $\text{argmin}_F F$ are dramatically different from those of $d_{PE}$, it is a good measure of surface similarity with most of the important properties of the true partial embedding distance, yet less susceptible to noise and numerical errors. We will henceforth denote the $\ell_p$ version of $d_{PE}$ as $d_{PE}^p$.

\(^1\)Note that since our optimization problem is non-convex, most practical optimization algorithm do not guarantee convergence to the global minimum. Yet, in Section 3.2 we show that, in practice, local convergence can be avoided using multi-resolution methods.
The Gromov-Hausdorff distance is computed essentially in the same way as $d_{PE}$, by solving

$$
\min_{s_i', d_k, \epsilon \geq 0} \epsilon \quad \text{s.t.} \quad \left| d_Q(q_i', q_j') - d_S(s_i, s_j) \right| \leq \epsilon
$$

(12)

$$
\left| d_S(s_k', s_l') - d_Q(q_k, q_l) \right| \leq \epsilon
$$

$$
\left| d_S(s_i, s_k') - d_Q(q_k, q_i') \right| \leq \epsilon
$$

where $j > i, l > k$. Here two additional terms standing for $\text{dis} \, \varphi$ and $\text{dis}(\varphi, \psi)$ are added. As in the case of $d_{PE}$, it might be advantageous to replace the $\ell_\infty$ norm with some more robust non-linearity, yielding the stress

$$
F = \sum_{i,j=1}^{N} \eta \left( d_Q(q_i', q_j') - d_S(s_i, s_j) \right) + \sum_{k,l=1}^{M} \eta \left( d_S(s_k', s_l') - d_Q(q_k, q_l) \right)
$$

$$
+ \sum_{i,j} \eta \left( d_S(s_i, s_k') - d_Q(q_k, q_i') \right).
$$

(13)

### 3.1 Iterative minimization algorithm

We now present a practical numerical algorithm for computation of $d_{PE}$. For clarity, we will henceforth consider only the unconstrained minimization of (11); in order to handle the constrained problem (12), it is first converted to an unconstrained one using the penalty-barrier method [1], which is then solved in an essentially similar way. The same technique can be used to compute the Gromov-Hausdorff distance.

Our goal is to bring $F(s_1', ..., s_M')$ to a (possibly local) minimum over $s_i' = (t_i, u_i)$ by starting with some initial guess $s_i' (0) = (t_i (0), u_i (0))$ of the points and proceeding by iteratively updating their locations producing a decreasing sequence of function values. Let $s_i' (k) = (t_i (k), u_i (k))$ be the optimization variables at $k$th iteration and let there be a set of directions $d_i (k)$ such that displacement of $u_i (k)$ along them by some step size $\alpha (k)$ decreases the value of $F$ (for convenience, we omit the iteration index $k$ wherever possible). The simplest way to select the directions is $d_i = -\nabla_u F$, which yields a family of algorithms usually termed as gradient or steepest descent. There exist a variety of more efficient strategies to select $d_i$, including conjugate gradients and quasi-Newton methods just to name a few [1].

The step size $\alpha$ has to be chosen to guarantee a sufficient decrease of $F$. When constant step is used, there is generally a tradeoff between too small steps, which result in slow convergence and too large steps, which are liable to increase the value of $F$. To provide guaranteed decrease of $F$, we adaptively select the step size at every iteration using the Armijo rule, which first sets $\alpha = \alpha_0$ and then successively reduces it by some factor $\beta \in (0, 1)$ until

$$
F(s_1', ..., s_M') - F(s_1'', ..., s_M'') \geq -\sigma \alpha \sum_i d_i^T \nabla_u F(s_1', ..., s_M'),
$$

where $\sigma \in (0, 1)$ and $s_i''$ represent the points $s_i'$ displaced by $\alpha$ along $d_i$. In our implementation, we used $\sigma = 0.3, \beta = 0.5$. The initial value $\alpha_0$ is first selected to be large, and gradually refined at each iteration. A similar rule can be applied when the update is performed for a single point per iteration, yielding a block-coordinate descent algorithm.
Fig. 2. Displacement of the point \( u \) in the triangle \( t = ABC \) along the direction \( d \). From left to right: the point is displaced along \( d \) until the triangle edge \( BC \) is reached; the adjacent triangle \( t' = BCD \) is rotated about \( BC \) and the direction vector \( d \) is transformed to the barycentric coordinates of \( t' \), \( d' \); the final point \( u' \) is distant \( \alpha \) from \( u \) along a poly-linear path.

Note that unlike common optimization problems, here in order to displace \( s'_i \) one cannot simply add \( \alpha d_i \) to \( u_i \), since the latter might leave the triangle \( t_i \), thus invalidating the barycentric representation. Instead, one the displacement is performed by following a poly-linear path starting at \( u_i \), propagating along a straight line in the direction \( d_i \) until the first intersection with the triangle boundary, then proceeding along a line inside the triangle adjacent to the intersected edge, and so on until the total length of the path is \( \alpha \) (see Figure 2).

The computation is carried out by the following algorithm, which accepts an initial point \( s = (t, u) \), a direction \( d \) and a step size \( \alpha \) and returns the displaced point \( s' = (t', u') \):

1. Compute \( u' = u + \alpha d \). If \( u' \) is inside the triangle \( t \), return \( s' = (t, u') \) and stop.

2. Find \( \alpha' < \alpha \), for which \( u' = u + \alpha' d \) lies on the boundary of \( t \).

3. Otherwise, if there is no triangle adjacent to the edge on which \( u' \) lies, we can proceed no more, since the shape boundary is reached. Return \( s' = (t, u') \) and stop.

4. Otherwise, let \( t' \) be a triangle sharing the edge on which \( u' \) lies. Transform \( u' \) and \( d \) to the barycentric coordinates of \( t' \), replace \( \alpha \) with \( \alpha - \alpha' \) and \( (t, u) \leftarrow (t', u') \), and go to Step 1.

The transformation of \( u' \) from barycentric coordinates in \( t \) to those in \( t' \) carried out in Step 4 is trivial, since \( u' \) is located on the edge shared by the two triangles and thus has the same barycentric representation in \( t' \) up to vertex numbering. In order to transform \( d \), we rotate the triangle \( t' \) about the edge shared with \( t \) so that both triangles lie in the same plane, and find such \( d' \) in the coordinate system of \( t' \), which defines the same planar direction as \( d \) in \( t \).

Since at each iteration the Armijo rule produces a decreasing sequence of step sizes, it is sufficient to compute the displacement of \( s_i^{(k)} \) by \( \alpha_0 d_i \) and cache the entire path; this allows to save path computations for subsequent smaller steps.
3.2 Multi-resolution minimization

GMDS, similarly to MDS algorithms in general [3], is liable to local convergence, since the objective function involved in the computation of \( \hat{d}_{GH} \) and \( \hat{d}_{PE} \) is non-convex. Here we propose multi-resolution optimization schemes as a practical remedy for this problem, in light of the results obtained in [9] for traditional MDS. An additional advantage of multi-resolution optimization is a significant speedup of convergence.

The key idea of a multi-resolution optimization scheme is to work with a hierarchy of problems, starting from a coarse version of the problem containing a small number of variables (points). The coarse level solution is interpolated to the next resolution level, and is used as an initialization for the optimization at that level. The process is repeated until the finest level solution is obtained. Such a multi-resolution scheme can be thought of as a smart way of initializing the optimization problem. Small local minima tend to disappear at coarse resolution levels, thus reducing the risk of local convergence which is more probable when working at a single resolution.

The main components of the multi-resolution scheme are the hierarchy of data which defines optimization problems at different resolution levels and the interpolation procedure, which allows to pass from coarse level to a finer one. Formally, let us denote by \( Q_{M_1} \subset Q_{M_2} \subset \ldots \subset Q_{M_L} = Q_M \) an \( L \)-level hierarchy of data. The points on the \((l+1)\)st resolution level are obtained by removing part of the points on the \(l\)th level. The corresponding distance matrices \( D_1, \ldots, D_L = D_Q \) are created as sub-matrices of \( D_Q \).

One possibility to construct such a hierarchy is the farthest point sampling (FPS) strategy [13]. As the coarsest resolution level \( Q_{M_1} \), we select \( M_1 \) points. In practical problems, a few landmark points can be selected on the surface (e.g. in the face recognition problem, it is usually easy to roughly locate points such as the nose and the eyes); otherwise, these points can be chosen randomly. At the next resolution level, we add points in the following manner: \( q_{M_l+1} \) is selected as the most distant point from \( Q_{M_1} \), and so on. \( q_{M_l+k} = \arg\min_{q_i \in Q_M} d_Q(q_i, \{q_1, \ldots, q_{M_l+k-1}\}) \). Taking the first \( M_l \) points from the sequence produced in this manner, we obtain \( Q_{M_l} \) (see example in Figure 3).

Let us assume that at the \( l \)th resolution level, \( Q_{M_l} = \{q_1, \ldots, q_{M_l}\} \) were embedded into \( S_N \) using the iterative minimization algorithm outlined in Section 3.1. As the result, the set of images \( \psi(Q_{M_l}) = \{s_1', \ldots, s_{M_l}'\} \) on the polyhedron \( S_N \) was obtained. At the next
resolution level, we have to embed a larger set \( Q_{M+1} \) into \( S_N \), solving the minimization problem for \( \psi(Q_{M+1}) = \{s'_1, ..., s'_{M+1}\} \). The initialization for the first \( M \) points is readily available from the solution at the previous level. The initial locations for the remaining points \( s'_i \) for \( i = M+1, ..., M+1 \) have to be interpolated.

It is reasonable to initialize \( s'_i \) with such a point on \( S_N \) that the geodesic distances from it to the points \( s'_1, ..., s'_{M} \) are as close as possible to the geodesic distances from \( q_i \) to \( q_1, ..., q_{M} \). Formally, \( s'_i \) can be expressed as

\[
(15) \quad s'_i = \arg \min_s \sum_{j=1}^{M} (d_S(s, s'_j) - d_Q(q_i, q_j))^2.
\]

Note that practically the minimum can be found by exhaustively searching over all \( s_1, ..., s_N \) or even a coarser subset of \( S_N \). The complexity of such a search is negligible compared to the complexity of the iterative minimization process.

### 3.3 Interpolation of geodesic distances

So far, we considered the distance terms in (10) readily available, though it is obvious that only the distances between the surfaces samples, namely \( d_S(s_i, s_j) \) are available, whereas distances between two arbitrary points in \( S_N \) have to be computed. Let us now devise a numerical procedure for computation of the geodesic distances between a pair of points on the polyhedron \( S_N \).

As an input, we are given

1. Euclidean coordinates \( \{s_i\}_{i=1}^N \) of the polyhedron vertices \( s_i \).
2. The distance matrix \( D_i \), with elements \( d_{ij} = d_S(s_i, s_j) \).
3. Two arbitrary points \( s = (t, u) \), \( s' = (t', u') \) in barycentric coordinates.

The output of our numeric procedure is the approximate geodesic distance \( \hat{d}_S(s, s') \approx d_S(s, s') \).

Although the geodesic distance between two arbitrary points \( s, s' \) is generally unavailable, each point lies inside a triangle with vertices on the grid. Let \( s_1, s_2, s_3 \) and \( s'_1, s'_2, s'_3 \) be the vertices of the triangles \( t \) and \( t' \) enclosing \( s \) and \( s' \), respectively. Note that the mutual geodesic distances \( d_S(s_i, s'_j) \) are known. A naïve approach is to use the average of the above distances as \( \hat{d}_S(s, s') \), i.e.

\[
(16) \quad \hat{d}_S(s, s') = \frac{1}{3} \sum_{i=1}^{3} d_S(s_i, s'_i).
\]

By the triangle inequality, the approximation error has the order of the radius of the largest triangle, that is, \( O(r) \). In order to make the interpolant \( \hat{d}_S(s, s') \) smooth as a function \( u \) and \( u' \), the average is replaced by a weighted sum, where the weights are chosen according to proximity of \( u \) and \( u' \) to each of the enclosing triangle vertices. Such an interpolation procedure has very low computational complexity; yet, its main disadvantage is the fact that it ignores the underlying geometry. Here we develop an alternative geometric interpolation approach, which is a bit more elaborate, yet provides significantly more accurate approximation of the geodesic distances.

The basic idea of our interpolator is to compute first an approximation to the geodesic distances \( d_S(s, s'_i) \) between the point \( s \) and each of the three vertices \( s'_i \) of \( t' \). Computation
of each of these distances can be regarded as the problem of finding an approximation to \(d_S(s, s')\) given \(d_S(s_i, s_i')\). We handle the solution of this problem by a numerical procedure that we named the three-point geodesic distance approximation, which is detailed in Appendix B. Once the approximate \(d_S(s, s')\) are available, we invoke again the three-point geodesic distance approximation, this time to compute the desired distance \(d_S(s, s')\) from \(d_S(s_i, s_i')\).

### 3.4 Selection of weight for partial matching

So far, we have assumed that the set \(P\) of the consistent pairs of points in (10) or, alternatively, the weights \(w_{ij}\) in (11) are given. Here we show a way to find it in practice. Let \(S\) be the model surface and let \(Q\) be the probe surface. We denote by \(Q'\) a patch of \(Q\) and assume that its sampled version is \(Q'_M\). The question is how to find the indices \((i, j) \in \{1, ..., M'\} \times \{1, ..., M'\}\) of pairs of points for which \(d_{Q'_M}(q_i, q_j) \neq d_Q(q_i, q_j)\), and thus must be excluded from \(P\).

If a sampled version \(Q_M\) of \(Q\) is given (this is a common situation in face recognition, where one has the full probe surface but wishes to deliberately remove some parts of it [7]), the solution is straightforward: we remove the pairs of points \((i, j)\) for which \(|d_{Q'_M}(q_i, q_j) - d_{Q_M}(q_i, q_j)| < \delta\), where \(\delta\) is some small threshold, related to distance approximation accuracy. In case when only the patch \(Q'\) is available, we note that the minimal geodesics may change due to the change in the boundary of \(Q\). We therefore remove pairs of points \((i, j)\) for which \(d_{Q'}(q_i, \partial Q') + d_{Q'}(q_j, \partial Q') < d_{Q'}(q_i, q_j)\). Here \(\partial Q'\) denotes the boundary of \(Q'\). Since in practice a sampled version \(Q'_M\), is given, this criterion must be discretized, and the distances approximated numerically.

### 4 Numerical results

We now present numerical experiments that assess the validity of our theoretical and computational framework. In the first experiment, we show the performance of the partial embedding distance in matching between spheres with different radii. In the second experiment, matching of articulated objects is performed using the Gromov-Hausdorff distance. The Elad-Kimmel canonical form distance [12] is used as the baseline reference in the two experiments. In the third experiment, we show matching of partially missing objects using the partial embedding distance. Additional numerical evaluations of the partial embedding distance can be found in [7, 8].

#### 4.1 Partial embedding distance between spherical patches

In the first experiment, we measured the distance between a unit two-dimensional sphere sampled at 3200 points, and spheres with radii in the range \(R = 0.5 \div 2.5\) sampled at a smaller number of points according to the farthest point sampling strategy with a random seed. Ten random samplings were used for each radius. Two distance measures were compared: \(d^P_{PE}\) with \(M = 100\) points and \(d_{CF}\) with \(M = 100, 250\), and 500 points. \(d^P_{PE}\) was computed using the multi-resolution minimization algorithm.

Figure 4 presents the normalized distances as a function of the sphere radius. In Figure 5, a close-up look is shown for radii in the range of \(R = 0.88 \div 1.1\). The grayed areas represent the variance of the distance computed for the ten different samplings of the spheres. This variance makes spheres within a certain interval of radii around \(R = 1\) indistinguishable by the distance function. The partial embedding distance appears to be extremely sensitive to the geometry of the surfaces. A change as small as 0.1\% in the
spherical patch radius (from 1.000 to 0.999) increases $d_{PE}^2$ by a value exceeding the variance due to sampling. Similar results are achieved using $d_{PE}^1$ and slightly inferior with $d_{PE}$. For comparison, with the same number of points (100), $d_{CF}$ is unable to discern between spheres differing in radius by more than 10%. Increasing the number of points to 500 makes $d_{CF}$ sensitive to radius differences of about 2.64%, which is still one order of magnitude below the sensitivity of $d_{PE}^2$. From these results, $d_{PE}$ appears to be a promising tool for very accurate comparison of surfaces.

Reasonable execution times were observed. The generalized stress function (11) and its gradient implemented in C are evaluated in about 40 msec on a Pentium® IV PC for $M = 100$ points. The entire computation of $d_{PE}$ using non-optimized MATLAB and C code took about $10^{-60}$ sec².

4.2 Gromov-Hausdorff distance between isometric surfaces
In the second experiment, we evaluated the performance of the Gromov-Hausdorff distance on six articulated objects and their approximate isometries. The $\ell_2$ and $\ell_\infty$ approximations of $d_{GH}$ were used. All surfaces were represented as triangulated polyhedral meshes (see Figure 6). The embedding spaces consisted of the full-density meshes, whereas the embedded meshes were sampled at $M = N = 50$ points. Optimization was performed using the multi-resolution scheme. Geodesic distances were computed using the fast marching method for triangulated domains [18]. As a reference, the canonical form distance was computed between the surfaces, by comparing a full-density canonical form to canonical forms sampled at $M = 100, 250$ and 600 points as well as to full-density canonical forms.

---

2The codes and datasets used in this paper will be published at http://tosca.cs.technion.ac.il.
Moment signatures and an iterative closest point algorithm were used to compare between the canonical forms.

Figure 7 presents the distance matrices obtained using the $\ell_2$ and $\ell_\infty$ approximations of $d_{GH}$. As few samples as 50 give perfect recognition of all the objects using both $d_{GH}^2$ and $d_{GH}$. For comparison, the best result of $d_{CF}$ with full-density canonical forms and ICP (Figure 7, bottom right) yielded about 11% equal error rate. Figure 8 depicts Euclidean representations of the dissimilarities between the objects measured using $d_{GH}^2$ and $d_{CF}$. Note the tight clusters produced by $d_{GH}^2$ compared to looser clusters obtained from $d_{CF}$.

Fig. 5. A close-up look at the normalized distance between a unit sphere and spheres of different radii measured using $d^2_{PE}$ with 50 points (top) and $d_{CF}$ with 100 and 500 points (two bottom rows). Gray areas denote the variance measured over ten random furthest point samplings. Intervals of sphere radii that fall below the sensitivity threshold of the distance measure are indicated.
The objects Dog and Paper exhibit larger intra-cluster dissimilarities, which we attribute to larger deviations from isometricity.

In order to accurately quantify the discriminative power of the compared distances, we measured the cluster separability as

\[
\sigma = \min_n \left\{ \frac{\min_{i,j,m\neq n} d(S_{mi}, S_{mj})}{\max_i \min_j d(S_{ni}, S_{nj})} \right\},
\]

where \(S_{ni}\) denotes the \(i\)th instance of the \(n\)th object. In simple words, \(\sigma\) expresses the worst-case ratio between the inter-cluster and the intra-cluster distances. The higher is \(\sigma\), the better is the discrimination between the objects. \(d_{\text{GH}}^2\) yielded \(\sigma = 3.47\), which is a manifest of excellent separation; slightly inferior results were exhibited by \(d_{\text{GH}}\). For comparison, the best result obtained by \(d_{\text{CF}}\) was \(\sigma = 0.58\).

Increasing the number of points from 100 to 600 in \(d_{\text{CF}}\) increases the computational complexity with small or no benefit in accuracy. We also noted that starting from about 250 points, the use of ICP in the computation of the canonical form distance gives negligible accuracy improvement compared to the faster moment signature comparison. This leads us to the conclusion that in this experiment, the discriminative power of the canonical form distance is essentially limited by metric distortion introduced by the embedding the surface into \(\mathbb{R}^3\).

4.3 Partial matching of isometric surfaces

In the third experiment, six partial probes (Figure 9) were matched against the objects from the previous experiment using the \(\ell_2\) and \(\ell_\infty\) approximations of \(d_{\text{PE}}\). As before, the embedding spaces consisted of full-density meshes, whereas the partial probes were sampled at \(M = N = 50\) points. Optimization was performed using the multi-resolution scheme. Perfect matching was achieved by both distance functions, yielding the separability of \(\sigma = 1.62\) and 1.29 for \(d_{\text{PE}}^2\) and \(d_{\text{PE}}\), respectively.

Additional numerical assessment of the partial embedding distance was presented in [8], where it was used for face recognition from partially occluded data.

5 Conclusions

In this paper, we introduced a new computational framework for isometry-invariant surface matching. This framework allowed us to formulate the Gromov-Hausdorff distance between two surfaces as a solution of the generalized MDS problem. We also mentioned that the same numerical framework can be used to perform isometry-invariant matching of partially missing surfaces [7]. The GMDS problem is posed as a continuous optimization problem and is solved efficiently using the proposed iterative optimization algorithm. A multiresolution scheme can be used to practically avoid convergence to local minima. In light of recent results [9], we foresee that further performance boosting can be obtained using multigrid optimization methods.

Acknowledgment

We are grateful to D. Burago, A. Brook and V. Lin for their valuable comments and fruitful discussions. We thank our reviewers for their illuminating remarks that helped us to improve the clarity of this paper. This work was supported by the H. & R. Sohnis Cardiology Research Fund.
References

A Proofs of properties of the partial embedding distance

Property 1. Proof. (i) is trivial since $d_{\psi} \geq q$.

The first direction of (ii) is straightforward, since if $Q$ is isometrically embeddable into $S$, there exists $\psi : Q \rightarrow S$ with $d_{\psi} = 0$. The second direction is more elaborate; our proof closely follows the proof of Theorem 7.3.30 in [10]. From $d_{PE}(S, Q) = 0$ it follows that there exists a sequence of mappings $\{\psi_n : Q \rightarrow S\}$ satisfying $\lim_{n \rightarrow \infty} \psi_n \leq \frac{1}{n}$. Fix a countable dense set $Q' \subset Q$. Using the Cantor diagonalization procedure, one can choose a subsequence $\{\psi_{n_k}\}$, such that for every $q' \in Q'$ the sequence $\{\psi_{n_k}(q')\}$ converges in $S$. Let us assume without loss of generality that such a sequence is $\{\psi_n\}$ itself. We define a map $\psi : Q' \rightarrow S$ as the point-wise limit of $\{\psi_n\}$, namely $\psi(q') = \lim_{n \rightarrow \infty} \psi_n(q')$ for every $q' \in Q'$. Since $|d_S(\psi_n(q'), \psi_n(q')) - d_Q(q', q')| \leq d_{\psi_n}(q', q')$ for all $q', q'' \in Q$, i.e. $\psi$ is a distance preserving embedding of $Q'$ into $S$. Since $Q'$ is dense, by Proposition 1.5.9 in [10], $\psi$ can be extended to a distance preserving embedding of the entire $Q$ into $S$.

To prove property (iii), let $d_{PE}(\mathcal{R}, Q) = \delta_1$ and $d_{PE}(\mathcal{S}, \mathcal{R}) = \delta_2$. Then, there exist two sequences of mappings $\{\varphi_n : \mathcal{Q} \rightarrow \mathcal{R}\}$ and $\{\psi_n : \mathcal{R} \rightarrow \mathcal{S}\}$, satisfying $\lim_{n \rightarrow \infty} \varphi_n \leq \frac{1}{n}$ and $\varphi_n \leq 2\delta_1 + \frac{1}{n}$. Denote by $\{\zeta_n : Q \rightarrow S\}$ the sequence of the compositions $\zeta_n = \psi_n \circ \varphi_n$. Invoking the triangle inequality for real numbers, one has

$$|d_S(\zeta_n(q), \zeta_n(q')) - d_Q(q, q')| \leq |d_S(\zeta_n(q), \zeta_n(q')) - d_R(\varphi_n(q), \varphi_n(q'))| + |d_R(\varphi_n(q), \varphi_n(q')) - d_Q(q, q')| \leq d_{\varphi_n} + d_{\psi_n} \leq 2(\delta_1 + \delta_2) + \frac{2}{n},$$

for all $q, q' \in Q'$. Hence, $d_{\zeta_n} \leq 2(\delta_1 + \delta_2) + \frac{2}{n}$ and $d_{PE}(\mathcal{S}, \mathcal{Q}) \leq \delta_1 + \delta_2$. □

Property 2. Proof. (i) This property follows straightforwardly from the fact that $d_{PE}(\mathcal{S}, \mathcal{Q}) \leq d_{GH}(\mathcal{Q}, \mathcal{S})$.

Property 3. We divide the proof into the following two separate statements:

(i) $d_{PE}(\mathcal{S}, \mathcal{Q}) - r \leq d_{PE}(\mathcal{S}, \mathcal{Q'}) \leq d_{PE}(\mathcal{S}, \mathcal{Q})$

(ii) $d_{PE}(\mathcal{S}, \mathcal{Q}) \leq d_{PE}(\mathcal{S'}, \mathcal{Q}) \leq d_{PE}(\mathcal{S}, \mathcal{Q}) + r'$

The combination of (i) and (ii) gives Property 3.

Proof. The right side of inequality (i) is straightforward, since $\sup_{q, q' \in Q} |d_Q(q, q') - d_S(\psi(q), \psi(q'))| \leq |d_Q(q, q') - d_S(\psi(q), \psi(q'))|$. To show the left side, we invoke Property 2(ii), which yields $d_{PE}(\mathcal{S}, \mathcal{Q}) \leq d_{PE}(\mathcal{S}, \mathcal{Q'}) + d_{PE}(\mathcal{Q'}, \mathcal{Q})$. It remains to show that $d_{PE}(\mathcal{Q'}, \mathcal{Q}) \leq r$. Let the mapping $\psi : \mathcal{Q} \rightarrow \mathcal{Q'}$ be defined in the following way: for $q \in \mathcal{Q'}$, $\psi(q) = q$; and for $q \in \mathcal{Q} \setminus \mathcal{Q'}$, $\psi(q) = \arg \min_{q' \in \mathcal{Q'}} d_Q(q, q')$ (the minimum exists, since $\mathcal{Q'}$ can be replaced by its finite sub-covering). From the fact that $\mathcal{Q'}$ is an $r$-covering, it follows that $d_Q(q, \psi(q)) \leq r$. Let $q_1, q_2$ be two points in $Q$; if both are in $\mathcal{Q'}$, then $|d_Q(q_1, q_2) - d_Q(\psi(q_1), \psi(q_2))| = 0$. If both $q_1$ and $q_2$ are in $\mathcal{Q'}$, invoking the triangle
Inequality one has $|d_Q(q_1, q_2) - d_Q(\psi(q_1), \psi(q_2))| \leq d_Q(q_1, \psi(q_1)) + d_Q(q_2, \psi(q_2)) \leq 2r$. Similarly, when only $q_1$ is in $Q'$, $|d_Q(q_1, q_2) - d_Q(\psi(q_1), \psi(q_2))| \leq r$. Hence, $d \psi \leq 2r$.

The left side of inequality (i) is straightforward. To show the right side, let us be given some $\psi : Q \to S$. We define a new mapping $\tilde{\psi} : Q \to S'$ in the following way

$$
\tilde{\psi}(q) = \begin{cases} 
\psi(q) & \text{if } \psi(q) \in S' \\
\arg\min_{s \in S'} d_S(s, \psi(q)) & \text{else}
\end{cases}
$$

Since $S'$ is an $r'$-covering of $S$, $\sup_{q,q' \in Q} |d_Q(q, q') - d_S(\psi(q), \psi(q'))| \leq \sup_{q,q' \in Q} |d_Q(q, q') - d_S(\psi(q), \psi(q'))| + 2r'$ for all $\psi$, and consequently $d_{PE}(S', Q) \leq d_{PE}(S, Q) + r'$. □

**Property 4.** Proof. The right side of the inequality follows from the combination of Property 3 and Property 1ii, $d_{PE}(S_M', Q_N') \leq d_{PE}(S_M', Q) + d_{PE}(Q, Q_N') \leq d_{PE}(S, Q) + r'$. The left side of the inequality follows from Property 3, $d_{PE}(S, Q) \leq d_{PE}(S, Q_N') + r \leq d_{PE}(S_M', Q_N') + r$. □

### B Three-point geodesic distance approximation

We now outline a numerical scheme for interpolation of geodesic distance $d_S(s, s')$ between two arbitrary points $s = (t, u)$ and $s' = (t', u')$ on a polyhedral surface $S_N$ from the matrix $D_S$ of geodesic distances between all $s_i$ and $s_j$. Let us denote the vertices of the triangle $t$, in which the point $s$ lies, as $s_1, s_2, s_3 \in S_N$. Similarly, the vertices of $t'$ will be denoted by $s_1', s_2', s_3'$.

As a starting point, let us consider the particular case where $s'$ is one of the polyhedron vertices. In this case, the distances $d_j = d_S(s_i, s')$ for $i = 1, 2, 3$ can be obtained from the matrix $D_S$. We will assume without loss of generality that the triangle $s_1, s_2, s_3$ lies in $\mathbb{R}^2$ with $s_1$ at the origin and $s_2$ on the horizontal axis, denoting by $x_1 = (0, 0)^T$, $x_2 = (x_2, 0)^T$, and $x_3 = (x_3, y_3)^T$ the planar coordinates of the corresponding vertices. The original triangle can always be brought to this canonical coordinate system by a simple pre-computed transformation. In other words, the actual vertex coordinates $s_i$ in a triangle are not required by the algorithm.

Let us assume that there exists a (virtual) source point $\bar{x}_0 = (\bar{x}_0, \bar{y}_0)^T$ in the plane of the triangle, whose Euclidean distance from $x_i$ is $d_i$. When the surface $S$ is planar, there, indeed, exists such a point consistent with the data, and it can be found by computing the intersection of three circles of radii $d_i$ centered at the corresponding triangle vertices $x_i$ (Fig. 10, left). This gives rise to the following system of quadratic equations

$$
\begin{align*}
\bar{x}_0^2 + \bar{y}_0^2 &= d_1^2 \\
(\bar{x}_0 - x_2)^2 + \bar{y}_0^2 &= d_2^2 \\
(\bar{x}_0 - x_3)^2 + (\bar{y}_0 - y_3)^2 &= d_3^2.
\end{align*}
$$

Once $\bar{x}_0$ is determined, the Euclidean distance $||\bar{x}_0 - x||$ serves as the approximation to the geodesic distance $d_S(\bar{x}_0, x)$. In the above particular case, the two coincide.

However, in the general case when the surface has non-zero Gaussian curvature, the Euclidean distances $||\bar{x}_0 - x||$ will be inconsistent with the geodesic distances $d_i$, and thus the three circles will not generally intersect at a single point (Fig. 10, right). Even if $\bar{x}_0$ can be found to satisfy (18) in the “best possible way”, the Euclidean distance $||\bar{x}_0 - x||$ will not generally coincide with the true geodesic distance $d_S(x, x_0)$. Yet, this scheme still gives a good approximation to $d_S(x, x_0)$.

---

3Theoretically, the scheme is first order, yet, in practice we observed that its accuracy is often superior
Here, we find the virtual source $\hat{x}_0$ by a weighted least-squares solution of (18), namely,

$$\dot{x}_0(\mathbf{X}, \mathbf{d}, \mathbf{w}) = \arg\min_{x_0} \ell(x_0; \mathbf{X}, \mathbf{d}, \mathbf{w}),$$  \hspace{1cm} (19)$$

where

$$\ell(x_0; \mathbf{X}, \mathbf{d}, \mathbf{w}) = \sum_{i=1}^{3} w_i \left( \|x_0 - x_i\|^2 - d_i^2 \right)^2, \hspace{1cm} (20)$$

$\mathbf{X} = (x_1, x_2, x_3), \mathbf{d} = (d_1^2, d_2^2, d_3^2)^T,$ and $\mathbf{w} = (w_1, w_2, w_3)^T$ is a vector of weights controlling the influence of each of the vertices. For a moment, we assume that the weights are given; later we will discuss their choice.

In our implementation, the minimization is carried out using Newton steps [1]. The gradient $\mathbf{g}$ and the Hessian matrix $\mathbf{H}$ of $\ell(x_0; \mathbf{X}, \mathbf{d}, \mathbf{w})$ with respect to $x_0$, required by the Newton method, are given by

$$\mathbf{g} = 4 (\mathbf{w}^T \mathbf{f} \cdot \mathbf{x} - \mathbf{X} \cdot \mathbf{w} \odot \mathbf{f}) \hspace{1cm} (21)$$

$$\mathbf{H} = 4 \mathbf{w}^T \mathbf{f} \cdot \mathbf{I} + 8 \sum_{i=1}^{3} w_i (x_0 - x_i) (x_0 - x_i)^T \hspace{1cm} (22)$$

where $(\mathbf{f})_i = \|x_0 - x_i\|^2 - d_i^2,$ and $\mathbf{w} \odot \mathbf{f}$ denotes the element-wise (Hadamard) product. When the vertices $\mathbf{x}_i$ are not collinear, which normally happens in a valid triangulation, the Hessian is full rank for $w_i \neq 0$. When one of the weights, say $w_3$, is zero, the first two equations in (18) are satisfied and hence the first terms of $\mathbf{f}$ vanish. Consequently, the first term in the Hessian is zero, whereas the second term is a sum of two outer products of linearly independent vectors, which is full rank. When only one weight, say $w_1$ is non-zero, the Hessian assumes the form $\mathbf{H} = 8 w_1 (x_0 - x_1) (x_0 - x_1)^T,$ which is a rank one matrix. Since in this case the solution $\hat{x}_0$ admits only the first equation in (18), the minimum of $\ell(x_0; \mathbf{X}, \mathbf{d}, \mathbf{w})$ is achieved on the circle described by the said equation, and the null vector of $\mathbf{H}$ corresponds to the tangential direction, in which the function does not change.

Due to the lack of convexity of $\ell(x_0; \mathbf{X}, \mathbf{d}, \mathbf{w})$, we need to pay attention to the initialization. In our implementation, we first find simultaneous solutions of different pairs of equations from (18), giving at most six different points, and then initialize the minimization algorithm by the one yielding the smallest value of $\ell$. Solution of a pair of quadratic equations has an analytic expression; for example, the first two equations yield

$$\hat{x}_0 = \frac{d_1^2 - d_2^2 + x_2^2}{2 x_2},$$

$$\hat{y}_0 = \pm \sqrt{d_1^2 - \hat{x}_0^2}, \hspace{1cm} (23)$$

wherever two, one, or none of the solutions exist. Since by its virtue such an initialization is sufficiently close to the optimal $\hat{x}_0$, the algorithm converges in about two or three iterations.

**Lemma B.1.** Let the vertices $\mathbf{x}_i$ be non-collinear and let $\mathbf{w}$ contain at least two non-zero weights. Then, the first-order derivatives of $\mathbf{x}_0(\mathbf{X}, \mathbf{d}, \mathbf{w})$ with respect to $\mathbf{d}$ and $\mathbf{w}$ are given by

$$\mathbf{J}_w = \frac{\nabla_\mathbf{w} \dot{x}_0^T}{\mathbf{G}_w \mathbf{H}^{-1}} = - \mathbf{G}_w \mathbf{H}^{-1},$$

$$\mathbf{J}_d = \frac{\nabla_\mathbf{d} \dot{x}_0^T}{\mathbf{G}_d \mathbf{H}^{-1}} = - \mathbf{G}_d \mathbf{H}^{-1}, \hspace{1cm} (24)$$

about two order of magnitudes compared to the straightforward “non-geometric” linear interpolation of geodesic distances on articulated objects.
where \( G_w = \nabla_w g^T \) and \( G_d = \nabla_d g^T \) are given by
\[
G_w = 4F (x_0 1^T - X), \\
G_d = -4W (x_0 1^T - X),
\]
(25)

\( W \) and \( F \) are diagonal matrices containing the elements of \( w \) and \( f \), respectively, on the diagonal, and \( 1 = (1, 1, 1)^T \).

Proof. Let us start with the derivative with respect to \( w \). By (19), \( x_0 \) is a local minimum of \( \ell (x_0; X, d, w) \), hence \( g (x_0; X, d, w) = 0 \), where \( g \) stands for the gradient \( \nabla_{x_0} \ell \). Adding an infinitesimal perturbation to \( w \) results in \( g (x_0; X, d, w + dw) \neq 0 \), yet there exists an infinitesimal perturbation \( dx_0 \), such that \( g (x_0 + dx_0; X, d, w + dw) = 0 \). From the first-order Taylor expansion,
\[
g (x_0 + dx_0; X, d, w + dw) \approx g (x_0; X, d, w) + H (x_0; X, d, w) dx_0 + G_w^T (x_0; X, d, w) dw,
\]
(26)
where \( H \) stands for the Hessian matrix of \( \ell \) with respect to \( x_0 \) given in (22). Substituting \( g (x_0; X, d, w) = 0 \) and neglecting second-order terms, we obtain the following equation
\[
H dx_0 + G_w^T dw = 0.
\]
(27)
Since for non-collinear \( x_i \) and at most one zero weight the Hessian is invertible,
\[
dx_0 = -H^{-1} G_w^T dw,
\]
(28)
from where \( J_w \) is straightforwardly obtained. In the same way, \( J_d \) is derived.

Having the approximant of the virtual source \( \bar{x}_0 (X, d, w) \) and its first-order derivatives, we can construct the three-point geodesic distance approximant
\[
d_S^2 (x; X, d, w) = \| x - \bar{x}_0 (X, d, w) \|^2
\]
(29)
and its derivatives
\[
\nabla_w d_S^2 (x; X, d, w) = 2 (x - \bar{x}_0), \\
\nabla_w d_S^2 (x; X, d, w) = 2 J_w (\bar{x}_0 - x), \\
\nabla_d d_S^2 (x; X, d, w) = 2 J_d (\bar{x}_0 - x),
\]
(30)
where for convenience we use the squared distance. Assuming that \( s = (t, u) \) is given in normalized barycentric coordinates \( u_1 + u_2 + u_3 = 1 \), one has \( x = Xu = u_1 x_1 + u_2 x_2 + u_3 x_3 \), where \( X \) is a pre-computed matrix defining the geometry of the triangle \( t \). Hence, we have a tool to compute the approximate geodesic distance \( d_S^2 (s, s') \) given \( d = (d_{s1}^2 (s_1, s'), d_{s2}^2 (s_2, s'), d_{s3}^2 (s_3, s'))^T \) and a vector of weights \( w \). The derivatives of \( d_S^2 \) with respect to \( u \) are obtained using the chain rule, \( \nabla_u d_S^2 = X^T \nabla_x X d_S^2 (Xu) \).

Let us now address the issue of weights. We fix \( s' \) to be a point on \( S_N \) and assume that the vector \( d = (d_{s1}^2, d_{s2}^2, d_{s3}^2) \) of the geodesic distances \( d_S (s_i, s_i) \) for \( i = 1, 2, 3 \) is known. It is straightforward that for non-zero weights \( w \) and \( s = (t, u) \) belonging to the interior of the triangle \( s_1, s_2, s_3 \) (i.e., \( u_i > 0 \)), the function \( d_S^2 (u) \) is \( C^1 \). However, the transition of \( s \) to an adjacent triangle, say \( s_4, s_3, s_2 \) will cause the vector of distances to change to \( d = (d_{s1}^2, d_{s2}^2, d_{s3}^2) \). As a result, \( d_S^2 (u) \) is liable to be discontinuous along the triangle edges.
This complicates the numerical solution of (12). Therefore, our goal is to achieve at least $C^1$ continuity for $d_S^2$ on entire $\mathcal{S}_N$.

In order to handle this problem, we set the weights $w$ proportional to the barycentric coordinates of $s, u$. Inside the triangle one has $u_i > 0$ and, consequently, the weights are set to some positive values inside the triangle. For $s$ located on the edge $s_2s_3$, one has $u_1 = u_1 = 0$, which cancels the influence of the vertex $s_1$ in the least squares problem. Hence, the same vertices participate with the same weights when $d_S^2$ is computed both from the triangle $s_1, s_2, s_3$ and $s_2, s_3, s_4$, yielding a continuous transition of $d_S^2$ across the edge. In order to enforce $C^1$ continuity for $d_S^2$, a change of $u$ by a unit vector $\Delta u$ and of $u'$ by a unit vector $\Delta u'$ must displace the point $x$ inside the triangles $t$ and $t'$ by the same Euclidean distance. Since barycentric coordinates are defined up to scale, this can be enforced by selecting an appropriate scaling for $u$ in each triangle.

Since the least-squares problem (19) is under-determined when one of the weights is set to zero and generally yields two distinct solution, one has to select the solution consistent with the case when the weight is slightly larger than zero. In practice, it is easier to avoid the second inconsistent solution by setting the weight to some small positive constant $\epsilon$ rather than strictly to zero. At the triangle vertices two weights vanish, which makes the least squares problem under-determined. In this case, however, there is no need for the least squares solution, since the geodesic distances between the triangle edges to $s'$ are assumed to be known.

Our final step is to put everything together to obtain the approximant $d_S^2(s, s')$ of the geodesic distance between two arbitrary points $s = (t, u)$ and $s' = (t', u')$ on $\mathcal{S}_N$, given the mutual geodesic distances $d_S(s_i, s'_i)$ between the vertices $s_1, s_2, s_3$ of $t$ and $s'_1, s'_2, s'_3$ of $t'$. We first compute the approximate geodesic distances from $s$ to $s'_i$, $d_S^2(s, s'_i) = d_S^2(u; X, d_i, w = u)$ for $i = 1, 2, 3$, where $X$ is the matrix with the canonical coordinates of $s_1, s_2, s_3$ and $d_i = (d_S^2(s_1, s'_1), d_S^2(s_2, s'_2), d_S^2(s_3, s'_3))^T$. Using the vector of distances $d = (d_S^2(s, s'_1), d_S^2(s, s'_2), d_S^2(s, s'_3))^T$ and operating in the triangle $t'$, we compute the approximate geodesic distance between $s$ and $s'$, $d_S^2(s, s') = d_S^2(u'; X', d, w = u')$, where $X'$ are the canonical coordinates of $s'_1, s'_2, s'_3$.

Using the chain rule, it is straightforward to show that the derivatives of $d_S^2(s, s')$ with respect to $u$ and $u'$ are given by

\[
\nabla_u d_S^2(s, s') = \nabla_u d_S^2(u'; X', d, u') + \nabla_w d_S^2(u'; X', d, u') \quad \nabla_u d_S^2(s, s') = \left(\nabla_u d_S^2, \nabla_u d_S^2, \nabla_u d_S^2\right)^T \nabla_d d_S^2(u'; X', d, u'),
\]

where

\[
\nabla_u d_S^2 = \nabla_u d_S^2(u; X, d_i, u) + \nabla_w d_S^2(u; X, d_i, u).
\]
Fig. 6. Six objects (Human, Dog, Giraffe, Crocodile, Paper and Spider) with their approximate isometries.
Fig. 7. Matrices of distances (dissimilarities) between the objects obtained using $d_{GH}^2$ (top left) and $d_{CH}$ (top right) with $M = N = 50$; and $d_{CF}$ with $M = 600$ (bottom left) and full-density canonical forms (bottom right).
Fig. 8. Euclidean proximity pattern representing the distances between different objects measured according to $d^2_{GH}$ with $M = N = 50$ (left), and $d_{CF}$ with $M = 600$ and full-density canonical forms (middle and right, respectively). Each point represents an object; Euclidean distance between two points represents dissimilarity between the corresponding objects.

Fig. 9. Parts of the objects Human, Dog, Giraffe, Crocodile, Paper and Spider used as probes in the third experiment.
Fig. 10. Geodesic distances associated with the triangle vertices $x_1, x_2, x_3$ are used to find a virtual “source” point, distant $d_1, d_2$ and $d_3$ from the corresponding vertices. When the surface is a plane, the three circles intersect at one point $x_0$ and such a source exists (left); in the general case, only an approximate source $\hat{x}_0$ can be found, yet every pair of circles usually intersect at two distinct points (right).