

Learning Robust Similarity Measures for 3D Partial Shape Retrieval

Yi Liu^{1,2,*} · Xu-Lei Wang¹ · Hua-Yan Wang³ · Hongbin Zha^{1,*} · Hong Qin⁴

Received: date / Accepted: date

Abstract In this paper, we propose a novel approach to learning robust ground distance functions of the Earth Mover's distance to make it appropriate for quantifying the partial similarity between two feature-sets. First, we define the ground distance as a monotonic transformation of commonly used feature-to-feature base distance (or similarity) measures, so that in computing the Earth Mover's distance, the algorithm could better turn its focus on the feature pairs that are correctly matched, while being less affected by irrelevant ones. As a result, the proposed method is especially suited for 3D partial shape retrieval where occlusion and clutter are serious problems. We prove that when the transformation satisfies certain conditions, the metric property of the base distance is sufficient to guarantee the ground distance is a metric (and so is the Earth Mover's distance), which makes fast shape retrieval on large databases technically possible. Second, we propose a discriminative learning framework to optimize the transformation function based on the real Adaboost algorithm. The optimization is performed in the space of the piecewise constant approximations of the transformation without making any parametric assumption. Finally, extensive experiments on 3D partial shape retrieval convincingly demonstrate the effectiveness of the proposed techniques.

Keywords Partial similarity measure · 3D shape retrieval · Earth Mover's distance · Adaboost

1 Introduction

Content-based information retrieval (CBIR) has received an emerging research interest in the past decade. [27,32,41]. At the same time, as a new type of multimedia, 3D models have been widely used in virtual reality, computer animation, computer aided design and the entertainment industry. The need for easily organizing and reusing the fast growing number of available 3D models has prompted a new trend of research on 3D model retrieval. However, most previous work has only focused on searching for 3D models that are globally similar to a query shape [12,34,42]. Obviously, this is a severe restriction in many practical scenarios since partial similarity is more pervasive among 3D shapes. For example, in the 3D imaging process, not only is the foreground object of interest pictured, but also a lot of background clutter is recorded. As a result, when such images are compared, it is important to ignore the unrelated background information robustly. Besides, because of the occlusions and/or limited visual fields of the 3D range scanners, different range images could only be partially overlapped in general, which is another instance of partial shape similarity. In 3D model retrieval, correctly identifying those partial shape similarities will provide us with newer and more valuable information.

To address this problem, it is important to investigate the fundamental principles for correctly evaluating the partial similarity between 3D shapes. First, we have to introduce a suitable feature representation. Since an ideal partial similarity measure should be robust to clutters and occlusions, global shape descriptors are obviously out of consideration. Instead, it is more appropriate to employ a set of local shape

¹ Key Laboratory of Machine Perception (Ministry of Education), Peking University, Beijing

² Chinese Academy of Sciences Key Laboratory of Molecular Developmental Biology, Center for Molecular Systems Biology, Institute of Genetics and Developmental Biology, Chinese Academy of Sciences, Beijing

³ Department of Computer Science and Engineering, Hong Kong University of Science and Technology

⁴ Department of Computer Science, State University of New York at Stony Brook, Stony Brook, NY 11794-4400

* To whom the correspondence should be addressed: {liuyi, zha}@cis.pku.edu.cn

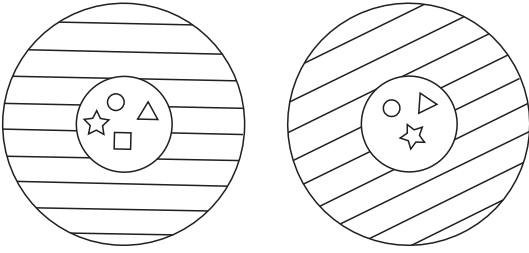


Fig. 1 Feature set comparison with irrelevant and missing features: Left: feature set 1. Right: feature set 2. Irrelevant features are in the dashed regions; features of the foreground objects are inside the small circles. One relevant feature is missing in set 2.

descriptors to represent a 3D object, where each descriptor characterizes a local shape part around a basis point. In this way, it is reasonable to expect some of these local shape descriptors are unchanged in the presence of clutter and occlusion, due to their localized nature. This invariance provides us with valuable partial similarity cues for discrimination. Accordingly, it enables robust shape recognition and 3D partial shape retrieval. As a result, we represent each 3D model by a set of localized features, which is referred to as the bag-of-features representation. Now, the main technical challenge has reduced to the problem of developing a robust similarity/distance measure for two sets of shape signatures. However, the problem is still not trivial. This is because: 1) There is no *a priori* correspondence between the local shape signatures in two sets; 2) As mentioned earlier, irrelevant features as to the interested shape parts may appear, while some relevant features might be missing; 3) For part-in-whole 3D shape retrieval, it is necessary to define a robust similarity measure when the partial shape’s feature set is much smaller than that of the full shape. From now on, for the ease of presentation, we will interchangeably use the word “feature” to denote a shape signature. A schematic illustration of the technical challenges is shown in Fig. 1.

To tackle these difficulties, one might hope to first specify the subset of features that are relevant to the object of interest before feature set comparison, which assumes a pre-segmentation of the 3D scene implicitly. In fact, in image analysis, many previous approaches [3, 4, 20, 24] rely on such an assumption. However, up to now, automatically segmenting the contents of 3D shapes, images and videos [47] is still considered to be an open problem. Besides, it is also impractical to produce such segmentations manually for the unlimited and ever growing numbers of 3D shapes.

In contrast to previous methods, our approach does not require a pre-segmentation of the 3D shapes. As it will not be affected by a moderate amount of irrelevant and/or missing features, the method is able to work on the raw feature-set representation of un-segmented 3D shapes directly. Specifically, to meet the needs of partial similarity based retrieval,

we propose to define the ground distance of the Earth Mover’s distance as a monotonic transformation of commonly used base distance/similarity measures. Harnessed with a well-defined ground distance between two features, the Earth Mover’s distance could turn its focus on the reliable feature pairings across two sets, while ignoring those irrelevant and redundant features that are isolated from stable feature matches. Besides, we also propose a discriminative learning approach to optimizing the transformation function to obtain the best ground similarity measure for a particular task. As a result, the proposed method has a high degree of flexibility and it could potentially be applied to a broad spectrum of applications.

The main contributions of this paper are three-folds: First, we propose a new, flexible transformational mechanism to define the inner feature-to-feature ground distance measure for the Earth Mover’s distance. Specifically, the ground distance is converted from common distance/similarity measures by applying a monotonic transformation. Here, we list a number of criteria that such transformations should comply. Second, we propose a supervised learning algorithm for optimizing such transformations based on the real Adaboost algorithm, which helps us avoid the need of manual specification. The results reveal an interesting fact that most transformations learnt have a sigmoid-shaped profile. Third, the proposed method is applied to develop a novel 3D partial shape retrieval system without explicit shape alignment. The scalability issue is also addressed technically by proving the transformation preserves the desirable metric property under some moderate conditions. We also obtained a deeper understanding of previous approaches to quantifying feature-sets similarity in the proposed new framework. Finally, we conduct extensive experiments to demonstrate the effectiveness and robustness of our approach to 3D partial shape retrieval.

The rest of the paper is organized as follows: Section 2 reviews related work. The construction of the transformational mechanism for defining robust ground distance function in the Earth Mover’s distance and an analysis of its properties are introduced in Section 3. Section 4 presents an algorithm for learning the ground distance measure by optimizing the transformation function based on Adaboost. The experimental results on 3D partial shape retrieval are reported in Section 5. Finally, we conclude this paper and discuss possible further research directions in Section 6.

2 Related Work

In recent years, 3D objects retrieval has received a considerable amount of research interests from people working on computer graphics, computer vision and multimedia. A common approach to this problem is to represent the overall shape of a 3D model using a global shape descriptor. Then,

the similarity search can be performed efficiently in the descriptor space. It is generally expected that the shape descriptor is invariant to the free rotations of a 3D model. Otherwise, each 3D model should be normalized to a canonical coordinate system before the shape descriptor is applied for feature extraction. One of the first global shape descriptors is shape distributions [34], where a Monte-Carlo algorithm is employed to draw random points on a 3D shape uniformly. Then, the distribution of the Euclidean distances between two point samples is proposed as the D2 shape signature. Another method is to represent a 3D shape using a group of spherical functions [12,44]. By applying the spherical harmonic transformation, rotational invariant features can be extracted by computing the energies at each l -frequency of the spherical harmonic expansion. An experimental comparison of some global shape descriptors can be found in [8] and please refer to [42] for a comprehensive survey of algorithms for 3D shape retrieval.

These global shape descriptors are computationally very efficient for shape comparison. However, they could not be applied to evaluate the partial similarity between 3D shapes. A direct approach to tackle this problem is to try to align two shapes to detect their overlapped parts. However, the computational complexity far exceeds the requirements of performing online shape retrieval. Moreover, the alignment of deformable shape parts is a challenging problem itself. To this end, a number of alignment-free methods have been proposed for 3D partial shape retrieval [13,27,28,38]. All these methods use a collection of local shape descriptors to characterize the detailed shape of a 3D model. In particular, [13,28] model the spatial distributions of local features in the three-dimensional space while [27,38] do not. Moreover, the original local shape features are used in [13] while a codebook strategy is employed in [27,28,38] to simplify the feature representation and similarity computation.

The codebook strategy is also termed as the bag-of-words paradigm in image analysis [27,41,47], where a representative set of local features is pre-computed via a clustering algorithm to form the codebook. In this manner, the need for matching two sets of local features is reduced to comparing two frequency histograms of the clusters in a codebook. Though this approach is very intuitive and easy to implement, it has two main limitations. First, by clamping features to the indices in a codebook, fine-grained information in the original features is totally lost. Second, the codebook is application dependent. It is unclear whether a codebook generated from one database could be used for shape retrieval on another database.

Other 3D partial shape retrieval methods based on original local shape descriptors include [14,39,43]. Specifically, in [39], the authors used local spherical harmonic descriptors to characterize the local shape region around a point. Shape retrieval is performed by matching a local shape de-

scriptor of the query to its closest partners in the feature-sets of database models. The retrieval performance, which is quantified using the DCG score, reflecting the distinctiveness of the descriptor on the query shape. [14] proposes new local shape descriptor and develops algorithms to compose low-level descriptors to high-level salient features for partial shape matching, where the geometric hashing algorithm is used to compute the alignments between different shapes using a voting approach. In [43], three different approaches to representing a 3D model as a weighted point set are discussed and a variant of the Earth Mover's distance is used to measure the similarity between two sets. Though these approaches have their attractive properties, none of them is very suited for 3D partial shape retrieval. In [39], only a single local shape descriptor is used to represent the query shape, which is not appropriate to depict relatively large shape regions. In [14], the shape alignments should be explicitly computed, which is rather time consuming. Finally, since the point-set representation is not rotationally invariant in [43], a global PCA pose normalization procedure is required, which is not robust in the case of partial shape retrieval.

For partial shape matching algorithms based on the structural information, in [5], the authors constructed rooted trees using the spectral decomposition of 3D shapes. Then, shape matching and comparison is performed via graph matching using a dynamic programming algorithm. In [9], the curve-skeleton of 3D models is first extracted using a vector field based approach. Then, the correspondence between curve segments is computed by iteratively computing the Earth Mover's distance and the transformation. And in [6], the authors represent the structure of 3D shapes using extended Reeb graphs, where each node of the graph is associated with a geometric descriptor to characterize the local shape properties. Then, graph matching algorithms are used to extract common sub-graphs to identify the partial similarity between two shapes. The main problem with these approaches is the high computational complexity. Besides, it is unclear to what extent the graph matching algorithms are robust to large-scale shape changes, clutter and occlusion. For the method in [9], if the initial transformation is far from optimal, the algorithm may converge to a poor solution.

Since our approach essentially computes the partial similarity between two feature sets, we also review related work in this aspect. Specifically, to establish the correspondence of features across two sets, the simplest greedy approach is to match each feature to the nearest one from the other set [32]. Once the feature pairing is established, it is possible to take the maximum (the Hausdorff distance [35]), the minimum, or average (the "average linkage" in [19]) distances between the feature pairs as the feature-set distance. The main drawback of this approach is the allowance of unbalanced one-to-many feature matches.

By imposing the mass conservation constraint in feature matching, the Earth Mover’s distance (EMD) [36] **g**over the drawback of greedy feature matching above. Essentially, based on a pre-defined feature-to-feature ground distance, the EMD distance computes the minimum cost of moving one set of features to the other set. However, with an ill-defined ground distance, the EMD distance may generate problematic feature pairings and yield very poor retrieval results. To avoid this problem, in this paper, we systematically investigate the question of how to define an appropriate ground distance for making the EMD distance be a good partial similarity measure.

In the case that the ground distance is Euclidean, a space embedding method [15,21] is proposed to approximate the Earth Mover’s distance for reducing its computational complexity. It works by partitioning the feature space hierarchically and perform feature matching from the finest level to the coarsest level. In each hierarchy, the number of matched features is counted and each one is associated with a cost proportional to the bin size. The unmatched features are left to the next level. Finally, the total cost is computed by summing over all hierarchies. It approximates the Earth Mover’s distance within a multiplicative constant factor. Since the factor is proportional to the dimensionality of the feature space, the Euclidean embedding method is not accurate for high-dimensional features.

In high resemblance to the Euclidean embedding method, the pyramid match kernel (PMK) [16] computes the similarity between two feature sets by replacing the distance cost for each feature pairing with a similarity score, which is inversely proportional to the space partition granularity in that hierarchy. It is shown that PMK is better for measuring partial similarities. However, like the embedding method, the PMK is also not suited for matching high dimensional features except a recent variant proposed by us [29]

Although uniform space partitioning suffers **o**urse of dimensionality problem, it is shown empirically that the hierarchical bag-of-words approach generalizes well to highly dimensional features [17]. Furthermore, the embedding methods implicitly assume that feature matches are formed in a Euclidean space, which is not quite realistic for some image/shape signatures [25,26].

3 The Transformational Mechanism for Defining Robust Ground Distance

In this section, we introduce the technical details of the proposed transformational mechanism for defining the ground distance in the EMD distance. First, for the ease of problem analysis, we present an equivalent, dual optimizing criterion and link it to the original EMD distance. Instead of minimizing the total cost of transporting a set of features to the other set, in the dual formulation, we maximize the total excited

similarities of matching two sets of local features. Specifically, a transformation mechanism is proposed to map common (base) distance/similarity measures to the ideal ground similarity measure in the dual formulation. The key benefit and flexibility of our approach owe to this transformational function, which is able to help the EMD distance get rid of irrelevant features and focus on relevant ones. To facilitate designing such functions, we propose a number of key properties that an ideal ground similarity measure should satisfy. Finally, we prove that under certain conditions, the transformation could well transfer the metric property of the base distance to the ground distance, which in turn implies the EMD distance is also a metric when the mass of feature sets is equal. This is important to make our proposed algorithm scalable to large databases. After presenting the technical route of our method, we compare it with related approaches.

3.1 The Dual Formulation of EMD and the Transformation Mechanism

Suppose there are two feature sets, $P = \{(p_i, u_i)\}$ and $Q = \{(q_j, v_j)\}$, p_i and q_j are the i -th and j -th feature in the two feature sets, and u_i and v_j are their weights. Our problem is to compute a score which quantifies the distance or similarity of P and Q . It is expected that an ideal partial similarity measure for P and Q will have a certain degree of resistance to outlier and/or missing features.


For the ease of analyzing partial similarity measures, we introduce an equivalent, dual formulation of the EMD distance [36] **e**ned “partial similarity measure” and PSM for short), which is defined as the maximum total excited similarities when a set of features is transported to the other set of features:

$$PSM = \max_{F'=\{f_{ij}\}} \frac{\sum_{i,j} f_{ij} s_{ij}}{\sum_{i,j} f_{ij}}, \quad (1)$$

with the following constraints:

$$\begin{aligned} \sum_j f_{ij} &\leq u_i, \quad \sum_i f_{ij} \leq v_j, \quad f_{ij} \geq 0, \\ \sum_{i,j} f_{ij} &= \min \left\{ \sum_i u_i, \sum_j v_j \right\}. \end{aligned} \quad (2)$$

Note that this criterion is not a new contribution itself, but it greatly enhance **o**ur intuition about how to define good partial similarity measures. Similar to the terminology of the EMD distance, here $s_{i,j}$ is defined as the ground similarity between two features (p_i, q_j) , which is induced from a base distance (or base similarity) measure $d(p_i, q_j)$ by a monotonic transformation function $s_{i,j} = F(d)$. And $f_{i,j}$ is the weight that the i -th feature of P transports to the j -th


feature of Q .  four conditions are the mass conservation constraints that must be satisfied in solving the supply-demand transports, which dictates that: 1) For each feature, the sum of transported weights must be smaller or equal to its own weights. 2) The smaller feature set must be fully transported to the other set. 3) The feature transports $\{f_{i,j}\}$ must be non-negative.

Here, we note that the dual PSM formulation is very similar to the original EMD [36] distance. Specifically, the four constraints are exactly the same. However, there are two differences. First, in PSM, we maximize the similarity between two feature-sets rather than minimizing their distance in EMD. This is more intuitive and straightforward in detecting partial similarities across two sets. Second, a monotonic distance transformation is added in the PSM formulation, which offers great flexibility in specifying the ground similarity $s_{i,j}$. In fact, we only expect the base distance (or similarity) $d(p_i, q_j)$ computes an approximate affinity for feature pairs, e.g., which feature pairs are matched better than other feature pairs. The exact value of the ideal ground similarity $s_{i,j}$ may not be linearly related to $d(p_i, q_j)$, but could be linked by a non-linear transformation function. Here, our basic idea is that it would be much easier to decide the relative order of the distances (or similarities) between feature pairs, rather than the exact values.

Here, our key contribution is the introduction of the transformation mechanism for flexibly defining the ground similarity measure. As we shall prove below, the PSM formulation is equivalent to the EMD distance and the ground similarity measure in PSM is related to the ground distance in EMD. Therefore, the transformation mechanism also provides a new, powerful way to specify the ground distance of EMD. To make a comparison of using and not using the transformation in EMD, in this paper, we will only map a base “distance” $d(p_i, q_j)$ to the ground similarity in PSM, so the transformation function $s_{i,j} = F(d)$ is monotonically decreasing, as similarity is reversely related to distance. However, it would be straightforward to use a base similarity measure (and here the transformation function would be monotonically increasing). Now, we prove the dual PSM formulation is equivalent to the definition of the original EMD distance.

Define the ground distance $GD(p_i, q_j)$ in EMD as:


$$GD(p_i, q_j) = A - s_{i,j}, \quad (3)$$

where A is the tight upper bound of the ground similarity s  which guarantees that the ground distance $GD(p_i, q_j)$ being non-negative. Now, we show the corresponding Earth Mover’s distance:

$$EMD = \min_{F'=\{f_{ij}\}} \frac{\sum_{i,j} f_{ij} GD_{ij}}{\sum_{i,j} f_{ij}}, \quad (4)$$

with $GD(p_i, q_j)$ as its ground distance and the same constraints in (2), is equivalent to the PSM formulation (1). Specifically, by unraveling the definition of GD in (3), we have:

$$EMD = A - \max_{F'=\{f_{ij}\}} \frac{\sum_{i,j} f_{ij} s_{ij}}{\sum_{i,j} f_{ij}} = A - PSM. \quad (5)$$


Now it is clear that the feature transports $\{f_{i,j}\}$ which minimizes the EMD formulation (4) will maximize the PSM formulation (1), and the sum of PSM and EMD is a constant A . As a result, we can always solve PSM by optimizing its corresponding EMD formulation. Despite that the two formulations are equivalent, via  the optimization problem as maximizing the similarity across two feature sets will help us explore which desirable properties that an ideal ground similarity measure should have.

3.2 Properties of Robust Ground Similarity Measures

As introduced above, the ground similarity measure $s_{i,j}$, is computed by transforming the base distance $d(p_i, q_j)$ through the function $s_{i,j} = F(d)$. Before proceeding ahead, we will study how to design an ideal ground similarity measure $s_{i,j}$ in order to make the resulting PSM robust to missing and outlier features. We will show that the transformation mechanism is a powerful way to generate an appropriate ground similarity measure for 3D partial shape retrieval. Here, I will list the desirable properties of a good ground similarity and explain why.

- (1) The ground similarity score should be non-negative.

Suppose that two 3D shapes have a similar part. If the ground similarity between two features can be negative, then the dissimilar parts of the two objects could contribute an overall negative score which may cancel the positive score from the similar part of the objects. In that case, we are not able to discover the similarity between the two objects from the overall score.

- (2) The ground similarity should be a monotonic  increasing function of the corresponding base distance.
- (3) The ground similarity should approach zero when the base distance approaches infinity.

This is because a partial similarity measure should reflect the strength of the similar part of two 3D shapes, while being undisturbed by their dissimilar parts. Otherwise, if the ground distance is also not bounded, like the base distance, the EMD would be largely determined by the distinct parts of two objects. For example, suppose that feature pairs will be considered similar if their distances are less than 10. If we change the very large distance of two distinct features from,

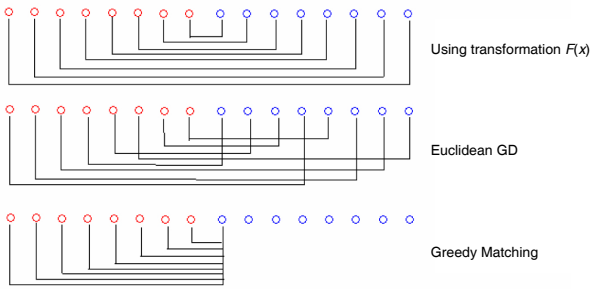


Fig. 2 A comparison of feature matching schemes. Top: The feature matching histogram of PSM with a non-linear transformation function $F(x)$, which correctly pairs nearby features. Middle: The EMD with the original Euclidean ground distance. Here, short-distance matches can be affected by long distance matches. Bottom: Greedy matching. Mass-conservation is violated, and the estimated similarity score can be too optimistic.

saying to 200, the final similarity score output by PSM (or EMD) with the transformation function will be nearly invariant, while the distance will change significantly if the base distance is used directly as the ground distance in EMD. Also, this property implies that in the EMD formulation, an ideal ground distance GD will approach the upper bound A when the base distance becomes very large.

- (4) The similarity score should have a (reasonable) upper bound.

If not, when two exact local features are matched, a huge similarity score will be excited in PSM. In this case, it is hard to judge whether the two 3D shapes have a large similar part, or they have just a few exact feature matches.

With an ideal ground distance/similarity generated by the transformation, the EMD/PSM formulation could also produce a better feature matching pattern than the EMD whose naïve ground distance definition does not satisfy the properties above. As shown in Fig. 2, three feature matching schemes are compared. The PSM with the transformation function $F(x) = \frac{1}{1+x}$ correctly pairs the corresponding features across the two sets. In this case, it is not hard to verify that PSM will yield a smaller similarity score for any other feature pairing satisfying the four constraints. However, in the EMD formulation with a Euclidean ground distance, the pairing of the closely related features can be severely affected by irrelevant feature matches. This time, the irregular feature pairing shown in the middle panel of the figure generates the same EMD distance as the correct feature pairing shown in the upper panel. As a result, the partial similarity between the two feature-sets is not identified in this scenario. In the greedy feature matching scheme, since one-to-one feature correspondence is not explicitly enforced, the estimated similarity could be too optimistic: From the lower panel, we can see that the average distance between feature pairs is reduced.

3.3 The Analysis of Our Method and Related Approaches

In this section, we discuss the relations of our method with previous feature set comparison approaches [15–17, 21]. As shown before, the Euclidean embedding [21] methods, including pyramid match kernels, could not be directly applied to high dimensional features, such as spin images for 3D shapes [23], and SIFT descriptors for 2D images [30].

Furthermore, we have shown in Fig. 2 that using the Euclidean ground distance, the EMD algorithm may produce poor feature matches. However, both the Euclidean embedding and the PMK methods approximate the feature correspondence in a Euclidean space, although PMK sums up similarities in the later step. As a result, these approaches may also suffer from low quality feature matches. In the hierarchical bag-of-words approach [17, 33], the feature codebook is database-dependent, making it a less general approach to measuring partial similarity.

Since the EMD distance is computationally slow, it could not be used directly to perform fast partial shape retrieval on very large 3D model databases. However, as proven in [36], the EMD distance is a metric when the ground distance is a metric and the total weights of the two feature sets are equal. The Lipschitz embedding approach [7] tells us that when the distance measure between objects is a metric, there exists an embedding of these objects in a normed-space that approximates their pair-wise distances. After that, the distance computation can be conducted in the normed-space very efficiently. More remarkably, the locality sensitive hashing [10] technique could be used to further speed up the similarity retrieval in a normed-space to sublinear time complexity. In our approach, we are interested in exploring under which conditions, the transformation function is able to preserve the metric property of the base distance to the ground distance. Fortunately, we obtained the positive result in Theorem 1.

To complete the proof, we have to assume that the base distance is a metric and the transformation function $s = F(x)$ should satisfy an additional property:

- (5) The second derivative of the transformation function, which maps a base distance to the ground similarity in PSM, is positive.

Note that this is not a stringent requirement. Many transformation functions satisfy this property. For example, $F(x) = \frac{1}{1+x}$. Based on the relation between the ground similarity in PSM and the ground distance in EMD (3), an alternative expression of this property is that the transformation from the base distance to the ground distance in EMD should have a negative second derivative with x .

Theorem 1 *If the base distance is a metric; the transformation function, which maps a base distance to the ground similarity of PSM, satisfies the five properties mentioned above;*

and the total weights of two feature sets are equal, then the corresponding EMD distance is a metric.

As mentioned above, the key steps to prove this theorem show that under these conditions, the transformation is able to preserve the metric property of the base distance to the ground distance. Specifically, we have:

Proof. To prove the EMD distance is a metric, we first prove its ground distance GD is a metric. It is easy to verify that GD satisfies the following two metric properties:

- 1) $GD(x, y) = 0$, iff $x = y$; (Since A is the upper bound of the ground similarity)
- 2) $GD(x, y) = GD(y, x)$.

Now, it suffices to prove the triangle inequity of GD . Suppose that we have 3 local features A, B, C . Their pairwise base distances AB, AC , and BC satisfy the triangular inequity, since the base distance is a metric: $AB + AC \geq BC, AC + BC \geq AB, AB + BC \geq AC$.

Without loss of generality, we assume that $AB \leq AC \leq BC$. As a result, the inequity $AB + AC \geq BC$ is non-trivial, while the other two inequities hold naturally. Denote the corresponding ground distance of AB, AC , and BC after transformation by AB', AC' , and BC' . We have:

$$\begin{aligned} AB' &= G(AB) = \int_0^{AB} G'(x)dx, \\ AC' &= G(AC) = \int_0^{AC} G'(x)dx, \\ BC' &= G(BC) = \int_0^{BC} G'(x)dx. \end{aligned} \quad (6)$$

Here $G(x)$ is the mapping from the base distance to the ground distance GD . Since the second derivative of $G(x)$ is negative, $G'(x)$ is monotonically decreasing. With this property, and note $AB + AC \geq BC$, it is easy to see that $AB' + AC' \geq BC'$ hold. And since $G(x)$ preserves order, $AB' \leq AC' \leq BC'$, the other two inequities hold naturally.

Now we have proved that the ground distance GD is a metric. Based on the fact that a EMD distance is a metric when 1) its ground distance is a metric; 2) the total weights of two feature sets are equal [36], we have proved that the EMD distance with the transformation function is also a metric. Therefore, we could speed up object retrieval in large databases using metric embedding approaches. \square

Note that in part-in-whole and part-to-part retrieval, the total weights of two feature sets might not be equal. Therefore, the metric property of the resulting EMD distance is not guaranteed and the metric embedding approach could not be used in this case. However, as a rule of thumb, we could randomly duplicate a few features in a set or adjust the

weights of some features to make the total weights equal. As we have discussed above, the transformational mechanism greatly enhances the robustness of the EMD distance to these artificially added irrelevant features. As a result, the performance could still be fairly good.

4 Approximate Learning of Distance Transformation Function

In the PSM formulation, to well discriminate well-matched features and irrelevant clutter, we need an optimal ground similarity measure between two features. As shown in Section 3, the transformation function is a flexible mechanism to map a common base distance measure to a desirable ground similarity measure. However, if we specify the functional form, as well as the parameters of the transformation manually, the PSM may not have satisfactory performance. In this section, we present a novel method to learn the transformation function under the supervision of human agents, who offer guidance on similarities between objects (by category labels or relative comparisons). What based on equation (3), the function also decides how to convert the base distance to an ideal ground distance in EMD.

The basic idea of learning the transformation function is to iterate the following two steps:

- 1) Given the transformation function, we solve the correspondence of local features by maximizing the total similarity between two 3D shapes.
- 2) Given the correspondence of local features between 3D shapes, we optimize the transformation function to maximize the percentage of correct similarity judgments by PSM.

The first step is trivial. Essentially, we just have to solve the original PSM optimization criterion (alternatively, the corresponding EMD distance) itself, that is, to compute how the features are optimally matched (by solving the demand-supply transports $F = \{f_{ij}\}$ between two 3D shapes, see Equations (1) and (2)).

Hereafter, we will primarily deal with the second part of the problem: When the correspondences of local features are established, how to compute the optimal transformation function that maximizes the retrieval performance.

Since there is no *a priori* information about the functional form of the transformation function, it is not appropriate to make any parametric assumption in learning. Instead, we take a nonparametric approach: The base distance of local features is discretized into uniform intervals, and the problem is reduced to compute the corresponding ground distance GD for each interval, shown in Fig. 3.

Now we describe the proposed approach to learning the distance transformation function. Our learning scheme is similar to the BoostMap approach [1, 2]. However, there are two

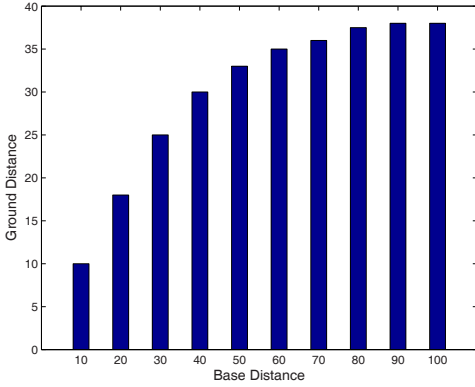


Fig. 3 The histogram representation of the distance transformation function.

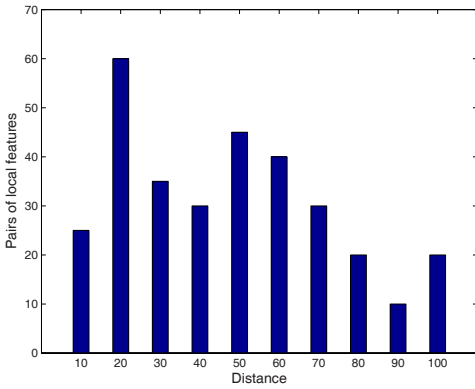


Fig. 4 The feature matching histogram. It counts the total weights of the feature pairs that fall into each distance interval.

major differences: 1) The objective in this paper is to learn the distance transformation function of PSM, while theirs is to learn an embedding in a normed-space for approximating a computational expensive distance function. 2) We proposed a new method to guarantee the parameters to be non-negative in the Adaboost optimization procedure, which guarantees the monotonicity of the distance transformation function.

Let A be the set of training multimedia objects. $D_A(a_1, a_2)$ is the distance measure between objects $a_1, a_2 \in A$. An ordinal function is defined as

$$\bar{F} = \begin{cases} 1, & D_A(q, a_1) < D_A(q, a_2) \\ 0, & D_A(q, a_1) = D_A(q, a_2) \\ -1, & D_A(q, a_1) > D_A(q, a_2) \end{cases}. \quad (7)$$

\bar{F} has three discrete output values: -1, 0, +1, which is a quantized version of a continuous function $\tilde{F}(q, a_1, a_2)$, defines as:

$$\tilde{F} = D_A(q, a_2) - D_A(q, a_1). \quad (8)$$

\tilde{F} can be regarded as a real-valued classifier, and we will employ the real Adaboost learning algorithm [37] to approximate it by a number of weak classifiers.

Our training set S consists of T triplet of objects:

$$S = ((q_1, a_1, b_1), \dots, (q_T, a_T, b_T)). \quad (9)$$

For each triplet, a label is attached: If (q, a) belongs to the same class, while (q, b) belongs to different classes, then the corresponding label of the triplet is 1, otherwise, the label is -1. We denote the label of (q_t, a_t, b_t) by y_t .

As introduced above, the distance between local features is discretized into many intervals. As a result, after solving the feature correspondence between two objects, we can count the total weights of the feature pairs that fall into each distance interval. The result is the *feature matching histogram* (FMK) of the two objects, shown in Fig. 4. Note that, since our learning method is recursive, at first round, the distance to be discretized is the initial base distance between local features. At later rounds, the distance to be discretized could be the ground distance learnt in the last round, or still the original base distance.

Real-Adaboost algorithm is employed to learn the distance transformation function. More specifically, we fit the real-valued classifier $\tilde{F}(q, a_1, a_2)$ by the additive composition of its weak classifiers $\tilde{F}_k' = c_k(q, a_2) - c_k(q, a_1)$, where $k = 1, \dots, K$, K is the dimension of the *feature matching histogram* (FMK), and $c_k(q, a)$ is the number in the k -th dimension of the FMK between objects q and a .

$$\begin{aligned} \tilde{F} &= D_A(q, a_2) - D_A(q, a_1) \\ &= \sum_k \alpha_k c_k(q, a_2) - \sum_k \alpha_k c_k(q, a_1) \\ &= \sum_k \alpha_k [c_k(q, a_2) - c_k(q, a_1)] \\ &= \sum_k \alpha_k \tilde{F}_k', \end{aligned} \quad (10)$$

where the coefficients $A = \{\alpha_k\}, k = 1, \dots, K$ are the outputs of the Adaboost learning procedure. They define a distance transformation function f , which maps the k -th distance interval to α_k .

Now suppose the learning procedure stops after M iterations. In each iteration, if the discretization is always performed on the ground distance learnt in the last round, the resulting transformation should be $F = f_M f_{M-1} \dots f_2 f_1$, where f_m is the transformation learnt in the m -th round, defined by the coefficients A_m . Otherwise, if the discretization is always performed on the original base distance, then the resulting transformation function is $F = f_M$.

In practice, we found that 1) when the transformation function F is monotonic, the feature matching pattern computed using the transformation is not very different at of not using the transformation. So there would not be much

difference in the first step of each learning round. 2) The coefficients A_m only approximate the transformation function with a histogram. So it is hard to define the convergence of the learning procedure strictly. As a result, in our current implementation, we only run one round of the learning process. However, good results are obtained in this simple setting.

Now we introduce the details of the learning procedure. It is known that the Adaboost algorithm [37] takes a number of rounds. At each round, a weak learner is trained with the weighted version of the original training samples. After that, each training sample is re-weighted according to the confidence of being correctly classified by the weak learner. Briefly, the samples that are falsely classified by the weak classifier will be assigned to larger weights after this iteration and *vice versa*. The procedure continues until the testing error no longer decreases or some pre-determined number of rounds is reached.

More precisely, at each Adaboost learning round r , a weight $w_{r,t}$ will be assigned to each triplet (q_t, a_t, b_t) , satisfying $\sum_{t=1}^T w_{r,t} = 1$. In the beginning, all weights are initialized equally: $w_{1,t} = \frac{1}{T}$.

In the beginning, the composition coefficients of $A = \{\alpha_k\}, k = 1, \dots, K$ of the weak classifiers \tilde{F}'_k are set to 0. At the r -th round, we try to select a weak classifier \tilde{F}'_{k^*} from the pool $C = \{\tilde{F}'_k, k = 1, 2, \dots, K\}$ that best minimizes the overall empirical training error. To quantify this notion, a measure Z_r was proposed [37]:

$$Z_r(\tilde{F}'_k, \alpha_k) = \sum_{t=1}^T w_{r,t} \exp(-\alpha_k y_t \tilde{F}'_k(q_t, a_t, b_t)). \quad (11)$$

In (11), $r = 1, 2, \dots, R, k = 1, 2, \dots, K$ and t ranges from $1, 2, \dots, T$, where R is the maximum number of iterations of the Adaboost algorithm, K is the dimension of the feature matching histogram, i.e., the number of weak classifier and T is the number of triplets (training data). Generally speaking, $Z_r(\tilde{F}'_k, \alpha_k)$ represent the benefit of adding the k -th weak classifier \tilde{F}'_k with weight α_k to the current classifier composition in minimizing the empirical training error. The smaller the Z_r , the larger the benefit. When $Z_r(\tilde{F}'_k, \alpha_k) > 1$, adding \tilde{F}'_k with weight α_k actually deteriorates the classification performance. Therefore, at the r -th iteration, we choose the weak classifier k^* with weight α'_{k^*} : $(\tilde{F}'_{k^*}, \alpha'_{k^*})$ that minimizes Z_r :

$$(\tilde{F}'_{k^*}, \alpha'_{k^*}) = \arg \min_{(\tilde{F}'_k, \alpha_k) \in (C, A)} Z_r(\tilde{F}'_k, \alpha_k). \quad (12)$$

The overall Adaboost learning procedure is summarized in Algorithm 1.

Algorithm 1: The Adaboost algorithm for learning the distance transformation function.

Initialization:

Set $r = 1, \alpha_k = 0, k = 1, \dots, K, w_{1,t} = \frac{1}{T}, t = 1, \dots, T$.

Main loop:

For the r -th iteration,

- 1) Compute weak classifier \tilde{F}'_{k^*} and weight α'_{k^*} that minimize $Z_r(\tilde{F}'_k, \alpha_k)$, i.e.

$$(\tilde{F}'_{k^*}, \alpha'_{k^*}) = \arg \min_{(\tilde{F}'_k, \alpha_k) \in (C, A)} Z_r(\tilde{F}'_k, \alpha_k),$$

where

$$Z_r(\tilde{F}'_k, \alpha_k) = \sum_{t=1}^T w_{r,t} \exp(-\alpha_k y_t \tilde{F}'_k(q_t, a_t, b_t)).$$

- 2) If $Z_r(\tilde{F}'_{k^*}, \alpha'_{k^*}) < 1$, set $\alpha_{k^*} = \alpha_{k^*} + \alpha'_{k^*}$; else terminate.
- 3) Re-set the weights associated with the training samples:

$$w_{r+1,t} = \frac{w_{r,t} \exp(-\alpha'_{k^*} y_t \tilde{F}'_{k^*}(q_t, a_t, b_t))}{Z_r(\tilde{F}'_{k^*}, \alpha'_{k^*})}, t = 1, 2, \dots, T.$$

If $r > R$, where R is the pre-determined maximum number of iterations of Adaboost, terminate, else, $r \leftarrow r + 1$

A key step of the process is how to choose α'_{k^*} , given \tilde{F}'_{k^*} . In their original paper [37], they solve this problem by setting the first order derivative of Z_r over α equal to 0. By denoting $u_t = y_t \tilde{F}'(q_t, a_t, b_t)$, it is easy to obtain:

$$Z'_r(\alpha) = - \sum_{t=1}^T w_{r,t} u_t \exp(-\alpha u_t). \quad (13)$$

And they show that under common circumstance (there exists t_1, t_2 such that $u_{t_1} < 0, u_{t_2} > 0$, i.e., the empirical error is not zero), $Z'_r(\alpha)$ is monotonically increasing and has one zero point. So we could simply perform a line search to find the root α' .

However, as shown in Fig. 5, the learnt distance warping function is zigzagged, i.e., the ground distance is not monotonically increasing with the base distance. This is mainly due to 1) we only have limited training data; 2) the weights in different base distance intervals are unbalanced; 3) we may not obtain a globally optimal solution by the Adaboost algorithm. On the whole, the ground distance has the trend to increase with the base distance. However, such zigzagged functions will generalize poorly beyond training data, i.e., the transformation function is overfitted by only focusing on minimizing the empirical risk, while ignoring intrinsic of the intrinsic property of the transformation function (monotonicity).

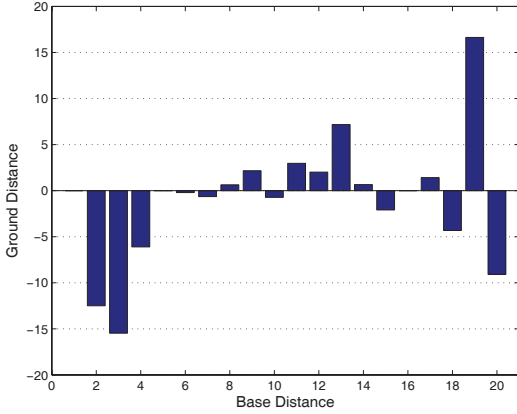


Fig. 5 The profile of the distance transformation function learnt by the original Adaboost algorithm. It is zigzagged.

To overcome this drawback, we propose a restricted-Adaboost learning algorithm with the *cumulative feature matching histogram*, which guarantees that the resulting transformation function is monotonically increasing. First, we define the *cumulative feature matching histogram*. Let $C = \{c_1, c_2, \dots, c_K\}$ denote the original feature matching histogram. We define the *cumulative feature matching histogram* as $E = \{e_1, e_2, \dots, e_K\}$, where

$$e_k = \sum_{j=k}^K c_j, k = 1, \dots, K. \quad (14)$$

The corresponding weak classifiers are $H'_k = e_k(q, a_2) - e_k(q, a_1)$, and the additive composition coefficients of these weak classifiers are $B = \{\beta_k, k = 1, \dots, K\}$, i.e., the strong classifier is $\tilde{F} = \sum_k \beta_k \tilde{H}'_k$. Unraveling this equation, we obtain:

$$\tilde{F} = \sum_{k=1}^K \beta_k \tilde{H}'_k = \sum_{k=1}^K \beta_k \sum_{j=k}^K \tilde{F}'_j = \sum_{j=1}^K \tilde{F}'_j \sum_{k=1}^j \beta_k. \quad (15)$$

By comparing the above equation with $\tilde{F} = \sum_j \alpha_j \tilde{F}'_j$, we have:

$$\alpha_j = \sum_{k=1}^j \beta_k. \quad (16)$$

Having established this relationship, if we could guarantee that $\beta_i \geq 0$ in the Adaboost learning process, then it is clear that α_j is non-decreasing.

To this end, we modified the original Adaboost learning procedure to find a non-negative β value. Recall that

$$Z'_r(\beta) = - \sum_{t=1}^T w_{r,t} u_t \exp(-\beta u_t). \quad (17)$$

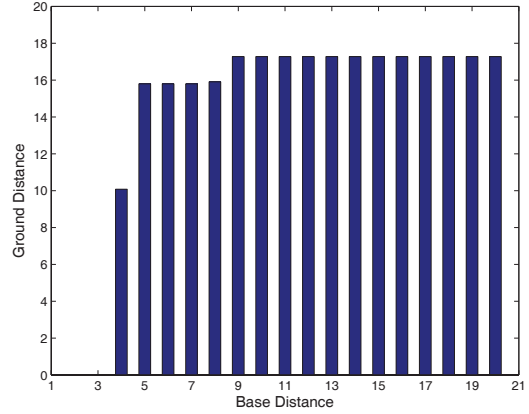


Fig. 6 The profile of the distance transformation function learnt by ordinal Adaboost. This time, the function is monotonical.

Here, we test whether $Z'_r(0) > 0$. If this is true, then $\beta' < 0$, where β' is the root of $Z'_r(\beta)$. (Since under ordinary conditions, i.e. when there exists t_1, t_2 such that $u_{t_1} < 0, u_{t_2} > 0$, $Z'_r(\beta)$ is monotonically increasing and $Z'_r(\beta) \rightarrow -\infty$, when $\beta \rightarrow -\infty$; $Z'_r(\beta) \rightarrow +\infty$, when $\beta \rightarrow +\infty$.) As we restrict $\beta \geq 0$, the minimum value $Z_r(\beta)$ occurs at $\beta = 0$, because $Z'_r(\beta) > 0, \beta \geq 0$. So we return $\beta = 0$ in this case.

When $Z'_r(0) < 0$, it is clear that the root $\beta' > 0$. So we perform a line search in the range $[0, +\infty)$ to find the root β' of $Z'_r(\beta)$.

We tried to apply the restricted Adaboost learning algorithm to our problem. This time, the monotonicity of the transformation function is guaranteed, as shown in Fig. 6.

We can see from the figure that the distance transformation behaves like a soft-thresholding function (sigmoid). This is an interesting result since the soft-thresholding signal transformation has sound physiological evidence, and it is widely used in artificial neural networks. Besides, it is clear that the ground distance no longer increases as the base distance exceeds some point. In other words, learning from real-world data indeed validates the proposed ‘‘saturation’’ property of the ground distance.

In many image/shape recognition tasks, it is necessary to define a kernel to measure the similarity between two training/testing instances, so that standard kernel machines, such as SVMs, could be used directly for classification. The Gaussian kernel of EMD [48] with a naïve ground distance has been proposed for object recognition and is empirically shown to be very successful (though still unknown whether it satisfies the Mercer’s theorem). We believe that if the proposed transformation function is used in the definition of the ground distance of the EMD kernel, the performance of object recognition can be further boosted. Note that the functional calculus transformation [46] could be used here to avoid large diagonal entries in the kernel matrix.

5 Experimental Results

In this section, we test the proposed techniques through extensive experiments on 3D shape retrieval. The experimental results suggest that the proposed algorithms have superior performance in general and are especially suited for 3D partial shape retrieval tasks. The results also suggest that the best transformation function for different shape classes could be quite different. By training such “class specific” transformation functions, the retrieval performance can be further improved. This finding also prompts the fact that our approach is able to incorporate user-feedbacks conveniently.

Table 1 The composition of the PSB-52 database

biplane. airplane	fighter. jet. airplane	human. biped	face. body. part	hand. body. part	ship. see. vessel	head. body. part	aircraft. airplane. glider
1119	1168	118	290	323	1427	340	1267
1121	1170	120	292	325	1429	342	1269
1123	1172	122	295	327	1431	344	1271
1125	1174	124	297	329	1433	346	1273
1127	1176	126	298	331	1435	348	1275
1129	1178	128	299	333	1437	350	1277
-	1180	130	301	335	-	-	-

Numbers are the indices of the 3D models in the Princeton Shape Benchmark.

To demonstrate the effectiveness of our algorithm, we perform three experiments in this section. The purpose of the first experiment is to gain a deep understanding of the proposed algorithms. As a result, we exhaustively test a large number of retrieval scenarios and learning strategies to investigate how the different choices in algorithm design affect the performance in different tasks. For the ease of study, this experiment is tested on a relatively small database constructed on the Princeton shape benchmark. The aim of the second experiment is to study how the proposed algorithm generalizes to different and much larger databases. To this end, we conduct partial-to-partial shape retrieval on the first 200 models of the SHREC 2007 partial matching database [31]. The results show that without changing the parameters, the EMD distance with the three transformational functions constructed for the previous small database still consistently outperform the EMD with the original Euclidean distance. The third experiment, however, is to compare the performance of our approach with the state-of-the-art partial shape matching algorithms based on structural information [6,9]. To this end, we use the 30 hybrid query models to retrieve the 400 database models in the SHREC 2007 partial matching track. Here, a similar learning approach to the one presented in this work is used to obtain a better cross-bin base distance, which further improves the effectiveness of our algorithm. The results suggest that the performance of our algorithm is better than [9] and closely matches that of

[6]. However, it is easy to see that 1) Our algorithm is much simpler than these graph matching approaches and is probably well scaled to large databases (by using the metric embedding techniques). 2) The proposed method is applicable to a wider range of 3D shape formats, such as meshes with holes, polygon soups and even oriented point clouds, while the graph matching algorithms are only tested on watertight 3D shapes without holes and topology errors. 3) The local shape descriptor in our approach can be selected arbitrarily. In the current implementation, we only used the classic spin image descriptor but other more powerful local shape descriptors can be used later to further improve the retrieval performance.

Now, let us introduce the first experiment. Our testing database (termed PSB-52) is composed of 52 3D models in 8 balanced categories manually selected from the Princeton shape benchmark [40]. As shown in Table 1, only the first 6/7 3D models from each category is included in the database since we would like the sizes of each classes to be approximately equal. Although the database is small, some of the shape classes are hard to distinguish. As a result, it is still challenging enough to contrast the performance of different algorithms.

Since we would like to test the effectiveness of the proposed partial similarity measure, in all the three experiments, each 3D model is represented as a bag of local features without recording their 3D spatial relations. The local features here essentially denote the spin image shape signatures [23]. Each spin image characterizes the local shape around its basis point. Therefore, each 3D model is represented as a set of spin images. For detailed feature extraction steps, we first sample $N = 300$ basis points uniformly on the meshes of each 3D model, using the Monte-Carlo approach [34]. Then, Spin-image signatures [23] are computed at these 300 points using 50000 uniform Monte-Carlo point samples on the meshes to avoid the affect of irregular mesh tessellation. The support region for a spin image is set to be within $0.4R$ horizontally (on the tangent plane) and vertically (along the normal) of the basis point, where R is the R.M.S. of the distances from the mesh points of a 3D model to the shape centroid (i.e., R is the “radius” of the shape). The resolution of the spin image signature is set to be 15 by 15. Each spin image is L1-normalized so that all the 225 bins sum to a constant 3500.

To summarize, after the feature extraction procedure, each 3D model is represented by a set of 300 spin image signatures, and there is no explicit ordering between them. Each spin image is itself a 225-dimensional vector. This representation is much simpler than the graph matching based partial shape retrieval approaches [6,9]. In the offline preprocessing stage, the features of all 3D models in the database are calculated and stored. While in the online stage, when a 3D model is specified as query, we would like to find 3D shapes

in the database that are most similar to it. This is achieved by comparing the query model’s feature set with those stored feature sets. Then, the 3D models in the database are ranked according to their similarities to the query.

To measure the performance of the retrieval algorithm, the Precision-Recall plots [40] are evaluated based on the retrieval results and the ground-truth classification of the database models. “Precision” is defined to be the fraction of the retrieved 3D models that have the same category with the query, while “recall” is the fraction of 3D models of the query’s category that have been found in the retrieval process. By this definition, P-R plot measures the complete sensitivity-specificity tradeoffs of a retrieval algorithm. Due to its sensitivity to the retrieval performance, the P-R plot is used in the first two experiments for contrasting different algorithms. In the third experiment, to compare our approach with the two algorithms [9, 6] tested on the SHREC07 partial shape matching track, the Normalized Discounted Cumulated Gain Vector (ND^{CGV}) [22] is used for performance evaluation.

5.1 Global-to-Global Shape Retrieval on the PSB-52 database

We first conduct “Global-to-Global” 3D model retrieval on the PSB-52 database, i.e., both the query shape and the database shapes are in their complete-form. Specifically, four different distance/similarity measures are tested. The first one is the “Earth Mover’s distance” with the original Euclidean distance between spin images as the ground distance without the transformation function. The latter three instances are the “Earth Mover’s distance” with three different parametric transformation functions for ground distance definition. The parameters are manually specified and have not been tuned to be optimal. The four transformation functions (that map the Euclidean base distance to the ground distance in EMD) are listed below:

- 1) $f(x) = x$, i.e. the original Euclidean ground distance;
- 2) $f(x) = a - \frac{1.0}{x+b}$, $a = 10.0$, $b = 0.1$, i.e. “reciprocal” transformation;
- 3) $f(x) = a - \exp(-x/b)$, $a = 2.0$, $b = 100.0$, i.e. “exponential” transformation;
- 4) $f(x) = a + \exp(\frac{x-b}{c}) / (1 + \exp(\frac{x-b}{c}))$, $a = 2.0$, $b = 150.0$, $c = 30.0$, i.e. “sigmoid” transformation.

And we note that the EMD solver is set to take 100 iterations in all the experiments in this section.

The P-R plot corresponding to the four distance/similarity measures is shown in Fig. 7. From the figure, we can see that both the three non-linear transformation functions have better retrieval performance than the Euclidean ground distance. Later, we will show that the profile of the transforma-

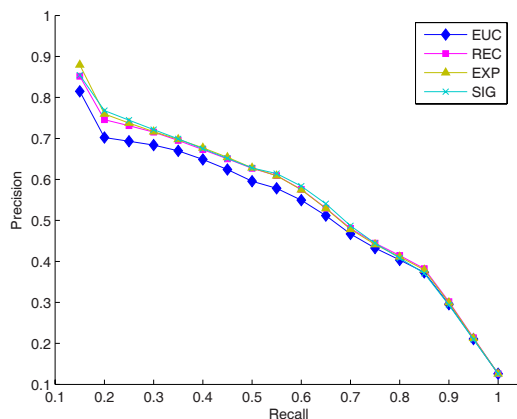


Fig. 7 A comparison of different distance measures for Global-to-Global retrieval (with parametric transformation functions): The EMD distance with EUC: the original Euclidean ground distance; REC: “reciprocal”; EXP: “exponential”; SIG: “sigmoid” transformations. And we note that the EMD solver is set to take 100 iterations in all the experiments in this section.

tion function learnt by the Adaboost algorithm and compare it with the three transformation functions.

Besides, it is worth noting that although the three non-linear transformation functions outperform the Euclidean ground distance in this case, the improvement is not very big. This is because the transformational mechanism is specifically designed for partial similarity based retrieval. To gain a further understanding of this result, we compare the retrieval performances of the four transformation functions for each shape category. The Discounted Cumulative Gain (DCG) [40], is used here for performance comparison. In general, larger DCG scores correspond to higher precision-recall curves.

We could get some detailed observations from Table 2. First, for the shape classes that the Euclidean ground distance has a low DCG score, the three non-linear distance transformations have consistently better retrieval performance. While for shape classes that EUC has a high DCG score already, the transformation may not further increase the retrieval performance. On average, non-linear transformations have better retrieval performance over all classes. However, for some classes, (e.g., hand_body_part and ship_see_vessel) the non-linear transformations actually perform slightly worse than the Euclidean ground distance. In fact, as we will show later, the optimal transformation function tends to be quite dissimilar for different shape classes. So, it is not surprising that a universal transformation function may not perform well on some categories. To this end, we implemented a class (or query) specific learning scheme to obtain the optimal transformation for a given shape category (or query object).


Let us take a closer look at this effect. The non-linear transformations tend to perform better than EUC on shape

Table 2 A comparison of different distance measures for global-to-global retrieval on different shape classes:

	biplane_ airplane	fighter- jet- airplane	human- biped	face- body- part	hand- body- part	ship- see- vessel	head- body- part	aircraft- airplane- glider
EMD	0.612	0.619	0.843	0.770	0.868	0.995	0.791	0.521
REC	0.657	0.704	0.879	0.751	0.868	0.987	0.810	0.535
EXP	0.689	0.730	0.880	0.753	0.841	0.979	0.811	0.577
SIG	0.717	0.705	0.877	0.812	0.769	0.959	0.825	0.617

The EMD distance with EUC: the original Euclidean ground distance; REC: “reciprocal”; EXP: “exponential”; SIG: “sigmoid” transformations. Numbers are the Discounted Cumulative Gain (DCG) scores.

classes with large intra-class variation. This is because two shapes in one of such classes tend to have some dissimilar parts, which may induce a number of large feature-to-feature Euclidean distances. However, for the three non-linear transformations, the “total similarities” between these objects are less affected. As a result, the non-linear transformations excel the Euclidean distance on these shape classes.

On the other hand, for shape classes that  little intra-class variance, the EMD with EUC ground distance tends to have good retrieval performance since the distances between objects in the same class are small. This time, although non-linear transformations will also compute large intra-class similarity scores; it could be more easily affected by a large inter-class similarity. In other words, when two objects in different classes happen to have a similar part, they would produce a false high score. So for these shape classes, the non-linear transformations might not perform better than the Euclidean ground distance.

To summarize, the EMD with the original Euclidean ground distance is sensitive to the discrepancy between objects, while the transformation functions can make the EMD sensitive to the similarity among objects. For the task of partial similarity based retrieval, particularly on a large, un-segmented 3D shape database, discrepancy is nearly universal between these shapes (even from the same class, due to the un-segmented background), while similarity is mainly limited to the intra-class shape pairs. So any direct focus on discrepancy will make us at best find some nearly identical 3D shapes and fail to get more interesting results in general. However, the EMD harnessed with the transformation functions is suited for this task, although some occasional inter-class similarities will slightly affect the retrieval performance. This effect could be mitigated by learning class specific transformation functions.

5.2 Part-based Similarity Retrieval

In this section, we consider part-based similarity retrieval. That is, the 3D shapes in the same class are only partially similar. This is the common scenario in many CBIR applications, and is therefore the main focus of this paper. For ex-

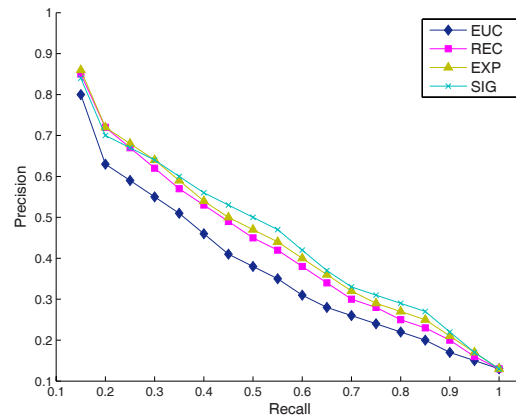


Fig. 8 A comparison of different distance measures for Partial-to-Partial retrieval (with parametric transformation functions): EUC: the Euclidean ground distance; REC: “reciprocal”; EXP: “exponential”; SIG: “sigmoid” transformations.

ample, un-segmented range images contain both foreground and background parts. Only the former is informative, while the latter is irrelevant.

To simulate this effect, we designed a new experimental protocol. The main idea is to randomly extract a surface patch to represent a 3D model, simulating the occlusion effect. More precisely, we randomly select an interest point on the surface of a 3D shape, using the Monte-Carlo sampling technique [34]. Then, we rank the 300 spin image basis points according to their distances to the interested point. Only the nearest 150 (50%) basis points are reserved to represent a 3D shape, while other basis points are discarded, simulating that roughly half of the shape is occluded randomly. Therefore, when two shapes from the same class are compared, it is expected that about 25% of the overall shapes are in correspondence, since they are occluded independently. We first examine the task of partial-to-partial shape retrieval, where both the query shape and the database shapes are using this “occluded” shape representation. This is a very challenging task, which is as hard as retrieving un-segmented range images using an un-segmented range image as query, where the foreground takes about 50% of the area.

Fig. 8 shows the precision-recall curves for four distance measures: the EMD with Euclidean ground distance and three non-linear transformation functions. We can see that the three transformations consistently perform much better than EMD, which demonstrates the obvious advantage of the proposed approach for 3D partial shape retrieval. Particularly, the PSM with sigmoid transformation has the best performance among the four methods. This suggests that the optimal distance mapping may have the soft-thresholding characteristics. Later, we will show that the transformation function learnt by the Adaboost algorithm indeed has the thresholding characteri

To compare with the case of global shape retrieval, we also present the DCG scores of the four distance measures on different shape classes in this partial-to-partial retrieval task, as shown in Table 3. We can see from the table that, in this case, the transformations perform better than (or as good as) the Euclidean ground distance on most shape classes. Contrary to global-to-global shape retrieval, there is no shape category that the three transformations are consistently worse than EUC. This intuitively makes sense, as the “simulated occlusion” feature extraction strategy significantly increases the intra-class variation, making the similarity-based criterion guided by the transformations excel the discrepancy-based EMD with the Euclidean ground distance.

Another related retrieval task is “part-in-whole” search, where a user selects an interested region on a query 3D shape, and she/he wants to find the database shapes that contain a similar part to the interested region. In our experiments, we apply the “simulated occlusion” feature extraction strategy to the query 3D model, and use the whole shapes for the 3D models in the database. Note that the total weights for the query and a database shape might not be equal. Two weight assignment strategies are tested here: Type I: We know each feature of the query shape precisely corresponds to one feature of a database shape. In this case, each spin image feature of both the two types of shapes associated with weight $1/300$. So the total weights of the query are 0.5, and the total weights for a database model are 1.0. Here, relevant features could match perfectly. Type II: We do not know the relative feature density on the two types of shapes. In this case, identical total weights, e.g. 1.0, are adopted for both of the query and database shapes. In this experimental setting, the features of the query object are essentially duplicated and many irrelevant feature-matches are found in solving the EMD distance.

Fig. 9 shows the precision-recall plot for the Type I case of “part-in-whole” retrieval. Again, the three non-linear transformation functions perform better than the Euclidean ground distance. However, the advantages are not as significant as that of the “partial-to-partial” retrieval case, as there is no enforcement of forming irrelevant feature pairings.

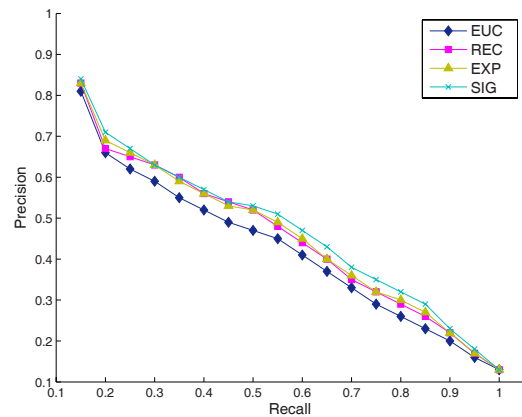


Fig. 9 A comparison of different distance measures for Type I part-in-whole retrieval (with parametric transformation functions): EUC: the Euclidean ground distance; REC: “reciprocal”; EXP: “exponential”; SIG: “sigmoid” transformations.

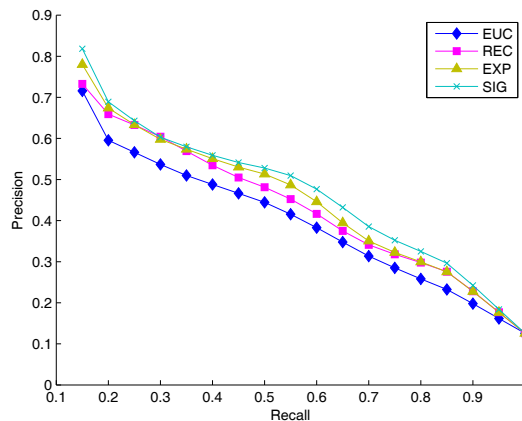


Fig. 10 A comparison of different distance measures for Type II part-in-whole retrieval (with parametric transformation functions): EUC: the Euclidean ground distance; REC: “reciprocal”; EXP: “exponential”; SIG: “sigmoid” transformations.

Fig. 10 shows the precision-recall plot for the Type II case of “part-in-whole” retrieval. It is easy to see that the performance gaps between the transformation functions with the Euclidean ground distance are enlarged. This suggests that the proposed distance transformational mechanism is a powerful way to get rid of the affect of irrelevant feature pairings. Besides, as discussed in Section 3.3, we can unify the total weights of different feature sets in order to exploit the metric property of our approach (Theorem 1) to speed up retrieval on large databases.

5.3 Learning Distance Transformation Function

Up to now, the functional forms as well as the parameters of the distance transformation functions are specified manually. We have shown that the three non-linear functions ob-

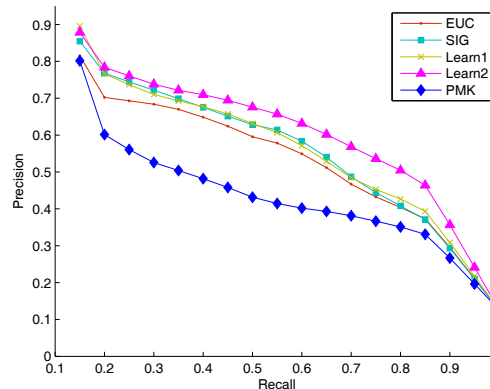
Table 3 A comparison of different distance measures for partial-to-partial retrieval on different shape class:

	biplane_ airplane	fighter_ jet_ airplane	human_ biped	face_ body_ part	hand_ body_ part	ship_ see_ vessel	head_ body_ part	aircraft_ airplane_ glider
EUC	0.562	0.647	0.798	0.694	0.734	0.675	0.635	0.501
REC	0.699	0.693	0.831	0.681	0.737	0.738	0.654	0.545
EXP	0.699	0.687	0.821	0.692	0.736	0.766	0.618	0.573
SIG	0.727	0.636	0.836	0.745	0.646	0.756	0.661	0.629

tained consistently better results than the Euclidean ground distance on partial shape retrieval, which convincingly demonstrates the effectiveness of the transformation mechanism and its good generalization property, as the improved performance does not rely on a particular function. However, it is more interesting to try to learn the transformation directly from data. In particular, we would like to see what shape the learnt transformation function looks like. Note that in this section, our primary focus is to improve the performance of the current retrieval task by exploiting available distance comparison information. For this purpose, the learning and testing procedures are conducted on the same database. This is the case for many practical retrieval tasks where prior knowledge and/or user feedbacks are available. However, from the machine learning viewpoint, it is important to demonstrate the good generalization property of a learning algorithm. We will address this issue in great detail in Section 5.6. The results there suggest that the difference between the empirical risk and structural risk is often negligible, i.e., the proposed learning algorithm generalizes very well beyond training data. As a result, the current experimental results also make sense in the generalization context.

We consider two kinds of learning. One is to learn an all-purpose transformation function for all 3D shapes in a database, while the other is to learn a class-specific transformation for each shape class, like the query-sensitive embedding approach [2]. In this latter learning strategy, the training set $S = \{(q_1, a_1, b_1), \dots, (q_T, a_T, b_T)\}$ consists T triplets of objects, where $q_t, t = 1, \dots, T$ are sampled from all object categories. Without loss of generality, q_t, a_t belong to the same class, while q_t, b_t belong to different classes. In the latter learning strategy, the training set only consists of the triplets where q_t, a_t belong to a particular shape class. As a result, in both of the two cases, it is expected that $D_A(q_t, a_t) < D_A(q_t, b_t)$ for an ideal distance measure, i.e., all the triplets are associated with label 1.

Following the learning method introduced in Section 4, we initialize the feature matching histogram between two shapes by solving the EMD with Euclidean ground distance. Then, the composition coefficients $A = \{\alpha_k\}, k = 1, \dots, K$ are learnt by the restricted Adaboost algorithm. The distance between two shapes is computed based on the feature matching histogram and the learnt coefficients α_k . Strictly speaking, this is not perfect in theory, since the computation of

**Fig. 11** A Comparison of different similarity measures for global-to-global retrieval. EUC: the Euclidean ground distance; SIG: the “sigmoid” parametric transformation. Learn1: Learning an all-purpose transformation function. Learn2: Learning class-specific transformation functions. PMK: The pyramid match kernel algorithm (the best performance of using it as a distance or similarity measure is shown here)

feature matching histogram (based on Euclidean distance) and the summation of transformed (ground) distances between feature-pairs in EMD are based on different criteria. To further investigate this issue, we also test using the EMD with a parametric transformation function, e.g., sigmoid, to compute the feature matching histogram, and then learn the coefficient α_k . The results suggest that using a better feature matching histogram indeed improves the performance of the learning algorithm, which validates the consistency of the proposed framework.

In Fig. 11, we compare the two distance learning methods for global-to-global shape retrieval by plotting their precision-recall curves. “Learn1” denotes the method for learning an all-purpose transformation function; while “Learn2” is to learn a specific transformation function for each category. For reference, we also reproduce the precision-recall curves of the EMDs with the Euclidean ground distance (EUC) and with the sigmoid transformation (SIG). Furthermore, we also plot the P-R curve of an efficient, state-of-the-art algorithm for feature-sets comparison, the pyramid match kernels (PMK) [18]. We can see that all the three approaches using the transformation mechanism (SIG, Learn1 and Learn 2) have better performance than the original Euclidean ground

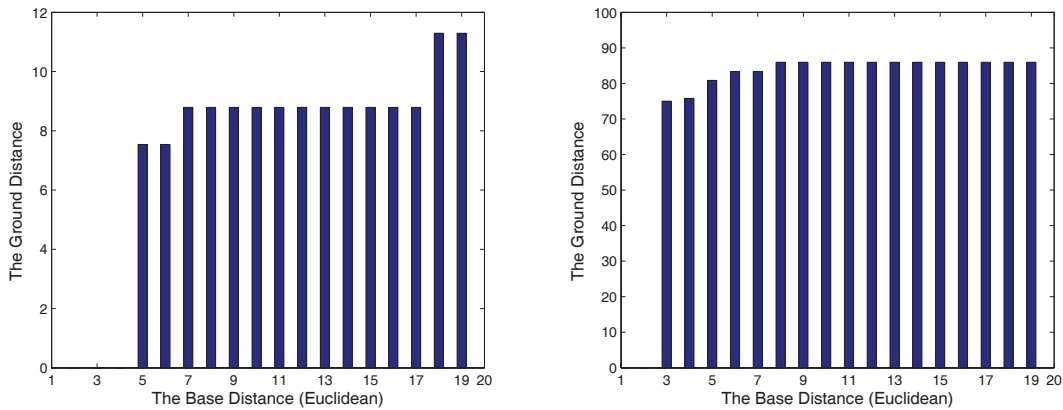


Fig. 12 The learnt class-specific distance transformation functions for two classes. Left: biplane_airplane class Right: fighter_jet_airplane.

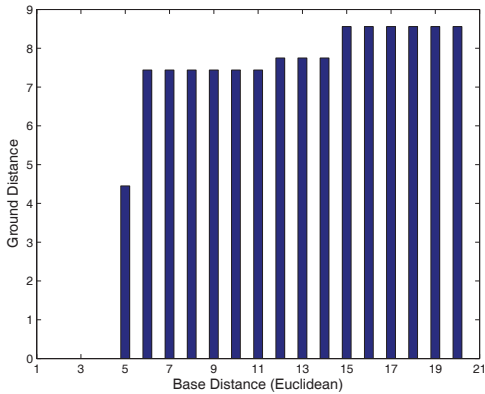


Fig. 13 The profile of the all-purpose distance transformation function.

distance (EUC) and the PMK algorithm. Specifically, the performance of the parametric transformation function (SIG) is comparable with the all-purpose learning scheme (Learn1), while the class-specific learning scheme (Learn2) has the best performance among the five methods. Finally, we would like to remark that though the PMK compares favorably with many other feature-sets comparison approaches [18], the main focus of it is on the efficiency issue. So it's not surprising that the PMK has the worst performance among the five algorithms.

Note that the class (query)-specific learning approach is very flexible. Since user intervention and other types of knowledge can be conveniently represented as a number of triplets (each encoding a relative distance comparison), and the learning is very efficient, the transformation function can be tuned dynamically in light of new knowledge.

In Fig. 13, we plot the profile of the all-purpose distance transformation function. We can see from the figure that it looks like a soft-thresholding function. When the base distance is small, the ground distance is zero. When the base distance increases to a particular value, the ground distance

suddenly jumps to a very large value. Then, as the base distance continues to increase, the ground distance augments very slightly until it becomes saturated. In Fig. 12, we also plot the profiles of two class-specific transformation functions. It can be seen that the two functions basically have the soft-thresholding shape, like the all-purpose transformation function. However, their threshold value differs, due to their class-specific characteristics. In Fig. 11, we have shown that the retrieval performance increases significantly by exploiting this class-specific information.

We also apply the learning methods to “partial-to-partial” and “part-in whole” retrieval. Fig. 14 shows the precision-recall plots for the “partial-to-partial” retrieval task. There are two types of training methods “general purpose” and “class specific”, and the transformation functions are learnt either from “global-to-global” or “partial-to-partial” feature matching histograms. The purpose here is to study whether the transformations learnt from “global-to-global” feature matching histograms are suitable for “partial-to-partial” retrieval. “Learn1” is to learn a “general purpose” transformation function on “partial-to-partial” feature matching histograms; “Learn2” is to learn a “class-specific” transformation function on “partial-to-partial” feature matching histograms. “Learn3” and “Learn4” are the “general purpose” and “class-specific” learning schemes based on “global-to-global” feature matching histograms. “EUC” is the precision-recall curve of the Euclidean ground distance for “partial-to-partial”, reproduced here for comparison.

From Fig. 14, we can see that both the four learning schemes outperform the Euclidean ground distance. The transformation functions (Learn3 and Learn4) learnt from “global-to-global” feature pairing are in general appropriate for “partial-to-partial” retrieval. However, the transformation functions learnt from “partial-to-partial” feature pairing (Learn1 and Learn2) perform better in the same task. Particularly, “Learn2” is the best one among the five methods, which again demonstrates the superiority of “class-specific” learning.

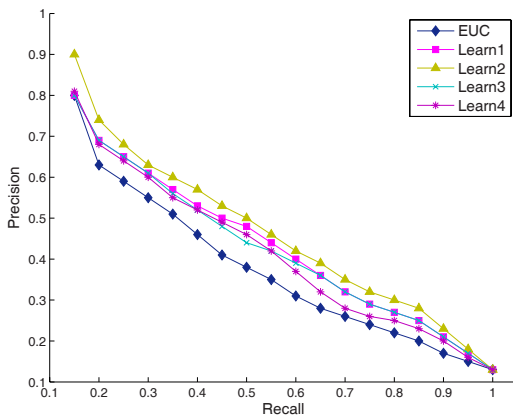


Fig. 14 A Comparison of different learning methods for partial-to-partial retrieval. See texts for details of Learn1-Learn4.

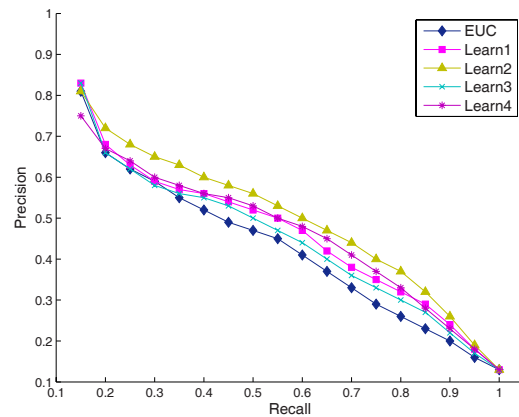


Fig. 16 A Comparison of different learning methods for Type I part-in-whole retrieval. See texts for details of Learn1-Learn4.

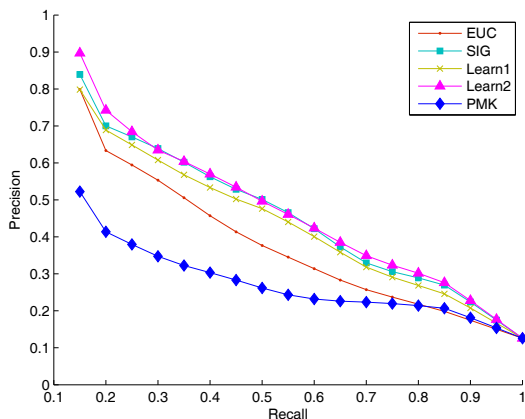


Fig. 15 A Comparison of different similarity measures for partial-to-partial retrieval. EUC: the Euclidean ground distance; SIG: the “sigmoid” parametric transformation. Learn1: Learning an all-purpose transformation function. Learn2: Learning class-specific transformation functions. Both Learn1 and Learn2 are based on “partial-to-partial” feature matching histograms. PMK: The pyramid match kernel algorithm (the best performance of using it as a distance or similarity measure is shown here)

Fig. 15 compares the performance of learnt transformation functions with the parametric transformation functions in the partial-to-partial retrieval task. Here, “Learn1” and “Learn2” are reproduced from the two curves with the same name in Fig. 14, while EUC, SIG are the EMDs with the Euclidean ground distance and the sigmoid transformation function, respectively. Again, PMK denotes the pyramid match kernel algorithm [18]. It is clear from the figure that all the three EMD distances with non-linear transformation functions obtained better performance than EUC and PMK. Besides, the performance of the class-specific learning scheme (Learn2) is comparable to the sigmoid parametric transformation function (SIG), while the performance of the all-

purpose learning scheme (Learn1) is slightly worse than SIG and Learn2.

We also compare the different learning methods for Type I part-in-whole retrieval, shown in Fig. 16. Similarly, “Learn1” and “Learn2” are the “general purpose” and “class-specific” learning schemes based on “partial-to-global” feature matching histograms; while “Learn3” and “Learn4” are the “general purpose” and “class-specific” learning schemes based on “global-to-global” feature matching histograms. “EUC” denotes using the EMD with the Euclidean ground distance for “part-in-whole” retrieval, reproduced here for comparison. Clearly, once again, the four learning methods perform better than “EUC”. And, as we expected, “Learn2” again has the best performance among the five methods, demonstrating the advantage of “class-specific” learning.

5.4 The Advantage of a Good Feature Matching Histogram

As mentioned before, the feature matching histograms are generated by solving the EMD with the Euclidean ground distance (EUC). And the learning of transformation function is performed on the feature matching histograms. This is not in full consistency, as the feature matching patterns produced by EUC may be somewhat affected by irrelevant feature pairings. Ideally, the learning should be conducted recursively, so that the transformation function learnt in the last round is used by EMD to generate the current feature matching histogram. However, the transformation functions are only learnt in a histogram representation, which prevents well-defined recursive learning in a strict sense.

To examine whether better learning results could be obtained by using a better feature matching histogram, we run EMD with a parametric transformation function to compute the feature matching patterns. Then, learning is performed on the resulting feature matching histogram. Here, the sigmoid function introduced in Section 5.3 is chosen as the

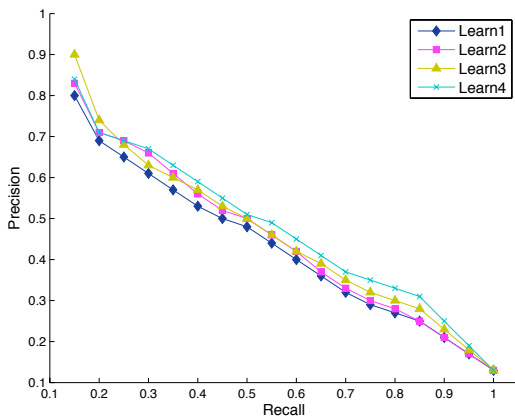


Fig. 17 A Comparison of different feature matching histograms for partial-to-partial retrieval.

parametric transformation function, as its shape is closest to the learnt functions.

The results of using different feature matching histograms for partial-to-partial retrieval are shown in Fig. 17. “Learn1” and “Learn3” are the “general purpose” and “class-specific” learning schemes based on the feature matching histograms of EUC, reproduced here for comparison; while “Learn2” and “Learn4” are the “general purpose” and “class-specific” learning schemes based on the feature matching histograms induced by the sigmoid transformation function. We can see from the figure that “Learn2” is better than “Learn1”, and “Learn4” is better than “Learn3”, though the differences are not large. These findings indicate that the feature pairings generated by the sigmoid function are also better than those with the original Euclidean distance, which justifies the theoretical consistency of the proposed approach.

5.5 The Effectiveness and Efficiency of Our Algorithm on Larger Databases

We have carefully tested how different parameter settings and building blocks of our algorithm may affect its performance on a variety of retrieval tasks. Now, our primary focus is to study the scalability and time-efficiency issues of our algorithm on larger databases. For this purpose, the database of SHREC07 partial matching track [31] is used, which is composed of 400 watertight 3D models in 20 categories (with 20 models in each category) and 30 mixed query models. In this section, we only perform partial-to-partial shape retrieval on the first 200 models in the database (i.e., the models in the first 10 classes). While in Section 5.6, the 30 query models will be used to retrieve the 400 database models for comparing the performance of our algorithm with two partial shape matching algorithms based on structural information of 3D shapes [9, 6].

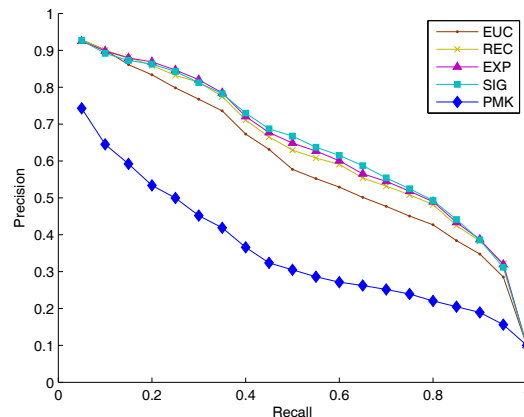


Fig. 18 A comparison of different distance measures for Partial-to-Partial retrieval (with parametric transformation functions) on the first 200 models of the SHREC07 database. EUC: the Euclidean ground distance; REC: “reciprocal”; EXP: “exponential”; SIG: “sigmoid” transformations. The performance of the pyramid match kernel algorithm (PMK) is also plotted.

We use the same partial shape feature extraction strategy as described in Section 5.2. That is, each 3D model is represented using the 150 (50%) spin image features nearest to a randomly selected interest point. On an ordinary PC computer, the average time of extracting the 300 spin images for each full shape is 3.02 sec. Then, each 3D model is used to retrieve the remaining shapes in the database. We first examine whether the transformational mechanism still works well in this scenario. To the end, we use the EMD distance with the four transformation functions in Section 5.1 (EUC, REC, EXP, SIG) to quantify partial shape similarities. The average per query time for each transformation function is: EUC: 8.37 sec; REC: 8.48 sec; EXP: 8.99 sec; SIG: 9.58 sec. As a result, the transformation mechanism only slightly increases the computational time of the Earth Mover’s distance. The Precision-Recall plots of these methods are shown in Fig. 18.

It is clear from Fig. 18 that the three non-linear transformation functions (REC, EXP, SIG) in EMD have better performance than the Euclidean ground distance (EUC) and the pyramid match kernel algorithm (PMK), and the sigmoid transformation (SIG) has the best performance among the five approaches.

We also try to learn the distance transformation functions using the algorithm presented in Section 4. Specifically, the learning is performed on the partial-to-partial feature matching histograms induced by the Euclidean ground distance (EUC) and the sigmoid transformation (SIG), and the all-purpose and the class-specific learning schemes are also tested. Fig. 19 compares the performance of the above four combinatorial learning strategies, Learn1: all-purpose learning on the EUC feature matching histogram; Learn2: all-purpose learning on the SIG feature matching histogram;

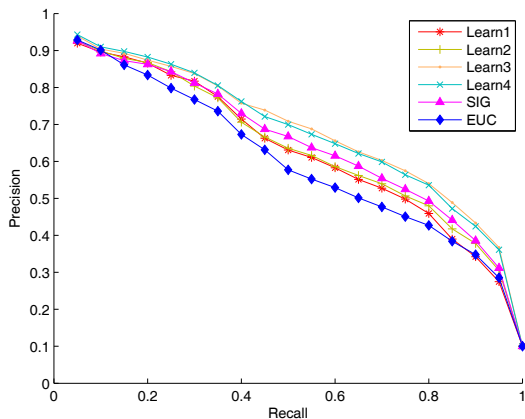


Fig. 19 A comparison of different learning methods for Partial-to-Partial retrieval on the first 200 models of the SHREC07 database. See the texts for details of Learn1-Learn4. For comparison purpose, we also reproduce the sigmoid (SIG) parametric transformation function in Fig. 18

Learn3: class-specific learning on the EUC feature matching histogram; Learn4: class-specific learning on the SIG feature matching histogram. The amortized learning time for each query (i.e., the learning time divided by 200) is, Learn1: 0.535 sec; Learn2: 0.495 sec; Learn3: 1.275 sec; Learn4: 1.125 sec. The time for generating the feature matching histogram is essentially similar to the time for computing the pairwise EMD distances between the 200 3D models.

We can see from Fig. 19 that both the four learning methods have better retrieval performance than the Euclidean ground distance (EUC). Specifically, the performance of the all-purpose learning scheme is slightly worse than the sigmoid transformation function (SIG) and the performance of class-specific learning scheme is better than SIG.

5.6 The Generalization Performance of the Proposed Learning Algorithm

In the previous sections, we have shown that the proposed learning algorithm is very effective for improving the performance of partial shape retrieval by exploiting the relative distance comparison information in the triplets. However, from the machine learning perspective, it is important to show that the learnt distance transformation function is not simply over-fitted to the training data, i.e., it should also generalize well on other related datasets.

To this end, we randomly divide the 200 3D models from the first 10 classes of the SHREC07 partial matching track [31] into a training set and a test set. Specifically, for each class, 10 out of the 20 3D models are randomly included in the training set while the remaining 10 models are assigned to the test set. Therefore, both of the training and test

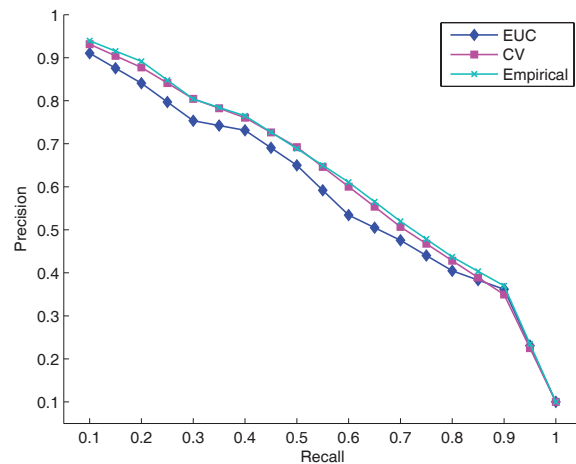


Fig. 20 The Precision-Recall plots for three partial-to-partial shape retrieval approaches on the SHREC07 test dataset. 1) EUC: the EMD distance with the Euclidean ground distance; 2) CV: The cross-validation experimental scheme, where the distance transformation function is learnt from the SHREC07 training dataset, and is used to perform shape retrieval on the test dataset. 3) Empirical: Both the learning and testing of the transformation function are performed on the test dataset. In this case, the Precision-Recall plot could be over-fitted and is only representative of the empirical risk.

composition. Then, similar to Section 5.5, we represent each 3D model in the two sets with 150 spin images nearest to a random interest point to conduct partial-to-partial shape retrieval on the two datasets. To test the efficacy of the proposed learning algorithm, the classification of the 3D models in the training set is employed to learn an all-purpose distance transformation function. Then, the function is applied to the feature matching histogram of the test set to perform partial-to-partial shape retrieval. In this approach, both the feature-matching histograms of the training set and test set are computed by the EMD with Euclidean ground distance, and no class-label information about the test set is used. The retrieval results of the aforementioned approach (Cross-Validation) are compared with two alternative methods on the test set: 1) Empirical: The distance transformation function is also learnt from the test set; 2) EUC: The EMD with Euclidean ground distance is used directly to perform shape retrieval. The precision-recall plots of all the three experimental settings are compared in Fig. 20.

From Fig. 20, we can see that the “empirical” approach is only slightly better than the “cross-validation” approach. However, both of the two methods have much better retrieval performance than the “EUC” setting. This finding suggests that the proposed learning algorithm generalizes very well beyond training data, since there is not much difference if the learning process is performed on the training set (“cross-validation” approach) or on the test set (the “empirical” approach). Besides, the advantage of distance transformation function over a naïve ground distance definition (EUC) is demonstrated again.

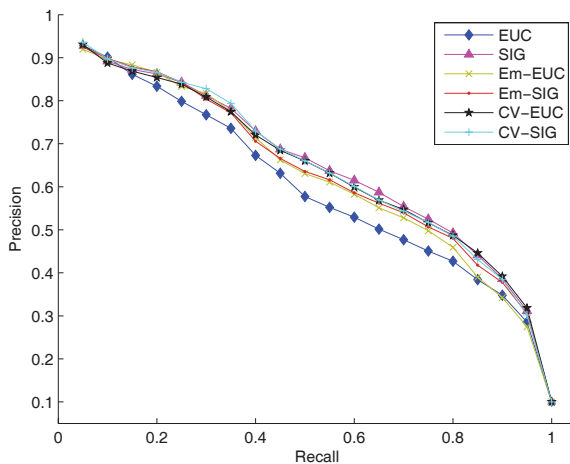


Fig. 21 The Precision-Recall plots for six partial-to-partial shape retrieval approaches on the SHREC07 200 model database. 1) EUC: the EMD distance with the Euclidean ground distance (EUC in Fig. 18); 2) SIG: The EMD distance with the sigmoid transformation function (SIG in Fig. 18); 3) Em-EUC: The transformation function is learnt from the SHREC07 200 model database with Euclidean distance induced feature matching histogram (Learn1 in Fig. 19); 4) Em-SIG: Similar to Em-EUC, but with the sigmoid function induced feature matching histogram (Learn2 in Fig. 19). 5) CV-EUC: The transformation function is learnt from the PSB-52 3D model database with Euclidean distance induced feature matching histogram; 6) CV-SIG: Similar to CV-EUC, but learning is performed on the sigmoid function induced feature matching histogram.

Motivated by the fact that the three nonlinear parametric transformation functions introduced in Section 5.1 work well across different 3D model databases, we are interested in examining whether the distance transformation functions learnt on one database still work well on another heterogeneous database. To this end, we consider a more challenging retrieval task: First, we learn an all-purpose distance transformation on the PSB-52 database. Then, this function is used to perform shape retrieval on the first 200 models of the SHREC07 database. Since the two databases are constructed by different researchers and having very different shape classes, this task is much harder than traditional supervised learning problems, in which both the training set and test set are assumed to follow the same, but unknown distribution. Specifically, we first learn two all-purpose transformation functions from the PSB-52 database, based on the partial-to-partial feature matching histograms generated by the EMDs with Euclidean ground distance and the sigmoid transformation function, respectively. Then, the two transformation functions are applied to the SHREC07 200 model database to perform partial-to-partial shape retrieval (termed CV-EUC and CV-SIG, respectively). Note that the feature matching histogram of the SHREC07 200 model database is generated by the EMD with Euclidean ground distance. As a result, no *a priori* information about the SHREC07 200 model database or its transformation function is used in these experimental settings.

Fig. 21 compares the performance of the two cross-validation approaches with four other retrieval settings on the SHREC07 200 model database. We can see from the figure that both the 5 approaches with a non-linear distance transformation function have much better retrieval performance than the EMD with the Euclidean ground distance. More remarkably, the performance of the two cross-validation schemes (CV-EUC and CV-SIG) is comparable to or even slightly better than the two empirical learning schemes (Em-EUC and Em-SIG). This observation again strongly demonstrates the good generalization property of our algorithm: The learnt distance transformation function works equally well on some potentially heterogeneous dataset.

It can be concluded from the experiments above that the performance improvement of the learnt transformation function is not simply an artifact of over-fitting to a particular dataset. It generalizes very well over a broad application scope. As a result, not only the proposed learning algorithm is useful for a particular retrieval task at hand, but also it is able to work well on many pattern recognition and machine learning problems which involve quantifying the partial similarities between feature-sets, e.g., learning the Gram-matrix for kernel machines, as we pointed out at the end of Section 4.

5.7 Comparison with Previous 3D Partial Shape Matching Algorithms

It is interesting to compare the performance of our approach with previous 3D partial shape matching algorithms. To this end, we use the 30 hybrid query models in the SHREC07 partial matching dataset to retrieve the 400 database models [31], as the performance of two graph-matching based partial retrieval algorithms [6,9] on the same task is available.

As we have mentioned before, all the 430 shapes in this database are watertight 3D models. Each of the 400 database models belongs to one of 20 equal-sized shape classes, and each of the 30 hybrid query models is highly-relevant or marginally relevant to several classes of database models, as they are generated by fusing several parts of different 3D shapes. The ground-truth classification of these 3D models is available at [31].

Different from the above experiments on partial shape retrieval, our task here is to evaluate the partial similarity between the full shapes of 3D models. As a result, we use the feature extraction strategy described in Section 5.1, i.e., each 3D model is represented by 300 spin image features that are uniformly distributed on the meshes. To make the comparison between different algorithms feasible, the “Normalized Discounted Cumulated Gain Vector” (NDCG) [22] is used in this section for performance evaluation. Similar to the Precision-Recall plot, higher NDCG curves correspond to better retrieval results.

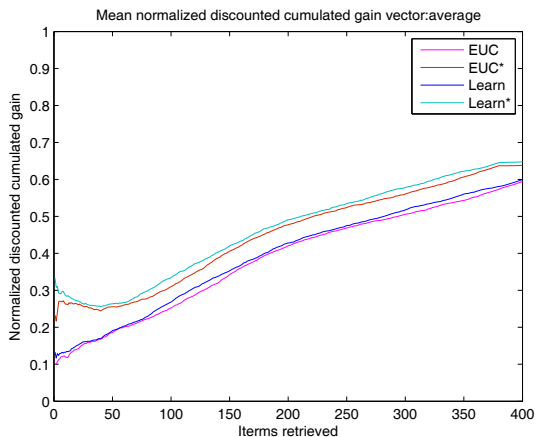


Fig. 22 The performance of our algorithms on the original spin-image features. Both highly and marginally relevant models are considered. “EUC” is the EMD distance with Euclidean ground distance and “Learn” is to learn an all-purpose transformation function based on the feature matching patterns generated by the “EXP” transformation function. “EUC*” and “Learn*” are the results of the two algorithms after removing the influence of the “spring” class.

Fig. 22 shows the performance of two representative algorithmic settings (“EUC” and “Learn”) of our approach. Details can be found from the caption of Fig. 22. Unfortunately, the performance of these two algorithms is not satisfactory. Upon a closer look at the retrieval results, we found that spin image features of the model in the spring class are degenerated. As a result, these models often appear on the top of the retrieval list of many query models. This is because the springs are highly structured 3D shapes with very large curvatures. The rapid change of the direction of normals makes the spin images quite unstable. To eliminate this effect, the distances between unrelated query models to the spring shapes are set to be a large value (Spring shapes can be easily identified from the curvature distributions, so this is feasible). The NDCG curves of “Euc*” and “Learn*” in Fig. 22 show that the retrieval results are greatly improved. From the panel (b) of Figure 3 in [31], we can see that the performance of our algorithms are comparable to, or even slightly better than the Many-to-Many shape matching approach [9]. However, the performance of our algorithms still does not reach that of the sub-part correspondence method [6].

To further boost the performance of our approach, we propose a new approach to learn a better base distance function for spin image features. Our basic observation is that, since different bins in the spin image descriptor are spatially related, a bin-to-bin Euclidean base distance is not an ideal choice, a cross-bin base distance could be better. As a typical example of cross-bin distance between vectors is the (squared) Mahalanobis distance, we try to learn a Maha-

lanobis distance as the base distance in our framework:

$$d_A(p_i, q_j) = (p_i - q_j)^T A (p_i - q_j), \quad (18)$$

where p_i, q_j are the i -th and j -th (column) feature of two 3D models, and A is a positive semi-definite matrix. There are a number of approaches for learning a Mahalanobis distance, e.g. [11]. However, the class labels of each features are required in these methods. In our problem, only the class label of *feature-sets* are available, so it is necessary to overcome this bottleneck. To this end, we propose a new learning scheme similar to the recursive Adaboost learning procedure introduced earlier. In the first step, we compute the feature matching patterns of two sets: f_{ij} by solving the Earth Mover’s distance with a Mahalanobis ground distance. In the second step, based on the observation that:

$$\begin{aligned} & \sum_{i,j} f_{ij} (x_i - x_j)^T A (x_i - x_j) \\ &= \sum_{i,j} f_{ij} \text{tr} \left((x_i - x_j)^T A (x_i - x_j) \right) \\ &= \sum_{i,j} f_{ij} \text{tr} \left(A (x_i - x_j) (x_i - x_j)^T \right) \\ &= \text{tr} \left(A \sum_{i,j} f_{ij} (x_i - x_j) (x_i - x_j)^T \right) \end{aligned} \quad (19)$$

we can directly use the approach [11] to learn the matrix A in the Mahalanobis distance, since it suffices to replace $(x_i - x_j)(x_i - x_j)^T$ in the original optimization procedure [11] with $\sum_{(i,j)} f_{ij} (x_i - x_j)(x_i - x_j)^T$. The above two steps are iterated until convergence. More details can be found in our technical report [45].

Besides, we can always factorize the learned matrix A as $A = R^T R$, e.g., using Cholesky decomposition, as A is a positive semi-definite matrix. Then the Mahalanobis distance can be rewritten as

$$d_A(p_i, q_j) = [R(p_i - q_j)]^T [R(p_i - q_j)]. \quad (20)$$

So if we replace p_i and q_j with Rp_i and Rq_j , we can simply compute the Mahalanobis distance between p_i and q_j as a squared Euclidean distance between Rp_i and Rq_j .

We conduct the aforementioned learning procedure on a database with 52 models from the Princeton Shape Benchmark, obtaining the matrix A and R . Then we project each spin image feature from p_i to Rp_i . To avoid the influence of irrelevant feature pairings, we only use the Mahalanobis distance as a base distance in EMD, and define the ground distance by using a parametric transformation function $y = \sqrt{x}$. (This approach is denoted as “Maha” in Fig. 23.) Besides, we also learn an all-purpose transformation function based on the resulting feature matching histogram (denoted as “Learn” in Fig. 23). Finally, the affect of the spring class

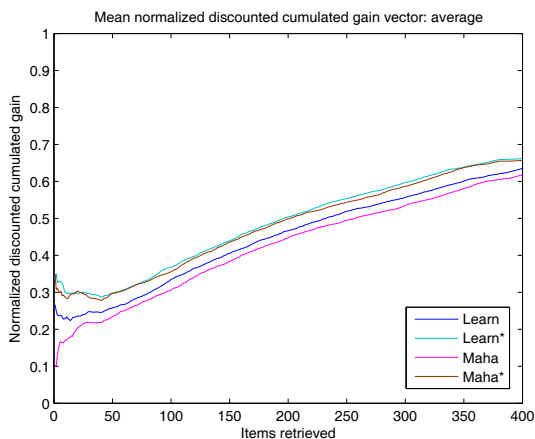


Fig. 23 The improved retrieval performance with an improved base distance.

is removed, resulting “Maha*” and “Learn*”, also shown in Fig. 23.

Now, we can see that the performance of “Learn*” (the rightmost NDCG score is 0.66) closely matches the final performance of sub-part correspondence method (which is smaller than 0.7) [6]. Our approach is lower on a small range at the left-hand side), and it is better than the Many-to-Many shape matching approach [9]. This is good news for our approach since: 1) It is much simpler than the above two graph matching algorithms and is provably well up to large databases (by using the metric embedding techniques); 2) Our method is applicable to a wider range of 3D shape formats, such as meshes with holes, polygon soups and even oriented point clouds, while the graph matching algorithms are only tested on watertight 3D shapes without holes and without topology errors; 3) In our approach, the local shape descriptor can be selected arbitrarily. In the current implementation, we only used the classic spin image descriptor but other more powerful local shape descriptors can be used to further improve the retrieval performance. Note that, albeit our current approach currently is not suitable for handling highly structured shape classes (spring), it can be compensated by using a structured local shape descriptor [6]. However, it would be harder for the graph matching algorithm to deal with general 3D shapes which could have many openings and with many topological errors.

6 Conclusion

In this paper, we propose a novel transformation mechanism for better evaluating the partial similarity between two feature sets. Specifically, the robustness of the EMD distance with respect to irrelevant feature pairings can be greatly improved with the introduction of distance transformation functions. Also, we prove that under certain conditions, the trans-

formation function is able to keep the metric property from the base distance to the ground distance, thereby enabling the use of metric embedding methods to scale-up our method to large databases.

We also propose a supervised learning approach for approximately learning the distance transformation function based on the Adaboost algorithm. Specifically, a modification of the original Adaboost optimization procedure is developed to guarantee the monotonicity of the learnt transformation function. The class-specific learning scheme is also developed to further enhance the power of the learning method. Finally, extensive experiments on 3D partial shape retrieval convincingly demonstrate the effectiveness of the proposed algorithms and their superiorities on a variety of CBIR applications.

Acknowledgements We sincerely thank for the thoughtful comments anonymous reviewers. Yi Liu, Xu-Lei Wang and Prof. Hongbin Ma are supported by NKBRP grant 2004CB318005 and NSFC grant 60803067. Professor Hong Qin (Stony Brook University)’s current research in this paper is partially supported by NSF grants: IIS0949467, IIS0710819, and IIS0830183.

References

1. V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios. BoostMap: A Method for Efficient Approximate Similarity Rankings. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2004.
2. V. Athitsos, M. Hadjieleftheriou, G. Kollios, and S. Sclaroff. Query-Sensitive Embeddings. In Proceedings of ACM SIGMOD conference, 2005.
3. S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Context, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, pp. 509-522, 2002.
4. S. Berretti, A. Del Bimbo, and P. Pala. Retrieval by Shape Similarity with Perceptual Distance and Effective Indexing, IEEE Transactions on Multimedia, vol. 2, pp. 225-239, 2000.
5. D. Bespalov, A. Shokoufandeh, W. C. Regli and W. Sun. Scale space representation of 3D models and topological matching. In Proceedings of Symposium on Solid Modeling and Applications, 2003.
6. S. Biasotti, S. Marini, M. Spagnuolo, B. Falcidieno. Sub-part correspondence by structural descriptors of 3D shapes. Computer Aided Design, vol. 38, pp. 1002-1019, 2006.
7. J. Bourgain. On Lipschitz embedding of finite metric spaces in Hilbert space. Israel J. of Math, vol. 52, pp. 46-52, 1985.
8. B. Bustos, D. A. Keim, D. Saupe, T. Schreck, and D. V. Vranic. An experimental effectiveness comparison of methods for 3D similarity search. International Journal on Digital Libraries, vol. 6, pp. 39-54, 2006.
9. N. D. Cornea, M. F. Demirci, D. Silver, A. Shokoufandeh, S. J. Dickinson and P. B. Kantor. 3D Object Retrieval using Many-to-many Matching of Curve Skeletons. In Proceedings of International Conference on Shape Modeling and Applications, 2005.
10. K M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-Sensitive Hashing Scheme Based on p-Stable Distributions, In Proceedings of ACM Symposium on Computational Geometry, 2004.
11. J. V. Davis and B. Kulis and P. Jain and S. Sra and I. S. Dhillon. Information-Theoretic Metric Learning. In Proceedings of the 24th International Conference on Machine Learning, 2007.

12. T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs. "A Search Engine for 3D Models," *ACM Transactions on Graphics*, vol. 22, pp. 83-105, 2003.
13. T. Funkhouser and Philip Shilane. "Partial Matching of 3D Shapes with Priority-Driven Search," In *Proceedings of Symposium on Geometry Processing*, 2006.
14. R. Gal and D. Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics*, vol. 25, pp. 130-150, 2006.
15. K. Grauman and T. Darrell. Efficient Image Matching with Distributions of Local Invariant Features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005.
16. K. Grauman and T. Darrell. The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features. In *Proceedings of International Conference on Computer Vision*, 2005.
17. K. Grauman and T. Darrell. Approximate Correspondences in High Dimensions. In *Proceedings of Advances in Neural Information Processing Systems 19 (NIPS)*, 2007.
18. K. Grauman, T. Darrell. The Pyramid Match Kernel: Efficient Learning with Sets of Features. *Journal of Machine Learning Research (JMLR)*, vol. 8, pp. 725-760, 2007.
19. T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*, Springer-Verlag, New York, 2001.
20. D. Huttenlocher and P. Felzenszwalb, Pictorial Structures for Object Recognition, *International Journal of Computer Vision*, vol. 61, pp. 55-79, 2005.
21. P. Indyk and N. Thaper. Fast Image Retrieval via Embeddings. In *Proceedings of the 3rd International Workshop on Statistical and Computational Theories of Vision*, 2003.
22. K. Järvelin and J. Kekäläinen. Cumulated Gain-based Evaluation of IR Techniques. *ACM Transactions on Information Systems (TOIS)*, vol. 20, pp. 422-446, 2002.
23. A. E. Johnson and M. Hebert. Using Spin-images for efficient multiple model recognition in cluttered 3-D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 433-449, 1999.
24. H.-K. Kim and J.-D. Kim. Region-Based Shape Descriptor Invariant to Rotation, Scale and Translation, *Signal Processing: Image Communication*, vol. 16, pp. 87-93, 2000.
25. H. Ling and K. Okada. An efficient Earth Mover's distance algorithm for robust histogram comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 840-853, 2007.
26. H. Ling and K. Okada. Diffusion Distance for Histogram Comparison. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006.
27. Y. Liu, H. Zha and H. Qin. Shape topics: A Compact Representation and Algorithms for 3D Partial Shape Retrieval. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006.
28. Y. Liu, H. Zha, and H. Qin. The Generalized Shape Distributions for Shape Matching and Analysis. In *Proceedings of International Conference on Shape Modeling and Applications*, 2006.
29. Y. Liu, X.-L. Wang, and H. Zha. Dimension Amnesic Pyramid Match Kernel. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, 2008.
30. D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, vol. 60, pp. 91-110, 2004.
31. S. Marini, L. Paraboschi, S. Biasotti. SHape REtrieval Contest 2007 (SHREC07): Partial Matching Track. In R. C. Veltkamp, F. B. ter Haar (eds.). *SHREC2007: 3D Shape Retrieval Contest*, Technical Report UU-CS-2007-015, 2007.
32. K. Mikolajczyk and C. Schmid. Indexing Based on Scale Invariant Interest Points. In *Proceedings of International Conference on Computer Vision*, 2001.
33. D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2006.
34. R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics*, vol. 21, pp. 807-832, 2002.
35. G. Rote. Computing the minimum Hausdorff distance between two point sets on a line under translation. *Information Processing Letters*, vol. 38, pp. 123-127, 1991.
36. Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, vol. 40, pp. 99-121, 2000.
37. R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, vol. 37, pp. 297-336, 1999.
38. Y. Shan, H. S. Sawhney, B. Matei, R. Kumar. Shapeme Histogram Projection and Matching for Partial Object Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 568-577, 2006.
39. P. Shilane and T. A. Funkhouser. Distinctive regions of 3D surfaces. *ACM Transactions on Graphics*, vol. 26, 2007.
40. P. Shilane, P. Min, M. Kazhdan and T. Funkhouser. The Princeton shape benchmark. In *Proceedings of International Conference on Shape Modeling and Applications*, 2004.
41. J. Sivic, and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Proceedings of International Conference on Computer Vision*, 2003.
42. J. W. H. Tangelder, R. C. Veltkamp. A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications*, vol. 39, pp. 441-471, 2008.
43. J. W. H. Tangelder and R. C. Veltkamp. Polyhedral Model Retrieval Using Weighted Point Sets. *International Journal of Image and Graphics*, vol. 3, pp. 209-229, 2003.
44. D. V. Vranic. An improvement of rotation invariant 3D-shape based on functions on concentric spheres. In *Proceedings of International Conference on Image Processing*, 2003.
45. X.-L. Wang, Y. Liu and H. Zha. Learning Robust Cross-bin Similarities for the Bag-of-Features Model. Technical Report, available at <http://www.cis.pku.edu.cn/vision/Visual&Robot/people/wang/pubs/tr2009.draft.pdf>, 2009.
46. J. Weston, B. Scholkopf, E. Eskin, C. Leslie, and W. Noble. Dealing with large diagonals in kernel matrices. In *Principles of Data Mining and Knowledge Discovery*, vol. 243 of *SLNCS*, 2002.
47. J. Winn, A. Criminisi, and T. Minka. Object Categorization by Learned Universal Visual Dictionary. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2005.
48. J. Zhang, M. Marszalek, S. Lazebnik and C. Schmid. Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *International Journal of Computer Vision*, vol. 73, pp. 213-238, 2007.