Asian Journal of Information Technology 16 (6): 576-587, 2017
ISSN: 1682-3915
© Medwell Journals, 2017

A New View Based Algorithm for Non-Rigid 3D Object Retrieval

H. Ahmed, H. Shdeed, S. Hamad and D. Hegazy
Department of Scientific Computing, Faculty of Computers and Information Sciences,
Ain Shams University, 11566 Cairo, Egypt

Abstract: This study introduces an efficient view based algorithm for 3D non-rigid object retrieval that accepts all the models. The algorithm is based on describing the 3D Model using multi-scale local visual features. To reduce the cost of distance computation and features storage, the extracted features integrated into histogram using Bag of Features (BoF) approach. The proposed algorithm used only 42 depth images to extract the features which reduced the runtime. The proposed Enhanced Ray Tracing algorithm was used for generating best quality depth images with great reduction in runtime also. The codebook for the vector quantization is learned via. our proposed Modifed Extremely Randomized Clustering trees (MERC-trees). In comparison with other algorithms, the proposed algorithm achieved high performance on SHREC'11 and 15 datasets; the well-known benchmarks. Experimental results show that, the proposed algorithm is very fast, achieved high retrieval result and robust against 11 types of different transformations with different levels of strength when tested on SHREC'11-Robust dataset. The proposed algorithm achieved the 6th best performance among 37 methods participated in SHREC'15 contest which none of them was view based and all used topological based features that are easier to discriminate the non-rigid objects.

Key words: Content based retrieval, non-rigid 3D objects retrieval, SIFT, ERC trees, extracted, easier to discriminate

INTRODUCTION

The 3-Dimensional (3D) Models are an important multimeda data type. They have become ubiquitous for games running on mobile-phones and on game consoles for such web-based applications as the Google Earth for medical diagnostics and architectural design. In addition, to the increasing in the applications that create 3D Models as CAD, 3D Max, Maya, ..., etc. and the rapid development of 3D scanning technologies that produce a huge number of 3D Models, the need to organize these 3D Models for effective reuse has prompted research into shape-based retrieval of the 3D Models. So, the main issue has shifted from the creation of new 3D Models to the retrieval of existing 3D Models based on their content. The 3D object retrieval has become a challenging research area that the retrieval method must satisfy several requirements for invariance. A typical set of requirements includes invariance to similarity transformations with its different types and degrees. Invariance to shape representations as the 3D Model may be in different representation as point clouds, polygonal mesh, ..., etc. Invariance to geometrical and topological noise. Invariance to articulation or global deformation. Content-based 3D object retrieval based on the comparison of the geometry and topological properties of the 3D Models which based is complicated by the fact that many 3D models manifest rich variability especially in case of non-rigid (deformable) shapes. Non-rigid shapes include a wide range of shape transformations such as bending and articulated motion.

Up to now, a large number of algorithms for non-rigid 3D shape retrieval have been proposed. Each method of the 3D non-rigid retrieval methods has its own advantages and disadvantages. However, most retrieval approaches specifically designed for non-rigid 3D objects can only process watertight single-component meshes.

This study presents a view based retrieval method for non-rigid 3D objects which accepts different 3D representations so it suites for many different applications. The proposed method is based on rendering the 3D Model as a set of range images and for each range image a set of local, multi-scale, salient, visual features are extracted using Scale Invariant Feature Transform (SIFT) algorithm proposed by Lowe (2004). As each depth image yields a few dozen features and there are a few dozen range images per model, a 3D Model is associated with hundreds or even thousands of local features. The

Corresponding Author: H. Ahmed, Department of Scientific Computing, Faculty of Computers and Information Sciences, Ain Shams University, 11566 Cairo, Egypt

computation of dissimilarity between those sets will be expensive especially for large datasets which will make the search very slow process. The proposed method avoids the costly pair-wise distance computation by integrating all the local features of the model into a single feature vector by using the Bag-of-Features (BoF) approach.

The experimental results showed that the proposed algorithm achieved retrieval accuracy between 92 and 99% according to the different setting for SHREC'11 but for robust SHREC'11, 91% accuracy was achieved which reveals that this approach is highly discriminative for articulation and for different types of transformations and noise with different degrees. The retrieval accuracy on SHREC'15 was between 88 and 94%. The proposed algorithm compared with other 37 algorithms in SHREC'15 which are based on the topology or the geometry of the 3D Models, it was found that the proposed algorithm achieved the 6th best retrieval performance among those algorithms, although, the proposed one is view based and the topological and geometric properties algorithms are more discriminative than view based algorithms.

Literature review Non-rigid retrieval algorithms have great challenging criteria because they should have a reasonable computation complexity, robustness to arbitrary topological degeneracies and discriminate the same shapes in different postures. Non-rigid retrieval algorithms can be divided to algorithms employing: local features, topological structures, isometry invariant global geometric properties, direct shape matching and canonical forms.

The 1st division is to measure the dissimilarity between two models based on their local features that are insensitive to isometric transformations. The 2nd division used topological structures to compare deformable 3D objects (Hilaga *et al*., 2001; Reeb, 1946; Sundar *et al*., 2003).

For the 3rd division, isometric-invariant global geometric properties are used such as: geodesic distance which was used by Reuter *et al*., (2005) and the model's Laplace-Beltrami spectra which proposed to use by Jain and Zhang (2007).

The 4th division is to compare the mesh itself (direct shape matching) as by Memoli and Sapiro (2005). The last division is the canonical form. The utilization of canonical forms is also a promising solution for non-rigid 3D shape retrieval. The idea of generating canonical forms in 3D domain was initially proposed by Elad and Kimmel (2003).

The approach of local features is used in this research as it can be applied to rigid, non-rigid and partial 3D objects retrieval. Local features convey enough information to discriminate 3D objects and at same time they can handle different kinds of noise (Lian *et al*., 2010).

Local features algorithms can be divided into view based and model based approach. The advantage of view based approach that it can deal with 3D objects with different representations, more flexible, benefit from existing image processing technologies (Chen *et al*., 2003; Furuya and Ohbuchi, 2009; Ohbuchi *et al*., 2008). The proposed view-based method uses salient local features that are extracted using Scale Invariant Feature Transform (SIFT) algorithm (Lowe, 2004) which is one of the highest descriptors accuracy and invariant to scale and rotations (Mikolajczyk and Schmid, 2005).

According to our study BF-DSIFT (Furuya and Ohbuchi, 2009) and BF-SSIFT (Ohbuchi *et al*., 2008) are promising algorithms as they are view based methods which accepts different 3D representations, so they suite different applications and they perform well in SHREC'10 track for non-rigid 3D retrieval (Lian *et al*., 2010). BF-DSIFT and BF-SSIFT used SIFT descriptor (Lowe, 2004) which is one of the most famous descriptors in image recognition they have achieved high retrieval results. Moreover, the use of BoF paradigm made them compact from the storage point of view which makes them suitable for large detailed databases.

MATERIALS AND METHODS

The proposed approach: The proposed research is a based on the synergy between the two algorithms BF-DSIFT (Furuya and Ohbuchi, 2009) and BF-SSIFT (Ohbuchi *et al*., 2008) (Fig. 1). The algorithm is divided into 6 phases: model normalization, rendering the model as a set of depth images, extract SIFT features, extract training set and build visual codebook, vector quantization and finally, the matching phase.

Phase 1; Model normalization: In the 1st phase, pose normalization is performed which include position normalization followed by scale normalization. The position normalization is performed exactly as BF-SSIFT (Ohbuchi *et al*., 2008) in which the model centroid is computed using the quasi Monte-Carlo sampling of mass distribution on the surfaces of the model (Ohbuchi *et al*., 2008) which is inspired by Osada *et al.* (2002)'s method. The main problem of scale normalization is to find the scale value. In the proposed approach to find the scale value, the absolute values of the minimum and the maximum coordinates of x, y and z coordinates of the model must be found after performing the position

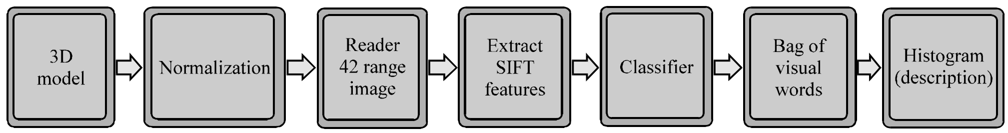

Fig. 1: The steps of the proposed algorithm

normalization process then the maximum absolute is chosen to be the scale value to guarantee that all the coordinates absolute are <1, so the model can be rendered inside unit sphere.

Phase 2; Multi-view rendering: In rendering the multi-view range images, the viewpoints are spaced evenly in the solid angle by placing them at vertices of regular or near regular polyhedrons enclosing the model. Viewpoints are placed at the vertices of an 80 face (42 vertices) semi regular polyhedron (Pentakis Icosidodecahedron) generated with the same methodology in BF-SSIFT (Ohbuchi *et al*., 2008). However, for generating or rendering the depth image itself the proposed enhanced Ray Tracing (enhanced-RT) presented by Ahmed *et al*. (2015) was used. Enhanced RT algorithm was based on sorting the mesh's triangles according to its minimum x coordinate and get the range of triangles that may intersect with each ray using binary search. The experimental results showed that the enhanced-RT decreased the execution time by 99.8% than the traditional ray tracing algorithm with high quality for the produced depth images for the standard benchmark models.

Phase 3; Extract SIFT features: After the range images are rendered, the SIFT algorithm (Lowe, 2004) is applied on the image to detect the interest points and then to compute the features at these interest points using SIFT descriptor. However, in the proposed algorithm, since, the interest points must lay on the objectwe neglect the key points on the image background, so the features become more discriminative to the model itself, this enhanced the performance of the algorithm used to build the visual codebook.

Phase 4; Extracting training set and build visual codebook

Extracting the training set: Extracting the training set is one of the most important phases. It affects the overall performance of the algorithm and specially the efficiency of the codebook. The efficiency of the visual codebook affects the representation of the model's SIFT features and hence the resultant descriptor of the model.

In the proposed algorithm, extracting the training set were done with specific percentage from the total features of each class. This percentage must be applied to all classes in the benchmark. The effect of using different percentage ratios of features from each class which will be used in the training process is studied and presented in the experiments and results discussions. The result show that, the performance enhanced with the increasing in the percentage ratio used.

Building the visual codebook: To build the visual codebook the training set is used to train the classifier (used as the input to the clustering algorithm). In the proposed approach, Extremely Randomized tree Clustering (ERC) algorithm was used as a classifier according to Moosmann *et al*. (2006) with proposed modifications to enhance the classifier accuracy. The proposed Modified ERC (MERC) is a supervised classifier in which discrete labels were assigned to the descriptors. The descriptors from the same class have the same label. The trees were trained using a labeled training set. At each node n, the space S will be divided into two children S_l and S_r which must satisfy the following conditions:

$$S = S_l \cup Sr$$
$$S_r \cap S_l = \emptyset$$

Recursion continues until further subdivision is impossible when one of the following occur:

- All surviving training examples belong to the same class
- All have identical values for all attributes
- $|S|<n_{min}$where n_{min} is the minimum sample size for splitting a node

When one of the previous conditions occurs the node will be a leaf node. In MERC algorithm, the leaf node is labeled by the class frequencies which represent how much this feature may occur in each class. This label will be considered a visual word.

The rest of the proposed MERC algorithm can be summarized as follow: to build an M ensemble tree from S, for each m where $1 \le m \le M$, do the following: select randomly K attributes, $\{attr_1, attr_2, ..., attr_K\}$ without

Table 1: The difference between the ERC and the proposed MERC Clustering algorithms

Parameters	ERC	MERC	Advantage of MERC
Attributes	Select K non constant random attributes	Select K non constant random attributes	
Splitting	Select random threshold according to the normal distribution of the attribute	Select T_r random thresholds according to the normal distribution of the attribute and select the one with the maximum score	This guarantees that the split will be done at the which maximum score will achieve best classification for this attribute
Leaf label	The class with maximum occurrence	Labeled by the class frequencies	Represents how much this feature may occur in each class
Histogram of the feature	The maximum occurrence in each tree of M	The summation of all class frequencies returned from the M ensemble trees	Represents the net result which gives accurate percentage of the occurrence of this feature at each class

replacement among all (non constant attributes in S) candidate attributes. For each attribute $attr_i$ in $\{attr_1, attr_2, ..., attr_K\}$. Generate split threshold s_i randomly according to the normal distribution of the $attr_i$ attribute. Calculate score for the split s_i according to attribute $attr_i$. Repeat the previous two steps T_r (number of tries) time and get the maximum score s* from the K attributes at each try of T_r. The T_r tries will guarantee that the split will be done at the maximum score which will achieve best classification for those attributes. Split S into subsets S_l and S_r according to the test s*. Build the left tree t_l using the subset S_l and build right tree t_r using the subset S_r by repeating the previous steps and replace S with S_l and S_r.

Create a node n with the split s*, attach t_l and t_r as left and right subtrees of this node and return the resulting tree t.

The MERC-tree parameters are set as follows: K = 128, n_{min} = 2, M = 10 and T_r = 50. The score of each split will be computes. The difference between the ERC and MERC and the advantages of MERC can be summarized in Table 1.

Phase 5; Vector quantization: After extracting the SIFT features from the training range images of each model. Each model will have a few hundreds features. This costs a very huge storage especially for large databases. Therefore, to save the storage the BoF is used to quantize the features of each 3D Model into a visual word using visual codebook which is constructed using the proposed MERC-trees (Hess, 2010).

The model features will be reduced in a histogram which will be used as the model descriptor. The histogram of each model was constructed by using the trained MERC that is traversed using every SIFT feature. Each SIFT feature is replaced with the frequencies of classes contained in the leaf that the feature retched to in each ensemble tree. The bin's number of the histogram equals the number of the visual words (vocabulary size) in the visual codebook. The histogram will be the accumulation of the frequencies and is used as the descriptor vector of the model. The histograms generated using MERC are more accurate than generated by the ERC as they represent the similarity percentage among the model and all the classes.

Phase 6; Matching phase: The two 3D Models can be compared by comparing their corresponding bags of features. Therefore, 3D object similarity problem is reduced to the problem of comparing vectors of feature frequency. Now for comparing two 3D objects, the bags of features are treated as the descriptors (histograms) such that the degree of similarity or dissimilarity can be determined through calculating the distance between their bags of features. Similar, 3D Models tend to have a small dissimilarity value or a large similarity value and vice versa.

In the model, we use Euclidean distance to compare among the bag of features:

$$\sqrt{\sum_{i=1}^{N_v}\left(x_i - y_i\right)^2} \tag{1}$$

Where:

N_v = The vocabulary size

x_i and y_i = The ith element of the descriptor vector

Experimental and evaluations: The experiments are designed to test the following:

- The effect of using different clustering algorithms for visual codebook learning
- The effect of using different vocabulary sizes
- The effect of using different training set sizes
- The effect of using different distance function to compare the BoFs
- The robustness of the retrieval algorithm

In order to evaluate, the proposed approach performance, the shape retrieval dataset SHREC'11 (Lian *et al*., 2011), SHREC'11-Robustness and SHREC'15 benchmarks are selected for applying the proposed algorithm on them, SHREC is an annual 3D shape retrieval contest, whose general objective is to evaluate the effectiveness of 3D-shape retrieval algorithms.

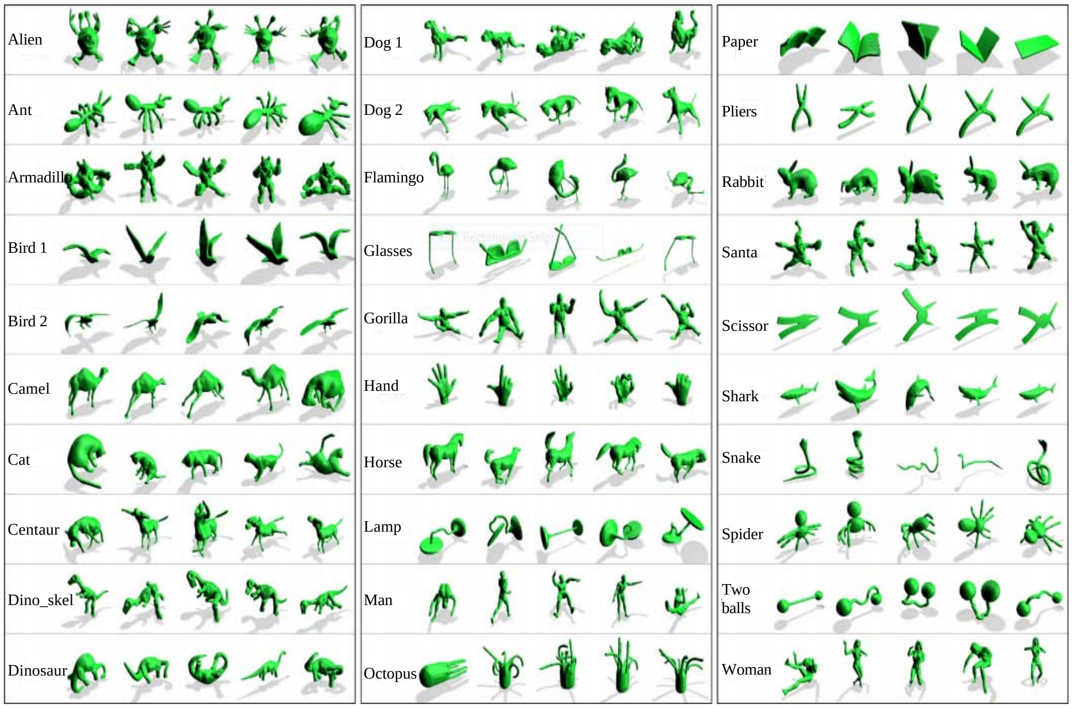

Fig. 2: Examples of 3D Models in SHREC'11 database that is classified into 30 categories

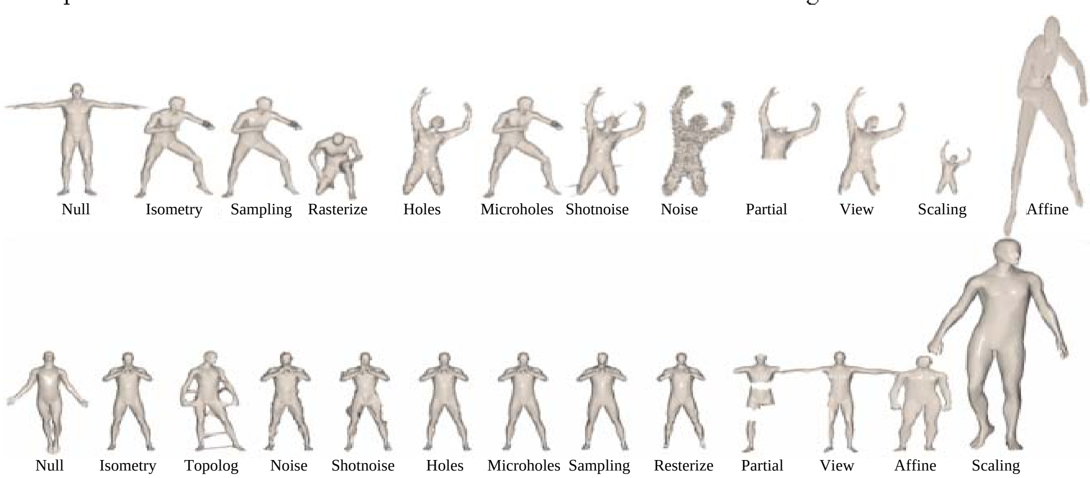

Fig. 3: Examples of 3D Models in SHREC'11 robust database

SHREC'11 is a large scale dataset consisting of 600 non-rigid 3D Models that are equally classified into 30 categories (Fig. 2). The 3D Models are represented as watertight triangular meshes that are derived from 30 original models, represented in different deformed versions.

SHREC'11-Robustness benchmark is a dataset that includes simulated transformations of different types and strengths. For each shape, transformations are split into 12 classes (isometry, topology, small and big holes, scaling, noise, shot noise, missing parts, sampling, affine and rasterize) (Fig. 3). In each class, the transformation appears in five different strength levels. In the experiments, the training dataset is used with a total of 684 shapes. The set includes 13 shape classes with null shapes, each shape have the 12 transformation classes with different degrees.

SHREC'15 consists of 1200 watertight triangle meshes (Fig. 4) that are equally classified into 50 categories. The algorithm is implemented on a standard PC with an Intel Core I7-4702MQ 2.20 GHz CPU and 8 GB memory.

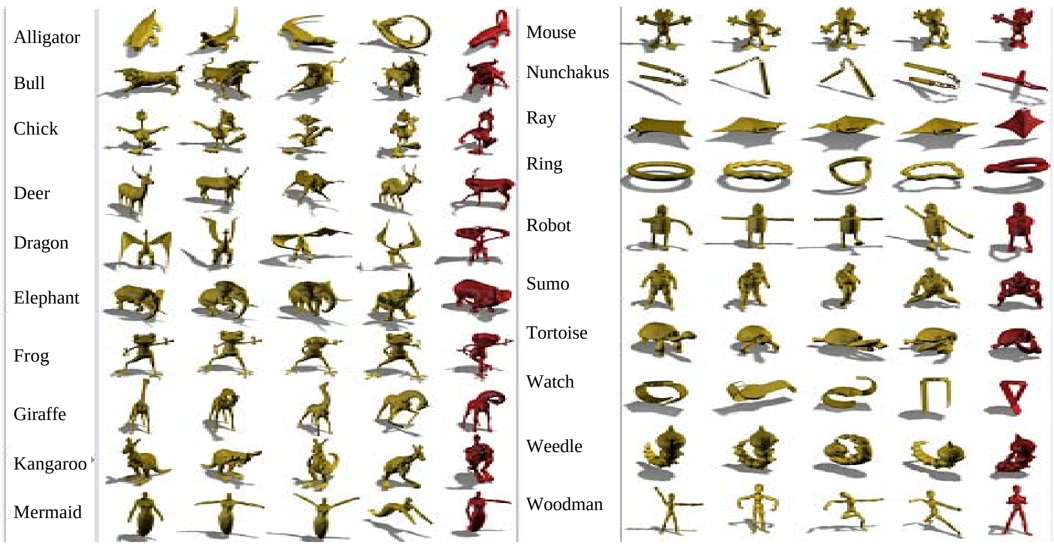

Fig. 4: Examples of 3D Models in SHREC'15 Robust database

The performance is measured by using Princeton Shape Benchmark (PSB) statistics (Shilane *et al*., 2004) which are: Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), e-Measure and Discounted Cumulative Gain (DCG).

Learning the visual codebook: The first step to apply the BoF approach is to build the visual codebook. The visual codebook is built using different clustering algorithms:

- K-means
- Gravitational Search Algorithm (GSA) (Rashedi *et al*., 2009) with 50 run iterations to run it
- Proposed MERC-tree classifier which is a modification of the ERC (Geurts *et al*., 2006; Moosmann *et al*., 2006) where K = 128, n_{min} = 2, M = 10 and t_r = 50

K is the number of attributes randomly selected at each node, M is the number of trees per ensemble, n_{min} is the minimum sample size for splitting a node and t_r is the maximum number of tries to get the maximum score for splitting as by Moosmann *et al*. (2006). Both K-means and GSA are unsupervised learning but the proposed MERC-tree is a supervised learning algorithm.

Vocabulary size: In this experiment, the relationship between the vocabulary size (codebook size) and the retrieval performance was investigated. So, different vocabulary size values were used. The change in vocabulary size means the change in the number of histogram bins and hence, the dimension of the descriptor vector. The used vocabulary sizes were 30, 60 and 120 which are the number of classes in the SHREC'11 and its multipliers.

Distance function: The used distance functions were Euclidean distance and Kullback Leibler Divergence (KLD) as shown in Eq. 1 and 2. The smaller values of these measures means the models are most similar and the larger values means less similar. We also used Normalized KLD to compare its results with KLD results. The normalized KLD is the same as KLD but with normalizing the histograms:

$$\mathrm{Div}(x,y)=\sum_{i=1}^{N_v}(y_i\text{-}x_i)\ln\frac{y_i}{x_i} \qquad (2)$$

Where:
(x, y) = (y_i x_i) = The feature vectors
N_v = The dimension of the vectors

The KLD is sometimes referred to as information divergence or relative entropy.

RESULTS AND DISCUSSION

The impact of changing the different settings was studied in order to select the optimum settings that achieve the best retrieval results. Intuitively, settings attaining high values for most evaluation metrics are considered as the one achieving the best retrieval results.

There are two different scenarios for evaluating the performance: The first scenario is to change the training set size with different clustering algorithms and the same distance function. Table 2-4 study the effect of changing the training set size on different clustering algorithms with the Euclidean distance used as distance function. Table 5-7 study the effect of changing the training set size on different clustering algorithms with KLD as distance function. Table 8-10 study the effect of changing the training set size on different clustering algorithms with normalized KLD as distance function.

The second scenario is to change the distance function with different clustering algorithms and the same training set size. Table 2, 5 and 8 study the effect of changing the used distance function with the training set size equals 50% on different clustering algorithms. Table 3, 6 and 9 study the effect of changing the used distance function with training set size equals 70% on different clustering algorithms. Table 4, 7 and 10 study the effect of changing the used distance function with training set is 100% on different clustering algorithms.

Table 2: Performance comparison of the proposed algorithm using different clustering algorithm with 50% training set of each class features and Euclidean distance as distance function for SHREC'11

No. of clusters (N_v)	Clustering algorithm	NN	FT	ST	E	DCG
30	MERC-trees	0.9417	0.6471	0.6965	0.5132	0.8482
30	K-means	0.4033	0.2059	0.2912	0.1997	0.5275
60	MERC-trees	0.9333	0.4859	0.5262	0.3865	0.7797
60	K-means	0.4133	0.2082	0.2894	0.1974	0.5300
120	MERC-trees	0.9250	0.3685	0.4189	0.3020	0.7125

Table 3: Performance comparison of the proposed algorithm using different clustering algorithm with 70% training set of each class features and Euclidean distance as distance function for SHREC'11

No. of clusters (N_v)	Clustering algorithm	NN	FT	ST	E	DCG
30	MERC-trees	0.9633	0.7485	0.7710	0.5731	0.8845
30	K-means	0.5667	0.2737	0.3627	0.2537	0.5920
60	MERC-trees	0.9567	0.4849	0.5016	0.3707	0.7833
60	K-means	0.5683	0.2799	0.3631	0.2559	0.6003
120	MERC-trees	0.9500	0.3440	0.3652	0.2687	0.7065
120	K-means	0.6017	0.2823	0.3656	0.2572	0.6028

Table 4: Performance comparison of the proposed algorithm using different clustering algorithm with 100% training set of each class features and euclidean distance as distance function for SHREC'11

No. of clusters (N_v)	Clustering algorithm	NN	FT	ST	E	DCG
30	MERC-trees	0.9967	0.9818	0.9872	0.7345	0.9929
30	K-means	0.7683	0.4313	0.5639	0.3967	0.7428
30	GSA	0.1167	0.1257	0.2460	0.1567	0.4662
60	MERC-trees	0.9900	0.4769	0.4982	0.3650	0.8203
60	K-means	0.7917	0.4419	0.5757	0.4033	0.7530
120	MERC-trees	0.9867	0.2283	0.2568	0.1842	0.6717
120	K-means	0.8300	0.4445	0.5773	0.4056	0.7611

Table 5: Performance of the proposed algorithm using 50% training set of each class features using KLD as distance function for SHREC'11

No. of clusters (N_v)	Clustering algorithm	NN	FT	ST	E	DCG
30	MERC-trees	0.935	0.5999	0.6591	0.4818	0.8334
60	MERC-trees	0.930	0.4571	0.5081	0.3685	0.7688
120	MERC-trees	0.925	0.3685	0.4189	0.3020	0.7125

Table 6: Performance of the proposed algorithm using 70% training set of each class features using KLD as distance function for SHREC'11

No. of clusters (N_v)	Clustering algorithm	NN	FT	ST	E	DCG
30	MERC-trees	0.9667	0.6906	0.7181	0.5333	0.8622
60	MERC-trees	0.9583	0.4506	0.4764	0.3501	0.7683
120	MERC-trees	0.9433	0.3133	0.3397	0.2493	0.6869

Table 7: Performance comparison of the proposed algorithm using different clustering algorithm with 100% training set of each class features and using KLD as distance function for SHREC'11

No. of clusters (N_v)	Clustering algorithm	NN	FT	ST	e-Measure	DCG
30	MERC-trees	0.9200	0.9168	0.9200	0.6847	0.9452
30	K-means	0.7667	0.4435	0.5838	0.4101	0.7527
30	GSA	0.1117	0.1261	0.2458	0.1563	0.4664
60	MERC-trees	0.8917	0.4422	0.4584	0.3337	0.7740
60	K-means	0.7633	0.4283	0.5693	0.3986	0.7417
120	MERC-trees	0.9033	0.2171	0.2441	0.1741	0.6458

Table 8: Performance of the proposed algorithm using 50% training set of each class features using normalized KLD as distance function for SHREC'11

No. of clusters (N_v)	Clustering algorithm	NN	FT	ST	e-Measure	DCG
30	MERC-trees	0.9467	0.6395	0.6839	0.5033	0.8445
60	MERC-trees	0.9283	0.4862	0.5292	0.3865	0.7806
120	MERC-trees	0.8950	0.3932	0.4346	0.3170	0.7212

Table 9: Performance of the proposed algorithm using 70% training set of each class features using normalized KLD as distance function for SHREC'11

No. of clusters (N_v)	Clustering algorithm	NN	FT	ST	e-Measure	DCG
30	MERC-trees	0.9433	0.7126	0.7307	0.5428	0.8621
30	K-means	0.4950	0.2057	0.2744	0.1910	0.5412
60	MERC-trees	0.9350	0.4709	0.4861	0.3599	0.7722
120	MERC-trees	0.9117	0.3310	0.3509	0.2586	0.6909

Table 10: Performance comparison of the proposed algorithm using different clustering algorithm with 100% training set of each class features and using normalized KLD as distance function for SHREC'11

No. of clusters (N_v)	Clustering algorithm	NN	FT	ST	e-Measure	DCG
30	MERC-trees	0.9200	0.9168	0.920	0.6847	0.9452
30	K-means	0.7183	0.3839	0.5078	0.3553	0.703
30	GSA	0.0333	0.0333	0.0649	0.0405	0.3452
60	MERC-trees	0.8917	0.4422	0.4584	0.3337	0.7741
60	K-means	0.7267	0.3931	0.5218	0.3656	0.7117
120	MERC-trees	0.9033	0.2171	0.2441	0.1741	0.6457

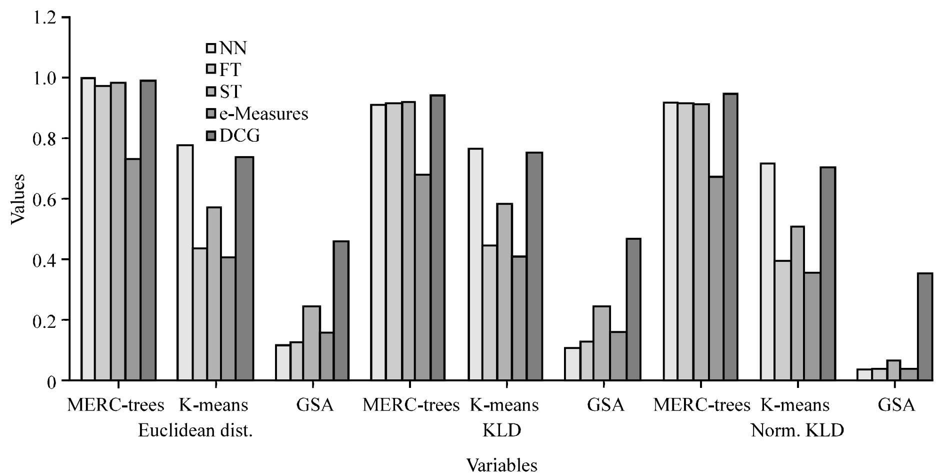

Fig. 5: Performance of the proposed algorithm on the SHREC'11 using different clustering algorithms and distance functions with training set 100% and vocabulary size 30

Table 2 shows the effect of changing the clustering algorithm and the number of clusters using 50% every class features as a training set and using the Euclidean distance as distance function. The proposed MERC-trees achieved the best retrieval results with 30 clusters (the number of classes in the dataset). Increasing the Number of clusters (N_v) than 30 decreases the efficiency of the retrieval results because it means that, the class is divided to more than one cluster which makes those clusters have the same characteristics. Hence, the classifier is confused and its efficiency of classification is decreased. But in K-means the retrieval performance enhanced with increasing the number of clusters because the error in clustering decreased with increasing the number of clusters until it reached to zero when each data object has its own cluster. In Table 3 and 4, the performance of the retrieval algorithm is improved with increasing the training set size as the characteristics of each class will be more obvious, so the clustering algorithm will be more able to distinguish among different classes (Fig. 5).

Table 5-7 show the effect of increasing the training set size and different number of clusters with KLD as distance function. In Table 7 with 100% training set size, different clustering algorithms were compared. The performance of MERC-trees with 30 clusters achieved the best retrieval results.

From Table 7, it is not noticeable that the NN measure decreased with 100% training set size. This because that the KLD which is used as distance function is divergence measure which may decreased with the increasing in the details.

Table 8-10 show the effect of increasing the training set size with different clustering algorithms and different

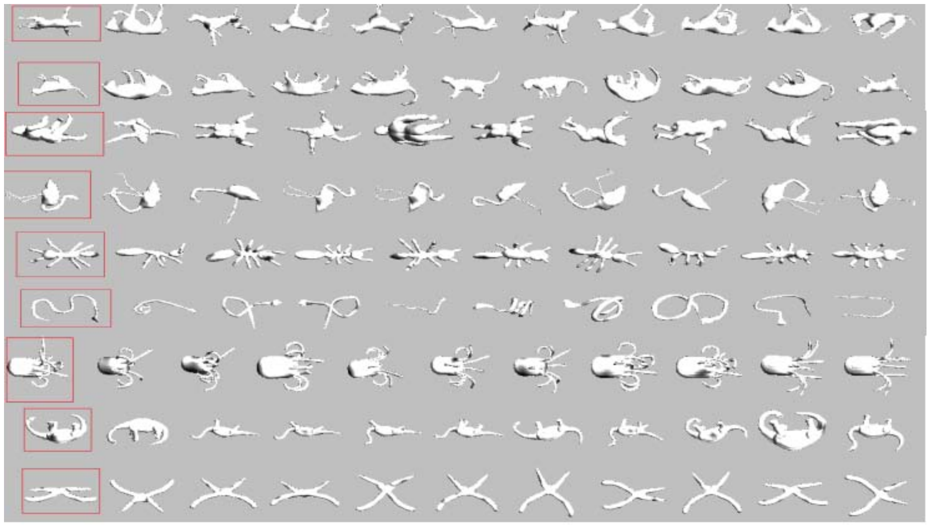

Fig. 6: Sample of 3D object retrieval results. Leftmost column: queries with red squares where each of the queries belongs to different category; ten following columes represent the closest ten retrieved 3D Models for each query

number of clusters with normalized KLD as distance functions. The proposed MERC-trees with 30 clusters achieved also the best retrieval results.

To conclude and as revealed from the result, Euclidean distance outperforms both KLD and the normalized KLD as it uses all the elements in the feature vector but the others uses the non-zero elements only because of the used "In" function (Eq. 2).

Figure 6 shows the visual retrieval results of the proposed algorithm with the vocabulary size N_v = 30, training set size = 70% and the proposed MERC to build visual codebook. The first model is the query model and the first retrieved model at the same time (which explains the highly accuracy for our proposed algorithm with only 70% not 100% training set size. The other nine models are the followed retrieval results.

Table 2-10 show the performance of the proposed algorithm (using different clustering algorithms) on SHREC'11. Table 2-10 show that the proposed MERC-trees achieved best retrieval result with different training set sizes and distance functions. Hence, MERC-trees is applied on SHREC'15 with various training set sizes and distance function (Fig. 7).

Figure 7 shows that Euclidean distance achieved the best retrieval results over all training set sizes and the retrieval performance is improved with the increase in training set size.

Robustness results: To test the robustness of the proposed algorithm against different transformations with different degrees, SHREC11-Robutness was used with 12 classes of transformations each with 5°. For the

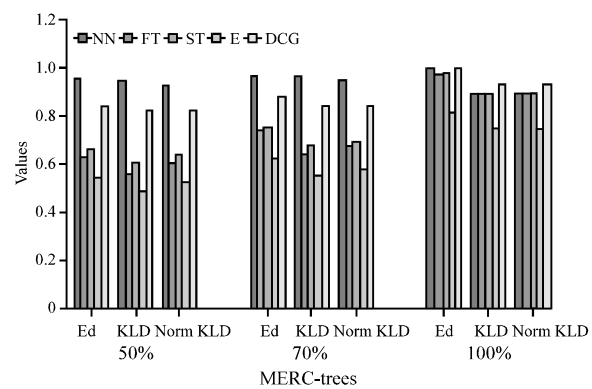

Fig. 7: Performance of the proposed algorithm on the SHREC'15 using MERC-trees for building the visual codebook

training set we used only the null models which are 12 models from the total dataset (684 models). For building the visual codebook only MERC algorithm was used with vocabulary size 12 as each original model and its transformed versions considered as one class. Table 11 shows the retrieval results for the transformed models but after excluding rasterization class as rasterize class is simulating non-pointwise topological artifacts due to occlusions in 3D geometry acquisition which a topological wise deformation and not in the scope of the study. The retrieval results shown in Table 11 are almost perfect as it explains that the proposed algorithm is robust against 11 different type of transformation with 5 different strength degrees. The normalized KLD is the best distance function on SHREC'11-Robust because the

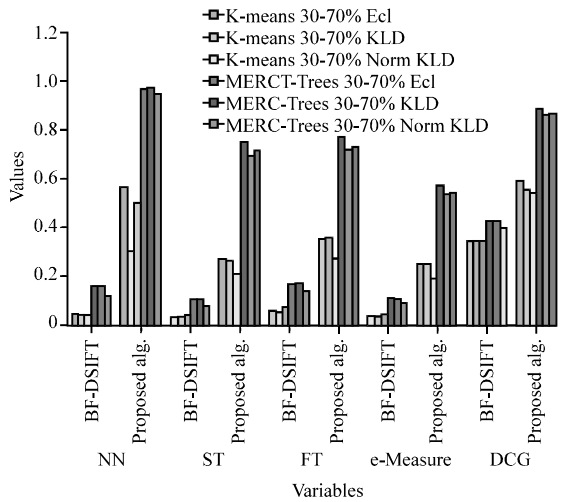

Fig. 8: Performance using the MERC-trees or K-means as a clustering algorithm and using Euclidean distance or KLD as a distance function

normalized KLD is applying KLD distance function on normalized histograms. The normalization decreases the effect of noise.

Evaluations and comparisons: In this study, the performance of the proposed algorithm is compared with our implementation to BF-DSIFT on SHREC 2011 and the implementation of the BF-DSIFT (Furuya and Ohbuchi, 2009) on SHREC 2010 with only 200 models.

The proposed algorithm is compared with BF-DSIFT and Multidimensional Scaling, Clock Matching and Bag-of-Features (MDS-CM-BOF) (Lian *et al*., 2010) algorithms which are view based algorithms participated in the shape retrieval contest for non-rigid 3D Models in 2011 and also, compared with algorithms shared in SHREC 2015. There is no view based algorithms shared in SHREC 2015 contest, all the algorithms were topological, geometrical and non-view based local features which are easier to discriminate the non-rigid objects.

Figure 8 shows that our implementation to BF-DSIFT achieved very low retrieval performance using the MERC-trees or K-means as a clustering algorithm and using Euclidean distance or KLD as a distance function. BF-DSIFT achieved low retrieval performance as it extracts SIFT descriptors at random points taken on each image which maybe not discriminative to its image.

Figure 9 shows that our proposed algorithm which applied on SHREC'11 with 600 models achieved better than MDS-CM-BOF (Lian *et al*., 2010) applied on SHREC'11 and BF-DSIFT performance presented by Furuya and Ohbuchi (2009) on SHREC 2010 with

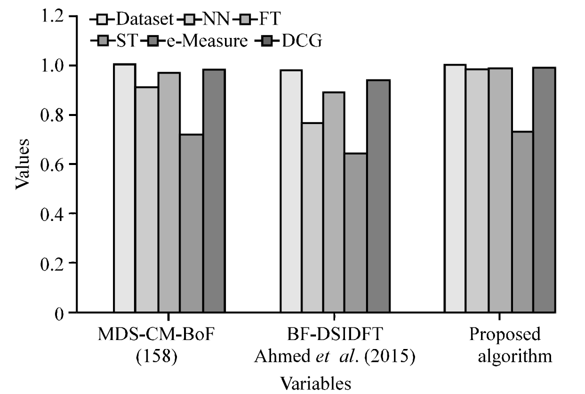

Fig. 9: Performance comparison between the proposed Algorithm and the implemented "BF-DSIFT" algorithms on SHREC 2011

Table 11: The retrieval results of the proposed algorithm (using the proposed MERC classifier and different distance function) on SHREC'11 Robust without rasterize class

Distance function	NN	FT	ST	e-Measure	DCG
Euclidean	0.8926	0.5446	0.6600	0.5346	0.8557
KLD	0.8990	0.5403	0.6557	0.5318	0.8551
Normalized KLD	0.9151	0.7609	0.8639	0.6660	0.9313

Table 12: Performance comparison between the proposed algorithm and the algorithms participate on SHREC'15 (Lian *et al*., 2015)

Algorithm names	NN	FT	ST	e-Measure	DCG
SV-LSF-kpca50	0.1000	0.9972	0.9997	0.8357	0.9997
SV-LSF	0.1000	0.9797	0.9974	0.8292	0.9977
HAPT-run3	0.1000	0.9613	0.9795	0.8117	0.9915
HAPT-run2	0.9983	0.9657	0.9821	0.8150	0.9919
HAPT-run1	0.9975	0.9658	0.9818	0.8150	0.9918
Proposed-100%-ECL	0.9933	0.9716	0.9768	0.8150	0.9882
SPH-sparse-1024	0.9975	0.9568	0.9696	0.8047	0.9885
SPH-sparse coding-256	0.9967	0.9510	0.9662	0.8008	0.9868
SPH-SPEM-VLAD-64	0.9958	0.9343	0.9570	0.7911	0.9811
SPH-SPEM-VLAD-32	0.9942	0.9106	0.9460	0.7778	0.9757
Compact BoFHKS-10D	0.9842	0.8714	0.9082	0.7465	0.9582
Compact BoFHKS-4D	0.9817	0.8722	0.9080	0.7476	0.9581
Compact BoFHKS-19D	0.9825	0.8672	0.9059	0.7440	0.9575
Compact BoFHKS-5D	0.9775	0.8582	0.8988	0.7356	0.9533
FVF-SIHKS	0.9800	0.8249	0.8826	0.7178	0.9503
FVF-WKS	0.9767	0.8225	0.8945	0.7242	0.9518
EDBCF-NW	0.9775	0.7931	0.8839	0.7076	0.9431
EDBCF-AV	0.9750	0.7699	0.8680	0.6899	0.9358
SG-L1	0.9725	0.7596	0.8143	0.6597	0.9192
SID-4	0.9767	0.7188	0.8213	0.6482	0.9200
SID-5	0.9767	0.7109	0.8164	0.6422	0.9171

200 models. The MDS-CM-BOF takes 66 depth images and extracts the SIFT features from them. Although, the MDS-CM-BOF performance is close to the proposed algorithm, performance, the proposed one takes only 42 depth images which mean that, it has better runtime. In addition, the use of our proposed Enhanced Ray Tracing algorithm presented by Ahmed *et al*. (2015) reduced the time for generating the depth images with the same quality as the Naive Ray Tracing algorithm.

In SHREC'15 (Lian *et al*., 2015) 37 methods were participated which none of them are view based method. Table 12 shows a comparison among the top 20 methods and the proposed algorithm performance. The proposed algorithm (proposed-100%-ECL), using our proposed MERC classifier with training set size = 100% and the Euclidean distance as a distance function, achieved the 6th best performance.

CONCLUSION

In this study, an efficient view based approach for retrieving 3D non-rigid objects was introduced. The proposed algorithm is performed in five execution phases and a great improvements were done on most phases. The first phase is normalizing the model which includes position normalization followed by scale normalization. The second phase is multi-view rendering which includes placing the viewpoints at vertices of regular or near regular polyhedrons enclosing the model and render the depth image for the model from each one using our proposed Enhanced Ray Tracing algorithm presented by Ahmed *et al*. (2015) which achieved a great reduction in rendering time reached 99.8% with the same quality as the naive ray tracing. The third phase is extracting the silent points from each depth images and extracts the SIFT descriptor at each point. The fourth phase is extracting the training set, building the visual codebook using our proposed MERC algorithm and building a histogram for each model in the database. The histogram is the descriptor vector of each model which represents the frequency of each visual word in the visual codebook. The fifth and final phase is the descriptor matching where the Euclidean distance was used to calculate the dissimilarity between the models descriptors.

The experimental results lead to the conclusion that the proposed technique is quite effective for the purpose of 3D object retrieval, showing very high retrieval accuracy. It achieved high performance on SHREC 2011 dataset, SHREC 2015 and 2011-Robust; the public well known benchmark s of non-rigid 3D Models. The results have indeed confirmed that the proposed descriptor is invariant against different kinds of deformations. Moreover, it is significant that the proposed technique is computationally efficient. Through employing an extensive comparative study, the proposed algorithm is compared against other state of art view based algorithms. These methods had already been evaluated on SHREC 2011 and SHREC 2010 datasets (which contains only 200 non-rigid 3D watertight models and considered only as a part of SHREC'11) by the shape retrieval contest of non-rigid 3D watertight meshes. It was shown that the proposed algorithm clearly outperforms all these state of the art retrieval methods applied on SHREC 2011 dataset and on SHREC 2010, regarding different evaluation metrics. The proposed algorithm was ranked first among all other methods that we compared it with. To sum up, all these results lead to conclude that the proposed approach is:

- Highly discriminative
- Computationally efficient
- Robust against different transformations with different degrees

RECOMMENDATIONS

For future research, we intended to explore in depth the following topics: adapting the proposed 3D object Retrieval technique to handle partial matching problem where retrieval is required inside 3D scenes containing an instance of the query object or when there are missing parts in an incomplete 3D Models. Extending the application of the proposed 3D object retrieval technique on volumetric 3D Models. Testing the proposed technique on domain-specific benchmarks (e.g., proteins, CAD models, faces, etc.); such that it can be reformed for adapting different domains of knowledge.

REFERENCES

Ahmed, H., H. Shedeed and D. Hegazy, 2015. Enhanced ray tracing algorithm for depth image generation. Br. J. Math. Comput. Sci., 11: 1-11.

Chen, D.Y., X.P. Tian, Y.T. Shen and M. Ouhyoung, 2003. On visual similarity based 3D Model retrieval. Comput. Graphics Forum, 22: 223-232.

Elad, A. and R. Kimmel, 2003. On bending invariant signatures for surfaces. IEEE. Trans. Pattern Anal. Mach. Intell., 25: 1285-1295.

Furuya, T. and R. Ohbuchi, 2009. Dense sampling and fast encoding for 3D Model retrieval using bag-of-visual features. Proceedings of the ACM International Conference on Image and Video Retrieval, July 08-10, 2009, ACM, Santorini, Fira, ISBN: 978-1-60558-480-5, pp: 1-26.

Geurts, P., D. Ernst and L. Wehenkel, 2006. Extremely randomized trees. Mach. Learn., 63: 3-42.

Hess, R., 2010. An open-source SIFT library. Proceedings of the 18th ACM International Conference on Multimedia, October 25-29, 2010, ACM, Firenze, Italy, ISBN:978-1-60558-933-6, pp: 1493-1496.

Hilaga, M., Y. Shinagawa, T. Kohmura and T.L. Kunii, 2001. Topology matching for fully automatic similarity estimation of 3D shapes. Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, Aug. 12-17, ACM, Los Angeles, pp: 203-212.

Jain, V. and H. Zhang, 2007. A spectral approach to shape-based retrieval of articulated 3D Models. Comput. Aided Des., 39: 398-407.

Lian, Z., A. Godil, B. Bustos, M. Daoudi and J. Hermans *et al*., 2011. Shape retrieval on non-rigid 3D watertight meshes. Proceedings of the Euro graphics Workshop on 3D Object Retrieval (3DOR) April 10-10, 2011, Mitsubishi Electric Research Laboratories Inc., Llandudno, Wales, ISBN: 978-3-905674-31-6, pp: 1-12.

Lian, Z., A. Godil, T. Fabry, T. Furuya and J. Hermans *et al*., 2010. SHREC'10 track: Non-rigid 3D shape retrieval. Proceedings of the Eurographics/ACM SIGGRAPH Symposium on 3D Object Retrieval, May 2-2, 2010, Eurographics Association, Norrkoping, Sweden, pp: 101-108.

Lian, Z., A. Godil, X. Sun and H. Zhang, 2010. Non-rigid 3D shape retrieval using multidimensional scaling and bag-of-features. Proceedings of the 17th IEEE International Conference on Image Processing (ICIP), September 26-29, 2010, IEEE, Hong Kong, China, ISBN:978-1-4244-7992-4, pp: 3181-3184.

Lian, Z., J. Zhang, S. Choi, H. El-Naghy and J. El-Sana *et al*., 2015. SHREC'15 Track: Non-rigid 3D shape retrieval. Proceedings of the Eurographics Workshop on 3D Object Retrieval, May 2-3, 2015, Eurographics Association, Zurich, Switzerland, pp: 153-160.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision, 60: 91-110.

Memoli, F. and G. Sapiro, 2005. A theoretical and computational framework for isometry invariant recognition of point cloud data. Found. Comput. Math., 5: 313-347.

Mikolajczyk, K.I. and C. Schmid, 2005. A performance evaluation of local descriptors. IEEE Trans. Pattern Anal. Mach. Intell., 27: 1615-1630.

Moosmann, F., B. Triggs and F. Jurie, 2006. Randomized clustering forests for building fast and discriminative visual vocabularies. Proceedings of the Conference on Neural Information Processing Systems (NIPS'06), December 4-7, 2006, Vancouver, British Columbia, pp: 1145-1152.

Ohbuchi, R., K. Osada, T. Furuya and T. Banno, 2008. Salient local visual features for shape-based 3D model retrieval. Proceedings of the IEEE International Conference on Shape Modeling and Applications SMI, June 4-6, 2008, IEEE, Stony Brook, New York, USA., ISBN:978-1-4244-2260-9, pp: 93-102.

Ohbuchi, R., T. Minamitani and T. Takei, 2005. Shape-similarity search of 3D models by using enhanced shape functions. Intl. J. Comput. Appl. Technol., 23: 70-85.

Osada, R., T. Funkhouser, B. Chazelle and D. Dobkin, 2002. Shape distributions. ACM Trans. Graphics (TOG), 21: 807-832.

Rashedi, E., H. Nezamabadi-Pour and S. Saryazdi, 2009. GSA: A gravitational search algorithm. Inform. Sci., 179: 2232-2248.

Reeb, G., 1946. On the singular points of a completely integrable pfaff form or of a numerical function. Comptes Rendus Acad. Sci. Paris, 222: 847-849.

Reuter, M., F.E. Wolter and N. Peinecke, 2005. Laplace-spectra as fingerprints for shape matching. Proceedings of the ACM Symposium on Solid and Physical Modeling, June 13-15, 2005, ACM, Cambridge, Massachusetts, ISBN:1-59593-015-9, pp: 101-106.

Shilane, P., P. Min, M. Kazhdan and T. Funkhouser, 2004. The princeton shape benchmark. Proceedings of the Shape Modeling International, (SMI'04), Los Alamitos, pp: 167-178.

Sundar, H., D. Silver, N. Gagvani and S. Dickinson, 2003. Skeleton based shape matching and retrieval. Proceedings of the Shape Modeling International, May 12-15, 2003, IEEE Computer Society Washington, DC, USA., pp: 130-139.