3D Recognition Using Neural Networks

Elhachloufi Mostafa

Faculty of Science Semlalia, Dept. Computer Science, University Caddy Ayyad, Marrakech, Morocco

Tel: 212-667-879-031 E-mail: elhachloufi@yahoo.fr

El Oirrak Ahmed

Faculty of Science Semlalia, Dept. Computer Science, University Caddy Ayyad, Marrakech, Morocco

Tel: 212-675-367-724 E-mail: aboutaj@fsr.ac.ma

Aboutajdine Driss

Faculty of Science, Dept. Computer Science, University Mohamed V, Marrakech, Morocco

Tel: 212-665-767-028 E-mail: oirrek@yahoo.fr

Kaddioui Mohamed Najib

Faculty of Science Semlalia, Dept. Computer Science, University Caddy Ayyad, Marrakech, Morocco

Tel: 212-661-347-099 E-mail: kaddioui@ucam.ac.ma

Received: November 17, 2011 Accepted: December 5, 2011 Published: March 1, 2012

doi:10.5539/cis.v5n2p105 URL: http://dx.doi.org/10.5539/cis.v5n2p105

Abstract

With the advent of the Internet, exchanges and the acquisition of information, description and recognition of 3D objects have been as extensive and have become very important in several domains, which require the establishment of methods to develop description and recognition techniques to access intelligently to the contents of these objects.

This paper deals for 3D models recognition. Thus under general affine transform we propose an approach based on neural network.

The recognition is done by measuring the similarity between a sample of object and its transformed obtained by parameters extracted from neural networks using the euclidean distance.

Keywords: 3D Object, Recognition, Neural networks, Affine transformed

1. Introduction

Databases of 3D objects that are used in different fields (e-commerce, gaming, medical, *et al.*) become bigger, which mandates the establishment of effective methods for users to find the corresponding 3D objects to the request object.

In this context, several research systems by the content search engines and 3D models are available on the web. Many methods have been developed in this direction.

The shape spectrum descriptor proposed by Zaharia et al based on surface geometry, is recommended by MPEG-7.

Filali et al. proposed the descriptor based on 2D views named Adaptive Views Clustering (AVC). It is a probabilistic Bayesian method that selects the most interesting views from several views of a 3D model.

Based on the statistics, Osada et al. proposed the descriptor named shape distribution.

(D2). Paquet et al. proposed the method of cord histograms.

The ratio area/volume is used as a feature vector to describe the 3D models by Zhang and Chen. Even if this

descriptor computes the feature vectors easily and quickly, it needs a high quality of mesh.

Vranic and Saupe proposed the method named Ray based descriptor. It uses the extents obtained from the center of mass of the model to intersection with its surface in directions which are constructed by an icosahedron. This approach is not robust to noise and needs a high dimension of feature vectors, this is why the authors construct the feature vectors from the complex function on a sphere, composed with Ray based feature vectors and shading based feature vectors, presented in frequency domain by applying the spherical harmonics.

2. Representation of 3D Objects

3D object is represented by a set of points denoted $M = \{P_i\}_{i=1.....n}$ where $P_i = (x_i, y_j, z_i) \in \Re^3$, arranged in a matrix 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 equation:

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

With A invertible matrix and B is a vector translation.

3. The Artificial Neural Networks

The artificial neural networks (ANN) are mathematical models inspired by the structure and behavior of biological neurons. They are composed of interconnected units called artificial neurons capable of performing specific and precise functions.

ANN can approximate nonlinear relationships of varying degrees of complexity and significant to the recognition and classification of data. Figure1 illustrates this situation.

3.1 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.

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.

3.2 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. The architecture of this type of network is illustrated in Figure 2.

According to R. LEPAGE and B. Solaiman, 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.

3.3 Learning

The learning algorithm used is the gradient back propagation algorithm 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 criterion is verified.

3.4 Learning Algorithme

The objective of this algorithm is to minimize the maximum possible error between the outputs of the 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 g , 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:

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

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 an output layer is calculated:

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

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

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

It updates the weights in all layers:

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

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

4. Recognition for 3D Volume Using Neural Networks Applied on 2D Slice

Typical scalar volume data is composed of a 3-D array of data and three coordinate arrays of the same dimensions. The coordinate arrays specify the x-, y-, and z-coordinates for each data point. The units of the coordinates depend on the type of data. For example, flow data might have coordinate units of inches and data units of psi. Slice plans provide a way to explore the distribution of data values within the volume by mapping values to colors.

We can orient slice plans at arbitrary angles, as well as use nonplanar slices. In this case each volume V is characterized by a set of slices, each slice is a 2D image having coordinates (x, y) and color information.

In this experiment only slices 1, 8 and 27 are considered. For each one six normalized color coefficients are extracted.

This technique supposes that V and $\tilde{V}$ are stocked with the same number of slices. We can use another

technique based on isosurface. An isosurafce define the 3D volume bounded by a particular isovalue. The volume inside contains values greater (or less) than the isovalue. The volume outside contains values less (or greater) than the isovalue. The isovalue can be an interval [min, max] in this case the isosurface is a list of all cells for which the isovalue is contained in the interval [min, max]. Also the choice of isovalue on V and $\tilde{V}$ is another problem. Figure 4 shows some subvolumes obtained by this technique.

Divided 3D volume V and it's transformed $\tilde{V}$ into five slices $(S_1,...,S_5)$ respectively $(\tilde{S}_1,...,\tilde{S}_5)$ as shown in figure 5 and figure 6. Those volumes figure 1 are initially stocked into 24 slices. We take as a first step two points: $p_{i_0} = (x_{i_0}, y_{i_0}) \in S_{i_0}$ and $\tilde{p}_{i_0} = (\tilde{x}_{i_0}, \tilde{y}_{i_0}) \in \tilde{S}_{i_0}$.

After we study the relation between p_{i_0} and $\tilde{p}_{i_0}$. To do this we extract the parameters α_0 and β_0 that can transmit p_{i_0} and $\tilde{p}_{i_0}$ as follows:

$$\tilde{p}_{i_0} = \alpha_0 . p_{i_0} + \beta_0 \tag{5}$$

using neural networks as shown in the figure below:

In our case, the structure of neural network used has two inputs corresponding to the coordinates of points of the slice and a single hidden layer of one neuron and two outputs corresponding to the coordinates of points of the slice. The training set consists of two samples of slices and the transfer function is the linear function.

In the second step, we calculate the points $\hat{p}_j$ using the previously extracted parameters α_0 and β_0 by the formula (1) as follows:

$$\hat{p}_j = \alpha_0 . p_j + \beta_0 \tag{6}$$

then we proceed to calculate the errors err_j defined as follows:

$$err_j = \hat{p}_j - \tilde{p}_j \tag{7}$$

corresponding to the pairs of samples $(p_j, \tilde{p}_j)$, with $j = 1,...,5$.

According to the table above we conclude that all points of $S_i (i = 1,...,5)$ are converted into points of $\tilde{S}_i (i = 1,...,5)$ by the same parameters α_0 and $\beta_0 \Rightarrow \tilde{S}$ is an affine transformation $S \Rightarrow \tilde{V}$ is an affine transformation V.

6. Conclusion

In this work, we have developed an approach for recognizing 3D objects based on neural networks. The principal advantage of proposed approaches is the possibility to handle affine transform and 3D volume.

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 approach.

In future the 3D search engine can be extended for medical databases (MRI magnetic resonance imaging), this data typically contains a number of slice plans taken through a volume, such as the human body.

References

A. Benyettou, A. Mesbahi, H. Abdoune, & A. Ait-ouali. (2002). La reconnaissance de formes spatio-temporelles par les réseaux de neurones à délais temporels, Conf. Nationale sur l'Ingénierie de l'Electronique –CNIE'02, pp. 159-163, univ. USTOran, Algérie, 15-16 Décembre 2002.

B. Muller, J. Reinhardt, & M. T. Strckland. (1995). Neural networks an introduction Springer-verlang Berlin Heidlberg.

Chen D. Y., Ouhyoung M., Tian X. P., & Shen Y. T. (2003). On visual similarity based 3d model retrieval. In *Eurographics*, 223-232, Granada, Spain.

D.V. Vranic & D. Saupe. (2001). 3d model retrieval with spherical harmonics and moments. In *B. Radig and S. Florczyk (Eds.), DAGM 2001*, 392-397, Munich, Germany.

E. A. Lmaati, A. El Oirrak, & M. N. Kaddioui. (2009). A 3d search engine based on 3d Curve analysis. *SIViP: Signal, Image and Video processing*.

E. A. Lmaati, A. El Oirrak, & M. N. Kaddioui. (2010). 3d model retrieval based on 3d Discrete cosine transform. *IAJIT, International Arab Journal of Information Technology*.

E. Paquet & M. Rioux. (1999). A query by content software for three-dimensional models databases management In *Conf. on Recent Advances 3-D Digital Imaging and Modeling*, 345-352.

F. Bloyo-M.Verleysen. (1996). Les réseaux de neurones artificiels. Presse Universitaire de France.

I. Khanfir, K. Taouil, M. S. Bouhlel, & L. Kamoun. (2003). Strategie de traitement des images de lesions dermatologiques, Sciences Electronique, Technologies de l'Information t des Télécommunications, ed M.S.Bouhlel, Solaiman et L.Kamoun. ISBN 9973-41- 685-6, Mars 2003.

Maglogiannis I. Pavlopoulos, D. Koutsouris & D. Koutsouris. (2005). An Integrated Computer Supported Acquisition, Handling, and Characterization System for Pigmented Skin Lesions in Dermatological Images" IEEE Transactions on Information Technology in Biomedicine, 9(1), March 2005. http://dx.doi.org/10.1109/TITB.2004.837859

MPEG-7 Video Group. (2001). Information Technology Multimedia Content Description Interface. Part 3: Visual, ISO/IEC FCD, 15938:3/N4062, MPEG-7.

R. Lepage, & B. Solaiman. (2003). Les réseaux de neurones artificiels et leurs applications en imagerie et en vision par ordinateur, Ecole de technologie supérieure.

R. Osada, T. Funkhouser, B. Chazelle, & D. Dobkin. (2001). Matching 3d models with shape distributions. In *SMI*, pages 154-166, Genova, Italy.

S. E Fahlman. (1988). An empirical study of learning speed in backpropagation networks",Carhengie Mellon University Computer Science Department, 1988.

T. Filali Ansary, M. Daoudi, & J. P. Vandeborre. (2007). A bayesian 3d search engine using adaptive views clustering. *IEEE Transaction on Multimedia*, 9, 78-88. http://dx.doi.org/10.1109/TMM.2006.886359

Table 1. Representation of errors

	$\left(p_j, \tilde{p}_j\right)$
err_1	0.0008 (10^{-4})
err_2	0.0007 (10^{-4})
err_3	0.00013 (10^{-4})
err_4	0.00024 (10^{-4})
err_5	0.00037 (10^{-4})

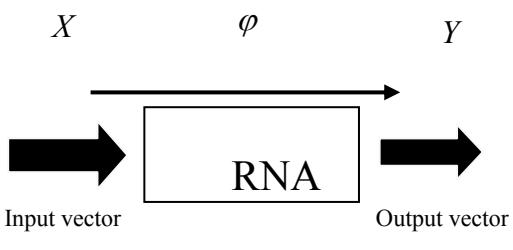


Figure 1. Black box of Artificial Neural Networks

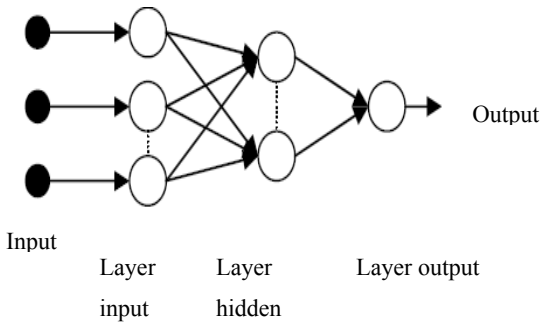


Figure 2. Architecture of a multilayer perceptron network

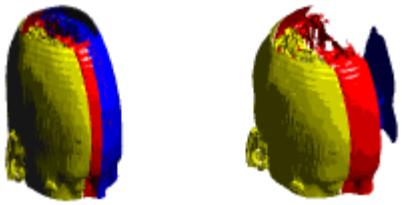


Figure 3. Subvolumes obtained by isosurface technique

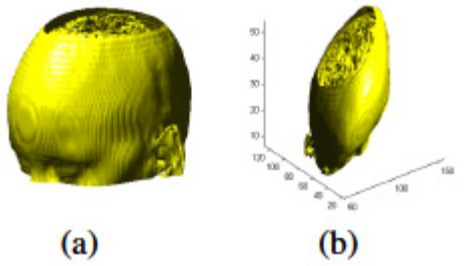


Figure 4. 3D volume and its transformed. (a) MRI volume. (b) Transformed volume

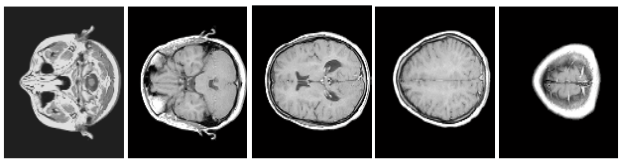


Figure 5. Five 2D slices of 3D volume

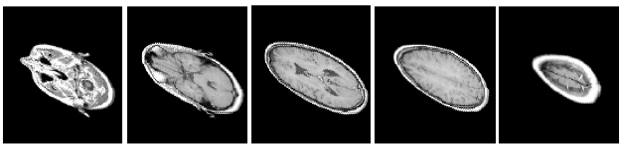
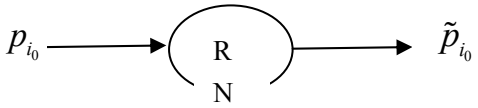


Figure 6. Five 2D slices of transformed 3D volume



p_{i_0} : Represents the vector of input data

$\tilde{p}_{i_0}$: Represents the vector of output data

Figure 7. Parameters extraction α_0 and β_0

ISSN 1913-8989 E-ISSN 1913-8997