

Three-Dimensional Shape Recognition and Classification Using Local Features of Model Views and Sparse Representation of Shape Descriptors

Hussein Kanaan* and Alireza Behrad*

Abstract

In this paper, a new algorithm is proposed for three-dimensional (3D) shape recognition using local features of model views and its sparse representation. The algorithm starts with the normalization of 3D models and the extraction of 2D views from uniformly distributed viewpoints. Consequently, the 2D views are stacked over each other to form view cubes. The algorithm employs the descriptors of 3D local features in the view cubes after applying Gabor filters in various directions as the initial features for 3D shape recognition. In the training stage, we store some 3D local features to build the prototype dictionary of local features. To extract an intermediate feature vector, we measure the similarity between the local descriptors of a shape model and the local features of the prototype dictionary. We represent the intermediate feature vectors of 3D models in the sparse domain to obtain the final descriptors of the models. Finally, support vector machine classifiers are used to recognize the 3D models. Experimental results using the Princeton Shape Benchmark database showed the average recognition rate of 89.7% using 20 views. We compared the proposed approach with state-of-the-art approaches and the results showed the effectiveness of the proposed algorithm.

Keywords

Shape Classification, Sparse Representation, 3D Local Features, 3D Shape Recognition, View Cube

1. Introduction

Recently the rapid development of three-dimensional (3D) scanners and sensors as well as graphic accelerated hardware have attracted an enormous amount of interest for 3D image processing and analysis [1]. Nowadays, 3D shape models are used in various fields such as archaeology, cultural heritage, computer-aided design (CAD), medical applications, 3D object classification, and biometrics. For instance, in industrial applications, 3D CAD models are variously used in a new form of prototyping called digital prototyping. Digital prototyping allows for the easy evaluation of industrial components before fabrication. One of the most important applications in the field of medical diagnostic is the automatic analysis and recognition of abnormalities in the anatomic structures. The recognition of abnormalities in 3D images requires automatic representation and the classification of 3D models, which is a challenging issue.

The recognition and classification of 3D models are of great importance in several applications.

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Manuscript received May 17, 2017; first revision April 18, 2018; accepted May 31, 2018.

Corresponding Author: Alireza Behrad (behrad@shahed.ac.ir)

* Dept. of Electrical Engineering, Shahed University, Tehran, Iran (dr.kanaan1984@gmail.com, behrad@shahed.ac.ir)

However, existing approaches for the classification and recognition of 2D images cannot be applied to 3D images directly because of its different nature and characteristic. The recognition and classification of a 3D model require the extraction of global or local characteristics of a 3D shape and its representation in an appropriate approach that is called a 3D shape descriptor or signature. The performance of a 3D shape recognition algorithm highly depends on the appropriateness of a 3D shape descriptor.

Existing algorithms for 3D shape representation can be roughly categorized into feature-based, view-based and graph-based approaches [2,3]. In feature-based approaches, global or local features of a 3D shape model are used as the descriptors for shape classification. Most early approaches for 3D shape classification employ global features for shape recognition. Global features are easy and fast methods for 3D shape representation; however, they cannot describe the local deformation of a 3D shape. Therefore, the local features have been recently considered as a solution to handle the shortcomings of global features and increase the discriminating power of shape descriptors.

In comparison with feature-based approaches that consider only the geometric characteristic of a 3D shape, graph-based approaches employ the relations and linkages between the various components of a model for the shape representation. In other words, graph-based approaches construct a graph representing the topological information of a 3D object as a descriptor. In general, graph-based approaches are computationally inefficient and their applications are mostly restricted to CAD/CAM models. Two similar 3D models are looked similar from various views, therefore view-based approaches extract the 2D images of a 3D model from various views to describe it. The number of views and the method to combine views have a major impact on the effectiveness of a view-based algorithm.

This study proposes a new approach for 3D shape classification and recognition using 3D local features of model views. The proposed approach can be considered as a combination of view-based and feature-based approaches. The algorithm commences with the normalization of 3D models. This stage makes the extracted views invariant into rotation and scale changes. Then, 2D images of 3D models are captured from various views. In contrast to existing view-based approaches that mostly extract the silhouette of 3D models, we consider the distance of points from the center of mass of 3D models as the intensity of the extracted views. Consequently, the 2D views of the 3D models are stacked over each other to form the view cubes. The algorithm employs the descriptors of 3D local features in the view cube after applying Gabor filters in various directions. To combine the descriptors of local features, various descriptors from different classes of 3D models are saved during the training stage. These descriptors are called the prototype dictionary of local descriptors. To extract an intermediate feature vector, we measure the similarity between the local descriptors of an input model and the local features of the prototype dictionary. We represent the intermediate feature vectors of 3D models in the sparse domain to obtain the final descriptors of models. Finally, support vector machine (SVM) classifiers are used for the recognition of 3D models. In brief, the main contributions of this paper are as follows:

- A new approach is proposed to represent a 3D model as a cube of model views. In contrast to most view-based algorithms that use the silhouette of a shape model as the view image, our algorithm utilizes grey-level images. The pixel intensities of the grey-level images are obtained by the calculation of distances of points from the center of mass of the shape model.
- The proposed approach leverages a new descriptor that is extracted from 3D local descriptors of the view cube. Therefore, the proposed algorithm can be considered as a combination of view-based and feature-based approaches.
- A new approach is proposed to represent the descriptor in the sparse domain.

The remainder of this paper is organized as follows. In Section 2, a survey of related work is presented. Section 3 introduces the proposed algorithm for 3D shape recognition. Section 4 presents the experimental results and their analysis and we conclude the paper in Section 5.

2. Related Work

Various applications of 3D shape recognition and classification have attracted a large amount of attention from researchers during the past decade. Therefore, several approaches have proposed for 3D shape classification during the past decade. Also, several survey papers have discussed various algorithms for 3D shape recognition, classification, and retrieval [3-6]. As mentioned before, early approaches for 3D object representation mostly focused on global features like area, volume and moments of 3D shapes [7-10]. For Instance, Elad et al. [7] calculated moments of 3D shape and represented them as a descriptor for 3D shape retrieval. Osada et al. [8] used shape a distribution sampled from a shape function measuring the global geometric properties of an object as the shape descriptor. An enhanced version of these features was also used in [9].

Some existing approaches for 3D shape recognition employ descriptors based on a spatial map that uses the spatial relations between different sections of a 3D model [11-13]. For instance, Saupe and Vranic [11] used spherical harmonic coefficients as a shape descriptor. The descriptor measures the maximal extent of shape across all rays from the origin. Spatial maps are generally sensitive to the geometric transformations like the scale and rotation of a 3D object; therefore, a pose normalization stage is inevitable. Kazhdan et al. [12] used a spherical harmonic representation that transforms a rotation-dependent shape descriptor into the rotation-independent one. However, spherical harmonics that are based on the latitude-longitude parameterization of a sphere result in singularities in the poles. To handle the problem, Laga et al. [13] uniformly sampled points on the sphere and used the wavelet transform to extract the spherical wavelet descriptors that are the extended version of Zernike moments and spherical harmonics.

Recent feature-based approaches for 3D shape representation mostly focus on local features [14-19]. Descriptors based on local features generally give rise to better discrimination for inter-class shape recognition. The main stage of constructing a descriptor based on local features is the detection of salient points in a 3D shape model. Shilane and Funkhouser [14] defined the distinctive regions of a 3D shape as the areas that their shapes are consistent with the objects of the same type and differ from the objects of other types. Inspired by the characteristics of human visual perception, Zhao et al. [15] used two features called retinex-based importance feature (RIF) and relative normal distance (RND) for salient point extraction. Atmosukarto et al. [16] used the histogram of low-level features like Besl–Jain and Gaussian curvatures for salient point extraction by using a trained SVM. Salient points are then transformed into a 2D longitude-latitude spatial map as a descriptor to classify 3D models. In [17], an extension of Harris method is used to extract the salient points of a 3D shape. The method uses an adaptive technique to determine the neighborhood of a vertex to apply Harris operator. Bu et al. [18] employed deep belief networks (DBN) to learn high-level features that are obtained from local descriptors. They adopted a scale-invariant heat kernel signature and average geodesic distance as the local descriptors that are robust against non-rigid and complex shape deformations.

Graph-based approaches are another group of algorithms for 3D shape recognition [20-24]. In contrast

to feature-based methods that describe 3D objects using 3D geometrical properties, graph-based algorithms use the topological information and the relation between various parts of the object for 3D shape recognition. A graph shows how various parts of an object are linked together. Various graph representations have been proposed in the literature, such as skeletal graph [20], Reeb graph [21], and spectral Reeb graph [23,24], to name a few. Graph-based algorithms are computationally expensive and sensitive to small topological changes.

Another category of 3D shape recognition algorithms, which are called view-based algorithms, capture several 2D images from various directions of a 3D model [25-27]. In [25], a set of 2D images is automatically generated from a 3D object, by taking the views from uniformly distributed viewpoints. Then, a set of 2D rotation-invariant shape descriptors are extracted for each image. Finally, a similarity measure is used for 3D model retrieval. In [26], convolutional neural networks (CNNs) are used for 3D shape recognition using multiple views of a 3D model. The method first trains a set of CNNs to combine several views as a single and more informative view. Finally, another trained CNN is used to generate the shape descriptor. Most view-based methods can be considered as an extension to global feature-based methods that extract a global descriptor from the view images. However, Ding and Liu [27] used a view-based descriptor called sphere image for 3D shape retrieval that takes into account the relation between various views by constructing a star graph. In [28], a feature fusion method and multi-modal graph learning are used for view-based 3D object retrieval. After extracting different visual features, including 2D Zernike moments, 2D Fourier descriptors, and 2D Krawtchouk moments, the Hausdorff distance is computed to measure the similarity between two 3D objects with multiple views. Finally, several graphs are employed for the feature fusion task.

3. Proposed Method

This study aims to classify 3D objects using the local features of model views and the sparse representation of the extracted features. Fig. 1 shows the general block scheme of the proposed algorithm. The proposed method includes two parts comprising training and testing phases. The proposed algorithm is a combined feature and view-based approach that utilizes the 3D local features of model views to construct the required features for 3D shape classification. Both training and testing phases start with the normalization of 3D models. This stage makes the extracted views invariant into the rotation and scale changes. Then a set of 2D images is extracted from 3D shapes by taking views from uniformly distributed viewpoints. Consequently, the 2D views of the 3D models are stacked over each other to form view cubes. We use 2D Gabor filters in companion with a 3D max filter to extract proper characteristics of the view images and take into account the relation between various views. In the training phase of the system, we select some 3D local features from the training set of 3D models as the prototype dictionary of local features. Both in the training and testing phases, the intermediate features of 3D models are computed by calculating the similarity between 3D patches of a 3D model and the prototype dictionary of local features. We use sparse representation to calculate the final descriptors for the classification. The sparse representation of descriptors needs the selection of appropriate basic signals or atoms. This stage that is called dictionary learning is conducted during the training phase. This dictionary is then used to obtain the final descriptors for 3D shape classification via sparse representation. Then the trained SVM classifiers are used for 3D shape recognition.

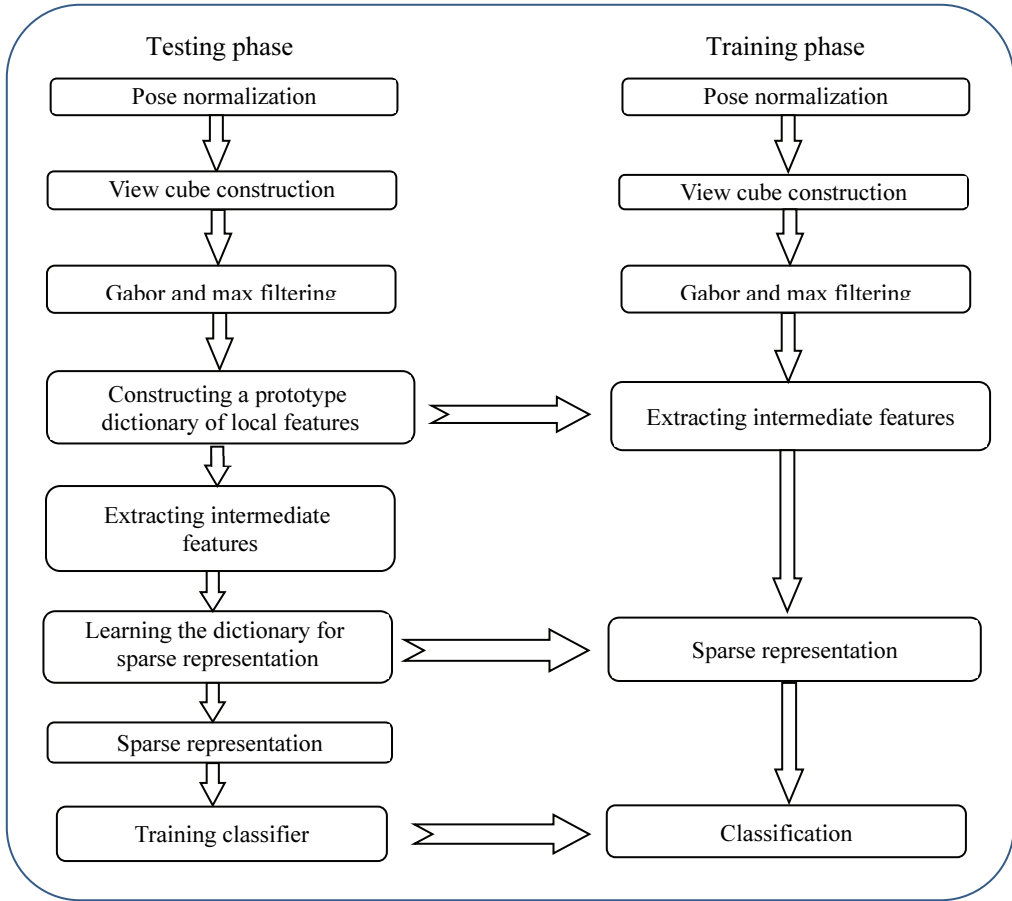


Fig. 1. The general block scheme of the proposed algorithm.

3.1 Pose Normalization

The 3D shape models may contain an arbitrary scale, orientation, and position. To make the extracted features robust against the scale, rotation, and position of a model, a pose normalization stage is used before the feature or view extraction. Pose normalization algorithms aim to transform a 3D shape into a new canonical coordinate frame where the representation of the shape is independent of its scale, orientation, and position. Various normalization algorithms such as weighted principal component analysis (PCA) [29], continuous PCA [30] and PCA on the normals of the model (NPCA) [31] have been proposed in the literature. We use the weighted PCA [29] approach for the pose normalization.

In the weighted PCA, the mean vector and covariance matrix of vertex coordinates are calculated as follows:

$$\mathbf{m}_V = \frac{1}{n} \sum_{i=1}^n w_i \mathbf{v}_i \quad (1)$$

$$\mathbf{C}_V = \frac{1}{n} \sum_{i=1}^n w_i (\mathbf{v}_i - \mathbf{m}_V)^T (\mathbf{v}_i - \mathbf{m}_V) \quad (2)$$

where n is the number of vertices in a 3D shape model, \mathbf{v}_i are the coordinates of the shape vertices and \mathbf{m}_V and \mathbf{C}_V are the mean vector and covariance matrix of the vertex coordinates, respectively. Here ω_i is the weight of vertex \mathbf{v}_i that is defined as the proportion of the sum of surfaces of all triangles that have \mathbf{v}_i as a vertex to the sum of surfaces of all triangles in the shape. Let \mathbf{A} be a matrix consisting of the ordered eigenvectors of the covariance matrix \mathbf{C}_V , as the row vectors. The normalized coordinates of the vertices are then calculated as follows:

$$\hat{\mathbf{v}}_i = \mathbf{A}(\mathbf{v}_i - \mathbf{m}_V) \tag{3}$$

Fig. 2 shows the result of the pose normalization stage on some 3D shape models.

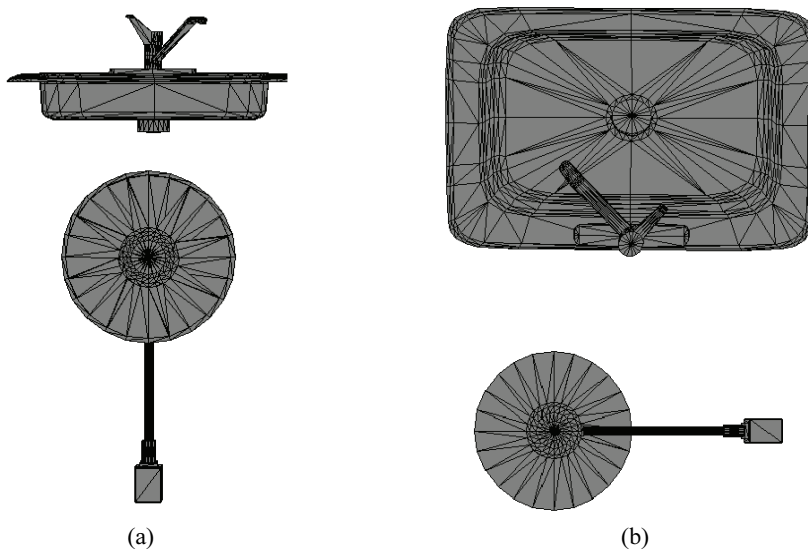


Fig. 2. Results of pose normalization on two typical 3D shapes: (a) before applying the pose normalization and (b) after applying the pose normalization.

3.2 View Cube Construction

After the normalization of 3D shape models, different views of the shape models are extracted to construct view cubes. To build a view cube, a set of 2D images is extracted from a 3D shape by taking views from uniformly distributed viewpoints. The view cube is constructed by stacking of various views of a shape model. The aim of constructing a view cube is to take into account the relation between various views of a shape model for the extraction of the shape descriptor. In the proposed approach, not only the 3D information of a view cube is used in the filtering steps, but also the intermediate features are based on the 3D information of adjacent views. Fig. 3 shows the extracted views and the constructed view cube for a typical 3D shape model with nine views. In contrast to the most view-based algorithms that use the silhouette of a shape model as a view image, we use the distances of points from the center of mass as the intensities of pixels of a view image. This feature provides us the capability of using intensity-based filters such as Gabor filter to extract the proper characteristics of a shape model from its view images.

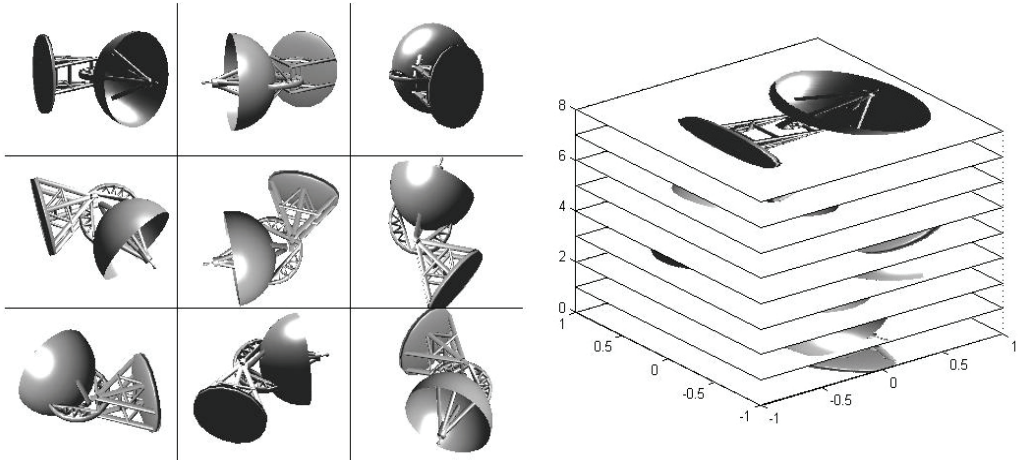


Fig. 3. The extracted views and the constructed view cube for a typical 3D shape model.

3.3 Gabor and Max Filtering

The use of distances of points from the mass center of shape models as the intensity of view images provides us the capability of utilizing filters such as Gabor filter for the texture-based feature extraction. Inspired by the biological characteristics of the human visual system, Gabor filters are found to be appropriate for texture representation and discrimination. The weights of 2D Gabor filter are defined as follows:

$$\hat{x} = x \cos(\theta) - y \sin(\theta) \quad (4)$$

$$\hat{y} = x \sin(\theta) + y \cos(\theta) \quad (5)$$

$$G(x, y) = \exp\left(-\frac{\hat{x}^2 + \gamma^2 \hat{y}^2}{2\sigma^2}\right) \cos\left(\frac{2\pi}{\lambda} \hat{x}\right) \quad (6)$$

where θ denotes the direction of the Gabor filter, γ is the aspect ratio, σ is effective width and λ is wavelength. Our approach for texture representation using Gabor filter is based on the method of [32]. We apply 11×11 pixels 2D Gabor filters to all 2D images of view cube in 16 directions and select the maximum value in 16 directions as the output of Gabor filter. Taken all from [32], γ , σ , and λ are set to 0.3, 4.5, and 5.6, respectively.

After applying the 2D Gabor filter to the images of the view cube, 3D local max filter is employed to take into account the relations between various view images. To this intent, a 3D local max filter with the dimension of $5 \times 5 \times 5$ pixels is experimentally used in this study.

3.4 Extracting Intermediate Features

After applying Gabor and max filtering, the intermediate features are calculated by comparing all 3D local patches in the view cube by some previously-stored local patches, i.e., the prototype dictionary of local features. To this end, in the training stage, we randomly select N_i 3D local patches with a size of $16 \times 16 \times 3$ pixels. By comparing all the 3D local patches in view cube with the prototype dictionary of local features, intermediate features are calculated using the following algorithm:

- Select a 3D patch from the prototype dictionary of local features, i.e., $\mathbf{p}^j, j=1,2,\dots, N_i$.
- Slide a 3D window with a size of $16 \times 16 \times 3$ pixels in the view cube.
- Compare the 3D local patches of the view cube with \mathbf{p}^j as follows:

$$r_i^j = \exp(-\|\mathbf{x}_i - \mathbf{p}^j\|)$$

where \mathbf{x}_i defines i-th 3D local patch in the view cube and $\|\cdot\|$ denotes 2-norm.

- Calculate the max difference rm^j as follows:

$$rm^j = \max_i(r_i^j)$$

- Construct the intermediate feature vector, i.e., $\mathbf{r} = [rm^1, rm^2, \dots, rm^{N_i}]^T$.

3.5 Sparse Representation

In this study, the sparse representations of intermediate features are used as the descriptors for 3D shape recognition. In the sparse representation, a feature vector $\mathbf{r}_i \in R^{N_i}$, which denotes an intermediate feature vector, is represented as follows:

$$\mathbf{r}_i = \mathbf{D}\boldsymbol{\alpha}_i + \mathbf{e}_i \tag{7}$$

where $\boldsymbol{\alpha}_i \in R^k$ is the sparse representation of the feature \mathbf{r}_i , $\mathbf{D} \in R^{N_i \times k}$ is the dictionary or basic signals and $\mathbf{e}_i \in R^{N_i}$ is the error vector. Here k is the number of atoms or basic signals in the dictionary. The sparse representation of an intermediate feature vector \mathbf{r} , can be obtained by solving an underdetermined system of linear equations as follows:

$$\hat{\boldsymbol{\alpha}} = \arg \min \|\boldsymbol{\alpha}\|_0 \quad \text{s.t. } \mathbf{r} = \mathbf{D}\boldsymbol{\alpha} \tag{8}$$

where $\|\cdot\|_0$ denotes the number of nonzero elements in a vector. To make the solution of the Eq. (8) more feasible, the sparse representation problem may be expressed as:

$$\hat{\boldsymbol{\alpha}} = \arg \min \frac{1}{2} \|\mathbf{r} - \mathbf{D}\boldsymbol{\alpha}\|_2 + \lambda \|\boldsymbol{\alpha}\|_1 \tag{9}$$

where $\|\cdot\|_1$ refers to the L_1 norm. Since both \mathbf{D} and $\boldsymbol{\alpha}$ in the Eq. (8) is unknown, the sparse representation of local descriptors consists of two different stages: 1- dictionary learning and 2- the calculation of sparse codes. The aim of dictionary learning is the selection of appropriate basic signals or atoms. We use the K-SVD algorithm [33] in the training stage of the proposed algorithm to learn a dictionary for sparse representation. To learn a dictionary for the sparse representation, we use the intermediate feature vectors of the training shape models. Let \mathbf{R}_t denote the intermediate feature vectors of the training shape models and \mathbf{A}_t represent the corresponding sparse coefficients of \mathbf{R}_t as follows:

$$\mathbf{R}_t = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{n_t}\} \tag{10}$$

$$\mathbf{A}_t = \{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_{n_t}\} \tag{11}$$

where n_t is the total number of training 3D shape models. The calculation of matrix \mathbf{D} , i.e., dictionary using the K-SVD algorithm comprises the following steps:

Step 1. Initialize dictionary \mathbf{D} with L_2 normalized columns.

Step 2. Calculate sparse coefficients using the following equation:

$$\mathbf{A}_t = \arg \min_{\mathbf{A}_s} \{ \|\mathbf{R}_t - \mathbf{D}\mathbf{A}_t\| \} \quad \text{s.t.} \quad \|\mathbf{a}_i\|_0 \leq T_0 \quad i=1, 2, \dots, n_t \quad (12)$$

Step 3. Update the dictionary as:

$$\mathbf{D} = \arg \min_{\mathbf{D}} \|\mathbf{R}_t - \mathbf{D}\mathbf{A}_t\|_2 \quad (13)$$

Step 4. Repeat steps 2 and 3 until the convergence.

After the calculation of the dictionary, the feature-sign search algorithm [34] is used to calculate the sparse coefficients for intermediate features as the descriptors of 3D shape models.

3.6 Classification

In this study, SVM classifiers are used to classify 3D shape models using the extracted descriptors. An SVM is a binary classifier that categorizes an input feature vector by evaluating the classifier function $f(\mathbf{x})$ as follows [35]:

$$f(\mathbf{x}) = \text{sgn}(\boldsymbol{\omega}^T \varphi(\mathbf{x}) + b) \quad (14)$$

where \mathbf{x} is the feature vector, b and $\boldsymbol{\omega}$ are the bias and vector of SVM coefficients respectively, φ defines a kernel function, and sgn denotes the sign function. Since an SVM is inherently a binary classifier, one against one approach is used in this study to create a multi-class SVM classifier for 3D shape classification.

4. Experimental Results

The proposed algorithm was implemented in MATLAB environment and evaluated with two different 3D shape datasets comprising the McGill 3D shape benchmark (MSB) [36] and Princeton Shape Benchmark (PSB) databases [37]. Fig. 4 shows examples of 3D shape models for the MSB database. This database consists of 19 different kinds of 3D objects with the emphasis is on including models with articulating parts. Fig. 5 illustrates samples from the PSB database. This benchmark contains a database of 3D polygonal models collected from the World Wide Web. This dataset includes 1,814 models divided into three levels of classification. In our experiment, we used the 3rd level classification of the PSB database to assess the performance of the proposed algorithm and compare its results with those of other methods.

To construct a view cube, we use a set of nine 2D images with a resolution of 150×200 pixels. These images are extracted from 3D shapes by taking views from uniformly distributed viewpoints. In this study, the prototype dictionary of local features comprises 5,000 local patches of size $16 \times 16 \times 3$ pixels.

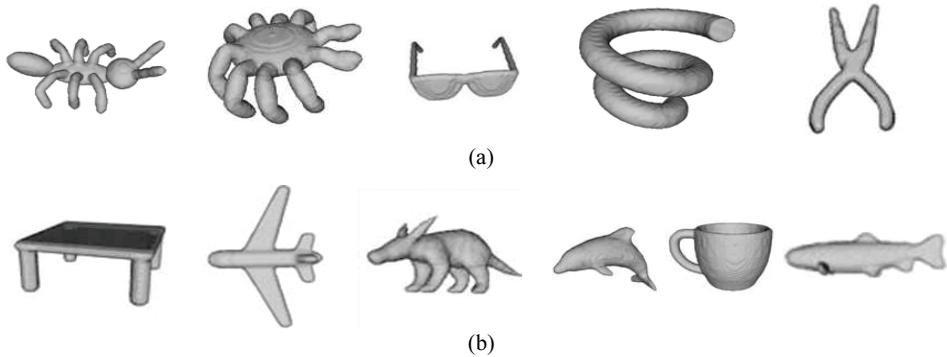


Fig. 4. Samples of 3D objects of the MSB database, (a) objects with articulating parts, (b) objects with moderate or no part articulation.

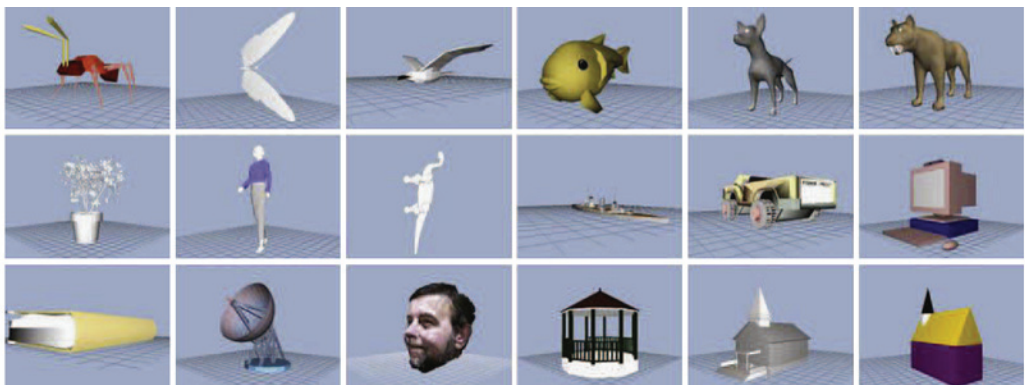


Fig. 5. Samples of 3D objects of the PSB database.

Fig. 6 shows the correct classification rate of the proposed algorithm on the MSB database. Our experimental results demonstrate that an SVM classifier with linear kernel function results in better classification accuracy. Therefore, one against one linear SVM classifiers are used in the experiments to create a multi-class SVM classifier. Fig. 6 illustrates the percentage of correct classification for the various dimensions of the sparse representation, i.e., k values. Since the MSB database does not specify test and train shape models for the classification, we use 190 random object models for the training and the remaining models for the test. The experiments are repeated ten times and the average classification rates are reported in the figure. The results of Fig. 6 demonstrate that the maximum allowable dimension of sparse representation, i.e., $k=190$, results in the maximum percentage of correct classification. Table 1 shows the correct classification rate for various classes of the MSB database using $k=190$. The results of Table 1 show the average classification rate of 83.7% and 100% for seven and 19 classes of the MSB database, respectively. Table 2 illustrates the confusion matrix for the classification of 3D objects in the MSB database. As the results of Table 2 demonstrate, some sources of errors for the proposed algorithm are related to the misclassification of airplanes as birds, tables as chairs and dinosaurs as four-limbs, to name a few. We also tested the proposed algorithm for 3D shape classification with and without the sparse representation. Table 3 compares the percentage of correct classification for the proposed algorithm with and without the sparse representation. The results of Table 3 show that the sparse representation of intermediate features enhances the classification rate up to 6.4%.

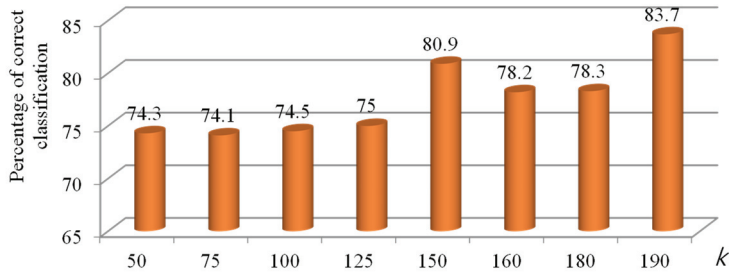


Fig. 6. Correct classification rates of the proposed algorithm on the MSB database for the various dimensions of the sparse representation i.e. k values.

Table 1. Correct classification rates (%) for various classes of the MSB database using $k=190$

Class name	Correct classification rate (%)
Airplanes	70.0
Ants	100
Birds	62.5
Crabs	80.0
Chairs	100
Cups	100
Dinosaurs	87.5
Dolphins	100
Fishes	80.0
Four-limbs	83.3
Hands	50.0
Humans	75.0
Octopus	70.0
Pliers	100
Snakes	100
Spectacles	84.6
Spiders	76.1
Tables	71.4
Teddy-bears	100
All classes	83.7

Table 2. Confusion matrix for the classification of 3D objects in the MSB database

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1. Airplane	0.70	-	0.20	-	-	-	0.10	-	-	-	-	-	-	-	-	-	-	-	-
2. Ants	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3. Birds	-	-	0.63	-	-	-	-	-	-	-	0.13	-	0.13	-	0.13	-	-	-	-
4. Crabs	-	-	-	0.80	0.05	-	-	-	-	-	0.05	-	-	-	0.10	-	-	-	-
5. Chairs	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6. Cups	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-
7. Dinosaurs	-	-	-	-	-	-	0.88	-	-	0.13	-	-	-	-	-	-	-	-	-
8. Dolphins	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-
9. Fishes	-	-	-	-	-	-	0.20	-	0.80	-	-	-	-	-	-	-	-	-	-
10. Four-limbs	0.08	-	-	-	-	-	0.08	-	-	0.83	-	-	-	-	-	-	-	-	-

Table 2. (Continued)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
11. Hands	-	-	-	-	-	-	-	-	-	-	0.50	0.20	0.10	0.10	0.10	-	-	-	-
12. Humans	-	-	-	0.08	-	-	0.08	-	-	-	0.08	0.75	-	-	-	-	-	-	-
13. Octopus	-	0.10	-	-	-	-	-	-	-	-	0.20	-	0.70	-	-	-	-	-	-
14. Pliers	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-
15. Snakes	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-
16. Spectacles	-	-	-	-	-	0.08	-	-	-	-	-	-	-	0.08	-	0.85	-	-	-
17. Spiders	-	-	-	0.05	-	-	-	-	-	-	-	-	0.19	-	-	-	0.76	-	-
18. Tables	0.14	-	-	-	0.14	-	-	-	-	-	-	-	-	-	-	-	-	0.71	-
19. Teddy-bears	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1

Table 3. Percentage of correct classification for the proposed algorithm on the MSB database with and without the sparse representation stage

	With the sparse representation	Without the sparse representation
Percentage orrect classification	83.7	77.3

To show the effectiveness of the proposed algorithm, we also compared the results of the proposed algorithm on the MSB database with those of method of Atmosukarto et al. [16]. This method uses the histogram of low-level features like Besl–Jain and Gaussian curvatures for salient point extraction. Salient points are then transformed into a 2D longitude-latitude spatial map as a descriptor to classify 3D models. Fig. 7 illustrates the correct classification rate of method of Atmosukarto et al. [16] on the MSB database using K nearest neighbors (KNN) classifier with various K values. The results of Fig. 7 show the maximum correct classification ratio of 65.47% for K=1. Table 4 illustrates the confusion matrix for the classification of 3D objects in the MSB database using Atmosukarto et al. method. A comparison between the results of Tables 2 and 4 shows the effectiveness of the proposed algorithm.

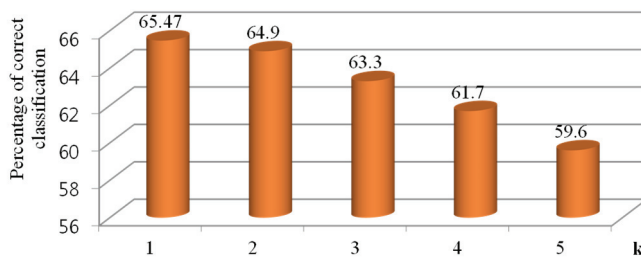


Fig. 7. Correct classification rates of method of Atmosukarto et al. [16] on the MSB database using KNN classifier for various K values.

Table 4. Confusion matrix for the classification of 3D objects in the MSB database using method of Atmosukarto et al. [16]

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1. Airplane	0.80	-	-	-	-	-	-	-	0.10	-	-	-	-	-	-	-	-	0.10	-
2. Ants	-	0.64	-	0.07	-	-	-	-	-	-	-	0.28	-	-	-	-	-	-	-
3. Birds	0.22	-	0.44	-	-	-	-	-	-	0.11	-	-	-	-	-	0.11	-	0.11	-
4. Crabs	-	-	-	0.84	-	-	0.08	-	-	0.08	-	-	-	-	-	-	-	-	-
5. Chairs	-	-	0.10	-	0.40	-	-	-	0.10	-	-	-	-	-	-	0.10	-	0.30	-

Table 4. (Continued)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
6. Cups	-	-	-	-	-	0.83	-	-	-	0.08	-	-	-	-	-	-	-	-	0.08
7. Dinosaurs	-	-	-	-	-	-	0.76	-	0.07	0.16	-	-	-	-	-	-	-	-	-
8. Dolphins	0.20	-	-	-	-	-	-	0.30	0.05	-	-	-	-	-	-	-	-	-	-
9. Fishes	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-
10. Four-limbs	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-
11. Hands	-	0.20	-	-	-	-	-	-	-	-	0.30	0.10	-	-	-	-	0.40	-	-
12. Humans	-	-	-	-	-	-	-	-	-	-	-	0.81	-	0.09	-	-	0.09	-	-
13. Octopus	-	-	-	-	-	0.08	-	0.08	-	0.16	-	0.08	0.33	0.16	-	-	0.08	-	-
14. Pliers	-	-	-	-	-	-	-	-	-	-	-	-	-	0.80	-	0.20	-	-	-
15. Snakes	0.16	0.16	-	-	-	-	-	-	0.16	-	-	-	-	-	0.33	-	-	0.16	-
16. Spectacles	0.08	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0.91	-	-	-
17. Spiders	-	-	-	-	-	0.07	0.07	0.07	-	0.35	-	0.07	-	-	-	-	0.35	-	-
18. Tables	-	-	-	-	-	-	-	-	0.40	-	-	-	-	-	-	-	-	0.60	-
19. Teddy-bears	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1

Fig. 8 shows the correct classification rate of the proposed algorithm on the PSB database. SVM classifiers with linear kernel function are experimentally used in this experiment as well. Fig. 8 illustrates the percentage of correct classification for the various dimensions of the sparse representation, i.e., k values. In this experiment, 907 objects are used to train the classifiers as well as obtaining the dictionary for sparse representation. The 907 remaining objects are also utilized to test the proposed algorithm. The results of Fig. 8 demonstrate that the maximum allowable dimension of sparse representation, i.e., $k=907$, results in the maximum percentage of correct classification.

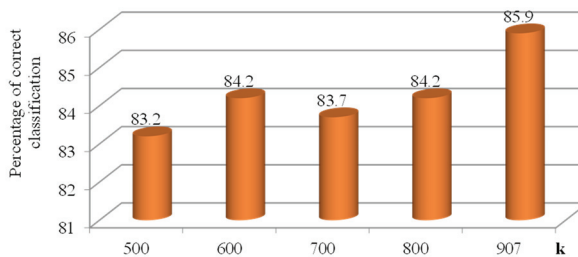


Fig. 8. Correct classification rates of the proposed algorithm on the PSB database for the various dimensions of the sparse representation, i.e., k values.

Table 5 shows the correct classification rate of the proposed algorithm on the PSB database with and without applying the sparse representation stage on intermediate features. The results of Table 5 show that the sparse representation of intermediate features increases the correct classification rate of the proposed algorithm up to 16.99%.

Table 5. Percentage of correct classification for the proposed algorithm on the PSB database with and without the sparse representation stage

	With the sparse representation	Without the sparse representation
Percentage correct classification	85.9	68.91

We also compared the results of applying the proposed algorithm on the PSB database with the methods of Bu et al. [18] and Hamid and Nakajima [38]. Bu et al. [18] leveraged the geodesics-aware bag-of-features (GA-BoF) as the initial features. Then DBNs were used for dimensionality reduction and 3D shape classification. They also reported the results of shape classification using GA-BoF features and some of other dimensionality reduction algorithms such as PCA, multidimensional scaling (MDS), linear discriminant analysis (LDA), and locally linear embedding (LLE) algorithms. Table 6 compares the results of the proposed algorithm on the PSB database with GA-BoF features with DBNs and some other dimensionality reduction algorithms as well as the method of Hamid and Nakajima [38]. The results of this table demonstrate the higher performance of the proposed algorithm.

Table 6. Comparison of the results of the proposed algorithm with a method of Bu et al. [18] using DBNs and other dimensionality reduction algorithms as well as Hamid and Nakajima [38] approach

Algorithm	Percentage of correct recognition
GA-BOF	71.4
PCA	76.5
MDS	77.1
LDA	72.5
LLE	73.7
DBN	85.1
Hamid and Nakajima [38]	70.05
Proposed method with 9 views	85.9
Proposed method with 20 views	89.7

5. Conclusions

This study reports a new approach for 3D shape recognition using 3D local features of model views. In comparison with the existing view-based approaches that mostly extract the silhouette of 3D models, we consider the distances of the points from the center of mass of a 3D model as the intensities of the pixels of extracted views. Then, we use Gabor-based filters to extract proper features for shape classification. To take into account the relationships between various views, we use a new approach for 3D local feature extraction by the construction of view cubes. Additionally, we describe intermediate features in the sparse domain to enhance the accuracy of classification.

Experimental results on both the PSB and MSB databases demonstrate the effectiveness and higher performance of the proposed method on the classification of 3D objects. Additionally, the comparisons of the results generated by the proposed method with those of other state-of-the-art approaches show that more reliable results could be obtained using the proposed method.

The experimental results also show that the sparse representation of intermediate features increases the accuracy of the proposed algorithm. Our experiments using SVM classifiers with a radial basis function (RBF) as a kernel demonstrate that the correct classification rate decreases with an RBF kernel function. This effect shows that the sparse representation of intermediate features makes them linearly separable.

According to the results obtained by the proposed approach, the increase in the number of coefficients for the sparse representation enhances the performance of the proposed approach. In the proposed approach, we experimentally used the maximum number of allowable coefficients in the training phase,

i.e., the number of intermediate features of shape models. This effect indicates the scalability for the proposed algorithm, because by increasing the number of classes and shape models it is possible to increase the number of coefficients for the sparse representation.

References

- [1] J. S. Nam, H. Gao, M. K. Kang, K. T. Kim, S. C. Son, C. U. Pom, and K. Heo, "Scenario-based 3D Objects Synthesizing System Design," *Journal of Information Processing Systems*, vol. 2, no. 1, pp. 18-22, 2006.
- [2] I. Atmosukarto, "3D shape analysis for quantification, classification, and retrieval," Ph.D. dissertation, University of Washington, Seattle, WA, 2010.
- [3] J. W. Tangelder and R. C. Veltkamp, "A survey of content based 3D shape retrieval methods," *Multimedia Tools and Applications*, vol. 39, article no. 441, 2008.
- [4] M. A. Savelonas, I. Pratikakis, and K. Sfikas, "An overview of partial 3D object retrieval methodologies," *Multimedia Tools and Applications*, vol. 74, pp. 11783-11808, 2015.
- [5] C. Li and A. B. Hamza, "Spatially aggregating spectral descriptors for nonrigid 3D shape retrieval: a comparative survey," *Multimedia Systems*, vol. 20, no. 3, pp. 253-281, 2014.
- [6] G. L. Lopez, A. P. P. Negron, A. D. A. Jimenez, J. R. Rodriguez, and R. I. Paredes, "Comparative analysis of shape descriptors for 3D objects," *Multimedia Tools and Applications*, vol. 76, pp. 6993-7040, 2017.
- [7] M. Elad, A. Tal, and S. Ar, "Content based retrieval of VRML objects: an iterative and interactive approach," in *Multimedia 2001*. Vienna: Springer, 2002, pp. 107-118.
- [8] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, "Shape distributions," *ACM Transactions on Graphics*, vol. 21, no. 4, pp. 807-832, 2002.
- [9] R. Ohbuchi, T. Minamitani, and T. Takei, "Shape-similarity search of 3D models by using enhanced shape functions," *International Journal of Computer Applications in Technology*, vol. 23, no. 2-4, pp. 70-85, 2005.
- [10] M. Mahmoudi and G. Sapiro, "Three-dimensional point cloud recognition via distributions of geometric distances," *Graphical Models*, vol. 71, no. 1, pp. 22-31, 2009.
- [11] D. Saupe and D. V. Vranic, "3D model retrieval with spherical harmonics and moments," in *Pattern Recognition*. Heidelberg: Springer, 2001, pp. 392-397.
- [12] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors," in *Proceedings of Eurographics Symposium on Geometry Processing*, Archen, Germany, 2003, pp. 156-164.
- [13] H. Laga, H. Takahashi, and M. Nakajima, "Spherical wavelet descriptors for content-based 3D model retrieval," in *Proceedings of IEEE International Conference on Shape Modeling and Applications*, Matsushima, Japan, 2006.
- [14] P. Shilane and T. Funkhouser, "Distinctive regions of 3D surfaces," *ACM Transactions on Graphics*, vol. 26, no. 2, article no. 7-es, 2007.
- [15] Y. Zhao, Y. Liu, Y. Wang, B. Wei, J. Yang, Y. Zhao, and Y. Wang, "Region-based saliency estimation for 3D shape analysis and understanding," *Neurocomputing*, vol. 197, pp. 1-13, 2016.
- [16] I. Atmosukarto, K. Wilamowska, C. Heike, and L. G. Shapiro, "3D object classification using salient point patterns with application to craniofacial research," *Pattern Recognition*, vol. 43, no. 4, pp. 1502-1517, 2010.
- [17] I. Sipiran and B. Bustos, "Harris 3D: a robust extension of the Harris operator for interest point detection on 3D meshes," *The Visual Computer*, vol. 27, no. 11, article no. 963, 2011.
- [18] S. Bu, Z. Liu, J. Han, J. Wu, and R. Ji, "Learning high-level feature by deep belief networks for 3-D model retrieval and recognition," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2154-2167, 2014.

- [19] A. Flint, A. Dick, and A. Van den Hengel, "Local 3D structure recognition in range images," *IET Computer Vision*, vol. 2, no. 4, pp. 208-217, 2008.
- [20] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson, "Skeleton based shape matching and retrieval," in *Proceedings of 2003 Shape Modeling International*, Seoul, South Korea, 2003, pp. 130-139.
- [21] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii, "Topology matching for fully automatic similarity estimation of 3D shapes," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, CA, 2001, pp. 203-212.
- [22] B. Leng, C. Du, S. Guo, X. Zhang, and Z. Xiong, "A powerful 3D model classification mechanism based on fusing multi-graph," *Neurocomputing*, vol. 168, pp. 761-769, 2015.
- [23] A. Kacem, W. Mohamed, and A. B. Hamza, "Spectral geometric descriptor for deformable 3D shape matching and retrieval," in *Image Analysis and Recognition*. Heidelberg: Springer, 2013, pp. 181-188.
- [24] W. Mohamed and A. B. Hamza, "Deformable 3D shape retrieval using a spectral geometric descriptor," *Applied Intelligence*, vol. 45, no. 2, pp. 213-229, 2016.
- [25] P. Daras and A. Axenopoulos, "A compact multi-view descriptor for 3D object retrieval," in *Proceedings of 2009 Seventh International Workshop on Content-Based Multimedia Indexing*, Chania, Greece, 2009, pp. 115-119.
- [26] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 2015, pp. 945-953.
- [27] K. Ding and Y. Liu, "A probabilistic 3D model retrieval system using sphere image," in *Computer Vision – ACCV 2012*. Heidelberg: Springer, 2012, pp. 536-547.
- [28] S. Zhao, H. Yao, Y. Zhang, Y. Wang, and S. Liu, "View-based 3D object retrieval via multi-modal graph learning," *Signal Processing*, vol. 112, pp. 110-118, 2015.
- [29] D. V. Vranic and D. Saupe, "3D model retrieval," in *Proceedings of the Spring Conference on Computer Graphics and its Applications (SCCG2000)*, Budmerice, Slovakia, 2000, pp. 89-93.
- [30] D. V. Vranic, D. Saupe, and J. Richter, "Tools for 3D-object retrieval: Karhunen-Loeve transform and spherical harmonics," in *Proceedings of 2001 IEEE 4th Workshop on Multimedia Signal Processing (Cat. No. 01TH8564)*, Cannes, France, 2001, pp. 293-298.
- [31] P. Papadakis, I. Pratikakis, S. Perantonis, and T. Theoharis, "Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation," *Pattern Recognition*, vol. 40, no. 9, pp. 2437-2452, 2007.
- [32] J. Mutch and D. G. Lowe, "Object class recognition and localization using sparse features with limited receptive fields," *International Journal of Computer Vision*, vol. 80, no. 1, pp. 45-57, 2008.
- [33] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311-4322, 2006.
- [34] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," *Advances in Neural Information Processing Systems*, vol. 20, pp. 801-808, 2007.
- [35] P. H. Chen, C. J. Lin, and B. Scholkopf, "A tutorial on v-support vector machines," *Applied Stochastic Models in Business and Industry*, vol. 21, no. 2, pp. 111-136, 2005.
- [36] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, and S. Dickinson, "Retrieving articulated 3-D models using medial surfaces," *Machine Vision and Applications*, vol. 19, no. 4, pp. 261-275, 2008.
- [37] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, "The princeton shape benchmark," in *Proceedings Shape Modeling Applications*, Genova, Italy, 2004, pp. 167-178.
- [38] L. Hamid and M. Nakajima, "Supervised learning of salient 2D views of 3D models," *The Journal of the Society for Art and Science*, vol. 7, no. 4, pp. 124-131, 2008.

**Hussein Kanaan** <https://orcid.org/0000-0001-5688-6953>

He received the B.S. degree in biomedical engineering in 2008. In 2012, he received the M.S. degree in communication engineering from Shahed University, Tehran, Iran. He has been working towards the PhD degree in electronic engineering at Machine Vision and Image Processing laboratory of Shahed University, Tehran, Iran, since 2012. His current research interests include 3D object recognition, classification and machine vision.

**Alireza Behrad** <https://orcid.org/0000-0002-1990-6668>

He received the B.S. degree in electronic engineering from Electrical Engineering Faculty, Tabriz University, Tabriz, Iran, in 1995. In 1998, he received M.S. degree in digital electronics from Electrical Engineering Faculty, Sharif University of Technology, Tehran, Iran. He received Ph.D. degree in electronic engineering from Electrical Engineering Faculty, Amirkabir University of Technology, Tehran, Iran, in 2004. Currently, he is an associate professor of Electrical Engineering Department, Shahed University, Tehran, Iran. His research fields are image and video processing and machine vision.