Eurographics Workshop on 3D Object Retrieval (2010)
I. Pratikakis, M. Spagnuolo, T. Theoharis, and R. Veltkamp (Editors)

SHREC'10 Track: Generic 3D Warehouse

T. P. Vanamali[†,1], A. Godil[†,1], H. Dutagaci[†,1], T. Furuya[2], Z. Lian[1,3], R. Ohbuchi[2]

[1]National Institute of Standards and Technology, USA,
[2]University of Yamanashi, Japan
[3]Beihang University, Beijing, PR China

Abstract

In this paper we present the results of the 3D Shape Retrieval Contest 2010 (SHREC'10) track Generic 3D Warehouse. The aim of this track was to evaluate the performances of various 3D shape retrieval algorithms on a large Generic benchmark based on the Google 3D Warehouse. We hope that the benchmark developed at NIST will provide valuable contributions to the 3D shape retrieval community. Three groups have participated in the track and they have submitted 7 set of results based on different methods and parameters. We also ran two standard algorithms on the track dataset. The performance evaluation of this track is based on six different metrics.

Categories and Subject Descriptors (according to ACM CCS): I.5.4 [Pattern Recognition]: Computer Vision—, H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—

1. Introduction

With the increasing number of 3D models that are created everyday and stored in databases, the development of effective 3D shape-based retrieval algorithms has become an important area of research. Benchmarking allows researchers to evaluate the quality of results of different 3D shape retrieval approaches. Here, we propose a new publicly available large benchmark called Generic 3D Warehouse with 3168 models, based on the Google 3D Warehouse to advance the state of art in 3D shape retrieval. The motivation behind the creation of this Generic benchmark is to have larger number of 3D models in the benchmark to challenge the 3D Shape Retrieval research community to be able handle, process and calculate feature vectors and distance measure in short time. Also presumably the reliability of the performance evaluation of algorithm increases with the increase in number of models in benchmark. The remainder of the paper is organized as follows. Section 2 describes the track dataset and how it was created. Section 3 discusses performance evaluation methods used to compare the results. Section 4 discusses the participant's information. Afterwards, Section 5 discusses briefly the different methods tested in this track and Section 6 describes the evaluation results. Finally, the conclusion of this paper is provided in Section 7.

2. Dataset

All the 3D models in the Generic 3D Warehouse track were acquired by a web crawler from the Google 3D Warehouse. The database consists of 3168 3D objects categorized into 43 categories. The number of objects in each category is not same and varies between 11 and 177. This dataset is a collection of 3D models downloaded from Google 3D Warehouse [1] which is an online collection of 3D models and 3D scenes created in sketchup format (*.skp) format using Google Sketchup [2] or converted from other 3D databases. which have been modified and converted into ASCII Object File Format(*.off). The categories of objects are given in Table 1. Even though the models have been retrieved from a single source of google 3d warehouse, it has still a varied type of models as they have been made and uploaded by the enthusiastic 3D model designers all across the world. There were quite a few complications in making the dataset because many models were repetitive and most of the time it was not a single object in the 3d model but rather a scene with more than one related object. Also many of the models were found to be corrupt. For every shape we made a list of keywords which different people might have used in the pro-

[†] Organizer of this SHREC track.

DOI: 10.2312/3DOR/3DOR10/093-100

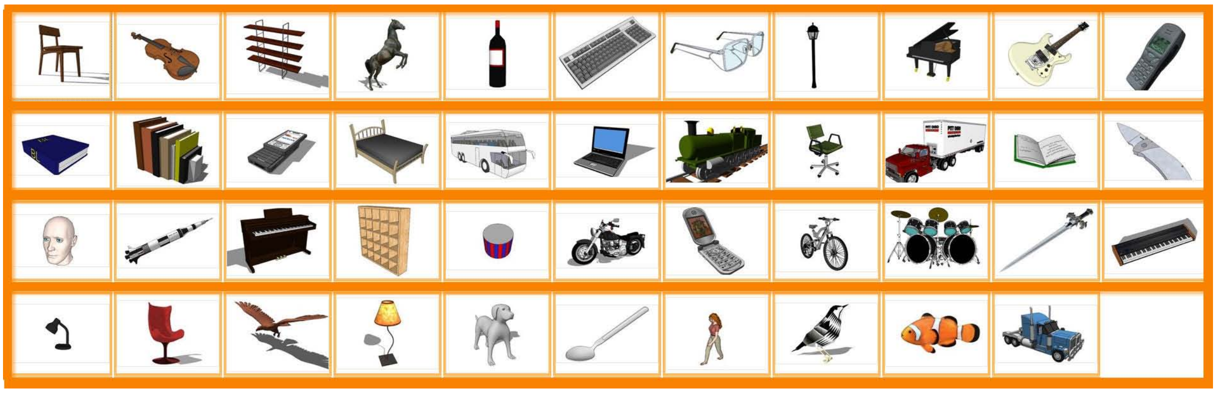

Figure 1: *One example image from each class of the Generic 3D Warehouse is shown.*

cess of uploading the models. For every key word we have downloaded 683 .skp models on a maximum (if available) spanning across 57 search result pages of Google 3D Warehouse. As sketchup program was mostly used to generate a multi-object models rather than a single object model many objects were repeating themselves across various skp files. Also almost every model was part of a scene with other objects and had a background. Then after manually verifying each of the downloaded file and removing the unnecessary content, we converted them to Object File Format (OFF).

The sub classification of each class was done based on the shape forms of the models that were downloaded by keywords, like a book model has been sub classified as openbook and closed-book and book-set.

The classes are defined with respect to their semantic categories and are listed in Table 1.

Both Google and end users have a non-exclusive, perpetual, worldwide license to use any content uploaded under the terms of the Term of Service (ToS). So, we as the end user can use the content in any manner described in the ToS, including: to reproduce the content; to create and reproduce derivative works of the Content; to display publicly and distribute copies of the Content; to display publicly and distribute copies of derivative works of the content.

3. Performance Evaluation

The evaluation of the shape retrieval contest is based on standard metrics. Using the dissimilarity matrix the following evaluation measures are calculated: 1) Nearest Neighbor (NN), 2) First Tier (FT), 3) Second Tier (ST), 4) E-measure (E), and 5) Discounted Cumulative Gain (DCG) [SMKF04]. In addition to these scalar performance measures, the precision-recall curves are also obtained.

Bed	Bicycle	Bookshelf	Bottle
Bus	Cat	Fish	Guiter
Head	Horse	Keyboard	Knife
Motorbike	Rocket	Spectacles	Spoon
Sword	Train	Violin	Woman
FlyingBird	PerchedBird	BookSet	OpenBook
SingleBook	FlipCellphone	NormalCellphone	SliderCellphone
4LeggedChair	0LeggedChair	WheeledChair	DrumSingle
DrumSet	BedLamp	FloorLamp	StudyLamp
ClosedLaptop	OpenLaptop	ClosedPiano	OpenPiano
PianoBoard	ContainerTruck	NoncontainerTruck	

Table 1: *43 classes of the target database.*

4. Participants

There are three groups that participated in the Contest and submitted 7 dissimilarity matrices. The groups are: R. Ohbuchi and T. Furuya with a Methods, Bag-of Densely-Sampled Local Visual Features for 3D Model Retrieval; Z. Lian and A. Godil with View based 3D Shape Descriptor Using Local and Global Features; and, H. Dutagaci and A.Godil with View based PCA. We also ran the dataset on two of the standard retrieval methods, LightField Descriptor by Chen et al.[CTSO03] and Hybrid method by Vranic [Vra05].

5. Methods

Brief descriptions of the methods are provided in the following subsections.

5.1. Bag-of Densely-Sampled Local Visual Features for 3D Model Retrieval by R. Ohbuchi and T. Furuya

The algorithm compares 3D shapes based on their appearances i.e., range images of the object rendered from multiple viewpoints. The algorithm is designed so that it could handle (1) diverse range of shape representations, including polygon soup, point set, or B-rep solid (2) models having articulation or deformation.

Appearance based comparison gives the algorithm its

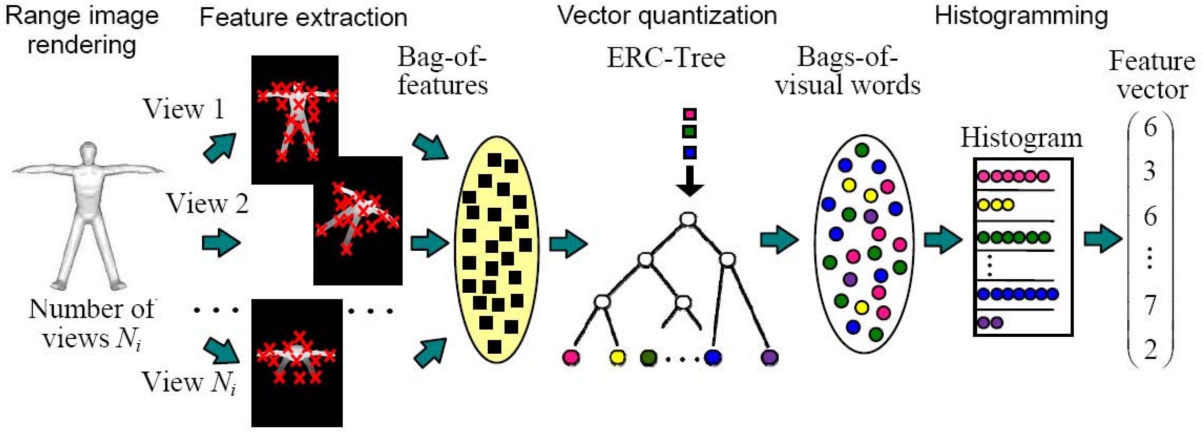

Figure 2: *Flow of the Bag-of-Feature Dense-SIFT with ERC-Tree* (BF-DSIFT-E) *algorithm.*

ability to handle diverse shape representation i.e., multiple-viewpoint rendering of dozens of range images coupled with (2D) rotation invariant image feature gives the algorithm its rotation invariance. Invariance to articulation and/or global deformation is achieved through the use of a set of multi-scale, local, visual features integrated into a feature vector per 3D model by using *Bag-of-Features (BoF)* approach. A feature vector per 3D model makes the cost of comparison among a pair of 3D models much lower than comparing sets of features, each consisting of thousands of local features. The algorithm is called *Bag-of- Features Dense-SIFT with Extremely randomized tree (BF-DSIFT-E)*. Please refer to the paper by Furuya et al. [FO09] for the details. While most 3D model retrieval algorithms deal with invariance to similarity transformation, very few algorithms achieve invariance to articulation while being able to accept 3D models having diverse set of 3D shape representations.

Figure 2 shows the processing flow of the BF-DSIFT- E algorithm. After normalizing the model for its position and scale, dozens of range images of the model are generated by using multiple virtual cameras looking inward at the model sitting at the coordinate origin. From each range image, our algorithm densely and randomly samples a few hundreds of local, multiscale image feature *Scale Invariant Feature Transform (SIFT)* by David Lowe [Low04]. Salient point detector of the SIFT is disabled for the dense sampling. A SIFT feature, typically having 128 dimensions, encodes position, orientation, and scale of gray-scale gradient change of the image about the sample point.

A 3D model is rendered into 42 depth images, each one of which then is sampled at 300 or so random locations. Thus, a 3D model is described by a set of about 12k SIFT features. The set of thousands of visual features is integrated into a feature vector per model by using BoF (e.g., [CDF*04] [SZ03]). The BoF approach vector quantizes, or encodes, a SIFT feature into a representative vector, or a "visual word", using a previously learned codebook. Visual words are accumulated into a histogram, which is the feature vector for the 3D model. The optimal dimension of the histogram, that is, the dimension of BoF-integrated feature vector, depends on the database. The dimension of feature vector is experimentally chosen as about 30k for the non-rigid model.

To extract this many features quickly, a fast GPU-based implementation of SIFT algorithm called SiftGPU by Wu [Wu] is applied. The Extremely Randomized Clustering Tree, or ERC-Tree, by Guerts et al. [GEW06], is also used for both feature set clustering during codebook learning and for vector quantization of SIFT features. With a small penalty in retrieval performance, the ERC-Tree is much faster than k-means clustering during codebook learning and naive linear search during VQ.

To derive ranked list of retrieval results given a query, two methods are employed: simple distance computation using Kullback-Leibler Divergence (KLD), and a distance-metric learning approach called Manifold Ranking (MR) [ZWG*03] with a small modification. In the retrieval experiment, the version that used KLD is named BF-DSIFT-E, while the one used MR is named MR-BF-DSIFT-E. The KLD below performs well for comparing two histogram-based feature vectors x and y.

$$d_{KLD}(x,y) = \sum_{i=1}^{n} (y_i - x_i) ln \frac{y_i}{x_i}$$

The MR first estimates the distribution of features in a low-dimensional subspace, or "manifold" approximated by a mesh. The mesh is created based on the proximity of feature points, and its edges are weighted based on the distance among the feature points. It then computes similarity among the features on the manifold using a process similarity solving a diffusion equation. A relevance rank is diffused from the source, that is, the query. At an equilibrium state, the concentration of the relevance rank at a feature point indicates its closeness to the query. KLD is utilized to form the manifold mesh on which the diffusion takes place.

5.2. VLGD: View based 3D Shape Descriptor Using Local and Global Features by Z. Lian and A. Godil

The visual similarity based method has been widely considered as the most discriminative approach in the field of content-based 3D object retrieval. We developed three such kind of methods, denoted as CM-BOF, GSMD, and VLGD, respectively, for the Warehouse Benchmark of SHREC'10. Basically, these two algorithms are similar due to the fact that, they all utilize a particular visual similarity based framework, and the only difference between them is how to describe the depth-buffer views captured around a 3D object. More specifically, CM-BOF uses a local feature based shape descriptor to represent a view as a histogram, and GSMD describes the view by a global feature vector, while VLGD utilizes a linear combination of above mentioned 2D shape descriptors. Finally, a *Modified Manifold Ranking* (MMR) method is applied to try to further improve the retrieval performance of the VLGD method.

5.2.1. A Visual Similarity based Framework

As demonstrated in Figure 3, our visual similarity based 3D shape retrieval framework is implemented subsequently in four steps:

1. ***Pose Normalization:*** Given a 3D object, we first translate the center of its mass to the origin of canonical coordinate frame and then scale the maximum polar distance of the points on the surface to one. Rotation invariance is achieved by applying the PCA technique to find the principal axes and align them to the canonical coordinate frame. Note that, we only employ the information of eigenvectors to fix the positions of three principal axes, namely, the direction of each axis is still undecided and the x-axis, y-axis, z-axis of the canonical coordinate system can be located in all three axes. That means 24 different orientations are still plausible for the normalized 3D object, or rather, 24 matching operations should be carried out when comparing two models. It should also be pointed out that, the exact values of the surface moments used in our PCA-based pose normalization are calculated via the explicit formulae introduced by [ST01].
2. ***View Rendering:*** After pose normalization, 66 depth-buffer views with size 256×256 are captured on the vertices of a given unit geodesic sphere whose mass center is also located in the origin, such that a 3D model can be represented by a set of images. We render the views base on OpenGL.
3. ***Feature Extraction:*** For each view, a specific image processing technique is applied to represent the view as a compact feature vector. Based on the different 2D shape descriptors used (see corresponding subsections), our algorithms are classified as the following three categories: local feature based (i.e. CM-BOF), global feature based (i.e. GSMD), and composite (i.e. VLGD) methods.
4. ***Dissimilarity Calculation:*** The last step of our framework is the dissimilarity calculation for two shape descriptors. The basic idea is that, after we get the principal axes of an object, instead of completely solving the problem of fixing the exact positions and directions of these three axes to the canonical coordinate frame, all possible poses are taken into account during the shape matching stage.
   The dissimilarity between the query model q and the source model s is defined as,
   $$Dis_{q,s} = \min_{0 \le i \le 23} \sum_{k=0}^{65} D\left(FV_q(p_0'(k)), FV_s(p_i'(k))\right),$$
   where $FV_m = \{FV_m(k) | 0 \le k \le 65\}$ denotes the shape descriptor of 3D object m, $FV_m(k)$ stands for the feature vector of view k, the permutations $p_i' = \{p_i'(k) | 0 \le k \le 65\}$, $0 \le i \le 23$ indicate the arrangements of views for all (24) possible poses of a normalized model, and $D(\cdot,\cdot)$ measures the dissimilarity between two views. For more details about this multi-view shape matching scheme, we refer the reader to our previous papers [LRS] [LGS].

5.2.2. A Local feature based Method: CM-BOF

In our CM-BOF algorithm, each view is described as a word histogram obtained by the vector quantization of the view's salient local features, and the distance between two histograms H_1, H_2 with N_w bins is evaluated by the formula,

$$D(H_1, H_2) = 1 - \frac{\sum_{j=0}^{N_w-1} \min(H_1(j), H_2(j))}{\max(\sum_{j=0}^{N_w-1} H_1(j), \sum_{j=0}^{N_w-1} H_2(j))}.$$

Note that, the 2D salient local feature is calculated using the *VLFeat* matlab source code developed by Vedaldi and Fulkerson [VF]. On average, this 3D shape descriptor contains about 2618 integers when $N_w = 1500$.

5.2.3. A Global feature based Method: GSMD

In our GSMD algorithm, each view is represented as a global feature vector with 47 elements including 35 Zernike moments, 10 Fourier coefficients, eccentricity and compactness, and the dissimilarity between two feature vectors is measured by their L_1 difference.

Note that, the global feature vector is calculated using the C++ source code provided by [DYM03], and the vector is normalized to its unit L_1 norm. The dimension of this 3D shape descriptor is 3102.

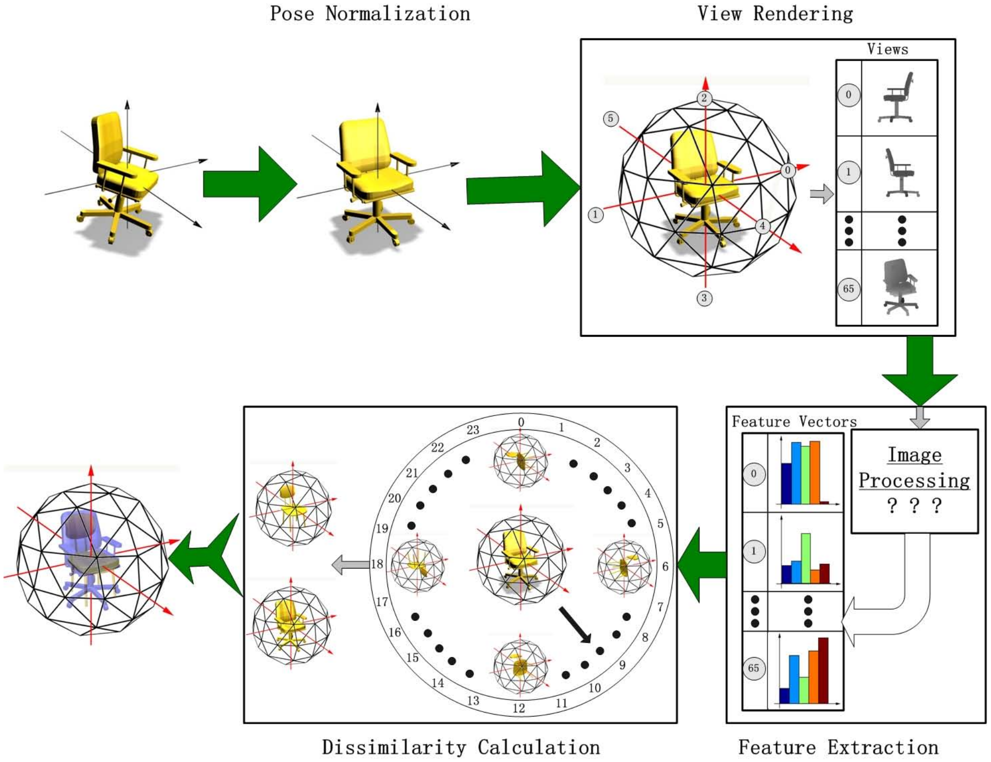

Figure 3: *The uniform framework of our methods.*

5.2.4. A Composite Method: VLGD

Our VLGD algorithm is a composite method based on a linear combination of CM-BOF and GSMD. More specifically, in this method, a view is expressed by a feature vector consisting of two kinds of shape descriptors, which are used in CM-BOF and GSMD, with pre-specified weights. We experimentally select the weights as $W_{local} = 7.0$ and $W_{global} = 1.0$ for local and global features, respectively, by maximizing the retrieval accuracy on PSB train set with base classification.

5.2.5. A Modified Manifold Ranking: MMR

Given a query model M_0 and a database with N target models, the initial retrieval result can be derived by sorting the dissimilarities between the query and other models in ascending order. The sorted data set can be denoted as $DM = \{M_0, M_1, M_2, ..., M_N\}$. We observe that the manifold ranking method (originally presented in [ZWG*03]) can further improve the 3D shape retrieval performance. In order to make the algorithm more effective and efficient for our application, we slightly modify the original approach in three aspects:

1. Instead of iterating to create a connected graph, our manifold is formed by connecting edges between the models and their N_k nearest neighbors. This modification not only accelerates the graph construction but also ensures enough connections for each model.
2. Instead of directly using the distance of every pair of models to compute the weights of the edges, we assign weight values to the edges according to their corresponding initial ranking order with respect to the reference model. That means the graph is normalized and an edge may have different weights in its two directions.
3. Instead of just considering the query model as labeled data, we also label the nearest neighbor as the query. It is reasonable and actually effective because the initial overall successful rate of best matches is above 90%.

Let the ranking scores of the data set DM be denoted by $S = \{s_0, s_1, s_2, ..., s_N\}$, where $s_i(t)$ is the score after t iterations and $s_i(0) = 1$ if $i \in \{0, 1\}$, else $s_i(0) = 0$. We also compute a matrix $X = [x_{ij}]_{(N+1)\times(N+1)}$ in which x_{ij} stands for the position of M_j in a retrieval ranking list given that the query model is M_i. For example, $x_{ii} = 0$ and $x_{ij} = 1$ when M_j is the nearest neighbor of M_i. Our modified manifold ranking method performs in the following five steps:

1. Create edges between the models and their N_k nearest neighbors to form a graph $G = (M, E)$.

2. Define the matrix K computed by $k_{ij} = \exp(-||x_{ij}||^2/\sigma)$ if there is an edge $e(i,j) \in E$ or else $k_{ij} = 0$.
3. Calculate the matrix $L = D^{-1}K$where D is the diagonal matrix with $d_{ii} = \sum_{j=0}^{N} k_{ij}$.
4. Iterate $s_i(t+1) = \sum_{j=0}^{1} l_{ij} s_j(0) + \alpha \sum_{j=2}^{N} l_{ij} s_j(t)$, where $\alpha \in [0,1]$ and $t \leq N_r$, for all unlabeled models $(1 < i \leq N)$.
5. Denote the iteration result $s_i(N_r)$ by s_i^* which is considered as the final ranking score of the model M_i. Then output the ranking list by sorting $s_i^* (0 < i \leq N)$ in descending order.

Note that the parameters are experimentally selected as $\alpha = 0.995$, $\sigma = 32$, $N_k = 100$ and $N_r = 300$.

5.3. Principal Component Analysis of Depth Images by H. Dutagaci and A.Godil

One of the methods in this track is based on Principal Component Analysis of depth images. PCA is used as a data driven subspace model to describe the shape information of the views. The depth images obtained from various points of the view sphere are treated as 2D intensity images. A PCA-based subspace is trained to extract the inherent structure of the views within a database.

5.3.1. View sampling

Prior to view sampling, a 3D triangular mesh is rotated into a reference frame via CPCA pose normalization, which was proposed by [Vra05]. Then the views are obtained using the vertices of a geodesic sphere surrounding the model. A regular octahedron is subdivided to obtain finer geodesic spheres with more vertices. In this competition, a geodesic sphere with 18 vertices is used for view sampling. The depth images are captured as seen from the vertices of these two spheres, then they are mapped onto a regular grid of resolution 128x128. Figure 4 shows two examples of the view extraction process.

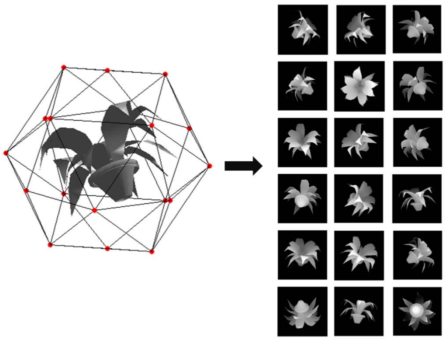

Figure 4: *View sampling.*

5.3.2. Principal Component Analysis

Subspace methods find a set of vectors that describe the significant statistical variations among the observations. These vectors form the basis of a subspace where most of the meaningful information for certain class of processes is preserved. These methods have the additional advantage of greatly reducing the dimensionality of the observations. In this specific application, the observations are the depth images generated from the complete 3D models.

A data matrix X is constructed by collecting N depth images from a training set, converting them to one dimensional vectors of length M, and putting these vectors $\{x_1, x_2, \cdots x_N\}$ into the columns of the data matrix X. Then, PCA is applied to the data matrix.

PCA is an analysis technique that is based on the decorrelation of the data using second order statistics. The eigenvectors of the $M \times M$ covariance matrix, $G = (X - \bar{X})(X - \bar{X})^T$ gives the principal directions of variations. Here, $\bar{X}$ denotes the mean of the training vectors. Let $\{v_1, v_2, \cdots v_K\}$ be the first K eigenvectors of G with corresponding eigenvalues $\{\alpha_1 \geq \alpha_2 \ldots \alpha_K\}$. These vectors model the largest variations among the training samples, therefore are considered to capture most of the significant information. The amount of information maintained depends on K and the spread of eigenvalues. The projection of an input vector x onto the PCA subspace is given by $b = V^T x$, where V represents the $M \times K$ projection matrix formed as $[v_1, v_2, \cdots v_K]$.

A separate data set other than the target dataset provided by SHREC track organizers were used to extract the PCA subspace. The data matrix X is formed by collecting six canonical views of each of the 907 training models in the Princeton Shape Benchmark (PSB) [2], so there are 6 907 observations to be analyzed. Figure 2 shows the first four principal modes of variations among the depth images of PSB training models.

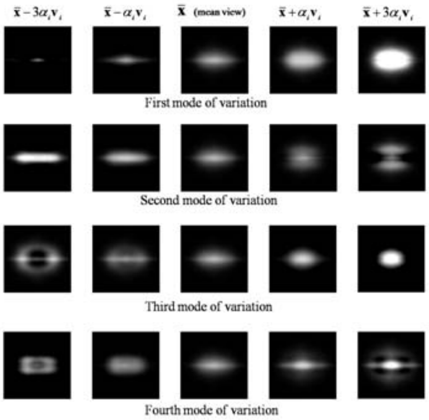

Figure 5: *View sampling.*

5.3.3. Axis rotations and reflections

One problematic issue with the CPCA normalization is the ambiguity of axis ordering and reflections. Most of the misalignment errors are due to inconsistent within-class axis orderings and orientations given by the normalization procedure. These axis ordering and reflection ambiguities are resolved by reordering and reconfiguring the views of the model according to the 48 possible reflections and orientations of the objects. Then subspace features are extracted for each of the 48 sets of depth images. The feautures of a query model are then compared to each of the 48 sets of feature vectors and the one giving the minimum distance is chosen.

6. Results

In this section, we present the performance evaluation results of the SHREC'10 Generic 3D Warehouse track. Three research groups participated in the contest and submitted 7 dissimilarity matrices obtained by different methods. In addition to the methods of the participants we tested 5 state-of-the-art methods on our benchmark and obtained the corresponding distance matrices. The dissimilarity matrices were analyzed based on the following evaluation measures: Nearest Neighbor (NN), First-Tier (FT), Second-Tier (ST), E-Measure (EM), Discounted Cumulative Gain (DCG) and Precision Recall Curve.

Table 2 shows the retrieval statistics yielded by the methods of the participants, four previous methods proposed by Vranic [Vra04] and the Light Field Descriptor (LFD) [DYM03]. All the methods except the view-based PCA gave better results than all of the Vranic's methods and the LFD. The high retrieval results suggest that emerging 3D retrieval algorithms are capable of handling databases containing thousands of models.

PARTICIPANT	METHOD	NN	FT	ST	E	DCG
Ohbuchi	BF-DSIFT-E	0.884	0.531	0.668	0.360	0.841
	MR-BF-DSIFT-E	0.897	0.606	0.733	0.389	0.869
Dutagaci	View-based PCA - 18 view	0.825	0.433	0.557	0.314	0.789
Lian	GSMD	0.875	0.491	0.624	0.344	0.824
	CM-BOF	0.862	0.534	0.662	0.358	0.836
	VLGD	0.889	0.565	0.696	0.377	0.855
	VLGD+MMR	0.889	0.647	0.791	0.390	0.880
AUTHOR	PREVIOUS METHODS	NN	FT	ST	E	DCG
Vranic	DSR472 with L1	0.871	0.498	0.639	0.356	0.831
	DBD438 with L1	0.809	0.407	0.532	0.306	0.770
	SIL300 with L1	0.807	0.412	0.548	0.300	0.780
	RSH136 with L1	0.783	0.385	0.508	0.275	0.758
Chen	LFD	0.864	0.48	0.613	0.336	0.816

Table 2: *The retrieval statistics for all the methods and runs.*

We have selected the best runs of each participant and displayed them in Figure 6, which shows their performances results in a bar graph. Figure 7 gives the precision-recall curves of the all methods. Observing these figures, we can state that Lian's VLGD+MMR method yielded highest results in terms of all the measures but Nearest Neighbor. In terms of Nearest Neighbor, Ohbuchi's MR-BF-DSIFT-E method performed best.

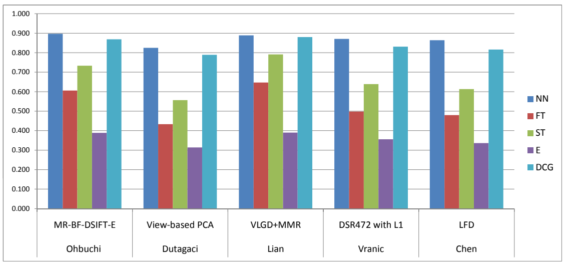

Figure 6: *Bar plot of the Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), E-measure(E) and Discounted Cumulative Gain (DCG) for the best runs of each participant. We also include the results of LFD descriptor of Chen and the DSR472 descriptor of Vranic.*

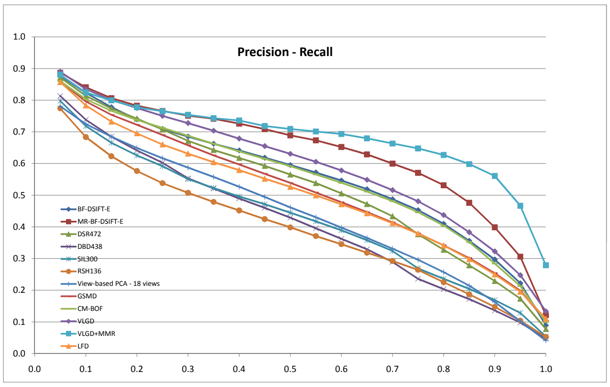

Figure 7: *Precision-recall curves of the best runs of each participant.*

The view-based PCA didn't perform as well as other participants' methods. One main reason is that the PCA basis was trained on a different dataset (the PSB training set) other than the target database; therefore the shape structure of the target database is not well-represented by the PCA basis. The participant didn't choose to train the PCA basis using the target database, since it would require to conduct long leave-one-out experiments where the query object is left out and the PCA basis is trained using the rest of the objects.

The GSMD method is based on global descriptors of views whereas the CM-BOF of Lian and SIFT-based descriptors of Ohbuchi use bag of words approach to desribe local features of the views. We can consider these four methods and the view-based PCA approach as individual methods. Among these individual methods, Ohbuchi's MR-BF-DSIFT-E method performed better. Lian's VLGD method is a combination of the GSMD and CM-BOF; i.e. it is a hybrid method that fuses local and global characteristics of the object. The results in Table 2 show that the hybrid method performed better than the individual methods of Lian. It is also apparent that application of the Non-Manifold Ranking to the hybrid method boosted the performance.

7. Conclusions

In this paper, we have described and compared the performance of different algorithms submitted by three research groups that participated in this track. The participants have submitted 7 sets of dissimilarity matrices in total, based on different methods and parameters. We hope that the track database will provide valuable contributions to the 3D shape retrieval community.

References

[CDF*04] Csurka G., Dance C. R., Fan L., Willamowski J., Bray C.: Visual categorization with bags of keypoints. In *Proc. ECCV'04 workshop on Statistical Learning in Computer Vision* (2004), pp. 59–74. 3

[DYM03] D.-Y.Chen X.-P.Tian Y.-T., M.Ouhyoung: On visual similarity based 3D model retrieval. In *Computer Graphics Forum* (2003), vol. 22, pp. 223–232. 4, 7

[FO09] Furuya T., Ohbuchi R.: Dense sampling and fast encoding for 3d model retrieval using bag-of-visual features. In *Proc. ACM CIVR 2009* (2009). 3

[GEW06] Guerts P., Ernst D., Wehenkel L.: Extremely randomized trees. *Machine Learning 36*, 1 (2006), 3–42. 3

[LGS] Lian Z., Godil A., Sun X.: Visual similarity based 3D shape retrieval using bag-of-features. *(in review)*. 4

[Low04] Lowe D. G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110. 3

[LRS] Lian Z., Rosin P. L., Sun X.: Rectilinearity of 3D meshes. *International Journal of Computer Vision (in press)*. 4

[SMKF04] Shilane P., Min P., Kazhdan M., Funkhouser T.: The princeton shape benchmark. In *Shape Modeling International* (2004). 2

[ST01] Sheynin S. A., Tuzikov A. V.: Explicit formulae for polyhedra moments. *Pattern Recognition Letters 22* (2001), 1103–1109. 4

[SZ03] Sivic J., Zisserman A.: Video google: A text retrieval approach to object matching in videos. In *Proc. ICCV 2003* (2003), vol. 2, pp. 1470–1477. 3

[VF] Vedaldi A., Fulkerson B.: VLFeat: An open and portable library of computer vision algorithms. *http://www.vlfeat.org/*. 4

[Vra04] Vranic D. V.: *3D Model Retrieval*. PhD thesis, University of Leipzig, 2004. 7

[Vra05] Vranic D. V.: Desire: a composite 3D-shape descriptor. In *ICME* (2005), pp. 962–965. 6

[Wu] Wu C.: SiftGPU: A GPU implementation of david lowe's scale invariant feature transform (SIFT). *http://cs.unc.edu/ ccwu/siftgpu/*. 3

[ZWG*03] Zhou D., Weston J., Gretton A., Bousquet O., Scholkopf B.: Ranking on data manifolds. In *Proc. the Conference on Advances in Neural Information Processing Systems (NIPS)* (2003). 3, 5