

Ryutarou Ohbuchi, Jun Kobayashi, Akihiro Yamamoto, and Toshiya Shimizu,
Comparison of dimension reduction method for database-adaptive 3D model retrieval,
In *Proc. Fifth International Workshop on Adaptive Multimedia Retrieval (AMR 2007)*, Paris, France, July 2007.

Comparison of dimension reduction methods for database-adaptive 3D model retrieval

Ryutarou Ohbuchi, Jun Kobayashi, Akihiro Yamamoto, Toshiya Shimizu

4-3-11 Takeda, Kofu-shi, Yamanashi-ken, 400-8511, Japan.
ohbuchi AT yamanashi.ac.jp, jun066 AT quartz.ocn.ne.jp,
g05mk039 AT yamanashi.ac.jp, g06mk009 AT Yamanashi.ac.jp

Abstract. Distance measures, along with shape features, are the most critical components in a shape-based 3D model retrieval system. Given a shape feature, an optimal distance measure will vary per query, per user, or per database. No single, fixed distance measure would be satisfactory all the time. This paper focuses on a method to adapt distance measure to the database to be queried by using learning-based dimension reduction algorithms. We experimentally compare six such dimension reduction algorithms, both linear and non-linear, for their efficacy in the context of shape-based 3D model retrieval. We tested the efficacy of these methods by applying them to five global shape features. Among the dimension reduction methods we tested, non-linear manifold learning algorithms performed better than the other, e.g. linear algorithms such as principal component analysis. Performance of the best performing combination is roughly the same as the top finisher in the SHREC 2006 contest.

1. Introduction

Research on shape-based retrieval of 3D models [24, 13, 26] has recently gained attention. A shape-based 3D model retrieval system retrieves, given a query, a set of shape models ranked by their shape-based similarity to the query. The query may be texts, 2D sketches, 3D sketches, or 3D models. In this paper, we assume the query is a 3D model defined, for example, as a set of polygons and that the system retrieves 3D models similar in their shape to the query.

Two of the most significant technical challenges for shape-based retrieval of 3D models are feature extraction and distance computation. We first have to devise a compact yet expressive shape feature that can be extracted and compared with reasonable computational cost. In feature extraction, compatibility of the shape feature with various shape representations, e.g., polygon soup and voxel enumeration solid, is an important factor to be considered. Then, distance, or dissimilarity among a pair of models to be compared must be computed. It is desirable that the feature and distance measure be adaptive to a database, to a user, or even to a specific query. For example, a combination of feature and distance metric that works well for comparing human face models may be sub-optimal for comparing screws. Or, the bunny model I wanted yesterday may be different from a bunny model I want today. So-called “curse

of dimensionality”, which discourages higher dimensional shape feature vectors, also comes into play as the dimensionality of 3D shape feature tend to be quite high.

In our previous work [20], we adopted Xhaofei He’s approach [12], producing significant improvement in retrieval performance. Instead of the distance in the original feature (or input) space, the method uses geodesic distance on a *subspace*, or a *manifold*, spanned by the features of 3D models in a database. He’s method used the *Laplacian Eigenmaps* [3] for learning a manifold from the set of features for learning-based dimension reduction.

Our previous work [20] has several limitations, however. The experimental results are derived using only one non-linear dimension reduction algorithm, LE. And the LE is combined with only two shape features, the AAD [19] and SPRH [28]. Also, the training set size in our previous work [20] was limited to about 4000 samples, mostly due to memory space limitation.

In this paper, we try to experimentally explore the approach further in the following three aspects;

1. How do various subspace learning methods compare?
2. How well does the approach work when it is applied to various shape features?
3. What is the impact of the number of learning samples?

To answer the first question, we compared five dimension reduction methods, all of which are based on unsupervised subspace learning. The set of dimension reduction methods we compared are the *Principal Component Analysis (PCA)*, *Kernel PCA (KPCA)* [10], *Locality Preserving Projections (LPP)* [11], *Laplacian Eigenmaps (LE)* [3], *Locally Linear Embedding (LLE)* [21], and *Isometric feature mapping (Isomap)* [25]. To answer the second question, we experimented with five shape features and their respective multiresolution variations [18]. The five features we compared are; the *Ray-based Spherical Harmonics (RSH)* [27], *Exponentially-decaying EDT (EDED)* [27], *Spherical Harmonics (SH)* [14], *Absolute Angle-Distance histogram (AAD)* [19], and the *Surflet-Pair Relation Histogram (SPRH)* [28]. We also applied the multiresolution shape feature extraction approach by Ohbuchi et al [18] on these shape features; all in all, we experimented with 16 different shape features. To answer the third question, we increased the number of training samples from 4,000~6,000 to 10,000, by using CPU, OS, and application codes that support 64bit addressing.

Measured using the PSB [22] test set, the original, single-resolution (SR-) SPRH, which uses *Kullback-Leibler divergence* for its distance, has R-Precision of 37.4%, while its multiresolution SPRH has R-Precision of 42.5%. After the dimension reduction using the LLE, multi-resolution (MR) SPRH clicked in at 49.3% for R-Precision, a 12% overall increase in performance from the original. The best performing combination we have experimented with, the MR-SPRH feature dimension reduced by using the LLE trained by 10,000 samples, tied with the best performer in the SHREC 2006 contest, the method by Makadia et al [26].

This paper is organized as follows. In the following section, we will review learning-based approach to 3D model retrieval. In Section 3, our algorithm is described. In Section 4, we will report on the results of experimental evaluations. We conclude the paper in Section 5 with some remarks on 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 re-searchers for 3D model retrieval [8, 1, 15, 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. [15] performed *Kernel Principal Component Analysis (Kernel PCA)* [10] 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. [15] 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 sub-space includes Principal Component Analysis (PCA) and Multi-Dimensional Scaling (MDS), both of which are quite effective if the feature points lie on or near a linear sub-space 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 non-linear manifold learning algorithms are *Isomap* [25], *Locally Linear Embedding (LLE)* [21], *Laplacian Eigenmaps (LE)* [3], and *Local Tangent Space Alignment (LTSA)* [29].

Manifold learning algorithms aim at preserving their own metric for spatial distortion. The LLE, LE, and LTSA are classified as a “local” approach, which tries to preserve local geometry of the feature space in the learned manifold. These methods might not preserve global geometric shape of the original feature space in the lower dimensional embedding. However, they tend to preserve local metric structure better than a “global” approach such as the *Isometric feature mapping (Isomap)* [25]. “Global” approaches, such as Isomap tries to preserve metric at both global and local spatial scales. In doing so, however, Isomap may incur more local distortions. Also,

Isomap could only be applied to flat manifold, e.g., a cylinder, embedded in a higher-dimensional space.

For our purpose, which is to rank 3D shape features based on their similarity, we expect the locality preserving nature of the LLE, the LE, or the LTSA is preferable to global approaches. What we want are good matches in the top k retrievals, whose features must be positioned close to the feature of the query on the manifold. A property (drawback) of the LE, the LLE, and the Isomap is that the mapping they produce is defined only at 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 [12] solved this problem by using *Radial Basis Function* (RBF) network [10] for a continuous approximation of the manifold.

3. The Method

The method employed in this paper is essentially that of Xhaofei He's approach for 2D image retrieval [12] with some modifications. The method first learns, unsupervised, the subspace spanned by the 3D shape features computed from the models in the training database. The subspace is then used for dimension reduction of the features in the database, to be used for distance computation at the retrieval step. (See Fig. 1.) As mentioned before, we compared the total of six dimension reduction algorithms based on subspace, or manifold learning, linear and non-linear, for their effectiveness. In case of the LE, LLE, and the Isomap, the manifolds learned by using these methods are defined only at the input samples. As we need the manifold defined at an out-of-sample (i.e., out-of-training-set) query models, the learned manifold is approximated continuously and smoothly by using RBF network [10] as in [12].

The following explains the steps the method uses for 3D model retrieval;

Learning:

- (1) **Extract feature:** Extract n -dimensional feature vectors from the K models in the training database (i.e., corpus).
- (2) **Select training samples:** To reduce computational costs, sub-sample, if necessary, the training set down to L ($L \leq K$) feature vectors.
- (3) **Learn the manifold:** Perform unsupervised learning of the m -manifold ($m \leq n$) from the n -dimensional training samples by using a manifold learning algorithm. Certain learning algorithms, e.g., LE and LLE, produce a manifold defined only at the set of training samples. In such a case, to handle queries outside of the training set, continuously approximate the manifold by using RBF network [6].

Database pre-processing:

- (1) **Extract feature:** Extract an n -dimensional feature vector from all the models in the database to be retrieved.
- (2) **Reduce dimension of features of the database models:** Project features of all the models in the database onto the m -manifold (or its approximation), and store the resulting m -dimensional feature together with the corresponding 3D models.

The retrieval phase:

- (1) **Extract feature:** Extract an n -dimensional feature vector from the query model.
- (2) **Reduce dimension:** Project the n -dimensional vector of the query's feature onto the (approximated) m -manifold to obtain a dimension reduced feature of the query.
- (3) **Compute distances on the manifold:** Compute distances from the query model to all the models in the database using the dimension reduced features.
- (4) **Retrieve top p matches:** Retrieve the models in the database having the p -smallest distances from the query model, and present the result to the user.

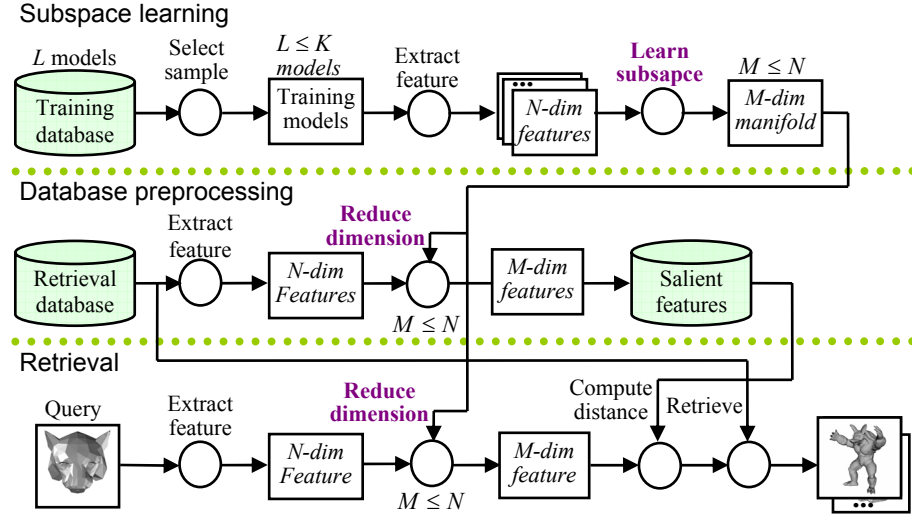


Fig. 1. An unsupervised learning approach to database-adaptive 3D model retrieval.

We used the following five dimension reduction algorithms, all of which are based on unsupervised learning; (1) *PCA*, (2) *LE* [3], (3) *LPP* [11], (4) *LLE* [21], (5) *Isomap* [25], and (5) *KPCA* [10]. The *PCA* and *LPP* learn linear subspaces, while the others learn globally non-linear subspaces. The *LLE*, *LE*, and *Isomap* are sometimes called *graph-based algorithms*, for their computation starts with the construction of a graph that connects feature points in the input m -dimensional space. The *LLE* and *LE* are local methods that try to preserve local distance. The *Isomap* is a global method, which tries to preserve global distance as well. While *LLE* and *LE* produces a sparse connectivity graph, *Isomap* produces a dense one. The *Isomap* thus incur a higher computational cost than the others.

Fig. 2 and Fig. 3 show examples of dimension reduction using the *LE*, *LLE*, and *Isomap*. Data points are synthetic 3D points sampled on 2D surfaces embedded in 3D space. Points have colours to depict the embeddings that resulted. In Figure 2, the *PCA* failed to dissolve four classes in the subspace. All three manifold learning methods, *LE*, *LLE*, and *Isomap*, mapped the data points so that the four colours do not overlap. Globally, the *Isomap* appears to produce the best result without distortion, albeit an increased computational cost. However, local distance metric of the *Isomap* may have suffered compared to the other two, in order to satisfy global geometric

constraints. In Figure 3, the original data points are distributed on a non-manifold surface that has no “correct” 2D embedding. As expected, all three methods failed to represent one of the colours (brown) properly in their 2D embeddings. Representation of the remaining three colours appears about the same among the three methods. The efficacies of these methods applied to higher dimensional data are not obvious from these simple, low-dimensional examples.

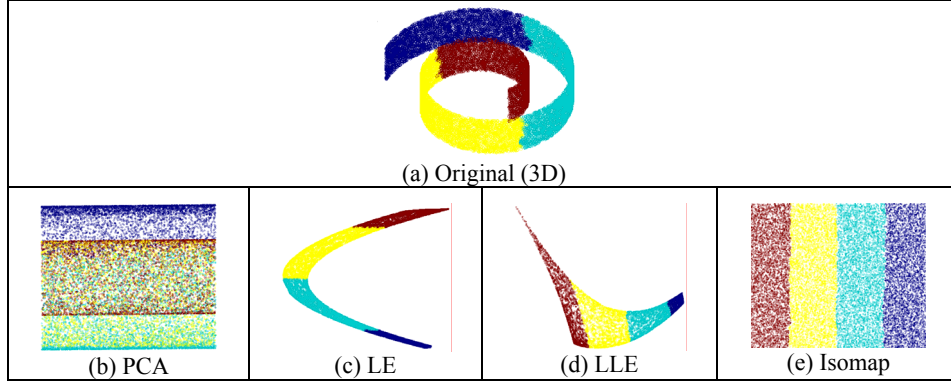


Fig. 2. A toy example for dimension reduction from a 2D manifold “Swiss roll” embedded in 3D space. The LE, LLE, and the Isomap successfully found a 2D manifold. The PCA failed, whose result shows points from different parts of the “Swiss roll” manifold overlapped.

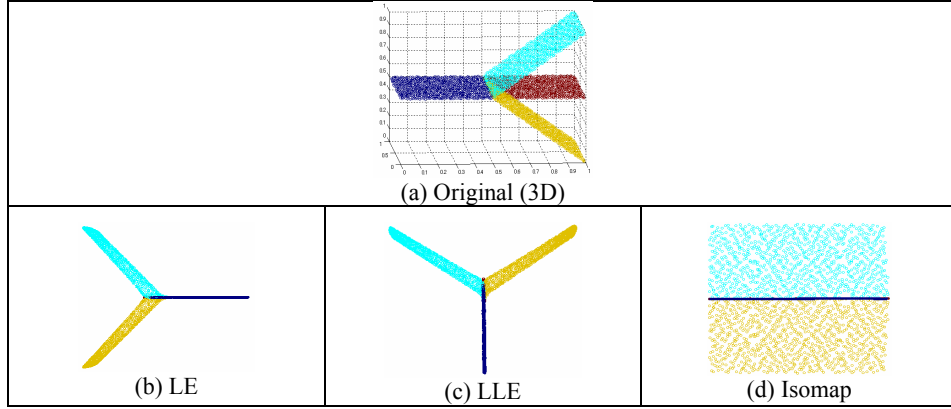


Fig. 3. Another toy example for dimension reduction from a non-manifold surface embedded in 3D space. Finding a flat 2D manifold may be quite difficult or impossible in such a case.

We paired these five dimension reduction algorithms with the five features, the EDED [27], RSH [27], SH [14], AAD [19], and SPRH [28]. We also applied the multiresolution shape feature extraction approach by Ohbuchi et al [18] on these shape features. All in all, we experimented with 10 different shape features, which are, five single-resolution (SR) features and five multi-resolution (MR) features. The multiresolution shape comparison method [18] will be explained in the next section.

3.1. Multiresolution shape comparison method

A better shape comparison may be possible if shapes are compared at multiple scales. For example, for some queries, trees should be compared with each other by their overall shape, not by the shapes of their leaves or branches. Ohbuchi et al proposed an approach [18] for 3D shape similarity comparison that uses a mathematical morphology-like multiresolution (MR) representation (See Fig. 4). The approach first creates a set of 3D MR shape models by using 3D alpha shapes algorithm. The L alpha values for an L -level MR representation is computed from the diameter of the model. Once the MR set of 3D models is obtained, appropriate (single resolution) shape feature is extracted at each resolution level for an L -level MR feature.

Dimension reduction may be applied to an MR set of features in several different ways, e.g., independently at each resolution level, or to a big feature vector created by concatenating feature vectors from all the resolution levels. In the experiment reported in this paper, a dimension reduction method is applied separately at each resolution level. That is, if the MR representation has L resolution levels, L unsupervised learning and then L dimension reductions are performed independently at each level. To compare a pair of 3D models, each having a set of L -level MR features, a distance is calculated at each of the L levels of the MR representation. The L distance values are then combined into an overall distance among the pair of models by using a fixed-weight linear combination of distances. In the experiments described in this paper, all the weights are fixed at 1.0.

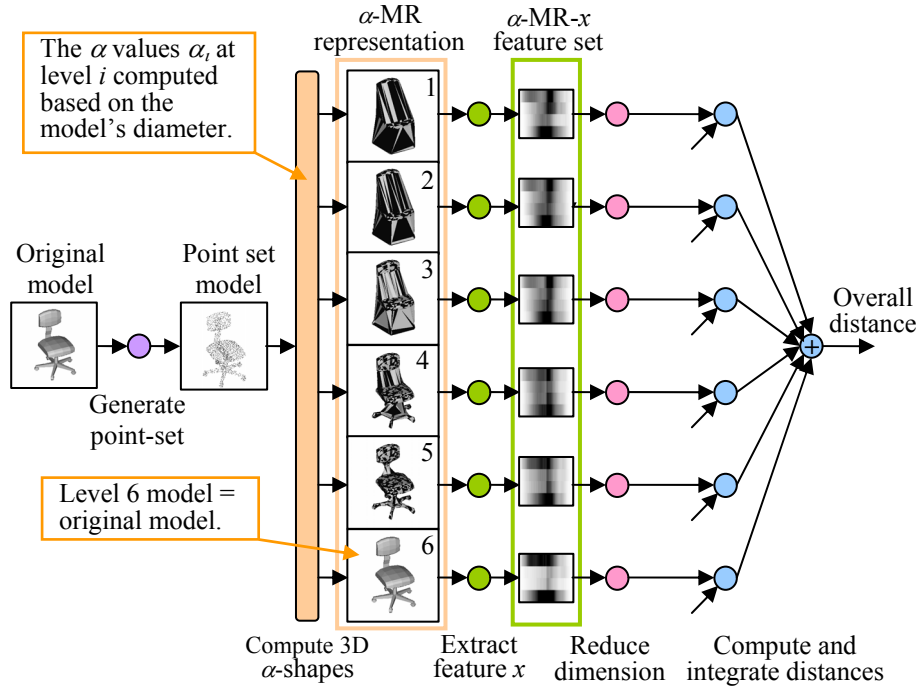


Fig. 4. Multiresolution shape comparison using morphological hierarchy.

4. Experiments and results

We conducted experiments (1) to evaluate the effectiveness of the six dimension reduction methods, and (2) to find the best performing pair of the feature and the dimension reduction method.

The experiments use 3D model databases for two purposes; (1) to train dimension reduction algorithms, and (2) to be queried, e.g., for performance evaluation. To train learning-based dimension reduction algorithms, we used the union of the training set of the *Princeton Shape Benchmark* (PSB) database [22] containing 907 models and the *National Taiwan University 3D Model Database* (NTU) ver. 1.0 containing 10,911 models [17]. The NTU database does not have any labels. The labels in the PSB training set are simply ignored. By using *Niederreiter* sequence, we sub-sampled the union (12,775 models) down to 4,000, 5,000, and 10,000 models to make the learning tractable. We used the SHREC 2006 benchmark [26] protocol as well as its set of tools for computing performance indices for the experiments. Thus, the database to be queried is the union of the PSB training set and the PSB test set as specified in the SHREC 2006. The 30 queries are out-of-database models. While the SHREC 2006 computes six different performance figures, we show only a subset of them in the following.

We computed five shape features using our own code and executables available on the Internet by the original authors of the methods [14, 19, 27]. Codes are also available on the Internet for some of the learning-based dimension reduction algorithms, such as LE, LLE, and Isomap. We wrote our own set of codes in C++ and *MatLab* for retrieval and a part of performance evaluation.

Some of the learning-based dimension reduction algorithms have parameters, e.g., number of output dimensions m , neighborhoods size k for manifold reconstruction, and spreads of RBF kernels σ for the RBF-network approximation. We chose, through experiments, the numbers listed in Table 1 for the experiments below.

Table 1. Features and their parameters for the learning-based dimension reduction.

Learning algorithms	feature	Original feature dimension n	Reduced feature dimension m	Neighborhood size k (%)	RBF kernel size σ
LE	AAD	256	220	0.30	0.6
	SPRH	625	500	1.00	1.2
	RSH	130	90	0.55	5.5
	EDED	544	200	0.75	8.0
	SH	544	200	0.40	9.0
LLE	AAD	256	220	2.00	0.3
	SPRH	625	400	2.00	0.9
	RSH	130	90	1.00	5.5
	EDED	544	200	0.67	8.0
	SH	544	300	0.67	9.0
Isomap	AAD	256	60	1.30	0.6
	RSH	130	90	1.30	5.5

4.1 Effectiveness of various learning-based dimension reduction methods

We first compare the effectiveness of the six learning-based dimension reduction algorithms. Due to space limitation, we show the results for two features only, the AAD (Fig. 5) and the RSH (Fig. 6). Both single-resolution (SR) and multi-resolution (MR) versions of the features are shown in the graphs. For the PCA, we employed the manifold dimensions that resulted in the contribution of 99%, which are listed in Table 2.

Several observations can be made. First, effectiveness of learning methods clearly depends on the feature. The AAD and the RSH behaved differently when dimension reduction methods are applied. For example, LLE method improved the performance of AAD by 4% for the SR case and 6% for the MR case. However, for the RSH, dimension reduction using LLE actually lowered the performance score, while KPCA produced small performance gain. Second, with or without dimension reduction, multiresolution shape features are better than single resolution shape features. Also, learning based dimension reduction methods seem to work better for multi-resolution features. For example, in the case of the RSH, the performance often drops after dimension reduction in case of the single resolution features. However, for the multiresolution features, the performance improves after dimension reduction using the LLE and LE, non-linear, local dimension reduction methods.

Overall, the LE and the LLE, which are local, non-linear dimension reduction methods, produced better performance than the other dimension reduction methods. This tendency is more apparent for the multiresolution cases. Global non-linear methods such as Isomap, and global linear methods such as PCA and LPP did not perform well.

Table 2. Dimensions N of the manifold that resulted in 99% contribution for PCA.

Features	Single resolution	Multi-resolution
AAD	62~58	168~151
RSH	79~78	314~311

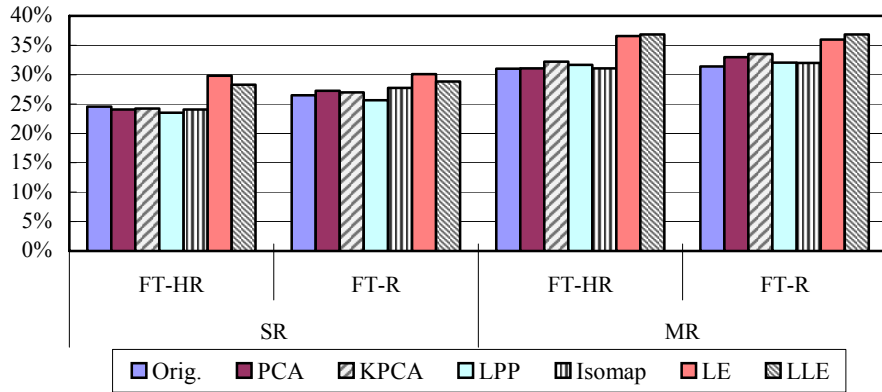


Fig. 5. Dimension reduction methods and retrieval performance for the SR-AAD feature.

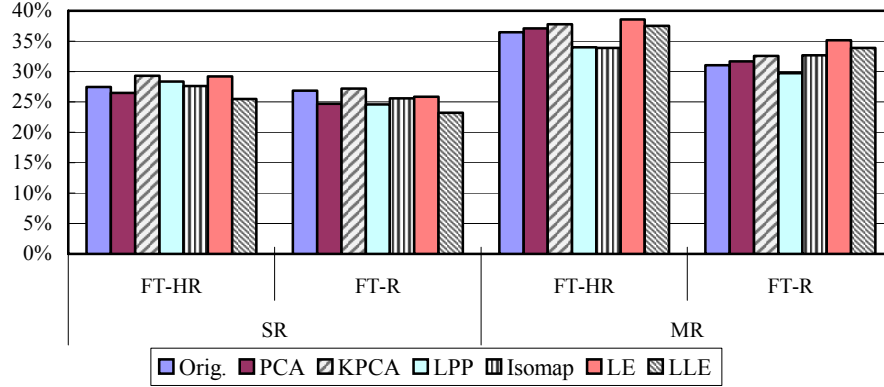


Fig. 6. Dimension reduction methods and retrieval performance for the SR-RSH feature.

4.2 Effectiveness of the LE and LLE methods on five shape features

In this section, we compare the five shape features processed by using two of the most successful dimension reduction methods, the LE and the LLE. The results are shown in Fig. 7 for the single resolution features and in Fig. 8 for the multi-resolution features.

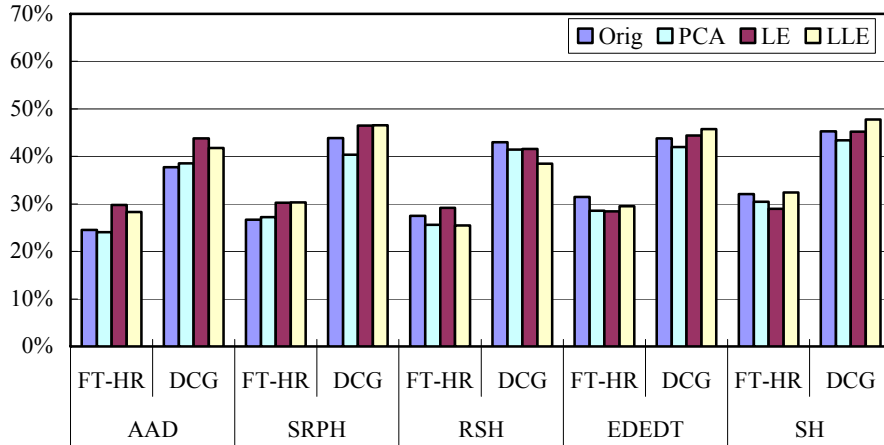


Fig. 7. Retrieval performances in FT-HR [%] of various single-resolution (SR) features after dimension reduction using LE and LLE.

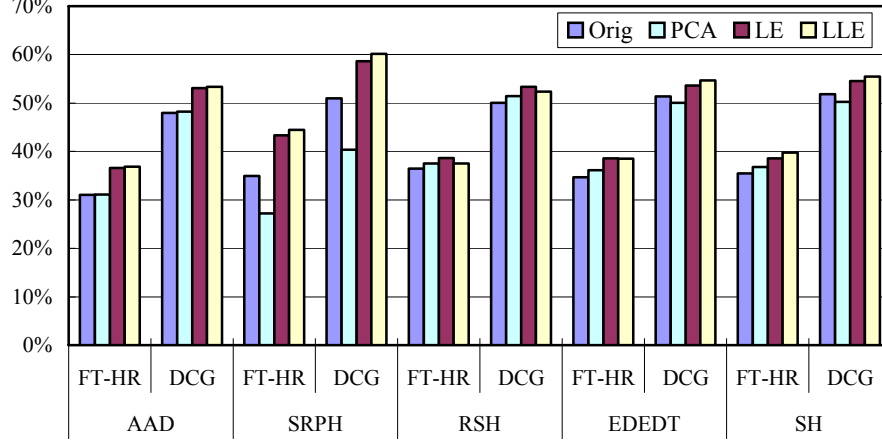


Fig. 8. Retrieval performances in FT-HR [%] of various multi-resolution (MR) features after dimension reduction using LE and LLE.

Gain in retrieval performance depends on the shape feature. For the single resolution cases, all but one feature gained performance by applying dimension reduction using the LE or LLE method. An exception is the RSH feature, in which the dimension reduction produced small degradation in performance. For the multiresolution cases, all the features including the RSH gained performance after dimension reduction using the manifold estimated by using LE or LLE. There appears to be a synergistic relationship between the LE (or LLE) and the multiresolution shape comparison approach [18].

Among the features, before the dimension reduction, and for SR features, the SH and the EDED performed the best. Comparing among the MR features without dimension reduction, the RSH and the SPRH almost caught up with the EDED and the SH in retrieval performance. After the dimension reduction, the MR version of the SPRH somehow outperformed the rest of the features. The reason is unclear, but it is possible that structure of the subspace produced by the SPRH feature was easier to estimate using the LE or the LLE than those produced by the EDED and SH.

4.3 Number of training samples and retrieval performance

In our previous paper [20], we used LE to estimate the manifold spanned by AAD and SPRH features having the number of samples up to 5,000 samples. The experiment showed that the unsupervised learning of the manifold for dimension reduction becomes effective if the number of training samples exceeded about 1,500 samples. The performance kept increasing up to 5,000 samples, which was at the time the maximum number of sample we were able to process using the LE running on the *MatLab*.

Table 3. Performance of the EDED and SPRH features trained by using 5,000 and 10,000 training samples.

Learning methods	SR/MR	Learned models	AP-HR	FT-HR [%]	DAR	NCG @25	NDCG @25
EDED							
-	SR	-	0.3250	31.46	0.4027	0.3862	0.4377
	MR	-	0.3724	34.68	0.4604	0.4692	0.5135
PCA	SR	5,000	0.3030	28.00	0.3733	0.3749	0.4167
		10,000	0.3049	28.55	0.3755	0.3739	0.4198
	MR	5,000	0.3697	36.11	0.4515	0.461	0.5005
		10,000	0.3707	35.64	0.4529	0.4627	0.5022
LE	SR	5,000	0.2949	28.10	0.3722	0.3832	0.4273
		10,000	0.3004	28.42	0.3817	0.3948	0.4437
	MR	5,000	0.3817	38.55	0.4768	0.4955	0.535
		10,000	0.3785	37.28	0.4763	0.5003	0.5363
LLE	SR	5,000	0.3168	29.54	0.3891	0.4083	0.4573
		10,000	0.3088	29.14	0.3877	0.3816	0.4359
	MR	5,000	0.3988	38.49	0.4861	0.5043	0.5442
		10,000	0.3978	37.57	0.4836	0.5098	0.5469
SPRH							
no	SR	-	0.2886	26.68	0.3990	0.3920	0.4384
no	MR	-	0.3761	34.93	0.4631	0.4519	0.5101
PCA	SR	5,000	0.2726	27.20	0.3722	0.3560	0.4035
		10,000	0.2726	26.61	0.3718	0.3560	0.4037
	MR	5,000	0.2726	27.20	0.3722	0.3560	0.4035
		10,000	0.2726	26.61	0.3718	0.3560	0.4037
LE	SR	5,000	0.2958	28.89	0.3819	0.4131	0.4423
		10,000	0.3234	30.27	0.4116	0.4275	0.4647
	MR	5,000	0.4490	43.29	0.5177	0.5377	0.5759
		10,000	0.4586	42.24	0.5250	0.5421	0.5867
LLE	SR	5,000	0.2810	29.64	0.3768	0.3889	0.4295
		10,000	0.3097	30.32	0.4047	0.4294	0.4658
	MR	5,000	0.4614	44.46	0.5341	0.5604	0.5966
		10,000	0.4747	44.44	0.5382	0.5584	0.6013
SHREC 2006 top 2 results (excerpts).							
Makadia (run 2)			0.4869	44.77	0.5499	0.5498	0.5906
Daras (run 1)			0.4475	42.75	0.5242	0.5246	0.5791
Other methods							
LFD [7]			0.4014	38.48	0.4867	0.4889	0.5426
Hybrid [27]			0.4499	44.13	0.5136	0.5032	0.5626

AP-HR: Mean Average Precision (Highly Relevant)

FT-HR: Mean First Tier (Highly Relevant)

DAR: Mean Dynamic Average Recall

NCG @25: Mean Normalized Cumulated Gain @25

NDCG @25: Mean Normlized Discounted Cumulated Gain @25

Here we compare the performance of the features processed using LE and LLE that are trained by using 5,000 as well as 10,000 training samples. We used the *MatLab* 2006b having 64bit address space for the experiments below using 10,000 training samples.

Table 3 shows the results of a pair of features processed by using LE and LLE dimension reduction methods and 5,000 as well as 10,000 training samples. Due to space limitations, only the results for the EDEDT and the SPRH features could be included. In case of the PCA, the number of training sample has had essentially no effect. In cases of the LE and LLE applied to the SPRH feature, the increase in the number of training samples from 5,000 to 10,000 increased the retrieval performance. In the case of the EDEDT feature, however, the increase in the number of training samples from 5,000 to 10,000 often resulted in a small but consistent performance loss.

Table 3 also shows the performance of the 1st and 2nd finishers for the SHREC 2006 [26]. The best performing of our methods is the multiresolution SPRH trained by using LLE and 10,000 samples. It performed on a par with the 1st finisher in the SHREC 2006, and outperformed other powerful methods such as LFD [7] and Hybrid [27]. (We have conducted the experiments that produced SHREC 2006 based performance figures for the LFD and the Hybrid (HBD) features.)

Note that, in the table (and also in Fig. 5 and Fig. 6.) the performance of the SPRH feature is the same for its SR and MR versions. This is not an editorial error, but because the trained MR version actually ended up using only the highest resolution level (Level 6). At all the other levels, i.e., from Level 1 to Level 5, the PCA subspace collapsed due to numerical instability. Subspace learning failures also happened when we tried to use LE or LLE to learn subspace of the Hybrid (HBD) [27], Depth-buffer (DB) [27], and Silhouette features (SHL) [27]. While the exact cause is not known, we suspect that the pose normalization, i.e., normalization of orientation, position, and size, of 3D models employed in all these three features may be related. Imperfect pose normalization, especially in terms of orientation, may have created feature distribution that is quite difficult to learn.

5. Conclusion and future work

Effective retrieval of 3D models based on their shape similarity requires a salient and compact shape feature as well as a good distance measure. In our previous paper, we adopted, for shape-based 3D model retrieval, the approach originally proposed by He et al [12] for 2D image retrieval. The idea is to learn, unsupervised, the subspace, or manifold of features spanned by the set of models in the database, and to use the manifold for dimension reduction. A distance computed using the dimension reduced features is database-adaptive, improving the retrieval performance. Our previous paper showed that the approach using Laplacian Eigenmaps (LE) [3] for manifold learning improved 3D model retrieval performance. However, the feature we tried was limited to two, and the dimension reduction algorithm was limited to the LE only.

In this paper, we explored the approach more comprehensively by comparing six learning-based dimension reduction methods, namely, Principal Component Analysis

(PCA), Kernel-PCA (KPCA), Locality Preserving Projections (LPP), Locally Linear Embedding (LLE), Isometric feature mapping (Isomap), and the Laplacian Eigenmaps (LE). We applied these six dimension reduction methods to five shape features to find a best performing pair. The experimental evaluation showed that two local, non-linear methods, the LE and LLE, produced significant gains in retrieval performance for many of the shape features we tested. A learning based dimension reduction appears to be especially effective if it is combined with the multiresolution shape comparison approach we have previously proposed [19]. The linear, global methods such as PCA and LPP did not significantly improve retrieval performance.

Several avenues of future exploration exist. We would like to see if the off-line unsupervised learning approach used in this paper could be combined effectively with an on-line supervised learning approach based on relevance feedback, or with a supervised learning of pre-defined categories. Such a combination would be able to capture both short-term, local knowledge and long-term, universal knowledge for more effective shape-based 3D model retrieval. We would also like to explore the ways to effectively combine heterogeneous features by using, again, a learning-based approach. Such a combination would produce a very powerful shape comparison and retrieval method.

Acknowledgements

The authors would like to thank those who created benchmark databases, those who made available codes for their shape features, and those who made available codes for various learning algorithms. This research has been funded in parts 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. Ribeiro-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. 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).
9. T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, D. Jacobs, A Search Engine for 3D Models, *ACM TOG*, **22**(1), pp. 83-105, (2003)
10. S. Haykin, *Neural network a comprehensive foundation*, Second Edition, Prentice Hall, (1999).
11. X. He, P. Niyogi, Locality Preserving Projections, *Advances in Neural Information Processing Systems*, **16**, Vancouver, Canada, (2003).
<http://people.cs.uchicago.edu/~xiaofei/LPP.html>
12. X. He, W-Y. Ma, H-J. Zhang, Learning an Image Manifold for Retrieval, *Proc. ACM Multimedia 2004*, pp. 17-23 (2004).
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. M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation Invariant Spherical Harmonics Representation of 3D Shape Descriptors, *Proc. Symposium of Geometry Processing* 2003, pp. 167-175 (2003). <http://www.cs.jhu.edu/~misha/>
15. 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.
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. Pacific Graphics 2003*, pp. 293-302, (2003).
19. R. Ohbuchi, T. Minamitani, T. Takei, Shape-similarity search of 3D models by using enhanced shape functions, *IJCAT*, **23**(3/4/5), pp. 70-85, (2005).
20. R. Ohbuchi, J. Kobayashi, Unsupervised Learning from a Corpus for Shape-Based 3D Model Retrieval, *Poster paper, Proc. ACM MIR 2006*, pp. 163-172, (2006).
21. S. T. Roweis, L.K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, *Science*, **290**(5500), pp. 2323-2326, (2000).
22. 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>
23. *Statistical Pattern Recognition Toolbox for Matlab*,
<http://cmp.felk.cvut.cz/cmp/software/stprtool/index.html>
24. J. Tangelder, R. C. Veltkamp, A Survey of Content Based 3D Shape Retrieval Methods, *Proc. SMI '04*, pp. 145-156.
25. J. B. Tanenbaum, V. de Silva, J.C. Langford, A Global Geometric Framework for Nonlinear Dimensionality Reduction, *Science*, **290**(5500), pp. 2319-2323, (2000).
26. R. C. Veltkamp, R. Ruijsenaars, M. 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>
27. D. V. Vranić, 3D Model Retrieval, *Ph.D. Thesis*, University of Leipzig, 2004.
<http://merkur01.inf.uni-konstanz.de/CCCC/>
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).
29. Zhang, Zhenyue, Zha, Hongyuan, Principal manifolds and nonlinear dimension reduction via local tangent space alignment. *SIAM Journal of Scientific Computing*, **26**(1):313–338, 2004.