Applied Mathematical Sciences, Vol. 5, 2011, no. 33, 1595 - 1611

New Methods of Recognition of 3D Objects Based on Neural Network, Genetic Algorithms, Fuzzy Logic and Principal Component Analysis

M. Elhachloufi[1], A. El Oirrak[1], D. Aboutajdine[2] and M. N. Kaddioui[1]

[1]Faculte des sciences Semlalia
Dept. Informatique, Marrakech, Maroc

[2]Senior Member IEEE
Faculte des sciences, LEESA-GSCM
BP 1014, Rabat, Maroc

elhachloufi@yahoo.fr, oirrek@yahoo.fr
aboutajadine@fsr.ac.ma, kaddioui@ucam.ac.ma

Abstract

The increasing number of objects 3D available on the Internet or in specialized databases require the establishment of methods to develop description and recognition techniques [1] [2] [3]to access intelligently to the contents of those objects. In this context, our work whose objective is to present the methods of description and recognition of 3D objects based on a set of techniques that are part of the artificial intelligence as neural networks, fuzzy logic, genetic algorithms and the principal component analysis. In fact, it consists of determining invariant descriptors [4][5] and recognizing the objects of a database similar to a given object (query object). The 3D objects of this database are transformations of 3D objects by one element of the overall transformation. The set of transformations considered in this work is the general affine group. The measure of similarity between two objects is achieved by a similarity function using the Euclidean distance.

Keywords: Affine Invariant, neural networks, fuzzy logic, genetic algorithms, Principal Component Analysis

1 Introduction

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. On the other hand, the size of 3D objects used on the Internet and in computer systems has become enormous, particularly due to the rapid advancement technology acquisition and storage which require the establishment of methods to develop description and recognition techniques to access intelligently to the contents of these objects. In fact, several approaches are used : in terms of statistical approaches, the statistical shape descriptors for recognition in general consist either of calculating various statistical moments [6] [7] and [8], or of estimating the distribution of the measurement of a given geometric primitive, when either deterministic or random. Among the approaches by statistical distribution, we mention the specter of 3D shape (SF3D) [9] which is invariant to geometric transformations and algebraic invariants [10], which provide global descriptors, which are expressed in terms of moments of different orders. For structural approaches, approaches representative of the object segmentation in 3D plot of land and performances by adjacency graph are presented in [11] and [12]. Similarly, Tangelder and al [13] have developed an approach based on representations by interest points. In transform approaches a very rich literature emphasizes any interest in approaches based transform Haugh [14], [15] and [16] which consists in detecting different varieties of dimension (n-1) immersed in the space. In the same vein, this work focuses on defining methods for the description and recognition of 3D objects using neural networks, genetic algorithms, fuzzy logic and the principal component analysis.

2 Representation of the 3D object

3D object is represented by a set of points denoted M=$(P_i)_{i=1,\ldots,n}$ where $P_i = (x_i, y_i, z_i) \in \mathbb{R}^3$, 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 procedure:

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

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

3 Recognition of 3D objects by neural networks

3.1 Neural networks

Neural networks [17] [18] are very robust tools, they are widely used in pattern recognition, classification and knowledge representation. In this network, neurons are arranged in layers. There is no connection between neurons of one layer, and connections are made only with layers of swallow neurons. Usually, each neuron of one layer is connected to all neurons of the next layer and to it only. In our network , weights are first initialized with random values. Then the network receives the input vector. The output of this network is the vector. The objective is to determine the weights and biases which are represented respectively by and which transforms into . For this, the signal is propagated forward in the layers of the neural network $X_k^{(n-1)} \mapsto X_j^{(n)}$, $Y_k^{(n-1)} \mapsto Y_j^{(n)}$. The forward spread is calculated using the activation function g , the aggregation function h (often a scalar product between the weights and the inputs of neuron) and synaptic weight w_{jk} between the neuron $X_k^{(n-1)}$ and the neuron $X_j^{(n)}$, where $X_k^{(n)} = g^{(n)}(h_j^{(n)}) = g^{(n)}(\Sigma_k w_{jk}^{(n)} X_k^{(n-1)})$. When the forward propagation is complete, we get the output result R. We calculate the error between the output given by the R and the output vector desired T for this sample. For each neuron i of output layer , we calculate : $e_i^{sortie} = g^{'}(h_i^{sortie}[T_i - R_i])$. We propagate the error backwards $e_k^{(n)} \mapsto e_j^{(n-1)}$ and we update the weights in all layers: $\triangle w_{ij}^{(n)} = \lambda e_i^{(n)} X_j^{(n-1)}$ where $\triangle$is the learning rate (low magnitude and less than 1.0). Finally we return the weights and biases (a_p and b_p)that transforms X to Y.

3.2 3D objects recognition

The principle of the proposed method is as follows:

`Step 1:`

Given two 3D objects X and Y, in the first time we take random sample points of X respectively points of Y named X_p respectively Y_p, with p is the size of the sample and n is the maximum size of the points X and Y. Then we pass to study the connection between X_p and Y_p, for this we extract the parameters α_p and β_p which can transmit X_p to Y_p as follows: $Y_p = \alpha_p \cdot X_p + \beta_p$ using neural networks as shown in following figure:

`Step 2:`

In this step, at first we calculate the points of 3D object X_{n-p}^c obtained by the following formula:$Y_{n-p}^c = \alpha_p \cdot X_{n-p} + \beta_p$ where X_{n-p} is the set of points of 3D

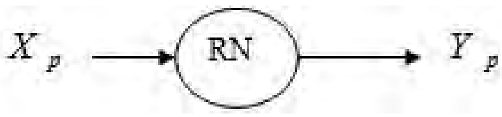

Figure 1: Architecture of Neural network

object remaining after the draw without replacement of the random sample. Then we compare Y^c_{n-p} to Y_{n-p} where Y_{n-p} is the set of points of the 3D object remaining after the given drawing in the random sample Y_p, using the Euclidean distance metric defined as follows: $D = d(Y_{n-p}, Y^c_{n-p}) = \Sigma_i (Y_{n-p}(i) - Y^c_{n-p}(i))^2 = \sum_i d_i^2$: (3) with $d_i = Y^c_{n-p}(i) - Y_{n-p}(i)$. The recognition is done by measuring the similarity using the formula (3).

3.3 Results and evaluation

We consider two 3D objects X and Y related by an affine transformation. According to the equation $Y_p = \alpha_p \cdot X_p + \beta_p$ obtained by using neural networks a is an affine transformation of X_p. Knowing $Y = (Y_p, Y_{n-p})$ that so if Y is a transformation affine of $X = (X_p, X_{n-p})$ it must be $Y_{n-p} = \alpha_p \cdot X_{n-p} + \beta_p$, because all points of X are transformed to Y by the same parameters. So This means that $Y_{n-p} \simeq Y^c_{n-p}$. Thus according to the formula (3) we will have $d_i \simeq 0 \ \forall i$, as shown in figure 3.

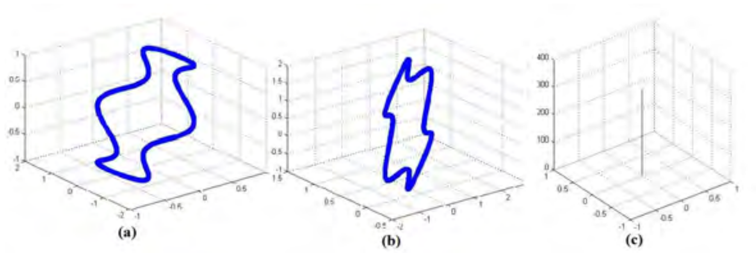

Figure 2: (a): Original 3D Objet , (b): Transformed 3D Objet , (c): Representation of errors d_i

4 An affine description invariant of 3D objects by the canonical correlation coefficients

4.1 The canonical analysis of two vectors

Let $X = (X_1, X_2, ..., X_n)^t$ and $Y = (Y_1, Y_2, ..., Y_n)^t$ are two vectors. The principle of this analysis is to search at the first a couple of variables (V^1, W^1)

where $V^1 = \alpha_1 X^j$ is a normalized linear combination of variables X^j (X^j is jth column of X and $W^1 = \beta_1 Y^k$ a normalized linear combination of variables Y^k(Y^k is kth column of Y, such that V^1 and W^1 are correlated as possible, i.e: that maximizes the quantity vailable $|\rho_1 = Corr(V^1, W^1)|$. Then we search the normed pair (V_2, W_2) where $V^2 = \alpha_2 X^j$ as being a linear combination of X^j uncorrelated to V^1, i.e:$\rho_{12}^v = Corr(V^1, V^2) = 0$, and $W^2 = \beta_2 Y^2$ linear uncorrelated to W^1 i.e: $\rho_{12}^w = Corr(W^1, W^2) = 0$, which V^2 and W^2 are correlated as possible, i.e: maximizes the quantity available $|\rho_2 = Corr(V^2, W^2)|$. And so on ... The canonical analysis product p of pairs of variables where s = 1, .., p. The variables V^s is an orthogonal basis of the space generated by X^j. The variables W^j is an orthogonal space generated by Y^k. The couples (V^s, W^s), particularly the first of them, reflect the linear connections between two groups of initial variables. The variables are called canonical variables. Their successive correlations $|\rho_k = Corr(V^k, W^k)|$(decreasing) are called canonical correlation coefficients (or canonical correlations). The canonical correlation coefficients $(\rho_s)_{s=1,...,p}$ are invariant quantities over an affine transformation and are all equal to 1 in this case.

4.2 Results and evaluation

We consider two 3D objects X and Y related by an affine transformation (figure 2). The calculation of canonical correlation coefficients from these objects requires computing the first pairs of canonical variables $(V^s, W^s)_{s=1,...,p}$ of X and Y then the coefficients of canonical correlations $(\rho_s)_{s=1,...,p}$ from these couples. Figure 3.a shows that the values of canonical correlation coefficients are all equal to 1. In another, Figures 3.b shows that the canonical variables of origin objects and its transformation are the same which leads us to conclude that Y is an affine transformation of X according to the procedure of canonical analysis.

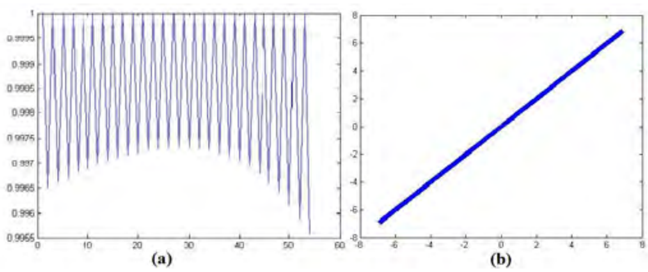

Figure 3: (a):Representation of the canonical correlation coefficients ,(b):Representation of the canonical variables of 3D object original as function of the 3D object transformed

5 Affine Invariant description of 3D objects with the coefficients of barycenter, genetic algorithms and neural Networks

5.1 Genetic algorithms

Genetic algorithms (GA) [19][20]were originally developed by John Holland. They are used in order to find a solution, usually numerical solving of a given problem, without having a prior knowledge on the search space. The basic structure of the genetic algorithm is as follows:

- Creation of initial population
- Evaluation of adaptive function

While population numbers are less than maximum number of generations, do:

- Generating new population
- Selection of best individuals
- Generation progeny by crossing
- Mutation of individuals

These algorithms are generally computationally implemented using genetic operation such as: *Creating the initial population:* the production of initial population
Coding: Consists of associates to each point of the state of space a data structure called chromosome
Selection: choose pairs of individuals surviving from one generation to another and those involved in the reproduction procedure of the future population
Crossing: applies to two individuals randomly in the previous population. These individuals are matched to give birth to two offsprings
Mutation: change randomly the value of the individual parameter
Operation parameters:
The implementation of a genetic algorithm requires the adjustment of certain parameters : population size, survival rate, the mutation rate and number of generations as in any iterative algorithm, we must define a stopping criterion, a maximum number of iterations or detection of an optimum.
Learning and Simulation
Learning by genetic algorithm involves the evaluation of the behavior of each individual before generating new individuals who will in turn be evolved. In our case the individuals are the coefficients of barycenter.

5.2 Definition and properties of barycenter of 3D object

Definition of barycenter of a 3D object

The barycenter of a 3D object is defined as follows:

$$\begin{cases} f : \mathbb{R}^{3n} \to \mathbb{R}^3 \\ \tilde{p_1}^{(X)}, \tilde{p_2}^{(X)}, ..., \tilde{p_n}^{(X)} \to \frac{\sum_i a_i \tilde{p_i}^{(X)}}{\sum_i a_i} \end{cases}$$

where $\tilde{p_i}^{(X)} = (x_i, y_i, z_i)$, $\tilde{g_X} = (x_G, y_G, z_G)$ and $\sum_i a_i \neq 0$. We note $CoefBary(X) = (a_1, a_2, ..., a_n)$ barycenter coefficients of $\tilde{g_X}$

Properties of barycenter of Y

i) The barycenter of Y (corresponds to the 3D object) is defined as $\tilde{g_Y} = \frac{\sum_i a_i \tilde{p_i}^{(Y)}}{\sum_i a_i}$
ii) $\tilde{g_Y} = A \times \tilde{g_X} + B$ where $\tilde{g_X}$ is the barycenter of Y

Note:$CoefBary(X) = CoefBary(Y) = (a_1, a_2, ..., a_n)$

5.3 Procedure for determining an affine invariant descriptive for 3D object

Given two objects 3D X and Y, for verifying that Y is an affine transformation of X, it suffices to verify that the barycenter coefficients of X and Y are equal, i.e: $CoefBary(X) = CoefBary(Y) = (a_1, a_2, ..., a_n)$

5.4 Calculation of the barycenter of X corresponds to the 3D object

According to the previous definition of barycenter we have :

$\tilde{g_X} = (x_G, y_G, z_G) = \frac{\sum_i a_i \tilde{p_i}^{(X)}}{\sum_i a_i} = \frac{\sum_i a_i (x_i, y_i, z_i)}{\sum_i a_i} \Rightarrow x_G = \frac{\sum_i a_i x_i}{\sum_i a_i}$, $y_G = \frac{\sum_i a_i y_i}{\sum_i a_i}$ et $z_G = \frac{\sum_i a_i z_i}{\sum_i a_i} \Rightarrow \chi_G = x_G - y_G - z_G = \frac{\sum_i a_i (x_i - y_i - z_i)}{\sum_i a_i} = \frac{\sum_i a_i \psi_i}{\sum_i a_i}$ where $\psi_i = x_i - y_i - z_i$ and $\tilde{z}_G = (\sum_i a_i) \chi_G$. Then the problem amounts to determining the coefficients $(a_1, a_2, ..., a_n)$ verifying the equation (1) where:

- ψ_i known
- $\tilde{z}_G$ unknown

Under our approach, we choose to use genetic algorithms to determine the barycenter coefficients of $\tilde{g}_G$.

5.5 Problem formulation

According to the previous definition of barycenter we have : Our problem is as follows: let $h(a_i, a_i, ..., a_i) = \sum_i a_i \psi_i$ .To determine the coefficients of barycenter$(a_i, a_i, ..., a_i)$, we search to minimize a function $h(a_i, a_i, ..., a_i)$ using genetic algorithms under the following constraint : $\sum_i a_i \neq 0$.
The result obtained by the genetic algorithm permit to determine the following coefficients $(a_i^0, a_i^0, ..., a_i^0)$ and $h(a_i^0, a_i^0, ..., a_i^0)$ which $\tilde{g_X} = (x_G, y_G, z_G) = (\frac{\sum_i a_i^0 x_i}{\sum_i a_i^0}, \frac{\sum_i a_i^0 y_i}{\sum_i a_i^0}, \frac{\sum_i a_i^0 z_i}{\sum_i a_i^0})$: (3) led to calculate the barycenter.
The next step is to calculate the barycenter of Y using the equation: $\tilde{g}_Y = \alpha \tilde{g}_X + \beta$: (4) , then we compare it with the barycenter obtained by equation (3) . If the difference between the two barycenters $\tilde{g}_X$ and $\tilde{g}_X^c$ is almost equal (as a threshold prefix) or nearly zero then we conclude that the coefficients $(a_i^0, a_i^0, ..., a_i^0)$ of barycenter corresponds to X are those of barycenter of Y , i.e : $CoefBary(X) = CoefBary(Y) = (a_1^0, a_2^0, ..., a_n^0)$, consequently Y is an affine transformation of X .

5.6 Results and evaluation

Consider two 3D objects related by an affine transformation (figure 2). To calculate the vector of invariants from one of these objects we calculate at first the coefficients of barycenter from one of two objects using genetic algorithms, and secondly we extract the parameters α and β which can transform X and Y using neural network.
Finally we compare the barycenter obtained by using coefficients barycenters previously calculated in equation 3 and that obtained by using the parameters α and β extracted by the neural network as shown by equation 4.
The figure 4 show that the two coefficients barycenters are equal. This leads us to conclude that the coefficients of a barycenter are invariant vectors for the two 3D objects.

6 3D object recognition by neural networks and normalized principal component analysis

6.1 normalized Principal component analysis

The principal component analysis normalized is a method factorial analysis of multidimensional data [21] [22]. It determines a decomposition of a random vector X with uncorrelated components, orthogonal and adjusting to better

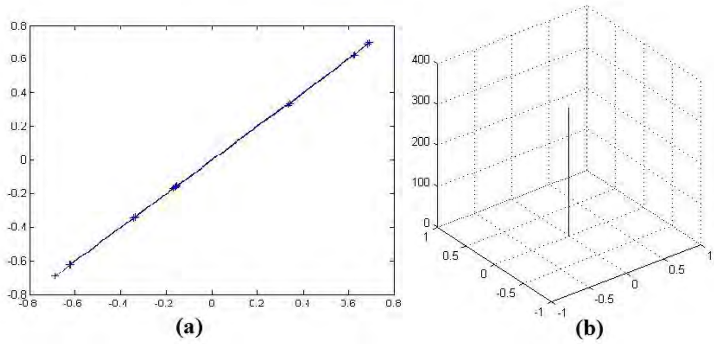

Figure 4: (a): barycenter of the original 3D object and its transformed , (b):L'erreur $\tilde{g_Y^c} - \tilde{g_Y}$

distribution of X. In this sense the components are called principal components and are arranged in descending order according to their degree of adjustment. The calculation of normalized principal components of the vector is carried out initially by calculating the covariance matrix as follows:$V = \frac{1}{n}X^tX$ where X^t transposed of X . Then we passes to extract eigenvalues and eigenvectors associated to V by the following process

1. $Det(V - \lambda I) = 0 \Rightarrow \lambda = [\lambda_1, \lambda_2, ..., \lambda_p]$
2. $X.u = \lambda.u \Rightarrow u = [u_1, u_2, ..., u_p]$

λ and u are eigenvalues and eigenvectors associated V . The eigenvalues τ and eigenvectors ν associated V^t are λ and $\nu = [\nu_1, \nu_2, ..., \nu_n]$ where $\nu_i = \frac{X.u_i}{\sqrt{\lambda_i}}$ and is the number of line of . The reconstruction of X from vector $\psi = (\lambda, u, \nu)$ is doing as follows $X = \Sigma_i \sqrt{\lambda_i} u_i \nu_i^t$, we say that the vector $\psi = (\lambda, u, \nu)$ is a vector characteristic of X , then we write $:X \rightarrow \psi = (\lambda, u, \nu)$.

Remark:

Let $Y = \alpha.X + \beta$, knowing that $var(F_a) = \lambda_a$ where F_a means a^{th} component factor defined as follows:$F_a = X.u_a$, then $\lambda_a^y = (u_a^y)^t var(\alpha X + \beta)(u_a^y)$ $\Rightarrow \lambda_a^y = (u_a^y)^t(\alpha^t)var(X)(\alpha)(u_a^y) = \lambda_a^y = (u_a^y\alpha)^t var(X)(\alpha u_a^y)$. If $var(X) = 1$ then $\lambda_a^y = (u_a^y\alpha)^t(\alpha u_a^y) = (u_a^y)^t\alpha^t\alpha(u_a^y)$.

7 Principle of the proposed recognition method

`Step 1:`

Given two 3D objects (object of database) and (query object) that we search if they are related by an affine transformation. For this we extract the parameters α and β which can transmit X to Y as follows:$Y = \alpha X + \beta$ using neural Networks.

Note: X represents the vector input of points and Y represents the vector output of points

```
Step 2:
```

In this step we use the parameter α extracted by the neural network in equation (2) from the principal component analysis to calculate the following error: $err = \lambda^y - (u^y)^t \alpha^t \alpha (u^y)$ then we select the object corresponding to the error below a given threshold s (usually very close to zero). The following diagram illustrates this situation.

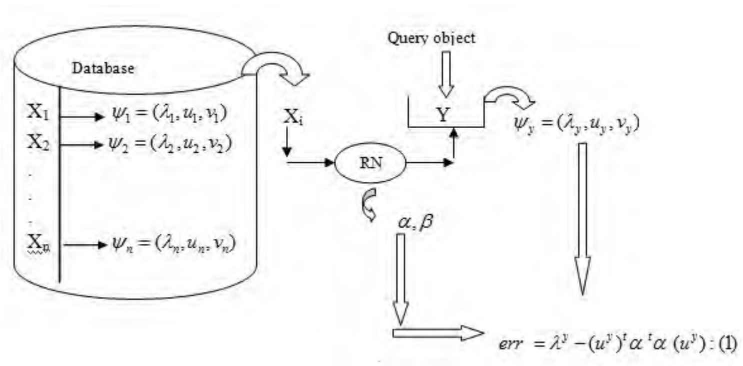

Figure 5: system of recognition

X_i (resp. Y_i) corresponds to centered points reduced of the database object i (resp. the query object), i.e:$var(X_i) = 1$ and $var(Y) = 1$, and $\psi_x = (\lambda_x, u_x, \nu_x)$ (resp. $\psi_y = (\lambda_y, u_y, \nu_y)$) is the characteristic vector associated to Xi (resp. query object Y) and α is the parameter of neural networks. The recognition is done by measuring the error described in equation (3) , if the error is below a given threshold (usually very close to zero), we say that the object Y is an affine transformation of X_i , object of the database .

8 Results and evaluation

Consider two 3D objects (object of a database) and (query object) related by an affine transformation (figure 2). After extracting a parameter α by neural networks as shown in figure 1 , we uses the latter in equation (1) to calculate the error of equation (3). The results show the performance of the proposed method. In fact, according to the graph of the error we conclude that $err_i \simeq 0$. So Y is an affine transformation of X

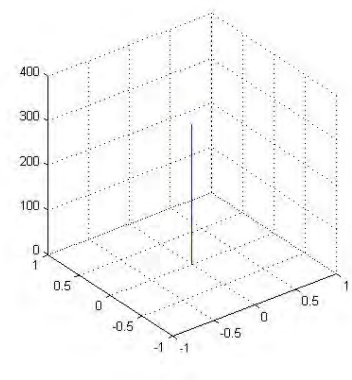

Figure 6: Representation of erreur err_i

9 Recognition of 3D object by neural networks, genetic algorithms and fuzzy logic

9.1 Fuzzy Logic

Definition of Fuzzy Logic

A fuzzy model of a system [23] [24]is a representation of his behavior by the concepts of the theory of fuzzy subsets. This representation characterizes the relationship between input variables and output system. In the classical or Boolean logic, any element of a set to only one value 1 or 0, or in fuzzy logic, a set A of universe of discourse U = (x) is defined as a distribution by which each value of x is assigned any number in the interval [0, 1], indicating the degree of membership of x in set A, that is to say .

Fuzzy Model

In general, any system modeling (linear or nonlinear) can be decomposed three inter-related elements:

- *Input* (input variable)
- *Model* (mathematical formulation)
- *Output* (output variable)

In case of non-linearity or the lack of a mathematical model describing a system, fuzzy logic can be an alternative to system, but we must have basic information on our system.

Fuzzy Model Elements

The Fuzzy Model Elements are:

- *Fuzzification:* the definition of linguistic variables and membership functions

- *Inference:* the establishment of rules of inference in the form: IF ... THEN
- *Deffuzzification:* the determination of the output variable in the calculation center gravity, or by the maximum ... etc.

9.2 Definition and properties of a barycenter of 3D objects

The barycenter of a 3D object is defined as follows:

$$\begin{cases} f : \mathbb{R}^{3n} \rightarrow \mathbb{R}^3 \\ \tilde{p_1}^{(X)}, \tilde{p_2}^{(X)}, ..., \tilde{p_n}^{(X)} \rightarrow \frac{\sum_i a_i \tilde{p_i}^{(X)}}{\sum_i a_i} \end{cases}$$

where $p_i^{(X)} = (x_i, y_i, z_i)$ and a_i the a_i are the coefficients of barycenter. Let $g_{X,Y}$ a barycenter of $(p^{(X)}, a)$ and $(p^Y, \tilde{a})$ then $g_{X,Y} = (\frac{a}{a+\tilde{a}} p^{(X)} + \frac{\tilde{a}}{a+\tilde{a}} p^{(Y)})^2$
Knowing that $(p^{(Y)} p^{(X)})^2 = (p^{(X)})^2 + (p^{(X)})^2 - 2\ p^{(X)} p^{(Y)} cos\theta$ then
$(a + \tilde{a})^2 g_X = a^2 (p^{(X)} p^{(Y)})^2\ - a^2 (p^{(Y)})^2\ 2(a\tilde{a} + 1) p^{(X)} p^{(Y)} cos\theta \tilde{a}^2 (p^{(X)})^2$.

Remark 1:

we suppose that $p^Y = \alpha \cdot p^X + \beta$ where α and β are scalars, this means that all points of y is an affine transformation of points X , then y is an affine transformation of X .

Let $p^Y = \alpha \cdot p^X + \beta$ then :
$g_{X,Y} = \frac{a^2}{(a+\tilde{a})^2} (p^{(Y)} p^{(X)})^2 + \frac{a^2 - \tilde{a}^2}{(a+\tilde{a})^2} (p^{(Y)})^2 + \frac{2(a \cdot \tilde{a}+1)}{(a+\tilde{a})^2} p^{(X)} (\alpha \cdot p^{(X)} + \beta) cos\theta \Rightarrow g_{X,Y} = \frac{a^2}{(a+\tilde{a})^2} (p^{(Y)} p^{(X)})^2 + \frac{a^2 - \tilde{a}^2}{(a+\tilde{a})^2} (p^{(Y)})^2 + \frac{2(a \cdot \tilde{a}+1) \cdot \alpha}{(a+\tilde{a})^2} (p^{(X)})^2 cos\theta + \frac{2(a \cdot \tilde{a}+1)}{(a+\tilde{a})^2} p^{(X)} \cdot \beta cos\theta$

We set $\lambda = \frac{a^2}{(a+\tilde{a})^2}$, $\phi = \frac{\tilde{a}^2 - a^2}{\tilde{a}^2 - a^2}$,$d = (p^{(X)} p^{(X)})^2$, $\gamma = \frac{2\alpha(1+a\tilde{a})}{(a+\tilde{a})^2}$, $\delta = \frac{2\beta(1+a\tilde{a})}{(a+\tilde{a})^2}$ and $\tau = \frac{(p^{(X)})^2 + (p^{(Y)})^2 - d}{2 p^{(X)} p^{(Y)}}$ Then : $g_{X,Y} = \lambda d + \phi (p^{(X)})^2 \tau + \delta p^{(X)}) \tau$: (1)

9.3 Principle of the proposed recognition method

`Step 1:`

In this step we calculate the barycenter g_i and the vector $\nu_i = (a_i, \tilde{a}_i)$of barycenter coefficients of points $(p_i^{(X)}, p_i^{(Y)})$,$\forall i \in [1, 2, ..., r]$ where r is the number of object point X. According to the definition of previous barycenter we have : $g_i = (x_{g_i}, y_{g_i}, z_{g_i}) = \frac{a_i p^{(X)} + \tilde{a}_i p^{(Y)}}{a+\tilde{a}} = \frac{a_i (x_i, y_i, z_i) + \tilde{a}_i (\tilde{x}_i, \tilde{y}_i, \tilde{z}_i)}{a+\tilde{a}} \Rightarrow, x_{g_i} = \frac{a_i x_i + \tilde{a}_i \tilde{x}_i}{a_i + \tilde{a}\tilde{i}}$, $y_{g_i} = \frac{a_i y_i + \tilde{a}_i \tilde{y}_i}{a_i + \tilde{a}\tilde{i}}$ and $z_{g_i} = \frac{a_i z_i + \tilde{a}_i \tilde{z}_i}{a_i + \tilde{a}\tilde{i}} \Rightarrow \chi_i = x_{g_i} - y_{g_i} - z_{g_i}\ \frac{a_i (x_i - y_i - z_i) + \tilde{a}_i (\tilde{x}_i + \tilde{y}_i + \tilde{z}_i)}{a_i + \tilde{a}_i}$.

$(a_i + \tilde{a}_i) \chi_i = a_i \gamma_i + \tilde{a}_i \tilde{\gamma}_i$ where $\gamma_i = x_i - y_i - z_i$ and $\tilde{\gamma}_i = \tilde{x}_i - \tilde{y}_i - \tilde{z}_i$ if we set $\delta_i = (a_i + \tilde{a}_i) \chi_i$ then $\delta_i = a_i \gamma_i + \tilde{a}_i \tilde{\gamma}_i$: (2)

Then the problem returns to determine the coefficients $(a_i, \tilde{a}_i)$ satisfying the equation (1) where:

- γ_i and $\tilde{\gamma}_i$ are known
- δ_i unkown

Under our approach, we choose to use genetic algorithms to determine the coefficients $(a_i, \tilde{a}_i)$ and g_i minimizing the following quantity $g_i = h(a_i, \tilde{a}_i) = a_i\gamma_i + \tilde{a}_i\tilde{\gamma}_i$: (3) under the following constraint :$a_i + \tilde{a}_i \neq 0$. The result obtained by the genetic algorithm determine the coefficients $(a_i^0, \tilde{a}_i^0)$ which led us to calculate the barycenter $g_i = (x_{g_i}, y_{g_i}, z_{g_i})$.

`Step 2:`

After calculating the vector $(a_i^0, \tilde{a}_i^0)$, g_i and extracting the parameters α and β by neural networks for transformation of $p_i^{(X)}$ to $p_i^{(Y)}$ as indicated by the figure 1 , we passes to calculate the following errors: $err_i = g_i - \lambda_i d_i - \phi_i (p_i^{(X)})^2 \tau_i - \delta_i p_i^{(X)} \tau_i$: (4) and variations of the error $\Delta err_i = err_i - err_{i-1}$: (5) corresponding to couples $(p_i^{(X)}, p_i^{(Y)})$ $\forall i \in [1, 2, ..., r]$. Finally, the recognition is done by the outputs S_{err_i} associated to errors and errors variation Δerr_i calculated from different pairs of samples $(p_i^{(X)}, p_i^{(Y)})$ obtained from the fuzzy logic. While these outputs are below a given threshold (very close to zero) s , then we concluded that the equation (4) is verified and therefore the object of the database X is a transformation of the object query Y According to note 1.

Operation of fuzzification The input quantities in our case to be fuzzified are error (denoted err_i), the variation of the error (denoted Δerr_i) and output (denoted S_{err_i}). The input err_i has been partitioned into 4 linguistic values as well as Δerr_i and the output S_{err_i} $\forall i \in [1, 2, ..., r]$ as follows : $VS = [0.01, 0.05]$:Very Small,$S = [0.05, 0.1]$:Small $A = [0.5, 0.1]$:Average and $B = [0.1, 1]$:Big

Rule-based "inference"

The basis of fuzzy rules defined many rules. These rules were developed manually. We adopted the intuitive approach to build the knowledge base
- R1:IF (err_i) is VS and (Δerr_i) is VS then(S_{err_i}) is VS
- R2:IF (err_i) is VS and (Δerr_i) is S then(S_{err_i}) is S
- R3:IF (err_i) is S and (Δerr_i) is S then(S_{err_i}) is S
- R4:IF (err_i) is S and (Δerr_i) is A then(S_{err_i}) is A
- R5:IF (err_i) is A and (Δerr_i) is A then(S_{err_i}) is A
- R6:IF (err_i) is A and (Δerr_i) is B then(S_{err_i}) is B
- R7:IF (err_i) is B and (Δerr_i) is B then(S_{err_i}) is B

Defuzzification"

For output, we use the more often the rule "the center of mass" i.e: we calculate the barycenter of the surface (the intersection between the membership function of linguistic value of the relevant output and ? found by the aggregation rules) .

10 Results and evaluation

Consider two 3D objects X (object of a database) and Y (query object) related by an affine transformation (figure 2). After extraction the parameter α and β_i by neural networks as shown in figure 1 for $i = 1, 2, ..., r$ which r denotes the number of sample drawn randomly from objects, we using them in equation (3) to calculate the error of equation (4) and equation (5). The results show the performance of the proposed method. In fact, the outputs corresponding to inputs: error and variation of errors of r samples of the two objects are all very close to zero as it is shown in figure 8 which shows the recognition of the object in question (the input object).

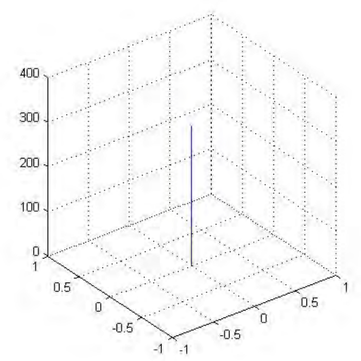

Figure 7: Representation of S_{err_i} of 3D objet

11 Conclusion

In this work we present methods for the description and recognition of 3D objects based on several approaches: The first approach is based on canonical correlation coefficients obtained from the canonical analysis , this approach consists of extracting the canonical variables from the original object and its transformation, then we calculate their correlations, those are called canonical correlation coefficients, which are invariant quantities over an affine transformation. The second approach relies on the extraction of barycenter coefficients from the 3D object using genetic algorithms and neural networks, these coefficients are invariant over an affine transformation of the object. Finally, a third approach which aims to recognize the object (s) of database (s) similar to a given object (query object), using :

- Scores obtained from fuzzy logic as shown in recognition procedure previously
- A feature vector which satisfies an equation obtained from the principal component analysis and neural networks

References

[1] A. Eloirrak, M. Daoudi, and D. Aboutajdine. "affine invariant descriptors using fourier series". *Pattern Recognition Letters*, 23(10):1109–1118, 2002.

[2] A. Eloirrak, M. Daoudi, and D. Aboutajdine. "affine invariant descriptors for color images using fourier series". *Pattern Recognition Letters*, 24(9):1339–1348, 2003.

[3] A. Eloirrak, M. Daoudi, and D. Aboutajdine. Estimation of general 2d affine motion using fourier descriptors. *Pattern Recognition*, 35(1):223–228, 2002.

[4] M. Petrou and A. Kadyrov. Affine invariant features from the trace transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):30–44, 2004.

[5] A. Choksuriwong, H. Laurent, and B. Emile. A comparative study of objects invariant descriptors. In *ORASIS*, 2005.

[6] T. Murao. Descriptors of polyhedral data for 313-shape similarity search. In *Proposal P177, MPEG-7 Proposal Evaluation Meeting, UK*, 1999.

[7] M. Elad, A. Tal, and S. Ar. Directed search in a 3d objects database using svm. In *Hewlett-Packard Research Report HPL-2000-20R1*, 2000.

[8] C. Zhang and T. Chen. Efficient feature extraction for 2d/3d objects in mesh representation. In *Proc. of the International Conference on Image Processing (ICIP 2001)*, 2001.

[9] T. Zaharia and F. Prteux. 3d-shape-based retrieval within the mpeg-7 framework. In *Proc. SPIE Conf. on Nonlinear Image Processing and Pattern Analysis XII*, volume 4304, pages 133–145, 2001.

[10] Taubin and D.B. Cooper. *Object recognition based on moment (or algebraic) invariants.* J.L. Mundy and A. Zisserman, editors, Geometric Invariants in Computer Vision, MIT Press, 1992.

[11] 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 Pattern Analysis and Machine Intelligence*, 19(3):259–267, 1997.

[12] S. Dickinson, A. Pentland, and S. Stevenson. Viewpoint-invariant indexing for content based image retrieval. *Proc. IEEE Int. Workshop on Content-Based Access of Image and Video Database*, pages 20–30, 1998.

[13] 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.

[14] P. V. C. Hough. *Method and means for recognizing complex patterns.* U.S. Patent 3 069 654, 1962.

[15] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[16] J. Illingworth and J. Kittler. A survey of the hough transform. *Computer Vision, Graphics and Image Processing*, 44:87–116, 1988.

[17] A. Benyettou, A. Mesbahi, H. Abdoune, and A. Ait-ouali. La reconnaissance de formes spatio-temporelles par les rseaux de neurones dlais temporels. In *Conf. Nationale sur l'Ingnierie de l'Electronique - CNIE'02,Algrie, univ. USTOran*, pages 159–163, 2002.

[18] Dreyfus and al. *Rseaux de neurones, Mthodologie et Applications.* Eyrolles, 2002.

[19] Ludovic. M. *Audit de scurit par algorithmes gntiques.* Thse de Doctorat,Universit de Rennes 1, France, 1994.

[20] E. Lutton. *Algorithmes gntiques et Fractales.* Dossier d'habilitation diriger des recherches, Universit Paris XI Orsay, 1999.

[21] Bouroche J.M. and G. Saporta. *L'analyse de donnes.* Presses Universitaires de France, Collection Que sais-je ?, 2006.

[22] G. Saporta. *Probabilits, analyse des donnes et statistiques.* Technip, 2me dition, 2006.

[23] L.A.Zadeh. Fuzzy logic, neural networks, and soft computing. In *ACM'94*, pages 77–84, 1994.

[24] A.Idri and A.Abran. *La Logique Floue applique aux modles d'estimation du cout in Sminaire dpartemental*. Universit du Quebec Montreal, 1999.

Received: May, 2010