Accepted as an oral paper, Proc. 9th ACM SIGMM International Workshop on Multimedia Information Retrieval (ACM MIR 2007), September 28-29, 2007, University of Augsburg, Germany.

Learning Semantic Categories for 3D Model Retrieval

Ryutarou Ohbuchi
University of Yamanashi
4-3-11 Takeda, Kofu-shi, Yamanashi-ken, Japan
ohbuchi A T yamanashi . ac . jp

Akihiro Yamamoto
University of Yamanashi
4-3-11 Takeda, Kofu-shi, Yamanashi-ken, Japan
g05mk039 A T yamanashi . ac . jp

Jun Kobayashi
University of Yamanashi
4-3-11 Takeda, Kofu-shi, Yamanashi-ken, Japan
jun066 A T quartz . ocn . ne . jp

ABSTRACT

A shape similarity judgment among a pair of 3D models is often influenced by their semantics, in addition to their shapes. If we could somehow incorporate semantic knowledge into a "shape similarity" comparison method, retrieval performance of a shape-based 3D model retrieval system could be improved. This paper presents a 3D model retrieval method that successfully incorporates semantic information from human-made categories (labels) in a training database. Our off-line, 2-stage semi-supervised approach learns efficiently from a small set of labeled models. The method first performs unsupervised learning from a large set of unlabeled 3D models to find a non-linear subspace on which the shape features are distributed. It then performs a supervised learning from a much smaller set of labeled 3D models to learn multiple semantic categories at once. Our experimental evaluation showed that the retrieval performance using proposed method is significantly higher than those of both supervised-only and unsupervised-only learning methods.

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.

General Terms

Algorithms, Performance, Experimentation, Measurement.

Keywords

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

1. INTRODUCTION

In the recent years, a number of papers has been published on 3D model retrieval systems that are based on shape similarity [26, 12]. Retrieval performance of such a system has not been satisfactory, however. One of the most promising ways to improve retrieval performance of such a system is to exploit *semantic knowledge* associated with a 3D model. Using semantic knowledge, for example, a system might be able to distinguish bananas from dolphins by paying attention to small differences in their shapes, that are, fins, despite their overall shape similarity.

A semantic knowledge may be classified by its persistence and universality. A *short-term* and/or *local* semantic knowledge is defined for each search occasion or for a person. For example, a user may want an "antique-looking wooden rocking chair of ∩∫≈⌂ kind", where "∩∫≈⌂" being a knowledge still internal to the user. Such short-term and/or local knowledge is best learned *on-line* from each user. Quite a few methods that employ on-line, interactive learning in a relevance-feedback framework have been proposed to capture such knowledge for 3D model retrieval [7, 13, 15, 1]. During an iterative retrieval session, a user gives feedback on the relevance of retrieved set of models. The system learns online from the user feedback, and tries to improve the retrieved set of models for the next round of retrieval. A *long-term* and/or *universal* semantic knowledge stays for a long time and is shared among a group of people. For example, an "office chair" may be a long lived concept shared among a large number of people. This form of semantic knowledge may be captured from a *categorized*, or *labelled*, training database by using an *off-line*, *supervised learning*. Alternatively, long-term semantic knowledge may be captured gradually over a period of time via an on-line, supervised learning.

In the field of shape-based 3D model retrieval, the authors are aware of no published work that employed an off-line, supervised learning to improve retrieval performance. There are two major reasons for this; the training database has (1) a small overall size, and (2) small individual category size. One of the two well established 3D model retrieval benchmark datasets, *The Princeton Shape Benchmark* (*PSB*) database [23] for years had a provision for off-line supervised learning. The PSB, containing 1,814 models, is divided into two equal sized subsets, 907 models each, named "training set" and "test set". The training set and the test set are further divided into 90 and 93 each of "semantic" categories. The training set size of 907 is small considering a high feature dimension (typically tens to hundreds). Size of individual categories is also quite small; many of the categories have only 4 models in them. Increasing the overall size of the training set and/or individual category size would be costly. Our previous attempts to apply some of the supervised learning algorithms directly on the PSB training set was not successful (Figure 1(b)). It is quite laborious to discover a set of classes from, and classify models of, a large number of 3D models. It is not only laborious, but the resulting classification tends to be unstable and noisy.

An approach to deal with the small sample problem is *semi-supervised learning*, which employs both labelled and unlabeled samples for learning. In our previous work [19, 20], we have successfully applied unsupervised learning for 3D model retrieval. The method trains an *Unsupervised Dimension Reduction* (*UDR*) algorithm by using a large set (e.g., 5,000 samples) of 3D models (Figure 1(b)). By employing a locally-constrained, non-linear manifold learning algorithm such as the *Locally Linear Embedding* (*LLE*) [23] or *Laplacian Eigenmaps* (*LE*) [2], 3D model retrieval performance improved significantly. Our intuition then was to train a *Supervised Dimension Reduction* (*SDR*) algorithm in the subspace produced by the preceding UDR step (Figure 2(c)). We call the method *Semi-Supervised Dimension Reduction* (*SSDR*), for it uses information from both unlabeled samples and labelled samples. The UDR is trained, unsupervised, by using a large (dimension k, set size p) set of input, or original unlabeled features. Then, input features of labelled models (dimension k, set size q) are processed by the UDR to produces a set of *interim features* having dimension l ($l < k$). The SDR algorithm is trained, supervised, by using the interim features of the labelled models and their labels. The trained SDR then processes the input features for the models in the database to be searched and the query for a set of *salient* features having dimension m ($m < l < k$) for distance computation and retrieval.

We experimentally evaluated the method under both inductive and transductive settings using the PSB [23] and Shape Retrieval Contest (SHREC) 2006 [28], respectively. In both cases, the SSDR approach showed the largest performance gain. The UDR approach also showed consistent but smaller performance gain than the SSDR. The SDR only approach, however, often performed worse than the original, untreated feature. We also tested the fitness of the supervised learning algorithms under the presence of multimodal categories. A multimodal category arises when a category contains multiple disjunct clusters of input features. The results are yet to be conclusive, but the experiments seem to suggest advantage of a class of SDR methods over the other for retrieving multimodal categories.

In the inductive setting using PSB, the best performing SSDR

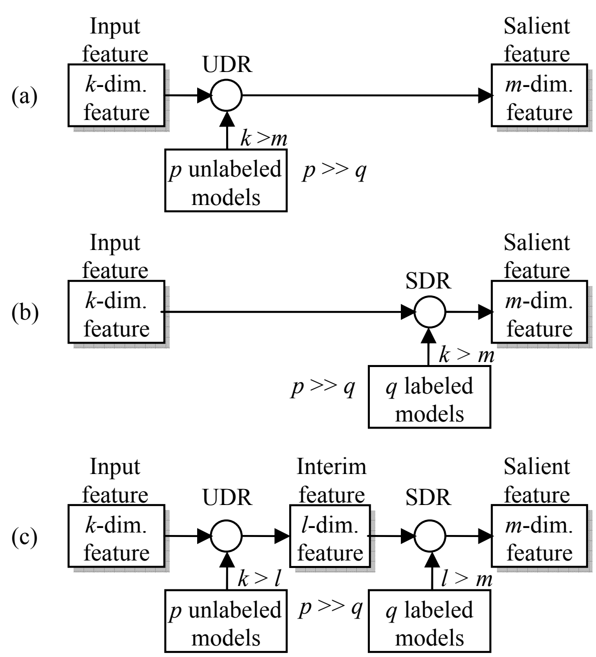


Figure 1: Three dimension reduction methods: *Unsupervised Dimension Reduction* (*UDR*) method (a), *Supervised Dimension Reduction* (*SDR*) method (b), and our proposed *Semi-Supervised Dimension Reduction* (*SSDR*) method (c).

method applied to the EDT feature produced the R-precision of 47%, compared to the 40% of the original EDT. The best-performing multi-resolution combination of SSDR-processed EDT features produced the R-precision of 53%. This R-precision outperforms the *Light Field Descriptor* (*LFD*) [6] whose R-precision is 46%. In the transductive setting measured using the SHREC 2006, the best performing SSDR combination showed First Tier (Highly Relevant) figure of 58%, a number significantly (13%~14%) better than the best result from the SHREC 2006. Furthermore, such a performance gain is obtained by using the feature having a significantly smaller dimension than the original.

In the next section, we will review the use of learning in the context of shape-based retrieval of 3D models. In Section 3, we will describe the proposed retrieval algorithm based on semi-supervised learning. Experiments and their results are described in Section 4, followed, in Section 5, by summary and future work.

2. PREVIOUS WORK

In the field of shape-based 3D model retrieval, *on-line, interactive learning* of semantic concepts in a relevance-feedback framework have been popular [7, 13, 15, 1]. The method used by Leifman, et al [13], for example, performs an UDR using *Kernel Principal Component Analysis* (KPCA) followed by a supervised learning of single class in a relevance feedback framework. The difference between our approach and Leifman's approach is that, their method learns *single semantic category* iteratively and interactively, while our method learns *multiple semantic categories* in a single batch, *off-line*. There is only one published method that exploits pre-categorized training samples. The method, proposed by Bustos, et al [4] may be considered as a "mild" form of off-line supervised learning. It is "mild" for the method uses the categorized training samples to estimate a goodness of shape features. The goodness, called *purity*, is then used to weight a linear combination of distances obtained from multiple (heterogeneous) shape features.

Learning based dimension reduction algorithms can be classified as supervised or unsupervised. The former used labelled, or categorized, training samples, while the latter uses unlabelled samples for the learning. Classical methods for unsupervised dimension reduction include *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 subspace of the ambient (input) space. The PCA tries to preserve covariance structure on the input space. If the subspace is non-linear, however, these linear methods do not work well. *Self-Organizing Map* and *KPCA* are two of the well-known examples of non-linear dimension reduction. (See, for example, Haykin et al [8].) Both PCA and KPCA produce continuous mapping that is defined everywhere in the input, high dimensional space. Recently, a class of geometrically inspired non-linear methods, called "*manifold learning*" has been proposed for learning a *manifold* of an input feature vector space quite effectively. Examples of manifold learning algorithms are the *Isomap* [27], *Locally Linear Embedding* (*LLE*) [22], and *Laplacian Eigenmaps* (*LE*) [2]. The LLE tries to preserve locally linear structure of nearby features. A drawback of *LE*, *LLE*, and *Isomap* is that their map is defined only for the feature vectors in the training set, i.e., a new query. To reduce dimension of a feature outside of the training set, the manifold must be defined everywhere in the input high-dimensional feature space. In a 2D image retrieval setting, He et al [11] solved this problem by approximating the manifold by using *Radial Basis Function* (*RBF*) network [5]. Ohbuchi, et al [19, 20] applied the algorithm proposed by He et al [11] to the task of 3D model retrieval. They showed that, by learning a large (e.g., >1500 models) set of 3D models, an unsupervised non-linear dimension reduction could significantly improve 3D model retrieval performance. In this paper, we employ UDR algorithms with the hope of improving efficiency of supervised learning stage that follows. Specifically, we experimented with three UDR algorithms, the PCA, KPCA, and the LLE.

There are a choice of supervised dimension reduction algorithms, such as the *Supervised-LLE* (*SLLE*) [21], *Supervised Locality Preserving Projections* (*SLPP*) [10], *Local Fisher Discriminant Analysis* (*LFDA*) [24] and its kernelized version *Kernel-LFDA* (*KLFDA*). The SLPP is a supervised version of the LPP (Locality Preserving Projections), which in turn is a linear version of the LE mentioned above. As in the LE, the LPP connects, by geometrical proximity in the input space, training samples into a mesh. In its supervised version SLPP, connections are made also for those feature points that are semantically close, even if they are not geometrically close in the input space. The SLPP then tries to maintain proximity in the input space, both geometrical and semantic, in the projected lower dimensional manifold. The LFDA can also be considered as a supervised extension to the LPP. The LFDA tries to maximize between-class separability while preserving within-class local structure. An advantage of the LFDA as a supervised learning algorithm is its capability to handle multimodality of data in the input space. Multimodality appears when a concept (especially a higher level concept) is consisted of multiple disjunct sets of feature clusters. The method proposed in this paper experimentally compared the SDR algorithms SLPP, LFDA, and KLFDA, in combinations with the UDR algorithms mentioned before, for the retrieval performance.

3. METHOD

The proposed 3D model retrieval algorithm incorporates semantic knowledge from the categorized training database in a single batch, *off-line* learning by using the 2-stage *Semi-Supervised Dimension Reduction* (*SSDR*) (Figure 2). In the 2-stage SSDR, an input (or original) feature having a high dimension k is processed first by using an *unsupervised dimension reduction* (*UDR*) to produce an interim feature having a lower dimension l than the input dimension k. The map for the UDR is computed based on an unsupervised learning from a large (size p) set of unlabeled 3D models. The interim feature is processed further by a *supervised dimension reduction* (*SDR*) algorithm to produce a "salient" feature having dimension m ($m < l < k$). The map that incorporates semantic knowledge used for the SDR is learned from a smaller (size q, $q << p$) set of labelled 3D models. Note that the proposed SSDR approach is not specific to a shape feature. It could be combined with many, if not all, of the existing and forthcoming shape features.

The SSDR-based 3D model retrieval algorithm proposed in

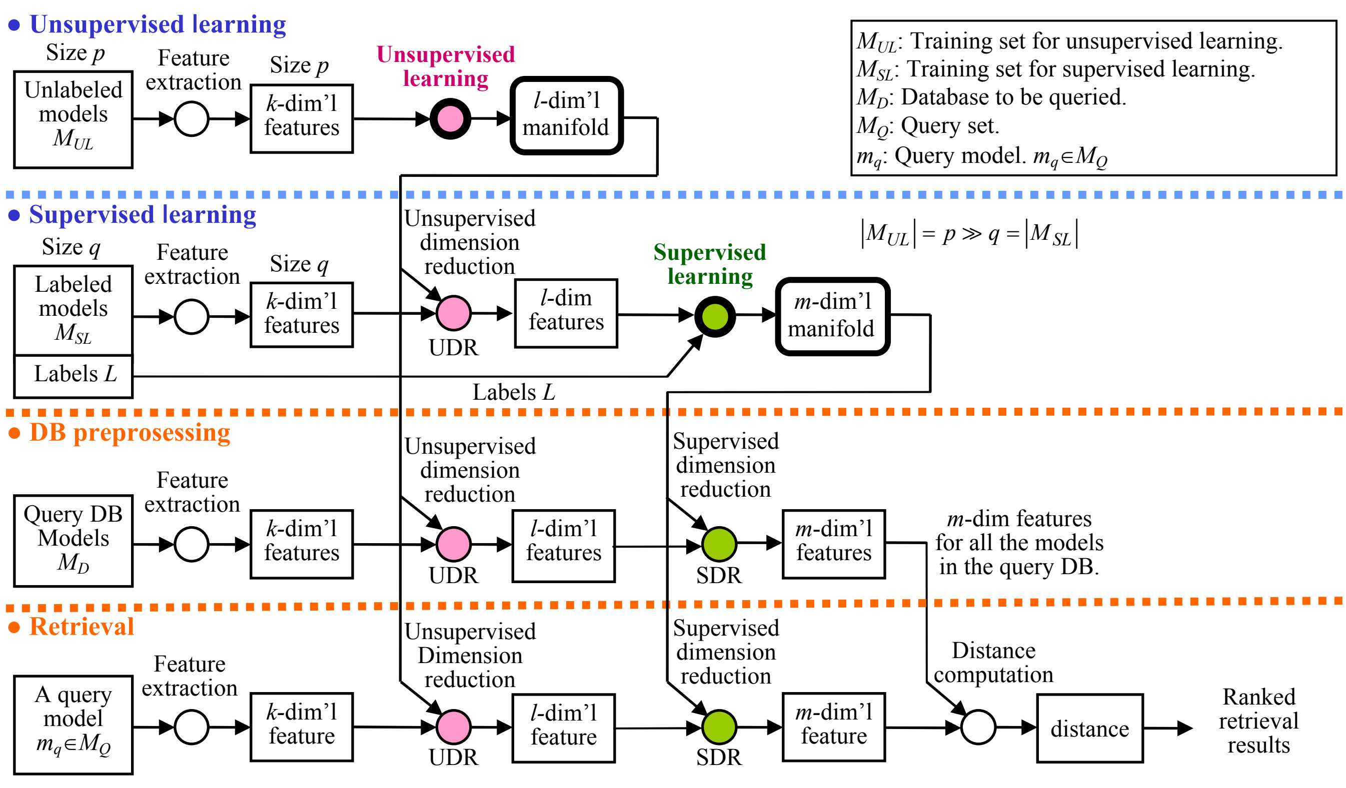


Figure 2. A shape feature is "purified" through the two-stage dimension reduction process which performs an unsupervised dimension reduction followed by a supervised one. The subspace, or manifold used for the dimension reduction steps are learned during the learning phase.

this paper proceeds according to the steps below:

(1) Unsupervised learning: Extract a *k*-dimensional feature from each model in the unlabeled set of 3D models M_{UL} of size *p* for the UDR. Given the set of *p* unlabeled features, an UDR algorithm learns the *l*-dimensional subspace S_{UDR} spanned by the set. The resulting S_{UDR} maps an input *k*-dimensional feature to the interim, *l*-dimensional feature in which $l < k$.

For some UDR algorithms, such as the LLE [22], we must approximate the S_{UDR} so that it is defined everywhere in the input space. We used the RBF-network [5] for the approximation, as proposed by He, et al [10]. The other UDR algorithms, such as PCA and KPCA, do not require the approximation step as their manifold are defined continuously in the input space.

(2) Supervised learning: Extract *k*-dimensional features from the set of labeled 3D models M_{SL} for the SDR. Note that the size *q* of the M_{SL} is typically much smaller than the size *p* of the M_{UL} . A SDR algorithm then learns categories from the *q* labeled features in a batch and encodes the knowledge into the *m*-dimensional subspace S_{SDR} to be used for later SDRs. The manifold maps an interim *l*-dimensional feature onto the salient, *m*-dimensional feature used for retrieval.

(3) DB pre-processing: For each *k*-dimensional input feature of all the models in the database M_D , employ the UDR and SDR in succession to produce *m*-dimensional salient feature that incorporates semantic concepts learned from the labeled models in the training set M_{SL} . Store the salient feature together with the corresponding 3D model for later retrieval. As the dimension *m* of a salient feature is much smaller than the dimension *k* of the corresponding input feature, cost of distance computation and feature storage are significantly reduced. .

(4) Retrieval: Extract the input feature of the query model $m_q \in M_Q$ and perform the UDR and SDR in succession to produce its salient feature having dimension *m*. Compute distances between the salient feature of the query and all the salient features stored in the database. Rank the models in the database according to the distances. Retrieve the *r* top matches from the ranked list having *r* smallest distances.

3.1 Unsupervised Dimension Reduction

For the UDR, we compared the *PCA*, *KPCA*, and *LLE*. For the KPCA and PCA, we used the code available in the *Statistical Pattern Recognition Toolbox* [25], and for the LLE, we used the code available in the *Statistical Learning Toolbox* [14], both for the *MatLab*. (We used the *MatLab*® R7.2.) Borrowing the approach from He, et al [10], we used the RBF network [5] available in the *Neural Network Toolkit* for the Matlab to create a continuous map defined everywhere in an input space from a discrete map produced by the LLE. Both of the KPCA and LLE require parameters. The KPCA may employ different kinds of kernel functions, and, depending on the kernel function, there are additional parameters. We used an RBF (Gaussian) kernel $k(x,x') = \exp(-\|x-x'\|^2 / \sigma^2)$ for the KPCA, and the spread σ is set to the average of distance among the features in the input (*k*-dimensional) space. For the LLE, we used the 0/1 weights (as opposed to the heat kernel) on the edges during the mesh construction step. For the RBF-network approximation of the map produced by the LLE, we used the parameters shown in Table 1, which were found experimentally. The neighbourhood size, that is, the number of neighbouring sample points to be connected, is in percentage points relative to the cardinality of the unsupervised training set M_{UL} . The spreads of the RBF depends on a feature, as the sample points distribution is different from a feature to another. We found the appropriate RBF spread thorugh preliminary set of experiments.

The LLE and Kernel-PCA could be quite expensive both in time and in space if the training set size *p* is large. Our method sub-samples the corpus for unsupervised learning to reduce costs. The UDR is performed by using M_{UL} having 3,500 or 4,000 samples, depending on the UDR algorithm, instead of the full 10,911 unlabeled models of the National Taiwan University (NTU) database [16]. For the sub-sampling, we used *Niederreiter's* quasi-random sequence [3], adopting the method described in [19, 20]

Table 1. Parameters for the RBF network approximation of an LLE map at the UDR learning step.

Features	Number of neighbors	RBF Spread
AAD	2.0	0.3
SPRH	2.0	0.9
EDT	1.0	5.5
SH	0.67	8.0
RSH	0.67	9.0

3.2 Supervised Dimension Reduction

For the SDR step, we compared the *SLPP*, *LFDA*, and *KLFDA*. The SLPP and the LFDA are linear methods, while the KLFDA is a non-linear method. For the SLPP, we used the code provided by Xaofei He at his web site [10]. For the LFDA and KLFDA, we used the code found at Sugiyama's web site [24].The neighbourhood size for the LFDA and KLFDA are fixed to three for all the features we experimented with. In the case of the KLFDA, we used the *inner product kernel* $k(x,x') = x \bullet x'$, and its regularization parameter is fixed to 0.001 for all the features we tried. As SLPP, LFDA, and KLFDA produces continuous maps defined everywhere in the input space, there is not need for approximation.

3.3 Shape features and distance measures

The proposed approach can be applied to almost any shape feature that produces a feature vector. We experimented with five shape features, the *Exponentially-decaying EDT* (*EDT*), *Ray-based Spherical Harmonics* (*RSH*) [29], *Spherical Harmonics* (*SH*) [12], *Absolute Angle-Distance* (*AAD*) [18], and *Surflet-Pair Relation Histograms* (*SPRH*) [30]. We implemented the AAD and SPRH ourselves. We used executable codes available on the net for the EDT, SH, and RSH. The dimensions of the original (input) features are; $AAD = 256$, $SPRH = 625$, $EDT = 544$, $SH = 544$, and $RSH = 136$.

To compute distance among features, we used *Cosine* distance for all the features that are processed by dimension-reduction. For the original (i.e., without dimension reduction) features, we experimentally picked the best performing one among the four distances, the *L1-norm*, *L2-norm*, *Cosine*, and the *Kullbuck-Leibler Divergence* (*KLD*). The selection is performed via a preliminary set of retrieval experiments. For example, we used the L1 for the original EDT, and the KLD for the original SPRH.

3.4 Multiresolution Shape Comparison

A better shape comparison may be possible if shapes are compared at multiple scales. For example, trees may be compared with each other by their overall shape, not by the shapes of their leaves or branches. Ohbuchi et al proposed [17] to use a mathematical morphology-like multi-resolution (MR) representation for a MR 3D shape similarity comparison. The MR approach by Ohbuchi et al. first creates a set of 3D MR shape models by using 3D alpha shapes algorithm given a set of predetermined values of *alpha*, or *scale*. Once an MR set of 3D models is obtained, appropriate (single resolution) shape feature is computed for the model at each resolution level to produce a set of multi-resolution shape features.

A dimension reduction method may be applied to the MR set of features in several different ways, e.g., to a feature vector at each resolution level, or to a big feature vector created by concatenating all the feature vectors from multiple resolution levels. In the experiment reported in this paper, a dimension reduction method is applied separately at each resolution level (Figure 3). That is, if the MR representation has L resolution levels, the total of L dimension reductions are performed independently. To compare a pair of 3D models, a distance is calculated at each of the L levels of the MR representation. These 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.

4. EXPERIMENTS AND RESULTS

We a priori did not know which combinations of UDR and/or SDR methods would work with which shape feature. We compared four dimension reduction strategies, namely, (1) UDR only, (2) SDR only, (3) UDR+UDR, and (4) UDR+SDR (= SSDR). The third strategy, UDR+UDR, is included later to see the cause of performance improvement in SSDR; whether it is the semantic concept learned from the SDR in the 2nd dimension reduction stage, or simply the reduced dimension of the interim feature that facilitated the 2nd stage learning. For the UDR+UDR strategy, we fixed the 2nd stage to the LPP, the unsupervised version of the SLPP, and paired it with the three UDR variations in the1st stage. All in all, we experimented with the total of 19 variations; three UDR, three SDR, three UDR+UDR, nine SSDR, and one original, i.e., no dimensional reduction.

An important consideration in evaluating learning-based retrieval systems is the sets of models, labeled or otherwise, used For example, if the supervised training set equals the test set used for evaluation, the learning is easier as the problem does not involve much generalization. We evaluated the method under the two settings below;

(1) Inductive retrieval using PSB: For this set of experiments, $M_{SL} \neq M_D$, and $M_Q = M_D$. That is, the database to be queried (for performance evaluation) M_D (=PSB test set) is different from the one used for the supervised learning M_{SL} (PSB train set). The query model m_q is drawn from the test database ($m_q \in M_D$), and compared against the rest of models ($M_D - m_q$) in the database. We used the *Princeton Shape Benchmark* (*PSB*) database [23] *training set* (907 models) as the supervised learning models M_{SL}, and the PSB *test set* (907 models) as the M_D. For the unsupervised learning set M_{UL}, we used *p* unlabeled models sub-sampled from the union of the *National Taiwan University 3D Model Database* (NTU) ver. 1 [16] containing 10,911 models and the entire PSB models (907+907=1,814 models). We are not sure if $M_{UL} \cap M_D$ is empty or not, as the M_D (= PSB test set) may share some models with M_{UL}, the union of the NTU database and PSB training set.

(2) Transductive retrieval using SHREC 2006: For this set of experiments, $M_{SL} = M_D$ and $M_Q \notin M_D$. That is, the database to be queried (for performance evaluation) M_D is the same as the set of models M_{SL} for the supervised learning. The query model m_q is not included in the database ($m_q \notin M_D$), however, for the SHREC 2006 [28]. The unsupervised learning is performed in a manner similar to the inductive setting above. In this case, $M_{UL} \cap M_D \neq \varnothing$, as the M_D (= PSB test set + PSB training set) shares models with M_{UL}.

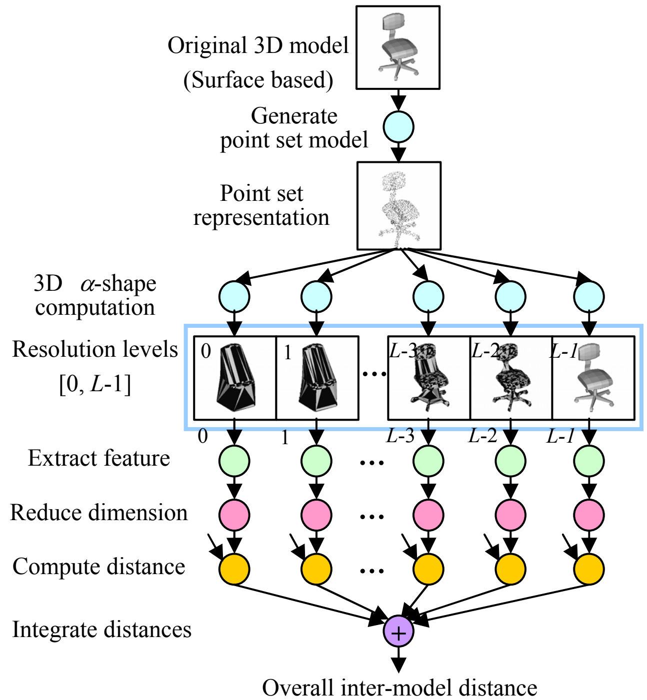


Figure 3. Comparing shapes by using a set of multiresolution models generated from the given 3D model. Dimension reduction and distance computation is performed per resolution level. An overall distance is computed as a linear combination of distance.

For the inductive retrieval experiments, we used *R-precision* (RP) as the performance index. 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 $|C_k - 1|$. The RP values presented below are a mean over all the 907 models in the PSB test set used for the experiment. For the transductive retrieval experiments, we used the indices used in the SHREC 2006, which are, Average Precision (AP), First Tier (FT), Second Tier (ST), Dynamic Average Recall (DAR), Normalized Cumulated Gain (NCG), and Normalized Discounted Cumulated Gain (DCG). All these figures are averages over the 30 query models of the SHREC 2006. For each of these indices, the larger number the better the retrieval performance. Please refer the SHREC 2006 report [28] for these performance indices.

4.1 Inductive Retrieval Using PSB

Table 2 shows, for the EDT feature, details of the relations between the learning strategies, learning algorithms, and the retrieval performance measured in R-precision. For the single-resolution (SR) case, all the combinations of dimension reduction methods we have tried are shown. For the multi-resolution (MR) case, only the best performing combination in each learning strategy is shown. Figure 4 and Figure 5 show further detail of experiments we did by varying the number of salient feature dimensions. We obtained similar tables and plots for all the five shape features, but those for the EDT feature only are shown here for brevity.

For the SR EDT feature, the best performing is the SSDR approach that combined KPCA for UDR and SLPP for SDR, which produced the R-precision RP = 46.98%. It is followed closely by the KPCA + KLFDA (RP = 45.76%) and KPCA + LFDA (RP = 45.34%) combinations. The LLE + KLFDA combination also did well. It is interesting to see that, while the KPCA did not do well if it used alone in the UDR-only setting, KPCA appears the best match for the EDT in the SSDR combinations. Preferred UDR and SDR methods for a SSDR combination depends on the feature. For example, the MR SPRH produced best retrieval performance by using LLE ($l = 90$) for UDR and KLFDA ($m = 90$) for SDR. Overall, a non-linear method, i.e., either the LLE or the KPCA is preferred for the UDR. The PCA, which is a linear UDR method, showed no performance gain.

The best result for the EDT feature was achieved by the SSDR processed MR EDT feature whose R-precision was 53.1%.. It used KPCA ($l = 250$) for the UDR and SLPP ($m = 60$) for the SDR. This performance is significantly better than those of the *LFD* [6] at 45.9% and the *Hybrid* [29] at 48.5% (using L1 norm).

Please note that a significant reduction in feature dimension is accompanied by a significant performance gain due to the SSDR approach. The best performing salient SR EDT feature processed by the SSDR (KPC + SLPP) has the subspace dimension $m = 80$. This is near 1/7 of the dimension $k = 544$ of the original EDT feature. For the MR EDT having six resolution levels, the overall dimension of $3,264 = 544 \times 6$ for the original feature has been reduced to $360 = 60 \times 6$. Such a reduction in dimensionality would have a large impact on the cost of distance computation, which would be repeated many times for a database containing a large number of models. Reduction in feature storage cost for a database is also significant.

Figure 4 compares, for the EDT feature, the performance of UDR method, while fixing the SDR method to the SLPP. It is plotted against the salient feature dimension m. The EDT-MR-PCA250-SLPP, EDT-MR-KPCA250-SLPP, and EDT-MR-LLE250-SLPP used the SSDR strategy, while the EDT-MR-PCA250, EDT-MR-KPCA250, and EDT-MR-LLE250 used UDR-only strategy. The two SSDR combinations, EDT-MR-KPCA250-SLPP and EDT-MR-LLE250-SLPP scored much higher than the others at almost all the value of m. Figure 5 compares, for the EDT feature, the retrieval performance among three SDR methods, fixing the UDR method to the most promising (for the EDT) KPCA. It is again plotted against the salient feature dimension m. This graph shows that the best performance is attained at a low subspace dimension of about m=60, compared to the k=544 of the input dimension for the EDT. While the performance of a SSDR variation does depend on the subspace dimension, the sensitivity is not very high.

Table 2. R-precision (RP) for the EDT combined with various learning methods. All the distance measures are *Cosine* distance, except for the SR and MR variations of the original EDT (Orig.), which used *L1* norm.

EDT		1st stage			2nd stage		RP[%]
k=544		method	p	l	method	m	
SR	Orig.						41.01
	UDR	KPCA	3500	200			40.84
		LLE	4000	250			42.64
		PCA	4000	200			40.01
	SDR				KLFDA	44	40.40
					LFDA	44	21.23
					SLPP	44	18.43
	UDR + UDR	KPCA	3500	200	LPP (UDR)	190	41.09
		LLE	4000	200		150	41.20
		PCA	4000	200		70	37.23
	SSDR (UDR + SDR)	KPCA	3500	250	KLFDA	230	45.76
				250	LFDA	100	45.34
				200	SLPP	80	46.98
		LLE	4000	250	KLFDA	240	43.68
				250	LFDA	40	38.81
				200	SLPP	60	42.77
		PCA	4000	200	KLFDA	40	40.69
				150	LFDA	30	36.98
				150	SLPP	40	37.24
MR	Orig.						42.91
	UDR	LLE		250			45.75
	SDR				KLFDA	44	44.68
	SSDR	KPCA		250	SLPP	60	53.06

Figure 6 and Figure 7 shows the summary of experimental results comparing the retrieval performance among various combinations of features and dimension reduction methods. Figure 6 shows the results for the SR cases, while Figure 7 shows the results for the MR cases. What we did was to choose the best performing combination for each of the UDR, SDR, UDR+UDR (=UDR+LPP), and SSDR (=UDR+SDR) dimension reduction methods. Details such as the details of the UDR and SDR methods, number of interim and salient feature dimensions, etc. are omitted from the graph for brevity. For example, in Figure 5, the

"UDR+SDR" (=SSDR) combination for the SR EDT is actually the UDR step using KPCA followed by the SDR step using SLPP.

Several observations can be made. First of all, the SSDR strategy (i.e., the UDR+SDR combination) outperformed all the others. In fact, for each of the five features, the best performing choice was the SSDR. The degrees of improvements vary from a feature to another and from the SR versions to the MR versions.

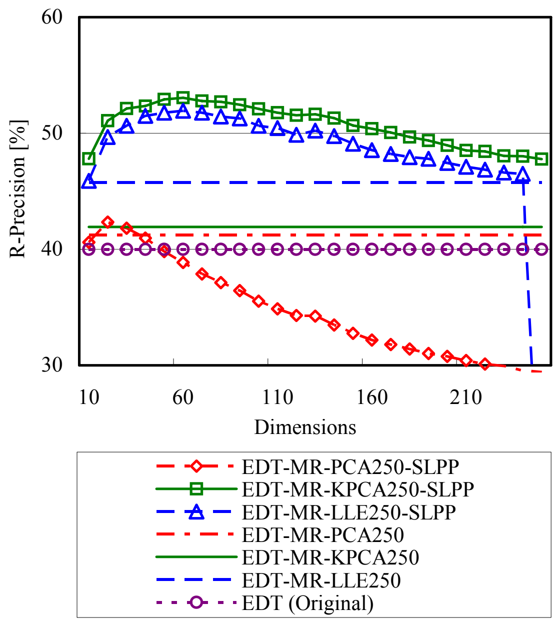


Figure 4. Choice of UDR method and retrieval performance for the MR EDT feature. The SDR method is fixed to the SLPP. The curves are plotted against the salient subspace dimension *m*.

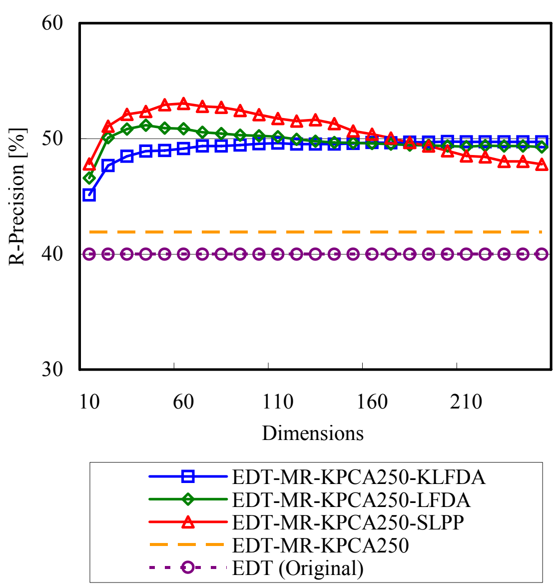


Figure 5. Choice of SDR method and retrieval performance for the MR EDT feature. The UDR method is fixed to the KPCA. The curves are plotted against the salient subspace dimension *m*.

For example, the SPRH appears to benefits more than the others from the SSDR as well as the UDR in the MR cases. Second, the SDR-only strategy did not perform well. The SDR-only variations performed worse than the UDR-only variations. In fact, the SDR processed features almost always performed worse than the original features. This partially explains why a method based on an off-line learning of multi-class semantic concepts has not been reported in the past for 3D model retrieval.

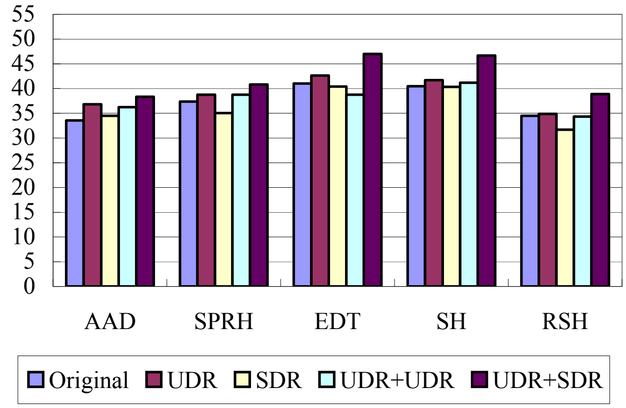


Figure 6. Retrieval performances for the five single-resolution (SR) features using five learning strategies.

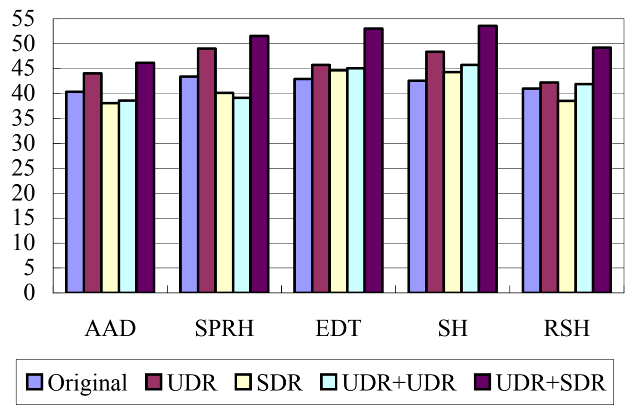
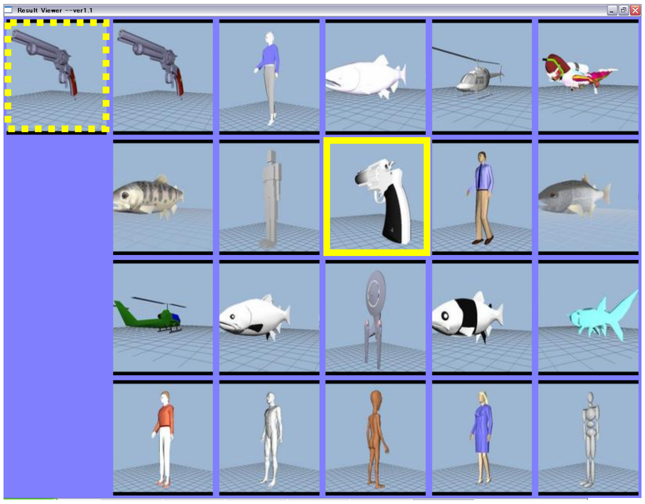
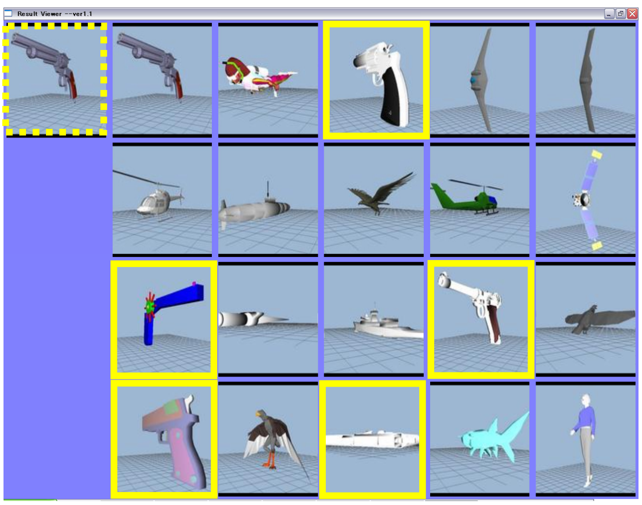


Figure 7. Retrieval performances for the five multi-resolution (MR) features using five learning strategies.

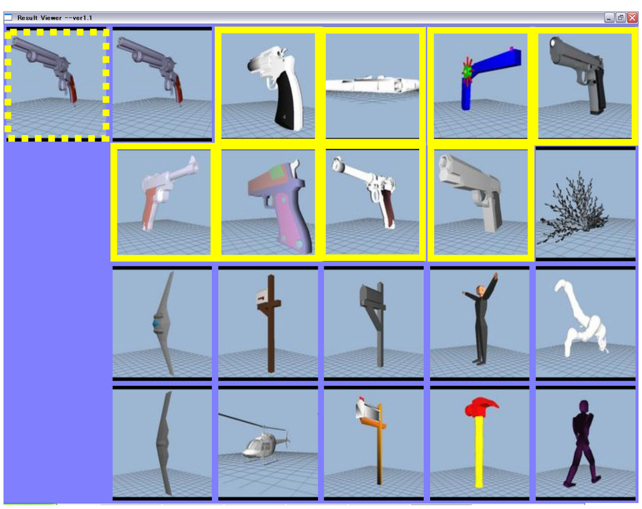
Figure 8 shows examples of retrieval results using four variations of the SH feature given the query "m660.off" (a handgun model) from the PSB test set. The database queried is the PSB test set. In the figure, the query is shown at the leftmost column. A set of top 20 retrievals is shown as a 4 by 5 matrix, in which the leftmost model in the topmost row is the closest match. The top match is often the query itself, and is not counted. A dotted box indicates the query, while a solid box indicates a correct retrieval. The result using the unprocessed multi-resolution SH feature, SH-MR, showed only one model from the correct category in the top 20 (Figure 8(a)). The UDR using LLE improved the performance significantly (Figure 8(b)), retrieving 5 models from the category. The proposed SSDR approach did the best, retrieving 8 models from the correct category in the top 20.



(a) Original MR SH (SH-MR).



(b) UDR-processed MR SH (SH-MR-LLE250).



(c) SSDR-processed MR SH (SH-MR-KPCA300-SLPP90).

Figure 8. Retrieval examples using the PSB. The query is "m660.off" in the PSB test set. The query model is indicated by the dotted box, while the models in the correct category are indicated by solid box.

4.2 Handling Multimodal Categories

A category for semantic concept is often multimodal. That is, a category may consist of multiple clusters that are geometrically distant in the input feature space. For example, a "mammal" model category in a 3D model database may be multimodal, by including models for a human, horse, rabbit, and dolphin. If a supervised learning algorithm forces features in a semantic class geometrically close in the dimension-reduced space, for example, multimodal clusters would disappear, potentially reducing the retrieval performance.

We tried to evaluate the SDR algorithms we employed for their ability to handle multimodality by using the PSB that contains four abstraction levels of categories (Table 3). The most detailed is the "Base" category, containing about 90 categories. It is followed by gradually coarser abstractions having less and less number of categories. The Coarse 1 level contains about 40 categories, the Coarse 2 level contains 7 categories ("Vehicle", "Animal", "Household", "Building", "Furniture", "Plant", and Others), and the coarsest Coarse 3 contains only two categories ("natural" and "manmade"). Comparison of performances of SDR methods using Coarse 2 level or Coarse 3 level categories won't be very useful as these categories are too abstract.

We trained the three SDR algorithms, the SLPP, the LFDA and the KLFDA by using the "Base" level 90 categories of the PSB train set. We then evaluated retrieval performance by using the four abstraction levels of categories. The results for the single resolution (SR) EDT feature (EDT-SR) and multi-resolution (MR) EDT feature (EDT-MR) are shown in Figure 9 and Figure 10, respectively. The SLPP scores the highest for the Base category for both SR and MR cases. The performance of the SLPP drops more quickly than those of the LFDA and KLFDA at the Coarse 1 level in which the categories are more abstract, containing more multimodality. This tendency may reflect the fact that the LFDA and the KLFDA are designed to handle multimodality [24]. These results are still tentative, and we need further study.

Table 3. Abstraction levels and number of categories in the PSB.

	Base	Coarse1	Coarse2	Coarse3
PSB train	90	42	7	2
PSB test	92	38	7	2

4.3 Transductive Retrieval Using SHREC

Table 4 shows the retrieval performance measured in a transductive setting using the SHREC 2006 benchmark. For the SPRH only, we list the performances of UDR-processed multiresolution (MR), unprocessed MR, and unprocessed single-resolution (SR) (i.e., the original) features as well. We also list, in the table, the top 2 performers among the SHREC 2006 contest entrants.

The best performer among the compared is the SSDR-processed MR SPRH feature. The SPRH feature gained more than 30% in *First Tier Highly Relevant* (FT-HR) figure, by going from the original SR SPRH feature (FT-HR of 26.68%) to the MR combination of SSDR-processed SPRH features (FT-HR of 58.13%). Note also the reduction in feature dimension that accompanied the performance gain. At each resolution level, the SSDR-processed MR SPRH feature has the dimension $m = 50$.

Thus, the overall dimension of the SSDR-processed 6-level *MR* SPRH feature is $50 \times 6 = 300$, which is smaller than that of the original SR SPRH feature having $k = 625$. The best performer, the SSDR-processed MR SPRH, performed about 13%~14% better than the best figure in the SHREC 2006 contest by *Makadia* et al. Even the least powerful combination, the SSDR-processed MR AAD, outperformed the SHREC 2006 winner.

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed and evaluated a shape-based 3D model retrieval system that incorporates semantic knowledge captured in a categorized training database. To our knowledge, *off-line supervised learning* of *multiple semantic categories* from a relatively small set of labelled 3D models, e.g., the PSB training set, has not been done previously. Our two-stage, *semi-supervised dimension reduction* (SSDR) method made such learning possible, improving the 3D model retrieval performance significantly. In the method, an unsupervised learning of non-linear subspace from unlabeled set of sample is followed by a supervised learning from labelled (categorized) training set of samples.

Experimental evaluations of the proposed method showed that the method learned a labelled set to achieve significant performance gain together with reduction in feature size. For example, in the set of experiments using the Princeton Shape Benchmark [23] for *inductive* retrieval, the retrieval performance measured in R-precision of the Exponentially-decaying EDT shape feature [30] improved by 7% from 40% to 47%. A linear combination of distances from multi-resolution (MR) set [18] of the SSDR-processed EDT shape features achieved the R-precision of 53%. In the set of experiment for *transductive* retrieval, using the SHREC 2006 benchmark [28], the *FT-HR* figure as high as 58% was obtained by using the SSDR-processed MR combination of the SPRH feature [30]. This figure is significantly better than the results obtained by using some of the previous methods.

Several avenues of future exploration exist. We need to investigate further the combinations of parameters, e.g., kernel functions and their parameters, etc. We would like to see if the off-line supervised learning approach proposed in this paper could be combined effectively with an on-line supervised learning approaches based on relevance feedback. Such a combination might be able to capture both short-term, local knowledge and long-term, universal knowledge for retrieval. We would also like to explore the ways to effectively combine heterogeneous features by using a learning-based approach.

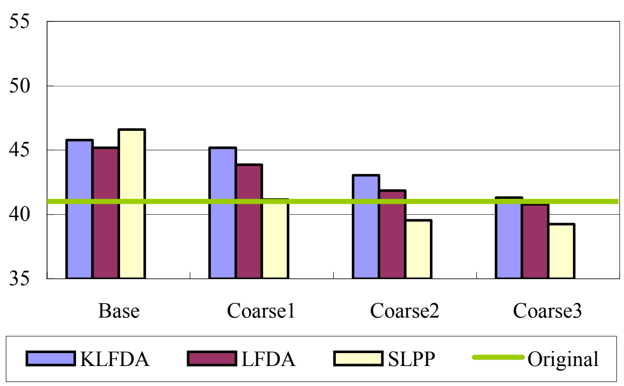


Figure 9. Learning higher level concepts using a single-resolution EDT (SR-EDT) feature.

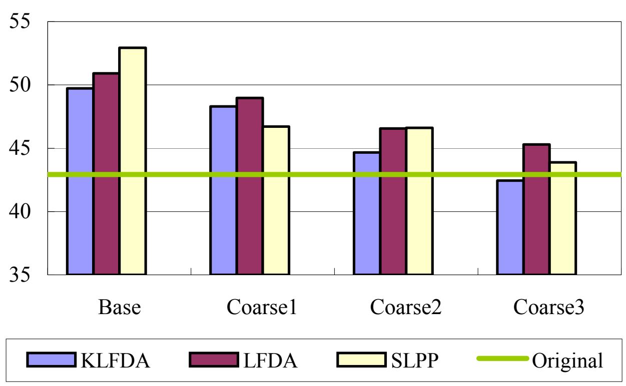


Figure 10. Learning higher level concepts using a multi-resolution EDT (MR-EDT) feature.

ACKNOWLEDGMENTS

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

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

[4] 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).

[5] 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)

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

[7] 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).

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

[9] 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

[10] X. He, W-Y. Ma, H-J. Zhang, Learning an Image Manifold for Retrieval, *Proc. ACM Multimedia 2004*, pp. 17-23 (2004).

[11] 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).

[12] 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/

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

[14] D. Lin, Statistical Learning Toolbox, http://www.mathworks.com/matlabcentral/files/12333/ content/ sltoolbox_r101/sltdoc/index.html

[15] 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).

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

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

[18] 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).

[19] 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).

[20] R. Ohbuchi, J. Kobayashi, A. Yamamoto, T. Shimizu, Comparison of dimension reduction methods for database-adaptive 3D model retrieval, *Proc. 5th International Workshop on Adaptive Multimedia Retrieval* (*AMR*) 2007, Paris, France, July, (2007).

[21] D. de Ridder, O. Kouropteva, O. Okun, M. Pietikäinen, R. P. W. Duin, Supervised locally linear embedding, *Proc. Joint Int. Conf. ICANN/ICONIP 2003*, Lecture Notes in Computer Science, **2714**, (2003).

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

[23] 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

[24] M. Sugiyama, Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis, *Journal of Machine Learning Research,* **8** (May), pp.1027-1061, (2007). http://sugiyama-www.cs.titech.ac.jp/~sugi/software/LFDA

[25] Statistical Pattern Recognition Toolbox for Mat-lab, http://cmp.felk.cvut.cz/cmp/software/stprtool/index.html

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

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

[28] R. C. Veltkamp, R. Ruijsenaars, M. Spagnuolo, R. Van Zwol, F. ter Haar, SHREC2006 3D Shape Retrieval Contest, Utrecht University Dept. Info. and Comp. Sciences, *Technical Report UU-CS-2006-03*0 (ISSN: 0924-3275) http://give-lab.cs.uu.nl/shrec/shrec2006/index.html

[29] D. V. Vranić, 3D Model Retrieval, *Ph.D. Thesis*, University of Leipzig, 2004. http://merkur01.inf.uni-konstanz.de/CCCC/

[30] 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 4. Retrieval performance of the SSDR approach in a *transductive* setting measured using the SHREC 2006 benchmark [28]. In this setting, the database for the supervised learning M_{SL} is the same as the database to be queried and retrieved M_D .

Feature	MR/SR	Dim. Reduc'n	1st stage method	1st stage l	2nd stage method	2nd stage m	distance	AP-HR	AP-R	FT-HR	FT-R	DAR	NCG @25	NDCG @25
SPRH	SR	(None)	-	-	-	-	KLD	0.2886	0.3179	26.68%	31.77%	0.3990	0.3920	0.4384
SPRH	MR	(None)	-	-	-	-	KLD	0.3761	0.3552	34.93%	32.84%	0.4631	0.4519	0.5101
SPRH	MR	UDR	LLE	350	-	-	cos	0.4687	0.4278	45.40%	42.18%	0.5413	0.5567	0.6049
SPRH	MR	SSDR	LLE	350	LFDA	50	cos	0.6196	0.5982	58.13%	55.19%	0.6537	0.6815	0.7017
AAD	MR	SSDR	KPCA	200	SLPP	50	cos	0.5083	0.4897	45.91%	45.86%	0.5797	0.5885	0.6224
EDT	MR	SSDR	KPCA	250	SLPP	70	cos	0.5863	0.5568	53.12%	51.44%	0.5994	0.6462	0.6592
SH	MR	SSDR	KPCA	300	SLPP	70	cos	0.6015	0.5766	54.16%	52.77%	0.6178	0.6677	0.6717
Makadia (run 2)		-	-	-	-	-	-	0.4869	0.4364	44.77%	40.55%	0.5499	0.5498	0.5906
Daras (run 1)		-	-	-	-	-	-	0.4475	0.3952	42.75%	37.03%	0.5242	0.5246	0.5791

AP-HR: Mean Average Precision (highly relevant)
AP-R: 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