Ryutarou Ohbuchi, Jun Kobayashi, Unsupervised Learning from a Corpus for Shape-Based 3D Model Retrieval, accepted, *ACM MIR 2006*, Santa Barbara, CA, U.S.A., Oct. 2006

Unsupervised Learning from a Corpus for Shape-Based 3D Model Retrieval

Ryutarou Ohbuchi
University of Yamanashi
4-3-11 Kofu-shi, Yamanashi-ken
400-8511 Japan
+55-220-8570
ohbuchi at yamanashi dot ac dot jp

Jun Kobayashi
University of Yamanashi
4-3-11 Kofu-shi, Yamanashi-ken
400-8511 Japan
+55-220-8570
g05mk014 at ccn dot yamanashi dot ac dot jp

ABSTRACT
Arguably the most important issues in shape-based 3D model retrieval are methods to extract powerful yet compact shape features and methods to properly and promptly compare the shape features. In this paper, we explore a method to improve feature distance computation by employing unsupervised learning of the subspace of 3D shape features from a corpus. We employ an algorithm called *Laplacian Eigenmaps* proposed by *Belkin*, et al. to learn a manifold spanned by shape features of 3D models in the corpus. The learned manifold is approximated by an RBF network, onto which shape features are projected. Distances among shape features can then be computed effectively on the learned manifold. We combine this learning-based distance-computation method with a method to extract multiresolution shape features proposed by Ohbuchi, et al. Our experimental evaluation showed that the proposed method could significantly improve retrieval performance. Learning alone improved performance of two shape features we tried by about 5%. A combination of learning and multiresolution shape feature allowed about 10% gain in performance. As an example, the trained, multiresolution version of the SPRH gained 10% over the original single resolution, untrained SPRH shape feature.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Information filtering. I.3.5 [**Computational Geometry and Object Modeling**]: Surface based 3D shape models. I.4.8 [**Scene Analysis**]: Object recognition. I.2.6 [**Learning**]: Induction.

General Terms
Algorithms, Performance, Experimentation, Measurement.

Keywords
Shape-based 3D model retrieval, content-based retrieval, manifold learning.

1. INTRODUCTION
An increased popularity of 3D models, from games on cellular phones to CAD models for automobile parts, has prompted research into effective reuse of 3D models through shape-based retrieval of 3D models [25, 13]. While a number of shape comparison methods have been proposed, the retrieval performance has not been satisfactory. There are many possible reasons for this. For one, the shape feature may be inadequate; the feature might not capture shape features that are relevant. Or, it could be the inadequacy of similarity, or dissimilarity (distance) computation method. Even if the feature contains salient information, the shape dissimilarity measure may not be using it effectively. Added factors are both inter-personal and intra-personal variability of shape similarity criteria. My shape similarity judgment may be different from yours. Or, my judgment today may be different from that of yesterday.

In this paper, we explore a method to improve distance computation among shape feature vectors. Given a set of *n*-dimensional feature vectors (points in $\mathbb{R}^n$), we assume that the feature vectors spans a *m*-dimensional ($m \le n$) non-linear subspace of the $\mathbb{R}^n$. We employ unsupervised learning to estimate the non-linear subspace. Specifically, we use the *Laplacian Eigenmaps* proposed by Belkin, et al [3] for the learning. Once such a subspace could be learned, *n*-dimensional feature vectors of 3D models could be projected onto the subspace so that a distance among a pair of feature vectors can be computed as a geodesic on the subspace. Our experimental evaluation showed that learning from a corpus of 3D shape models does improve retrieval performance. Retrieval performances of both AAD [19] and SPRH [28] shape features, measured in R-precision, increased by more than 5% by learning from a 4,000 model 3D model corpus. A combination of multiresolution shape feature extraction approach by Ohbuchi et al [18] and the learning produced about 10% increase in performance compared to the original (single-resolution) shape features. As an example, the trained, multiresolution (MR) version of the SPRH surpassed the performance of the *Light Field Descriptor* by Chen et al. [7].

This paper is organized as follows. In the following section, we will review learning applied to shape-based retrieval of 3D models. We will describe, in Section 3, our retrieval algorithm based on manifold learning and multiresolution shape features.

Experiments and their results are described in Section 4, followed, in Section 5, by conclusion and future work.

2. PREVIOUS WORK

Learning based approach to similarity retrieval can be classified into on-line learning and off-line learning. The on-line learning approach tries to learn human intentions interactively, e.g., through iterative relevance feedback or by interactive grouping of examples. An advantage of this approach is its capability to adapt to personal preference or even to changes in personal preference over time or occasion. The off-line learning approach learns from a prescribed training database prior to actual retrieval. The learning may be unsupervised to learn the structure of subspace on which the measured features exist. Or, the learning may be supervised, e.g., by using a pre-categorized database.

Relatively small number of work exploiting learning has so far been published for shape-based 3D model retrieval. Interactive relevance feedback, a form of on-line interactive learning, has been explored by several researchers for 3D model retreival [9, 1, 14, 16]. Elad et al. is among the first to apply Support Vector Machines (SVM) learning in an on-line learning setting to improve 3D model retrieval [8]. Leifman et al. [14] performed Kernel Principal Component Analysis (Kernel PCA) for an unsupervised learning of a feature subspace before applying a relevance feedback technique that employs Biased Discriminant Analysis (BDA) or Linear Discriminant Analysis (LDA) on the learned subspace. Novotni et al. [16] compared several learning methods, SVM, BDA, and Kernel-BDA, for their retrieval performance in a relevance feedback setting. Unlike relevance feedback, unsupervised off-line learning has seen very little attention in 3D model retrieval. The Kernel-PCA employed by Leifman et al. [14] is an example. The purity proposed by Bustos et al. [5] can also be considered as a weak form of unsupervised off-line learning. Purity is an estimate of the performance of a shape descriptor determined by using a pre-classified training database. Bustos used the purity to weight distance obtained from multiple shape descriptors.

Classical methods for unsupervised learning of subspace includes PCA and MDS, both of which are quite effective if the feature points lie on or near a linear subspace of the input space. However, if the subspace is non-linear, these methods do not work well. Many non-linear methods have been proposed for unsupervised learning of subspace; Self-Organizing Map (SOM) and Kernel-PCA are some of the well-known examples [10]. Recently, a class of geometrically inspired non-linear methods, called "manifold learning" has been proposed for learning the m-manifold of measured feature vectors. Some of the examples of manifold learning algorithms are *Isomap* [26], *Locally Linear Embedding* (LLE) [23], and *Laplacian Eigenmaps* (LE) [3]. Manifold learning has been applied to many problems, including motion analysis, multimedia data retrieval [11], and others. A property (drawback) of LE, LLE, and Isomap is that these mapping are defined only for the feature vectors in the training set. To query a 3D model outside of the training set, however, its feature vector must have an image on the manifold. In a 2D image retrieval setting, He et al [11] solved this problem by using Radial Basis Function (RBF) network [6, 10] for a continuous approximation of the manifold. The work presented in this paper is essentially He's method applied to shape-based 3D model retrieval with some modifications.

3. METHOD

Basic approach is similar to the 2D image retrieval proposed by He, et al. in [11]. He's method learns the manifold of 2D image features in the training set by using the *Laplacian Eigenmaps* (LE) by Belkin, et al. [3] and approximate the manifold by using a *Radial Basis Function* (RBF) *network*. We combine this learning based framework for similarity comparison with a multiresolution approach to feature extraction for 3D models proposed by Ohbuchi, et al. [18], and add a database subsampling step to contain computational cost.

We first describe the method for a single resolution shape feature. We then extend the approach to deal with multiresolution shape features in Section 3.5. The proposed method (using a single resolution shape feature) can be divided into two phases; the *learning phase* and the *retrieval phase* (See Figure 1). Each phase consists of the following steps:

The learning phase:

(1) **Extract shape feature vector:** Extract n-dimensional feature vectors from the K models in the training database (i.e., corpus).
(2) **Select training samples:** To reduce computational costs, subsample the training set down to L ($L \le K$) features vectors.
(3) **Learn manifold:** Perform unsupervised learning of the m-manifold ($m \le n$) spanned by the n-dimensional training samples by using Laplacian Eigenmaps by Belkin, et al. [3].
(4) **Approximate the manifold:** Construct a continuous approximation $\tilde{g}$ of the manifold by using the RBF network [5].
(5) **Project features of the models in the database:** Project features of all the models in the database onto the m-manifold using the approximation, and store the results together with the corresponding 3D models.

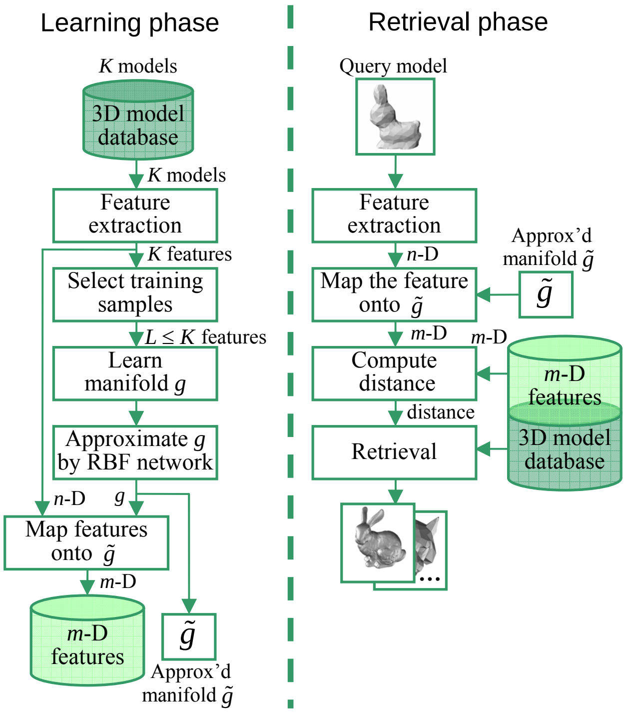
The retrieval phase:

(1) **Extract shape feature vector:** Extract an n-dimensional feature vector from the query model.
(2) **Map the feature onto the manifold:** Map the n-dimensional feature vector onto the approximated m-manifold $\tilde{g}$.
(3) **Compute distance on the manifold:** Compute distances from the query model to all the models in the database on $\tilde{g}$.
(4) **Retrieve and present the top *p* matches:** Retrieve the models in the database having the p-smallest distances from the query model.

3.1 Shape feature

For the experiments, we chose the *AAD* by Ohbuchi et al. [19] and the *SPRH* by Wahl, et al. [28] as the basic shape features for the experiments described in this paper. We call these single-resolution (SR) versions of the feature *SR-AAD* and *SR-SPRH*, respectively. (Multiresolution shape features and their integration with the learning framework will be described in Section 3.4.)

Both AAD and SPRH are global shape features that are inherently invariant to rigid body transformation. They are extensions of the D2 by Osada et al. [21]. The D2 is a 1D histogram of a distance computed between a pair of points generated at random locations on the surface of the 3D models to be compared. The AAD is a 2D joint histogram of a distance and an angle formed by the normal orientations of a pair of points, while the SPRH is a 4D joint histogram of three angles and a distance. Any shape feature can be used in the proposed learning framework provided that the feature produce feature *vector*. Some shape comparison methods do not produce feature vectors, and thus the proposed learning framework can't be applied. For example, the LFD [6], which involves combinatorial method to find a distance among a pair of models, can't be used with the proposed learning approach.



g : Mapping from n-D feature vector to m-D feature vector, defined only at samples learned.

$\tilde{g}$: Approximate, continuous mapping from n-D feature vector to m-D feature vector.

Figure 1. The learning phase (left) and the retrieval phase (right) of the proposed method.

3.2 Training sample selection

Learning a manifold by using LE could be expensive both in time and in space. A naïve method for the eigendecomposition used for the LE costs $O(n^2)$ space and $O(n^3)$ time. The RBF network regression also costs significant amount of spatial and temporal computational costs. To reduce these costs, we tried to estimate an approximate manifold by using a sampled subset of the corpus. We tried several subsampling methods, namely: (1) Pseudo-random sequence in C++ library, (2) *Sobol*'s quasi-random sequence [22], (3) *Niederreiter*'s quasi-random sequence [4], and (4) Max-min approach proposed by Liu, et al. [15]. In *Monte-Carlo* methods, quasi-random sequences are known to reduce variance. The max-min approach by Liu, et al try to quickly approximate the "outermost shape" of the feature point distribution by concentrating samples at points that are well spaced yet farthest from each other.

3.3 Manifold learning and approximation

Belkin's LE performs estimation of an m-manifold by following the steps below;

(1) **Construct an adjacency graph:** A feature vector of a model in the training database is a point in the n-dimensional feature space $U = \mathbb{R}^n$. Construct an mesh G by connecting k-nearest points using Euclidian distance in U.

(2) **Create mesh Laplacian matrix for G:** Create a *mesh Laplacian* matrix $L = D - W$, in which W is an adjacency matrix for G,

$$w_{ij} = \begin{cases} 1, & \text{if vertices } i \text{ and } j \text{ are adjacent;} \\ 0, & \text{otherwise.} \end{cases}$$

and D is the diagonal matrix satisfying the following equation.

$$D_{ii} = \sum_j w_{ji}$$

(3) **Perform an Eigenanalysis of L**: Find eigenvalues λ_i $(1 \le i \le n)$ and eigenvectors $\mathbf{f}_i$ $(1 \le i \le n)$ of L by solving the generalized eigenproblem $L\mathbf{f} = \lambda D\mathbf{f}$.

(4) **Find a manifold *g***: Sort eigenvectors in an ascending order. The least m eigenvectors (but excluding the first eigenvector $\mathbf{f}_0$) defines mapping $\varphi : \mathbf{x}_i \to (\mathbf{f}_1(i), \ldots, \mathbf{f}_m(i))$ that maps a point $\mathbf{x}_i \in U = \mathbb{R}^n$ onto m-manifold g.

Note that g is defined only for the points in the training set. To project a feature vector (e.g, a query) not in the training samples onto g, He et al. [11] approximated g by using RBF network. We used an implementation of Chen's RBF algorithm [6] included in the *Neural Network Toolkit* of the *MatLab* for the approximation.

3.4 Distance computation

We used *Cosine* distance for distance computations, for both learned and not-learned versions of the AAD and SPRH shape features. Cosine distance is computed simply as the inner product of a pair of normalized feature vectors. In addition, for comparison, we used *Kullback-Leibler divergence* (KLD) [28] to compute distance for the SPRH shape feature without learning.

Cosine distance was selected from four distance measures, L1, L2, Cosine, and KLD, after experimental evaluation of their performances.

In the case of AAD, our experiment showed that the Cosine distance and KLD performed identically, outperforming both L1 norm and L2 norm. (The original paper [19] uses L2 norm.)

In the case of SPRH, the Wahl et al. used KLD [28], and that the KLD outperformed three other distance measures we experimented. (See Table 1 for the comparison among KLD and Cosine distance.) However, KLD can't be used if a feature vector contains negative values, which is the case for learned feature vectors. We thus used the second best, the Cosine distance, for the SPRH in all the experiments that follows. For comparison, we included the results for SPRH using KLD for the cases without learning.

Table 1. Distance measure and performance of the SPRH.

Feature	Distance measure	RP	11A
SR-SPRH	KLD	37.4	40.1
	Cosine	35.6	38.4

3.5 Exploiting multiresolution shape features

The multiresolution (MR) feature extraction method proposed by Ohbuchi et al. [18, 20] is known to improve performance of some of the shape comparison methods. The multiresolution approach may be able to extract more shape feature from a model. Also, the multiresolution approach may conform better to the human perception and/or cognition in comparing 3D shapes. The approach by Ohbuchi et al. first creates a set of 3D MR shape features. Once a set of MR models is obtained, appropriate (single resolution) shape feature is computed at each of the multiple resolution levels to produce a set of multi-resolution shape feature vectors. An advantage of this multiresolution (MR) representation is that it can be computed for polygon soup models or even for point set models (by skipping the step 1 below.) Briefly, the method computes a set of multiresolution shape feature as below (see Figure 2);

(1) **Compute a multiresolution representation**: The surface-based input model is converted into a point-based model by *Monte-Carlo* sampling of the surfaces of the model. A set of L-1 scale values α_i, i=1…L-1 is computed based on the size of the model. The set of scale values are used to normalize size among shape models. Then, compute L-1 3D alpha shapes from the point set model by using the L-1 scale values α_i. Of L resolution levels, coarser (L-1) levels are computed by using the 3D alpha shapes [8]. The finest resolution level L uses the original polygon soup model.

(2) **Compute multiresolution shape feature set**: Apply a shape feature extraction method x to a model at each resolution level of the MR, creating a set of multiresolution shape features MR-x.

Figure 3 shows an example of the MR representation for a surface-based 3D model of a biplane.

Given the MR representation, learning based approach described above can be applied in two different ways.

(1) **Learning features at each resolution level:** In the learning phase, learn a manifold at each of the L-level MR representation. In the retrieval phase, a distance is calculated at each of the L levels of MR representation. Then L distance values are combined into a single overall distance. Combination can be performed, for example, by using fixed-weight linear-combination or by using *purity* [5].

(2) **Learning features as a multiresolution feature set:** Concatenate L feature vectors from the L resolution levels into a big feature vector, and learn a manifold from the concatenated vector. Distance among a pair of models is computed using the learned manifold.

In the experiments described in this paper, we chose to apply the former approach. Principal reason for this choice is the high computational and spatial cost of the latter approach.

Distances of a pair of MR-shape features are combined using a fixed weight linear combination. In the experiments described in this paper, the weights are fixed at 1.0 across the resolution levels.

In the following, a single resolution shape features are prefixed with "SR", and a multiresolution shape features are prefixed with "MR". Also, learned shape feature are prefixed with "L". In addition, postfix "C" or "K" may accompany these acronyms indicating distance computation methods; "C" for Cosine distance or "K" for KLD. Thus, for example, "SR-SPRH-K" stands for the "single-resolution SPRH using KLD", while "L-MR-AAD-C" stands for the "learned multi-resolution AAD using Cosine distance".

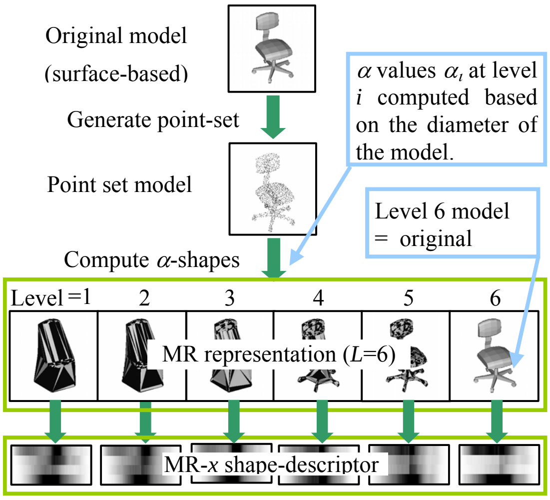


Figure 2. Computing a MR-x multiresolution shape feature set. (This example shows 2D histogram of the AAD [19] shape feature x.)

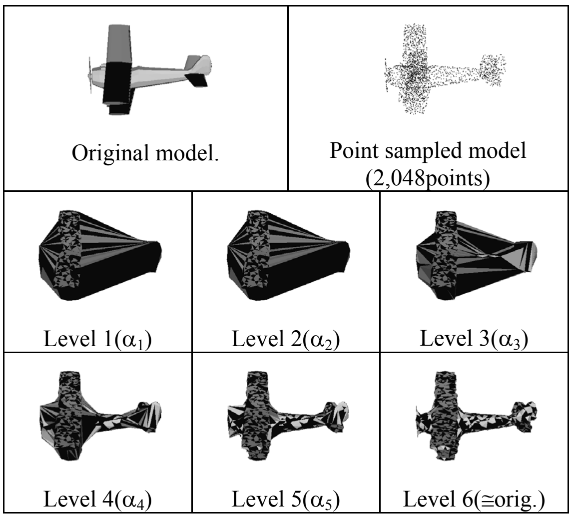


Figure 3. An example of MR representation for the surface-based model of a biplane. A feature per level is computed for the set for shape comparison.

4. EXPERIMENTS AND RESULTS

We implemented the learning algorithm in *Matlab* (Release 12.1). Others, such as, feature extraction, distance computation, retrieval and results evaluation are implemented in C++.

We use 3D model databases for two purposes; (1) unsupervised learning, and (2) performance evaluation. For learning, we used the union of the training set of the *Princeton Shape Benchmark* (PSB) [24] database containing 907 models and the *National Taiwan University 3D Model Database* (NTU) ver. 1 [17] containing 10,911 models. As our method uses unlabeled samples for learning, we ignored the category attached to the PSB training set. (The NTU database does not have any category or labeling.) We sampled the union, containing 11,818 models by using one of the four sampling strategies described in Section 3.2. For evaluation, we used the test set of the PSB that contains 907 models classified into 92 "base" categories. We used this categorization as the ground truth for the following experiments. For the performance evaluation, we picked a model from the test database of size *K* as the query, and ranked all the other (*K*-1) models in an ascending order of the distance computed.

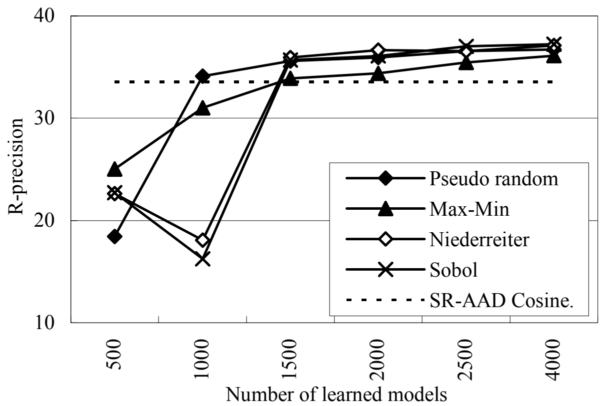
As quantitative measures of performance, we used *R-precision* (RP) and *11 point average precision* (11P) figures and the precision-recall plot [2]. The R-precision is the ratio, in percentile, of the models retrieved from the desired class C_k (i.e., the same class as the query) in the top *R* retrievals, in which *R* is the size of the class $|C_k|$. In computing the R-precision, we did not count the query q among the retrieved model, i.e., the numerator, which is divided by $|C_k - 1|$. (Note that RP is similar but different from the First Tier used in the SHREC 2006 benchmark [27].) The 11-point average 11A is the average of precision values taken at 11 equally spaced recall values {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}. An 11P average precision value can be considered as a summary of a recall-precision plot, which emphasizes overall performance. The RP and 11P values presented below are a mean over all the *K*=907 queries. As a reference of performance, we included the performance figure obtained by using *Light Field Descriptor* (LFD) by Chen, et al. [7] in some of the results.

Several parameters of the LE and the RBF need to be fixed. We performed some preliminary experiments to determine the number of neighborhood *k* for the mesh construction and the spread σ of the RBF (a Gaussian function) for the RBF network approximation. For the SR-AAD and MR-AAD, we chose $k = 12$ and $\sigma = 0.6$, after experiments that varied *k* in the range [8, 20] and σ in the range [0.4, 1.0]. For the SR-SPRH and MR-SPRH, we chose $k = 40$ and $\sigma = 1.2$, after experiments that varied *k* in the range [12, 50] and σ in the range [0.2, 1.4].

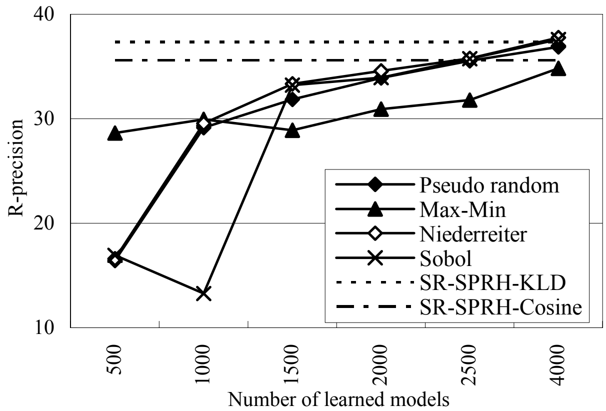
4.1 Sub-sampling methods and performance

We compared four sub-sampling schemes for their retrieval performance by varying the number of learned samples. The experiment was done by using the SR-AAD and SR-SPRH shape features.

Figure 4 shows the results. While the Max-Min farthest point sampling scheme [15] performs the best at a fewer number of samples (e.g., 500~1000 samples), other sampling schemes produce better results with a larger number of samples (>1500 samples). At the largest number of training samples tested, the *Niederreiter*'s sequence and the *Sobol*'s sequence performed the best, closely followed by the pseudo-random number sequence. The performance difference between the *Niederreiter*'s sequence and the *Sobol*'s sequence was negligible. Note that we used the default parameters for the *Niederreiter*'s sequence [4], whose results might have been different if we were to use a different set of parameters. In the experiments that follow, we will use the *Niederreiter's* sequence for subsampling unless otherwise stated.



(a) R-precision (RP) values for the SR-AAD feature by using four subsampling schemes. Dotted line shows the RP value for the original SR-AAD feature.



(b) R-precision (RP) values for the SR-SPRH shape feature obtained by using four subsampling schemes. Dotted line shows the RP value for the original SPRH shape feature (i.e., SR-SPRH-KLD), and the broken line shows the RP value for the SR-SPRH using Cosine distance.

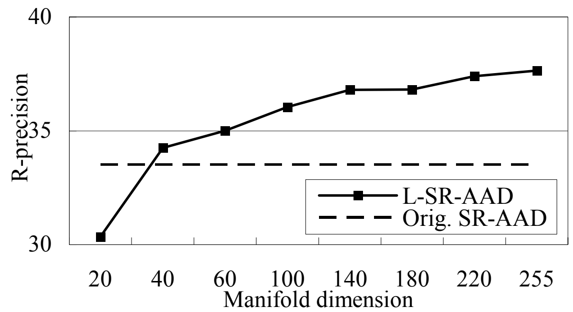
Figure 4. Database subsampling schemes and performance. For AAD (a) and for SPRH (b), plots of R-precision at different number of training samples *L* are shown for four subsampling schemes.

4.2 Manifold dimension vs. retrieval performance

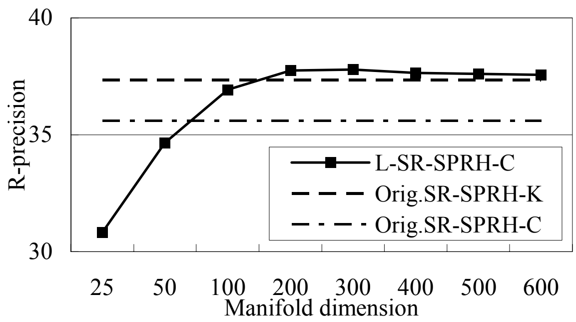
We performed experiments to determine the relationship between the dimension *m* of the manifold and the retrieval performance. The original SR-AAD feature has 256 dimensions [19] and the original SR-SPRH has 625 dimensions [28].

Figure 5 shows, for SR-AAD and SR-SPRH, changes in retrieval performance due to the number of dimension *m* of the manifold. In the case of the AAD, the retrieval performance of the trained version (L-SR-AAD-C) increased with the number of dimensions. The performance of the trained version (solid line with markers) surpasses that of the original at about *m*=40, a dimension significantly smaller than the original 256. In other words, for SR-AAD-C, *m*=40 or 1/5 of the size of original vector, is sufficient to achieve the performance level equal or better than the original. In the case of the SPRH, the performance of the learned version L-SR-SPRH-C (using Cosine distance) topped out at about 200 dimensions. Compared to the untrained original using KLD (SR-SPRH-K), the trained version is only marginally better at about *m*>200. If compared to the untrained SPRH using Cosine distance (SR-SPRH-C), however, the trained version performed significantly better. Interestingly, as we will discuss in Section 4.3, L-MR-SPRH-C, the learned MR-version of the SPRH using Cosine distance behaves differently; it somehow achieves significant performance gain of about 5% over the untrained MR-SPRH-K using *KLD*.

In terms of dimension reduction, the method could not find clear low-dimensional manifold, in both SR-AAD and SR-SPRH features. This is not very surprising, as 3D shapes most likely won't have a clear low-dimensional manifold of fixed dimension. (As opposed to, for example, human animation data constrained physically by skeletal structure, etc.)



(a) Trained L-SR-AAD feature (solid line) and the original AAD without learning (broken line).



(b) Trained L-SR-SPRH-C feature using Cosine distance (solid line) and the original SPRH using both KLD (SR-SPRH-K) and Cosine distance (SR-SPRH-C).

Figure 5. Dimensionality of feature vectors after dimension reduction vs. retrieval performance measured in RP figure [%].

Based on the experiment, dimensions of manifold *m* for each shape feature to be used for the experiments in Section 4.3 were picked. When in doubt, we tried err on the side of having more dimension, disregarding computational costs. For L-SR-AAD and for each resolution level of L-MR-AAD, *m*=220 (< *n*=256) is used. For L-SR-SPRH and for each resolution level of L-MR-SPRH (at each resolution level), *m*=500 (< *n*=625) is used. (While the assessment based only on Figure 4(b) suggests the dimension of about 200 for the SR-SPRH, other figures such as 11Pwere taken into consideration.)

4.3 Number of samples vs. retrieval performance

Figure 6 shows the change in retrieval performance due to learning for the SR-AAD, MR-AAD, SR-SPRH, and MR-SPRH shape features. Table 2 shows a summary of performance evaluation results. Overall, the manifold learning increased the performance by about 5%, and the multiresolution method added another 5%, for an overall increase of about 10% for both AAD and SPRH. As a result of the combined improvement, the performance of the SPRH measured in RP increased from 37.4% to 47.7%, surpassing that of our benchmark *Light Field Descriptor* [7].

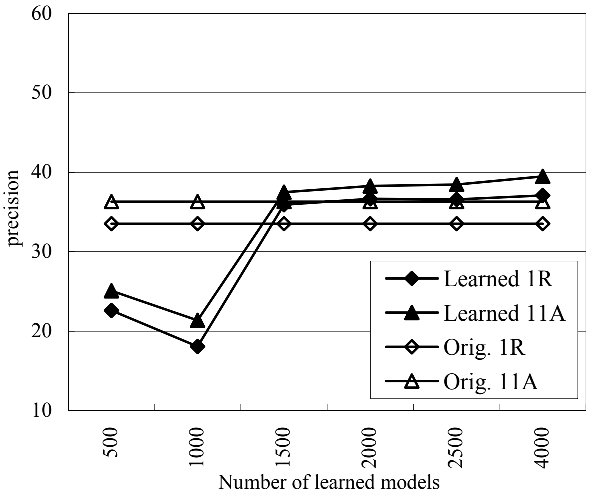
Notice in Figure 6 that the performances of every shape feature drop well below the original (i.e., untrained) versions. This can be explained as follows; the manifold estimated using a small number of samples are so warped that the distance computation suffered, compared even to the original space. As the number of samples increase, the learning captures the shape of the manifold enough to produce equal or better results. Please note that, considering the number of dimensions for the features, training samples size *L* is very small. Consider the SR-AAD feature having 256 dimensions. A set of 256 uniformly distributed samples means, on average, only one sample are taken per dimension.

In both Table 2 and Figure 6, the numbers of learned models are limited to either 4000 or 5000 in spite of the corpus size of 11,818. This limitation is due to the large memory consumption of the RBF network regression algorithm we used, and also to the memory limitation of the *MatLab* we used.

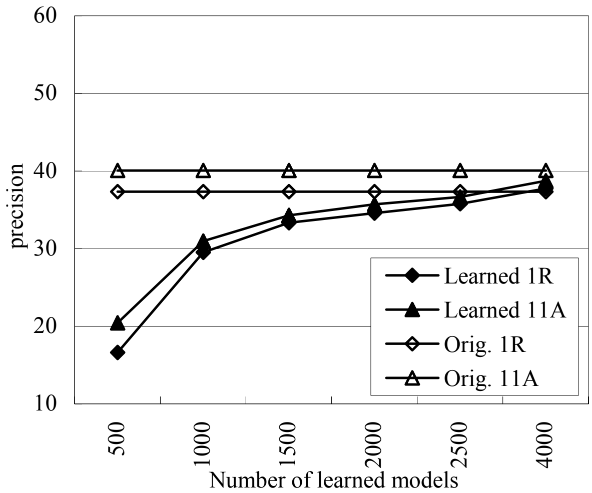
Table 2. A summary of retrieval performance showing the effect of learning..

Base feature	MR/ SR	Train?	Feature name	*L*	RP [%]	11P [%]
AAD	SR	No	SR-AAD	-	33.5	36.3
		Yes	L-SR-AAD	4000	37.1	39.5
	MR	No	MR-AAD	-	40.3	43.9
		Yes	L-MR-AAD	4000	43.9	46.8
SPRH	SR	No	SR-SPRH-K	-	37.4	40.1
		No	SR-SPRH-C	-	35.6	38.4
		Yes	L-SR-SPRH	5000	37.8	38.7
	MR	No	MR-SPRH	-	42.5	45.7
		Yes	L-MR-SPRH	5000	47.5	50.1
LFD	-	-	-	-	45.9	49.3

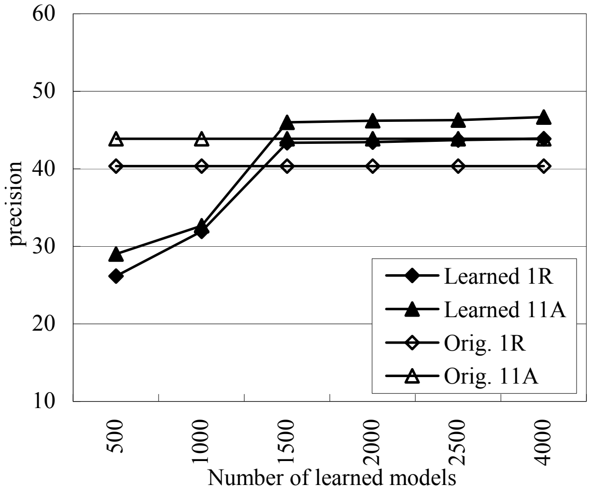
SR: Single Resolution MR: Multi Resolution
L: Number of training samples.



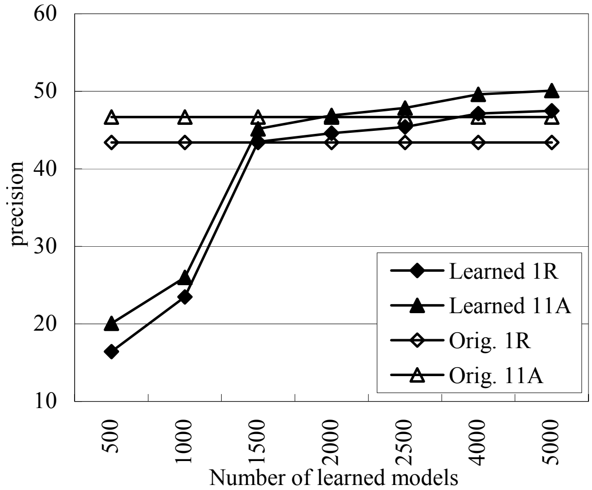
(a) Number of learned models and retrieval performance for the SR-AAD shape descriptor.



(b) Number of learned models and retrieval performance for the SR-SPRH shape descriptor.



(c) Number of learned models and the performance for the MR-AAD shape descriptor.



(d) Number of learned models and the performance for the MR-SPRH shape descriptor.

Figure 6. Size of the learned models L vs. retrieval performance for the four shape features. All the training sets are sampled by using *Niederreiter's* sequence.

4.4 Retrieval example

Figure 7 shows examples of retrieval by using the model of an office chair (#801) in the PSB test set as a query. The model belongs to the category that contains total of 20 models.

The interface displays the query model at the top left corner, in addition to the 20 retrieval results in the 4-rows by 5-columns matrix format. The top left position of the 4 by 5 matrix displays the best match, which is almost always the query itself. (Note that, some shape comparison method might return a model other than itself as the top match.) Thus, up to 19 "true" retrieval results can be displayed in this "first page". In the image, those in the correct category (office chair) are marked by solid rectangle, indicating "highly relevant", results. Those marked by broken line are "relevant" results, indicating models that are close to the models in the correct category.

It can be seen that both unsupervised learning (L-) and multiresolution shape feature extraction (MR-) produced performance gain. The combination of the two approaches (L-MR-xxx) produced the best results for both AAD and SPRH shape features. For the MR feature extraction, the untrained (multiresolution) MR-AAD (Figure 7(e)) retrieved 10 out of 20 office chair models in the top 19 retrievals, compared to 8 models for the untrained (single resolution) SR-AAD (Figure 7(a)). Unsupervised learning improved performance as well; those with learning (on the right hand side of the Figure 7) performed better than their counterparts without learning (on the left hand side of the Figure 7.). For example, trained L-MR-SPRH retrieved 12 out of 20 office chair models (Figure 7(h)), compared to only 8 office chair models for the untrained MR-SPRH (Figure 7(g)). Notice also that the relevant results ("near misses") increased significantly by learning across features, often displacing such irrelevant retrievals as trucks, houses, and computer monitors. For example, while MR-SPRH (Figure 7(g)) contained 12 relevant retrievals (R=12/20) in the top 19, L-MR-SPRH (Figure 7(h)) contained 18 relevant retrievals (R=18/20) in the top 19.

4.5 SHREC 2006 benchmark results

We did not enter the SHREC 2006 3D shape retrieval contest [27], but we performed the benchmark as specified in the contest using the method described in this paper. The SHREC 2006 uses the 1,814 models of both training and test sets of the PSB as the test database. The 30 query models are not included in the test database. We learned the manifold by sampling a union of the PSB (1,814 models) and the NTU (10,911 models) as described in Section 4.1 using the *Niederreiter's* quasi-random sequence. The sizes of training sets are 5,000 for AAD and 4,500 for the SPRH.

Excerpts of benchmark results are shown in Table 3. The table includes, for reference, the top two performers from the SHREC 2006 results. The performance of the learned, MR version of the SPRH would have placed at the second or third place overall if we were to enter the contest.

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed and evaluated a method for shape similarity based retrieval of 3D models that employs *unsupervised learning* of a 3D shape corpus (i.e., a collection of 3D shape models). Our motivation was to improve distance computation among shape features that are suspected to lie on a non-linear *m*-dimensional ($m \leq n$) subspace of the *n*-dimensional vector space spanned by the feature vectors. The proposed method learns the non-linear subspace, or manifold, spanned by the training samples, i.e., shape feature vectors of 3D model corpus, by using the *Laplacian Eigenmaps* [3]. The manifold is approximated by RBF network [6], onto which the feature vectors having high dimension are projected. The distance computation among shape features are then performed on the manifold. We combined this learning approach with the multiresolution shape feature extraction method proposed by Ohbuchi et al. [18].

The experimental evaluation showed that our proposed method significantly improved shape-based 3D model retrieval performance. For example, the learning improved retrieval performances of both AAD [19] and SPRH [28] shape features by about 5%, measured in R-precision. Combined with the multiresolution approach, both AAD and SPRH gained about 10% in R-precision, rivaling one of the best performing shape comparison method *Light Field Descriptor* by Chen, et al [7].

Probably the most significant issue with the proposed method is the computational cost, both spatial and temporal, of the learning phase. As stated before, a naïve method for the eigendecomposition used for the manifold estimation costs $O(n^2)$ space and $O(n^3)$ time. The RBF network regression, currently the limiting factor in increasing the training set size, also costs significant amount of spatial and temporal computational. Thus, one of the foremost issues is to reduce computational costs of eigendecomposition and RBF network approximation. Note that the computational cost of the retrieval phase is not particularly expensive. It only requires projection of the shape feature vector of the query model onto the manifold, followed by cosine distance computations.)

Our experiments on the relation of subspace dimension and retrieval performance found no clear dimension of the subspace. This is not very surprising, as shape features for 3D shapes won't span a well-defined low-dimensional subspace of a low dimension. It is unclear if some other non-linear unsupervised learning algorithms, such as *LLE* and *Isomap*, might be able to do better in discovering the subspace for improved retrieval performance. We definitely would like to experiment with these other learning algorithms.

ACKNOWLEDGMENTS

This research has been funded by the Ministry of Education, Culture, Sports, Sciences, and Technology of Japan (No. 17500066 and No. 18300068).

REFERENCES

[1] I. Atmosukarto, W.K. Leow, Z. Huang, Feature Combination and Relevance Feedback for 3D Model Retrieval, *Proc. MMM 2005*, pp. 334-339, (2005).

[2] R. Baeza-Yates, B. Ribiero-Neto, *Modern information retrieval*, Addison-Wesley (1999).

[3] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Computation* **15**, pp. 1373–1396, (2003).

[4] P. Bratley, B. L. Fox, H. Niederreiter, Algorithm 738: Programs to Generate Niederreiter's Low-discrepancy Sequences, ACM TOMS Algorithm 738.

[5] B. Bustos, D. Keim, D. Saupe, T. Schreck, D. Vranić, Automatic Selection and Combination of Descriptors for Effective 3D Similarity Search, *Proc. IEEE MCBAR'04*, pp. 514-521, (2004).

[6] Chen, S., C.F.N. Cowan, P. M. Grant, Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks, *IEEE Trans. on Neural Networks*, **2**(2), pp. 302-309, (1991)

[7] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, M. Ouhyoung, On Visual Similarity Based 3D Model Retrieval, *Computer Graphics Forum*, **22**(3), pp. 223-232, (2003).

[8] H. Edelsbrunner, E. P. Mücke, Three-dimensional Alpha Shapes, *ACM TOG*, **13**(1), pp. 43-72, (1994).

[9] M. Elad, A. Tal, S. Ar., Content based retrieval of vrml objects – an iterative and interactive approach, Proc. *EG Multimedia* **39**, pp. 97-108, (2001).

[10] S. Haykin, *Neural network a comprehensive foundation*, Second Edition, Prentice Hall, 842pages, (1999).

[11] Xiaofei He, Wei-Ying Ma, Hong-jiang Zhang: Learning an Image Manifold for Retrieval, *Proc. ACM Multimedia 2004*, pp. 17-23 (2004).

[12] Iyer, N., Kalyanaraman, Y., Lou, K., Jayanti, S., Ramani, K., A Reconfigurable, Intelligent 3D Engineering Shape Search System Part I: Shape Representation, *Proc. ASME DETC* '03, 23rd CIE Conf. (2003).

[13] M. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, K. Ramani, Three Dimensional Shape Searching: State-of-the-art Review

and Future Trends, *Computer Aided Design*, **5**(15), pp. 509-530, (2005).

[14] G. Leifman, R. Meir, A. Tal, Semantic-oriented 3d shape retrieval using relevance feedback, *The Visual Computer (Pacific Graphics),* **21**(8-10)*,* pp. 865-875*, October 2005.*

[15] Rong Liu, Varun Jain, Hao Zhang, Sub-sampling for Efficient Spectral Mesh Processing, *Proc. CGI 2006*, LNCS **4035**, pp. 172-184, Springer-Verlag, (2006).

[16] M. Novotni, G.-J. Park, R. Wessel, R. Klein Evaluation of Kernel Based Methods for Relevance Feedback in 3D Shape Retrieval, *Proc. The Fourth International Workshop on Content-Based Multimedia Indexing* (*CBMI'05*), (2005).

[17] NTU 3D Model Database ver.1 http://3d.csie.ntu.edu.tw/

[18] R. Ohbuchi, T. Takei, Shape-Similarity Comparison of 3D Shapes Using Alpha Shapes, *Proc. PG 2003*, pp. 293-302, (2003).

[19] Ryutarou Ohbuchi, Takahiro Minamitani, Tsuyoshi Takei, Shape-similarity search of 3D models by using enhanced shape functions, *International Journal of Computer Applications in Technology* (*IJCAT*), **23**(3/4/5), pp. 70-85, (2005).

[20] Ryutarou Ohbuchi, Yushin Hata, Combining Multiresolution Shape Descriptors for 3D Model Retrieval, *Proc WSCG 2006* (2006).

[21] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, Shape Distributions, *ACM TOG*, **21**(4), pp. 807-832, (2002).

[22] W. H. Press et al., *Numerical Recipes in C-The Art of Scientific Programming*, *2nd Ed*., Cambridge University Press, Cambridge, UK, 1992.

[23] S.T. Roweis, L.K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, Science, **290**(5500), pp. 2323-2326, (2000).

[24] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, The Princeton Shape Benchmark, *Proc. SMI '04*, pp. 167-178, (2004). http://shape.cs.princeton.edu/search.html

[25] J. Tangelder, R. C. Veltkamp, A Survey of Content Based 3D Shape Retrieval Methods, *Proc. SMI '04*, pp. 145-156.

[26] J. B. Tanenbaum, V. de Silva, J.C. Langford, A Global Geometric Framework for Nonlinaer Dimensionality Reduction, Science, **290**(5500), pp. 2319-2323, (2000).

[27] R. C. Veltkamp, R. Ruijsenaars, Michela Spagnuolo, R. Van Zwol, F. ter Haar, SHREC2006 3D Shape Retrieval Contest, Utrecht University Dept. Information and Computing Sciences *Technical Report UU-CS-2006-030* (ISSN: 0924-3275) http://give-lab.cs.uu.nl/shrec/shrec2006/index.html

[28] E. Wahl, U. Hillenbrand, G. Hirzinger, Surflet-Pair-Relation Histograms: A Statistical 3D-Shape Representation for Rapid Classification, *Proc. 3DIM 2003*, pp. 474-481, (2003).

Table 3. An excerpt of performance figures obtained by using the SHREC 2006 benchmark [27].

Train?	SR/ MR	Feature name	AP-HR	AP-R	FT-HR [%]	FT-R [%]	DAR	NCG @25	NDCG @25
No	SR	SR-SR-AAD-C	0.2374	0.2603	24.55	26.48	0.3364	0.3405	0.3771
		SR-SPRH-K	0.2886	0.3179	26.68	31.77	0.3990	0.3920	0.4384
		SR-SPRH-C	0.2656	0.2892	25.03	28.48	0.3650	0.3577	0.4038
	MR	MR-AAD-C	0.3322	0.3246	31.02	31.38	0.4320	0.4386	0.4793
		MR-SPRH-K	0.3761	0.3552	34.93	32.84	0.4631	0.4519	0.5101
		MR-SPRH-C	0.3585	0.3416	33.42	33.38	0.4344	0.4362	0.4856
Yes	SR	L-SR-AAD-C	0.2748	0.2795	29.81	28.31	0.3669	0.3962	0.4285
		L-SR-SPRH-C	0.2979	0.2837	30.85	30.31	0.3884	0.4117	0.4428
	MR	L-MR-AAD-C	0.3789	0.3564	36.58	34.35	0.4687	0.4720	0.5194
		L-MR-SPRH-C	0.4539	0.4105	43.12	39.98	0.5276	0.5379	0.5871
SHREC 2006	Makadia, (run 2)		0.4869	0.4364	44.77	40.55	0.5499	0.5498	0.5906
	Dars, (run 1)		0.4475	0.3952	42.75	37.03	0.5242	0.5246	0.5791

AP-HR: Mean Average Precision (highly relevant)
AP-HR: Mean Average Precision (relevant)
FT_HR: Mean First Tier (Highly Relevant)
FT_R: Mean First Tier (Relevant)
DAR: Mean Dynamic Average Recall
NCG @25: Mean Normalized Cumulated Gain @25
NDCG @25: Mean Normlized Discounted Cumulated Gain @25

* Prefix "L-" indicates "learned" version of the shape feature.

* AAD-C was trained by using 5,000 samples, and SPRH-C was trained by using 4,500 samples.

=

(a) SR-AAD, HR=8/20, R=11/20.

(b) L-SR-AAD. HR=10/20, R=15/20.

(c) SR-SPRH, HR=8/20, R=12/20.

(d) L-SR-SPRH, HR=9/20, R=13/20.

(e) MR-AAD, HR=10/20, R=15/20.

(f) L-MR-AAD, HR=14/20, R=19/20.

(g) MR-SPRH, HR=8/20, R=12/20.

(h) L-MR-SPRH, HR=12/20, R=18/20.

Figure 7. Examples of query using an office chair model. Learning clearly improved the retrieval results in this example