SIAM J. Sci. Comput.
Vol. 28, No. 5, pp. 1812–1836

EFFICIENT COMPUTATION OF ISOMETRY-INVARIANT DISTANCES BETWEEN SURFACES[*]

ALEXANDER M. BRONSTEIN[†], MICHAEL M. BRONSTEIN[†], AND 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–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.

Key words. generalized multidimensional scaling, isometric embedding, Gromov–Hausdorff distance, intrinsic geometry, isometry-invariant surface matching, iterative optimization, multiresolution methods

AMS subject classifications. 15A15, 15A09, 15A23

DOI. 10.1137/050639296

1. Introduction.

Many objects in nature can be described as nonrigid 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 surfaces. 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 a 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*

[*] Received by the editors August 31, 2005; accepted for publication (in revised form) May 2, 2006; published electronically October 27, 2006. This work was supported in part by the H. & R. Sohnis Cardiology Research Fund, United States–Israel Science Foundation grant 2004274, Israel Science Foundation grant 738/04, and the European FP 6 NoE 507752 (MUSCLE).

http://www.siam.org/journals/sisc/28-5/63929.html

[†] Department of Computer Science, Technion – Israel Institute of Technology, Haifa 32000, Israel (alexbron@ieee.org, bronstein@ieee.org, ron@cs.technion.ac.il).

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 [13] 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, 12, 25] and to texture mapping [28, 17]. Similar problems are analyzed in graph theory in the context of representation of discrete metric spaces [20]. Yet, when dealing with abstract metric spaces like graphs, or even point clouds with local distances, smoothness of the underlying geometry cannot 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 [16], 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 all 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 nonconvex optimization problem, which 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*, which 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 nature to MDS and therefore has been named *generalized MDS* or GMDS. GMDS can be thought of 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 isometry-invariant surface matching. There, we define the Gromov–Hausdorff dis-

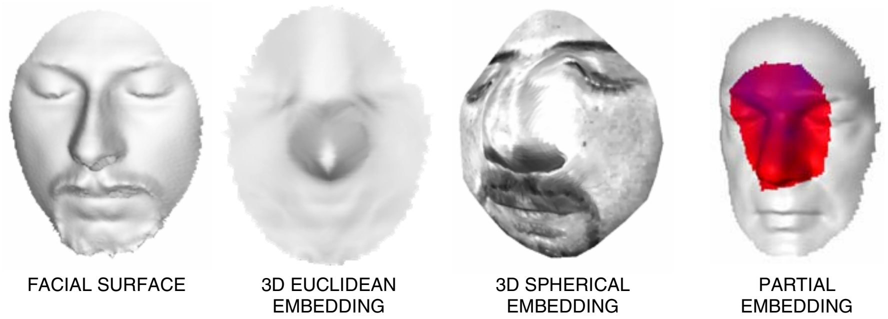

Fig. 1. *Evolution of isometry-invariant object recognition based on isometric embedding, exemplified on the problem of face recognition. Left to right: a facial surface; embedding into* $\mathbb{R}^3$ [6]*; embedding into* $\mathbb{S}^3$ *(shown is the maximum-variance projection onto* $\mathbb{R}^3$*)*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

Table 1
Notation and symbols.

$\mathbb{R}$; $\mathbb{R}^+$	Real numbers; nonnegative real numbers
$\mathbb{R}^m$	m-dimensional Euclidean space
$\mathbb{R}^{M\times N}$	Space of $M \times N$ matrices
$\mathbb{M}$	Set of compact metric spaces
$(\mathcal{S}, d_{\mathcal{S}})$; $\mathcal{S}$	Metric space $\mathcal{S}$ with metric $d_{\mathcal{S}}$ arising from a Riemannian surface
s ; $\mathbf{s}$	A point on $\mathcal{S}$; extrinsic coordinates of s
$d_{\mathcal{S}}\|_{\mathcal{S}'}$	Restriction of $d_{\mathcal{S}}$ to $\mathcal{S}'$
$\mathcal{S}_N$	Finite sampling of $\mathcal{S}$ consisting of N points
$\mathbf{D}_{\mathcal{S}}$	Matrix of geodesic distances on $\mathcal{S}_N$
$T_{\mathcal{S}}$; t^k	Triangulation of $\mathcal{S}_N$; kth triangle in $T_{\mathcal{S}}$
$\mathcal{S}^r$; $\mathcal{S}^r_N$	r-covering; finite r-covering of $\mathcal{S}$
$B_{\mathcal{S}}(s_0, r)$	Open ball of radius r in the metric space $\mathcal{S}$ around s_0
$\mathrm{diam}\,\mathcal{S}$	Diameter of metric space $\mathcal{S}$
$\mathrm{dis}\,\psi$	Distortion of the mapping ψ
$d_{\mathrm{H}}^{\mathcal{Z}}$	Hausdorff metric in the metric space $\mathcal{Z}$
d_{GH}	Gromov–Hausdorff metric between metric spaces
d_{P}	Permutation distance
d_{PE}	Partial embedding distance
Π_N ; π	Permutations of the set $\{1,\ldots,N\}$; a permutation in Π_N
$\mathbf{1}_{N\times N}$	$N \times N$ matrix of ones

tance 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. Notation and symbols used throughout the paper are summarized in Table 1.

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 $d_{\mathcal{S}} : \mathcal{S} \times \mathcal{S} \mapsto \mathbb{R}^+$ the geodesic distance function induced by the Riemannian metric. For simplicity, hereafter 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 $\mathbf{s} = (s^1, s^2, s^3)$.

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

$$\mathcal{S} = \bigcup_{s \in \mathcal{S}^r} B_{\mathcal{S}}(s, r). \tag{1}$$

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 $\mathcal{S}_N^r$. The space $(\mathcal{S}_N^r, d_{\mathcal{S}}|_{\mathcal{S}_N^r})$ constitutes a finite metric space, where the metric $d_{\mathcal{S}}|_{\mathcal{S}_N^r}$ is obtained by restricting the original metric $d_{\mathcal{S}}$ to the subset $\mathcal{S}_N^r$. An arbitrary finite sampling of $\mathcal{S}$ consisting of N points (a *point cloud*) is denoted by $\mathcal{S}_N = \{s_1, \dots, s_N\}$. In order to make a distinction between the original and the sampled surfaces, we refer to $\mathcal{S}$ as a *continuous* surface and to $\mathcal{S}_N$ as a *discrete* surface.

A sampled surface can be represented as a cloud of points $\mathcal{S}_N$ and an $N \times N$ matrix $\mathbf{D}_{\mathcal{S}} = (d_{\mathcal{S}_N}(s_i, s_j))$ of the corresponding geodesic distances. In addition, we assume that a triangulation $T_{\mathcal{S}}$ capturing the local geometry of the surface is given (here $T_{\mathcal{S}} = \{(t_{11}, t_{12}, t_{13}), \dots, (t_{K,1}, t_{K,2}, t_{K,3})\}$ is a list of K triangles, where each triangle is a triplet of sample indices, $t_{ki} \in \{1, \dots, N\}$). The resulting triangulated mesh can be thought of as a polyhedral approximation of the smooth surface $\mathcal{S}$.

Surface $\mathcal{S}$ is called *bounded* if $\operatorname{diam}\mathcal{S} = \sup_{s,s' \in \mathcal{S}} d_{\mathcal{S}}(s, s')$ is finite. $\mathcal{S}$ is said to be *compact* if all its open covers have a finite subcover. If $\mathcal{S}$ 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 $\mathcal{S}$ and $\mathcal{Q}$ in $\mathbb{M}$, a transformation $\varphi : \mathcal{S} \mapsto \mathcal{Q}$ is said to have a *distortion* or distort the metric by

$$\operatorname{dis}\varphi \equiv \sup_{s,s' \in \mathcal{S}} \left| d_{\mathcal{S}}(s, s') - d_{\mathcal{Q}}(\varphi(s), \varphi(s')) \right|. \tag{2}$$

If in addition for every $q \in \mathcal{Q}$ there exists $s \in \mathcal{S}$ such that $d_{\mathcal{Q}}(q, \varphi(s)) \leq \epsilon$ (we call this property *ϵ-surjectivity*), φ is called an *ϵ-isometry* and the spaces $\mathcal{S}$ and $\mathcal{Q}$ are termed *ϵ-isometric.* The image of $\mathcal{S}$, $\varphi(\mathcal{S})$ is called an *ϵ-Hausdorff approximation* of $\mathcal{S}$ in $\mathcal{Q}$. A transformation $\varphi : \mathcal{S} \mapsto \mathcal{Q}$ with $\operatorname{dis}\varphi = 0$ is called an *isometry*, and $\mathcal{S}$ and $\mathcal{Q}$ admitting such a transformation are called *isometric.* Unlike ϵ-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 nonnegative number to every pair of surfaces, measuring the degree of their similarity. The basic property we require is that for all isometries f of $\mathcal{S}$ and g of $\mathcal{Q}$, $d(f(\mathcal{S}), g(\mathcal{Q})) = d(\mathcal{S}, \mathcal{Q})$.

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 cannot guarantee completely avoiding the intrinsic local minima of the associated measure. Our starting point is the classical Hausdorff distance. Assume that the surfaces $\mathcal{S}$ and $\mathcal{Q}$ are subsets of some larger metric space $(\mathcal{Z}, d_{\mathcal{Z}})$. Then the *Hausdorff distance* can be defined in the following way:

$$d_{\mathrm{H}}^{\mathcal{Z}}(\mathcal{S},\mathcal{Q}) \equiv \max\left\{\sup_{s\in\mathcal{S}} d_{\mathcal{Z}}(s,\mathcal{Q}), \sup_{q\in\mathcal{Q}} d_{\mathcal{Z}}(q,\mathcal{S})\right\}. \tag{3}$$

Here for notational simplicity we use $d_{\mathcal{Z}}(s,\mathcal{Q}) = \inf_{q\in\mathcal{Q}} d_{\mathcal{Z}}(s,q)$, implying a point-to-set distance. It can be shown that $d_{\mathrm{H}}^{\mathcal{Z}}$ 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, 11]. 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 [16]. The *Gromov–Hausdorff distance* is defined by

$$d_{\mathrm{GH}}(\mathcal{S},\mathcal{Q}) \equiv \inf_{\substack{\mathcal{Z}\\ \rho:\mathcal{S}\mapsto\mathcal{Z}\\ \sigma:\mathcal{Q}\mapsto\mathcal{Z}}} d_{\mathrm{H}}^{\mathcal{Z}}(\rho(\mathcal{S}),\sigma(\mathcal{Q})), \tag{4}$$

where $\rho : \mathcal{S} \mapsto \mathcal{Z}$ and $\sigma : \mathcal{Q} \mapsto \mathcal{Z}$ are isometric embeddings into the metric space $\mathcal{Z}$. The Gromov–Hausdorff distance is invariant to isometries and thus allows a comparison 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 $\mathbb{M}$, in which a point is defined as an equivalence class of a surface and its isometries [15, 10]. In other words, d_{GH} satisfies the following axioms: (i) $d_{\mathrm{GH}}(\mathcal{S},\mathcal{Q}) \geq 0$; (ii) $d_{\mathrm{GH}}(\mathcal{S},\mathcal{Q}) = 0$ if and only if $\mathcal{S}$ and $\mathcal{Q}$ are isometric; (iii) $d_{\mathrm{GH}}(\mathcal{S},\mathcal{Q}) = d_{\mathrm{GH}}(\mathcal{Q},\mathcal{S})$; and (iv) $d_{\mathrm{GH}}(\mathcal{R},\mathcal{Q}) \leq d_{\mathrm{GH}}(\mathcal{R},\mathcal{S}) + d_{\mathrm{GH}}(\mathcal{S},\mathcal{Q})$, also known as the *triangle inequality.*
2. *Near-isometric similarity:* (i) If $d_{\mathrm{GH}}(\mathcal{S},\mathcal{Q}) \leq \epsilon$, then there exists a 2ϵ-Hausdorff approximation of $\mathcal{S}$ in $\mathcal{Q}$; (ii) if there exists an ϵ-Hausdorff approximation of $\mathcal{S}$ in $\mathcal{Q}$, then $d_{\mathrm{GH}}(\mathcal{S},\mathcal{Q}) \leq 2\epsilon$. This property can be thought of as an extension of the second metric axiom to ϵ-isometries (see [10] for details).
3. *Sampling consistency:* Given $\mathcal{S}^r$ an r-covering of $\mathcal{S}$ and $\mathcal{Q}^{r'}$ and r'-covering of $\mathcal{Q}$, $|d_{\mathrm{GH}}(\mathcal{S},\mathcal{Q}) - d_{\mathrm{GH}}(\mathcal{S}^r,\mathcal{Q}^{r'})| \leq r + r'$. 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 $d_{\mathrm{GH}}(\mathcal{S},\mathcal{Q})$ 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 $\mathcal{S}$ and $\mathcal{Q}$ to each other, thus avoiding the cumbersome construction of the metric space $\mathcal{Z}$. Formally, the Gromov–Hausdorff distance can be redefined as follows [18, 21]:

$$d_{\mathrm{GH}}(\mathcal{S},\mathcal{Q}) = \frac{1}{2}\inf_{\substack{\varphi:\mathcal{S}\mapsto\mathcal{Q}\\ \psi:\mathcal{Q}\mapsto\mathcal{S}}} \max\left\{\operatorname{dis}\phi, \operatorname{dis}\psi, \operatorname{dis}(\varphi,\psi)\right\}, \tag{5}$$

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

$$\operatorname{dis}(\varphi, \psi) \equiv \sup_{\substack{s\in\mathcal{S}\\ q\in\mathcal{Q}}} |d_{\mathcal{S}}(s, \psi(q)) - d_{\mathcal{Q}}(q, \varphi(s))| \tag{6}$$

measures how far φ and ψ are from being the inverse of each other. This reformulation, as we will show, allows us an efficient computation of the Gromov–Hausdorff distance between surfaces.

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

$$d_{\mathrm{P}}(\mathcal{S}_N, \mathcal{Q}_N) \equiv \frac{1}{2} \min_{\pi\in\Pi_N} \max_{1\le i,j\le N} \left| d_{\mathcal{S}}(s_i, s_j) - d_{\mathcal{Q}}(q_{\pi_i}, q_{\pi_j}) \right|,$$

where Π_N denotes all the permutations of $\{1, \dots, N\}$. d_{P} is not the exact Gromov–Hausdorff distance but a different measure of surface similarity related to d_{GH}. Obviously, $d_{\mathrm{GH}}(\mathcal{S}_N, \mathcal{Q}_N) \le d_{\mathrm{P}}(\mathcal{S}_N, \mathcal{Q}_N)$, 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, Mémoli 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].

Mémoli and Sapiro also noted that by restricting the $\mathcal{Z}$ space in definition (4) of d_{GH}, the metric actually boils down to the canonical form distance proposed in [13]. The canonical form distance is computed by first finding the best representation of $\mathcal{S}$ and $\mathcal{Q}$ in $\mathcal{Z}$ (in the sense of lowest metric distortion). Let $\rho : \mathcal{S} \mapsto \mathcal{Z}$ be the lowest distortion embedding of $\mathcal{S}$ in $\mathcal{Z}$; the image $\mathcal{S}_{\mathrm{can}} = \rho(\mathcal{S})$ is referred to as the *canonical form* of $\mathcal{S}$. Similarly, the canonical form of $\mathcal{Q}$ is computed. Once $\mathcal{S}_{\mathrm{can}}$ and $\mathcal{Q}_{\mathrm{can}}$ are available, the canonical form distance is computed according to

$$d_{\mathrm{CF}}(\mathcal{S}, \mathcal{Q}) = d^{\mathcal{Z}}(\mathcal{S}_{\mathrm{can}}, \mathcal{Q}_{\mathrm{can}}), \tag{7}$$

where $d^{\mathcal{Z}}$ is some distance between two sets in $\mathcal{Z}$ (e.g., the Hausdorff distance or its variant). The typical choice of $\mathcal{Z}$ is $\mathbb{R}^3$.

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 $\mathcal{S}$ and a *probe* $\mathcal{Q}$, which is a deformed version of the model. In practice, one is often required to find whether a *patch* $\mathcal{Q}' \subset \mathcal{Q}$ is similar to $\mathcal{S}$.

In [7] we defined the nonsymmetric *partial embedding distance* as

$$d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}') \equiv \frac{1}{2} \inf_{\psi:\mathcal{Q}\mapsto\mathcal{S}} \sup_{(q,q')\in P} \left| d_{\mathcal{Q}}(q,q') - d_{\mathcal{S}}(\psi(q),\psi(q')) \right|, \tag{8}$$

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

$$d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}) = \frac{1}{2} \inf_{\psi:\mathcal{Q}\mapsto\mathcal{S}} \operatorname{dis}\psi. \tag{9}$$

A comment about the potential usefulness of (9) for partial matching was made in [21].

The properties of d_{PE} resemble those of d_{GH}. We summarize them here; for proofs, see Appendix A.

1. *Some metric properties:* d_{PE} satisfies some of the metric axioms: (i) $d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}) \geq 0$; (ii) $d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}) = 0$ if and only if $\mathcal{Q}$ is isometrically embeddable into $\mathcal{S}$; (iii) $d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}) \leq d_{\mathrm{PE}}(\mathcal{S},\mathcal{R}) + d_{\mathrm{PE}}(\mathcal{R},\mathcal{Q})$, which is a nonsymmetric version of the triangle inequality.
2. *Near-isometric similarity:* (i) If there exists an ϵ-Hausdorff approximation of $\mathcal{S}$ in $\mathcal{Q}$, then $d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}) \leq 2\epsilon$. (ii) The converse appears to be true under some more restrictive assumptions (the proof is to be published elsewhere).
3. *Sampling consistency:* Given $\mathcal{S}^r$ an r-covering of $\mathcal{S}$ and $\mathcal{Q}^{r'}$ an r'-covering of $\mathcal{Q}$, $d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}) - r' \leq d_{\mathrm{PE}}(\mathcal{S}^r,\mathcal{Q}^{r'}) \leq d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}) + r$.
4. *Partial matching:* Given $\mathcal{Q}^{r'}$ an r'-covering of $\mathcal{Q}$, $d_{\mathrm{PE}}(\mathcal{Q}^{r'},\mathcal{Q}) = 0$.

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. Second, by virtue of property 4 (unlike the corresponding property $d_{\mathrm{GH}}(\mathcal{Q},\mathcal{Q}^r) \leq r$ of the Gromov–Hausdorff distance, as mentioned above) d_{PE} can be used for partial matching. Third, note that a particular case of property 3 is $d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}) - r \leq d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}^r) \leq d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q})$. When r is too large, the lower bound makes little sense, and one can obtain $d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}^r) \approx 0$ while $d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q})$ is large. This reveals a potential danger in partial matching: when the patch $\mathcal{Q}^r$ 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 with the computation of the simpler $d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q})$, of which the computation of $d_{\mathrm{GH}}(\mathcal{S},\mathcal{Q})$ will be a straightforward generalization. As input, let us be given the following:

1. A discretization of the continuous surface $\mathcal{Q}$, defined by the point cloud $\mathcal{Q}_M^r = \{q_i\}_{i=1}^M$.
2. A polyhedral approximation of the continuous surface $\mathcal{S}$ defined by the point cloud $\mathcal{S}_N^r = \{s_i\}_{i=1}^N$ and a triangulation $T_{\mathcal{S}} = \{(m_{11}, m_{12}, m_{13}), \dots, (m_{N_T,1}, m_{N_T,2}, m_{N_T,3})\}$. In our notation, the polyhedron $\mathcal{S}_N^r$ consists of N_T triangular faces, where the tth face is described by the vertices $s_{m_{t1}}, s_{m_{t2}}, s_{m_{t3}}$.
3. Two distance matrices $\mathbf{D}_{\mathcal{S}} = (d_{\mathcal{S}}(s_i, s_j))$ and $\mathbf{D}_{\mathcal{Q}} = (d_{\mathcal{Q}}(q_i, q_j))$, representing the geodesic distances between the samples of $\mathcal{S}$ and $\mathcal{Q}$, 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 $\mathcal{S}_N^r$ as a polyhedral r approximation of $\mathcal{S}$, we parameterize the points on $\mathcal{S}$ in local *barycentric coordinates* derived from the triangulation. Let s be a point on $\mathcal{S}_N^r$ belonging to some triangle $t \in \{1, \dots, N_T\}$; the Euclidean coordinates $\mathbf{s}$ of s can be represented as a convex combination of the triangle vertices $\mathbf{s}_{m_{t1}}, \mathbf{s}_{m_{t2}}, \mathbf{s}_{m_{t3}}$:

$$\mathbf{s} = \sum_{i=1}^{3} \mathbf{s}_{m_{ti}} u_i. \tag{10}$$

Hence, every point on $\mathcal{S}_N^r$ can be identified with a vector of barycentric coordinates $(t, \mathbf{u}) = (t, u_1, u_2, u_3)$, 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 $\mathcal{S}_N^r$ and $\mathcal{Q}_M^r$ can be formulated as the following constrained minimization problem:

$$\min_{s'_i, \epsilon \geq 0} \epsilon \quad \text{s.t.} \quad \left| d_{\mathcal{S}}(s'_i, s'_j) - d_{\mathcal{Q}}(q_i, q_j) \right| \leq \epsilon, \tag{11}$$

where $(i, j) \in P \subseteq \{1, \dots, M\} \times \{1, \dots, M\}$ (for the time being we assume P to be given; a way to find it is shown in section 3.4). Once the minimal ϵ is found, our approximation to the partial embedding distance is $\hat{d}_{\mathrm{PE}}(\mathcal{S}_N^r, \mathcal{Q}_M^r) = \epsilon/2$.

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 $d_{\mathcal{Q}}(q_i, q_j)$, which are the elements of the matrix $\mathbf{D}_{\mathcal{Q}}$. On the other hand, the distances $d_{\mathcal{S}}(s'_i, s'_j)$ 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 $\mathcal{S}_N^r$, respectively, and can fall inside any triangle on the latter polyhedron.

Theoretically, our computation is accurate up to the sampling radius, i.e., $|\hat{d}_{\mathrm{PE}}(\mathcal{S}_N^r, \mathcal{Q}_M^r) - d_{\mathrm{PE}}(\mathcal{S}, \mathcal{Q})| \leq 2r$. If in addition we take into account that the distances $d_{\mathcal{S}}$ and $d_{\mathcal{Q}}$ are computed with accuracy δ, we have $|\hat{d}_{\mathrm{PE}}(\mathcal{S}_N^r, \mathcal{Q}_M^r) - d_{\mathrm{PE}}(\mathcal{S}, \mathcal{Q})| \leq 2(r + \delta)$. In practice, a finer grid is used to approximate the distances, and 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]

[1] Note that since our optimization problem is nonconvex, no practical optimization algorithm guarantees convergence to the global minimum. Yet, in section 3.2 we show that, in practice, local convergence can be avoided using multiresolution methods.

Problem (11) can be viewed as a minimization of maximum distortion, or a min-max (ℓ_∞ norm) problem. Practical considerations may favor the choice of norms other than ℓ_∞. In order to handle them numerically, we abandon the constrained minimization formulation with the artificial variable ϵ, and minimize

$$F(s'_1, \dots, s'_M) = \sum_{i,j} w_{ij}\eta\left(d_{\mathcal{S}}(s'_i, s'_j) - d_{\mathcal{Q}}(q_i, q_j)\right) \tag{12}$$

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 nonlinear function (e.g., the choices $\eta(x) = x^2$ and $\eta(x) = |x|$ yield the ℓ_2 and the ℓ_1 norms, respectively). Note the resemblance of F to the *stress function* used in the MDS problems.

Though the properties of $\operatorname{argmin}_{s'_i} F$ 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 ℓ_p version of d_{PE} as d^p_{PE}.

The Gromov–Hausdorff distance is computed essentially in the same way as $\hat{d}_{\mathrm{PE}}$ by solving

$$\min_{s'_i, q'_k, \epsilon \geq 0} \epsilon \quad \text{s.t.} \quad \begin{aligned} \left|d_{\mathcal{Q}}(q'_i, q'_j) - d_{\mathcal{S}}(s_i, s_j)\right| &\leq \epsilon, \\ \left|d_{\mathcal{S}}(s'_k, s'_l) - d_{\mathcal{Q}}(q_k, q_l)\right| &\leq \epsilon, \\ \left|d_{\mathcal{S}}(s_i, s'_k) - d_{\mathcal{Q}}(q_k, q'_i)\right| &\leq \epsilon, \end{aligned} \tag{13}$$

where $j > i$, $l > k$. Here two additional terms standing for $\operatorname{dis} \varphi$ and $\operatorname{dis}(\varphi, \psi)$ are added. As in the case of $\hat{d}_{\mathrm{PE}}$, it might be advantageous to replace the ℓ_∞ norm with some more robust nonlinearity, yielding the stress

$$\begin{aligned} F = &\sum_{i,j=1}^{N} \eta\left(d_{\mathcal{Q}}(q'_i, q'_j) - d_{\mathcal{S}}(s_i, s_j)\right) + \sum_{k,l=1}^{M} \eta\left(d_{\mathcal{S}}(s'_k, s'_l) - d_{\mathcal{Q}}(q_k, q_l)\right) \\ &+ \sum_{i,j} \eta\left(d_{\mathcal{S}}(s_i, s'_k) - d_{\mathcal{Q}}(q_k, q'_i)\right). \end{aligned} \tag{14}$$

3.1. Iterative minimization algorithm. We now present a practical numerical algorithm for computation of $\hat{d}_{\mathrm{PE}}$. For clarity, we will henceforth consider only the unconstrained minimization of (12); in order to handle the constrained problem (13), 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 $F(s'_1, \dots, s'_M)$ to a (possibly local) minimum over $s'_i = (t_i, \mathbf{u}_i)$ by starting with some initial guess $s_i^{\prime(0)} = (t_i^{(0)}, \mathbf{u}_i^{(0)})$ of the points and proceeding by iteratively updating their locations, producing a decreasing sequence of function values. Let $s_i^{\prime(k)} = (t_i^{(k)}, \mathbf{u}_i^{(k)})$ be the optimization variables at kth iteration and let there be a set of directions $\mathbf{d}_i^{(k)}$ such that displacement of $\mathbf{u}_i^{(k)}$ 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 $\mathbf{d}_i = -\nabla_{\mathbf{u}_i} F$, which yields a family of algorithms usually termed as *gradient* or *steepest descent.* There exist a variety of more efficient strategies for selecting $\mathbf{d}_i$, including conjugate gradients and quasi-Newton methods, just to name a few [1].

The step size α 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

$$F(s'_1, \dots, s'_M) - F(s''_1, \dots, s''_M) \geq -\sigma\alpha \sum_i \mathbf{d}_i^{\mathrm{T}} \nabla_{\mathbf{u}_i} F(s'_1, \dots, s'_M), \tag{15}$$

where $\sigma \in (0, 1)$ and s''_i represent the points s'_i displaced by α along $\mathbf{d}_i$. In our implementation, we used $\sigma = 0.3$, $\beta = 0.5$. The initial value α_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.

Note that unlike common optimization problems, here, in order to displace s'_i, one cannot simply add $\alpha\mathbf{d}_i$ to $\mathbf{u}_i$, since the latter might leave the triangle t_i, thus invalidating the barycentric representation. Instead, the displacement is performed by following a polylinear path starting at $\mathbf{u}_i$, propagating along a straight line in the direction $\mathbf{d}_i$ 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 α (see Figure 2).

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

1. Compute $\mathbf{u}' = \mathbf{u} + \alpha\mathbf{d}$. If $\mathbf{u}'$ is inside the triangle t, return $s' = (t, \mathbf{u}')$ and stop.
2. Find $\alpha' < \alpha$, for which $\mathbf{u}' = \mathbf{u} + \alpha'\mathbf{d}$ lies on the boundary of t.
3. Otherwise, if there is no triangle adjacent to the edge on which $\mathbf{u}'$ lies, we can proceed no further, since the shape boundary is reached. Return $s' = (t, \mathbf{u}')$ and stop.
4. Otherwise, let t' be a triangle sharing the edge on which $\mathbf{u}'$ lies. Transform $\mathbf{u}'$ and $\mathbf{d}$ to the barycentric coordinates of t', replace α with $\alpha - \alpha'$ and $(t, \mathbf{u}) \leftarrow (t', \mathbf{u}')$, and go to step 1.

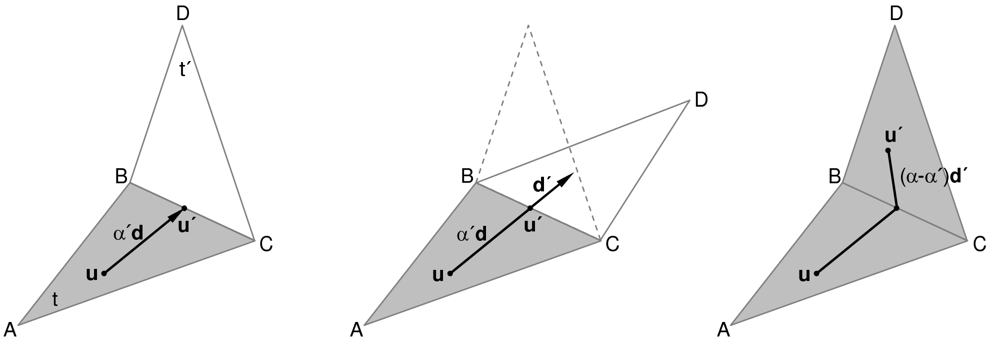

FIG. 2. *Displacement of the point* $\mathbf{u}$ *in the triangle* $t = ABC$ *along the direction* $\mathbf{d}$. *From left to right: the point is displaced along* $\mathbf{d}$ *until the triangle edge* BC *is reached; the adjacent triangle* $t' = BCD$ *is rotated about* BC *and the direction vector* $\mathbf{d}$ *is transformed to the barycentric coordinates of* t', $\mathbf{d}'$; *the final point* $\mathbf{u}'$ *is distant* α *from* $\mathbf{u}$ *along a polylinear path.*

The transformation of $\mathbf{u}'$ from barycentric coordinates in t to those in t' carried out in step 4 is trivial, since $\mathbf{u}'$ 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 $\mathbf{d}$, we rotate the triangle t' about the edge shared with t so that both triangles lie in the same plane, and we find such $\mathbf{d}'$ in the coordinate system of t', which defines the same planar direction as $\mathbf{d}$ 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 $\alpha_0 \mathbf{d}_i$ and cache the entire path; this allows us to save path computations for subsequent smaller steps.

3.2. Multiresolution minimization. GMDS, similarly to MDS algorithms in general [3], is liable to local convergence, since the objective function involved in the computation of $\hat{d}_{\mathrm{GH}}$ and $\hat{d}_{\mathrm{PE}}$ is nonconvex. Here we propose multiresolution optimization schemes as a practical remedy for this problem, in light of the results obtained in [9] for traditional MDS. An additional advantage of multiresolution optimization is a significant speedup of convergence.

The key idea of a multiresolution 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 multiresolution 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 a multiresolution 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 $\mathcal{Q}_{M_1} \subset \mathcal{Q}_{M_2} \subset \cdots \subset \mathcal{Q}_{M_L} = \mathcal{Q}_M$ 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 lth level. The corresponding distance matrices $\mathbf{D}_1, \ldots, \mathbf{D}_L = \mathbf{D}_{\mathcal{Q}}$ are created as submatrices of $\mathbf{D}_{\mathcal{Q}}$.

One possible means of constructing such a hierarchy is the *farthest point sampling* (FPS) strategy [14]. At the coarsest resolution level $\mathcal{Q}_{M_1}$, 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_1+1} is selected as the most distant point from $\mathcal{Q}_{M_1}$, and so on, $q_{M_1+k} = \mathrm{argmax}_{q_i \in \mathcal{Q}_M} d_{\mathcal{Q}_M}(q_i, \{q_1, \ldots, q_{M_1+k-1}\})$. Taking the first M_l points from the sequence produced in this manner, we obtain $\mathcal{Q}_{M_l}$ (see the example in Figure 3).

Let us assume that at the lth resolution level, $\mathcal{Q}_{M_l} = \{q_1, \ldots, q_{M_l}\}$ were embedded into $\mathcal{S}_N^r$ using the iterative minimization algorithm outlined in section 3.1. As a result, the set of images $\psi(\mathcal{Q}_{M_l}) = \{s_1', \ldots, s_{M_l}'\}$ on the polyhedron $\mathcal{S}_N^r$ was obtained. At the next resolution level, we have to embed a larger set $\mathcal{Q}_{M_{l+1}}$ into $\mathcal{S}_N^r$, solving the minimization problem for $\psi(\mathcal{Q}_{M_{l+1}}) = \{s_1', \ldots, s_{M_{l+1}}'\}$. The initialization for the first M_l points is readily available from the solution at the previous level. The initial locations for the remaining points s_i' for $i = M_l + 1, \ldots, M_{l+1}$ have to be interpolated.

It is reasonable to initialize s_i' with such a point on S_N^r that the geodesic distances from it to the points $s_1', \ldots, s_{M_l}'$ are as close as possible to the geodesic distances

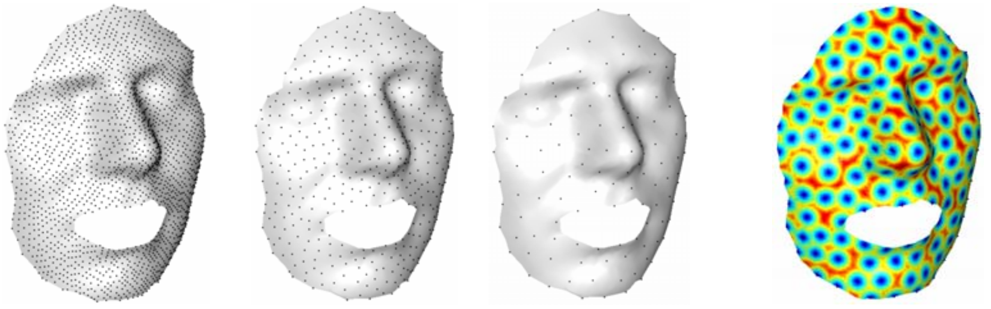

FIG. 3. *Example of data hierarchy construction for multiresolution optimization. Left: three resolution levels constructed using the farthest point sampling algorithm. Right: Geodesic distances from the coarsest grid points.*

from q_i to $q_1, \ldots, q_{M_l}$. Formally, s'_i can be expressed as

$$s'_i = \arg\min_{s} \sum_{j=1}^{M_l} \left(d_{\mathcal{S}}(s, s'_j) - d_{\mathcal{Q}}(q_i, q_j) \right)^2 . \tag{16}$$

Note that practically the minimum can be found by exhaustively searching over all $s_1, \ldots, s_N$ or even a coarser subset of $\mathcal{S}_N^r$. 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 (11) readily available, though it is obvious that only the distances between the surface samples, namely, $d_{\mathcal{S}}(s_i, s_j)$, are available, whereas distances between two arbitrary points in $\mathcal{S}_N^r$ 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 $\mathcal{S}_N^r$. As an input, we are given the following:

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

The output of our numeric procedure is the approximate geodesic distance $\hat{d}_{\mathcal{S}}(s, s') \approx d_{\mathcal{S}}(s, s')$.

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 $s'_1, s\prime_2, s\prime_3$ be the vertices of the triangles t and t' enclosing s and s', respectively. Note that the mutual geodesic distances $d_{\mathcal{S}}(s_i, s'_j)$ are known. A naïve approach is to use the average of the above distances as $\hat{d}_{\mathcal{S}}(s, s')$, i.e.,

$$\hat{d}_{\mathcal{S}}(s, s') = \frac{1}{3} \sum_{i=1}^{3} d_{\mathcal{S}} \left(s_i, s'_i \right) . \tag{17}$$

By the triangle inequality, the approximation error has the order of the radius of the largest triangle, that is, $\mathcal{O}(r)$. In order to make the interpolant $\hat{d}_{\mathcal{S}}(s, s')$ smooth as a function $\mathbf{u}$ and $\mathbf{u}'$, the average is replaced by a weighted sum, where the weights are chosen according to proximity of $\mathbf{u}$ and $\mathbf{u}'$ 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 $d_{\mathcal{S}}\left(s, s_i'\right)$ 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 $d_{\mathcal{S}}\left(s, s_1'\right)$ given $d_{\mathcal{S}}\left(s_i, s_1'\right)$. 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 $d_{\mathcal{S}}\left(s, s_i'\right)$ are available, we invoke again the three-point geodesic distance approximation, this time to compute the desired distance $d_{\mathcal{S}}\left(s, s'\right)$ from $d_{\mathcal{S}}\left(s_i', s\right)$.

3.4. Selection of weight for partial matching. So far, we have assumed that the set P of the consistent pairs of points in (11) or, alternatively, the weights w_{ij} in (12) are given. Here we show a way to find it in practice. Let $\mathcal{S}$ be the model surface and let $\mathcal{Q}$ be the probe surface. We denote by $\mathcal{Q}'$ a patch of $\mathcal{Q}$ and assume that its sampled version is $\mathcal{Q}'_{M'}$. The question is how to find the indices $(i,j) \in \{1,\dots,M'\} \times \{1,\dots,M'\}$ of pairs of points for which $d_{\mathcal{Q}'_{M'}}(q_i,q_j) \neq d_{\mathcal{Q}}(q_i,q_j)$, and thus must be excluded from P.

If a sampled version $\mathcal{Q}_M$ of $\mathcal{Q}$ 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 $|d_{\mathcal{Q}'_{M'}}(q_i,q_j) - d_{\mathcal{Q}_M}(q_i,q_j)| < \delta$, where δ is some small threshold, related to distance approximation accuracy. In the case when only the patch $\mathcal{Q}'$ is available, we note that the minimal geodesics may change due to the change in the boundary of $\mathcal{Q}$. We therefore remove pairs of points (i,j) for which $d_{\mathcal{Q}'}(q_i,\partial\mathcal{Q}') + d_{\mathcal{Q}'}(q_j,\partial\mathcal{Q}') < d_{\mathcal{Q}'}(q_i,q_j)$. Here $\partial\mathcal{Q}'$ denotes the boundary of $\mathcal{Q}'$. Since in practice a sampled version $\mathcal{Q}'_{M'}$ 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 [13] 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 $R = [0.5, 2.5]$ 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^2_{PE} with $M = 100$ points and d_{CF} with $M = 100$, 250, and 500 points. d^2_{PE} was computed using the multiresolution 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 $R = [0.88, 1.1]$. The gray 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

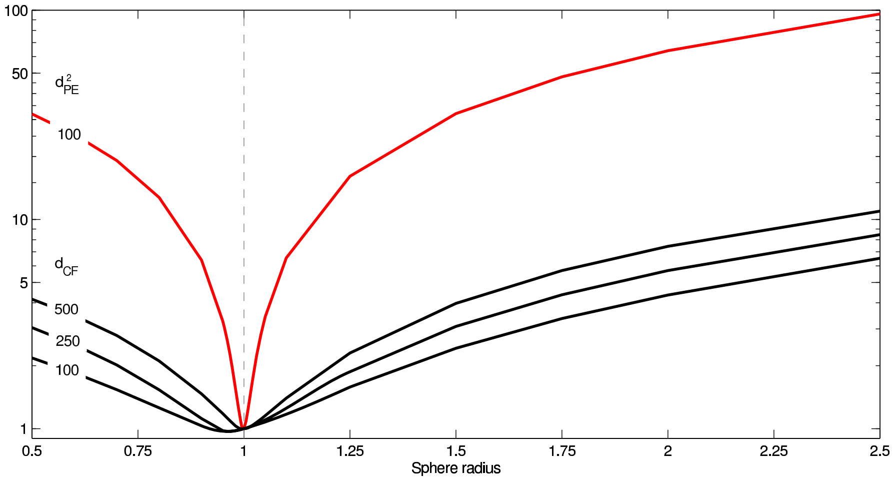

FIG. 4. *Normalized distance between a densely sampled unit sphere and spheres of different radii measured using the ℓ_2 approximation of the partial embedding distance (top line), and the canonical form distance (three bottom lines). The numbers on the plot stand for the number of points used for distance computation.*

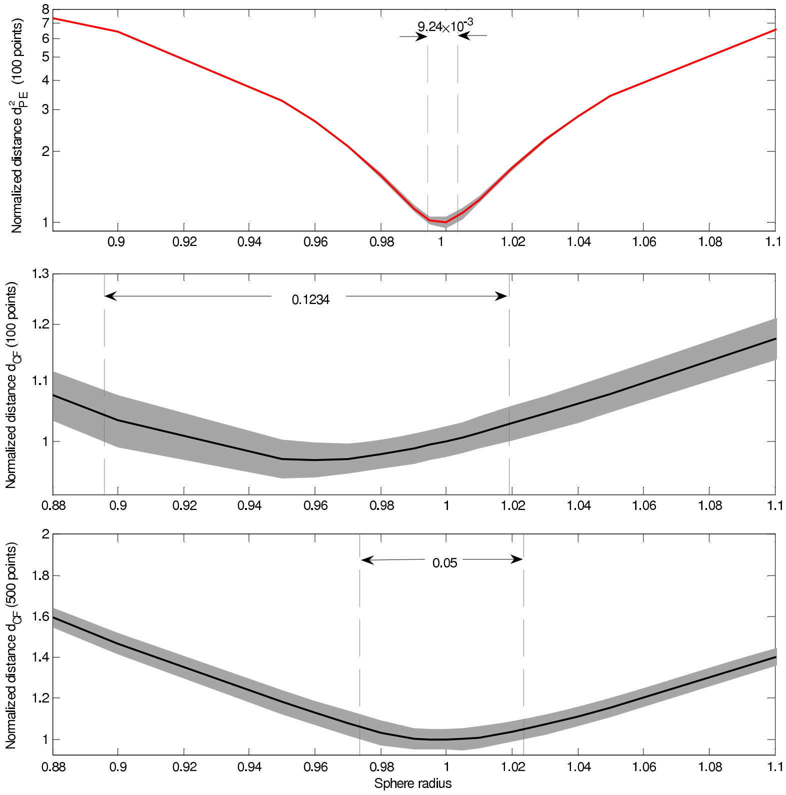

FIG. 5. *A close-up look at the normalized distance between a unit sphere and spheres of different radii measured using* d^2_{PE} *with* 50 *points (top) and* d_{CF} *with* 100 *and* 500 *points (middle and bottom plots). Gray areas denote the variance measured over ten random furthest point samplings. Intervals of sphere radii that fall below the sensitivity threshold of the distance measure are indicated.*

as 0.1% in the spherical patch radius (from 1.000 to 0.999) increases d^2_{PE} by a value exceeding the variance due to sampling. Similar results are achieved using d^1_{PE}, and with d_{PE} results are slightly inferior. 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^2_{PE}. 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 (12) 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 nonoptimized MATLAB and C code took about 10–60 sec.[2]

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 ℓ_2 and ℓ_∞ 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 multiresolution scheme. Geodesic distances were computed using the fast marching method for triangulated domains [19]. 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. Moment signatures and an iterative closest point algorithm were used to compare the canonical forms.

Figure 7 presents the distance matrices obtained using the ℓ_2 and ℓ_∞ approximations of d_{GH}. As few samples as 50 give perfect recognition of all the objects using both d^2_{GH} 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^2_{GH} and d_{CF}. Note the tight clusters produced by d^2_{GH} compared to looser clusters obtained from d_{CF}. The dog and paper objects exhibit larger intracluster 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

$$\sigma = \min_n \left\{ \frac{\min_{i,j,m\neq n} d(\mathcal{S}_{mi}, \mathcal{S}_{mj})}{\max_{i,j} d(\mathcal{S}_{ni}, \mathcal{S}_{nj})} \right\}, \tag{18}$$

where S_{ni} denotes the ith instance of the nth object. In simple words, σ expresses the worst-case ratio between the intercluster and the intracluster distances. The higher σ, the better the discrimination between the objects. d^2_{GH} yielded $\sigma = 3.47$, which is a manifest of excellent separation; slightly inferior results were exhibited by d_{GH}. For comparison, the best result obtained by d_{CF} was $\sigma = 0.58$.

Increasing the number of points from 100 to 600 in d_{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 discrimina-

[2] The codes and datasets used in this paper will be published at http://tosca.cs.technion.ac.il.

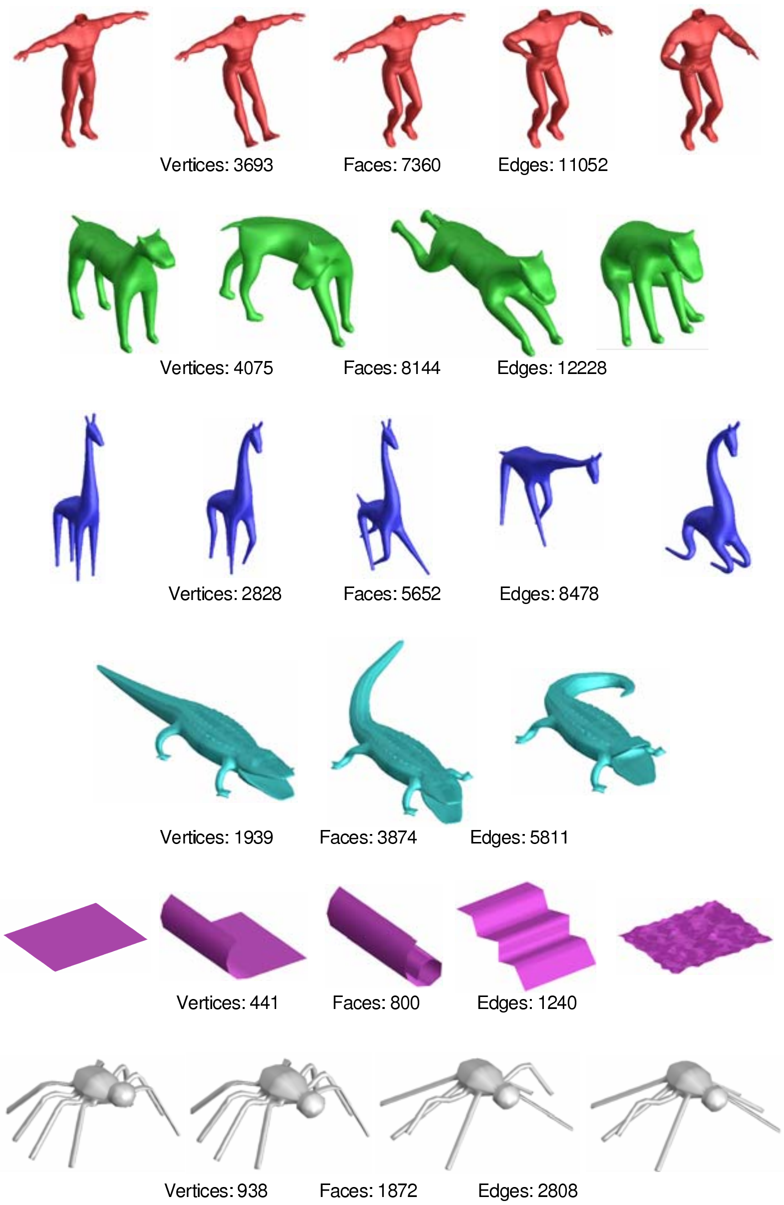

FIG. 6. *Six objects (human, dog, giraffe, crocodile, paper, and spider) with their approximate isometries.*

tive power of the canonical form distance is essentially limited by metric distortion introduced by 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 ℓ_2 and ℓ_∞ approximations of d_{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 multiresolution scheme.

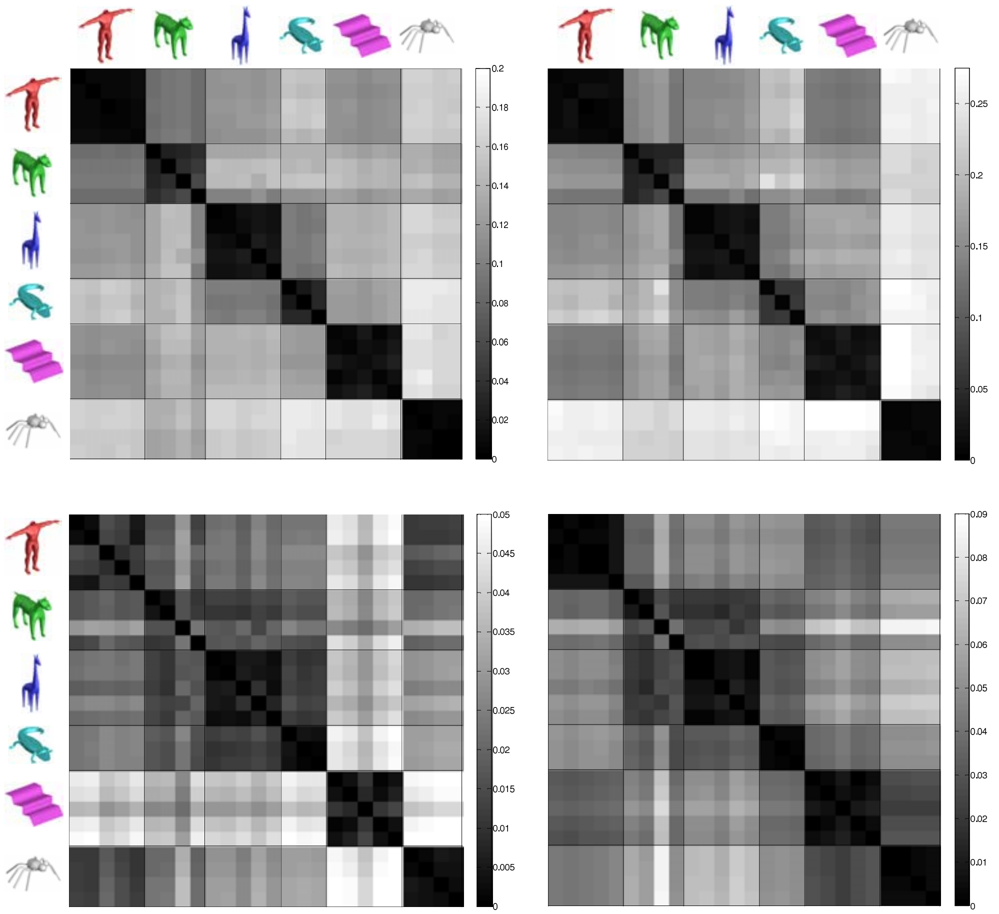

FIG. 7. *Matrices of distances (dissimilarities) between the objects obtained using* d^2_{GH} *(top left) and* d_{GH} *(top right) with* $M = N = 50$*; and* d_{CF} *with* $M = 600$ *(bottom left) and full-density canonical forms (bottom right).*

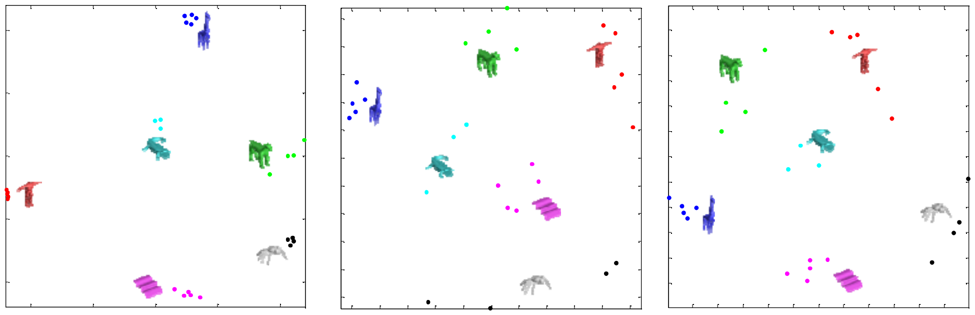

FIG. 8. *Euclidean proximity pattern representing the distances between different objects measured according to* d^2_{GH} *with* $M = N = 50$ *(left) and* d_{CF} *with* $M = 600$ *and full-density canonical forms (middle and right, respectively). Each point represents an object; Euclidean distance between two points represents dissimilarity between the corresponding objects.*

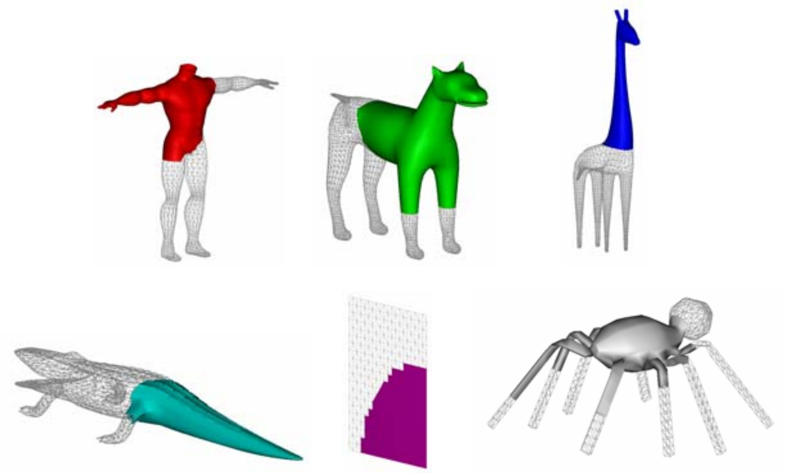

Fig. 9. *Parts of the human, dog, giraffe, crocodile, paper, and spider objects used as probes in the third experiment.*

Perfect matching was achieved by both distance functions, yielding the separability of $\sigma = 1.62$ and 1.29 for d^2_{PE} and d_{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 a practical means of avoiding convergence to local minima. In light of recent results [9], we foresee that further performance boosting can be obtained using multigrid optimization methods.

Appendix A. Proofs of properties of the partial embedding distance.

Proof of property 1. (i) is trivial since $\operatorname{dis}\psi \geq q$.

The first direction of (ii) is straightforward, since if $\mathcal{Q}$ is isometrically embeddable into $\mathcal{S}$, there exists $\psi : \mathcal{Q} \mapsto \mathcal{S}$ with $\operatorname{dis}\psi = 0$. The second direction is more elaborate; our proof closely follows the proof of Theorem 7.3.30 in [10]. From $d_{\mathrm{PE}}(\mathcal{S}, \mathcal{Q}) = 0$ it follows that there exists a sequence of mappings $\{\psi_n : \mathcal{Q} \mapsto \mathcal{S}\}$ satisfying $\operatorname{dis}\psi_n \leq \frac{1}{n}$. Fix a countable dense set $\mathcal{Q}' \subset \mathcal{Q}$. Using the Cantor diagonalization procedure, one can choose a subsequence $\{\psi_{n_k}\}$, such that for every $q' \in \mathcal{Q}'$ the sequence $\{\psi_{n_k}(q')\}$ converges in $\mathcal{S}$. Let us assume without loss of generality that such a sequence is $\{\psi_n\}$ itself. We define a map $\psi : \mathcal{Q}' \mapsto \mathcal{S}$ as the pointwise limit of $\{\psi_n\}$, namely, $\psi(q') = \lim_{n\to\infty} \psi_n(q')$ for every $q' \in \mathcal{Q}'$. Since $|d_{\mathcal{S}}(\psi_n(q''), \psi_n(q')) - d_{\mathcal{Q}}(q'', q')| \leq \operatorname{dis}\psi_n \to 0$, we have

$$d_{\mathcal{S}}(\psi(q''), \psi(q')) = \lim_{n\to\infty} d_{\mathcal{S}}(\psi_n(q''), \psi_n(q')) \ = \ d_{\mathcal{Q}}(q'', q')$$

for all $q', q'' \in \mathcal{Q}'$, i.e., ψ is a distance preserving embedding of $\mathcal{Q}'$ into $\mathcal{S}$. Since

$\mathcal{Q}'$ is dense, by Proposition 1.5.9 in [10], ψ can be extended to a distance preserving embedding of the entire $\mathcal{Q}$ into $\mathcal{S}$.

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

$$\begin{aligned}
\left|d_{\mathcal{S}}\left(\zeta_n\left(q\right),\zeta_n\left(q'\right)\right) - d_{\mathcal{Q}}\left(q,q'\right)\right| &\le \left|d_{\mathcal{S}}\left(\zeta_n\left(q\right),\zeta_n\left(q'\right)\right) - d_{\mathcal{R}}\left(\varphi_n\left(q\right),\varphi_n\left(q'\right)\right)\right| \\
&\quad + \left|d_{\mathcal{R}}\left(\varphi_n\left(q\right),\varphi_n\left(q'\right)\right) - d_{\mathcal{Q}}\left(q,q'\right)\right| \\
&\le \operatorname{dis}\varphi_n + \operatorname{dis}\psi_n < 2\left(\delta_1+\delta_2\right) + \frac{2}{n}
\end{aligned}$$

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

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

Proof of property 3. We divide the proof into the following two separate statements:

(i) $d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}) - r \le d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}^r) \le d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q})$.

(ii) $d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}) \le d_{\mathrm{PE}}(\mathcal{S}^{r'},\mathcal{Q}) \le d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}) + r'$.

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

The right side of inequality (i) is straightforward, since $\sup_{q,q'\in\mathcal{Q}} |d_{\mathcal{Q}^r}(q,q') - d_{\mathcal{S}}(\psi(q),\psi(q'))| \le \sup_{q,q'\in\mathcal{Q}} |d_{\mathcal{Q}}(q,q') - d_{\mathcal{S}}(\psi(q),\psi(q'))|$. To show the left side, we invoke property 1(ii), which yields $d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}) \le d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}^r) + d_{\mathrm{PE}}(\mathcal{Q}^r,\mathcal{Q})$. It remains to show that $d_{\mathrm{PE}}(\mathcal{Q}^r,\mathcal{Q}) \le r$. Let the mapping $\psi : \mathcal{Q} \mapsto \mathcal{Q}^r$ be defined in the following way: for $q \in \mathcal{Q}^r$, $\psi(q) = q$; and for $q \in \mathcal{Q}\backslash\mathcal{Q}^r$, $\psi(q) = \arg\min_{q'\in\mathcal{Q}^r} d_{\mathcal{Q}}\left(q,q'\right)$ (the minimum exists, since $\mathcal{Q}^r$ can be replaced by its finite subcovering). From the fact that $\mathcal{Q}^r$ is an r-covering, it follows that $d_{\mathcal{Q}}\left(q,\psi(q)\right) \le r$. Let q_1, q_2 be two points in $\mathcal{Q}$; if both are in $\mathcal{Q}^r$, then $\left|d_{\mathcal{Q}}\left(q_1,q_2\right) - d_{\mathcal{Q}}\left(\psi\left(q_1\right),\psi\left(q_2\right)\right)\right| = 0$. If both q_1 and q_2 are in $\mathcal{Q}^r$, invoking the triangle inequality, one has $\left|d_{\mathcal{Q}}\left(q_1,q_2\right) - d_{\mathcal{Q}}\left(\psi\left(q_1\right),\psi\left(q_2\right)\right)\right| \le d_{\mathcal{Q}}\left(q_1,\psi\left(q_1\right)\right) + d_{\mathcal{Q}}\left(q_2,\psi\left(q_2\right)\right) \le 2r$. Similarly, when only q_1 is in $\mathcal{Q}^r$, $|d_{\mathcal{Q}}(q_1,q_2) - d_{\mathcal{Q}}(\psi(q_1),\psi(q_2))| \le r$. Hence, $\operatorname{dis}\psi \le 2r$.

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

$$\tilde{\psi}(q) = \begin{cases} \psi(q), & \psi(q) \in \mathcal{S}^{r'}, \\ \operatorname{argmin}_{s\in\mathcal{S}^{r'}} d_{\mathcal{S}}(s,\psi(q)) & \text{else.} \end{cases}$$

Since $\mathcal{S}^{r'}$ is an r'-covering of $\mathcal{S}$, $\sup_{q,q'\in\mathcal{Q}} |d_{\mathcal{Q}}(q,q') - d_{\mathcal{S}}(\tilde{\psi}(q),\tilde{\psi}(q'))| \le \sup_{q,q'\in\mathcal{Q}} |d_{\mathcal{Q}}(q,q') - d_{\mathcal{S}}(\psi(q),\psi(q'))| + 2r'$ for all ψ, and consequently $d_{\mathrm{PE}}(\mathcal{S}^{r'},\mathcal{Q}) \le d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}) + r'$. □

Proof of property 4. The right side of the inequality follows from the combination of properties 3 and 1(ii), $d_{\mathrm{PE}}(\mathcal{S}_M^{r'},\mathcal{Q}_N^r) \le d_{\mathrm{PE}}(\mathcal{S}_M^{r'},\mathcal{Q}) + d_{\mathrm{PE}}(\mathcal{Q},\mathcal{Q}_N^r) \le d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}) + r'$. The left side of the inequality follows from property 3, $d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}) \le d_{\mathrm{PE}}(\mathcal{S},\mathcal{Q}_N^r) + r \le d_{\mathrm{PE}}(\mathcal{S}_M^{r'},\mathcal{Q}_N^r) + r$. □

Appendix B. Three-point geodesic distance approximation. We now outline a numerical scheme for interpolation of geodesic distance $d_{\mathcal{S}}(s,s')$ between two arbitrary points $s = (t,\mathbf{u})$ and $s' = (t',\mathbf{u}')$ on a polyhedral surface $\mathcal{S}_N$ from the matrix $\mathbf{D}_{\mathcal{S}}$ of geodesic distances between all s_i and s_j. Let us denote the vertices of

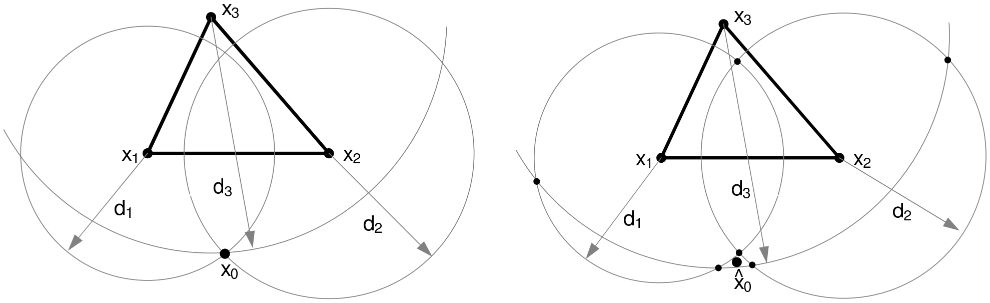

FIG. 10. *Geodesic distances associated with the triangle vertices* $\mathbf{x}_1$*,* $\mathbf{x}_2$*,* $\mathbf{x}_3$ *are used to find a virtual "source" point, distant* d_1*,* d_2*, and* d_3 *from the corresponding vertices. When the surface is a plane, the three circles intersect at one point* $\mathbf{x}_0$ *and such a source exists* (*left*)*; in the general case, only an approximate source* $\hat{\mathbf{x}}_0$ *can be found, yet every pair of circles usually intersects at two distinct points* (*right*)*.*

the triangle t, in which the point s lies, as $s_1, s_2, s_3 \in \mathcal{S}_N$. 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 $d_j = d_{\mathcal{S}}(s_i, s')$ for $i = 1, 2, 3$ can be obtained from the matrix $\mathbf{D}_{\mathcal{S}}$. 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 $\mathbf{x}_1 = (0,0)^{\mathrm{T}}$, $\mathbf{x}_2 = (x_2, 0)^{\mathrm{T}}$, and $\mathbf{x}_3 = (x_3, y_3)^{\mathrm{T}}$ the planar coordinates of the corresponding vertices. The original triangle can always be brought to this *canonical* coordinate system by a simple precomputed transformation. In other words, the actual vertex coordinates $\mathbf{s}_i$ in a triangle are not required by the algorithm.

Let us assume that there exists a (virtual) source point $\hat{\mathbf{x}}_0 = (\hat{x}_0, \hat{y}_0)^{\mathrm{T}}$ in the plane of the triangle, whose *Euclidean distance* from $\mathbf{x}_i$ is d_i. When the surface $\mathcal{S}$ 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 $\mathbf{x}_i$ (Figure 10, left). This gives rise to the following system of quadratic equations:

$$\begin{aligned} \hat{x}_0^2 + \hat{y}_0^2 &= d_1^2, \\ (\hat{x}_0 - x_2)^2 + \hat{y}_0^2 &= d_2^2, \\ (\hat{x}_0 - x_3)^2 + (\hat{y}_0 - y_3)^2 &= d_3^2. \end{aligned} \tag{19}$$

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

However, in the general case when the surface has nonzero Gaussian curvature, the Euclidean distances $\|\hat{\mathbf{x}}_0 - \mathbf{x}_i\|$ will be inconsistent with the geodesic distances d_i, and thus the three circles will not generally intersect at a single point (Figure 10, right). Even if $\hat{\mathbf{x}}_0$ can be found to satisfy (19) in the "best possible way," the Euclidean distance $\|\hat{\mathbf{x}}_0 - \mathbf{x}\|$ will not generally coincide with the true geodesic distance $d_{\mathcal{S}}(\mathbf{x}, \mathbf{x}_0)$. Yet this scheme still gives a good approximation to $d_{\mathcal{S}}(\mathbf{x}, \mathbf{x}_0)$.[3]

[3] Theoretically, the scheme is first order, but in practice we observed that its accuracy is often superior by about two orders of magnitude compared to the straightforward "nongeometric" linear interpolation of geodesic distances on articulated objects.

Here, we find the virtual source $\hat{\mathbf{x}}_0$ by a weighted least-squares solution of (19), namely,

$$\hat{\mathbf{x}}_0\left(\mathbf{X},\mathbf{d},\mathbf{w}\right) = \arg\min_{\mathbf{x}_0} \ell\left(\mathbf{x}_0;\mathbf{X},\mathbf{d},\mathbf{w}\right), \tag{20}$$

where

$$\ell(\mathbf{x}_0;\mathbf{X},\mathbf{d},\mathbf{w}) = \sum_{i=1}^{3} w_i\big(\left\|\mathbf{x}_0-\mathbf{x}_i\right\|^2 - d_i^2\big)^2, \tag{21}$$

$\mathbf{X} = (\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3)$, $\mathbf{d} = (d_1^2,d_2^2,d_3^2)^{\mathrm{T}}$, and $\mathbf{w} = (w_1,w_2,w_3)^{\mathrm{T}}$ 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 $\ell(\mathbf{x}_0;\mathbf{X},\mathbf{d},\mathbf{w})$ with respect to $\mathbf{x}_0$, required by the Newton method, are given by

$$\mathbf{g} = 4\left(\mathbf{w}^{\mathrm{T}}\mathbf{f}\cdot\mathbf{x} - \mathbf{X}\cdot\mathbf{w}\odot\mathbf{f}\right), \tag{22}$$

$$\mathbf{H} = 4\ \mathbf{w}^{\mathrm{T}}\mathbf{f}\cdot\mathbf{I} + 8\sum_{i=1}^{3} w_i\left(\mathbf{x}_0-\mathbf{x}_i\right)\left(\mathbf{x}_0-\mathbf{x}_i\right)^{\mathrm{T}}, \tag{23}$$

where $(\mathbf{f})_i = \left\|\mathbf{x}_0-\mathbf{x}_i\right\|^2 - d_i^2$, and $\mathbf{w}\odot\mathbf{f}$ denotes the elementwise (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 (19) 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 nonzero, the Hessian assumes the form $\mathbf{H} = 8w_1\left(\mathbf{x}_0-\mathbf{x}_1\right)\left(\mathbf{x}_0-\mathbf{x}_1\right)^{\mathrm{T}}$, which is a rank one matrix. Since in this case the solution $\hat{\mathbf{x}}_0$ admits only the first equation in (19), the minimum of $\ell(\mathbf{x}_0;\mathbf{X},\mathbf{d},\mathbf{w})$ 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 $\ell(\mathbf{x}_0;\mathbf{X},\mathbf{d},\mathbf{w})$, we need to pay attention to the initialization. In our implementation, we first find simultaneous solutions of different pairs of equations from (19), giving at most six different points, and then initialize the minimization algorithm by the one yielding the smallest value of ℓ. The solution of a pair of quadratic equations has an analytic expression; for example, the first two equations yield

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

$$\hat{y}_0 = \pm\sqrt{d_1^2 - \hat{x}_0^2} \tag{24}$$

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

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

$$\begin{aligned} \mathbf{J}_{\mathbf{w}} &= \nabla_{\mathbf{w}} \hat{\mathbf{x}}_0^{\mathrm{T}} = -\mathbf{G}_{\mathbf{w}} \mathbf{H}^{-1}, \\ \mathbf{J}_{\mathbf{d}} &= \nabla_{\mathbf{d}} \hat{\mathbf{x}}_0^{\mathrm{T}} = -\mathbf{G}_{\mathbf{d}} \mathbf{H}^{-1}, \end{aligned} \tag{25}$$

where $\mathbf{G}_{\mathbf{w}} = \nabla_{\mathbf{w}} \mathbf{g}^{\mathrm{T}}$ and $\mathbf{G}_{\mathbf{d}} = \nabla_{\mathbf{d}} \mathbf{g}^{\mathrm{T}}$ are given by

$$\begin{aligned} \mathbf{G}_{\mathbf{w}} &= 4\mathbf{F}\left(\mathbf{x}_0 \mathbf{1}^{\mathrm{T}} - \mathbf{X}\right), \\ \mathbf{G}_{\mathbf{d}} &= -4\mathbf{W}\left(\mathbf{x}_0 \mathbf{1}^{\mathrm{T}} - \mathbf{X}\right). \end{aligned} \tag{26}$$

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

Proof. Let us start with the derivative with respect to $\mathbf{w}$. By (20), $\hat{\mathbf{x}}_0$ is a local minimum of $\ell(\mathbf{x}_0; \mathbf{X}, \mathbf{d}, \mathbf{w})$, hence $\mathbf{g}(\hat{\mathbf{x}}_0; \mathbf{X}, \mathbf{d}, \mathbf{w}) = 0$, where $\mathbf{g}$ stands for the gradient $\nabla_{\mathbf{x}_0} \ell$. Adding an infinitesimal perturbation to $\mathbf{w}$ results in $\mathbf{g}(\hat{\mathbf{x}}_0; \mathbf{X}, \mathbf{d}, \mathbf{w} + d\mathbf{w}) \neq 0$, yet there exists an infinitesimal perturbation $d\hat{\mathbf{x}}_0$, such that $\mathbf{g}(\hat{\mathbf{x}}_0 + d\hat{\mathbf{x}}_0; \mathbf{X}, \mathbf{d}, \mathbf{w} + d\mathbf{w}) = 0$. From the first order Taylor expansion,

$$\begin{aligned} \mathbf{g}(\hat{\mathbf{x}}_0 + d\hat{\mathbf{x}}_0; \mathbf{X}, \mathbf{d}, \mathbf{w} + d\mathbf{w}) &\approx \mathbf{g}(\hat{\mathbf{x}}_0; \mathbf{X}, \mathbf{d}, \mathbf{w}) \\ &\quad + \mathbf{H}(\hat{\mathbf{x}}_0; \mathbf{X}, \mathbf{d}, \mathbf{w})\, d\hat{\mathbf{x}}_0 + \mathbf{G}_{\mathbf{w}}^{\mathrm{T}}(\hat{\mathbf{x}}_0; \mathbf{X}, \mathbf{d}, \mathbf{w})\, d\mathbf{w}, \end{aligned} \tag{27}$$

where $\mathbf{H}$ stands for the Hessian matrix of ℓ with respect to $\mathbf{x}_0$ given in (23). Substituting $\mathbf{g}(\hat{\mathbf{x}}_0; \mathbf{X}, \mathbf{d}, \mathbf{w}) = 0$ and neglecting second order terms, we obtain

$$\mathbf{H} d\hat{\mathbf{x}}_0 + \mathbf{G}_{\mathbf{w}}^{\mathrm{T}} d\mathbf{w} = 0. \tag{28}$$

Since for noncollinear $\mathbf{x}_i$ and at most one zero weight the Hessian is invertible,

$$d\hat{\mathbf{x}}_0 = -\mathbf{H}^{-1} \mathbf{G}_{\mathbf{w}}^{\mathrm{T}} d\mathbf{w}, \tag{29}$$

from which $\mathbf{J}_{\mathbf{w}}$ is straightforwardly obtained. In the same way, $\mathbf{J}_{\mathbf{d}}$ is derived. □

Having the approximant of the virtual source $\hat{\mathbf{x}}_0(\mathbf{X}, \mathbf{d}, \mathbf{w})$ and its first order derivatives, we can construct the three-point geodesic distance approximant,

$$\hat{d}_{\mathcal{S}}^2(\mathbf{x}; \mathbf{X}, \mathbf{d}, \mathbf{w}) = \left\| \mathbf{x} - \hat{\mathbf{x}}_0(\mathbf{X}, \mathbf{d}, \mathbf{w}) \right\|^2, \tag{30}$$

and it derivatives,

$$\begin{aligned} \nabla_{\mathbf{x}} \hat{d}_{\mathcal{S}}^2(\mathbf{x}; \mathbf{X}, \mathbf{d}, \mathbf{w}) &= 2(\mathbf{x} - \hat{\mathbf{x}}_0), \\ \nabla_{\mathbf{w}} \hat{d}_{\mathcal{S}}^2(\mathbf{x}; \mathbf{X}, \mathbf{d}, \mathbf{w}) &= 2\mathbf{J}_{\mathbf{w}}(\hat{\mathbf{x}}_0 - \mathbf{x}), \\ \nabla_{\mathbf{d}} \hat{d}_{\mathcal{S}}^2(\mathbf{x}; \mathbf{X}, \mathbf{d}, \mathbf{w}) &= 2\mathbf{J}_{\mathbf{d}}(\hat{\mathbf{x}}_0 - \mathbf{x}), \end{aligned} \tag{31}$$

where for convenience we use the squared distance. Assuming that $s = (t, \mathbf{u})$ is given in normalized barycentric coordinates $u_1 + u_2 + u_3 = 1$, one has $\mathbf{x} = \mathbf{X}\mathbf{u} = u_1 \mathbf{x}_1 + u_2 \mathbf{x}_2 + u_3 \mathbf{x}_3$, where $\mathbf{X}$ is a precomputed matrix defining the geometry of the triangle t.

Hence, we have a tool to compute the approximate geodesic distance $\hat{d}^2_{\mathcal{S}}(s,s')$ given $\mathbf{d} = (d^2_{\mathcal{S}}(s_1,s'), d^2_{\mathcal{S}}(s_2,s'), d^2_{\mathcal{S}}(s_3,s'))^{\mathrm{T}}$ and a vector of weights $\mathbf{w}$. The derivatives of $\hat{d}^2_{\mathcal{S}}$ with respect to $\mathbf{u}$ are obtained using the chain rule, $\nabla_{\mathbf{u}}\hat{d}^2_{\mathcal{S}} = \mathbf{X}^{\mathrm{T}}\nabla_{\mathbf{x}}\hat{d}^2_{\mathcal{S}}(\mathbf{X}\mathbf{u})$.

Let us now address the issue of weights. We fix s' to be a point on $\mathcal{S}_N$ and assume that the vector $\mathbf{d} = (d_1^2, d_2^2, d_3^2)^{\mathrm{T}}$ of the geodesic distances $d_{\mathcal{S}}(s', s_i)$ for $i = 1,2,3$ is known. It is straightforward that for nonzero weights $\mathbf{w}$ and $s = (t, \mathbf{u})$ belonging to the interior of the triangle s_1, s_2, s_3 (i.e., $u_i > 0$), the function $\hat{d}^2_{\mathcal{S}}(\mathbf{u})$ is $\mathcal{C}^1$. 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 $\mathbf{d} = (d_4^2, d_2^2, d_3^2)^{\mathrm{T}}$. As a result, $\hat{d}^2_{\mathcal{S}}(\mathbf{u})$ is liable to be discontinuous along the triangle edges. This complicates the numerical solution of (13). Therefore, our goal is to achieve at least $\mathcal{C}^1$ continuity for $\hat{d}^2_{\mathcal{S}}$ on all of $\mathcal{S}_N$.

In order to handle this problem, we set the weights $\mathbf{w}$ proportional to the barycentric coordinates of s, $\mathbf{u}$. 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 $\overline{s_2 s_3}$, one has $w_1 = u_1 = 0$, which cancels the influence of the vertex s_1 in the least squares problem. Hence, the same vertices participate with the same weights when $\hat{d}^2_{\mathcal{S}}$ is computed from both triangles s_1, s_2, s_3 and s_2, s_3, s_4, yielding a continuous transition of $\hat{d}^2_{\mathcal{S}}$ across the edge. In order to enforce $\mathcal{C}^1$ continuity for $\hat{d}^2_{\mathcal{S}}$, a change of $\mathbf{u}$ by a unit vector $\Delta\mathbf{u}$ and of $\mathbf{u}'$ by a unit vector $\Delta\mathbf{u}'$ must displace the point $\mathbf{x}$ 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 $\mathbf{u}$ in each triangle.

Since the least-squares problem (20) is underdetermined when one of the weights is set to zero and generally yields two distinct solutions, 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 ϵ rather than strictly to zero. At the triangle vertices two weights vanish, which makes the least-squares problem underdetermined. 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 $d^2_{\mathcal{S}}(s,s')$ of the geodesic distance between two arbitrary points $s = (t, \mathbf{u})$ and $s' = (t', \mathbf{u}')$ on $\mathcal{S}_N$, given the mutual geodesic distances $d_{\mathcal{S}}(s_i, s'_j)$ 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, $\hat{d}^2_{\mathcal{S}}(s, s'_i) = \hat{d}^2_{\mathcal{S}}(\mathbf{u}; \mathbf{X}, \mathbf{d}_i, \mathbf{w} = \mathbf{u})$ for $i = 1,2,3$, where $\mathbf{X}$ is the matrix with the canonical coordinates of s_1, s_2, s_3 and $\mathbf{d}_i = (d^2_{\mathcal{S}}(s_1, s'_i), d^2_{\mathcal{S}}(s_2, s'_i), d^2_{\mathcal{S}}(s_3, s'_i))^{\mathrm{T}}$. Using the vector of distances $\hat{\mathbf{d}} = (\hat{d}^2_{\mathcal{S}}(s, s'_1), \hat{d}^2_{\mathcal{S}}(s, s'_2), \hat{d}^2_{\mathcal{S}}(s, s'_3))^{\mathrm{T}}$ and operating in the triangle t', we compute the approximate geodesic distance between s and s', $\hat{d}^2_{\mathcal{S}}(s, s') = \hat{d}^2_{\mathcal{S}}(\mathbf{u}'; \mathbf{X}', \hat{\mathbf{d}}, \mathbf{w} = \mathbf{u}')$, where $\mathbf{X}'$ are the canonical coordinates of s'_1, s'_2, s'_3.

Using the chain rule, it is straightforward to show that the derivatives of $\hat{d}^2_{\mathcal{S}}(s, s')$ with respect to $\mathbf{u}$ and $\mathbf{u}'$ are given by

$$\begin{aligned}
&\nabla_{\mathbf{u}'}\hat{d}^2_{\mathcal{S}}(s,s') = \nabla_{\mathbf{u}}\hat{d}^2_{\mathcal{S}}\big(\mathbf{u}'; \mathbf{X}', \hat{\mathbf{d}}, \mathbf{u}'\big) + \nabla_{\mathbf{w}}\hat{d}^2_{\mathcal{S}}\big(\mathbf{u}'; \mathbf{X}', \hat{\mathbf{d}}, \mathbf{u}'\big),\\
&\nabla_{\mathbf{u}}\hat{d}^2_{\mathcal{S}}(s,s') = \big(\nabla_{\mathbf{u}}\hat{d}^2_1, \nabla_{\mathbf{u}}\hat{d}^2_2, \nabla_{\mathbf{u}}\hat{d}^2_3\big)^{\mathrm{T}}\nabla_{\mathbf{d}}\hat{d}^2_{\mathcal{S}}\big(\mathbf{u}'; \mathbf{X}', \hat{\mathbf{d}}, \mathbf{u}'\big),
\end{aligned} \tag{32}$$

where

$$\nabla_{\mathbf{u}}\hat{d}^2_i = \nabla_{\mathbf{u}}\hat{d}^2_{\mathcal{S}}(\mathbf{u}; \mathbf{X}, \mathbf{d}_i, \mathbf{u}) + \nabla_{\mathbf{w}}\hat{d}^2_{\mathcal{S}}(\mathbf{u}; \mathbf{X}, \mathbf{d}_i, \mathbf{u}).$$

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, which helped us to improve the clarity of this paper.

REFERENCES

[1] D. Bertsekas, *Nonlinear Programming*, 2nd ed., Atlanta Scientific, Belmont, MA, 1999.

[2] P. J. Besl and N. D. McKay, *A method for registration of 3D shapes*, IEEE Trans. PAMI, 14 (1992), pp. 239–256.

[3] I. Borg and P. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, Springer-Verlag, Berlin, Heidelberg, New York, 1997.

[4] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Expression-invariant face recognition via spherical embedding*, in Proceedings of the International Conference on Image Processing, Vol. 3, 2005, pp. 756–759.

[5] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *On isometric embedding of facial surfaces into* $\mathbb{S}^3$, in Proceedings of the International Conference on Scale Space and PDE Methods in Computer Vision, Lecture Notes on Comput. Sci. 3459, Springer-Verlag, Berlin, 2005, pp. 622–631.

[6] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Three-dimensional face recognition*, Internat. J. Comput. Vision, 64 (2005), pp. 5–30.

[7] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Generalized multidimensional scaling: A framework for isometry-invariant partial surface matching*, Proc. Natl. Acad. Sci. USA, 103 (2006), pp. 1168–1172.

[8] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, *Robust expression-invariant face recognition from partially missing data*, in Proceedings of the European Conference on Computer Vision (ECCV), 2006, pp. 396–408.

[9] M. M. Bronstein, A. M. Bronstein, R. Kimmel, and I. Yavneh, *Multigrid multidimensional scaling*, Numer. Linear Algebra Appl., 13 (2006), pp. 149–171.

[10] D. Burago, Y. Burago, and S. Ivanov, *A Course in Metric Geometry*, Grad. Stud. Math. 33, AMS, Providence, RI, 2001.

[11] Y. Chen and G. Medioni, *Object modeling by registration of multiple range images*, Image and Vision Comput., 10 (1992), pp. 145–155.

[12] D. Donoho and C. Grimes, *Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data*, Proc. Natl. Acad. Sci. USA, 100 (2003), pp. 5591–5596.

[13] A. Elad and R. Kimmel, *On bending invariant signatures for surfaces*, IEEE Trans. PAMI, 25 (2003), pp. 1285–1295.

[14] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi, *The farthest point strategy for progressive image sampling*, IEEE Trans. Image Processing, 6 (1997), pp. 1305–1315.

[15] M. Gromov, *Isometric immersions and embeddings*, Soviet Math. Dokl., 11 (1970), pp. 794–797.

[16] M. Gromov, *Structures métriques pour les variétés riemanniennes*, Textes Math. 1, Cedic, Paris, 1981.

[17] R. Grossmann, N. Kiryati, and R. Kimmel, *Computational surface flattening: A voxel-based approach*, IEEE Trans. Pattern Anal. Machine Intelligence, 24 (2002), pp. 433–441.

[18] N. J. Kalton and M. I. Ostrovskii, *Distances between Banach spaces*, Forum Math., 11 (1999), pp. 17–48.

[19] R. Kimmel and J. A. Sethian, *Computing geodesic on manifolds*, Proc. Natl. Acad. Sci. USA, 95 (1998), pp. 8431–8435.

[20] N. Linial, E. London, and Y. Rabinovich, *The geometry of graphs and some its algorithmic applications*, Combinatorica, 15 (1995), pp. 333–344.

[21] F. Mémoli and G. Sapiro, *A theoretical and computational framework for isometry invariant recognition of point cloud data*, Found. Comput. Math., 5 (2005), pp. 313–346.

[22] S. T. Roweis and L. K. Saul, *Nonlinear dimensionality reduction by locally linear embedding*, Science, 290 (5500) (2000), pp. 2323–2326.

[23] E. L. Schwartz, A. Shaw, and E. Wolfson, *A numerical solution to the generalized mapmaker's problem: Flattening nonconvex polyhedral surfaces*, IEEE Trans. PAMI, 11 (1989), pp. 1005–1008.

[24] J. A. Sethian, *A fast marching level set method for monotonically advancing fronts*, Proc. Nat. Acad. Sci. USA, 93 (1996), pp. 1591–1595.

[25] J. B. Tennenbaum, V. de Silva, and J. C Langford, *A global geometric framework for nonlinear dimensionality reduction*, Science, 290 (2000), pp. 2319–2323.

[26] J. Walter and H. Ritter, *On interactive visualization of high-dimensional data using the hyperbolic plane*, in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 123–131.

[27] Z. Y. Zhang, *Iterative point matching for registration of free-form curves and surfaces*, Internat. J. Comput. Vision, 13 (1994), pp. 119–152.

[28] G. Zigelman, R. Kimmel, and N. Kiryati, *Texture mapping using surface flattening via multi-dimensional scaling*, IEEE Trans. Visualization and Comput. Graphics, 9 (2002), pp. 198–207.