

Spectral-driven isometry-invariant matching of 3D shapes

Mauro R. Ruggeri · Giuseppe Patanè ·
Michela Spagnuolo · Dietmar Saupe

Revision date: May, 18, 2009.

Abstract This paper presents a matching method for 3D shapes, which comprises a new technique for surface sampling and two algorithms for matching 3D shapes based on point-based statistical shape descriptors. Our sampling technique is based on critical points of the eigenfunctions related to the smaller eigenvalues of the Laplace-Beltrami operator. These critical points are invariant to isometries and are used as anchor points of a sampling technique, which extends the farthest point sampling by using statistical criteria for controlling the density and number of reference points. Once a set of reference points has been computed, for each of them we construct a *point-based statistical descriptor* (PSSD, for short) of the input surface. This descriptor incorporates an approximation of the geodesic shape distribution and other geometric information describing the surface at that point. Then, the dissimilarity between two surfaces is computed by comparing the corresponding sets of PSSDs with bipartite graph matching or measuring the L_1 -distance between the reordered feature vectors of a proximity graph. Here, the reordering is given by the Fiedler vector of a Laplacian matrix associated to the proximity graph. Our tests have shown that both approaches are suitable for online retrieval of deformed objects and our sampling strategy improves the retrieval performances of isometry-invariant matching methods. Finally, the approach based on the Fiedler

vector is faster than using the bipartite graph matching and it has a similar retrieval effectiveness.

Keywords Isometry-invariant matching, 3D model retrieval, feature points, local statistical shape descriptors, Laplace-Beltrami operator.

1 Introduction

Recent developments in 3D modelling and acquisition techniques contributed to the large spread of 3D models in many fields such as CAD/CAM, architecture, computer entertainment, culture heritage, and medicine. In these contexts, 3D models represent non-rigid or deformable 3D objects with different postures or deformations. For instance, in component inspection for CAD/CAM engineering applications 3D models of articulated objects can be scanned, modelled, and stored in different postures. In an articulated object, components are attached through joints and can move about. Similarly, human faces and organs may be subject to different facial expressions and deformations, respectively. The deformations that transform a shape without stretching or tearing its surface, i.e., preserving the geodesic distances among points, are called *isometries*. Therefore, isometry-invariant comparison techniques are useful for shape recognition, retrieval, and classification. Fig. 1 shows an isometry-invariant deformation applied to an articulated object.

In this paper, which extends the work presented in [61], we consider the problem of sampling 3D shapes to achieve an efficient and effective isometry-invariant matching. Section 2 briefly reviews previous work on this and related topics. As 3D shape, we consider a surface represented by a triangle mesh; however, the

Mauro R. Ruggeri and Dietmar Saupe
Department of Computer and Information Science
University of Konstanz, Germany
E-mail: {mauro.ruggeri,dietmar.saupe}@uni-konstanz.de

Giuseppe Patanè and Michela Spagnuolo
Istituto di Matematica Applicata e Tecnologie Informatiche
Consiglio Nazionale delle Ricerche, Italy
E-mail: {patane,spagnuolo}@ge.imati.cnr.it

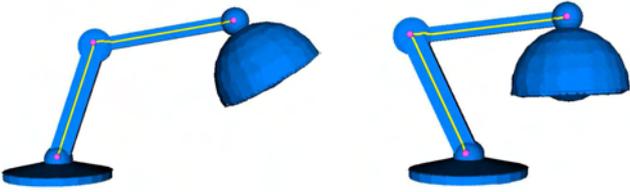


Fig. 1 An articulated object is shown in two different poses after applying an isometric deformation. The geodesic distance between the selected points as well as the geodesic path does not change with the deformation.

proposed technique is independent of the surface representation. The basic idea of our approach is to ensure a comparison of the most geometrically meaningful parts of shapes through a novel sampling technique, which uses the critical points of the eigenfunctions of the Laplace-Beltrami operator as *anchor points* (see Section 3). Low frequencies of the Laplace-Beltrami spectrum are related to information about the global structure and features of the input surface, while high frequencies reflect fine changes in its shape. These anchor points are located on geometrically and topologically meaningful regions of the shape and are invariant with respect to isometries. Then, we enrich these anchors with a set Q_r of sample points (called *reference points*), whose number, distribution, and density are controlled by statistical criteria.

Once the reference points have been computed, we describe a 3D shape with a set of *statistical descriptors*, each associated to a point $\mathbf{p} \in Q_r$ (see Section 4). Each descriptor incorporates an approximation of the geodesic shape distribution of the surface as seen from the corresponding point \mathbf{p} ; i.e., the histogram of the geodesic distances from \mathbf{p} to the others. Then, the histogram is enriched with other geometrical information to constitute a *point-based statistical shape descriptor* (PSSD, for short). Instead of considering a global histogram roughly describing the entire surface [24], the 3D shape is abstracted as a set of isometry-invariant PSSDs. This choice produces a more accurate description of the intrinsic properties of the surface and leads to a better retrieval effectiveness.

Similarly to [38, 40], we compare two 3D shapes by matching sets of reference sample points. We describe these sets through the corresponding sets of PSSDs, which are compared with a bipartite graph matching approach or by evaluating the L_1 -distance of the re-ordered PSSDs. To this end, we order the PSSDs with respect to the magnitude of the *Fiedler vector* components and form a single feature vector. This choice resembles spectral techniques for aligning embedding spaces [37]. In fact, we consider the ordering of the components of the feature vector induced by the first non-

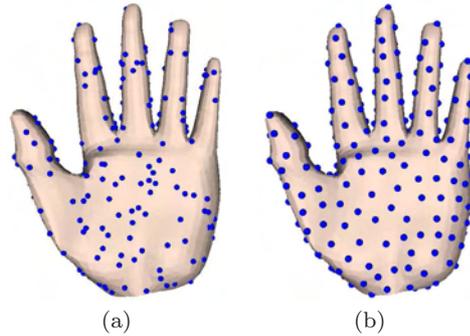


Fig. 2 (a) Uniform sampling *versus* (b) evenly spaced sampling of a 3D shape.

trivial eigenvector of the Laplacian matrices of weighted proximity graphs having our reference points as vertices. The matching approach using the Fiedler vector is faster than the bipartite matching and produces similar retrieval effectiveness. Moreover, the alignments *via* Fiedler vector can be performed in the preprocessing stage and provide a single feature vector that can be efficiently compared by using different metrics. Our matching algorithms are compared with several isometry-invariant methods and the sampling strategy is also applied to some of these techniques to evaluate its benefits for improving their retrieval effectiveness (see Section 5). Possible extensions of the proposed framework are discussed in Section 6.

2 Related work

In the following, we review previous work on the identification of reference points for shape correspondence and matching (see Section 2.1) and isometry invariant matching techniques (see Section 2.2).

2.1 Reference points for shape correspondence and matching

In several applications, which include shape comparison, cross parameterization, deformation transfer, shape morphing, and statistical shape modeling, an important step is the *detection of sample points*, i.e., points of the input surface with a high information content related to the surface geometry and topology. Several methods [15, 24, 48, 61, 66] approximate a surface with a triangle mesh that is uniformly sampled as a preliminary step of a shape feature extraction algorithm. The sampling strategy basically consists in picking random triangles with probabilities proportional to their area and generating random sample points inside them with equal probability per unit area. Although the uniform

sampling is widely adopted in many methods due to its simple implementation and fast execution, it may generate samples that are very close to each other. Furthermore, it might miss meaningful parts of a shape (see Fig. 2(a)) and create artefacts in the definition of high-level or statistical descriptors. Finally, as shown in [45, 61], we note that sampling strategies generating sample points evenly distributed on the surface (see Fig. 2(b)) can improve the retrieval performances of several algorithms [24, 48, 61]. Among these sampling techniques, we mention the farthest [17] and stratified [45] point sampling.

For shape analysis and matching, the spatial distribution of points should capture the shape of the object, so that a descriptor does not miss important geometric structures. To address these requirements, the *feature points* are commonly computed as the extrema of scalar functions that measure local properties of the input shape. Among them, we recall the distribution of the curvature values [29, 44, 53], which classify the local shape of 3D surfaces into planar, parabolic, and elliptic regions; saliency maps for 2D images [27] and 3D shapes [11, 33]; Euclidean- [1, 19] and geodesics-based functions [21, 25, 69], which identify surface protrusions [15, 22, 43]. These approaches can also be combined to take into account several surface properties; for instance, the work presented in [21] identifies salient features by measuring geometric properties at a point over different local views. The methods in [44, 53] estimate the local variation of the curvature values and their persistency over several scales. To improve the effectiveness of the selected feature points and reduce the influence of noise, sampling density, and surface tessellation on their identification, the shape descriptors are evaluated in a multi-scale manner using neighborhoods of increasing size. Alternatives are the Gaussian filters or the simplification of the surface representation with edge collapse and vertex removal.

Once the feature points have been identified, two surfaces are matched by establishing correspondences among feature points or using local statistical shape descriptors at these points. Automatic feature correspondence is usually best-suited for rigid shapes. To deal with large shape variations, the work presented in [70] uses a set of features located on the prominent parts of a shape, which are the maxima and minima of the average squared geodesic distance field [25, 69]. The local maxima of this function correspond to convex or concave shape extremities and the local minima are located on a region around the shape barycenter.

2.2 Isometry-invariant matching of 3D shapes

Techniques for comparing 3D objects deformed with isometric deformations were conceived by several authors. Hilaga *et al.* [25] presented a technique to match the topology of triangulated models by comparing *Multiresolution Reeb Graphs* (MRG, for short) [3, 51]. The MRG of a triangle mesh S , which approximates a smooth surface \mathcal{M} , was constructed by discretizing the function $\mu_1(\mathbf{v}) = \int_{\mathcal{M}} g(\mathbf{v}, \mathbf{p}) d\mathbf{p}$, where the value $g(\mathbf{v}, \mathbf{p})$ is the geodesic distance between the two points $\mathbf{v}, \mathbf{p} \in \mathcal{M}$. The algorithm for matching two MRGs is a coarse-to-fine strategy that searches the node pairs providing the largest value of similarity while maintaining topological consistency. As discussed in [4], this high-level structure can also be used to find sub-parts correspondences of 3D shapes. Similarly to [25], Hamza and Krim [24] considered an approximation of the squared geodesic distance function, which is defined as $\mu_2(\mathbf{v}) = \int_{\mathcal{M}} |g(\mathbf{v}, \mathbf{p})|^2 d\mathbf{p}$.

In both cases, the geodesic distance was computed by considering a set S' of centroids or base points, which are selected through a farthest point-like sampling. The choice of S' , the local shape noise, and an irregular surface tessellation may generate clustered feature points. To partially overcome these drawbacks, the resulting functions μ_1 or μ_2 are smoothed applying a polynomial kernel and clustered feature points in S' are removed *via* geodesic Poisson disk sampling [70]. These functions were then assumed as random variables with a common probability density map, which gives a statistical shape descriptor. The dissimilarity between two objects was calculated by computing the Jensen-Shannon divergence between the corresponding statistical shape descriptors.

Elad and Kimmel [15] proposed a canonical representation for triangulated surfaces, which is invariant with respect to isometries. A 3D surface was simplified to a low number of n vertices (e.g., $n := 2000$) and then transformed into canonical coordinates in the m -dimensional Euclidean space by applying a multi-dimensional scaling. In this canonical representation, the geodesic distances on the original surface were approximated by the corresponding Euclidean distances. The matching problem of non-rigid and deformed objects was reduced to the problem of matching rigid objects, which is approached with the iterative closest point algorithm [72], the moment comparison [16], and the method of eigenfaces [67].

Mémoli and Sapiro in [40] compared dense point clouds by computing an approximation of the *Gromov-Hausdorff* (GH, for short) *distance* between two compact metric spaces. This distance is an extension of the symmetric Hausdorff distance and intuitively mea-

sures how far two compact subsets of a metric space are from being isometric. They considered the geodesic distance as a metric on surfaces and showed that the computation of the GH distance leads to a combinatorial problem. They proposed an heuristic that progressively constructs approximations of the GH distance of subsets of the point clouds by minimizing the point-wise approximation error. In [38], Mémoli reformulated the problem of approximating the GH distance between compact metric spaces as a mass transportation problem, where the mass of each sample point of the metric spaces is expressed as a probability measure. This new formulation leads to a quadratic optimization problem with linear constraints, which is solved with an iterative procedure. Moreover, the author provided a theoretical framework that enables to understand the computational complexity of the GH distance and its relation to other metrics and matching methods presented in [7, 15, 24, 40]. Finally, Bronstein *et al.* [7] approximated the GH distance between two smooth surfaces using a generalized multi-dimensional scaling to compute the minimum distortion between those surfaces. They proposed a multi-resolution algorithm that minimizes an approximation of the distortion map between two triangulated surfaces. This approach was also used to compute an approximation of a non-symmetric partial embedding distance, which intuitively measures how similar a patch of the surface is to another.

Reuter *et al.* in [58] compared two shapes by computing the distance between two isometry-invariant feature vectors given by the smallest k , $10 \leq k \leq 100$, eigenvalues of the Laplace-Beltrami operator defined on the input shape. In [28], Jain and Zhang compared non-rigid objects by matching spectral embeddings, which are derived from the eigenvectors of affinity matrices computed considering geodesic distances. In [62], the first k Laplacian eigenvalues and eigenvectors are used to define an isometry-invariant shape representation. Then, these signatures are compared using a modification of the $D2$ -distribution [48], which is based on a set of histograms that capture the variation of distances among points within a set of spherical cells centered at the origin of a k -dimensional space.

A similar approach is proposed in [37]; here, spectral embeddings, constructed as *Local Linear Embedding* (LLE, for short) on eigenspaces of affinity matrices, are matched by using the *Expectation-Maximization* (EM, for short) framework. More precisely, a 3D surface is mapped in a d -dimensional space using a locally linear embedding, which is pose-invariant and preserves the distances between each point and its k -nearest neighbors in a least-squares sense. Then, two shapes are compared by aligning them through an orthogonal trans-

formation and estimating correspondences among the data points. For the alignment of the embedded spaces, the proposed approach computes the singular value decomposition of a $d \times d$ matrix. The parameters d and k need to be accurately tuned in order to avoid wrong correspondences between points or an over-fitting in the embedding. The EM framework serves to estimate a solution resolving the intrinsic ambiguities in eigenspace alignment. Among them, we mention the sign reversal of eigenfunctions, the eigenvalues switch, and residual transformation induced by the LLE embeddings, which preserve only local pairwise distances between data points.

Ohbuchi *et al.* [47] compared articulated 3D shapes by using the bag-of-features approach to match large sets of multi-scale local visual features. This method consists in accumulating multiple local features into a single histogram based on a specific clustering, and consider the histogram as a feature vector of the 3D shape. Ruggeri and Saupe [61] compared point-sampled surfaces by matching sets of histograms of geodesic distances with bipartite graph matching. These histograms can be efficiently extracted and compared by using various bin-to-bin distance functions [59]; e.g., the various L_p -distances, the χ^2 distance, the Kullback-Leibler divergence distance, and the Bhattacharyya distance. The aforementioned methods assume that the domains of the histograms are already aligned, although in practice histograms approximating the same probability density function might be misaligned and have different scales. To overcome this problem, techniques considering scale invariant cross-bin comparison of histograms (e.g., the Earth Mover’s distance) were developed in [35, 48, 59]. In practice, when histograms have a small number of bins a good compromise between matching efficiency and effectiveness can be obtained by using the L_1 -distance or the χ^2 distance defined as

$$d_{\chi^2}(\mathbf{H}_i, \mathbf{H}_j) = \frac{1}{2} \sum_{k=1}^B \frac{|\mathbf{H}_i(k) - \mathbf{H}_j(k)|^2}{\mathbf{H}_i(k) + \mathbf{H}_j(k)}, \quad (1)$$

where B is the number of bins of the histograms \mathbf{H}_i and \mathbf{H}_j , and $\mathbf{H}_i(k)$ is the value of the k -th bin of \mathbf{H}_i . Bins equal to zero (i.e., $\mathbf{H}_i(k) = \mathbf{H}_j(k) = 0$) are discarded. However, the choice of a good dissimilarity measure depends on the histogram space and can be estimated experimentally [8].

3 Selecting anchor and feature points on 3D shapes

In the following, we present a novel technique for sampling a 3D shape based on the critical points of isometry-

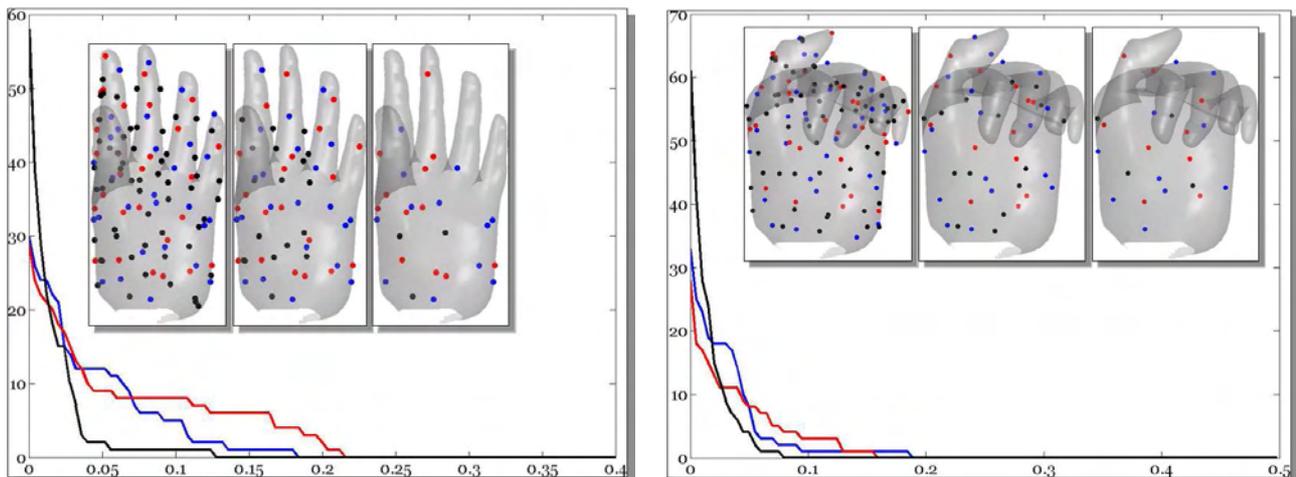


Fig. 3 Simplification of the critical points with respect to a given threshold ϵ . The big pictures show the evolution of the number of critical points (y -axis) with respect to a different ϵ (x -axis): the red, blue, and black curves represent the number of maxima, minima, and saddle points at each step. In the upper part of each picture, the left image depicts the input critical points and the other two images show the critical points preserved by the simplification.

invariant scalar functions, given by the eigenmaps of the Laplace-Beltrami operator related to its smaller eigenvalues. To this end, we discuss the definition (see Section 3.1) and selection (see Section 3.2) of the anchor points. Finally (see Section 3.3), we introduce the statistically controlled sampling used to select the anchors among the critical points of several Laplacian eigenfunctions.

3.1 Laplacian eigenfunctions: definition and main properties

The choice of the scalar functions for feature detection introduced by previous work (see Section 2.1) depends on the application and is a delicate task. As discussed in [46], an arbitrarily chosen function usually yields a complicated arrangement of many critical points that is impractical for several applications. Our intuition is that the critical points are meaningful for the description of the input surface \mathcal{M} if we select functions that are smooth [20]; intrinsically defined by the shape of \mathcal{M} ; invariant to isometric transformations. Furthermore, we expect that the critical points are located on prominent features of \mathcal{M} , which identify semantically meaningful regions [29, 44] (e.g., protrusions, symmetries), and are extracted on the base of local shape signatures [21, 23, 34].

In the following, we will verify that the Laplace-Beltrami eigenmaps and the corresponding critical points satisfy all the aforementioned properties. Furthermore, using the Laplacian eigenmaps avoids to select a specific function or choose the base points for its discretization, as it happens for the geodesic maps.

Definition of the Laplacian eigenfunctions. The Laplacian eigenfunctions on \mathcal{M} are defined as the solutions of the following *eigenvalue problem*

$$\text{find } f : \mathcal{M} \rightarrow \mathbb{R} \text{ such that } \Delta f = \lambda f, \quad \lambda \in \mathbb{R}. \quad (2)$$

In the discrete setting, let us consider a triangulated surface S with vertices $\{\mathbf{p}_1, \dots, \mathbf{p}_n\}$; as shown in [58, 68], equation (2) is equivalent to the *generalized eigenvalue problem*

$$L_{\text{cot}} \mathbf{f} = \lambda B \mathbf{f}, \quad \mathbf{f} := (f(\mathbf{p}_1), \dots, f(\mathbf{p}_n))^T,$$

where the $n \times n$ matrices L_{cot} and B are defined as

$$L_{\text{cot}}(i, j) := \begin{cases} \frac{\cot \alpha_{ij} + \cot \beta_{ij}}{2} & j \in N(i), \\ -\sum_{k \in N(i)} L_{\text{cot}}(i, k) & i = j, \\ 0 & \text{else,} \end{cases}$$

$$B(i, j) := \begin{cases} \frac{|t_l| + |t_r|}{12} & j \in N(i), \\ \frac{\sum_{k \in N(i)} |t_k|}{6} & i = j, \\ 0 & \text{else,} \end{cases}$$

$N(i) := \{j : (i, j) \text{ edge}\}$ is the 1-star of i ; $|t_i|$ is the area of the triangle t_i ; t_l and t_r are the triangles that share the edge (i, j) ; t_k , $k \in N(i)$, are the triangles sharing i ; and α_{ij} , β_{ij} are the angles opposite to the edge (i, j) . In the following, we assume that the generalized eigenvalues λ_k of the couple (L_{cot}, B) and the corresponding eigenvectors \mathbf{f}_k are reordered as follows

$$0 = \lambda_1 \leq \dots \leq \lambda_n, \quad L_{\text{cot}} \mathbf{f}_k = \lambda_k B \mathbf{f}_k, \quad k = 1, \dots, n,$$

where the first eigenfunction \mathbf{f}_1 is the constant vector $\mathbf{1}$ and its eigenvalue is null. Finally, we mention that this approximation can be improved by using higher degree

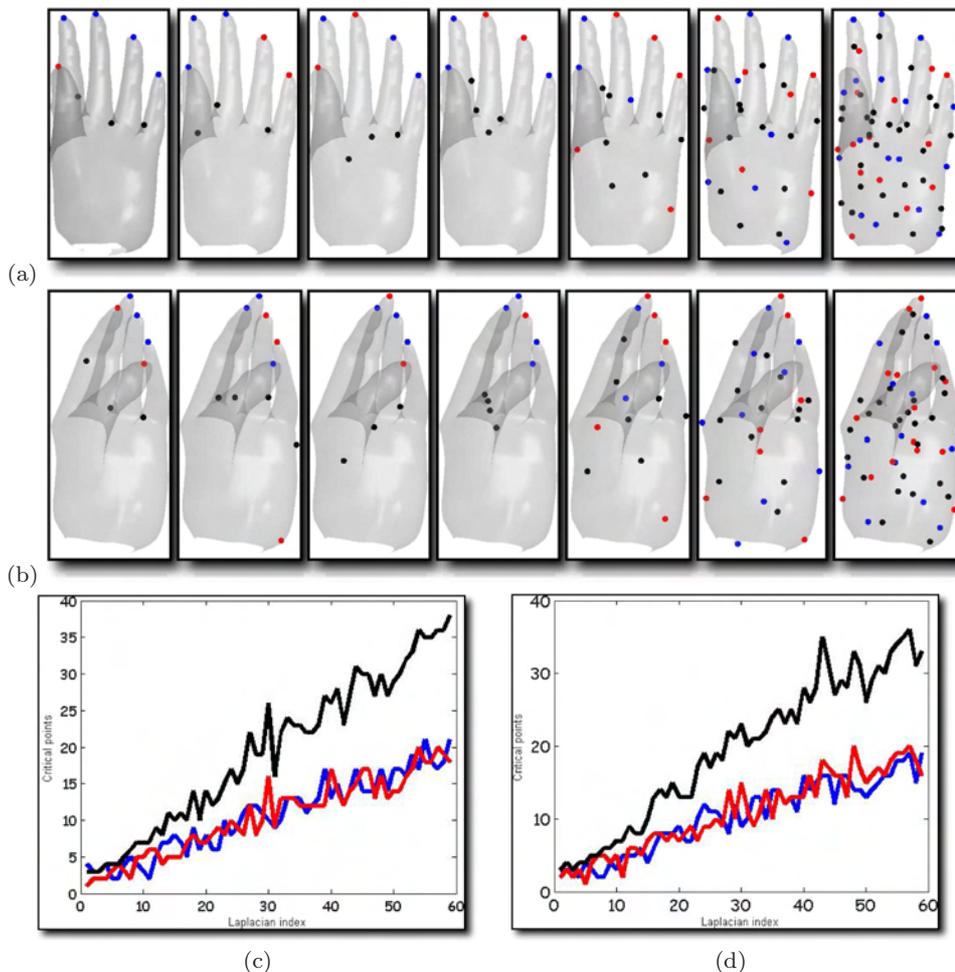


Fig. 4 (a) Location of the critical points of a set of representative Laplacian eigenfunctions \mathbf{f}_k , $k = 2, 3, 4, 6, 11, 31, 51$: red, blue, and black dots locate the maxima, minima, and saddle points, respectively. (b) Critical points of the previous eigenmaps after a deformation of the surface shown in (a). The example highlights that the maxima and minima can switch their classification but remain more stable than saddle points, which are affected by local non-isometric changes of the surface. (c, d) Evolution of the number of critical points. See also Fig. 5.

finite elements; for more details on the cubic case and the properties of the discrete Laplace-Beltrami operators for shape analysis, we refer the reader to [57, 58, 68].

Extraction of the critical points. To define the function $f : S \rightarrow \mathbb{R}$ on the triangulated surface S , the f -values are given at the mesh vertices and linearly interpolated along the edges and the faces of S by using barycentric coordinates. We assume that f is *general* (i.e., $f(\mathbf{p}_i) \neq f(\mathbf{p}_j)$, for each edge (i, j)); this hypothesis guarantees that the critical points of f (i.e., maxima, minima, and saddles) occur only at the mesh vertices and the Euler formula applies (c.f., Equation (3)). These points correspond to the maxima, minima, and saddles of f and are computed by analyzing for each vertex \mathbf{p}_i the distribution of the f -values on the neigh-

borhoods of \mathbf{p}_i [2]. Formally, if we let

$$Lk(i) := \{j_1, \dots, j_k \in N(i) : (j_s, j_{s+1})_{s=1}^k \text{ edges of } S\}$$

be the *link* of i then the *upper link* is defined as

$$Lk^+(i) := \{j_s \in Lk(i) : f(\mathbf{p}_{j_s}) > f(\mathbf{p}_i)\},$$

and the *mixed link* as

$$Lk^\pm(i) := \{j_s \in Lk(i) : f(\mathbf{p}_{j_{s+1}}) > f(\mathbf{p}_i) > f(\mathbf{p}_{j_s}) \text{ or } f(\mathbf{p}_{j_{s+1}}) < f(\mathbf{p}_i) < f(\mathbf{p}_{j_s})\},$$

where $j_{k+1} = j_1$. The *lower link* $Lk^-(i)$, is defined by substituting the inequality “>” with “<” in the definition of the upper link. If $Lk^+(i)$ (resp., $Lk^-(i)$) is empty, then \mathbf{p}_i a *maximum* (resp., *minimum*). If the cardinality of the set $Lk^\pm(i)$ is $2 + 2m_i$, $m_i \geq 1$, then \mathbf{p}_i is classified as a *saddle of multiplicity* m_i . A vertex that

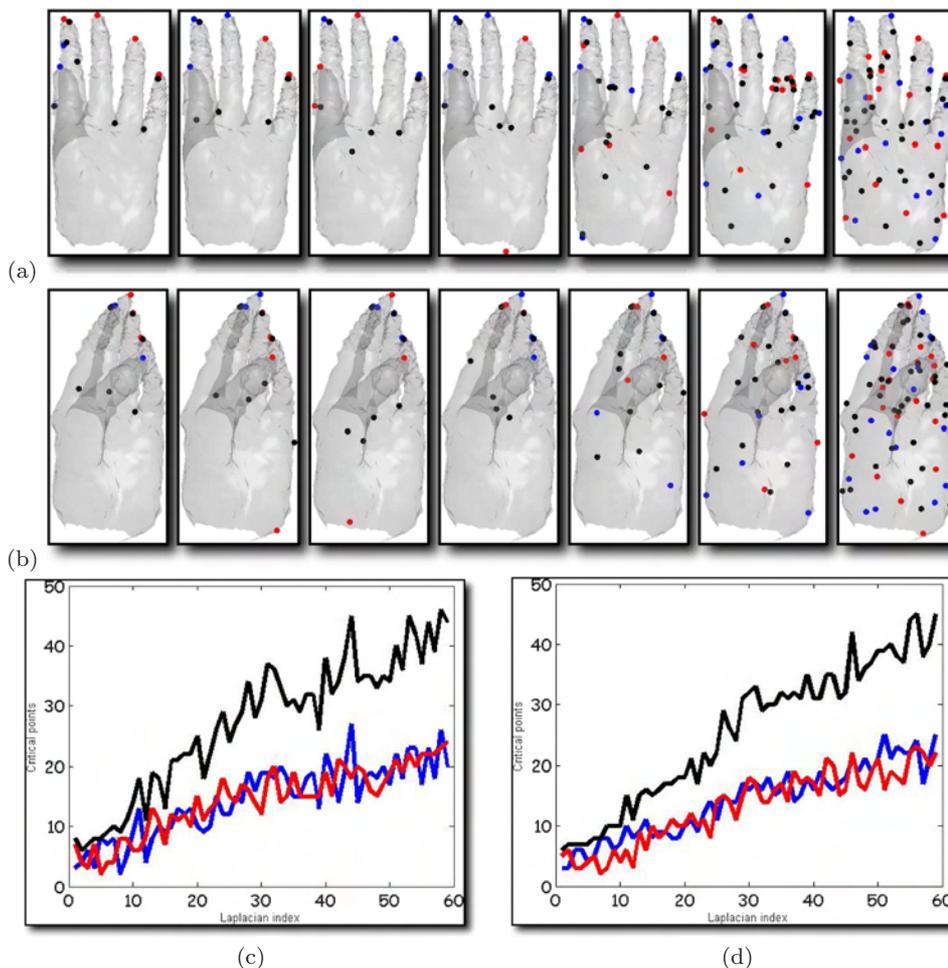


Fig. 5 In these examples, we added a Gaussian noise to the surfaces in Fig. 4 and tested the stability of (a,b) the location and (c,d) the number of the critical points of the corresponding Laplacian eigenfunctions. The example shows that the critical points, and especially the maxima and minima, are not affected by noise perturbation of medium level.

does not fall in the previous classification is called *regular*. Once the vertex-vertex relation has been extracted, the classification procedure requires $O(n)$ -time. Finally, we mention that in the previous definitions we can use a threshold in order to diminish the influence of the noise on the critical point classification (see Fig. 3). For more details on this part, we refer the reader to Section 3.2.

Main properties of the anchor points. The Laplacian eigenfunctions [57, 58, 68], and therefore their critical points, are intrinsically defined by the input shape and are invariant to isometric deformations. In fact, the eigenmaps related to the smallest eigenvalues are smooth and slowly varying functions with non-clustered critical points, which locate surface features and concisely characterize the whole surface. The eigenmaps related to the largest eigenvalues show rapid oscillations and a higher number of critical points. Therefore, the whole set of eigenmaps naturally provides a set of scalar functions organized in a coarse-to-fine hier-

archy; the larger k , the finer the geometric variations captured by the critical points of the Laplacian eigenmap \mathbf{f}_k . The aforementioned properties motivate our choice to consider the eigenmaps as an effective alternative to curvature-based, geodesic and saliency maps. Furthermore, the Laplacian critical points identify anchors useful in a wide class of matching algorithms that build on surface sampling such as [24, 40, 48, 61].

Considering a closed surface S and a Morse Laplacian eigenfunction (i.e., its critical points are not degenerate), the relation between the number of its critical points and the genus of S is given by the Euler formula [2, 41]

$$2(1 - g) = \text{minima} - \text{saddles} + \text{maxima}, \quad (3)$$

where each saddle is counted with its multiplicity m_i , that is, $\sum_{\mathbf{p}_i \text{ saddle}} m_i$. It follows that neglecting a generally low number of non-Morse eigenfunctions and the constant term given by $2(1 - g)$, the sum of the number of maxima and minima is of the same order as the

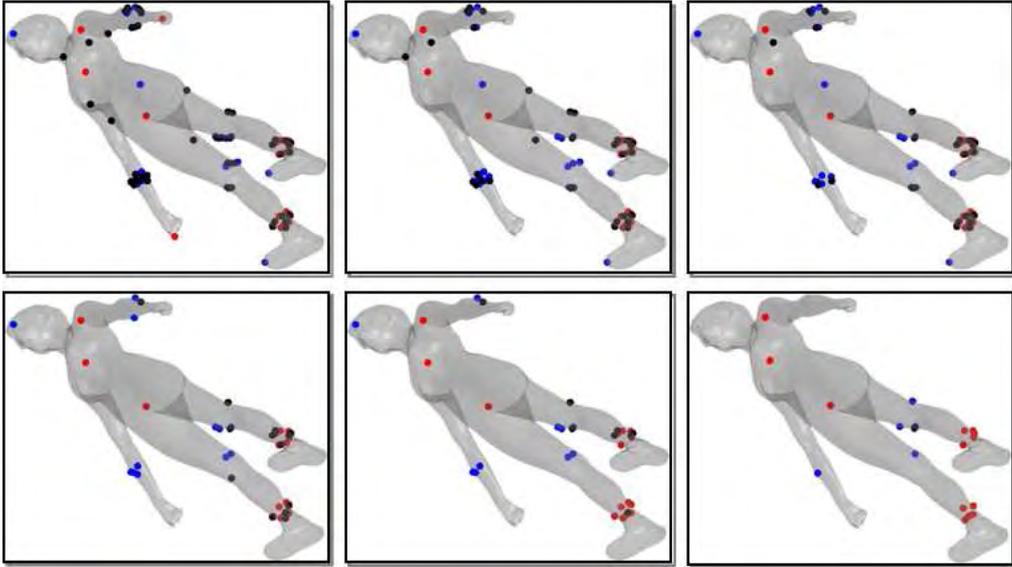


Fig. 6 Simplification of the critical points with respect to a different threshold, which increases from left to right; the initial set of critical points is shown on the left image of the first row.

number of saddle points. Finally, our tests (see Fig. 4 and Fig. 5) have shown that the maxima and minima of the Laplacian eigenfunctions identify protrusions of the input shape and are stable to noise and non-intrinsic deformations.

3.2 Selecting and clustering anchor points

In the following paragraph, we describe a classification of the critical points that is robust to noisy scalar functions and allows us to remove redundant and close critical points.

Robust classification and simplification of the critical points of an arbitrary scalar function.

Our tests have shown that the eigenfunctions related to the eigenvalues of high magnitude have a large number of critical points, which appear close to each other. Furthermore, triangle meshes with an irregular sampling density, connectivity, and small angles might make the computation of the spectrum unstable and provide a set of functions with degenerate critical points (e.g., plateaux regions, monkey saddles). To overcome these drawbacks, we introduce a simplification procedure of the critical points of an arbitrary function $f : S \rightarrow \mathbb{R}$. The idea is to check the changes of the sign of f along the edges of the 1-star of each vertex with respect to a threshold ϵ . If $\epsilon = 0$, then we will get the definition introduced in Section 3.1 and increasing ϵ will remove undesired critical points.

The ϵ -sensitive sign of f along the oriented edge $(\mathbf{p}_i, \mathbf{p}_j)$ is defined as positive if $f(\mathbf{p}_j) - f(\mathbf{p}_i) > \epsilon$; in

this case, we write $f(\mathbf{p}_j) >_\epsilon f(\mathbf{p}_i)$. In an analogous way, the ϵ -sensitive sign of f along the previous edge is considered as negative if $f(\mathbf{p}_j) - f(\mathbf{p}_i) < -\epsilon$; hence, we write $f(\mathbf{p}_j) <_\epsilon f(\mathbf{p}_i)$. Formally, if we let

$$Lk(i) := \{j_1, \dots, j_k \in N(i) : (j_l, j_{l+1})_{s=1}^k \text{ edges of } \mathcal{M}\}$$

be the *link* of i , then the (ϵ -sensitive) *upper link* is defined as

$$Lk^+(i) := \{j_l \in Lk(i) : f(\mathbf{p}_{j_l}) >_\epsilon f(\mathbf{p}_i)\},$$

and the (ϵ -sensitive) *mixed link* as

$$Lk^\pm(i) := \{j_l \in Lk(i) : f(\mathbf{p}_{j_{l+1}}) >_\epsilon f(\mathbf{p}_i) >_\epsilon f(\mathbf{p}_{j_l}) \text{ or } f(\mathbf{p}_{j_{s+1}}) <_\epsilon f(\mathbf{p}_i) <_\epsilon f(\mathbf{p}_{j_s})\}$$

where $j_{k+1} = j_1$. For the definition of the *lower link*, we replace the inequality “ $>_\epsilon$ ” with “ $<_\epsilon$ ” in the definition of the upper link. If $Lk^+(i) = \emptyset$ or $Lk^-(i) = \emptyset$, then \mathbf{p}_i is a ϵ -sensitive *maximum* or *minimum*, respectively. If the cardinality of the set $Lk^\pm(i)$ is $2 + 2m_i$, then \mathbf{p}_i is classified as a *saddle* of *multiplicity* $m_i \geq 1$. In our implementation, the parameter ϵ is proportional to the maximum variation of the f -values along the edges of S , i.e.,

$$\epsilon := 0.2 \max_{(\mathbf{p}_i, \mathbf{p}_j) \text{ edge}} \{|f(\mathbf{p}_i) - f(\mathbf{p}_j)|\}.$$

Examples are shown in Fig. 6; for more details on this approach, we refer the reader to [50]. Other simplification techniques are described in [5, 36].

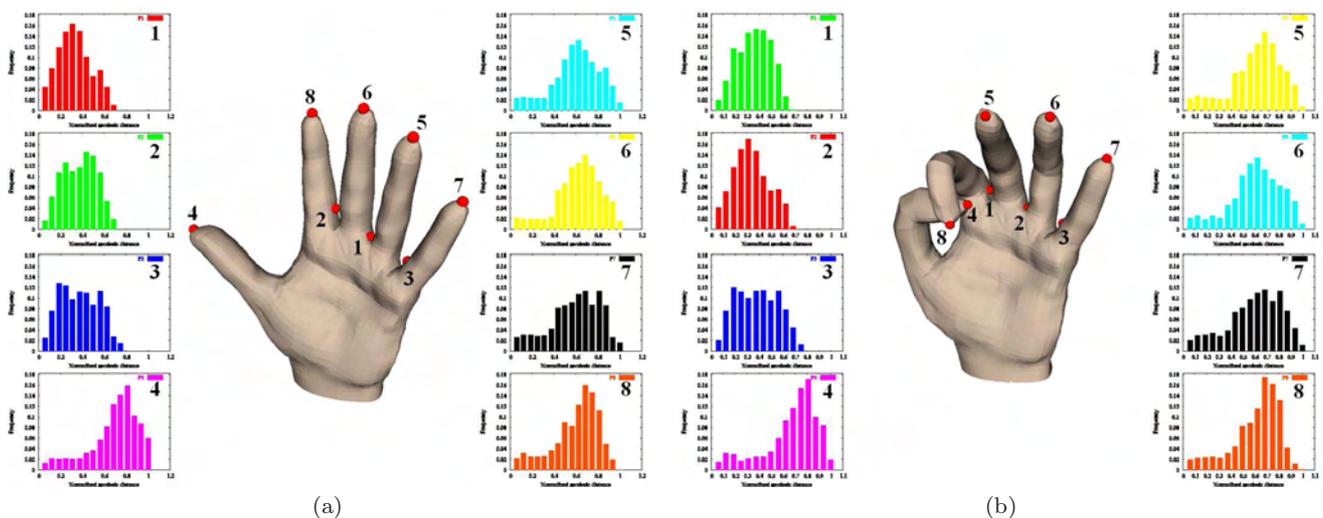


Fig. 7 Histograms of 16 bins approximating the geodesic shape distributions of two surfaces at their anchor points. The surface (b) is the isometrically deformed version of the surface (a), obtained by bending the hand fingers. Despite the isometric deformation, the anchor points of (a) are approximately located in the same regions of those of (b) and the corresponding histograms are similar.

Geodesic simplification of the anchor points of sets of Laplacian eigenfunctions. Let us now suppose that we consider as anchors the critical points of a large number of Laplacian eigenfunctions. In this case, it might happen that the critical points related to different eigenfunctions are closely located and cannot be simplified using the scheme previously introduced. In fact, these critical points are uncorrelated and correspond to different scalar functions. Therefore, we introduce a simplification scheme of the anchors which ensures that two anchors are sufficiently separated on the surface unless they have different geometric meanings. If a set of anchors has geodesic diameter smaller than a threshold g_t , then we keep its geodesic median point and discard the other points from the set. The geodesic median point is the point minimizing the sum of its squared geodesic distances to the other points of the set. The sets of anchors are iteratively computed by using a hierarchical clustering algorithm [54] considering the geodesic distance as metric. The choice of g_t affects the retrieval performance of a matching method. If g_t is too small, then we might have anchor points that are very close to each other, identify the same geometric feature, and degrade the effectiveness of a shape descriptor [45, 61]. To choose g_t , we first apply the FPS algorithm for selecting a number of points that is large enough to statistically describe the surface (see Section 3.3). Then, we set g_t as the minimal geodesic distance among the selected points.

We evaluate the *geodesic distance* between two points of the surface S as the length of the shortest path between the corresponding nodes of an *extended sphere-*

of-influence graph (eSIG, for short) [14]. For the evaluation of the shortest path, we use Dijkstra’s algorithm. The eSIG is the graph constructed from a dense set of points sampled from the triangular surface S . The sphere centered at $\mathbf{p}_i \in S$ with radius given by its distance to the k -nearest neighbor is called the sphere of influence of \mathbf{p}_i . Two distinct points \mathbf{p}_i and \mathbf{p}_j are connected by an edge e_{ij} if the corresponding spheres of influence intersect. Too long edges are discarded as outliers as well as edges, which do not lie on the triangular surface. Each edge e_{ij} is weighted by the Euclidean distance between the connected vertices. Although this approach is not as accurate as those presented in [30, 65], it is fast, reduces the approximation error with respect to considering only the mesh vertices, and can be directly applied to sets of points sampled from any surface representation.

3.3 Statistically controlled sampling of additional points

Our sampling method adds a set of *additional points* to the set Q of (filtered) anchor points previously computed for the triangulated surface S . These new sample points of S are generated through an extension of the *farthest point sampling* (FPS, for short) algorithm [17, 42], which produces sample points evenly distributed on a surface with respect to the anchor points. This distribution ensures a good covering of the surface [40] and is beneficial for increasing the retrieval performance of several matching methods [45, 61]. The farthest point sampling is a fast iterative algorithm,

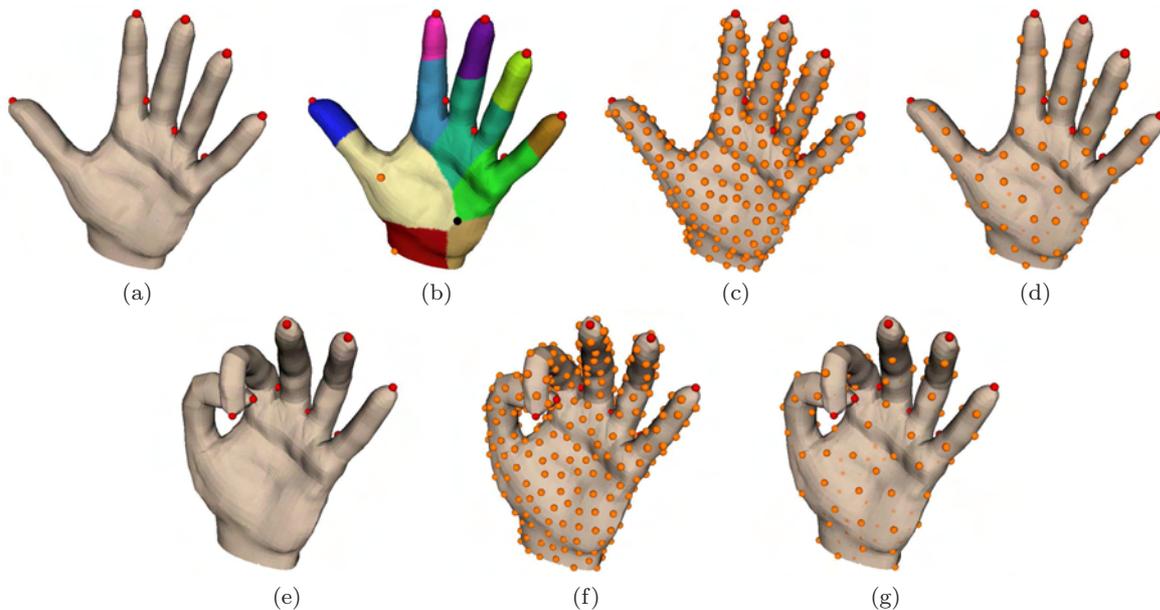


Fig. 8 (a) The model of the hand is sampled starting from a set of 8 anchor points; (b) shows an iteration of our sampling strategy, which adds the farthest and statistically meaningful geodesic Voronoi vertex (darkest point) to the currently selected points. Geodesic Voronoi cells are shown with different colors. From the final set of 346 sample points (c), we select a set of 100 reference points (d), which are considered by our matching algorithms. The hand (a) is isometrically deformed into the hand (e). The 8 anchor points, the 360 additional points, and the 100 reference points are shown in (e), (f), and (g), respectively. Despite the isometric deformation, the anchor points of (e) are located in the same regions of those of (a).

which at each iteration selects the point of the surface that is the farthest from all the currently selected points. Indicating with m the number of selected points, the corresponding cost of the farthest points sampling is $O(m \log m)$.

Once the eSIG of S has been computed, we extract the *geodesic Voronoi cell* of each point \mathbf{q} of Q , i.e., the set

$$GVC(\mathbf{q}, Q) = \{\mathbf{p}_i \in S : g(\mathbf{p}_i, \mathbf{q}) \leq g(\mathbf{p}_i, \mathbf{q}_k), \\ \mathbf{q}_k \neq \mathbf{q}, \mathbf{q}_k \in Q\},$$

which contains the points $\mathbf{p}_i \in S$ whose geodesic distance to the point $\mathbf{q} \in Q$ is not greater than to any other point $\mathbf{q}_k \in Q$. Finally, the (overlapping) partition

$$GVD(S, Q) := \{GVC(\mathbf{q}_i, Q), \mathbf{q} \in Q\}$$

of the surface S into m geodesic Voronoi cells is called *geodesic Voronoi diagram* of the set Q on S and a point of S , which belongs to more than two GVCs, is called *geodesic Voronoi vertex* (GVV, for short). We note that the partition in $GVD(S, Q)$ can be efficiently computed by simultaneously propagating the fronts of the geodesic distance functions from the points of Q to the other points of S till their GVCs completely cover S [42]. Moreover, if a new point \mathbf{p} of S is added to the set Q then the GVD can be updated by recomputing only the GVCs in the neighborhood of \mathbf{p} . In [42], Moenning *et al.* showed that at each iteration of the

FPS the farthest point on S can be efficiently computed as the GVV with maximal geodesic distance to the closest selected point. We extend this algorithm by selecting the farthest GVV $\bar{\mathbf{q}}$ only if its *statistical contribution* (defined in the next paragraph) sufficiently differs from those of its neighboring selected points, i.e., those points whose GVCs contain $\bar{\mathbf{q}}$. At each iteration, if $\bar{\mathbf{q}}$ is selected then we proceed further in the FPS fashion. Conversely, if $\bar{\mathbf{q}}$ is discarded then we consider all the GVVs starting from the second farthest GVV till the nearest GVV, as long as we find a good candidate point. If no good candidates are found, then we stop the iterations and consider the current set of selected points as final.

Statistical contribution of an additional point.

To define the *statistical contribution* of the additional point $\mathbf{p}_a \in Q$ for describing the surface S , we consider the outcome of $g(\mathbf{p}_a, \mathbf{p})$, $\mathbf{p} \in S$, as a random variable X . We then define the *geodesic shape distribution* of S at the point \mathbf{p}_a as the *probability density function* $F(X)$ describing the outcome of X . Thus, $F(X)$ describes the distribution of all the geodesic distances to the point \mathbf{p}_a on the surface S and intuitively gives a statistical description of S as seen from \mathbf{p}_a . We approximate $F(X)$ with a histogram $\mathbf{H}(\mathbf{p}_a, Q)$ of B bins constructed by taking only into account the geodesic

distances between \mathbf{p}_a and the points of Q , i.e.,

$$\left\{ \begin{array}{l} \mathbf{H}(\mathbf{p}_a, Q) = (b_1, \dots, b_B), b_k = \frac{\tilde{b}_k}{\sum_{k=1}^B \tilde{b}_k}; \\ \tilde{b}_k = \left| \left\{ \mathbf{p}_i \in Q : g(\mathbf{p}_a, \mathbf{p}_i) \in \left((k-1) \frac{g_{max}}{B}, k \frac{g_{max}}{B} \right] \right\} \right|, \\ 1 \leq k \leq B; \\ g_{max} = \max_{\mathbf{p}_i, \mathbf{p}_j \in Q} \{g(\mathbf{p}_i, \mathbf{p}_j)\}. \end{array} \right. \quad (4)$$

To make the histogram invariant with respect to scaling, the bins of $\mathbf{H}(\mathbf{p}_a, Q)$ are normalized to g_{max} . Histograms of 16 bins approximating the geodesic shape distributions of two deformed surfaces at their anchor points are shown in Fig. 7.

Sampling algorithm details. Our sampling algorithm aims at enriching the initial set of anchor points with those additional points, which contribute improving the statistical description of the surface S through their geodesic shape distributions. The algorithm is summarized as follows.

Require: The set Q of filtered anchor points of the input surface S .

Ensure: The final set of sample points.

1: For each $\mathbf{q} \in Q$, compute the geodesic Voronoi cell

$$GVC(\mathbf{q}, Q) = \{ \mathbf{p} \in S : g(\mathbf{p}, \mathbf{q}) \leq g(\mathbf{p}, \mathbf{q}_k), \mathbf{q}_k \neq \mathbf{q}, \mathbf{q}_k \in Q \}.$$

2: Compute the geodesic Voronoi diagram $GVD(S, Q)$ generated from the set Q on S , i.e.,

$$GVD(S, Q) := \{GVC(\mathbf{q}, Q), \mathbf{q} \in Q\}.$$

3: Construct the set $GVV(Q)$ of the geodesic Voronoi vertices of $GVD(S, Q)$.

4: Select the Voronoi vertex $\bar{\mathbf{q}} \in GVV(Q)$ with the maximum geodesic distance to the points of Q .

5: Compute $\mathbf{H}(\bar{\mathbf{q}}, Q)$, i.e., the histogram of the geodesic distances between $\bar{\mathbf{q}}$ and the points of Q (see Equation (4)).

6: Compute the set of points whose GVCs contain $\bar{\mathbf{q}}$, i.e.,

$$NN(\bar{\mathbf{q}}) = \{ \mathbf{p} \in Q : \bar{\mathbf{q}} \in GVC(\mathbf{p}, Q) \},$$

and compute $\mathbf{H}(\mathbf{p}, Q)$, $\mathbf{p} \in NN(\bar{\mathbf{q}})$.

7: Using Equation (1), evaluate

$$d_{\chi^2}^{max}(\bar{\mathbf{q}}) = \max_{\mathbf{p} \in NN(\bar{\mathbf{q}})} d_{\chi^2}(\mathbf{H}(\bar{\mathbf{q}}, Q), \mathbf{H}(\mathbf{p}, Q)).$$

8: If $d_{\chi^2}^{max} > d_{\chi^2}^t$, then add $\bar{\mathbf{q}}$ to Q and go to step 2; otherwise, go to step 9.

9: Remove $\bar{\mathbf{q}}$ from $GVV(Q)$.

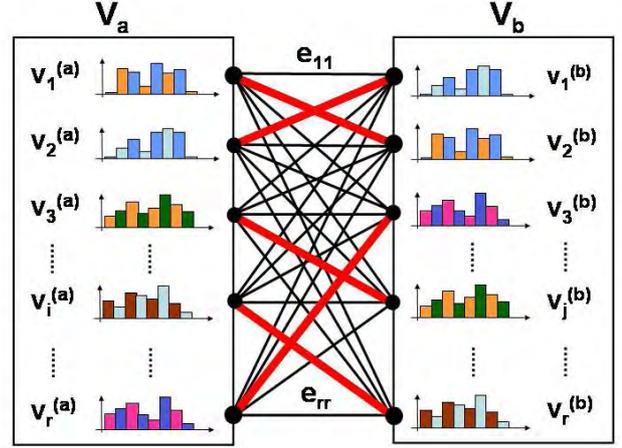


Fig. 9 Bipartite graph matching for comparing sets of PSSDs. Let V_a and V_b be two sets of PSSDs of cardinality r . Each PSSD of the set V_a is connected *via* r edges with all the PSSDs of V_b . An edge e_{ij} is weighted with the L_1 -distance between $\mathbf{v}_i^{(a)} \in V_a$ and $\mathbf{v}_j^{(b)} \in V_b$. A matching between V_a and V_b is that set of edges (thickened edges) minimizing the sum of the edge weights.

10: If $GVV(Q)$ is empty, then go to step 11; otherwise, go to step 4.

11: Consider Q as the final set of sample points.

The number of selected points depends on the surface geometry and on the threshold $d_{\chi^2}^t$. The choice of $d_{\chi^2}^t$ determines the precision at which the histograms $\mathbf{H}(\mathbf{q}, Q)$ approximate the actual geodesic shape distribution at the selected points $\mathbf{q} \in Q$. On our databases, we have set $d_{\chi^2}^t = 10^{-5}$. Using the maxima and minima of the first four non-trivial Laplacian eigenmaps leads to an average of 400 points per object. Fig. 8 shows the results of our sampling method applied on two isometrically deformed versions of a hand. For the models shown in Fig. 8(c,f), the final sets of sample points have cardinality 346 and 360, respectively.

Our sampling method ensures that at each iteration the generated sample points are evenly distributed on the surface with respect to the anchor points and give statistically meaningful contributions. Thus, it implicitly defines levels of detail in the sampled point set, which are refined at each iteration. Fig. 8(d,g) show the first 100 sample points, including the anchors, generated by our sampling method applied on two deformed surfaces. These points are considered as reference points for matching those surfaces, as described in Section 4.

4 Matching point-based statistical shape descriptors

Using the set of sample points Q , we compute a set of point-based statistical shape descriptors that describe the surface through histograms of geodesic distances (see Section 4.1). To compactly describe the surface, we only consider a set of shape descriptors associated to the set of reference points Q_r in Q . Then (see Section 4.2), the dissimilarity between two 3D shapes is computed by comparing the corresponding sets of descriptors with a bipartite graph matching or a reordering of their components.

4.1 Point-based statistical shape descriptor

Similarly to [61], given a set of feature points

$$Q = \{\mathbf{q}_i, i = 1, \dots, M\} \subseteq S,$$

we aim at compactly describing S with a set of *statistical shape descriptors* (SSDs, for short) constructed from a set of r *reference points* belonging to the set $Q_r = \{\mathbf{q}_{i_k} \in Q, k = 1, \dots, r\} \subseteq Q$. The set Q_r is constructed to include all the anchor points and the additional points of Q (in sampling order) that are necessary to reach a fixed cardinality $r := |Q_r|$. To define the descriptor of a point $\mathbf{q}_{i_k} \in Q_r$, we consider the histogram of B bins approximating the *geodesic shape distribution* of S at the point \mathbf{q}_{i_k} , as defined in Equation (4). Furthermore, we compute other information, which improves the matching of the histograms. Thus, to each point \mathbf{q}_{i_k} in Q_r we associate a *point-based statistical shape descriptor* (PSSD, for short), i.e., a vector

$$\mathbf{v}(\mathbf{q}_{i_k}) = (b_1, \dots, b_B, v_{B+1}, v_{B+2}, v_{B+3}, v_{B+4}),$$

containing the following information:

- $(b_1, \dots, b_B) = \mathbf{H}(\mathbf{q}_{i_k}, Q)$, i.e., the histogram of bins approximating the geodesic shape distribution of S at \mathbf{q}_{i_k} (see Equation (4)). It statistically describes the intrinsic properties of S as seen from \mathbf{q}_{i_k} .
- $v_{B+1} := b_{max}$ with $b_{max} := \max_{1 \leq j \leq B} b_j$. v_{B+1} improves the matching effectiveness of $\mathbf{H}(\mathbf{q}_{i_k}, Q)$. In fact, large values of b_{max} indicate that the geodesic distances are mostly concentrated in one range, which is typical of external points (i.e., points with high protrusions) that are far from the geodesic barycenter of the surface.
- $v_{B+2} = b_{max} - b_{min}$, where $b_{min} = \min_{1 \leq j \leq B} b_j$. Similarly to v_{B+1} , v_{B+2} serves to improve the matching effectiveness of $\mathbf{H}(\mathbf{q}_{i_k}, Q)$.

- $v_{B+3} = \frac{1}{M} \sum_{\mathbf{q}_j \in Q} |\hat{g}(\mathbf{q}_{i_k}, \mathbf{q}_j)|^2$, where $\hat{g}(\mathbf{q}_{i_k}, \mathbf{q}_j)$ is the geodesic distance between \mathbf{q}_{i_k} and \mathbf{q}_j normalized with respect to the maximal geodesic distance g_{max} . Intuitively, it indicates the protrusion of S at \mathbf{q}_{i_k} , i.e., how far from the geodesic barycenter of S the point \mathbf{q}_{i_k} is.
- $v_{B+4} = \sigma(\mathbf{q}_{i_k}) = \frac{3\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3}$, where $\lambda_1 \leq \lambda_2 \leq \lambda_3$ are the eigenvalues corresponding to the principal components of the geodesic Voronoi cell of \mathbf{q}_{i_k} generated by the set Q on S . The value $\sigma(\mathbf{q}_{i_k}) \in [0, 1]$ is called *surface variation* [52] and estimates how much the surface locally deviates from the tangent plane within the geodesic Voronoi cell of \mathbf{q}_{i_k} . Values of $\sigma(\mathbf{q}_{i_k})$ close to zero indicate that the portion of S is almost planar in the geodesic Voronoi cell of \mathbf{q}_{i_k} .

The choice of r influences the efficiency of the matching algorithm and depends on the database. The value r should be sufficient to describe all the objects of the database and to ensure good retrieval performances. Starting from the maximum number of anchor points detected in the database, we iteratively increase r till a good tradeoff between the retrieval effectiveness and execution time of the matching algorithm is achieved. For details on this part, we refer the reader to Section 4.2. By considering the reference points in sampling order, at each value of r the even distribution of the points is ensured without redundancies in their statistical contributions. Fig. 8(d,g) shows 100 reference points selected on two isometrically deformed models of a hand. Since the computed PSSD depends on the accuracy of the calculated geodesic distances, the presence of geometric and topological noise may diminish its quality. To alleviate this problem, various techniques for computing geodesics directly on point clouds [39, 60] can be used. The aforementioned methods are robust to both noise and outliers.

4.2 Matching statistical shape descriptors

Let $Q_a = Q_r(S_a) = \{\mathbf{q}_1, \dots, \mathbf{q}_r\} \subseteq S_a$ be the set of r reference points sampled on a surface S_a ; then, we compute a set of r PSSDs

$$V_a = V(Q_a) = \{\mathbf{v}(\mathbf{q}_1), \dots, \mathbf{v}(\mathbf{q}_r) : \mathbf{q}_j \in Q_a, j = 1, \dots, r\}.$$

Given a new surface S_b , the dissimilarity between S_a and S_b is estimated by comparing their corresponding sets of PSSDs V_a and $V_b := V(Q_b)$ with $Q_b := Q_r(S_b)$. Since the anchor points of Q_a and Q_b have an arbitrary order, to correctly match V_a with V_b we firstly need to find a good alignment between their elements. This problem implies searching among all the permutations



Fig. 10 Reference objects of our database. Each reference object represents a class of objects of the database. Each class contains a set of nearly isometrically deformed versions of its reference object.

of r indices the one that minimizes the dissimilarity between V_a and V_b . Formally, let $\pi : \{1, \dots, r\} \rightarrow \{1, \dots, r\}$ be a permutation of the set of indices $\{1, \dots, r\}$. The problem of comparing V_a and V_b can be stated as finding the permutation π , which minimizes the distance function

$$d_\pi(V_a, V_b) = \sum_{i=1}^r \|\mathbf{v}_i^{(a)} - \mathbf{v}_{\pi(i)}^{(b)}\|_1,$$

where $\mathbf{v}_i^{(a)}$ is the i th PSSD of V_a and $\mathbf{v}_{\pi(i)}^{(b)}$ is the PSSD of V_b related to $\pi(i)$. Thus, the permutation π establishes the correspondence between the elements of V_a and V_b . Then, the dissimilarity value between V_a and V_b is given by

$$\delta(V_a, V_b) = \min_{\pi \in \Pi_r} d_\pi(V_a, V_b), \quad (5)$$

where Π_r is the set of all permutations of $\{1, \dots, r\}$. This approach implies a combinatorial search over all permutations Π_r . The computation of $\delta(V_a, V_b)$ is time-consuming because the size of Π_r is $r!$. For this reason, the parameter r is fundamental to bound the computational complexity of the problem as well as the search of the permutation that solves (5). In the following paragraphs, to make the problem of matching two sets of PSSDs tractable we propose two simplifications: bipartite graph matching [32] and comparing reorderings of PSSDs based on Fiedler vectors [12].

Bipartite graph matching. A complete undirected bipartite graph $G = (V_{ab}, E)$ is constructed by taking as set of vertices $V_{ab} := V_a \cup V_b$. Each element $\mathbf{v}_i^{(a)}$ of V_a is connected to every element $\mathbf{v}_j^{(b)}$ of V_b through an edge e_{ij} , which is added to the set of edges E of G (see Fig. 9). Each edge $e_{ij} \in E$ is then weighted with the L_1 -distance between $\mathbf{v}_i^{(a)}$ and $\mathbf{v}_j^{(b)}$.

A *matching* in G is the set of edges $E_M \subseteq E$, in which no two edges share a common vertex of V_{ab} . We aim at finding the *minimum weight perfect matching*,

which is the minimum weight matching with cardinality $|E_M| = |V_{ab}|/2$ (e.g., the set of thickened edges in Fig. 9). This problem can be solved in $O(|V_{ab}|^2 \log |V_{ab}|)$ time with Edmonds' blossom algorithm as in [13]. For our tests, we used the implementation presented in [13]. Recently, Schwartz *et al.* [63] presented an algorithm achieving a running time of $O(|V_{ab}|^2)$ under the assumption that the edge weights are integers and uniformly distributed. The set E_M induces a bijection π^* of $\{1, \dots, r\}$ between the sets V_a and V_b . Therefore, the dissimilarity between V_a and V_b is estimated with the function

$$\delta^*(V_a, V_b) = \sum_{e_{ij} \in E_M} w(e_{ij}) = \sum_{i=1}^r \|\mathbf{v}_i^{(a)} - \mathbf{v}_{\pi^*(i)}^{(b)}\|_1,$$

which sums the weights of the edges in E_M . Even though the histograms could be compared with several different metrics, our experimental comparisons on different databases of isometric objects have shown that the L_1 -distance is the most effective metric for our PSSDs.

Fiedler vector based matching. The *Fiedler vector* is the eigenvector corresponding to the first non-null eigenvalue of the Laplacian matrix of a weighted graph [18]. Intuitively, the components of the Fiedler vector give rise to an embedding of the nodes of a graph on a straight line such that the sum of the weighted and squared edge lengths is minimal. In practice, the Fiedler vector can be used for ordering a set of entities, which can be related in form of a graph. Its usefulness has been demonstrated in several applications such as spectral graph partitioning [55] and drawing [31], mesh streaming [26] and processing [71].

Given the set of PSSDs V_a , we consider the complete graph having the PSSDs $\mathbf{v}_i^{(a)} \in V_a$ as nodes, which are connected to all the other PSSDs $\mathbf{v}_j^{(a)} \in V_a, i \neq j$ through edges weighted with $e_{ij} := \exp(-g(\mathbf{v}_i^{(a)}, \mathbf{v}_j^{(a)}))$. Here, the exponential function is used to smooth out the effect of large geodesic distances between reference

Matching method	Standard Classification					Finest Classification				
	P_{50}	P_{100}	BEP	RP	NN	P_{50}	P_{100}	BEP	RP	NN
OurBGM 100	99.0	92.1	91.9	87.2	100	99.4	95.8	95.2	93.3	100
Reuter06	98.2	90.7	89.1	87.8	98.9	98.7	95.7	95.2	94.0	98.9
OurFV 100	98.1	89.0	90.0	81.9	98.9	98.7	93.9	96.2	88.1	98.9
RuSa08 256	97.2	91.5	94.0	85.5	96.7	97.5	94.3	95.6	90.0	96.7
RuSa08 100	93.7	87.4	92.1	77.1	92.2	94.1	90.2	94.3	81.2	92.2
Osada02 G2 FPS	91.2	83.7	84.3	74.9	91.1	91.9	87.4	89.8	79.7	91.1
Hamza03 FPS	90.7	82.5	83.3	74.3	88.9	91.2	87.1	88.6	80.5	88.9
Memoli05	56.5	50.3	63.8	34.2	47.8	57.3	52.7	64.5	36.7	47.8

Table 1 Comparison of the retrieval effectiveness of different shape matching methods by using two different classifications: the standard one shown in Fig. 10 and the finest one, in which the class of 8 hands of two different humans is split in two different classes of 4 hands. The following retrieval effectiveness measures are reported in percentages: \bar{P}_{50} and \bar{P}_{100} = average precision over recall range 50% and 100%, respectively; RP=R-precision (first tier); BEP=Bull’s Eye Performance (second tier); NN=nearest neighbor.

points. For the computation of the Fiedler vector, we define the Laplacian matrix associated to the proximity graph as the $r \times r$ symmetric matrix whose entries are $-e_{ij}$, if $i \neq j$, and $d_{ii} := \sum_{j=1}^r e_{ij}$ if $i = j$. Then, the decreasing reordering of the components of the Fiedler vector are used to permute the elements of V_a . The re-ordered set of PSSDs is stored as a single feature vector

$$\mathbf{f}_a = (\mathbf{v}_{\pi_o(1)}^{(a)}, \dots, \mathbf{v}_{\pi_o(r)}^{(a)})$$

of $r \cdot (B + 4)$ entries, where $\pi_o : \{1, \dots, r\} \rightarrow \{1, \dots, r\}$ is the bijection induced by the ordering of the Fiedler vector components. The dissimilarity between V_a and V_b is then computed as the L_1 -distance between the re-ordered feature vectors \mathbf{f}_a and \mathbf{f}_b , i.e.,

$$\delta_L^*(V_a, V_b) = \|\mathbf{f}_a - \mathbf{f}_b\|_1.$$

This approach is faster than the bipartite matching and leads to approximately similar effectiveness values. The Fiedler vector of a $r \times r$ matrix can be computed in time $O(r \log r)$. This computation can be performed off-line during the production of the feature vectors, which makes this approach suitable for online retrieval applications. Indeed, the on-line comparison of two feature vectors can be efficiently performed by using the L_1 -distance or other suitable metrics.

5 Results

We evaluate our matching methods and our sampling strategy on a database containing a collection of 90 3D models, some of them kindly provided by Ron Kimmel (Technion-Israel), and some taken from the online database of R.W. Summer and J. Popovic [64]. These 3D models are subdivided in 17 classes of nearly isometric objects (see Fig. 10). Indeed, each class contains a set of nearly isometrically deformed versions of one reference object.

We perform a query for each object of this database. The dissimilarity values of the query objects with respect to all the other objects of the database are computed by using both the bipartite graph matching algorithm (*OurBGM*) and the approach based on the Fiedler vector (*OurFV*). Both methods match 100 PSSDs of 20 components per object, which correspond to the value of r and B in Section 4. We compared our two methods with the following techniques:

- *Reuter06* [58]. In the comparison, we used only the first 10 eigenvalues of the Laplace-Beltrami operator, computed with the finite element method described in Section 3.1;
- *RuSa08* [61]. Aside we report the number of sample points (local descriptors) considered by the method;
- *Hamza03 FPS* [24]. In this case, the descriptor is extended by using the FPS to select 1024 reference points;
- *Osada02 G2 FPS* is an extension of the method of Osada *et al.* in [48], which compares single global histograms of the geodesic distances between 1024 points sampled with the FPS;
- *Memoli05* [40] has been described in Section 2.2.

Table 1 reports a comparison of some well known measures of retrieval effectiveness [10, 66] of these methods. \bar{P}_{50} and \bar{P}_{100} are the average precision over recall range 50% and 100%, respectively. RP indicates the *R-precision* (first tier), BEP is the *Bull’s Eye Performance* (second tier), NN is the *nearest neighbor* which indicates the recognition ability, i.e., the ratio of the query objects that are successfully recognized. Indeed, the query object is judged as recognized if its class matches the class of the nearest neighbor. For the methods *OurBGM(FV)* and *RuSa08*, we also report the number of SSDs considered in the matching. A similar comparison is performed by calculating the precision *versus* recall diagram shown in Fig. 11. The rows in Table 1 and the plots in Fig. 11 are ordered with respect to \bar{P}_{50} . The

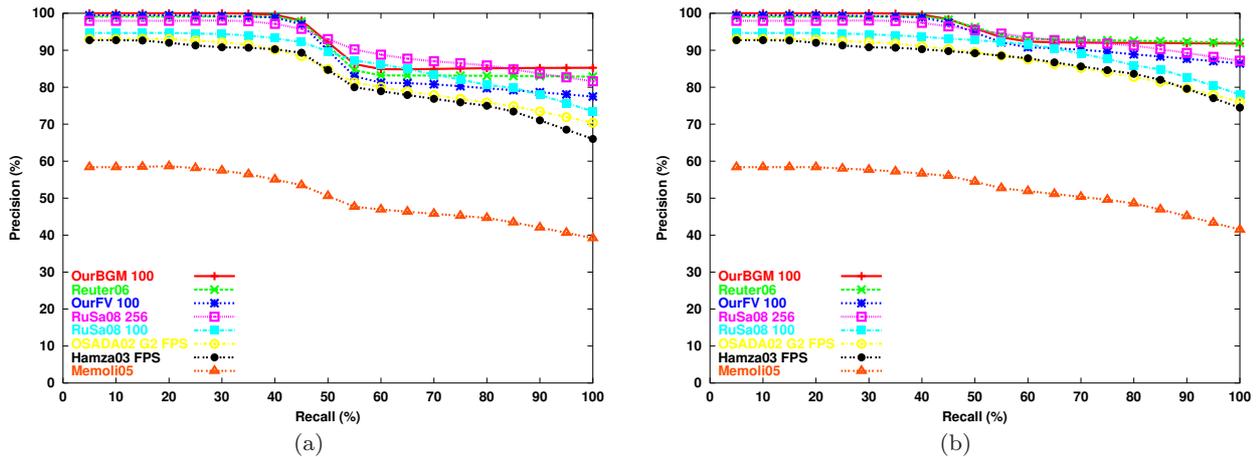


Fig. 11 Comparison of the retrieval effectiveness of different matching methods executed on the database of deformed objects shown in Fig. 10. Two classifications are considered: (a) the standard classification shown in Fig. 10 and (b) the finest classification, in which the class of 8 hands of two different humans is split in two different classes of 4 hands.

Matching method	Standard Classification					Finest Classification				
	P_{50}	P_{100}	BEP	RP	NN	P_{50}	P_{100}	BEP	RP	NN
OurBGM 100	99.0	92.1	91.9	87.2	100	99.4	95.8	95.2	93.3	100
OurFV 100	98.1	89.0	90.0	81.9	98.9	98.7	93.9	96.2	88.1	98.9
RuSa08 256 FPS	97.2	91.5	94.0	85.5	96.7	97.5	94.3	95.6	90.0	96.7
RuSa08 100 OURS	95.7	91.2	95.0	85.8	96.7	97.2	93.8	95.6	89.8	96.7
RuSa08 100 FPS	93.7	87.4	92.1	77.1	92.2	94.1	90.2	94.3	81.2	92.2
RuSa08 100 UNIF	77.2	65.1	69.2	53.5	75.6	78.8	66.8	68.9	55.5	75.6
Osada02 G2 OURS	91.4	82.3	83.1	73.4	92.2	92.2	86.4	89.0	78.5	92.2
Osada02 G2 FPS	91.2	83.7	84.3	74.9	91.1	91.9	87.4	89.8	79.7	91.1
Osada02 G2 UNIF	84.6	75.9	80.5	65.7	83.3	85.6	79.4	85.0	69.3	83.3
Hamza03 OURS	91.4	81.3	81.3	70.7	92.2	92.0	85.8	86.9	76.6	92.2
Hamza03 FPS	90.7	82.5	83.3	74.3	88.9	91.2	87.1	88.6	80.5	88.9
Hamza03 UNIF	90.4	79.4	81.8	67.8	88.9	90.9	83.9	87.1	74.0	88.9

Table 2 Comparison of the retrieval effectiveness of different shape matching methods applied on sets of sample points obtained with different sampling strategies, such as: our sampling method explained Section 3 (*OURS*), the FPS, and the *uniform sampling (UNIF)*. The following retrieval effectiveness measures are reported in percentages: P_{50} and P_{100} =average precision over recall range 50% and 100%, respectively; RP=*R-precision* (first tier); BEP=*Bull's Eye Performance* (second tier); NN=*nearest neighbor*. These measurements are computed by considering two different classifications of our database: the standard one shown in Fig. 10, and the finest one, in which the class of 8 hands of two different humans is split in two classes of 4 hands.

comparison is performed considering two different classifications: the standard one shown in Fig. 10 and the finest one, in which the class of 8 hands of two different humans is split in two different classes of 4 hands, each per human being. The finest classification is intended to show the sensitivity of the matching methods to small changes in the geometry like those present in the hands of different human beings. The value of NN=100% in Table 1 shows that our method *OurBGM* is able to recognize all the objects of the databases. Furthermore, it performs better than the other tested methods. The performance gap with respect to the methods *Reuter06* and *OurFV* is small and depends on the classification of our database. Indeed, considering the finest classification the method *Reuter06* has a slightly better value of RP than *OurBGM*. The contribution of our methods

in terms of retrieval effectiveness is particularly evident with respect to *RuSa08 100*, where the use of our sampling technique approximately improves the effectiveness of 10% by considering the same number of SSDs in the comparison.

We evaluate the benefits of our sampling strategy presented in Section 3 by applying it on the methods *RuSa08 100*, *Osada02 G2* and *Hamza03*. Table 2 reports the retrieval evaluation measures of these methods applied on sets of sample points generated by using our sampling method (*OURS*), the FPS, and the uniform sampling (*UNIF*). The use of our sampling method improves the retrieval effectiveness of these methods, in particular their recognition ability. Indeed, for the method of *RuSa08* it allows achieving with 100 point samples the same NN value as considering 256

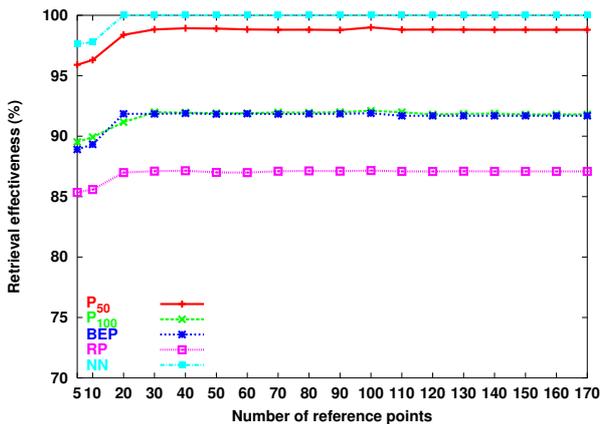


Fig. 12 Retrieval effectiveness measures of our method using BGM considering different number of reference points. This test is performed with the standard classification shown in Fig. 10.

of them. This fact considerably increases the efficiency of the *RuSa08* method, since its time complexity is $O(N^2 \log N)$, where N is the number of histograms to be matched with the χ^2 distance.

The retrieval tests are run on a Windows XP Professional system running on an Intel Pentium 4 2.80 GHz with 2GB of RAM. The average execution time per matching of the method *RuSa08 100 OURS* is 0.015s, almost 27 times less than *RuSa08 256*, i.e., 0.41s. Our method using the BGM on 100 PSSDs (*OurBGM 100*, for short) performs slightly better than the method *RuSa08 100 OURS*, because the PSSDs are compared with the L_1 -distance instead of χ^2 distance. Indeed, its average running time per matching is about 0.0106s. Since the alignment using the Fiedler vector is run off-line, our method *OurFV 100* is much faster than *OurBGM 100*. The on-line matching with the algorithm *OurFV 100* consists in comparing feature vectors of 2000 real numbers with the L_1 -distance, i.e., 20×100 PSSDs at once.

However, our method performs well already when considering few reference points. Fig. 12 shows different retrieval effectiveness measures of our method using BGM with respect to the number of reference points considered in the matching. The retrieval performances of our method stabilize already at 20 reference points, although the peak is reached at 100 points. This allows us to significantly decrease the execution time of our method BGM without sacrificing its effectiveness. While in [37] the input parameters need to be accurately tuned in order to avoid wrong correspondences between points or an over-fitting in the embedding, our approach reorders the nodes of the graph associated to each point-based statistical shape descriptor according to the Fiedler vector of a symmetric matrix. Further-

more, computing only one eigenvector is much faster than calculating the singular value decomposition as done in [37].

The contribution of the additional intrinsic quantities used for augmenting the histogram of geodesics in Section 4.1 is approximately 1.6% and 3.8% RP, for the standard and the finest classification respectively. Of these improvements, 50% can be roughly attributed to the values v_{B+1}, v_{B+2} , 20% to v_{B+3} , and 30% to v_{B+4} . However, on some collections of 3D shapes with topological and geometrical noise, our methods might mismatch local descriptors. This problem can be alleviated either by improving our local descriptors with further intrinsic geometry measures or by considering geometrical constraints between local descriptors. These constraints can also serve for improving the search of the optimal solution of our matching problem by applying algorithms like branch-and-bound heuristics [9] used in combinatorial optimization or the bag-of-features approach [6, 47] used in computer vision.

6 Conclusions and future work

We presented a matching method for 3D shapes, which is invariant with respect to isometric deformations. This method comprises a new technique for sampling 3D shapes and two matching algorithms using point-based statistical shape descriptors. Our sampling strategy generates a set of geometrically meaningful sample points by using the critical points of the Laplacian eigenmaps as anchor points and combining the farthest point sampling with statistical criteria. We only consider significant points that are not statistically redundant for describing the surface. However, this approach can be enhanced by considering database-dependent criteria like the tf-idf weighting used in text mining [6]. A set of PSSDs is constructed from the set of sample points. Using augmented histograms of geodesic distances, the PSSDs describe the surface as seen from their corresponding sample points. The dissimilarity between two 3D shapes is computed by matching the corresponding sets of PSSDs either with bipartite graph matching or by comparing reorderings of the PSSDs according to the Fiedler vector of the Laplacian matrix of the corresponding proximity graphs.

Both variants of the proposed spectral-driven isometry-invariant matching showed good retrieval effectiveness on a test database of 90 objects stored in different postures; in fact, it was able to recognize all or almost all objects in the database. Thus, it is an effective method for classifying and recognizing objects deformed with isometric transformations, e.g., non-rigid and articulated objects in different postures. The retrieval effec-

tiveness of our methods were compared against other isometry-invariant matching techniques. Our first matching technique based on a bipartite graph matching performed better than the other tested methods and recognized 100% of the objects of the database. The second approach based on the Fiedler vector is slightly less effective than the first one; however, it is more efficient and more suitable for online retrieval applications.

Our framework can be extended to address the problem of partial matching of two 3D shapes [6] by using either subgraph matching approaches or geometric hashing methods for finding similar surface regions identified by salient reference points [21]. Another interesting future application of our method is detecting intrinsic symmetries of non-rigid 3D shapes [49, 56], where similar local shape descriptors are used [56]. Other future work includes extending the proposed technique to point-sampled surfaces. In fact, most of its building blocks have already been treated in the context of point-based graphics. Actually, the main open problem is the stable computation of the critical points of a scalar function defined on a point set.

Acknowledgment

At the University of Konstanz this research was supported by the DFG Graduiertenkolleg 1042 “*Explorative Analysis and Visualization of Large Information Spaces*”. Giuseppe Patanè and Michela Spagnuolo have been partially supported by the FOCUS K3D Coordination Action and the Italy-Israel bilateral project SHALOM. We also acknowledge the use of the Edmonds’ blossom algorithm, kindly provided by W. Cook (University of Texas at Austin). Models are courtesy of Ron Kimmel (Technion-Israel), Robert W. Sumner (ETH-Zurich), and Jovan Popovic (MIT-Boston) [64]. We thank the anonymous reviewers for their comments and suggestions.

References

1. S. Baloch, H. Krim, I. Kogan, and D. Zenkov. Rotation invariant topology coding of 2D and 3D objects using morse theory. In *Proc. of the IEEE Conference on Image Processing*, pages 796–799, 2005.
2. T. Banchoff. Critical points and curvature for embedded polyhedra. *Journal of Differential Geometry*, 1:245–256, 1967.
3. S. Biasotti, B. Falcidieno, L. De Floriani, P. Frosini, D. Giorgi, C. Landi, L. Papaleo, and M. Spagnuolo. Describing shapes by geometric-topological properties of real functions. *ACM Computing Surveys*, December 2008. In press.
4. S. Biasotti, S. Marini, M. Spagnuolo, and B. Falcidieno. Sub-part correspondence by structural descriptors of 3D shapes. *Computer-Aided Design*, 38(9):1002–1019, 2006.
5. P.-T. Bremer, H. Edelsbrunner, B. Hamann, and V. Pascucci. A topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 10(4):385–396, 2004.
6. A. M. Bronstein, M. M. Bronstein, Y. Carmon, and R. Kimmel. Partial similarity of shapes using a statistical significance measure. *IPSJ Transactions on Computer Vision and Applications (CVA)*, 1:105–114, 2009.
7. A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Efficient computation of isometry-invariant distances between surfaces. *SIAM Journal on Scientific Computing*, 28(5):1812–1836, 2006.
8. R. Brunelli and O. Mich. Histograms analysis for image retrieval. *Pattern Recognition*, 34(8):1625–1637, 2001.
9. M. J. Brusco and S. Stahl, editors. *Branch-and-Bound Applications in Combinatorial Data Analysis*. Springer-Verlag, 2005.
10. B. Bustos, D. A. Keim, D. Saupe, T. Schreck, and D. V. Vranić. Feature-based similarity search in 3D object databases. *ACM Computing Surveys*, 37(4):345–387, 2005.
11. U. Castellani, M. Cristani, S. Fantoni, and V. Murino. Sparse points matching by combining 3D mesh saliency with statistical descriptors. *Computer Graphics Forum*, 27(2):643–652, 2008.
12. F. R. K. Chung. *Spectral graph theory*. American Mathematical Society, 1997.
13. W. Cook and A. Rohe. Computing minimum-weight perfect matchings. *Journal on Computing*, 11(2):138–148, 1999.
14. T. Darom, M. Ruggeri, D. Saupe, and N. Kiryati. Compression of textured surfaces represented as surfel sets. *Signal Processing: Image Communication*, 21(9):770–786, 2006.
15. A. Elad and R. Kimmel. On bending invariant signatures for surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1285–1295, 2003.
16. M. Elad, A. Tal, and S. Ar. Content based retrieval of vrmf objects: an iterative and interactive approach. In *Proc. of the Eurographics Workshop on Multimedia*, pages 107–118, 2002.
17. Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi. The farthest point strategy for progressive

- image sampling. *IEEE Transactions on Image Processing*, 6(9):1305–1315, 1997.
18. M. Fiedler. A property of eigenvectors of non-negative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25:619–633, 1975.
 19. A. Fomenko and T. L. Kunii. *Topological Modelling for Visualization*. Springer Verlag, 1997.
 20. T. Funkhouser and P. Shilane. Partial matching of 3D shapes with priority-driven search. In *Proc. of the Symposium on Geometry Processing*, pages 131–142, 2006.
 21. R. Gal and D. Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics*, 25(1):130–150, 2006.
 22. R. Gal, A. Shamir, and D. Cohen-Or. Pose-oblivious shape signature. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):261–271, 2007.
 23. N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *Proc. of the Symposium on Geometry Processing*, page 197, 2005.
 24. A. B. Hamza and H. Krim. Geodesic object representation and recognition. In *International Conference on Discrete Geometry for Computer Imagery*, pages 378–387, 2003.
 25. M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *ACM Siggraph*, pages 203–212, 2001.
 26. M. Isenburg and P. Lindstrom. Streaming meshes. In *Proc. of Visualization*, pages 231–238, 2005.
 27. L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.
 28. V. Jain and H. Zhang. A spectral approach to shape-based retrieval of articulated 3D models. *Computer Aided Design*, 39:398–407, 2007.
 29. S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Siggraph*, pages 954–961, 2003.
 30. R. Kimmel and J. A. Sethian. Computing geodesics on manifolds. In *Proc. of National Academy of Sciences*, volume 95, pages 8431–8435, 1998.
 31. Y. Koren, L. Carmel, and D. Harel. Ace: A fast multiscale eigenvectors computation for drawing huge graphs. In *Proc. of the IEEE Symposium on Information Visualization*, page 137, 2002.
 32. B. Korte and J. Vygen, editors. *Combinatorial Optimization, Theory and Algorithms*. Springer-Verlag, 2000.
 33. C. H. Lee, A. Varshney, and D. W. Jacobs. Mesh saliency. *ACM Transactions on Graphics*, 24(3):659–666, 2005.
 34. X. Li and I. Guskov. Multi-scale features for approximate alignment of point-based surfaces. In *Proc. of the Symposium on Geometry processing*, pages 217–226, 2005.
 35. H. Ling and K. Okada. Diffusion distance for histogram comparison. In *IEEE Conference on Computer Visualization and Pattern Recognition*, pages 246–253, 2006.
 36. Y.-S. Liu, M. Liu, D. Kihara, and K. Ramani. Salient critical points for meshes. In *Proc. of the Symposium on Solid and Physical Modeling*, pages 277–282, 2007.
 37. D. Mateus, F. Cuzzolin, R. Horaud, and E. Boyer. Articulated shape matching using locally linear embedding and orthogonal alignment. *IEEE International Conference on Computer Vision*, pages 1–8, 2007.
 38. F. Memoli. On the use of Gromov-Hausdorff distances for shape comparison. In *Proc. of the Symposium on Point-Based Graphics*, pages 81–90, 2003.
 39. F. Mémoli and G. Sapiro. Distance functions and geodesics on submanifolds of \mathbb{R}^d and point clouds. *SIAM Journal of Applied Mathematics*, 65(4):1227–1260, 2005.
 40. F. Mémoli and G. Sapiro. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Foundations of Computational Mathematics*, 5(3):313–347, 2005.
 41. J. Milnor. *Morse Theory*, volume 51 of *Annals of mathematics studies*. Princeton University Press, 1963.
 42. C. Moenning and N. A. Dodgson. Fast marching farthest point sampling. In *Eurographics*, 2003.
 43. M. Mortara and G. Patanè. Shape-covering for skeleton extraction. *International Journal of Shape Modelling*, 8(2):245–252, 2002.
 44. M. Mortara, G. Patanè, M. Spagnuolo, B. Falcidieno, and J. Rossignac. Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. *Algorithmica*, 38(1):227–248, 2004.
 45. D. Nehab and P. Shilane. Stratified point sampling of 3D models. In *Proc. of the Symposium on Point-Based Graphics*, pages 49–56, 2004.
 46. X. Ni, M. Garland, and J. C. Hart. Fair Morse functions for extracting the topological structure of a surface mesh. In *ACM Siggraph*, pages 613–622, 2004.
 47. R. Ohbuchi, K. Osada, T. Furuya, and T. Banno. Salient local visual features for shape-based 3D model retrieval. In *Proc. of Shape Modeling In-*

- ternational*, pages 93–102, 2008.
48. R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21(4):807–832, 2002.
 49. M. Ovsjanikov, J. Sun, and J. Guibas. Global intrinsic symmetries of shapes. In *Proc. of the Eurographics Symposium on Geometry Processing*, pages 1341–1348, 2008.
 50. G. Patanè and B. Falcidieno. Computing smooth approximations of scalar functions with constraints. *Computer & Graphics*, 2009. To appear.
 51. G. Patanè, M. Spagnuolo, and B. Falcidieno. A minimal contouring approach to the computation of the Reeb graph. *IEEE Transactions on Visualization and Computer Graphics*, 2009. To appear.
 52. M. Pauly, M. Gross, and L. P. Kobbelt. Efficient simplification of point-sampled surfaces. In *Proc. of Visualization*, pages 163–170, 2002.
 53. M. Pauly, R. Keiser, L. P. Kobbelt, and M. Gross. Shape modeling with point-sampled geometry. In *ACM Siggraph*, pages 641–650, 2003.
 54. W. H. Press, W. T. Vetterling, S. A. Teukolsky, and B. P. Flannery. *Numerical Recipes in C++: the art of scientific computing, 3rd edition*. Cambridge University Press, 2007.
 55. H. Qiu and E. R. Hancock. Graph partition for matching. In *Graph Based Representations in Pattern Recognition*, volume 2726, pages 178–189, 2003.
 56. D. Raviv, A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Symmetries of non-rigid shapes. In *IEEE Workshop on Non-rigid Registration and Tracking through Learning*, 2007.
 57. M. Reuter, S. Biasotti, D. Giorgi, G. Patanè, and M. Spagnuolo. Discrete laplace-beltrami operators for shape analysis and segmentation. *Computer & Graphics*, 2009. To appear.
 58. M. Reuter, F.-E. Wolter, and N. Peinecke. Laplace-Beltrami spectra as Shape-DNA of surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006.
 59. Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.
 60. M. R. Ruggeri, T. Darom, D. Saupe, and N. Kiryati. Approximating geodesics on point set surfaces. In *Proc. of the Symposium on Point-Based Graphics*, pages 85–93, 2006.
 61. M. R. Ruggeri and D. Saupe. Isometry-invariant matching of point set surfaces. In *Proc. of the Eurographics Workshop on 3D Object Retrieval*, 2008.
 62. R. M. Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proc. of the Symposium on Geometry processing*, pages 225–233, 2007.
 63. J. Schwartz, A. Steger, and A. Weißl. Fast algorithms for weighted bipartite matching. In *Experimental and Efficient Algorithms*, volume 3503, pages 476–487, 2005.
 64. R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. In *ACM Siggraph*, pages 399–405, 2004.
 65. V. Surazhsky, T. Surazhsky, D. Kirsanov, S. J. Gortler, and H. Hoppe. Fast exact and approximate geodesics on meshes. *ACM Siggraph*, 24(3):553–560, 2005.
 66. J. W. H. Tangelder and R. C. Veltkamp. A survey of content based 3D shape retrieval methods. In *Proc. of Shape Modeling International 2004*, pages 145–156, 2004.
 67. M. Turk and A. Pentland. Face recognition using eigenfaces. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–591, 1991.
 68. B. Vallet and B. Levy. Spectral geometry processing with manifold harmonics. *Computer Graphics Forum* 27(2), 2008.
 69. H. Zhang and R. Liu. Mesh segmentation via recursive and visually salient spectral cuts. In *Proc. of Vision, Modeling, and Visualization*, pages 429–436, 2005.
 70. H. Zhang, A. Sheffer, D. Cohen-Or, Q. Zhou, O. van Kaick, and A. Tagliasacchi. Deformation-drive shape correspondence. *Proc. of the Symposium on Geometry Processing*, 27(5):431–439, 2008.
 71. H. Zhang, O. van Kaick, and R. Dyer. Spectral methods for mesh processing and analysis. In *Proc. of Eurographics State-of-the-art Report*, pages 1–22, 2007.
 72. Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, 1994.