Journal of Software Engineering and Applications, **2011, 4, 306-310**
doi:10.4236/jsea.2011.45033 Published Online May 2011 (http://www.SciRP.org/journal/jsea)

3D Object Recognition by Classification Using Neural Networks

Mostafa Elhachloufi[1], Ahmed El Oirrak[1], Aboutajdine Driss[2], M. Najib Kaddioui Mohamed[1]

[1]University Cady Ayyad, Faculty Semlalia, Department of Informatics, Marrakech, Morocco; [2]University Mohamed V, Faculty of Science, Department of Physique, LEESA-GSCM, Rabat, Morocco.
Email: elhachloufi@yahoo.fr, oirrek@yahoo.fr, aboutajadine@fsr.ac.ma, kaddioui@yahoo.fr

Received March 27th, 2011; revised April 28th, 2011; accepted May 12th, 2011.

ABSTRACT

In this Paper, a classification method based on neural networks is presented for recognition of 3D *objects. Indeed, the objective of this paper is to classify an object query against objects in a database, which leads to recognition of the former.* 3D *objects of this database are transformations of other objects by one element of the overall transformation. The set of transformations considered in this work is the general affine group.*

Keywords: *Recognition, Classification,* 3D *Object, Neural Network, Affine Transformation*

1. Introduction

The growing number of 3D objects available on the Internet or in specialized databases, mandates the establishment of methods to develop techniques for recognition and description to access the content of these smartly objects. However the objects' 3D object recognition has been as extensive and has become very important in many areas.

In this framework, several approaches exist: In terms of statistical approaches, the statistical shape descriptors for recognition generally consist of either calculating various statistical moments [1-3] or estimating the distribution of the measurement of a given geometric primitive, which is deterministic [3] or random [2]. Among the approaches by statistical distribution, we mention the specter of a 3D shape (3D SSD) [4] which is invariant to geometric transformations and algebraic invariants [5], providing of global descriptors, expressed in terms of moments of different orders. For structural approaches, the approaches representing the segmentation of a 3D object into plot of land and representation by the adjacency graph are presented in [6] and [7].

In the same vein, Tang elder *et al.* [8] have developed an approach based on representations by points of interest. In approaches by transformation a very rich literature emphasizes any interest in approaches based on Haugh transformation [9-11] which include detecting different varieties of (n − 1) dimension diving into the space.

In the same vein, this work focuses on defining a method that allows the recognition of 3D objects by classification based on neural networks. Classification is a computing tool that expects as input a list of numbers, and which provides, at its output, an indication of class. A classifier must be able to model the best borders that separate classes from each other. This modeling uses the concept of discriminant function, which allows to express the classification criterion. Its role is to determine, among a finite set of classes, to which class a particular item belongs.

2. Representation of 3D Objects

3D object is represented by a set of points denoted $M = \{P_i\}_{i=1,\cdots,n}$ where $P_i = (x_i, y_j, z_i) \in \Re^3$, arranged in a matrix $\boldsymbol{X}$. Under the action of an affine transformation, the coordinates (x, y, z) are transformed into other coordinates $(\tilde{x}, \tilde{y}, \tilde{z})$ by the following procedure:

$$\begin{cases} f : \mathbb{R}^3 \to \mathbb{R}^3 \\ \boldsymbol{X}\left(x(t), y(t), z(t)\right) \to f(\boldsymbol{X}) = \boldsymbol{Y}\left(\tilde{x}(t), \tilde{y}(t), \tilde{z}(t)\right) \\ \boldsymbol{Y} = \boldsymbol{A} \times \boldsymbol{X}\left(x(t), y(t), z(t)\right) + \boldsymbol{B} \end{cases}$$

with $\boldsymbol{A} = \left(a_{i,j}\right)_{i,j=1,2,3}$ invertible matrix associated with $\boldsymbol{B}$ and $\mathbb{R}^3$ is a vector translation in $\Re^3$.

3. Classification

Classification is a research area that has been developed in the sixties. It is the basic principle of multiple support systems for diagnosis. It assigns a set of objects to a set of classes according to the description thereof. This description is done through properties or specific conditions typical to these classes.

Objects are then classified according to whether or not they check these conditions or properties. Classification methods can be supervised or unsupervised.

Supervised methods require the user a description of the classes while those unsupervised are independent of the user. Rather they are methods of statistical grouping that sort objects according to their properties and form sets with similar characteristics.

4. The Artificial Neural Networks

The artificial neural networks (ANN) are mathematical models inspired by the structure and behavior of biological neurons [12]. They are composed of interconnected units called artificial neurons capable of performing specific and precise functions [13]. ANN can approximate nonlinear relationships of varying degrees of complexity and significant to the recognition and classification of data. **Figure 1** illustrates this situation.

4.1. Architecture of Artificial Neural Networks

For an artificial neural network, each neuron is interconnected with other neurons to form layers in order to solve a specific problem concerning the input data on the network [14,15].

The input layer is responsible for entering data for the network. The role of neurons in this layer is to transmit the data to be processed on the network. The output layer can present the results calculated by the network on the input vector supplied to the network. Between network input and output, intermediate layers may occur; they are called hidden layers. The role of these layers is to transform input data to extract its features which will subsequently be more easily classified by the output layer. In these networks, information is propagated from layer to layer, sometimes even within a layer via weighted connections.

A neural network operates in two consecutive phases: a design phase and use phase. The first step is to choose the network architecture and its parameters: the number of hidden layers and number of neurons in each layer. Once these choices are fixed, we can train the network. During this phase, the weights of network connections and the threshold of each neuron are modified to adapt to different conditions of input. Once the training on this network is completed, it goes into use phase to perform the work for which it was designed.

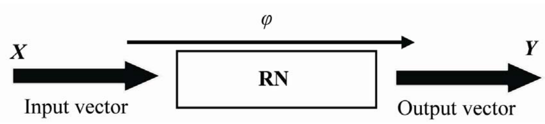

Figure 1. Black box of artificial neural networks.

4.2. Multilayer Perceptron

For a multilayer network, the number of neurons in the input layer and output layer is determined by the problem to be solved [14-16]. The architecture of this type of network is illustrated in **Figure 2**. According to R. LEPAGE and B. Solaiman [14],the neural network has a single layer with a hidden number of neurons approximately equal to: $J = 1 + \sqrt{N(M+2)}$ where:

N : number of input parameters.

M : the number of neurons in the output layer.

4.3. Figures and Tables

The learning algorithm used is the gradient back propagation algorithm. [17] This algorithm is used in the feed forward type networks, which are networks of neurons in layers with an input layer, an output layer, and at least one hidden layer. There is no recursion in the connections and no connections between neurons in the same layer.

The principle of backpropagation is to present the network a vector of inputs to the network, perform the calculation of the output by propagating through the layers, from the input layer to the output layer through hidden layers. This output is compared to the desired output, an error is then obtained. From this error, is calculated the gradient of the error which in turn is propagated from the output layer to the input layer, hence the term back propagation. This allows modification of the weights of the network and the refore learning. The operation is repeated for each input vector and that until the stop criter ion is verified [18].

4.4. Learning Algorithm

The objective of this algorithm is to minimize the maximum possible error between the outputs of the network

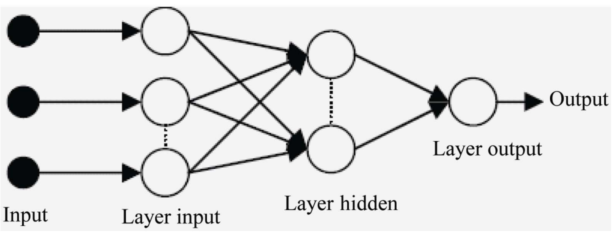

Figure 2. Architecture of a multilayer perceptron network.

(or calculated results) and the desired results. We spread the signal forward in the layers of the neural network: $x_k^{(n-1)} \mapsto x_j^{(n)}$ The spread forward is calculated using the activation function, the aggregation function h (often a scalar product between the weights and the inputs of the neuron) and synaptic weight w_{jk} between the neuron $x_k^{(n-1)}$ and the neuron $x_j^{(n)}$ as follows:

$$x_k^{(n)} = g^{(n)}\left(h_j^{(n)}\right) = g^{(n)}\left(\sum_k w_{jk}^{(n)} x_k^{(n-1)}\right) \qquad (1)$$

When the forward propagation is complete, we get the output result y. It then calculates the error between the output y given by the network and the desired vector s.

For each neuron i in output layer is calculated:

$$e_i^{sortie} = g'\left(h_i^{sortie}\right)\left[s_i - y_i\right] \qquad (2)$$

It propagates the error backward $e_i^{(n)} \mapsto e_j^{(n-1)}$ through the following formula:

$$e_i^{(n-1)} = g'^{(n-1)}\left(h_i^{(n-1)}\right)\sum w_{ij} e_i^{(n)} \qquad (3)$$

It updates the weights in all layers:

$$\Delta w_{ij}^{(n)} = \lambda e_i^{(n)} X_j^{(n-1)} \qquad (4)$$

where Δ is the learning rate (of low magnitude and less than 1.0).

5. Principle of the Proposed Method

The principle of the proposed method is as follows:

Step 1:
Given two 3D objects (object of database) $\boldsymbol{X}$ and $\boldsymbol{Y}$ (query object) that seeks to verify if they are related by a linear transformation and therefore are part of the same class (class of objects and transformed by an affine transformation), so we take as a first step r random samples of size p of points $\boldsymbol{X}$ respectively $\boldsymbol{Y}$ points named $\left(x_{e_1}, x_{e_2}, ..., x_{e_r}\right)$ respectively $\left(y_{e_1}, y_{e_2}, ..., y_{e_r}\right)$. After that we study the association between the samples x_{e_0} and y_{e_0} $i_0 \in [1,2,...,r]$. To do this we extract the parameters α_{e_0} and β_{e_0} that can transmit x_{e_0} to y_{e_0} as follows :

$$y_{e_{i_0}} = \alpha_{i_0} x_{e_{i_0}} + \beta_{i_0} \qquad (5)$$

using neural networks as shown in the **Figure 3**.

Step 2:
In this step, we first calculates the points of the vector $y_{e_i}^c$ using the previously extracted parameters α_{i_0} and β_{i_0} by the Formula (1) as follows:

$$y_{e_j}^c = \alpha_{i_0} x_{e_j} + \beta_{i_0} \qquad (6)$$

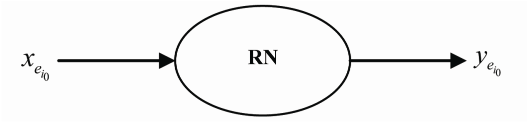

Figure 3. Parameters extraction α_{i_0} and β_{i_0}.

and $\forall j \in [1,2,\cdots,r]$ as such $j \neq i_0$, then we proceed to calculate the errors err_j defined as follows:

$$err_j = y_{e_j} - y_{e_j}^c = \sum_k \left(y_{e_j}(k) - y_{e_j}^c(k)\right)^2 \qquad (7)$$

corresponding to the pairs of samples $\left(y_{e_j}, y_{e_j}^c\right)$ which k represents the number of vector elements x_{e_j}.

Step 3:
This step involves the classification of objects using multilayer neural networks whose input vector is the vector of errors: $VectErr = \left[err_0, err_1, ..., err_r\right]$ and the output is the class c, with $c = 1$ (class c_1) if $\varphi\left(VectErr\right) \le \delta$ else $c = 0$ (class c_0) where δ a preset threshold small enough (very close to zero).

If $c = 1$ so all errors are part of the same class, a class where errors are very close to zero, *i.e.* that:

$$err_k \simeq 0 \Rightarrow y_{e_k} \simeq y_{e_k}^c, \Rightarrow y_{e_k} = \alpha_{i_0} x_{e_k} + \beta_{i_0} \qquad (8)$$

$\forall k \in [1,2,...,r]$

This means that all points of $\boldsymbol{X}$ are converted into points of $\boldsymbol{Y}$ by the same parameters α_{e_0} and $\beta_{e_0} \Rightarrow \boldsymbol{Y}$ is an affine transformation $\boldsymbol{X}$. Else, this is not the case of an affine transformation of $\boldsymbol{X}$ into $\boldsymbol{Y}$.

6. Results and Discussion

Consider two 3D objects X (object of a database) and Y (query object) related by an affine transformation (**Figure 4** and **Figure 5**) and divided into 20 samples. The goal is to reach neural structures capable of recognizing

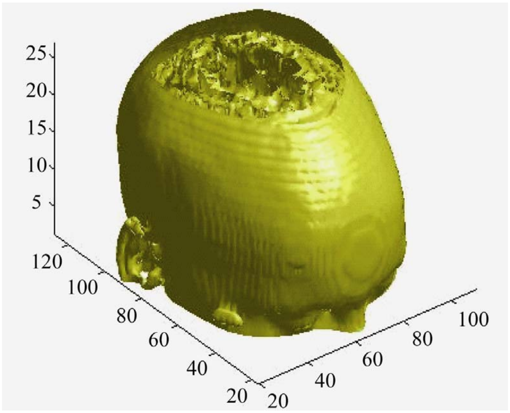

Figure 4. Origin Object.

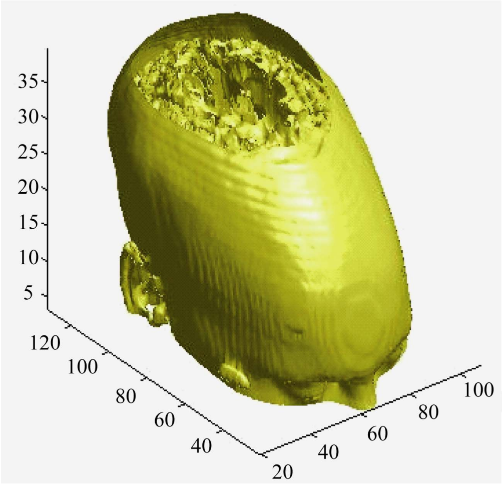

Figure 5. Transformed object.

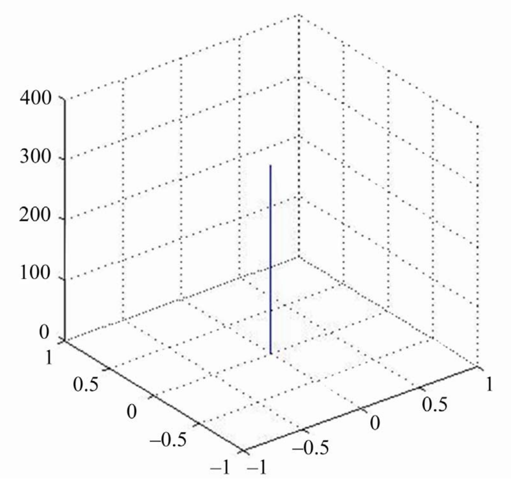

Figure 6. Representation errors err_i.

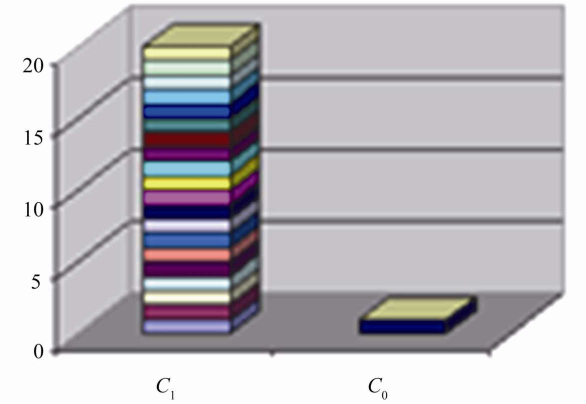

Figure 7. Affect of error classes built.

the errors corresponding to samples of objects belonging to the same class. The learning base is comprised of input variables (errors) and desired outputs variables (classes). The desired output will serve as a comparison with the calculated output during the learning phase. After training the network, tests were conducted on a number of data (errors) to check their performance. According to the results (**Figure 6** and **Figure 7**) of the validation test we notice that over 95% of errors are classified in the class c_1, which shows that according step 3 $\boldsymbol{Y}$ is an affine transformation of $\boldsymbol{X}$.

7. Conclusions

In this work, we presented a classification method for recognizing 3D objects. Indeed, we have developed an approach based on neural networks which is a first step to split the objects into *n* samples and then calculate the errors for these samples. Using the proposed method, they will be subject to classification for the recognition of these objects. Recognition is performed using the classification errors rate corresponding to objects (object of origin and its clone) as shown in **Figure 6**. The simulation results were presented and an evaluation of the designed system has been made. They were generally satisfactory and show the validity of the proposed method.

REFERENCES

[1] T. Murao, “Descriptors of Polyhedral Data for 313-Shape Similarity Search,” *Proposal P*177, *MPEG-*7 *Proposal Evaluation Meeting*, Lancaster, February 1999.

[2] M. Elad, A. Tal and S. Ar, “Directed Search in a 3D Objects Database Using SVM,” *Hewlett-Packard Research Report HPL*-2000-20*R*1, 2000.

[3] C. Zhang and T. Chen, “Efficient Feature Extraction for 2D/3D Objects in Mesh Representation,” *Proceeding of the International Conference on Image Processing* (*ICIP* 2001), Thessaloniki, Greece, 2001.

[4] T. Zaharia and F. Prêteux, “3D-Shape-Based Retrieval within the MPEG-7 Framework,” *Proceeding SPIE Conference on Nonlinear Image Processing and Pattern Analysis XII*, San Jose, Vol. 4304, 2001, pp. 133-145.

[5] G. Taubin and D. B. Cooper, “Object Recognition Based on Moment (or Algebraic) Invariants,” In: J. L. Mundy and A. Zisserman, Eds., *Geometric Invariants in Computer Vision*, MIT Press, Cambridge, 1992.

[6] S. J. Dickinson, D. Metaxas and A. Pentland, “The Role of Model-Based Segmentation in the Recovery of Volumetric Parts from Range Data,” *IEEE Transactions on PAMI*, Vol. 19, No. 3, 1997, pp. 259-267. doi:10.1109/34.584104

[7] S. Dickinson, A. Pentland and S. Stevenson, “Viewpoint-Invariant Indexing for Contentbased Image Retrieval,” *Proceedings of the* 1998 *International Workshop on Content-Based Access of Image and Video Databases*, Washington, 3 January 1998.

[8] J. W. H. Tangelder and R. C. Veltkamp, “Polyhedral

Model Retrieval Using Weighted Point Sets," *Rapport Technique No UU-CS*-2002-019, Universitd de Utrecht, Pays-Bas, 2002.

[9] P. V. C. Hough, "Method and Means for Recognizing Complex Patterns," US Patent 3 069 654, 1962.

[10] D. H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, Vol. 13, No. 2, pp. 111-122, 1981. doi:10.1016/0031-3203(81)90009-1

[11] J. Illingworth and J. Kittler, "A Survey of the Hough Transform," *Computer Vision*, *Graphics and Image Processing*, Vol. 44, No. 1, pp. 87-116, 1988. doi:10.1016/S0734-189X(88)80033-1

[12] A. Benyettou, A. Mesbahi, H. Abdoune and A. Ait-ouali, "La Reconnaissance de Formes Spatio-Temporelles par les Réseaux de Neurones a Délais Temporels," *Conference Nationale sur L'Ingénierie de L'Electronique—CNIE*'02, University USTOran, Algérie, 2002 .

[13] B. Muller, J. Reinhardt and M. T. Strckland, "Neural Networks: An Introduction," Springer-Verlag, Berlin, 1995.

[14] R. Lepage and B. Solaiman, "Les Réseaux de Neurones Artificiels et Leurs Applications en Imagerie et en Vision par Ordinateur," *Ecole de Technologie Supérieure*, 2003.

[15] I. Khanfir, K. Taouil, M. S. Bouhlel and L. Kamoun, "Strategie de Traitement des Images de Lesions Dermatologiques," In: M. S. Bouhlel, B. Solaiman and L. Kamoun, Eds., *Sciences Electronique*, *Technologies de L'Information et des Télécommunications*, ISBN 9973-41-685-6, 2003.

[16] I. Maglogiannis, P. D. Koutsouris and D. Koutsouris, "An Integrated Computer Supported Acquisition, Handling, and Characterization System for Pigmented Skin Lesions in Dermatological Images," *IEEE Transactions on Information Technology in Biomedicine*, Vol. 9, No. 1, March 2005, pp. 86-98. doi:10.1109/TITB.2004.837859

[17] S. E. Fahlman, "An Empirical Study of Learning Speed in Backpropagation Networks," Computer Science Department, Carhengie Mellon University, Pittsburgh, 1988.

[18] F. Bloyo and M.Verleysen, "Les Réseaux de Neurones Artificiels," Presse Universitaire de France, Paris, 1996.