Ryutarou Ohbuchi, Takahiko Furuya, Distance Metric Learning and Feature Combination for Shape-Based 3D Model Retrieval, *poster paper*, Proceedings of the ACM workshop on 3D object retrieval 2010, Firenze, Italy, (2010).

Distance Metric Learning and Feature Combination for Shape-Based 3D Model Retrieval

Ryutarou Ohbuchi
University of Yamanashi
4-3-11 Takeda, Kofu-shi
Yamanashi-ken, 400-8511, Japan
ohbuchiAT yamanashi.ac.jp

Takahiko Furuya
University of Yamanashi
4-3-11 Takeda, Kofu-shi
Yamanashi-ken, 400-8511, Japan
snc49925AT gmail.com

ABSTRACT

This paper proposes a 3D model retrieval algorithm that employs an unsupervised distance metric learning with a combination of appearance-based features; two sets of local visual features and a set of global features. These visual features are extracted from range images rendered from multiple viewpoints about the 3D model to be compared. The local visual features are bag-of-features histograms of a set of Scale Invariant Feature Transform (SIFT) features by Lowe [7] sampled at either salient or dense and random points. The global visual feature is also a SIFT feature sampled at an image center. The proposed method then uses an unsupervised distance metric learning based on the Manifold Ranking (MR) [15] to compute distances between these features. However, the original MR may not be effective when applied to a set of features having certain distance distribution. We propose an empirical method to adjust the distance profile so that the MR becomes effective. Experiments showed that the retrieval algorithm using a linear combination of distances computed from the proposed set of features by using the modified MR performed well across multiple benchmarks having different characteristics.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Information filtering. I.3.5 [Computational Geometry and Object Modeling]: Curve, surface, solid, and object representations. I.4.8 [Scene Analysis]: Shape.

General Terms

Algorithms, Experimentation.

Keywords

Content-based retrieval, manifold ranking, distance metric learning, bag-of-features, 3D geometric modeling, 3D object retrieval, feature combination.

1. INTRODUCTION

Effective and efficient management of 3D models, especially via content based retrieval by their shape [5][11], has become an important tool in such diverse areas as mechanical design, medical diagnosis, 3D game, as well as movie and 3D TV content production.

Of various 3D model retrieval methods, view-based approaches, the Light Field Descriptor (LFD) [1] being an early example, have enjoyed high retrieval performance. The view-based approaches also have an advantage, for they are relatively immune to variations in shape representations, e.g., voxels, B-Rep solids, polygon soup, point set, etc. Recently, view-based 3D model matching approaches employing multi-view, multi-scale, local visual features, e.g., the BF-SIFT [8] and the BF-DSIFT [3] has been proposed. To integrate thousands or tens of thousands of local features extracted from multiple images, the method employed a Bag-of-Features (BoF) approach [2][10][13]. The BoF integration produces a feature per 3D model, making the model to model comparison faster and easier than comparing sets of local features. The method produced very good retrieval performance for articulated models, and performed as well as the other methods for rigid models. However, their distance computation employed one of several fixed distance metrics. It has been shown in image retrieval and other fields that a distance metric adaptive to distribution of features in the feature space improves distance computation, and thus retrieval performance.

In this paper, to further improve retrieval performance, we apply Manifold Ranking of Zhou et al [15] on the features produced by bag-of-visual features algorithms for 3D model retrieval. In doing so, we discovered that certain distribution of distances made the MR ineffective for high-dimensional feature vectors produced by some of bag-of-features approaches. We thus propose an empirical method to adjust the distance distribution in order to improve efficacy of the MR algorithm. Experiments showed that our modified MR algorithm, called *DA-MR*, for *Distance Adjusted Manifold Ranking*, works quite well.

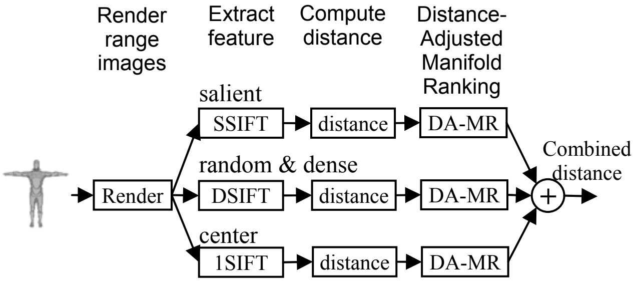


Figure 1. An overview of the proposed algorithm. The Distance Adjusted Manifold Ranking (DA-MR) is effective for high-dimensional features of the BF-DSIFT [3].

3DOR'10, October 25, 2010, Firenze, Italy.
Copyright 2010 ACM 978-1-4503-0160-2/10/10...$10.00.

We also experiment with combination of both local and global visual features for more *robust* 3D model retrieval algorithm. A method using local feature only could have difficulty distinguishing some of the shapes, e.g., pipes bent in U shape and in S shape, for they have almost identical local features. A global feature, on the other hand, would have difficulty handling articulated shapes, e.g., snakes in various different posture. We explore combinations of (1) *bag of salient local visual feature*, (2) *bag of dense and random local visual feature*, and (3) *global visual feature*, to see if a robust combination could be found.

Contributions of this paper can be summarized as follows.

(1) Identification of a difficulty in applying the Manifold Ranking [15] algorithm for certain high dimensional features often produced by bag-of-features approach. We propose an empirical remedy for the difficulty. Experimental evaluation of the proposed remedy shows its effectiveness.

(2) Demonstration of a combination of local and global features that could produce a 3D model retrieval method that perform robustly for a database containing unknown composition of rigid and articulated/globally deformed models.

The remaining parts of this paper are structured as follows. We will describe the proposed algorithm in Section 2. Empirical evaluation of the proposed algorithm will be presented in Section 3, followed by summary and future work in Section 4.

2. ALGORITHM

In this section, we first describe two variations of our bag of local visual features approach to 3D model feature extraction [8][3]. We then describe a method to improve distance computation among a pair of such feature vectors by using the Manifold Ranking [15] algorithm having distance profile modification.

2.1 Local and Global Visual Features

We combine three methods to compute feature a vector describing a 3D model. The first two, the *Bag-of-Features Salient SIFT* (*BF-SSIFT*) [8], *Bag-of-Features Dense SIFT* (*BF-DISFT*) [3] by Ohbuchi, Furuya, Osada, et al. are based on local features extracted from multi-view range image renderings. These local features are integrated into a feature vector per 3D model by using the bag-of-features approach.

For the global feature, we propose a simple one; *per-View Match One SIFT* (*VM-1SIFT*) based also on SIFT. The VM-1SIFT also employs multi-view range image renderings, but extract a global feature, which is a SIFT feature, per range image. Thus a 3D model is described by a set of global features whose number equals the number of views.

2.1.1 Bag-of-Local Visual Features

The BF-SSIFT and BF-DSIFT computes a feature per 3D model following the steps below;

1. **Partially normalize pose:** Both BF-SSIFT and BF-DSIFT perform partial pose normalization for translation and uniform scaling so that the model is centered at origin and fits within a sphere of radius 1.0.
2. **Render multi-view range images:** Render range images of the model from N_r viewpoints spaced evenly in the solid angle. We used N_r=42 for the experiments below, following the paper by Furuya, et al. [3]. The rendering is done on a GPU via OpenGL API.
3. **Extract SIFT features:** From the set of range images, extract local, multi-scale, multi-orientation, visual features by using the *Scale Invariant Feature Transform* (*SIFT*) [7] algorithm. A SIFT feature encodes position, orientation, and scale of gray-scale gradient change about its sample point. For the BF-SSIFT, we employ the original SIFT algorithm, which first detects interest, or "salient" points, and then computes 128D SIFT feature at each of these interest points. For the BF-DSIFT, we disable the interest point detector, and sampling positions are chosen randomly and densely. The BF-SSIFT extracts ~1k SIFT features per 3D model, while BF-DSIFT extracts about ~10k SFIT features per 3D model.
4. **Quantize SIFT features into visual words:** Encode a local feature into a *visual word* drawn from a vocabulary of size N_v by using a pre-learned codebook. The codebook is learned, unsupervised, from tens of thousands of SIFT features extracted from a set of models, e.g., the models in the database to be retrieved. The encoding, or Vector Quantization (VQ), is a closest point query in a high dimensional (e.g., 128D for SIFT feature) space. To speed up the learning and encoding, the algorithm uses *Extremely Randomized Clustering Tree* (*ERC-Tree*) by Guerts, et al [4].
5. **Generate histogram:** Visual words are accumulated into a histogram having N_v bins, which then becomes the feature vector for a 3D model.
6. **Compute distance:** Dissimilarity among a pair of feature vectors (the histograms) is computed. The distance may be computed by using "fixed" distance metrics, such as *Kullback-Leibler Divergence* (*KLD*) or L1-Norm, or by data-adaptive distance metric, e.g., by using the Manifold Ranking (MR) algorithm by Zhou et al [15].

There are many possible methods to compute distance $d(\mathbf{x}_i, \mathbf{x}_j)$ among features $\mathbf{x}_i$ and $\mathbf{x}_j$ We compared the following three. Let n be the dimension of the vectors. Then, L1 norm $d_{L1}(\mathbf{x}_i, \mathbf{x}_j)$ is defined as follows.

$$d_{L1}(\mathbf{x}_i, \mathbf{x}_j) = \sum_k^n \left\| x_{ik} - x_{jk} \right\| \tag{1}$$

The second, Cosine distance $d_{\cos}(\mathbf{x}_i, \mathbf{x}_j)$, gives angular cosine distance between the two vectors;

$$d_{\cos}(\mathbf{x}_i, \mathbf{x}_j) = 1 - \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\left\|\mathbf{x}_i\right\| \cdot \left\|\mathbf{x}_j\right\|} \tag{2}$$

Another distance is the *Kullback-Leibler Divergence* (*KLD*), which is sometimes referred to as *information divergence*, or *relative entropy*. It is known to work well for comparing histograms. We used its symmetric version as follows;

$$d_{KLD}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{i=1}^{n} \left(x_{kj} - x_{ki} \right) \ln \frac{x_{kj}}{x_{ki}} \tag{3}$$

Normally, distance among a pair of 3D models is the distance among their respective features computed by using one of the distances above. In this paper, we employ the Manifold Ranking [15], with a modification discussed in Section 2.3, so that the resulting distance is adapted to the distribution of features in the feature space. In experiments described in Section 3, we compare the three distances above with their MR-treated versions for retrieval performance.

2.1.2 *Global Visual Feature*

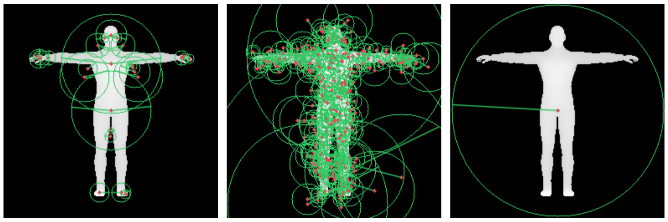
The global visual feature is called 1SIFT, and is based also on Lowe's SIFT. After rendering the 3D model from N_r viewpoints, as with the BF-DSIFT, the 1SIFT extracts only one SIFT feature (thus the name 1SIFT) at the center of each range image so that the SIFT feature acts as a global feature per range image. The VM-1SIFT does not employ the BoF approach, but uses a set of N_r SIFT features to describe a 3D model.

To compute distance among a pair of 3D models, the VM-1SIFT compute distances among all the $v \times v$ pairs of SIFT features, and choose the minimum of these distances as the overall distance among the 3D model pair. Let $\mathbf{x}_{ip}$ and $\mathbf{x}_{jq}$ be the 1SIFT features from the view *p* and *q* of the 3D models $\mathbf{X}_i$ and $\mathbf{X}_j$, respectively. Then the distance $D(\mathbf{x}_i,\mathbf{x}_j)$ between models $\mathbf{X}_i$ and $\mathbf{X}_j$ are defined as below. Here, the distance $d\left(\mathbf{x}_{ip},\mathbf{y}_{jq}\right)$ is one of the Cosine, L1, or KLD distances shown above.

$$D(\mathbf{x}_i,\mathbf{x}_j)=\sum_{p=1}^{N_i} min\left\{\sum_{q=1}^{N_i} d\left(\mathbf{x}_{ip},\mathbf{y}_{jq}\right)\right\} \tag{4}$$

The VM-1SIFT performed surprisingly well for a database of rigid shapes. As shown in Section 3.1, it performed better than the LFD [1] for the PSB [9] database, for example.

Figure 2a and Figure 2b show examples of sampling patterns for a range image of the SSIFT and the DSIFT. Number of samples per range image N_p for the SSIFT is determined "automatically" by the interest point detector of the SIFT algorithm. N_p for the DSIFT, on the other hand, is a user-defined parameter. Overall, we set the numbers of features per 3D model to be 1.3k for the SSIFT and 13k for the DSIFT in the experiments below. Figure 2c shows an obvious sampling pattern of the 1SIFT. The number of sample per range image for the 1SIFT is, by definition, one.



(a)SSIFT, N_p=38 (b)DSIFT, N_p=304 (c)1SIFT, N_p=1

Figure 2. Sampling patterns of a range image for the BF-SSIFT and BF-DSIFT.

2.2 Data-Adaptive Distance via Manifold Ranking

Intuitively, the MR algorithm simulates "diffusion" of relevance value from a feature point (e.g., that of a query) on an irregularly connected graph of high-dimensional features points (e.g., that of all the 3D models in a database). The graph, or mesh, is typically formed by connecting a point to its *k*-nearest-neighbors. The proximity used during mesh generation is determined by using a distance measure, e.g., *L1*-norm, in the ambient feature space. This distance metric affects the mesh topology. Each edge of the graph is added with a weight computed from the distance among the vertices of the edge. The weight represents "diffusion coefficients" of the relevance value over the edges; the higher the weights, the more easily the relevance value diffuses. The diffusion of relevance value happens on a graph from the source, i.e., a query.

The meshing step creates the affinity matrix $\mathbf{W}$ where $\mathbf{W}_{ij}$ indicates the similarity between feature points $\mathbf{x}_i$, and $\mathbf{x}_j$;

$$\mathbf{W}_{ij}=\begin{cases}\exp\left(-\dfrac{d(\mathbf{x}_i,\mathbf{x}_j)}{\sigma}\right) & if\ \ i \neq j \\ 0 & \text{otherwise}\end{cases} \tag{5}$$

Note that $\mathbf{W}_{ii}=0$ since there is no ark connecting a point with itself. The matrix $\mathbf{W}$ is positive symmetric.

After the meshing, the algorithm forms a normalized *graph Laplacian* $\mathbf{L}$,

$$\mathbf{L}=\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-\frac{1}{2}} \tag{6}$$

where $\mathbf{D}$ is a diagonal matrix in which $\mathbf{D}_{ij}$ equals to the sum of the *i*-th row of $\mathbf{W}$, that is, $\mathbf{D}_{ij}=\sum_j \mathbf{W}_{ij}$ The ranking vector $\mathbf{F}=\left[f_1,\cdots,f_n\right]^T$ can then be estimated by iterating the following until convergence;

$$\mathbf{F}^{(t+1)}=\frac{1}{1+\mu}(\mathbf{I}-\mathbf{L})\mathbf{F}^{(t)}+\frac{1}{1+\mu}\mathbf{S} \tag{7}$$

The parameter $\mu>0$ is a regularization parameter, and affects retrieval performance and the convergence of the iteration above. Let $\mathbf{F}^*$ be the limit of the above iteration. Rank each point x_i as a label $y_i=\arg\max_{j\le c} f_{ij}^*$.

The MR algorithm iteratively diffuses the initial value of label source vector $\mathbf{S}=\left[s_1,\cdots,s_n\right]^T$, in which $s_i=1$ corresponds to the query. At the equilibrium, the higher the diffused relevance value at a point, the higher the similarity rank of the point to the query. As the diffusion occurs on the manifold via multiple paths, similarity ranks thus computed are better than those computed directly in the input feature space. The distance metric $d\left(x_i,x_i\right)$ used for forming the affinity matrix $\mathbf{W}$ affects diffusion of rank values during the ranking process. If $\mathbf{W}_{ij}$ is very low, due to a very high $d\left(x_i,x_i\right)$, the rank value won't diffuse easily, possibly impeding the ranking process.

Computational cost is an issue to be considered for applying MR. MR is performed once per query, and cost of MR is dominated by the cost of meshing and computing $\mathbf{F}^*$. The cost increases with the size of matrix $\mathbf{L}$, that is, the number of features *n*.

2.3 Distance Statistics and Manifold Ranking

While MR is known to work in various multimedia information retrieval settings, our initial attempt in applying MR to the BF-DSIFT and BF-1SIFT degraded their retrieval performances. On the other hand, MR boosted the performance of the BF-SSIFT. We investigated the cause, and found that the MR did not work for certain distribution of distances. Figure 3 shows distance distribution for the BF-SSIFT, the BF-DSIFT, and the VM-1SIFT, computed by using the KLD. The profile is computed by using the MSB, and is an average over all the queries.

In the figure, "Original" indicate the distances without the proposed adjustment. The profile for the original BF-DSIFT shows a very large jump in distance from the query (rank 0) to its nearest neighbor (rank 1). As equation (5) for $\mathbf{W}_{ij}$ indicates, such a large distance from query to its nearest neighbor creates edges having very low diffusion coefficients, preventing the diffusion of relevance rank over the network of features. We thus devised a simple empirical *Distance Transformation* (*DT*) to remove the jump, which is to simply subtract from all the distance values the amount of the jump prior to computing weights $\mathbf{W}_{ij}$. In Figure 3, adjusted distance plots are indicated as "Adjusted"

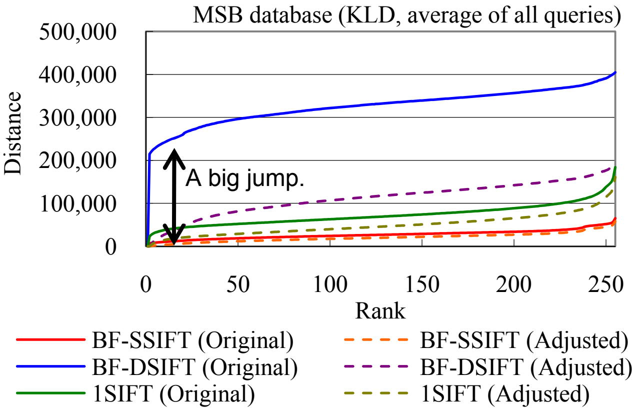


Figure 3. Nearest neighbor is very far from the query for certain (feature, distance metric) combination, e.g., (MSB BF-DSIFT, Kullback-Leibler Divergence).

2.4 Multiple Feature Combination

To combine multiple features, we employ a simple late-fusion approach, by computing linear combination of distances derived from multiple features. Here, each distance may or may be the result of Manifold Ranking.

3. EXPERIMENTS AND RESULTS

We performed the experiments using three benchmark databases: the *MSB* [14] for highly articulated but less detailed shapes, the *PSB* [9] for a set of diverse and detailed shapes. Figure 4 shows examples of 3D models from the two databases. The MSB consists of 255 models in 10 classes. The MSB include such articulated shapes as "humans", "octopuses", "snakes", "pliers", and "spiders". The PSB contains two equal-sized subsets, the training set and test set, each consisting of 907 models and about 90 classes. For our evaluation, we used the PSB test set partitioned into 92 classes. The PSB contains a more diverse set of 3D shapes than the MSB. To evaluated retrieval performance using the PSB, we used every model in the PSB test set of 907 models as queries to retrieve models in the test set. Similarly for the MSB, all the models in the database are used as queries.

Same database is used for both visual codebook learning and retrieval experiments. That is, the codebook generated by using the MSB database, for example, is used to query the MSB. We used the training set size $N_t = 50{,}000$ of SIFT features extracted from multi-view images of models in a database for training. For both codebook learning and for retrieval, we used the following set of parameters. We fixed, for the BF-SSIFT and BF-DSIFT, the number of views N_r =42, and range images size 256×256 pixels. Number of samples per range image for the BF-DSIFT is fixed at N_p=300. Vocabulary size N_v for the BF-SSIFT and BF-DSIFT depends on visual complexity of the range images, and thus on the complexity and diversity of shapes in the 3D model database.

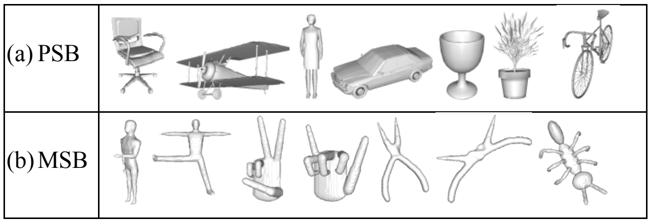


Figure 4. Examples of 3D models from the Princeton Shape Benchmark (PSB) (a) [9] and the McGill Shape Benchmark (MSB) (b) [14].

We chose the vocabulary size (codebook size) N_v based on preliminary experiments. The N_v used for the experiments are listed in Table 1. For the BF-SSIFT, we used *k*-means clustering for codebook learning, combined with a linear search for feature encoding (i.e., vector quantization). This is feasible since the number of SIFT features per 3D model of 700~1,100 and vocabulary size N_v~1.2k are both moderate. Due to the use of *k*-means clustering for codebook learning, the vocabulary sizes for BF-SSIFT are rounded numbers. In comparison, the BF-DSIFT generated, on average, about 13k SIFT features per 3D model. The ERC-Tree vector quantizer used in the BF-DSIFT is a randomized algorithm, and its number of vocabulary N_v is not rounded. Vocabulary size N_v for BF-DSIFT is controlled indirectly by a parameter. The MR has the regularization parameter μ and the region of influence parameter σ . In the following experiments, we varied μ and σ in the range $0.0025 \le \mu \le 50.0$, and $0.01 \le \sigma \le 0.5$, respectively, and used the best performing combination of the parameters.

As the performance index, we used Recall-Precision plot and *R-precision*. R-precision is a 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 $|C_k|$.

Table 1. Vocabulary sizes used for the experiments.

Database	Method	Vocabulary size N_v	R-Precision [%]
PSB	BF-SSIFT	1,200	44.8
	BF-DSIFT	30,215	54.1
MSB	BF-SSIFT	900	75.7
	BF-DSIFT	31,770	75.5

3.1 Distance Adjustment for the MR

This experiment evaluates effectiveness of the proposed *Distance Adjustment* (*DA*) in improving MR. Table 2 and Table 3 compare retrieval accuracies in R-Precision of five features, the BF-SSIFT, BF-DSIFT, VM-1SIFT, SHD and SPRH. We used the executables available online for the SHD and the SPRH. Table 2 and Table 3 show results for the PSB and MSB benchmarks, respectively. The distance $d(\mathbf{x}, \mathbf{y})$ is computed using three methods, the KLD, L1-norm, and Cosine distance (COS). Three rows for each feature are; (1) "None" for no MR, (2) "MR" for original MR, and

(3) "DA-MR" for MR with DA. For each table, "*" indicates the best performer for the feature.

Without the DA, the MR is not effective for the BF-DSIFT and the VM-1SIFT. However, with the DA, performances of the VM-1SIFT and the BF-DSIFT improved significantly. For example, MR with DA pushed the performance of the BF-DSIFT up to R-Precision=60.4% for rigid models (PSB), and R-Precision=90.7% for articulated models (MSB). For some other features, such as the BF-SSIFT and SPRH [12], the original MR (without DA) was more or less effective. However, in some cases, the DA improved efficacy of the MR for those features as well. For example, for the MSB, the SPRH feature benefited more from the DA-MR than the original MR. The SHD feature is an exception among those five features evaluated; the DA did not have clear influence on its retrieval performance using MR.

Figure 5 and Figure 6 show recall-precision plots for the VM-1SIFT, BF-SSIFT, and BF-DSIFT, each with and without the DA-MR. For comparison, performances of the SHD and LFD features without the MR (i.e., original features) are plotted.

Table 2. Manifold ranking and retrieval performance with and without Distance Adjustment (DA) (PSB database).

Algorithms		R-Precision [%]		
		KLD	L1	COS
BF-SSIFT	None	44.8	33.6	42.3
	MR	48.0	46.9	44.2
	DA-MR	* 50.3	*50.8	45.6
BF-DSIFT	None	54.1	54.4	51.1
	MR	53.1	54.5	51.5
	DA-MR	*60.4	59.3	55.8
VM-1SIFT	None	50.9	44.2	42.2
	MR	44.4	48.1	49.2
	DA-MR	*56.5	52.7	51.4
SHD	None	35.4	39.6	38.3
	MR	39.0	40.7	*42.0
	DA-MR	38.8	*42.5	41.7
SPRH	None	36.3	36.3	34.6
	MR	*39.4	38.1	36.6
	DA-MR	*39.9	39.1	36.7

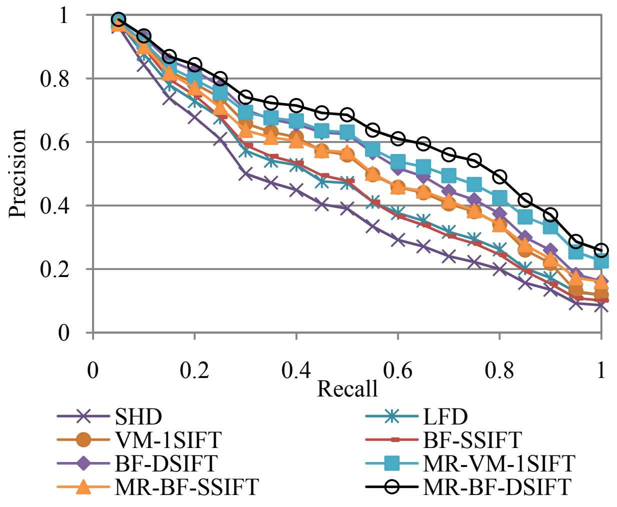


Figure 5. Recall-precision plot for the PSB database.

Table 3. Manifold ranking and retrieval performance with and without Distance Adjustment (DA) (MSB database).

Algorithms		R-Precision [%]		
		KLD	L1	COS
BF-SSIFT	None	75.7	64.6	70.3
	MR	84.4	78.2	79.9
	DA-MR	*86.8	83.7	79.9
BF-DSIFT	None	75.4	75.9	74.5
	MR	75.5	76.1	75.2
	DA-MR	*90.7	88.4	84.7
VM-1SIFT	None	48.2	43.3	41.7
	MR	42.7	47.2	46.0
	DA-MR	*54.9	49.9	48.1
SHD	None	48.4	55.6	51.3
	MR	55.5	*61.6	58.5
	DA-MR	54.7	*61.4	57.5
SPRH	None	51.1	53.0	57.3
	MR	58.9	56.8	63.6
	DA-MR	63.3	63.5	*65.2

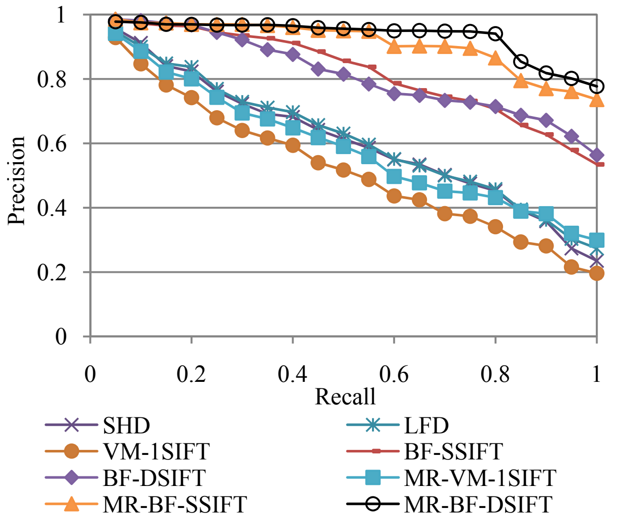


Figure 6. Recall-precision plot for the MSB database.

3.2 Feature Combinations

We tried several combinations of (1) *bag of salient local visual feature*, (2) *bag of dense and random local visual feature*, and (3) *global visual feature*, to see if they improve retrieval performance, and to see if they improve robustness of the retrieval algorithm against the composition of databases. For various combinations, Table 4 shows retrieval performances for the PSB, and Table 5 shows the retrieval performances for the MSB. For comparison, we included performance figures for the LFD [1] and the Spherical Harmonics Descriptor [6].

For the rigid shapes of the PSB, the combination of VM-1SIFT with BF-DSIFT performed the best, although the combination of VM-1SIFT, BF-DSIFT, and BF-SSIFT performed about as well. Note that the VM-1SIFT outperformed both LFD and SHD for the PSB, an indication of the power of the SIFT. For the articulated shapes of the MSB, as expected, combinations of two local features, "BF-SSIFT + BF-DSIFT", did the best. Note, however, that the combination "BF-SSIFT + BF-DSIFT + VM-1SIFT" performed nearly as well. In another word, adding VM-1SIFT to the mix of "BF-SSIFT + BF-DSIFT" did not hurt much. The combination of all the three features, including the global one, may be a good bet for a database of an unknown composition.

Table 4. Feature combination and retrieval performance with fixed and DA-MR distances (PSB benchmark).

Feature combinations	Ranking method	
	Fixed	DA-MR
LFD (original)	44.7	
SHD (original)	39.6	
BF-SSIFT	44.8	50.3
BF-DSIFT	54.1	60.4
VM-1SIFT	50.9	56.5
BF-SSIFT + BF-DSIFT	53.8	59.9
BF-SSIFT + VM-1SIFT	54.5	59.3
BF-DSIFT + VM-1SIFT	57.6	* 64.0
BF-SSIFT + BF-DSIFT + VM-1SIFT	57.1	* 63.6

Table 5. Feature combination and retrieval performance with fixed and DA-MR distances (MSB benchmark).

Feature combinations	Ranking method	
	Fixed	DA-MR
LFD (original)	55.5	
SHD (original)	55.6	
BF-SSIFT	75.7	86.8
BF-DSIFT	75.4	90.7
VM-1SIFT	48.2	54.9
BF-SSIFT + BF-DSIFT	76.1	* 91.9
BF-SSIFT + VM-1SIFT	69.0	82.0
BF-DSIFT + VM-1SIFT	70.8	88.0
BF-SSIFT + BF-DSIFT + VM-1SIFT	73.8	* 90.3

4. SUMMARY AND CONCLUSION

In this paper, we presented a 3D model retrieval algorithm that combines the BF-DSIFT algorithm [3] with a data-adaptive distance computation using the *Manifold Ranking* (*MR*) [15]. In doing so, we discovered that distance distribution produced by the BF-DSIFT made the MR less effective. We thus proposed an empirical remedy called *Distance Adjustment* (*DA*) to be applied prior to the MR. Our experimental evaluation showed that the DA does improve efficacy of the MR for certain features. The original MR was not effective at all if applied directly to BF-DSIFT features. However, the MR with DA was effective for such features. With the DA-MR, for example, the retrieval performance of the BF-DSIFT for the articulated models of the McGill Shape Benchmark [14] improved from 76% to 91% in R-Precision.

We had successfully employed the BF-DSIFT with DA-MR to enter the SHREC 2010 tracks on "Non-rigid shapes" and "Generic 3D Warehouse" for the 1st place and the 1st place tie, respectively. Using the DA-MR BF-DSIFT, a query for the SHREC 2010 Generic Warehouse is processed in about 2s, including the multi-view rendering, SIFT feature extraction, BoF integration, and DA-MR based ranking of the entire models in the database.

We also evaluated retrieval performances of combinations of local and global features. We found that combination of distances generated by three different features do improve robustness of the retrieval algorithm against composition of databases, with almost negligible impact on retrieval performance. Compared to the method we used for the SHREC 2010 tracks, the DA-MR BF-DSIFT, combination of features performed better on the PSB and about equal for the MSB.

5. ACKNOWLEDGMENTS

The authors would like to thank those who carefully reviewed the paper. This research has been funded in part by the *Ministry of Education, Culture, Sports, Sciences, and Technology of Japan* (No. 18300068).

6. REFERENCES

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

[2] G. Csurka, C.R. Dance, L. Fan, J. Willamowski, C. Bray, Visual Categorization with Bags of Keypoints, *Proc. ECCV '04 workshop on Statistical Learning in Computer Vision*, 59-74, (2004).

[3] T. Furuya, R. Ohbuchi, Dense sampling and fast encoding for 3D model retrieval using bag-of-visual features, *Proc. ACM CIVR 2009*, (2009).

[4] P. Guerts, D. Ernst, L. Wehenkel, Extremely randomized trees, *Machine Learning*, 2006, **36**(1), 3-42, (2006).

[5] M. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, K. Ramani, Three Dimensional Shape Searching: State-of-the-art Review and Future Trends, *CAD,* **5**(15), 509-530, (2005).

[6] M. Kazhdan, T. Funkhouser, S. Rusinkiewicz, Rotation Invariant Spherical Harmonics Representation of 3D Shape Descriptors, *Proc. Symposium of Geometry Processing* (*SGP*) 2003, 167-175 (2003).

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

[8] R. Ohbuchi, K. Osada, T. Furuya, T. Banno, Salient local visual features for shape-based 3D model retrieval, *Proc. SMI* '08, 93-102, (2008).

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

[10] J. Sivic, A. Zisserman, Video Google: A text retrieval approach to object matching in Videos, *Proc. ICCV 2003,* Vol. 2, 1470-1477, (2003).

[11] J.W.H. Tangelder, R. C. Veltkamp: A survey of content based 3D shape retrieval methods. *Multimedia Tools and Applications*. **39**(3), 441-471 (2008).

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

[13] J. Winn, A. Criminisi, T. Minka, Object categorization by learned universal visual dictionary, *Proc. ICCV05*, Vol. II, 1800-1807, (2005).

[14] J. Zhang, R. Kaplow, R. Chen, K. Siddiqi, The McGill Shape Benchmark (2005). http://www.cim.mcgill.ca/shape/benchMark/

[15] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Schölkopf, Learning with Local and Global Consistency, *Proc. NIPS 2003* (2003).