Yukinori Kurita, Ryutarou Ohbuchi, View-Clustering and Manifold Learning for Sketch-based 3D Model Retrieval,
short paper, *Proc. CyberWorlds 2013*, Oct. 21-23, 2013, Tokyo, Japan

View-Clustering and Manifold Learning for Sketch-based 3D Model Retrieval

Yukinori Kurita
Graduate School of Medicine and Engineering
University of Yamanashi
Yamanashi-ken, Japan
g11mk014_AT.yamanashi.ac.jp

Ryutarou Ohbuchi
Graduate School of Medicine and Engineering
University of Yamanashi
Yamanashi-ken, Japan
ohbuchi_AT.yamanashi.ac.jp

***Abstract*—Retrieval of 3D models by using 2D sketch query has recently become an active area of research. However, retrieval accuracy of the modality is still much lower than using a 3D model example as query. In this paper, we propose an algorithm that employs manifold-learning based dimension reduction for sketch-based 3D model retrieval. The algorithm compares multi-view rendering of 3D models with the 2D sketch. To improve distance computation, a distance metric adapted to data sample distribution is learned by using a manifold-learning algorithm, that is, Locally Linear Embedding (LLE). In order to lower the cost of training the LLE, we reduce number of training samples by clustering, either in feature space or in view space. Experimental evaluation has shown that both view space clustering and feature space clustering lowers training cost by more than 10 times while significantly improving retrieval accuracy. A compact 50 dimensional feature after the dimension reduction is much faster to compare, and its retrieval accuracy is 40% better than the original 30k dimensional feature. In terms of training cost, view clustering approach costs 20 times less than the one using full set of features without clustering.**

3D shape retrieval; content-based multimedia retrieval; 3D geometric modeling; manifold learning.

I. INTRODUCTION

Three-dimensional shape model (3D model) has been adopted in such wide variety of applications as mechanical design, medical diagnostics, and entertainment (e.g., games). As number of 3D models explodes, organization and management of 3D models through content-based search and retrieval has become an important subject of study.

3D shape models may be queried by 3D model example or by textual tags. But a 3D model appropriate for a query may not be available and 3D models usually has no keywords attached. One could also use photograph of a real object as a query, but such an object may not be around. A viable alternative in querying 3D shape models is to use a hand-drawn 2D sketch as the query [13][11][2][10]. With tablets and phones having touch and/or pen interface everywhere, this is a very attractive alternative.

Sketch-based 3D model retrieval algorithms compare a 2D line-drawing sketch with a 3D shape model by converting the 3D model into dozens of 2D images rendered from multiple viewpoints. The rendering often tries to mimic line-drawing, e.g., by using suggestive contour rendering algorithm [1]. Then, a feature extracted from the sketch is compared to features extracted from the multi-view rendered images of the 3D models.

State of the art sketch-based 3D model retrieval algorithms still suffer from low retrieval accuracies. A major cause for this is the gap between sketches and rendered images of 3D models. Hand-drawn sketches contain stylistic variation (e.g., shaded, v.s. contour), abstraction (e.g., stick figure man), inaccuracy and instability (e.g., wobbled lines). Intra-class variability also exists on the side of 3D models. Consequently, rendered images of 3D models are often not similar to hand-drawn sketches of the "same" class. Simple, fixed distance metrics used in previous algorithms often fails. We need a better approach than a simple distance metric, e.g., L2-norm, for comparing features of sketches and 3D models. A possible approach is to use semantic labels, if available in the database, for supervised or semi-supervised learning. However, labels are often not available for learning.

In this paper, we propose to use an *unsupervised learning* to improve *distance metric*. From the set of (unlabeled) features extracted from images of 3D models, we try to find a distance metric adapted to the distribution of the features in a high-dimensional feature space. Specifically, we proposed to employ non-linear manifold-learning based dimension reduction such as *Locally Linear Embedding* (LLE) [9]. Manifold learning tries to learn, or estimate, a low-dimensional manifold embedded in a high-dimensional input feature space. A distance measured along the lower dimensional manifold (a geodesic-like distance) often produces higher retrieval accuracy than a simple distance, e.g., L1-norm, in the high-dimensional input feature space. Furthermore, reduced dimension the processed features leads to faster comparison among features.

An issue with a manifold learning algorithm such as LLE is training cost. In case of the LLE, training cost depends on the size of the training set and the dimensionality of features. For sketch-based 3D model retrieval, M models in a database and N_v views per 3D model means training set size $M \cdot N_v$. In a bid to reduce training cost of the LLE, we explore two approaches; (1) *View clustering LLE* (*cv_LLE*): cluster features in view space of each 3D model prior to train LLE, and (2) *Anchor clustering LLE* (*anc_LLE*): cluster features of all the views of all the 3D models altogether to find "anchor" features prior to train LLE. We also explore a variation of query-by-sketch scheme that uses more than one sketches per query. To evaluate its effect, we created a multi-sketch query 3D model retrieval benchmark.

Our experiments showed that the LLE with both view clustering and anchor clustering produced better accuracy than without LLE. Retrieval accuracy in Mean Average Precision (MAP) improved from 0.162 without the LLE to

0.215 with the view-clustering LLE. Furthermore, two clustering-based methods did better than the LLE that spent a long time to learn the manifold from all the features. Meanwhile, dimension of feature is reduced from 30k to mere 50, leading to much faster search through the database. As expected, using multiple sketches per query did improve retrieval accuracy, and the improvements are more notable if multiple-sketch query is combined with dimension reduction by using anchor-clustering LLE or view-clustering LLE.

We will describe the proposed algorithm in the next section, followed by description of experiments and result in Section III. We conclude the paper in Section IV.

II. Algorithm

Our sketch-based 3D model retrieval algorithm converts both query sketch and 3D model into silhouette images for image-based comparison. Silhouette, instead of suggestive contour, is chosen as the middle ground as SIFT feature [8] performed better on silhouettes than on line drawings.

A 3D model placed at coordinate origin is rendered from N_v viewpoints spaced uniformly in solid angle. We use N_v=42 in the experiments below, after [4]. From each silhouette image, the algorithm extracts SIFT features densely sampled on or around the silhouette, as with our BF-DSIFT algorithm [4]. For 3D models, a set of SIFT features extracted from each silhouette image is integrated into a per-viewpoint 3D model feature by using Bag-of-Features (BoF) approach. For faster visual vocabulary (i.e., visual codebook) learning and vector quantization, the algorithm uses ERC-Tree [5] clustering algorithm. A 3D model is described by a set of N_v per-view features.

To extract a feature from a sketch, the line-drawing sketch is converted into a silhouette-like image. To do so, the algorithm first tries to close gaps in lines by applying dilation operation. Then, area-fill is applied to create silhouette-like images out of the sketch. This process succeeds on most, but not all, of the sketches. After converting to silhouette, densely sampled SIFT features are are integrated into a feature vector per sketch by using the same codebook learned from silhouette images of 3D models. Both sketch feature and per-view 3D model feature have dimensionality of 30k, typical of BoF histograms.

Our proposed algorithm uses LLE to learn a low dimensional manifold on which distance is computed. To reduce the cost of LLE training, we employ two clustering-based sample reduction algorithms. We compare following four methods for distance computation.

(1) Baseline: Baseline algorithm without dimension reduction.

(2) LLE: LLE is trained by using all the $M \cdot N_v$ features to find the low-dimensional manifold.

(3) cv_LLE: This approach tries to reduce N_v for each 3D model by clustering (Figure 1). The algorithm first computes many (N_v=42) per-view features. Then, *k*-means clustering is applied to find $N_{vc} < N_v$ "synthetic" viewpoints by setting $k = N_{vc}$. Total number of features per database to be learned by LLE is thus reduced to $M \cdot N_{vc}$.

(4) anc_LLE: This approach uses *k*-means algorithm, with $k = N_{anc}$, to cluster $M \cdot N_v$ per-view features of 3D models in a database at once to find $N_{anc} < M \cdot N_v$ "anchor" features (Figure 2). These N_{anc} anchor features, found as centroids of clusters, are used to train the LLE. Similar approach is used previously by [3] and [7].

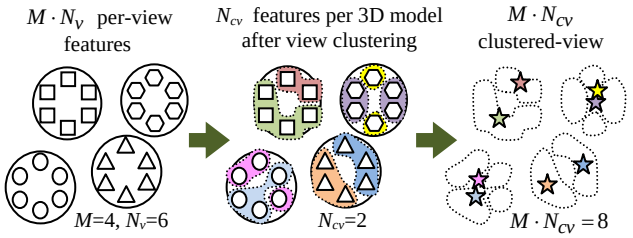


Figure 1. View clustering: For each 3D model, N_v per-view features are clustered into $k<N_v$ distinct "synthetic views".

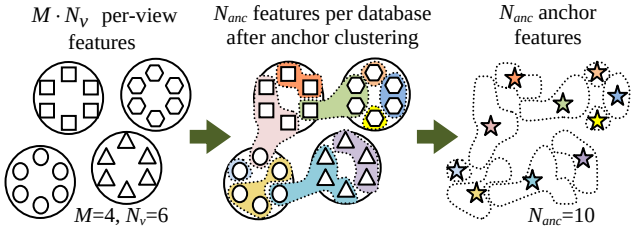


Figure 2. Anchor clustering: All the $M \cdot N_v$ features extracted from all the models are clustered into $N_{anc} < M \cdot N_v$ anchor features.

To reduce dimension of features using the LLE, the feature is projected onto the low-dimensional manifold found by training the LLE. However, the LLE can't handle out-of-sample features. Thus, low-dimensional manifold is approximated by using the RBF-network, as proposed by He, et al [6].

For both cv_LLE and anc_LLE, we used *k*-means cluster centers as new set of features. One could do otherwise, for example, one could select, for each cluster, one of original features closest to the computed centroid. We decided to use centroids since centroids produced better retrieval accuracy, esp. for cv_LLE, than using one of original features.

To compute distance among a sketch feature $\mathbf{x}_p$ and per-view feature $\mathbf{x}_q$ of 3D models, all the methods (1)~(4) use Kullback Leibler Divergence (KLD) below.

$$D_{KLD}(\mathbf{x}_p, \mathbf{x}_q) = \sum_{i=1}^{m} (x_{p,i} - x_{q,i}) \log\left(\frac{x_{p,i}}{x_{q,i}}\right) \quad \textbf{(1)}$$

An overall distance between a sketch and a 3D model is the average of all the distances from the sketch feature to all the features associated with the 3D model. For example, for the cv_LLE, it is the average of N_{cv} distances from the sketch to the N_{cv} (synthetic view) features of the 3D model.

III. Experiments and Results

We conducted experiments to compare evaluate three LLE-based algorithm for their retrieval accuracy and computational cost. For the cv_LLE and the anc_LLE, influences of N_{cv} and N_{anc} are evaluated. We also evaluated effectiveness of multi-sketch query.

LLE has a set of parameters, the neighborhood size k_{nn} to form a graph representing manifold, the size of reduced dimension R, and RBF kernel radius C. We fixed k=500, R=50, and C=250 for the experiments below.

As the index of retrieval accuracy, we use Mean Average Precision (MAP).

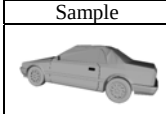
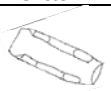
A. *Multi-sketch query benchmark*

We created a sketch-based 3D model retrieval benchmark whose query sketch set consists of 3 sketches per query. That is, an object the user wish to retrieve is illustrated by using 3 sketches, typically from different viewpoints. As the retrieval target, we selected M=497 models (classified into 40 categories) from the 907 models (classified into 92 categories) of the Princeton Shape Benchmark (PSB) test set [12]. We used the subset since some of the PSB models are not conducive to sketching. Three sketches per 3D model are generated as follows;

1. Present a 3D model to the subject by using a very simple interactive viewer that is capable of user-controlled rotation of the model.
2. Ask subject to draw, using a pen tablet, three views of the 3D model.

We asked 10 subjects to draw 3 sketches per 3D model for the 160 models in the database. Thus, we have 160 sets of queries, each having three sketches (that is, total of 480 sketches). Some of the experiments below used all the 3 sketches per query, while other experiments used only one of the three sketches per query. If only one of the sketches is used as a query in an experiment, the sketch having the smallest ID number is selected. Table I shows examples of 3D models and their respective query sketches drawn by the subjects.

TABLE I. EXAMPLES OF SKETCHES IN THE BENCHMARK (RIGHT) AND 3D MODELS (LEFT) SHOWN AS EXAMPLES TO SUBJECTS.

Sample	Sketch 1	Sketch 2	Sketch 3
			
			
			
			

B. *View clustering (cv_LLE)*

In this experiment, we evaluated the impact of number of clustered views N_{cv} on retrieval accuracy.

One might assume that clustering N_v=42 per-view features down to a smaller number N_{cv} of "synthetic" viewpoints would negatively impact retrieval accuracy, as information is inevitably lost. However, as Figure 3 shows, retrieval accuracy improved as the number N_{cv} decreased. The clustering might have eliminated spurious views and thus spurious matches.

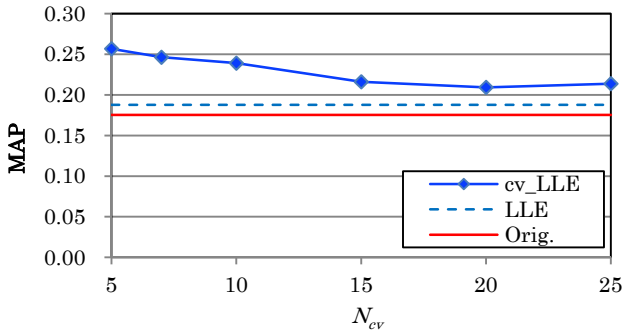


Figure 3. Number of view clusters and retrieval accuracy.

C. *Anchor LLE*

With M=497 models and N_v=42, there are 20,874 per-view image features of 3D models in the database. We clustered them down to N_{anc}=100~2000 using the anc_LLE, for retrieval experiments. As Figure 4 shows, retrieval accuracy stayed pretty much the same until it went down a bit at N_{anc} =100.

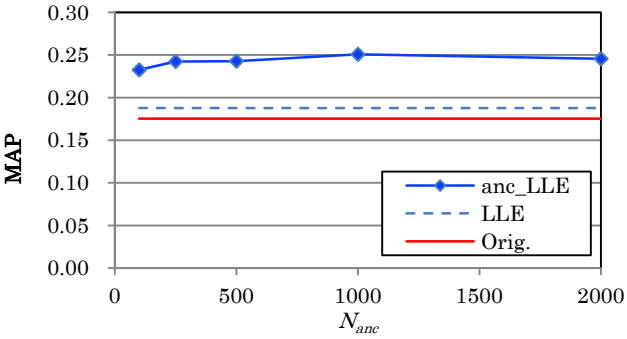


Figure 4. Number of anchors and retrieval accuracy.

D. *Multi-sketch query and manifold learning*

As we assumed, using multiple sketches per query did improve retrieval accuracy as shown in Figure 5. It is a trade-off for a user; accuracy or cost of drawing more sketches.

It appears that performance improvement due to multiple query sketches is more prominent for the cv_LLE and the anc_LLE, in which distance metric learning is more effective, than the other two cases. We also observe that the anc_LLE appears to work better than the cv_LLE for multiple sketches per query case. Tetrieval accuracy for the 1 sketch query is about equal among cv_LLE and anc_LLE. However, for the 3 sketch query, accuracy of the anc_LLE is better than the cv_LLE.

E. *Number fo sketches, retrieval accuracy, and computational cost*

Table II. lists computational costs of four methods along with their retrieval accuracies measured in MAP. This set of experiments is done by using 3 sketches per query.

If we compare retrieval accuracy, methods using the LLE did better than the one (Baseline) without the LLE. Of the three cases that employed the LLE, those using training sample reduction (the cv_LLE and the anc_LLE) produced higher accuracy than the one without, that is, the LLE traind by using all the training samples. For this 3 sketch query case, MAP scores for the cv_LLE and the anc_LLE are 40% and 47% better, respectively, than the Baseline without the LLE.

Dimension reduction impacted positively on the retrieval cost. Using compact 50-dimensional features, the LLE, the cv_LLE, and the anc_LLE took only 1/4[s] per query to search through the database. In comparison, the Baseline using the 30k dimensional features took nearly 20[s].

As for the training cost, clustering based approaches, the anc_LLE and the cv_LLE did significantly reduce the combined cost of LLE and RBF approximation. However, total cost of preprocessing a database must include the cost of clustering for the cv_LLE and the anc_LLE. The cv_LLE has an advantage here, as it is very quick to cluster 42 features. In comparison, the anc_LLE took a very long time, 2,489 min or about 40 hours, to cluster 20,874 per-view features in the entire database. Thus for the total cost of preprocessing, the cv_LLE wins handily; the cv_LLE took 223 min while the anc_LLE took 2,489 min to preprocess.

Overall, for this benchmark dataset, cv_LLE is the best trade-off among retrieval accuracy, retrieval cost, and database preprocessing cost. For databases that are significantly larger, a combination of approaches used in the cv_LLE and the anc_LLE may be necessary.

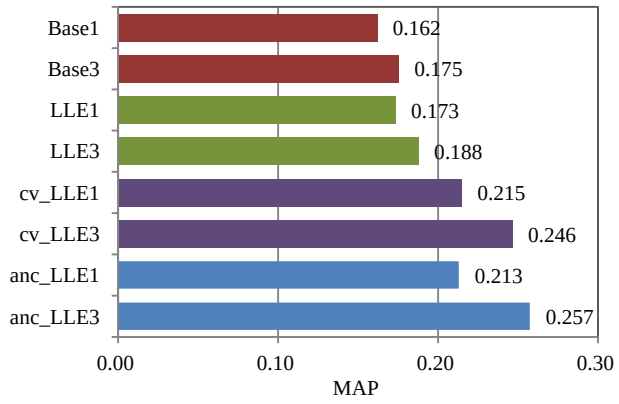


Figure 5. Retrieval accuracy of 4 algorithms measure by using 1 sketch (suffix 1) and 3 sketches (suffix 3) as query.

TABLE II. COMPUTATION TIMINGS. TIME TO PRE-PROCESS DATABASE (I.E., CLUSTERING, LLE, AND RBF-NETWORK APPROXIMATION) ARE IN MINUTES, WHILE TIME TO RETRIEVE SINGLE QUERY IS IN SECONDS.

3 sketches per query	Preprocessing database [min]			Retrieval (per query)[s]	Retrieval accuracy [MAP]
	k-means	LLE+ RBF	Total	Distance comp.	
Baseline.				19.62	0.175
LLE	0.00	4350.65	4350.65	0.24	0.188
cv_LLE	1.95	221.15	223.10	0.26	0.246
anc_LLE	2489.48	29.95	2519.43	0.24	0.257

IV. CONCLUSION AND FUTURE WORK

To improve retrieval accuracy and to reduce cost of feature comparison of sketch based 3D model retrieval, this paper explored manifold learning based dimension reduction of per-view image features by using the LLE [9]. We also explored effect of multiple-sketch query. To this end, we created a benchmark having a set of multiple sketch queries.

To reduce cost of training the LLE, we explored two approaches to reduce number of training samples of the LLE. The anc_LLE clusters the entire features set of the database at once, while the cv_LLE clusters, for each 3D model, per-view features of the 3D model.

Experimental evaluation showed that clustering based reduction of features, both the cv_LLE and the anc_LLE, do reduce LLE training cost while significantly improving retrieval accuracy. At the same time, dimension reduction positively impacted the cost of comparing a sketch to 3D models for retrieval. The experiments also showed that multi-sketch query and manifold learning based dimension reduction work synergistically to improve retrieval accuracy.

ACKNOWLEDGEMENTS

Funding provided by the Ministry of Education, Culture, Sports, Sciences, and Technology of Japan (No. 18300068).

REFERENCE

[1] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, A. Santella, Suggestive Contours for Conveying Shape, *ACM TOG*, **22**(3), pp. 848-855, (2003).

[2] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, and M. Alexa, Sketch-Based Shape Retrieval, *ACM TOG*, **31**(4) pp.1-10, (2012).

[3] M. Endoh, T. Yanagimachi, R. Ohbuchi, Efficient manifold learning for 3D model retrieval by using clustering-based training sample reduction, *Proc. IEEE ASSP 2012*, pp. 2345-2348, (2012).

[4] T. Furuya, R. Ohbuchi, Dense sampling and fast encoding for 3D model retrieval using bag-of-visual features, *ACM CIVR* 2009, Article No. 26, (2009).

[5] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, *Machine Learning Journal*, **63**(1), pp. 3-42 (2006).

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

[7] W. Liu, J. Wang, S-F. Chang, Hashing with graphs, *Proc. ICML* 2011, (2011).

[8] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, *Int'l Journal of Computer Vision*, **60**(2), November 2004.

[9] S. T. Roweis, L. K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, *Science*, **209**(22), pp. 2323–2326, (2000).

[10] J. M. Saavedra, B. Bustos, T. Schreck, S. M. Yoon, M. Scherer, Sketch-based 3D model retrieval using keyshapes for global and local representation, *Proc. 5th Eurographics conference on 3D Object Retrieval (EG 3DOR)* 2012, pp. 47-50, (2012).

[11] T. Shao, W. Xu, K. Yin, J. Wang, K. Zhou, B. Guo, Discriminative sketch-based 3D model retrieval via robust shape matching, *Computer Graphics Forum*, **30**(7), (*also as Proc. Pacific Graphics 2011*) , (2011).

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

[13] S. M. Yoon, M. Scherer, T. Schreck, A. Kuijper, Sketch-based 3D model retrieval using diffusion tensor fields of suggestive contours. *ACM Multimedia 2010*, pp. 193-200, (2010).