International Journal of Engineering & Technology, 7 (2.13) (2018) 103-108

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper



D2 shape distribution and artificial neural networks for 3D objects recognition

Safae Elhoufi [1] *, Aicha Majda [1], Khalid Abbad [1]

[1] *Intelligent System And Application Laboratory (SIA) FST Fez, Morocco*
**Corresponding author E-mail: elhoufi.safae@gmail.com*

Abstract

In this paper, we propose a 3D object recognition approach, based on the shape distribution D2 and artificial neural networks. The challenge is to discriminate between similar and dissimilar shapes by finding a shape signature that can be constructed and classified quickly. We propose a connectionist system to recognize 3D objects in VRML (Virtual Reality Modeling Language) format. The key idea is to represent the signature of an object as a shape distribution sampled from a shape function measuring global geometric properties of an object. The proposed strategy is the following: from a polygon object to be recognized, a triangulation is performed. Then, distances are calculated between two random points of the triangulated surface of the 3D object. The frequency of these distances will be represented by a normalized histogram. The values of these histograms feed a multi-layer neural network with back- propagation training. We demonstrate the potential of this approach in a set of experiments, which proved our system could achieve above 91.7% recognition rate. In addition, to evaluate the efficiency of our method, we compare our classifier with Support vector machine and k- nearest neighbours. The simulation results highlight the performance of the proposed approach.

Keywords: 3D Object; VRML; 3D/3D Indexing; Shape Distribution; Artificial Neural Network; Classification; 3D Retrieval; Recognition.

1. Introduction

The world is constantly changing, man evolves, and his lifestyle undergoes permanent changes. Nowadays, we notice a rise of computerization, where the traditional comes up against the need to evolve. This also applies to the world of computing that gives new impetus to 3D objects. The ease of creating these data, whether by personal digital cameras, camcorders, by 2D or 3D scanners, or by design software, has led to a significant increase of 3D objects bases and difficulties navigating through the 3D model databases. In this paper, we focus on the problematic related to the recognition and classification of 3D objects. This choice emanates from our conviction of the importance of this area that requires the implementation of new tools to handle these data.

During the last decade, the recognition of 3D objects has been widely studied and used in many areas [1]: including medical imaging, virtual reality, face detection, and quality control with face and fingerprint recognition. More 3D recognition applications can be found in [2]. The recognition process has three major steps: The first step is the computation of descriptors, which can be global [3], local [4], structural [5], transform-based [6], or view-based [7]. The second, is the similarity measure, which can be distance-based [8], probability-based [9], or graph-based [10], and finally, the decision-making process to classify 3D objects according to the similarity measure used. In a large base of 3D objects, searching for the closest model to the request object becomes more difficult, requiring a structuration of indexes to reduce the search space. To overcome this problem, the search for similar 3D objects is brought to a classification problem, where we calculate the closet class of 3D objects to the query object according to specific criteria. The search result will be a group or class of three-dimensional models with similar characteristics to the requested model.

In this paper, we propose a new method of 3D object recognition based on 3D shape signatures and artificial neural network Classifier. We analyse a method that calculates the 3D shape signatures for 3D polygonal models. The purpose is to represent the object's signature as a shape distribution, by a shape function that allows measuring global properties. The first challenge of this approach is to choose the discriminating shape functions. The second one is to calculate the efficient method for sampling them, and the last one is to classify these distributions. To achieve our goal, we use the shape distribution (D2) as a shape descriptor and artificial networks of neurons for classification. ANN functions are efficient for non-linear prediction problems. The goal behind using an ANN in this paper is to generate a better output function than the classical approximation methods. In the classical approximation methods, a set of data is adjusted to an n-degree polynomial function [11]. ANN consists of three kinds of layers: the input, hidden, and output layers. A weighted sum of the neuron inputs specifies the activation (i.e. sigmoid) function argument. The activation function is assumed nonlinear. This paper presents our results; we found the implemented method is not only simple and fast but also offers a classification with a promising recognition rate.

Methodically, the paper is shackled into sections. First, we will review the previous work on 3D shape similarity search and retrieval, and we will focus on those methods that target polygon models. In section 3, we will detail descriptions of issues and propose solutions for implementing our approach. Experiments results aimed at evaluating the efficiency and the robustness of our system will be the focus of Section 4. We will conclude the paper in Section 5.

Copyright © 2018 Safae Elhoufi et. al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

2. Related work

The framework of our study is information research by content, which involves indexation of three-dimensional objects. It aims to codify the content of a data to ease access to large databases. This operation involves analyzing the different properties of the 3D model to extract the main characteristics. This bibliographic review identifies two indexing method classes: The first class considers we can search for 3D objects regarding another object (3D/3D indexing): It captures the form directly from the three-dimensional model. Another set of methods characterize the 3D objects based on projection of the object in a two-dimensional space before its description (2D/3D indexing).
Several Researchers, such as Poggio and Riesenhuber, found that, in a system of human vision and recognition, 3D object is represented by a set of view. That is the reason many algorithms use this approach, which is based on information coming from 2D image of a 3D object. To extract these images, simply make projections of the object on a map from different vision angles. We split this approach into three groups: methods based on the silhouette, depth images, and those using curvatures map see Figure 1.
Approach based on 2D projection

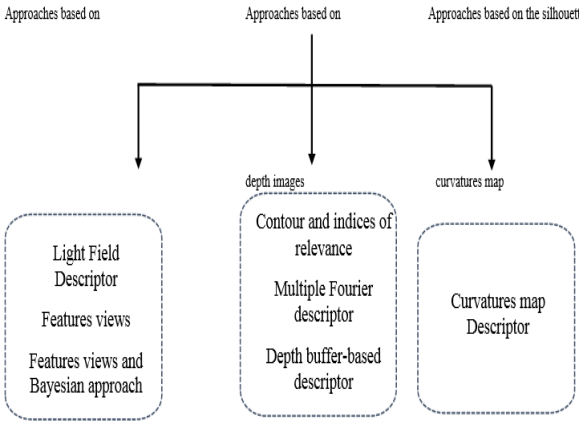


Fig. 1: Classification of 2D / 3D Approach into Three Subgroups.

Several articles introduce global descriptors. For example, [12, 22] introduce shape distribution as the probability distribution function of distance between two randomly selected points that reside on the 3D object surface. Since shape distribution is one of the simplest descriptors, it is appropriate to be used as a pre-classifier before real classification is executed. However, shape distribution is not the only approach that uses the probability distribution function or histogram as 3D shape representation. Ankerst et al. [23] suggests the 3D shape histogram showing the number of 3D points residing in each cell is divided by the radius and angle bin.
A study by Asari et al. [24] was used to compare four common 3D shape descriptors: shape distribution [12], local spin image [25], global spin image [26], and shape histogram [27]. They concluded that local spin image (using local similarity measurement) and shape distribution show excellent performance compared to other 3D shape descriptors. Local spin image outperformed shape distribution when only one instance per class is used in inter-class evaluation, and this is because of the local shape properties that are slightly similar in several depth images of the same instance. Shape distribution provides excellent performance compared to the local spin image, when several instances per class are used in intra-class evaluation. This is due to the local spin image suffering from invariant local shape properties within different instances per class, while global shape properties extracted by shape distribution is slightly retained. Since we are working with different objects, we have used the global descriptor, the distribution D2 shape [24]. The motivation behind our work is to develop a fast, simple, and robust method for recognition 3D polygonal models. We will describe the D2 shape feature and the classifier used in our work in the next section.

3. The proposed approach

Our focus in this paper is to represent the object's shape signature as a probability distribution. This representation produces informative global geometric properties for recognition and classification. The goal is to obtain a parameterized function from our 3D model and compare it easily with others, as shown in Figure 2. The steps to be followed in this approach are: 1) select discriminating shape functions, 2) construct the shape functions for each 3D object, and classify all distributions using artificial neural networks.

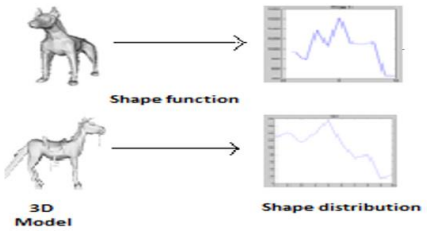


Fig. 2: Shape distributions facilitate shape matching because they represent 3D models as functions with a common.

Because we work directly on the original polygons of a 3D object, this approach can be used for similarity problems. Figure 3 presents the global diagram of our system.

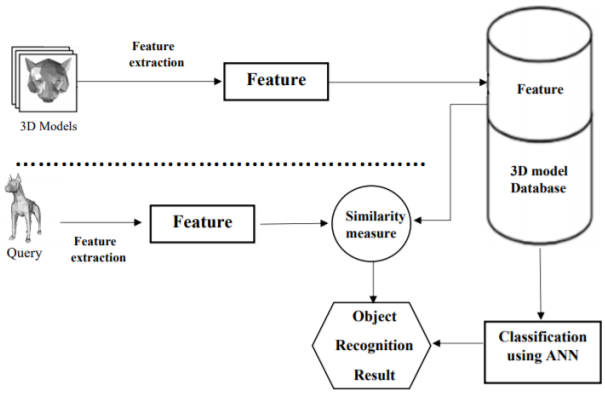


Fig. 3: Overview of the shape similarity search system.

The chosen strategy is the following: from a polygon object to be identified, triangulation is performed. Then, the distances are calculated between two random points of the triangulated surface of the 3D object. The frequency of these distances will be represented by a normalized histogram. The values of these histograms feed a multi-layer neural network trained with backpropagation algorithm.

3.1. The discriminating shape function

The first challenge is to select the right function that provides a good signature for the shape of a 3D polygonal model. The distribution should be invariant under similarity transformations and tessella- tions. In addition, it should be insensitive to noise, cracks, and in- sertion or removal of small polygons [12]. We propose a method to rep- resent 3D objects with probability distributions of geometric properties computed from randomly selected points on the object's sur- face. Five features, including A3, D1, D2, D3, and D4, are proposed for measuring shape distributions.

- A3: Measures the angle between three random points on the surface of a 3D model.
- D1: Measures the distance between a fixed point and one random point on the surface. We use the centroid of the boundary of the model as the fixed point.
- D2: Measures the distance between two random points on the surface.
- D3: Measures the square root of the area of the triangle between three random points on the surface.

- D4: Measures the cube root of the volume of the tetrahedron between four random points on the surface.

After analysing the five functions, the authors conclude the distances distributions between two random points give the best features. They are invariant to tessellation of 3D polygonal models, since points are randomly selected from the object's surface. However, they are sensitive to small deformation due to noise, cracks, or insertion/removal of polygons, since sampling is area weighted.

3.2. Constructing shape distributions

After selecting a shape function, the issue becomes how to compute and store a representation of its distribution. Computing the D2 distribution is easy if you have a point set; the object must be meshed with good triangles. The mesh is defined by triangulation of the 3D object and given as a list of points followed by a list of facets or triangles. Therefore, one issue we must be concerned with is sampling density, so in our experiments, we took all possible points of the 3D object. A second issue is sample generation, as our shape functions are described in random points on the surface of a 3D model; we implemented the triangulation algorithm of Delaunay: for each selected triangle with vertices (A; B; C), we construct a point on its surface by generating two random numbers, r1and r2, between 0 and 1, and evaluating the following equation:

$$= (1 - \sqrt{r_1}) + \sqrt{r_1}(1 - r_2)B + \sqrt{r_1}r_2C \quad (1)$$

Intuitively, r1 sets the percentage from vertex A to the opposing edge, while r2 represents the percentage along this edge (see Figure 6). Taking the square-root of r1 gives a uniform random point regarding surface area

The figure below (see Figure 7) summarizes the construction steps of D2 distribution shape. This latter represents in a normalized histogram the probability of occurrence of a distance between two random points of the triangular faces being taken randomly.

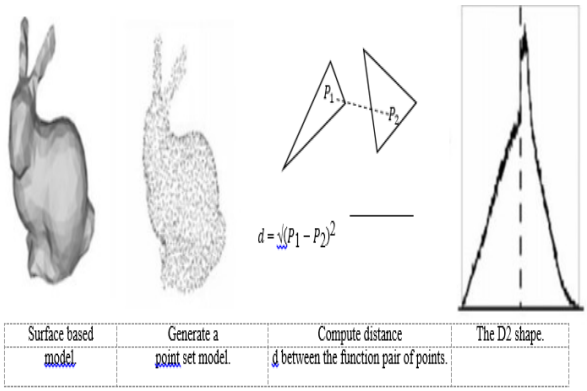


Fig. 7: Computing the D2 Shape Function from the Surface Based 3D Shape Model.

3.3. Classification and recognition using an

For a large database of 3D objects, searching for the closest models to the query object becomes difficult, which requires structuring indexes to reduce the search space. To overcome this problem, the research for similar 3D objects is reduced to a classification problem.The research result will be a group or class of three-dimensional models with similar characteristics to those of the query object. After constructing the shape distributions for two 3D models, we will use ANN to classify them.

Neural networks operate by apportioning the values of variables in neurons. These units are responsible to combine their information within each other to determine the value of the discrimination parameter. From the connection between these units emerges the ANN discrimination capacity. Each neuron receives digital information from neighboring neurons; to each value is associated a representative weight of the connection strength. Each neuron locally performs a calculation whose result is then transmitted to the downstream neurons. The multilayer perceptron (MLP) is a network layer propagation model (Figure 8). In our work, we use a supervised learning. The values of connection's weights are created initially at random, and the system searches by successive iterations to obtain data modelling. At each stage, an input is presented to the network; it spreads these values to output neurons. The calculated output is compared to the expected response, and the system modifies the weight. This alteration of the connections is obtained through backpropagation algorithm of the error gradient, where this calculation is back-propagating the errors identified in the outputs. In theory, we can never be sure this algorithm will eventually determine a set of weights suitable for all couples' Inputs-Outputs. In practice, we don't build a single ANN, but several models by varying the parameters of this algorithm and by seeking a model that fits the best the data.

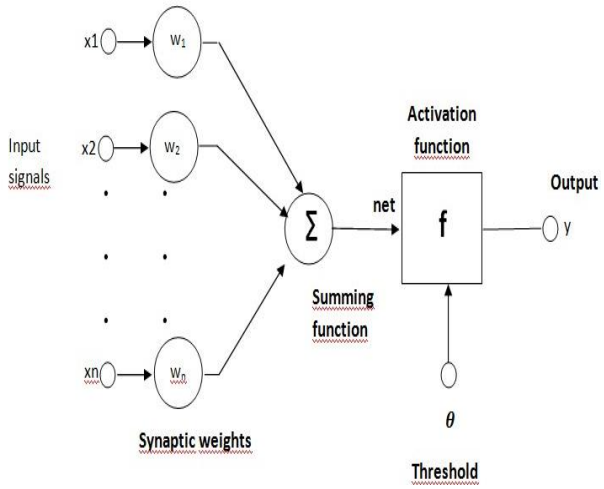


Fig. 8: The Architecture of the Multi-Layers Perception.

A multilayer perceptron gives the ability of learning and generalization. The idea consists of teaching this network several views from every possible object. At the end of the learning, the generalization capabilities of the multilayer perceptron allow him recognition on non-learned views. To reduce the amount of data to be processed, the object is characterized by its shape distribution.

4. Experimental results and discussion

We specified and experienced an approach for 3D object's recognition, based on its shape distribution. To test the effectiveness of this approach, we executed a series of shape matching experiments with a database that contains different 3D images in VRML Format. Our objects were extracted from Princeton benchmark standard database with ".off" extension. Figure 9 illustrates examples of 3D objects of each category of the 12 classes of the 3D object.

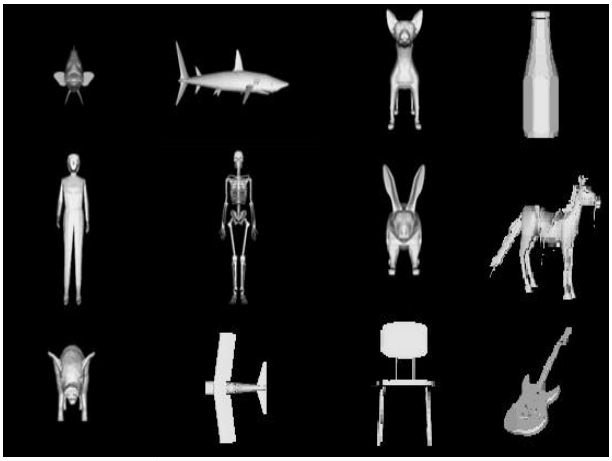


Fig. 9: Images of the Ten 3D Models Used in Our Initial Robustness.

Our 3D object recognition system can be summarized in Figure 4. At first, for each 3D object in the collection, a normalization stage is performed. Afterwards, the 3D object is described by its distribution in D2 form, which will be stored as a normalized histogram. The values of these histograms feed a multi-layer neural network trained by the backpropagation algorithm. The figure 10 shows the diversity within classes that contains pictures of three models from the shark, dog, women, rabbit, and skeleton. Note that the women are visually similar, while the dogs are different. Another class, rabbits, is even more diverse, where they are all similar in function, but different in shape

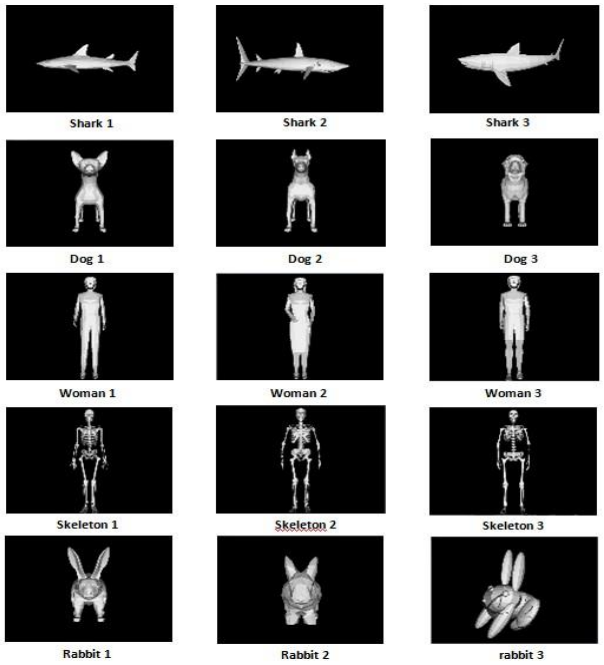


Fig. 10: Example Classes of Shapes in Our Database: Sharks, Dogs, Women, Skeleton and Rabbit.

To investigate the ability of our shape matching methods to discriminate between classes of objects, we run an experiment by computing D2 shape distributions for all models. Figure 11 shows examples. Examining these distributions qualitatively, we find the shape distributions for most objects within a single class are highly correlated, as multiple curves of the same class appear with the same form.

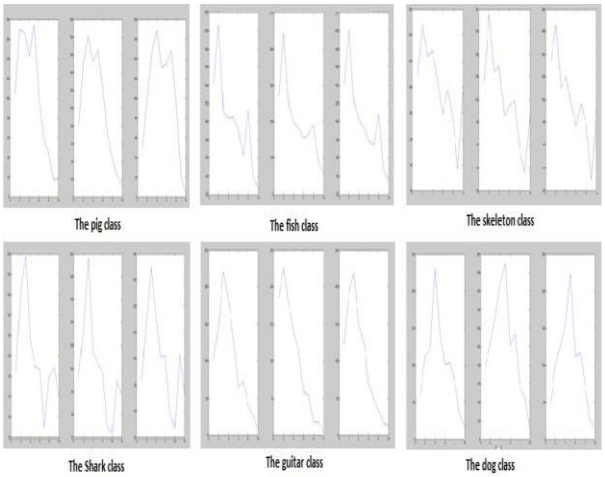


Fig. 11: D2 Shape Distributions for Some Objects. Each Plot Represents a Probability Distribution of Distance.

To classify the 3D objects characterized by their histograms, we used a system with 3 layers, driven by the backpropagation algorithm. The backpropagation algorithm requires a random initialization of synaptic weights. The network is connected between 2 layers and is thresholded. The input layer is supplied with the feature vectors. Once the learning is completed, the network can process new data. The neuron whose output is the highest determines the class. 10 classes were defined, and network output layer contains 10 neurons (one per class). The following table (Table 1) indicates the desired output vector for each of the ten classes:

Table 1: The Desired Output Vector for Each of the 10 Classes.

Class	Denomination	Desired outputs
1	Plan	(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
2	Bottle	(0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
3	Pig	(0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0)
4	Chair	(0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0)
5	Horse	(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0)
6	Dog	(0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)
7	Woman	(0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0)
8	Quitar	(0, 0, 0, 0, 0, 0, 0,1, 0, 0, 0, 0)
9	Rabbit	(0, 0, 0, 0, 0, 0, 0,0, 1, 0, 0, 0)
10	Fish	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0)
11	Shark	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0)
12	Skeleton	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1)

In this part, we will present the results out of the training and testing process of the algorithm and the performance of the neural network designed for 3D objects recognition. Our database contains 120 3D Objects. We used 70% in the learning phase and 30% in the test.
A different architecture of the networks has been tested. "Figure 12" shows the used neural network. If the number of neurons in the hidden layers is too large, the network will perform learning by heart of the training data and therefore have difficulty generalizing to new data. If it is too small, it will not possess enough internal variables to resolve the problem to treat. The choice of the number of neurons is a compromise between these two aspects. It's performed in an empirical way; we found that 32 neurons in the hidden layer give the best result.

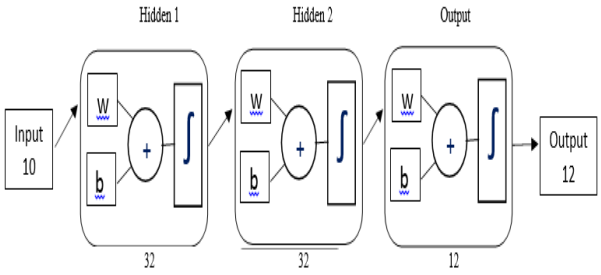


Fig. 12: The ANN Architecture.

4.1. Ann training

After the network has been structured for a specific application, it is ready to be trained. To start this process, the initial weights are chosen randomly. Then, the training or learning begins. Both the inputs and the outputs are given. The network processes the inputs and compares its results to the desired outputs. Errors are then transmitted back, to tell the system to adjust the weights that control the network. This process occurs repeatedly as the weights are continually tweaked. The set of data that enables the training is called the "training set." During the training of a network, the same set of data is processed many times as the connection weights are ever refined. The idea consists of training a neural network several objects of each class. At the end of the training phase, it should be able to recognize unlearned objects. Below, we give the confusion matrix corresponding to the learning phase. We obtain a recognition rate of 95,2 % with an error rate of 4.8% (see Table 2).

4.2. Ann testing

It is recommended to use an independent test base that hasn't been used during the training. The percentage of correct classifications will give a first indication. However, the performance with a test set is more appropriate. In the test phase, the output values of the input

data are compared to the desired ones. We attain a 91.7% recognition rate with 8.3% as error rate (see Table 2 and 3).

Table 2: Confusion Matrix for training set

Class	1	2	3	4	5	6	7	8	9	10	11	12	Rate Error
1	7	0	0	0	0	0	0	0	0	0	0	0	100% 0.0%
2	0	7	0	0	0	0	0	0	0	0	0	0	100% 0.0%
3	0	0	7	0	0	0	0	0	0	0	0	0	100% 0.0%
4	0	0	0	7	0	0	0	0	0	0	0	0	100% 0.0%
5	0	0	0	0	7	0	0	0	0	0	0	0	100% 0.0%
6	0	0	0	0	0	7	0	0	0	0	0	0	100% 0.0%
7	0	0	0	0	0	0	7	0	0	0	0	0	100% 0.0%
8	0	0	0	0	0	0	0	6	0	0	1	0	85.7% 14.3%
9	0	0	0	0	0	0	0	0	6	1	0	0	85.7% 14.3%
10	0	0	0	0	0	0	0	0	0	7	0	0	100% 0.0%
11	0	0	0	0	0	0	0	0	0	1	6	0	85.7% 14.3%
12	0	0	0	0	0	0	0	0	1	0	0	6	85.7% 14.3%
Rate/Error	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	85.7% 14.3%	77.8% 22.2%	85.7% 14.3%	100% 0.0%	**95.2%** **4.8%**

Table 3: Confusion Matrix for testing set

Class	1	2	3	4	5	6	7	8	9	10	11	12	Rate/Error
1	3	0	0	0	0	0	0	0	0	0	0	0	100% 0.0%
2	0	3	0	0	0	0	0	0	0	0	0	1	75.0% 25.0%
3	0	0	3	0	0	0	0	0	0	0	0	0	100% 0.0%
4	0	0	0	3	0	0	0	0	0	0	0	0	100% 0.0%
5	0	0	0	0	3	0	0	1	0	0	0	0	75.0% 25.0%
6	0	0	0	0	0	3	0	0	0	0	0	0	100% 0.0%
7	0	0	0	0	0	0	3	0	1	0	0	0	75.0% 25.0%
8	0	0	0	0	0	0	0	2	0	0	0	0	100% 0.0%
9	0	0	0	0	0	0	0	0	2	0	0	0	100% 0.0%
10	0	0	0	0	0	0	0	0	0	3	0	0	100% 0.0%
11	0	0	0	0	0	0	0	0	0	0	3	0	100% 0.0%
12	0	0	0	0	0	0	0	0	0	0	0	2	100% 0.0%
Rate/Error	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	66.7% 33.3%	66.7% 33.3%	100% 0.0%	100% 0.0%	66.7% 33.3%	**91.7%** **8.3%**

In this paragraph, we compare this ANN classifier's results to other methods, such as the Support-Vector-Machines (SVM) and k-nearest neighbour (KNN) method, which are the simplest to implement. They become a widely and successfully applied algorithm in numerous applications, like data mining domain and recognition of statistical forms.

To determine the correct kernel for our data, we have studied and compared the performance of the different linear and nonlinear types (polynomial and RBF) for both learning and test databases (See table 4).

Table 4: The Recognition Rate Obtained from the Training and Testing Sets Using Sum Methods with Different Kernels

Datasets	Linear	Polynomial	RBF
Training set	89.1%	83.6%	83.6%
Testing set	79.6%	73.8%	78.3%

We investigated SVM classifier performance with various kernels and found linear SVM perform the best in terms of recognition rate. As table 5 shows, the ANN obtains the best result regarding KNN method and SVM. The overall recognition rate is of 91.7% to be compared with an 80.3% obtained by KNN and 79.6% obtained by SVM. This proves the effectiveness of this technique for 3D objects recognition.

Table 5: The Recognition Rate for SVM, KNN and ANN

Datasets	SVM(linear)	KNN	ANN
Training set	89.1%	92.4%	95.2%
Testing set	79.6%	80.3%	91.7%

5. Conclusion

In this paper, we discussed the recognition of 3D objects. Specifically, we focused on the classification of objects based on the overall geometry. Our choice of processing these data was motivated by the huge number of items available today, which is constantly growing. The aim of this work is to present a new method to recognize 3D polygonal objects. To this end, we thought to combine the D2 shape distribution and artificial neural networks. The idea was to extract, in objects, some relevant information of the shape. Such information enables efficient classification of 3D objects. The challenging aspect of this paper is to find a suitable shape signature that can be constructed and classified quickly. The method used the learning and generalization abilities of a multilayer perceptron. The idea consists of teaching this network several examples of each possible class. At the end of the learning, the multilayer perceptron can recognize non-learned objects. Since the method distinguishes the overall shape of the object, it appears to be more suited to research of similar objects in databases in multiple shapes. In addition, to evaluate the efficiency of our method, we compare our classifier with Support vector machine and k-nearest neighbours. The simulation results are encouraging and demonstrate the performance of the proposed approach, with a value exceeding 91.7%.
For future work, several paths can be considered at the conclusion of this work. First, we plan to test this method on other bases and consider more classes to check if the approach is generic. Another approach would be applying the method with deep neural networks, using multiple hidden layers.

References

[1] Al-Batah, M.S., Mat Isa N.A., Zamli, K .Z. and Azizli, K.A. (2010) 'Modified Recursive Least Squares algorithm to train the Hybrid Multilayered Perceptron (HMLP) network', Applied Soft Computing, Vol. 10, No. 1.
[2] Ankerst, M., Kastenmüller, G., Kriegel, H.P. and Seidl, T. (1999) '3D Shape Histograms for Similarity Search and Classification in Spatial Databases', in Advances in Spatial Databases, Springer.
[3] Ansary, T.F. (2006) 'Model retrieval using 2d characteristic views'.
[4] Ansary, T.F., Daoudi, M. and Vandeborre, J.Ph. (2007) 'A bayesian 3d search engine using adaptive views clustering', In IEEE Transactions on Multimedia.
[5] Khosravi M., Mosaddeghi F., Oveisi, M., khodayari-b, A., Aerodynamic drag reduction of heavy vehicles using append devices by CFD analysis, Journal of Central South University, Volume 22, 2015, pp 4645–4652.
[6] Assfalg, J., Bimbo, A.D. and Pala, P. (2003) 'Retrieval of 3D objects using curvature maps and weighted walkthroughs'.
[7] Husnutdinov D. H., Aydarova S. H., Sagdieva R. K., Mirzagitov R. H., Tsaran A., Plotnikova H., Velikanova S. Information and Communication Tools for Tatar Language teaching, Astra Salvensis, Supplement No. 2, 2017, p. 15..
[8] Baareh, A.K., Sheta, A.F. and Al-Batah, M.S. (2012) 'Feature based 3D Object Recognition using Artificial Neural Networks', International Journal of Computer Applications, Vol. 44, No.5.
[9] Baareh, A.K., Sheta, A.F. and AL Khnaifes, K. (2006) 'Forecasting River Flow in the USA, A Comparison between AutoRegression and Neural Network Non-Parametric Models', Journal of Computer Science, Vol. 2.
[10] Baeza-Yates, R. and Valiente, G. (2000) 'An image similarity measure based on graph matching', Proceedings of the Seventh International Symposium on String Processing, Information Retrieval.
[11] Khodayari Bavil, A., Razavi, S.E., on the thermo-flow behavior in a rectangular channel with skewed circular ribs, Mechanics & Industry, 18 2 (2017) 225, https://doi.org/10.1051/meca/2016057.
[12] Beis, J.S. and Lowe, D.G. (1999) 'Indexing without invariants in 3D object recognition', IEEE Trans. Pattern.
[13] Thakurdesai PA, Kole PL & Pareek RP (2004), Evaluation of the quality and contents of diabetes mellitus patient education on Internet. Patient Education and Counseling 53, 309–313.
[14] J Kim, B Kim, S Savarese, Comparing Image Classification Methods: K-Nearest-Neighbor and Support-Vector-Machines. Applied Mathematics in Electrical and Computer Engineering (2012).