IJICTR International Journal of Information & Communication Technology Research ITRC

Volume 4- Number 1- December 2011 (9-20)

A Novel 3D Object Categorization and Retrieval System Using Geometric Features

Mohammad Ramezani
Computer Vision Lab, Electrical Engineering Faculty
Sahand University of Technology
Tabriz, Iran
mr_ramezani@sut.ac.ir

Hossein Ebrahimnezhad
Computer Vision Lab, Electrical Engineering Faculty
Sahand University of Technology
Tabriz, Iran
ebrahimnezhad@sut.ac.ir

Received: September 15, 2011- Accepted: November 28, 2011

Abstract — **In this paper, we propose a novel geometric features based method to categorize 3D models using probabilistic neural network and support vector machine classifiers. The employed features are extracted from face and vertex characteristics. In addition, we utilize the proposed features in 3D object retrieval. To achieve this end, each model is decomposed into a set of local/global geometrical features. We use histograms of two variables, i.e., deviation angle of normal vector on the object surface point from the vector that connect shape center to that point; and distance of object surface point from shape center. To achieve better separability of different models, mutual Euclidean distance histogram for the pairs of surface points is also used. The most advantage of using histogram to represent the features is that it shows the density of data and enables creating of low dimensional feature vector and consequently decreasing of computational cost in classification process. The effectiveness of our proposed 3D object categorization system has been evaluated on the generalized McGill 3D model dataset in terms of both accuracy and speed measures. Widespread experimental results and comparison with the other similar methods, demonstrate efficiency of the proposed approach to improve both accuracy and speed of categorization system.**

Keywords- 3D object; vertex normal vector; center- to-vertex vector; mutual Euclidean distance; histogram.

I. Introduction

In recent years, an incredible advance on 3D model acquisition and imaging technology has emerged and large number of 3D models has become available on the web or through other means. Now a days, 3D models are used in many applications such as computer game, computer aided design, e-business, home design, molecular biology, historical object preservation, etc, and grow to be the fourth multimedia type after sound, image and video [1]. 3D shapes retrieval has become important, both in practice and research. This had led to several systems implementation. In this topic, a 3D model is represented as a triangle mesh. At a high level, an algorithm of 3D object retrieval and classification works directly on 3D models or a set of projected images [2,3] taken from multiple views which called the object-space and the image-space approaches, respectively. Many of object-space shape descriptors make some spherical functions that acquire the geometric information in a 3D shape, extrinsically [4]. These spherical functions correspond to the distribution of some quantities. Since, the route of object recognition by human has not been completely understood, we are still not able to confirm theoretically that what kind of shape retrieval and classification approach is the best.

In the main, there are three major steps in both 3D model retrieval and classification systems:

A. 3D Pose Estimation and Normalization

3D models are shaped in random scale, orientation, and position in 3D space. Therefore, pose normalization of 3D models is a key step in many computer graphic applications; for instance 3D model retrieval and classification, 3D model recognition, 3D visualization, etc. The objective of 3D model pose normalization is to convert a model into a canonical coordinate framework, where the model is represented independent of its scale, direction, and position. An ideal canonical coordinate frame of a 3D model is identified with axes that are parallel to the front-back, left-right and top-bottom directions of the model. The normalization step is not requisite when we use intrinsic features.

B. Feature Extraction

Feature extraction is an important pre-processing step in pattern recognition and machine learning problems to generate feature vector for describing 3D shapes. Usually, feature vectors are selected in a fixed dimension. To improve the performance of classification, the size of feature vector should be increased while it leads to more required storage and computational complexity. The standard descriptors have been brought into the frame of MPEG-7 [5,6]. In this frame, several multimedia standards describe open design of various kind of audiovisual information.

C. Classification in Feature Vector Space

Neural network have been of interest over the last decades, and are being successfully applied across a wide range of problems in miscellaneous areas as finance, medicine, engineering, geology and physics. In fact, neural networks are being introduced anywhere we face with problems of prediction, classification or control. SVM is another powerful tool for solving the problems, especially in nonlinear classification, function estimation and density evaluation which has also led to many other recent improvements in kernel based learning methods, in general. Basically, SVMs have been introduced within the context of statistical learning theory and structural risk minimization. SVMs have practically good performance, successful applications in many fields such as bioinformatics, text and image recognition, etc.

In this paper, we introduce a 3D shape descriptor based on one-dimensional histogram of mutual Euclidean distance for each pairs of surface points and two-dimensional histogram of normal deviation angle and distance from object center for each surface point. To categorize the object, features are applied into both Probabilistic Neural Network (PNN) and Support Vector Machine (SVM) classifiers.

The remainder of this paper is organized as follows. At first, we present some related previous works in section 2. After brief discussion on preprocessing and pose normalization in section 3, we describe histogram of different extracted features from 3D model in section 4. Classification step is studied in section 5 and experimental results and discussions are followed in section 6. Finally, we conclude in Section 7.

II. Previous Work

Early research on 3D model processing was almost commenced by Paquet who caught some outstanding achievements [7,8]. Afterward, many scientists have been continued to work on several researches in 3D model processing fields. Now, many approaches of 3d model retrieval have been suggested that are generally divided into four main categories: shape based methods, topology based approaches, image based techniques and surface characteristics based algorithms. Anderst [9] modified 3D model search system using some features and calculated the proportions of point number of each unit in 3D model and formed shape histograms of those features. Anderst defined three methods to enhance and modify 3D models as shell model, sector model and spider web model. Therefore, 3D model retrieval can be accomplished by matching of shape histograms. Suzuki [10] discussed on different methods to modify 3D models and reduce the computational cost which called point density. By this method, the modified units are separately applied to classifiers. So, dimension of feature vector and computational quantity are significantly decreased. Mesh point's curvature as well as principle curvature direction and its magnitude have been used for classification and retrieval of different shapes [11] as other geometric features. Of course, point's curvature is very sensitive to noise. To deal with shape articulation, skeleton or other graph representation of the 3D shapes [12,13] is employed for matching in equivalent parts. But, the computational cost of skeleton extraction from 3D shapes would be great and similarly the subsequent graph matching is often computationally expensive, especially when medial axes are employed [13]. In addition to expensive computational cost, the shape descriptor itself is sensitive to topological noise. Another one of most general and applicable approaches for 3D model feature extraction is spectral domain that is enough fast to compute and automatically normalize 3D shapes against different rigid-body transformations, uniform scaling and bending. The resulting shape descriptors provided a more perceptive way of characterizing the shapes [14]. Moreover, the spectral approach is quite flexible and permits for different picking of graph edge weights and distance calculations. Also, rendering of the approach is more robust against topological noise. Elad and Kimmel [15] worked on manifold meshes and presented spectral embedding using Multi-Dimensional Scaling (MDS) based on geodesic distances. Osada [16] presented a 3D model retrieval system based on shape distributions. The important step of this method was defining the functions which could describe the models as fine. The main idea was to calculate and get large numbers of statistical data which could be served as shape distributions to describe the features of models. Osada presented five simple and computationally easy shape functions. Because of using large number of statistical data, this method is robust to noise, re-sampling and mesh simplification. Another sample to 3D objects descriptor is the work of Vranic [17] which applies energy scale in spherical frequency components and use spherical harmonics analysis called 2D Fourier Transform on unitary sphere. The process of feature extraction is slow, because this method requires sampling and harmonics transform. Utilizing of 3D objects spectrum characteristics is another field which attracted many research's attention. The knowledge of spectral approaches for data analysis have been used in many applications in clustering [18,19] and correspondence analysis [20-22]. Moment-based and wavelet transform-based descriptions [23] are another approaches. These descriptors are not robust to the level-of-detail of the model. There are also many researches on image-based 3D model retrieval [24,25] of which Chen made significant productions. In these methods, several images from different views of each 3D model are taken and employed for representing 3D models and making descriptors in 2D shapes. Afterward, matching of 3D models is accomplished by matching the corresponding 2D images in multiple views by those descriptors. Because of using multiple views image matching, the computation would be enormous. Chen [26] developed a web-based 3D model retrieval system in which a topology method using Reeb graph was introduced. Reeb graph could be created by different precisions, thus in this way, mutli-resolution retrieval is available.

In this paper, we focus on 3D object classification by extracting features of object as the combinational information of surface points that relates distance of surface points from center of object to the direction of normal at those points. To combine these geometric information, we use histogram of two variables: deviation angle of normal vector on the object surface point from radius vector that connect shape center to that point and distance of that point to the shape center. Also, mutual Euclidean distance histogram for pairs of surface points is used to obtain more shape information. To reduce the feature sensitivity to shape fluctuation and noise, we smooth 3D objects at first. Fig. 1 illustrates the whole process of the proposed method in this paper.

III. POSE NORMALIZATION

3D models may have an arbitrary position, orientation and scaling in 3D space. In order to capture the projected features of a model, the model has to be placed in a canonical coordinate frame. Thus, scaling, translating, rotating or flipping of the model will be same as placing of it into the canonical frame. Let $p=\{p_1,p_2,...,p_m\}; p_j=(x_j,y_j,z_j)\in R^3$ be the set of vertices on the surface of a model. The goal of pose normalization is to find an affine map that put the model normatively by translating, rotating and scaling [23]. The translation invariance is achieved by finding the center of the model as Eq. (1):

$$X_c=\frac{1}{m}\sum_{i=0}^{m-1}x_i; Y_c=\frac{1}{m}\sum_{i=0}^{m-1}y_i; Z_c=\frac{1}{m}\sum_{i=0}^{m-1}z_i \qquad (1)$$

Since, the point $p_c=(x_c,y_c,z_c)$ is the center of this model; we move this point to the origin of coordinate system. Therefore, for each point $p_j=(x_j,y_j,z_j)\subset p$, a corresponding transformation is performed as $p_j^{'}=(x_j-x_c,y_j-y_c,z_j-z_c)$. Based on this transformation, we define points set

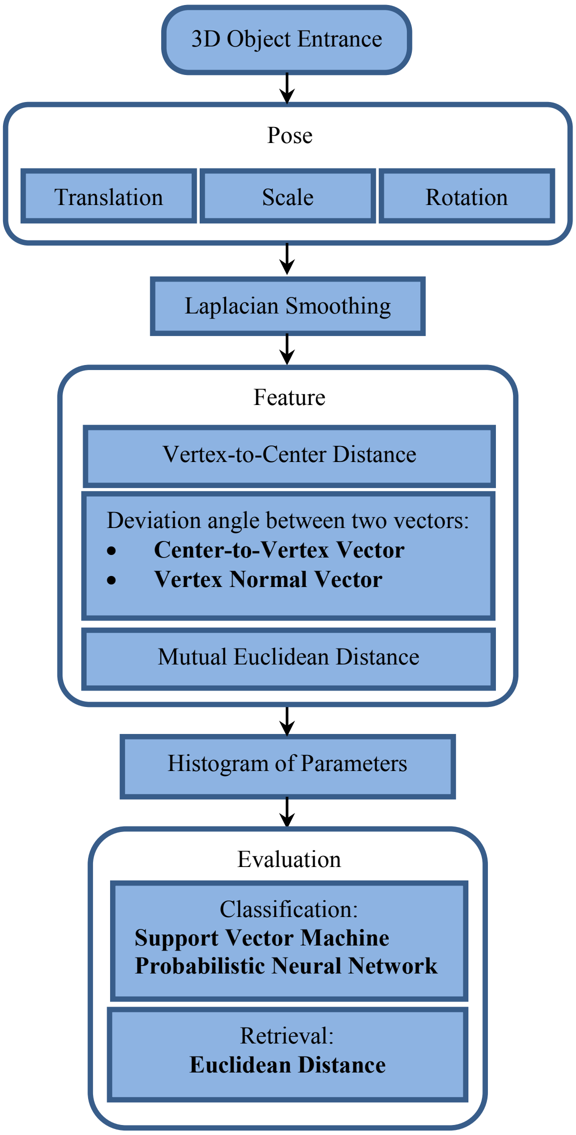

Figure 1. Overal diagram of the proposed method

$p^{'}=\{p_1^{'},p_2^{'},...,p_m^{'}\}$. The rotation invariance is done by the method of PCA [27]. The most well-known method to accomplish the pose normalization is the Principal Component Analysis (PCA). Usually, PCA is applied only to a set of surface points (e.g., vertices or centroid of faces), thus, the differing sizes of faces cannot be taken into account. In order to account the differing sizes of triangles of a mesh, an optimum algorithm is used which consider difference of triangles size using suitable weighting factors. We calculate the covariance matrix C as Eq. (2):

$$C=\sum_{k=0}^{m-1}p_k^T p_k=\begin{bmatrix}\sum_{k=0}^{m-1}x^2 & \sum_{k=0}^{m-1}xy & \sum_{k=0}^{m-1}xz\\ \sum_{k=0}^{m-1}yx & \sum_{k=0}^{m-1}y^2 & \sum_{k=0}^{m-1}yz\\ \sum_{k=0}^{m-1}zx & \sum_{k=0}^{m-1}zy & \sum_{k=0}^{m-1}z^2\end{bmatrix} \qquad (2)$$

This step must be carried out after translating of p_c to the origin of coordinate. Obviously, the matrix is a real symmetric and consequently its Eigen values are non-negative real numbers. Then, we sort the Eigen values in a non-increasing order and find the corresponding Eigen vectors. The Eigen vectors are scaled to Euclidean unit length and we construct the rotation matrix R with scaled Eigen vectors as its rows. All points are rotated in p' and a new points set is formed. As shown in Eq. (3), new coordinate of each vertex is obtained by multiplying of rotation matrix R to that vertex.

$$p^{"} = \{p_l^{"} \mid p_l^{"} = R \cdot p_l^{'}, p_l^{'} \in p^{i}, l = 1,2,3,\ldots,m\} \quad (3)$$

By applying PCA, the model is rotated in 3D space so that X-axis is posed in the direction of most important Eigen vector corresponds to the biggest Eigen value, Y-axis is posed in the direction of second important Eigen vector and finally Z-axis is posed in the direction of third important Eigen vector.

The scaling invariance is accomplished by zooming in (or out) of the model to a fixed size. After doing this step, the 3D model is embedded by a normalized cube with edge length of 2 and origin of coordinate system is posed in the center of cube. Assume that r_{max} is the maximum distance from a set of vertices to the origin of coordinate system. Then, to achieve the final coordinate of each vertex, we resize it using Eq. (4):

$$p^{'''} = \{(\frac{x_s^{"}}{r_{\max}}, \frac{y_s^{"}}{r_{\max}}, \frac{z_s^{"}}{r_{\max}})\}, s = 1,2,3,\ldots,m \quad (4)$$

Where $p_s^{"} = (x_s^{"}, y_s^{"}, z_s^{"}) \in p^{"}$. Obviously, for each point in $p^{"}$, the following inequality will be established:

$$p_t^{'''} = (x_t^{'''}, y_t^{'''}, z_t^{'''}), -1 \le x_t^{"}, y_t^{"}, z_t^{"} \le +1,$$
$$p_t^{'''} \in p^{'''}, t = 1,2,3,\ldots,m \quad (5)$$

An example of pose normalization is displayed in fig.2. After pose normalization, we smooth the surface of model. In this process, we enhance conditions of vertices and faces just by modifying their position. So the number of vertices and faces has no changes. Smoothing process reduces the surface noise and improves the surfaces in appearance. Laplacian smoothing is one simple and effective technique for polyhedral surface smoothing [28]. This technique changes the position of nodes without modifying the topology of the mesh. Each inner surface vertex is moved into a new position given by the average of neighboring vertices. This technique works very fine for all-triangular (or quadrilateral) meshes. To execute the smoothing procedure, consider a triangular surface and for each vertex V let us define the umbrella-operator as Eq. (6):

$$U(V) = \frac{1}{\sum_i \omega_i} \sum_i \omega_i Q_i - V \quad (6)$$

Where, summation is taken over all neighbors of V (i.e. Q_i) and ω_i are positive weights. The weights can be defined as the inverse distances between V and its neighbors Q_i:

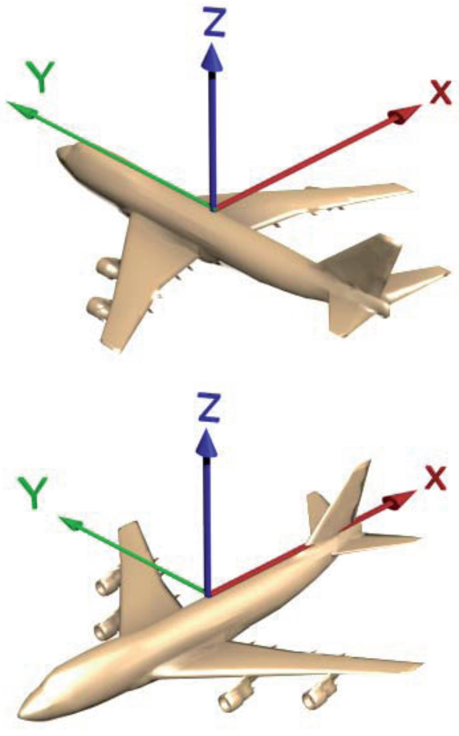

Figure 2. 3D objects pose normalization using PCA algorithm. **Up**: Primary 3D model with arbitrary position. **Bottom**: Normalized model into standard coordination after translation, rotation and scaling.

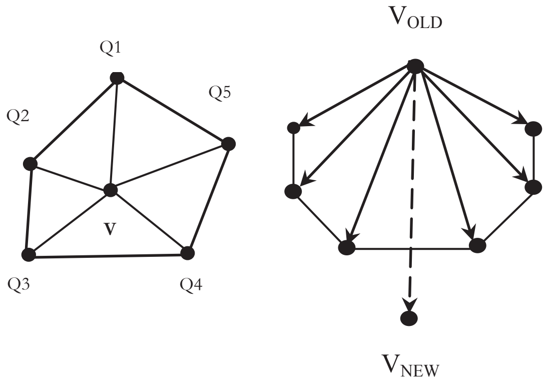

Figure 3. Umbrella operator for laplacian smoothing technique [28]

$$\omega_i = \left\| V - Q_i \right\|^{-1} \quad (7)$$

The simplest umbrella-operator is obtained by choosing $\omega_i = 1$ and the resulted umbrella-operator will be:

$$U(V) = \frac{1}{n} \sum_i Q_i - V \quad (8)$$

Where, n is the number of neighbors for vertex V. The local update rule is shown in fig. 3. This operation is applied all over the surface points and the smoothed form of 3D model is achieved (an example is shown in fig. 4).

IV. Feature Extraction

3D model consists of a set of vertices and faces that relate the vertices in triangular form. Each 3D model may include a large number of vertices and faces, which results in a high amount of required memory and computational cost. So, we summarize the surface of model by reducing the number of vertices and faces in a factor that the overall shape of the model does not loss. After reducing the number of patches, we extract mutual Euclidean distances for each pair of vertices, distance from points on the surface of object to the center of model and vertex normal vector angle versus vector that connect center of model to that vertex and use histograms of them as features. Histograms are applied to PNN and SVM classifiers. Mutual Euclidean distances are considered in one-dimensional histogram while the distances from surface points to the center of the model in conjunction with vertex normal vector angle versus vertex vector from the center are considered in two-dimensional histogram. Since, the number of vertices and faces in 3D model are so much, we can reduce high dimensionality of features by employing the histograms of parameters. Usually, 3D model surface contains several tens of thousands of vertices and several hundred thousands of triangular faces. In fact, employing such kind of feature is very exhausting. So, making histogram of these features will revise the problem and present more acceptable description of 3D model. Moreover, employing histogram form of parameters as the feature vector removes the need of ordering and arrangement between parameters in different models. The number of bins for each axis of 2D histogram and 1D histogram is empirically selected as 20 and 100, respectively. In fact, there is a trade-off between the number of histogram bins and computational cost, required storage and accuracy. We choose the best commodious number of bins in this paper. Fig. 5 displays a simple 2D histogram of two parameters. In this figure, the quantity of each bin is presented with the color and the height of bars. Vertex point in triangular mesh is the intersection point of some faces. For each face, the normal vector is defined as a outgoing vector which is perpendicular to the surface. To increase the robustness of normal direction against the shape noise, we define normal vector for each vertex of 3D model as the average normal vectors of its adjacent faces:

$$\vec{n} = \sum_{i=1}^{m} \vec{n}_i \Big/ \left| \sum_{i=1}^{m} \vec{n}_i \right| \tag{9}$$

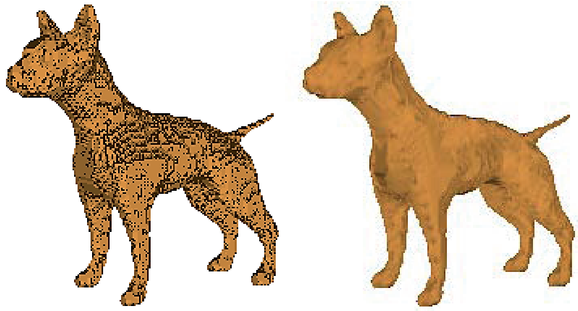

Figure 4. **Left**: Initial triangular mesh model. **Right**: Resulting triangular mesh by applying the laplacian smoothing technique

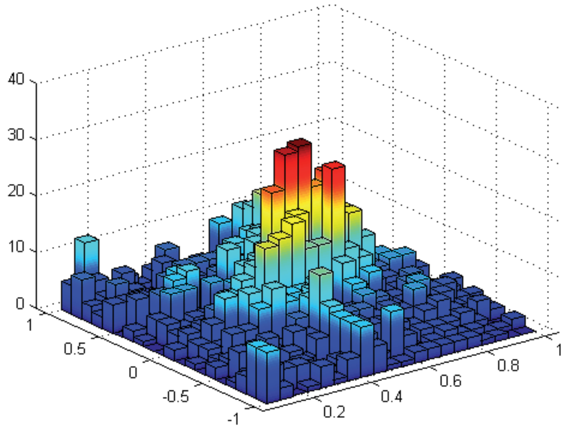

Figure 5. 2D Histogram sample. First axis shows distance (d), second axis shows angle (φ) and third axis shows the height of histogram.

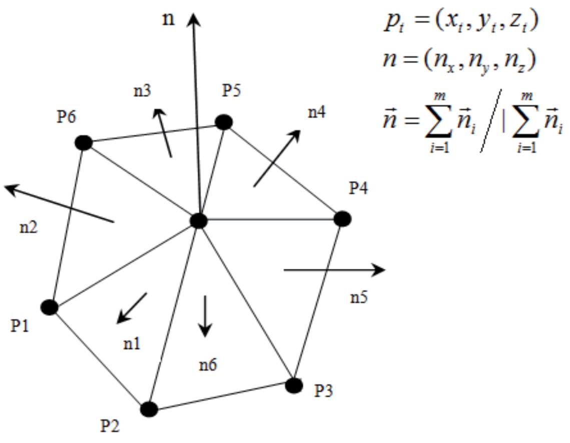
n
n3
P5
P6
n4
P4
n2
n5
P1
n1
n6
P3
P2
$p_t = (x_t, y_t, z_t)$
$n = (n_x, n_y, n_z)$
$\vec{n} = \sum_{i=1}^{m} \vec{n}_i \Big/ \left| \sum_{i=1}^{m} \vec{n}_i \right.$

Figure 6. Each vertex normal vector is computed by averaging the adjacent faces normal vectors

This idea has been illustrated in fig. 6 in which some faces join each other in a specific vertex (p). The resultant vertex normal (n) is computed by summing and normalizing of the related face normal vectors (n1, n2, n3, n4, n5, n6). Visual description of the geometrical features that used in this paper is shown in fig. 7. As it is shown, in middle triangular mesh model, point O is the center of model, red colored vector is a vector that leave center of 3D model and pass through a vertex O' on the surface, and the blue colored vector is the normal vector that is perpendicular to the surface at vertex O'. The 2D histogram of vertex distance from origin ($\overline{OO'}$) and deviation angle of blue vector from red one (ϕ) are considered as geometrical features. The last extracted feature is the histogram of mutual Euclidean distances between each pair of vertices that is shown in right model with red lines for some pairs.

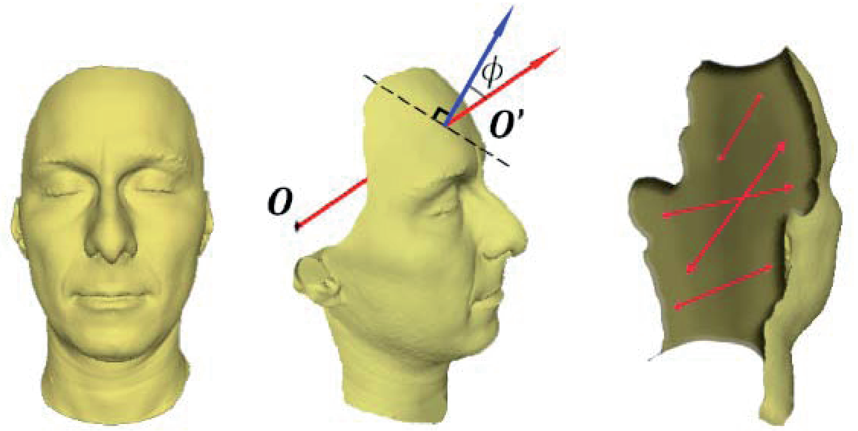

Figure 7. Visual description of employed geometric features. **Left**: Initial 3D model, **Middle**: Distance of vertex from center of object ($\overline{OO'}$), and deviation angle of vertex normal vector from center-to-vertex vector, **Right**: Mutual Euclidean distance for each pair of vertices.

V. CLASSIFICATION

In this stage, we employ PNN and SVM classifier to separate each group of the objects. In the following, we introduce these two classifiers, briefly. Probabilistic Neural Networks (PNN) has some advantages in classification problems against the multi-layer perceptron networks (MLP):

- Training of a PNN is usually much faster than a multi-layer perceptron network.
- PNNs are often more accurate than multilayer perceptron networks.
- PNNs are considerably insensitive to outlier data.
- PNNs generate accurate predicted target probability scores.

Probabilistic neural networks are conceptually similar to K-Nearest Neighbor (k-NN) models, but the implementation is very different. The basic idea is that two items with close values make likely predicted target values. In fact, PNN is an extension to the Bayes classifier that learns to estimate the probability density function of the training samples using radial basis function (RBF) to compute the weight or influence for each point, more accurately. In the other word, the PNN classifier is known as a function in which approximates the probability density of the distribution of primary examples. Probabilistic neural network is closely associated to Parzen window pdf estimator and contains various sub-networks, each of which is a Parzen window pdf estimator for each class. Fig. 8 shows the PNN architecture. The PNN consists of nodes allocated in four layers:

- **Input layer:** The first layer is a set of measurements. In this layer, there is only one neuron for each predictor variable. The input neurons normalize the range of values by subtracting the median and dividing to mid-spread.

- **Pattern layer:** This layer consists of Gaussian functions formed using the given set of data points as centers. For each training example, there is one pattern node which forms a product of the weight vector and the given example for classification. consequently, the product is passed through an activation function.

- **Summation layer**: This layer computes an average of the outputs from the pattern layer for each class.

- **Output layer**: The output nodes are binary neurons that make decision of classification. This layer use competitive activation function to perform a vote by selecting the largest value. As a result, the associated class label is determined.

The only factor that needs to be defined for training is the spread factor that is the standard deviation of the Gaussian functions. This factor adjusts the generalization and approximation. Too small deviations cause a very sharp approximation that cannot generalize well, while too large deviations smooth out details and decrease the approximation accuracy.

Beside PNN classifier, we use SVM as one powerful and common classifier. In SVM algorithm, a hyper-plane or set of hyperplanes in a high dimensional space is used for classification. Naturally, the hyper-plane that has the largest distance to the nearest training data points in any class, makes the best separation. Fig. 9 illustrates operation of SVM for two and three-class dataset. Accidentally, SVM classifiers minimize the empirical classification error and maximize the geometric margin. So, they are also known as maximum margin classifiers. This classifier is basically appropriate for two-class tasks. Therefore, to apply SVM in multi-class problems, we must reduce the multi-class task to several binary problems where each problem discriminates between one of the labels to the rest, one-versus-all [29] or between every pair of classes, one-versus-one [30]. By relating independently produced binary classifiers through the above-mentioned method, a multi-class SVM classifier is obtained. In one-versus-all approach, we seek for the classifier by training one class against the other classes so that a point is assigned to the class for which the distance from the margin, in the positive direction to this particular class, is the maximal. The one-versus-one approach formulates $N(N-1)/2$ classifiers by comparing each pair of classes. To classify a point, the method combines discrimination

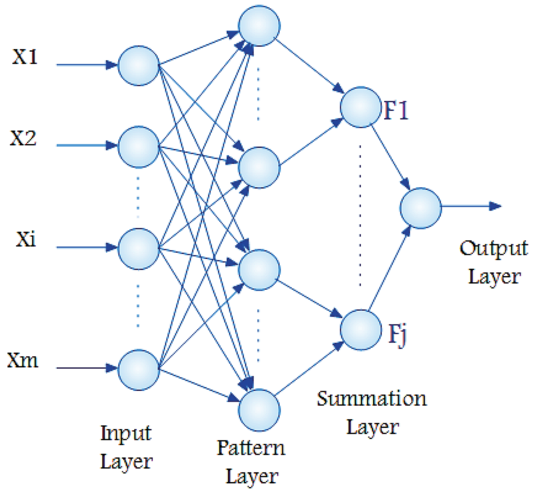

Figure 8. Probabilistic Neural Network architecture

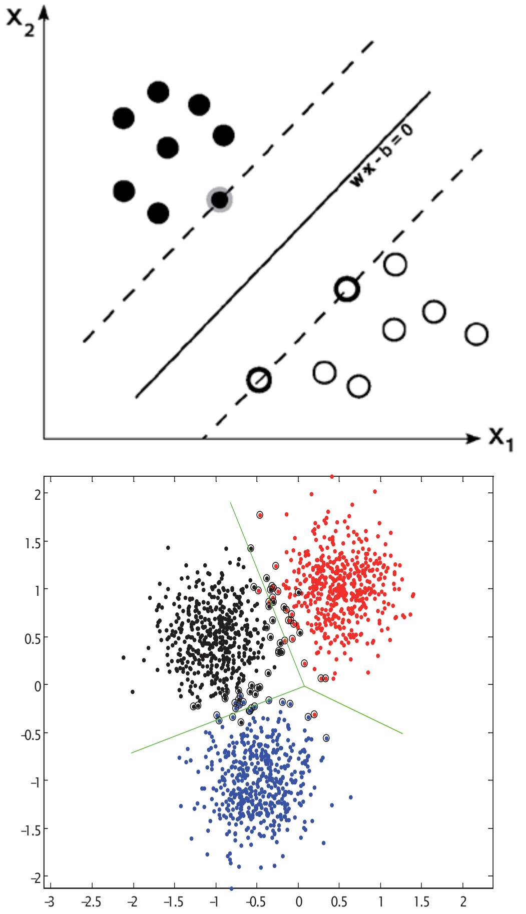

Figure 9. Two-class and three-class dataset separation using linear SVM.

function from these $N(N-1)/2$ classifiers using some voting scheme. Generally, SVM classifier performs the pattern recognition by mapping the original lower dimensional input space into a higher dimensional feature space via a nonlinear kernel function. The effectiveness of SVM classifier depends on selection of the best kernel function and its parameters values to improve the computational efficiency [29]. The most commonly used kernels for nonlinear pattern recognition are polynomial, Gaussian radial basis function, exponential radial basis function, multi-layer perceptron and many more including Fourier, splines, B-splines, additive kernels and tensor products. In this work, we use Gaussian radial basis function as the kernel function to classify multi-class dataset. Gaussian radial basis kernel function is defined as Eq. (10):

$$k(X_i, X_j) = \exp(-\gamma \left\| X_i - X_j \right\|^2) \qquad (10)$$

where, $\gamma = 1/2\sigma^2 > 0$ (In this work, we select $\gamma = 0.7$).

VI. Exprimental Results

We use Matlab software for simulation. All of experiments have been done by a computer with CPU speed of 2.5GHz and 2GB RAM.

A. Database:

McGill [31] dataset is used to verify accuracy of the proposed method. This dataset consists of 19 classes where each class includes different number of object samples. In fig. 10, samples of different shapes have been shown. The different classes are: Airplane, Ant, Bird, Chair, Crabs, Cup, Dinosaur, Dolphin, Fish, Four, Hand, Human, Octopus, Plier, Snake, Spectacle, Spider, Table and Teddy doll. This dataset includes both articulated (Ant, Spider, …) and non-articulated (Table, Cup, …) objects with different number of 12 to 31 objects in each class and total number of 456 objects in all classes.

We evaluate efficiency of the proposed geometric features in both categorization and retrieval scheme, which are described as follows:

B. Categorization scheme:

In categorization scheme, the objective is to define the class of object. Since, both of the classifiers i.e. PNN and SVM are supervised, we choose empirically 70 percent of models in each class as training data and the remaining 30 percent as test data. We have modified the number of training and test data in comparison to our previous work [32] to improve recognition rate. Mutual Euclidean distances for each pair of vertices, vertex-to-center distance and deviation angle of vertex normal from center-to-vertex vector are applied in 1D and 2D histogram form to PNN and SVM classifiers. The best Correct Classification Rate (CCR) obtained from PNN and SVM for the proposed features are achieved as %71.57 and %76.75, respectively. To implement SVM classification algorithm, we have used LIBSVM Toolbox [33] that employs one-versus-one classification method for multi-class problems. Table 1 illustrates accuracy of classification for different features that are used in separate and combination form. To compare with other approaches, we consider the methods that automatically classify 3D objects considering either the object geometry or its structure or both of them. For example, in [34], objects are described in terms of graph descriptors and a method is used for generation of a class representative, while some other methods use cognitive on geometric properties, like curvature, orientation and planarity [35]. Moreover, the method proposed in [36] divides

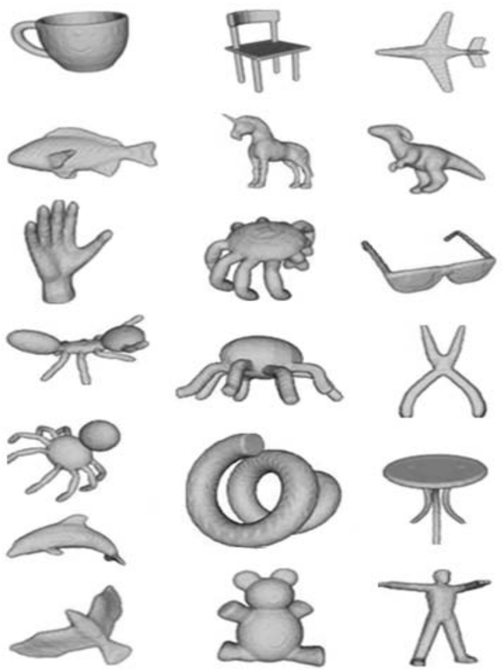

Figure 10. Sapmles of McGill 3D object database

the objects into sub-parts that are grouped into part classes using agglomerative clustering as a bottom-up hierarchical clustering method. We compare our work with an official workbench consisting of three different shape classifiers that are defined as the distance function between a query object and another objects in database and each of these classifiers has its own properties. Consequently, these properties are discussed and compared together. The classification process includes two main phases, choosing the best shape signature (descriptor) for describing each model and introduction of a suitable (dis-) similarity measure between descriptors. Object classification by shape signatures is formally expressed as follow:

1. The models are signified in the database by different kind of shape signatures $S_i, i = 1,2,3,...$ and the query by a signature q;

2. Signature representation is used to categorize the query, i.e. select the class which minimizes distance between the query and considered class.

We compare our work with three different definitions of distance: the Maximum Distance Classifier, the Centroid Distance Classifier and the Atypicity Distance Classifier. In table 2, the result of different classification methods for different extracted features have been shown for McGill dataset. These features [37] are Spherical Harmonics (SH), Light Field Descriptor (LFD) and Extended Reeb Graph (ERG). Spherical Harmonic Descriptor (SHD) [2] is a geometry-based representation of a given 3D object, which is invariant to rotations. It is obtained by recording the variation of the shape over a set of concentric spherical shells. These variations are captured by the norm of the spherical harmonic coefficients of appropriately defined shape functions. Light Field Descriptor (LFD) [38] represents a given 3D object using histograms of 2D silhouette or gray-scale images of the 3D object captured from a set of virtual cameras uniformly placed on an enclosing sphere. Dissimilarity between two 3D objects given by distance between two descriptors is minimized over all rotations with respect to the spheres. The main idea behind this descriptor is to define shape similarity based on projected visual images. Reeb graph [39] can be obtained assuming a function calculated over the 3D object surface. Reeb graphs are compact shape descriptors which convey topological information related to the level sets of a function defined on the 3D object.

As shown in table 2, the proposed method is compared with other classification method such as Maximum Distance Classifier (MaxDC), Centroid Distance Classifier (CDC) and Atypicity Distance Classifier (ADC). The MaxDC and CDC classifiers are introduced as Eq. (11), respectively:

$$\begin{aligned} \tilde{d}(q,C_k) &= \max\nolimits_{C_{\bar{k}}\in D} d(q,C_k) \\ d(q,C_k) &= \frac{\sum_{s\in C_k} d(q,s)}{|C_k|} \end{aligned} \tag{11}$$

Where, denominator is the cardinality of the class. The main disadvantage of the previous considered classifiers is the need of calculating of all the comparisons between query object and other objects in database, which motivates the introduction of the following Atypicity Distance Classifier (ADC).

TABLE I. COMPARISON OF FEATURES EFFICIENCY IN TWO CLASSIFICATION MODES, PNN AND SVM (CORRECT CLASSIFICATION RATE IN %).

Classifiers Features	Probabilistic Neural Network	Support Vector Machine
1D-Histogram of mutual Euclidean distance	59.05	71.79
2D-Histogram of the angle between vertex normal vector and vertex-to-shape center vector	64.48	68.31
Both features combination	**71.57**	**76.75**

TABLE II. DIFFERENT CLASSIFICATION METHOD COMPARISON RESULTS (CORRECT CLASSIFICATION RATE IN %)

	Features Classifiers	SH	LFD	ERG	Proposed features Combination
Different Classification Methods	**MaxDC**	33	38	38	54.45
	CDC	64	68	60	65.21
	ADC	63	68	58	65.32
	PNN used in [32]	67.50	68.32	65.50	68.42
	PNN used in this paper	68.33	68.98	69.88	**71.57**
	Support Vector Machine	68.60	65.35	67.43	**76.75**

structure. For this measure, the number of comparisons to be performed at run time is reduced by selecting a representative model for each class:

$$\tilde{d}(q, C_k) = \tilde{d}(q, \bar{s}) \qquad (12)$$

In fig. 11 and 12, we have shown two general correct classification rate charts to compare different classifiers and shape descriptors. In both methods, the correct classification rates are averaged over the shape descriptors and classifiers. The correct classification rates of different classifiers are compared in different colored bars and as they illustrate, PNN and SVM classifier in most of the classes, especially in term of our proposed features, have better performance in comparison with the other ones.

In table 3, confusion matrix of both PNN and SVM classifiers are presented in which its values show the classification accuracy for each pair of classes in percent. Also, they illustrate that how much two classes are similar in each classifier. For PNN and SVM classifiers, as shown in this table, some groups of McGill database such as chairs, spiders, Fishes, Pliers, Spectacles and Teddy are separated better than the other groups.

C. *Retrieval scheme:*

It would be very interesting to compare and demonstrate the effectiveness of our proposed features in retrieval performance scheme on the McGill database. Therefore, we report the results of our retrieval experiments in fig. 13 in term of Precision-Recall curves (PRc) based on Euclidean distance. In information retrieval framework, the precision and recall quantity are described in term of retrieved data.

- ***Precision and Recall Concepts Definition:***

Precision is the fraction of retrieved data that are relevant to the search data while Recall is the fraction of the data that are relevant to the query data that are successfully retrieved. They are computed as eq. (13):

$$\Pr = \frac{M \cap R}{R}, \ \mathrm{Re} = \frac{M \cap R}{T} \qquad (13)$$

where, *M, R* and *T* are the number of relevant, retrieved and total relevant of 3D object, respectively.

D. *Speed Evaluation:*

Besides improving the accuracy of the proposed categorization system, we compare our work in term of speed of both feature extraction and classification steps to other methods. The average time of feature extraction and classification of each 3D model in our computer system is shown in table 4. According to this table, in most of cases, the average time of both feature extraction and classification of our proposed features and classifiers is lower than the others. To clarify the comparison of different descriptors and classifiers in term of time, the speed of calculation for both feature extraction and classification, are shown in fig. 14 as colored bars chart for each feature and classifier, separately.

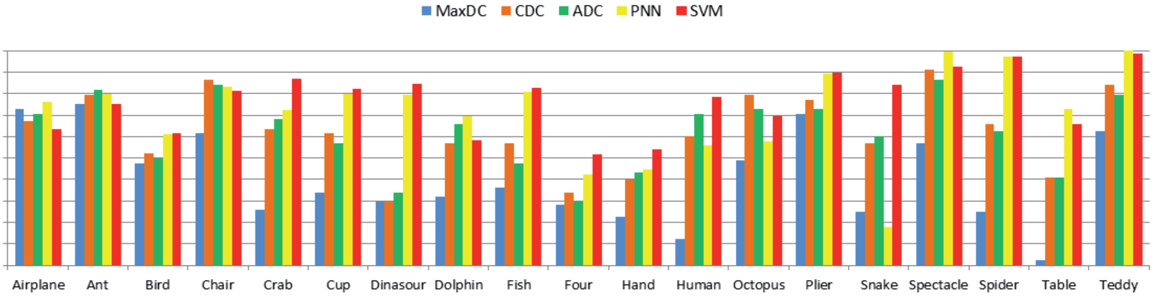

Figure 11. Correct Classification Rate (CCR) of different classifiers

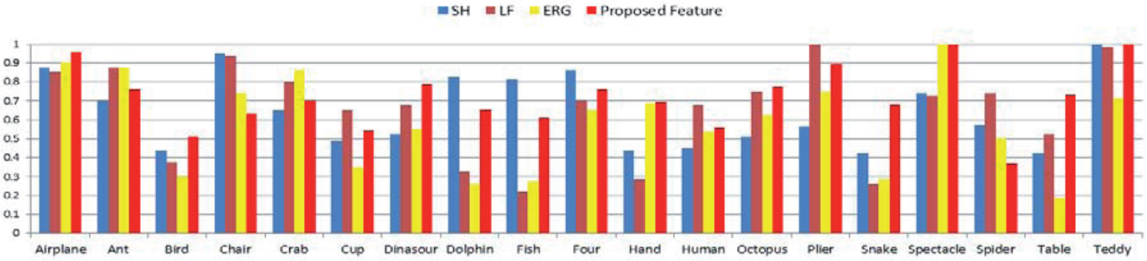

Figure 12. Correct Classification Rate (CCR) of different shape descriptor

TABLE IV.OBTAINED RESULTS OF CLASSIFICATION FOR PNN AND SVM IN CONFUSION MATRIX FORM (IN %), WHICH ROWS CORRESPOND TO REFERENCE WHILE COLUMNS GIVE THE PREDICTED VALUES OF THESE TWO CLASSIFIERS (WITH MAJORITY PREDICTION IN BOLDFACE AND COLORED SHAPE)

PNN SVM	Airplanes	Ants	Birds	Chairs	Crabs	Cups	Dinosaurs	Dolphins	Fishes	Fours	Hands	Humans	Octopus	Pliers	Snakes	Spectacles	Spiders	Tables	Teddy
Airplanes	76	1.33	22	0	0	0	0	0	0	0	0	0	0	0.67	0	0	0	0	0
	63.67	0	14.33	0	0.33	0	0	0	0	0	0	0.33	0	0	16	0	0	5.33	0
Ants	8	80	0	0	0	0	8.33	0	0	0	0.33	1.67	0.33	0	0	0	1	0.33	0
	15.33	75	5	0	0	0	0	1	0	0	0.33	2	0	0	0	0	1	0.33	0
Birds	10.67	0	61.33	0	15	0	2	0	0	1.67	0	1	1	0	0	0	7.33	0	0
	7.33	0	61.67	0	11	0	0	0	0	8.67	0	0.67	0.33	0	10	0	0.33	0	0
Chairs	5	1	0	83	0	0	0	0	0	0	10.33	0	0	0	0	0	0	0	0.67
	7.67	0	0	81.33	0	0	0	0	0	1	9	0	0	0	1	0	0	0	0
Crabs	0	2.33	6.67	0	72.33	0	0	0	0	0	0.67	0	1	0	0	0	9	0	8
	0	2.67	0.33	0	87	0	0	0	0	1	0.33	4.67	0	0	1.33	0	2	0.67	0
Cups	0	0	0	0	0.33	79.67	7	0	1	0	0	6.67	0	0	3.33	0	0	1.67	0.33
	0	0	0	0	0	82.33	0	0	0	0	0	0	0	0	17.67	0	0	0	0
Dinosaurs	0	0	0	0	0	0	79.33	1	4	0	13.67	0	2	0	0	0	0	0	0
	0	0	0	0	0	0	84.33	2	0.33	0	11.33	0	1.33	0	0.67	0	0	0	0
Dolphins	0	0	0	0	0	0	7.33	69.67	12.33	5	0	5.67	0	0	0	0	0	0	0
	0	0	0	0	0	0	8.67	58.33	6.33	5.33	0.67	8	0	0	11.67	1	0	0	0
Fishes	0	0	0	0	0	0	0.33	18.67	81	0	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	3	10	82.67	0	4.33	0	0	0	0	0	0	0	0
Fours	0	2.67	1	0	0	0	8.33	10	0.33	42.33	3.33	4.67	0	0	0	0	20	0	7.33
	0.67	0	3	0	0	0	4	1	0	51.67	8	11.67	2.33	0	8.33	0	9.33	0	0
Hands	0	16.33	0	0	7.33	0	2.67	0	6.33	3	44.67	9	0	0	0	0	8.33	0	2.33
	4	9	0	0	8.67	0	3.33	9	3	0	54	4.67	0.33	0	3	0	0	1	0
Humans	3	9.33	6	0	0	0	0	8	0	1	2	56	0	0	0	0	4.33	10.33	0
	1.67	0	0.67	0	0	0	0	5.67	0	5.67	5.33	78.33	0	0	2.67	0	0	0	0
Octopus	0	1.33	4.67	0	3	0	1.67	0	0	0	6	0	57.67	0	0	0	4.33	0	21.33
	0	0	2.33	1	8	0	0	0	0	0	5.33	0	69.67	0	5.67	0	1.67	0	6.33
Pliers	3.33	0	0	0.67	0	0	0	0	6.33	0	0.33	0	0	89.33	0	0	0	0	0
	1.33	0	0	0	0	0	0	0	0	0	0	1	0	89.67	8	0	0	0	0
Snakes	0	0	0	0	0	26.67	0	12.67	2	0	0	0	0	2.33	18	19.33	0	2	17
	0	0	0	0	0	6	0	0	0	0	6.33	0	0	0	84	0	0	0	3.67
Spectacles	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.33	99.67	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7.33	92.67	0	0	0
Spiders	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97	0	3
	0	0	0	0	0.67	0	0	0	0	0	0	0	2	0	0	0	97.33	0	0
Tables	1.67	0	0.33	0	19	0	0	0	0	0	0	5.67	0	0	0.33	0	0	73	0
	1	0	0	0	1.33	0	0	0	0	0	0	9	0	0	22.67	0	0	66	0
Teddy	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100
	0	0	0	0	1	0	0	0	0	0	0	0	0.33	0	0	0	0	0	98.67

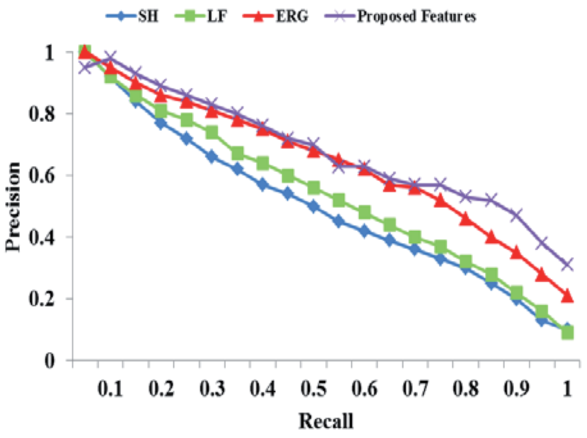

Figure 13. Precision-Recall (PR) curves based on Euclidean distance for different features extracted from McGill database of 3D objects

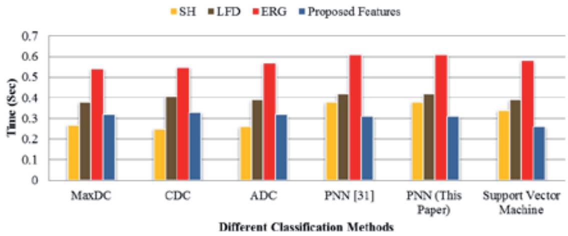

Figure 14. Average time of both feature extraction and classification process for each 3D object

TABLE III. THE AVERAGE TIME OF BOTH FEATURE EXTRACTION AND CLASSIFICATION STEPS OF EACH 3D MODEL (IN SECOND)

	Features Classifiers	SH	LFD	ERG	Proposed features Combination
Different Classification Methods	MaxDC	0.27	0.38	0.54	0.32
	CDC	0.26	0.41	0.55	0.33
	ADC	0.26	0.39	0.57	0.32
	PNN [32]	0.38	0.42	0.61	0.31
	PNN (This Paper)	0.38	0.42	0.61	**0.31**
	Support Vector Machine	0.34	0.39	0.58	**0.26**

VII. CONCLUSION

In this paper, we have used histogram of some geometric parameters as features to classify the 3D models and it has led to prominent reduction in computational cost and processing time. Furthermore, we have improved the correct classification rate using normal vector angle versus vertex distance to shape center in conjunction with the mutual Euclidean distance. In this paper, we have applied two kind of classifiers, i.e. PNN and SVM. Experimental result demonstrated the effectiveness of SVM against PNN, both in correct classification rate values and speed of classification.

In fact, in categorization scheme, the enhancement can be achieved in three parts:

- Employing more informative features of object can describe the object as more. Also, combining the features can make them to be more efficient as the features are approximately independent.
- Reducing the number of 3d object faces and vertices can speed up the feature extraction process. Moreover, employing the histogram of each parameter as the feature vector can reduce its dimension and consequently computational cost. Hence, we can reduce calculation time.
- Employing the better classification method can increase the accuracy of categorization, which this end can be achieved by using powerful classifiers like PNN and specially SVM.

In retrieval scheme, since we have to find the more similar objects of dataset to the query object, we can't use the classification tools like PNN and SVM and we have to define similarity of two objects as the easy distance formulations. Therefore, in this scheme, the enhancement can be achieved only in first two parts.

As it is clear, using histograms of some parameters are not the only way of feature extraction in 3D object categorization and retrieval. In our future works, we will try other efficient geometric features and employ feature selection algorithms as the standard methods of dimension reduction and outlier removal to improve the result of recognition rate.

REFERENCES

[1] Zh. Biqiang, X. Yuan and Y. Xueyu. "Fast generation of the topological information in STL for mesh simplification," *Journal of Shanghai Jiaotong University,* vol. 38, pp. 39-42, 2004.

[2] M. Kazhdan, T. Funkhouser and S. Rusinkiewicz. "Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors," *Symposium on Geometry Processing,* pp. 156-164, 2003.

[3] R.Osada, T.Funkhouseri, B.Chazelle and D.obkin. "Matching 3d shapes with shape distribution," *Proc. of the Shape Modeling International,* pp. 154–166, 2001.

[4] P. Shilane, P. Min, M. Kazhdan and T.Funkhouser. "The Princeton shape benchmark," *Proc. of Shape Modeling International,*2004.

[5] Mpeg Requirements Group. "Overview of the MPEG-7 Standard (version 3.0)," *Doc. ISO/MPEG N3445, MPEG Geneva Meeting,* 2000.

[6] Mpeg Video Group. "MPEG-7 Visual part of experimentation Model (version0)," *Doc.SO/MPEG N3914, MPEG Isa Meeting,* 2001.

[7] E. Paquet and M. Rioux. "A query by content software for three dimensions database management," *Proc. of International Conference on Recent Advances in 3D Digital Imaging and Modeling*, Ottawa, Canada, 1997, 345-352.

[8] E. Paquet and M. Rioux. "A content-based search engine for VRML database." *Proc. of the 1998 Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 541-546, 1998.

[9] M. Ankerst, G. Kastenmller and H. Kriegel. "3D Shape Histograms for Similarity Search and Classification in Spatial database." *Proc. of the 6th International Symposium on Large Spatial Database*, Hong Kong, pp. 207-226, 1999.

[10] T. Suzuki Motofumi, T. Kato and N. Otsu. "A Similarity Retrieval of 3D Polygonal Model Using Rotation Invariant Shape Descriptors." *Proc. of IEEE International on Systems, Man, and Cybernetics*, Nashville, Tennessee, vol. 4, pp. 2946-2952, 2000.

[11] H. Shum, M. Hebert and K. Ikeuchi. "On 3d shape similarity." *Proc. of CVPR*, pp. 526–531, 1996.

[12] M. Hilaga, Y. Shinagawa, T. Kohmura and T.L. Kunii. "Topology matching for fully automatic similarity estimation of 3d shapes." *SIGGRAPH*, pp. 203–212, 2001.

[13] J. Zhang, K. Siddiqi, D. Macrini, A. Shokoufandeh and S. Dickinson. "Retrieving articulated 3d models using medial surfaces and their graph spectra." *Int. Workshop on Energy Minimization Methods in CVPR*, pp. 285-300, 2005.

[14] R. Osada, T. Funkhouseri, B. Chazelle and D. Dobkin. "Matching 3d models with shape distribution." *Proc. of Shape Modeling International*, pp. 154–166, 2001.

[15] A. Elad and R. Kimmel. "On bending invariant signatures for surfaces." *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 25 no. 10, pp. 1285-1295, 2003.

[16] R. Osada, T. Funkhouseri and B. Chazelle. "Shape Distributions." *ACM Transactions on Graphics*, vol. 21, no. 4, pp. 807-832, 2002.

[17] D. Vranic and D. Saupe. "Description of 3D Shape Using a Complex Function on the Sphere." *Proc.of the IEEE International Conference on Multimedia and Expo*, Lausanne, Switzerland, pp. 177-180, 2002.

[18] A.Y. Ng, M.I. Jordan and Y. Weiss. "On spectral clustering: analysis and an algorithm." *Advances in Neural Information Processing systems*, pp. 849–856, 2001.

[19] H. Zhang and R. Liu. "Mesh segmentation via recursive and visually salient spectral cuts." *Proc. of Vision, Modeling, and Visualization*, pp. 429–436, 2005.

[20] M. Carcassoni and E.R. Hancock. "Spectral correspondence for point pattern matching." *Pattern Recognition*, vol. 36, no. 1, pp. 193–204, 2003.

[21] L.S. Shapiro and B.J. Michael. "Feature based correspondence: an eigenvector approach." *Image and Vision Computing*, vol. 10, no. 5, pp. 283-288, 1992.

[22] V. Jain and H. Zhang. "Robust 3d shape correspondence in the spectral domain" *Proc. of Shape Modeling International*, 2006.

[23] L. Wei and H. Yuanjun "3D Model Retrieval Based on Orthogonal Projections." *Proc. of the 9th International Conference on Computer Aided Design and Computer Graphics,* pp. 157-162, 2005.

[24] E. Paquet and M. Rioux. "Nefertiti: a Query by Content System for 3D Model and Image Databases Management." *Image and Vision Computing*, vol. 17, no. 2, pp. 157-166, 1999.

[25] J. Loffler. "Content-Based Retrieval of 3D Models in Distributed Web Databases by Visual Shape Information." *IEEE International Conference on Information Visualization*, pp. 82-87, 2000.

[26] D. Y. Chen and M. Ouhyoung. "A 3D Object Retrieval System Based on Multi-Resolution Reeb Graph." Proc. of Computer Graphics Workshop, pp. 16-20, 2002.

[27] M. Petrou and P. Bosdogianni. "Image Processing: The Fundamentals." *John Wiley*, 2th Ed., 2010.

[28] A. Belyaev, "Mesh Smoothing and Enhancing. Curvature Estimation,"http://www.mpi-inf.mpg.de/~ag4-gm/handouts/06gm_surf3.pdf

[29] V. Vapnik. "The Nature of Statistical Learning Theory." *Springer-Verlag*, 1st Ed., 1995.

[30] U. H. G. Kressel. "Pairwise Classification and Support Vector Machines." *In Advances in Kernel Methods – Support Vector Learning, The MIT Press*, pp. 255-268, 1999.

[31] Center for intelligent machines and school of computer science. "McGill 3D Shape Benchmark." 2005.

http://www.cim.mcgill.ca/~shape/benchMark/

[32] M.Ramezani and H.Ebrahimnezhad. "3D Object Categorization Based on Histogram of Distance and Normal Vector Angles on Surface Points," *7th Iranian Conference on Machine Vision and Image Processing* , Tehran, Iran, pp. 1-5, 2011.

[33] Ch. Chang and Ch. Lin. "LIBSVM a library for support vector machines." *ACM Trans. on Intelligent Systems and Technology*,

http://www.csie.ntu.edu.tw/~cjlin/libsvm

[34] K. Sengupta and K.L. Boyer. "Organizing large structural mordelbases." *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 4, pp. 321-332, 1995.

[35] P. Csakaky and A.M. Wallace. "Representation and classification of 3D objects." *IEEE Trans. on Systems, Man and Cybernetics - Part B: Cybern*, vol. 33, no. 4, pp. 638–647, 2003.

[36] D. Huber, A. Kapuria, R. Donamukkala and M. Hebert. "Part-based 3D object classification." *Proc. IEEE Conf. on Computer Vision and pattern Recognition*, vol. 2, pp. 82–89, 2004.

[37] S. Biasotti, D. Giorgi, S. Marini, M. Spagnuolo, and B. Falcidieno. "A Comparison Framework for 3D Object Classification Methods." *Multimedia Content Representation, Classification and Secuirity*, *Lecture Notes in Computer Science*, vol. 4105, pp. 314-321, 2006.

[38] D. Chen, M. Ouhyoung, X. Tian and Y. Shen. "On visual similarity based 3D model retrieval." *Computer Graphics Forum*, vol. 22, no. 3, pp. 223–232, 2003.

[39] S. Biasotti and S. Marini. "3D object comparison based on shape descriptors." *International Journal of Computer Applications in Technology,* vol. 23, pp. 57–69, 2005.

Mohammad Ramezani was born in Tehran, Iran in 1986. He received his B.Sc. degree in Electronic Engineering from Yazd University, Iran and his M.Sc. degree in Communication Engineering from Sahand University of Technology, Tabriz, Iran. His current research interests include 3D Image and Model Processing, Categorization and Retrieval. In addition, he has some experience in 3D Video Frame Compression.

Hossein Ebrahimnezhad was born in Iran 1971. He received his B.Sc. and M.Sc. degrees in Electronic and Communication Engineering from Tabriz University and K.N.Toosi University of Technology in 1994 and 1996, respectively. In 2007, he received his Ph.D. degree from Tarbiat Modares University. His research interests include image processing, computer vision, 3D model processing and soft computing. Currently, he is an assistant professor at Sahand University of Technology.