

## 3D model retrieval using meshSIFT descriptor and fuzzy C-means clustering

Najlaa Abd Hamza<sup>1</sup>, Shatha Habeeb Jafer<sup>2</sup> and Raghad Mohammed Hadi<sup>3</sup>

<sup>1</sup>College of Nursing, University of Baghdad, Iraq

<sup>2</sup>Computer Science Department, University of Technology, Iraq

<sup>3</sup>College of Administration and Economics, Mustansiriyah University, Iraq

---

### Article Info

#### Article history:

Received Sep 28, 2019

Revised Feb 9, 2020

Accepted Mar 27, 2020

---

#### Keywords:

3D model retrieval

Feature descriptor

Fuzzy c-means clustering

---

### ABSTRACT

A well-prepared abstract enables the reader to identify the basic content of a document quickly and accurately, to determine its relevance to their interests, and thus to decide whether to read the document in its entirety. The Abstract should be informative and completely self-explanatory, provide a clear statement of the problem, the proposed approach or solution, and point out major findings and conclusions. The Abstract should be 100 to 200 words in length. The abstract should be written in the past tense. Standard nomenclature should be used and abbreviations should be avoided. No literature should be cited. The keyword list provides the opportunity to add keywords, used by the indexing and abstracting services, in addition to those already present in the title. Judicious use of keywords may increase the ease with which interested parties can locate our article.

*Copyright © 2020 Institute of Advanced Engineering and Science.  
All rights reserved.*

---

### Corresponding Author:

Najlaa Abd Hamza,

College of Nursing,

University of Baghdad,

Bab Al-Muadham, Baghdad, Iraq.

Email: najlaaabd@conursing.uobaghdad.edu.iq

---

## 1. INTRODUCTION

Retrieving desired models from databases in a short time has become a large problem for research because of the increase in the number of 3D models in company or internet databases [1]. The number of requests for high-quality three-dimensional models is increasing each day. To create an innovative, straightforward model, however, is a very time-consuming procedure. In cases where three-dimensional models can be retrieved professionally from a database, the work efficiency in associated fields is enhanced by reprocessing the model. There are two methods for the retrieval of three-dimensional models from a database: a technique based on key words, and a technique based on content [2, 3].

For content-based 3D model retrieval, features of a model should be found to describe it, called shape descriptors. Some of these descriptors are robust, but applying them in real time is time-consuming; others are less time-consuming but also less accurate [1]. In this paper, we propose using a meshSIFT descriptor [4] which is robust against missing data, large rotations and translations and can thus find a balance between speed and robustness.

In a content-based retrieval engine, an input model, termed query, is selected, and models which are similar to the query are retrieved from a model database. Applying the retrieval algorithm directly to 3D models in a database is time-consuming [2, 5]. In this paper we propose a pre-processing step as an attempt to reduce the time it takes to search for 3D models in a dataset, achieved by reducing the size of that dataset through undertaking a comparison between different 3D model features using a KLT algorithm. These main features are extracted by means of a meshSIFT descriptor, resulting in one 3D model from any similar group representing an

input to a Fuzzy C-Means clustering approach of partitioning the vertices of both dataset and query models. Final steps are then taken to calculate the similarity between the two 3D models (dataset and query).

Mohammadhassan Rezaei and Erkan Gunpinar [1] have proposed a method for 3D mesh retrieval which first divides a model into clusters by utilizing a geodesic distance metric, using a skeleton-based, K-means clustering method, and then computes its descriptor using the area of these clusters. Each cluster is signified by an area-based descriptor which is invariant to orientation and scale.

Xiaohong Liu et al. [2] have implemented view-based, three-dimensional methods of model retrieval. In their methods, bag-of-words (BOW), regularization-based SIFT features are obtained from three projected views of a three-dimensional model, before a distributed K-means clustering approach (based on Hadoop platform) is employed for the calculation of feature vectors and for the clustering of three-dimensional models. The model correspondence is computed using the distance between the query model and every cluster's center, and the cluster closest to the query model is returned as a model of retrieval.

Kuan-Sheng Zou et al. [6] have introduced a combined shape distribution (CSD) descriptor for the retrieval of three-dimensional models based on group integration and principal plane analysis.

Jing Zhang et al. [7] have proposed a new sketch-based 3D model retrieval procedure which employs skeleton characteristics as the features to define the object shape.

## 2. FUZZY C-MEANS CLUSTERING

Clustering can be defined as the technique of grouping a collection of data objects as clusters or classes in such a way that objects within a cluster are similar to one another, but dissimilar to the other clusters' objects [8, 9].

Fuzzy C-Means (FCM) is one of the highest public fuzzy combination methods. It was proposed by Dunn [10] in 1973 and then improved by Bezdek [11] in 1981. This algorithm is an iterative clustering technique which produces an optimal partition by reducing the objective function  $J_m = (X, U, C)$  [12].

$$J_m = \sum_{i=0}^n \sum_{j=0}^c U_{ij}^m \|x_i - c_j\|^2$$

Where m denotes parameter of fuzziness which is used to modify the weighting effect of membership values,  $U=[u_{ij}]$  is a matrix of fuzzy partition that denotes the belongingness degree (membership) for  $x_i$  into center  $c_i$ , this matrix must satisfy the following limitations, for each j,  $1 \leq j \leq n: u_{ij} = 1$ , and  $\|x_i - c_j\|^2$  is the distance between sample  $x_i$  and center  $c_j$  [10].

The Fuzzy C-Means algorithm has been summarized in the stages below [8, 14]:

- a) Initialize matrix  $U=[u_{ij}]$  with initial value  $u^{(0)}$ .
- b) For n-step: compute the matrix of the cluster center  $Q^{(n)} = [c_i]$  with  $u^{(n)}$ .
- c) Update  $u^{(n)}$ ,  $u^{(n+1)}$ .
- d) If  $\|u^{(n+1)} - u^{(n)}\| < \epsilon$  then finish, else return to step2.

where  $\epsilon$  represents the termination state threshold.

### 2.1. MeshSIFT

The meshSIFT algorithm consists of three main stages [4]: detection of key-points, assignment of orientations and generation of the local feature descriptor. The following subsection gives a detailed account of each of these stages.

#### 2.1.1. Key-point detection

This stage detects significant points on the mesh. As in SIFT [15, 16], a method of scale space is followed. In this scale space significant points are identified. Initially, a scale space is built holding smoothed input mesh versions, which are achieved with the approximation of a Gaussian filter for meshes. This approximation involves the following mesh convolutions with a binomial filter [4]:

$$X_i = \begin{cases} X & i = 0 \\ \widehat{M}_{J_t} * X & \text{else} \end{cases}$$

The scale space is constructed using the following formula [15]:

$$\widehat{D_oM}_t = X_i - X_{i-1} = (\widehat{M}_{J_t} - \widehat{M}_{J_{t-1}}) * X$$

where  $X$  is the original mesh,  $\widehat{M}_{Jt}$  and  $\widehat{DoM}_i$  are respectively the approximations of Gauss and difference of Gauss. The Gaussian filter's standard deviation  $\widehat{M}_{Jt}$ , which has been approximated for finding a smoothed mesh  $X_i$ , has been given by  $Jt$ .

In order to discover significant points in the scale space, the mean curvature [15]

$$M(x, y) = \frac{(1+c_y^2)c_{xx}-2c_xc_yc_{xy}+(1+c_x^2)c_{yy}}{2(1+c_x^2+c_y^2)}$$

is subsequently calculated for every one of the vertices and at every one of the scales in the scale space, where  $M_i$  is a set holding the mean curvature for every one of the vertices at a scale  $i$ . Variances between consequent scales are calculated in the following way [4]:

$$DM_i = M_{i+1} - M_i$$

To identify the extrema of the scale space, the  $DM$  value at every one of the vertices is matched to the values of its neighbours, on the identical scale and on the lower and upper scales. A vertex is chosen only in that case where its value for  $DM$  is smaller or larger compared to all its neighbours. Lastly, the precise scale  $Jt$  is allocated to every scale-space extremum, directing to significant points (key-points) with an allocated scale [15].

### 2.1.2. Assignment of orientations

In this stage, every one of the key-points is allocated a recognized orientation, by stating each parameter in scale function  $Jt$ . The value of  $Jt$  around every one of the key-points is calculated. Initially, for every one of the vertices inside this area, the normal vector is calculated and the geodesic distance to the corresponding key-point is determined. Subsequently, all computed normal vectors are projected onto the tangent plane to the mesh  $X_i$  holding the key-point [4].

### 2.1.3. Local descriptor

The local descriptor now offers for every one of the key-points (with allocated scale and recognized orientation) a vector of the feature containing a sequence of concatenated histograms. Every one of those histograms has been intended over a minor circular area. In each region two histograms are computed where each one of them has 8 bins: one holding the index of the shape,

$$I(x, y) = \frac{2}{\pi} \tan^{-1} \left[ \frac{s_1(x, y) + s_2(x, y)}{s_1(x, y) - s_2(x, y)} \right]$$

(where  $s_1$  and  $s_2$  are the maximum and minimum curvatures respectively), and the other holding the slant angles (that is, the angle of the projected normal and the recognized orientations between each one). Initially, every item for both histograms is Gauss-weighted with the geodesic distance to the key-point ( $J = 4.5 \cdot Jt$ ) and Gaussian weighted with the geodesic distance to the region's centre ( $J = 4.5 \cdot Jt$ ). Every histogram is then normalized and cropped to  $\frac{1}{\sqrt{m}}$  (where  $m$  is the number of bins), decreasing the effect of big histogram values. In a last stage all histograms are concatenated [4]:

$$\begin{aligned} \widehat{VF}_i &= (\widehat{h} \widehat{S}_s) (\widehat{h} \widehat{S}_{slant \text{ angle}}), i = 1, \dots, 9 \\ VF &= (\widehat{VF}_1, \widehat{VF}_2, \dots, \widehat{VF}_9) \dots \end{aligned}$$

where  $VF$  is the vector of final feature [4].

## 3. 3D Mesh Model Representation

A 3D mesh  $m$  signifies the 3D surface, employing groups of mesh components – vertices  $E$ , edges  $G$ , and polygons (facets)  $T$  – where  $m = (E, G, T)$ . The mesh vertices signify 3D locations,  $E \subset R^3$ . Every edge,  $g_i \in G$ , is separated through two isolated vertices,  $G \subset \{g_i = \{e_j, e_k\} \mid e_j, e_k \in E, j \neq k\}$ . Where every facet,  $t_i \in T$ , is unique at the smallest three edges with the end area, each pair of edges share a vertex [17].

In the state of a triangular mesh (which is the most used kind of mesh because of its relative straight-forwardness), the facet  $t_i$  is exactly categorized by three edges.

$$\begin{aligned} T \subset \{t_i = \{d_m, d_b, d_p\} \mid d_m, d_b, d_p \in D, m \neq b, \\ m \neq p, \forall d_1, d_2 \in g_i \exists t (t \in d_1 \wedge t \in d_2)\}. \end{aligned}$$

Every facet is distinct through three different vertices,  $T \subset \{t_i = \{e_m, e_b, e_p\} | e_m, e_b, e_p \in E, m \neq b, m \neq p, b \neq p\}$  [18]. The mesh is part of a model, and defines the object vertices. Figure 1 shows the key components of each mesh [19]. There are several kinds of mesh representations, which vary depending on how the mesh data are structured and stored: data as polygon mesh, vertex-polygon mesh, simple adjacency mesh, partial adjacency mesh, triangular mesh and winged-edge mesh [20].

The triangular mesh is the most generally-used mesh representation because of its simplicity. It usually stores the triangles and the vertices, with guides to all neighbouring triangles and vertices, in isolated tables. The information from the edge is understood and can be determined at any triangle by its edges or vertex (distinct from the adjacent vertices) [19].

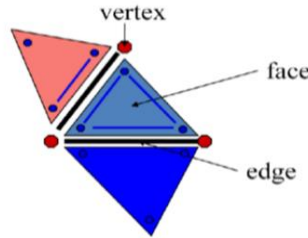


Figure 1. Example of vertices, faces, and edges

#### 4. Kanade Lucas Tomasi (Klt) Algorithm

A KLT algorithm, which provides the greatest possible breadth, is used for feature tracking. KLT procedure has been presented by Lucas and Kanade, who suggest a process for registering two images for stereo correspondence. Their work has been extended by Tomasi and Kanade [21]. Firstly, the displacement of the tracked feature from one mesh to another is calculated. From this displacement calculation it is easy to compute the movement of the feature. KLT tracking procedure tracks the feature in two steps: it determines the traceable feature in the initial mesh before then tracking the identified features in the next mesh by utilizing the calculated movement [22].

Simple notification has been used to display the movement  $d$  of a feature from one mesh to the next. For ease the second mesh  $B(x)$  is redefined as  $B(x) = I(x)$  and the first mesh as  $A(x + d) = I(x + d)$ . [22, 23]. The KLT tracker is the most efficient method of reducing

$$\epsilon(d) = \frac{1}{2} \iint_W [J(x + d) - I(x)]^2 \omega(x) dx$$

Compute the sum of squared error between mesh windows, then brute force search over  $d$  with normal cross correlation tracking. Here,  $I(x)$  is the value of the feature in the first mesh,  $J(x+d)$  is the value of the feature in the second mesh,  $x$  is the location of the feature in  $I$ ,  $w(x)$  is a weight which is usually 1, and  $d$  is the movement of the feature.

The simple approach to tracking (also known as cross-correlation search) involves scanning a window around a search box (trying all values of  $d$ ), and taking the one with the lowest cost,  $\epsilon$ , above. This is surprisingly effective, but slow. If the search box has size  $100 \times 100$  pixels, and each evaluation of  $\epsilon$  involves a  $15 \times 15$  pixel sum, then it takes at least  $15 \times 15 \times 100 \times 100 = 2,250,000$  floating point operations per tracked feature per frame. The KLT takes a different approach. The idea is to reduce the remainder by differentiating  $\epsilon(d)$  with respect to  $d$  and taking a truncated Taylor expansion of  $J(x + d)$ ,

$$J(x + d) \approx J(x) + \frac{\partial J(x)^T}{\partial x} d$$

where  $\frac{\partial J(x)}{\partial x}$  is the image derivative in the  $x$  and  $y$  directions. Substituting this into the equation for  $\epsilon$  and setting derivatives to zero yields

$$\underbrace{\left[ \iint_W \frac{\partial J(x)}{\partial x} \frac{\partial J(x)^T}{\partial x} \omega(x) dx \right]}_{Z \in R^{2 \times 2}} d + \underbrace{\left[ \iint_W [J(x) - I(x)] \frac{\partial J(x)^T}{\partial x} \omega(x) dx \right]}_{Z \in R^2} = Zd + e = 0$$

Tracking thus reduces to repeatedly solving the simple  $2 \times 2$  linear system  $Zd = e$ , where  $Z$  and  $e$  are readily computed from the images and gradients. The actual algorithm implemented is slightly more sophisticated, and uses multi-resolution image pyramids to enlarge the basin of attraction. By searching from the top of the pyramid down, it is possible to successfully track large motions [24].

## 5. System Overview

The proposed system uses three algorithms, as shown in Figure 2: Fuzzy C-Means clustering for 3D mesh models to partition the mesh into vertex clusters, a meshSIFT algorithm for deriving the key features of the 3D models, and a KLT algorithm for computing the similarity between the features in a pre-processing step. In the following sections, we give a brief overview of the process.

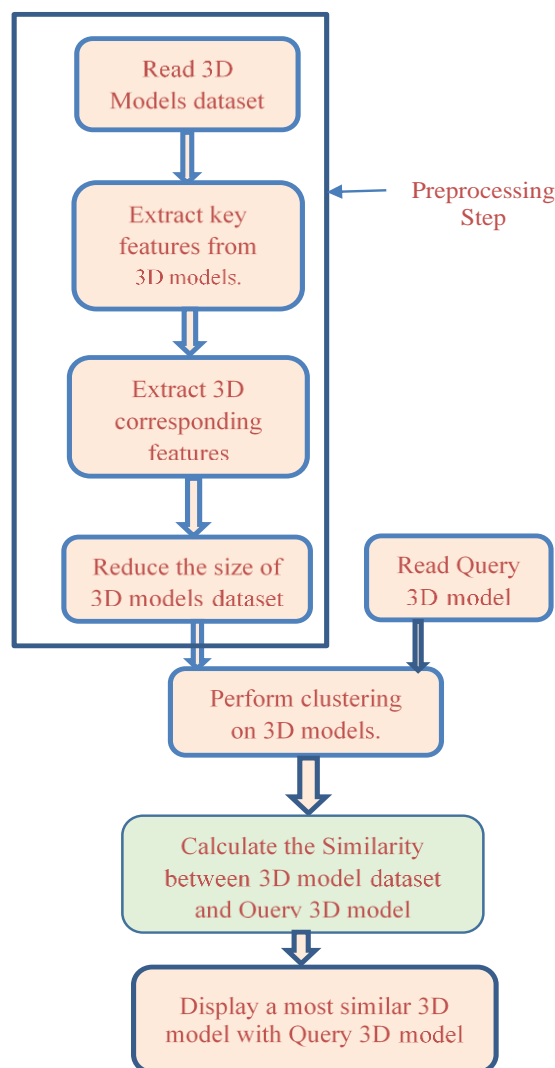


Figure 2. Block diagram of the proposed system

### 5.1. Pre-processing step

In this step, we attempt to reduce the size of the 3D model dataset by removing its redundancy. A detailed description of this step is provided in the following subsections.

#### 5.1.1. Extracting the important features

In this stage, 3D model dataset is read and the key features are extracted for all 3D model datasets using the meshSIFT algorithm.



**5.4. Model retrieval**

In this step, we calculate the similarity between the query 3D model and other 3D models. Similarity rate between two models are calculated based on the sum of squared differences (SSD) [6]:

$$Z = \text{sum}(m_1 - m_2)^2$$

The individual steps involved in model retrieval are as follows:

- a) Sort the clusters for each model from the smallest to largest.
- b) Compute the similarity between the largest cluster in the query 3D model and the largest cluster in the 3D model dataset.
- c) Sort the value of similarity from large to small.
- d) If the value is smaller than the threshold, reject the 3D model which contains this cluster.
- e) Calculate the similarity between the next largest cluster from the query 3D model and the other next largest cluster in the 3D model dataset, and go to step 3.































By applying these steps, we decrease the both the size of the 3D model data set and the time taken to process it.

**6. RESULTS AND DISCUSSION**

A set of testing models, comprising 80 3D mesh models from the Princeton shape benchmark [21], is used to certify the proposed algorithm. Retrieving is the procedure of discovering similar models to a query model. Table 1 shows the results of the proposed algorithm.






For a fish model, the most similar model is the fish model whose rate of similarity is 99%. For a face of women model, the most similar model is a women face model too. However, it has been observed that for a man face model, human model and a house model, the most similar models are the same model.

Table 1 Displays the results of model retrieval for different queries

Query Model	Recovered models respectively from greatest to smallest rate of similarity depending on Similarity rate				
					
	99%	94%	91%	91%	89%
					
	99%	97%	90%	88%	86%
					
	98.6%	98%	97.8%	97.2%	93%
					
	98.2%	95%	89%	88%	86%
					
	98%	95%	91%	90%	90%

According to these tests, it can be said that the proposed algorithm provides positive results in most cases. Computational time: time required for performing the proposed algorithm is shown in Table 2.

Table 2. Computational time

Model	Time for Model Retrieval (in minutes)
	15.2
	14.8
	24.1
	12.8
	24.5

## 7. CONCLUSION

In the present paper, we propose an innovative technique for retrieving 3D models denoted by meshes. Having first extracted the key features of the models, the models are then separated into clusters by applying a Fuzzy C-means method depending on key features. Performance of the suggested retrieval method is shown via five experiments where rigid and non-rigid models are applied. According to the results, the suggested technique is successful in terms of its retrieving performance. Furthermore, it has been shown that the technique is invariant to different model poses, and therefore can be used for non-rigid models.

## REFERENCES

- [1] E. G. Mohammadhassan Rezaei, "A K-means clustering based shape retrieval technique for 3D mesh models," *Selcuk Univ. J. Eng. Sci. Tech.*, vol. 6, no. 1, pp. 114-128, 2018.
- [2] X. Liu, D. Wu, Y. Chen, P. Li, Z. Qu, "Distributed cluster-based 3D model retrieval with map-reduce," *Journal of Computer and Communications*, vol. 6, no. 5, pp. 83-93, 2018.
- [3] D. Chen, X. Tian, Y. Shen and M. Ouhyoung, "On Visual Similarity Based 3D Model Retrieval", *The Eurographics journal*, vol.22, No.2,2003
- [4] C. Maes, T. Fabry, J. Keustermans, D. Smeets, P. Suetens and D. Vandermeulen, "Feature detection on 3D face surfaces for pose normalisation and recognition," in *Fourth IEEE International Conference on Theory Applications and Systems (BTAS)*, 2010.
- [5] B. Li, Y. Lu and H. Johan, "Sketch-Based 3D Model Retrieval by Viewpoint Entropy-Based Adaptive View Clustering", *Eurographics Workshop on 3D Object Retrieval*, 2013.
- [6] K. S. Zou, W. H. Ip, C. H. Wu, Q. Chen, K. L. Yung and C. Y. Chan, "A novel 3D model retrieval approach using combined," *Multimedia Tools and Applications*, vol. 69, no. 3, p. 799-818, 2014.
- [7] J. Zhang, B. S. Kang, B. Jiang, D. Zhang "A novel sketch-based 3D model retrieval approach based on skeleton," *International Journal of Informatics and Communication Technology (IJ-ICT)*, vol. 8, no. 1, pp. 1-12, 2019.
- [8] O. M. E. Zein, L. M. E. Bakrawy, and N. I. Ghali, "A robust 3D mesh watermarking algorithm utilizing fuzzy C-Means clustering," *Future Computing and Informatics*, vol. 2, pp. 148-156, 2017.
- [9] J. Iran, N. Pise and M. Phatak, "Clustering Techniques and the Similarity used in Clustering: A survey", *International Journal of Computer Applications*, Vol.134, No.7, January 2016.
- [10] J. C. Dun, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybernet*, vol. 3, no. 3, pp. 32-57, 1973.
- [11] J. C. Bezdek, "Pattern recognition with fuzzy objective function algorithms.," USA [5]: Kluwer Academic Publishers Norwell, 1981.



- [12] K. V. Rajkumar, A. Yesubabu, K. Subrahmanyam, "Fuzzy clustering and Fuzzy C-Means partition cluster analysis and validation studies on a subset of CiteScore dataset," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 4, pp. 2760-2770, 2019.
- [13] R. M. Hadi, "Text document clustering model using singular value decomposition and modified fuzzy C Mean," University of Technology, Department of Computer Science, Iraq, 2018.
- [14] N. V. Arup Guha, "Prediction of bankruptcy using big data analytic based on fuzzy c-means algorithm," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 8, no. 2, pp. 168-174, 2019.
- [15] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.
- [16] A. Jafari, M. abadi, M. Khojastehpour and B. Emadi, "A comparative evaluation of combined feature detectors and descriptors in different color spaces for stereo image matching of tree" , *Elsevier journals, Scientia Horticulturae*, Vol. 228, 26 Jan.2018,Pages:187-195.
- [17] N.A.Hamza, S.H.Jafeer, A.E.Ali," Encrypt 3D Model Using Transposition, Substitution, Folding, and Shifting (TSFS), *2019 2nd Scientific Conference of Computer Sciences (SCCS), IEEE*, PP.126 - 131
- [18] N. Pears, Y. Liu, and P. Bunting, "3D imaging, analysis and applications," London: Springer-Verlag London, 2012.
- [19] S. H. Shaker, N. A. Hamza "Surface shape descriptors on 3D faces," *Journal of Science*, vol. 58, no. 3C, pp. 1740-1750, 2017.
- [20] M. Daoudi, A. Srivastava, and R. Veltkamp, "3D face modeling, analysis and recognition," The John Wiley & Sons Ltd, 2013.
- [21] A. Xu and G. Namit, "SURF: Speeded - Up Robust Features," in *European Conference on Computer Vision*, 2008.
- [22] J. Pedersen, " SURF: Feature detection & description", *technical report* , 2011.
- [23] R. Gherardi, "Advances in 3D reconstruction," Ph.D. Thesis University of Verona, Department of Informatic, 2010.
- [24] R. Boda and M. J. P. Priyadarsini, "Face detection and tracking using KLT and Viola Jones," *ARPN Journal of Engineering and Applied Sciences*, vol. 11, no. 23, pp. 13472–13476, 2016.
- [25] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, "The Princeton Shape Benchmark," in *Shape Modeling Applications*, Genova, Italy, 7-9 June 2004.

## BIOGRAPHIES OF AUTHORS



**Najlaa Abd Hamza** presently working as Assistant Professor of the College of Nursing, University of Baghdad, Iraq. She received Ph.D from Department of Computer Science, University of Technology, Iraq. Her Research interest includes Multimedia Processing, 3D Model processing, Feature Extraction, Security and Data Mining



**Shatha Habeeb Jafer** presently working as Assistant Professor & Head of the Information Systems Branch, Department of Computer Science, University of Technology, Iraq. She received Ph.D from Department of Computer Science, University of Technology, Iraq. Her Research interest includes internet of things, web of things, network management web application, Data Mining and security.



**Raghad Mohammed Hadi** presently working as Assistant Professor of the College of Administration and Economics, Mustansiriyah University, Iraq. She received Ph.D, from Department of Computer Science, University of Technology, Iraq. Her Research interest includes security, Network systems, AI, and Data Mining.