AFFINE INVARIANT DESCRIPTOR AND RECOGNITION OF 3D OBJECTS USING NEURAL NETWORKS AND PRINCIPAL COMPONENT ANALYSIS

[1]**M. ELHACHLOUFI** , [1]A**. EL OIRRAK,** [2]**D. ABOUTAJDINE ,** [1]**M.N. KADDIOUI**
[1]Faculté des sciences Semlalia, Dept Informatique Marrakech, Maroc
[2]Faculté des sciences, LEESA-GSCM, BP 1014, Rabat , Maroc

ABSTRACT

The increasing number of objects 3D are available on the Internet or in specialized databases and require the establishment of methods to develop description and recognition techniques[1,2,3] to access intelligently to the contents of these objects . In this context, our work whose objective is to present the methods of description and recognizing of 3D objects are based on neural networks and 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) using neural networks, descriptor vectors extracted from the principal component analysis and concluded equations from the same analysis and neural networks. 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 descriptor vector objects is achieved by a similarity function using the Euclidean distance.

Keywords: *Invariant Descriptor , Recognition , 3D Objects, Neural Networks, Principal Component Analysis, Affine Transformation.*

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 affine invariant descriptor and recognition of 3D objects using neural networks, genetic algorithms, fuzzy logic and the principal component analysis.

2. METHOD 1 : RECOGNITION OF 3D OBJECTS BY NEURAL NETWORKS

2.1 Representation of the 3D object

3D object is represented by a set of points denoted $M = \left\{P_i\right\}_{i=1.......n}$ where

$P_i = (x_i, y_i, z_i) \in \Re^3$, 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 procedure:

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

$$Y(t) = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \times \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} + \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix}$$

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

2.2 Neural networks

Neural networks are very robust tools, they are widely used in pattern recognition, classification and knowledge representation [17]. In this network, neurons are arranged in layers. There is no connection between neurons of one layer, and connections 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 α_p and β_p which transforms X into Y . 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 $X_k^{(n)} = g^{(n)}(h_j^{(n)}) = g^{(n)}\left(\sum_k w_{jk}^{(n)} X_k^{(n-1)}\right)$.

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 : $e_i^{sortie} = g'\left(h_i^{sortie}\right)[T_i - R_i]$.

We propagate the error backwards $e_i^{(n)} \mapsto e_j^{(n-1)}$ through the following formula: $e_i^{(n-1)} = g'^{(n-1)}\left(h_i^{(n-1)}\right)\sum w_{ij} e_i^{(n)}$. We update the weights in all layers: $\Delta w_{ij}^{(n)} = \lambda e_i^{(n)} X_j^{(n-1)}$ where Δ is the learning rate (low magnitude and less than 1.0).

Finally we return the weights and biases in concluding that transforms X to Y designated by α_p and β_p .

2.3. Method of Recognition of 3D objects

The principle of the proposed method is as follows:

2.3.1 Step1

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 with $p << n$ where n is the maximum size of the points X and Y . After 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: $Y_p = \alpha_p \cdot X_p + \beta_p : (1)$ using neural networks as shown in following figure :

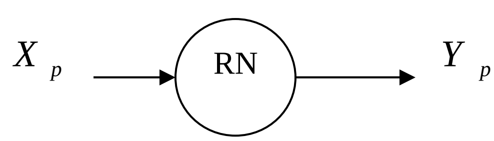

Figure 1: Extraction of parameters α_p and β_p

X_p represents the vector input of points

Y_p represents the vector output of points

2.3.2 Step 2

In this step , at first we calculate the points of 3D object Y_{n-p}^c obtained by the following formula: $Y_{n-p}^c = \alpha_p X_{n-p} + \beta_p : (2)$ where X_{n-p} is the set of points of 3D object remaining after the draw without replacement of the random sample. Then we compare Y_{n-p}^c to Y_{n-p} where

Y_{n-p} is the set of points of the 3D object Y remaining after the given drawing in the random sample Y_p that , using the euclidean distance metric defined as follows:

$$D = d(Y_{n-p}, Y_{n-p}^c) = \sum_i (Y_{n-p}(i) - Y_{n-p}^c(i))^2$$

$$= \sum_i d_i^2 : (3)$$

with $d_i = Y_{n-p}(i) - Y_{n-p}^c(i)$. The recognition is done by measuring the similarity between Y_{n-p} and Y_{n-p}^c using the formula (3) .

2.4 Results and evaluation

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

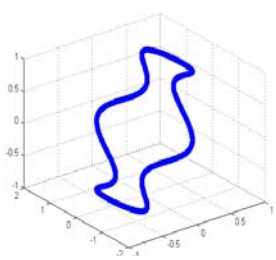

Figure 2 : 3D Objet

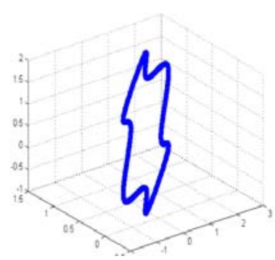

Figure 3: 3D Objet transformed

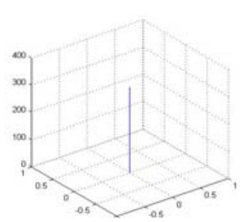

Figure 4: Representation of erreurs d_i

3. METHOD 2: INVARIANT DESCRIPTION AND RECOGNITION OF 3D OBJECTS BY PRINCIPAL COMPONENT ANALYSIS.

3.1 Representation of the 3D object

3D object is represented by a set of points denoted $\{P_i\}_{i=1.......n}$ with $P_i^t = (x_i, y_j, z_i)$, arranged in a matrix X ,
i.e: $X^t = (P_1, P_2, ..., P_n)$ where X^t and P^t are the matrix and vector transposed X and P.
Under the action of an affine transformation, the coordinates of X are transformed into other coordinates of Y by the following procedure:
$Y = \alpha X + \beta$ where α and β are scalar.
Let f function defined as follows

$: a_f = f(a) = \frac{a - \bar{a}}{\sigma_a} : (4)$, where a is 3D object , $\bar{a}$ and σ_a^2 are the mean and variance X .
We have

$$f(Y) = \frac{Y - \bar{Y}}{\sigma_Y} = \frac{\alpha X + \beta - (\alpha \bar{X} + \beta)}{\sqrt{\text{var}(\alpha X + \beta)}} = \frac{\alpha (X - \bar{X})}{\sqrt{\alpha^2 \text{var}(X)}}$$

$$= \frac{\alpha (X - \bar{X})}{\alpha \sigma_X} = \frac{X - \bar{X}}{\sigma_X} = f(X)$$

Then f is called an invariant function.

3.2. Principal component analysis

The principal component analysis is a method factorial analysis of multidimensional data [18]. It determines a decomposition of a random vector X with uncorrelated components, orthogonal and adjusting to better distribution of X [19]. 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:

$V = \frac{1}{n} X^t X$, 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- $X \cdot u = \lambda \cdot u \Rightarrow u = [u_1, u_2 .....u_p]$

λ and u are eigenvalues and eigenvectors associated V.
The eigenvalues τ and eigenvectors v associated V^t are λ and $v = (v_1, ..., v_n)$

where $v_i = \frac{X u_i}{\sqrt{\lambda_i}}$ and n is the number of line of X.

The reconstruction of X from vector $\psi = (\lambda, u, v)$ is doing as follows $X = \sum_i \sqrt{\lambda_i} \cdot u_i \cdot v_i^t$, we say that the vector $\psi = (\lambda, u, v)$ is a vector characteristic of X, so we write : $X \to \psi = (\lambda, u, v)$.

Remarks :

(a)- If $X = Y$ then $\psi_X = \psi_Y$.

(b)- We suppose $Y = \alpha X + \beta$ then $X_f = f(X) = f(Y) = Y_f$ and according to (a) we obtain: $\psi_{X_f} = \psi_{Y_f}$, thus $\psi_{(.)_f}$ is an invariant against affine transformation.

(c)- According to (4) we concluded that $X = X_f \cdot \sigma_X + \bar{X} : (5)$.

If $\psi_{X_f} = (\lambda, u, v)$ is a vector characteristic of X_f, i.e : $X_f = \sum_i \sqrt{\lambda_i} \cdot u_i \cdot v_i^t$ then $\nabla_{X_f} = (\psi_{X_f}, \bar{X}, \sigma_X)$ is a descriptor of X in the sense that the invariance check is only relative to the first component ψ_{X_f} and the reconstitution X require to use of $\psi_{X_f}, \bar{X}$ et σ_X as shown in equation (5).

(d) - Let $Y = a \cdot X + b$, knowing that $\mathrm{var}(F_\alpha) = \lambda_\alpha$ where F_α means la α^{th} component factor defined as follows: $F_\alpha = X \cdot u_\alpha$, then $\lambda_\alpha^y = (u_\alpha^y)^t \, \mathrm{var}(a \cdot X + b)(u_\alpha^y)$

$\Rightarrow \lambda_\alpha^y = (u_\alpha^y)^t a^t \, \mathrm{var}(X) a (u_\alpha^y)$

$\lambda_\alpha^y = (a u_\alpha^y)^t \, \mathrm{var}(X)(a u_\alpha^y)$.

If $\mathrm{var}(X) = 1$ then $\lambda_\alpha^y = (a u_\alpha^y)^t \cdot (a u_\alpha^y) \Rightarrow$

$\lambda_\alpha^y = (u_\alpha^y)^t \cdot a^t \cdot a \cdot (u_\alpha^y) : (6)$.

3.3 Principle of description and recognition proposed method

3.3.1 Step 1 : learning system

It can be decomposed into two phases: extraction of vector descriptor and recording.
The role of the first phase is to associate each object to learn the vector descriptor $\nabla_{X_f} = (\psi_{X_f}, \bar{X}, \sigma_X)$. This vector will be used later in the recognition system.
Registration vector descriptors are Phase 2.
The diagram of learning system is shown in Figure 1.

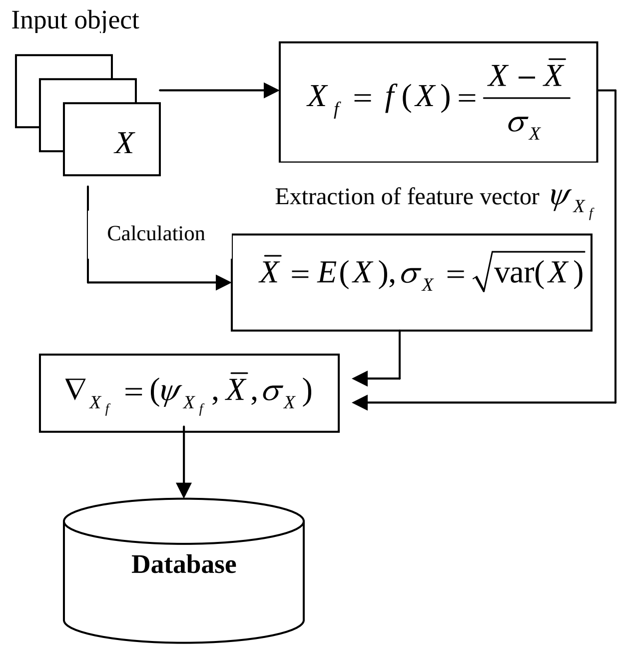

Figure 5:system diagram learning

3.3.2 Step 2 : system de reconnaissance

Recognition will be in determining, for each characteristic vector (first component) vector descriptor of the object database, the error resulting from the difference between this vector and that of the object (query object) to recognize.
We recognize the object that produced the error is almost zero.

This operation is illustrated in Figure 2.

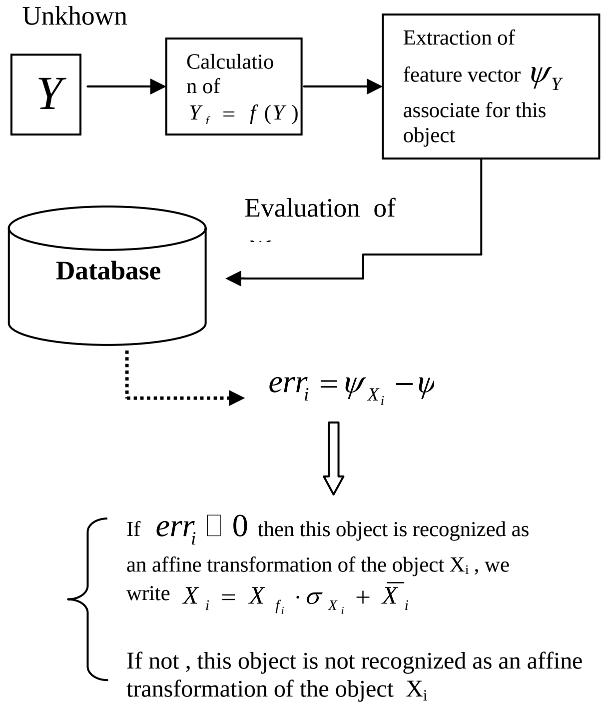

Figure 6 : Diagram of recogntion system

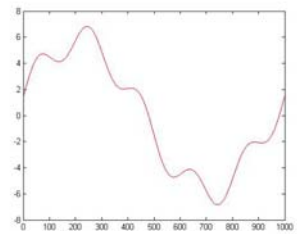

Figure 7 : descriptor of 3D objet origin

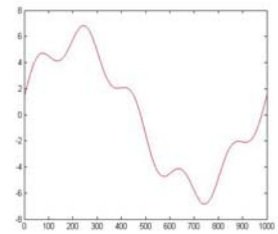

Figure 8 : descriptor of 3D objet transformed

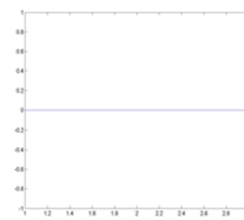

Figure 9: Representation of error $err_{x,y}$

3.4. Results and evaluation

Consider two 3D objects (object of a database) and (query object) related by an affine transformation as shown in figure 1 and figure 2. After extraction of vector descriptors $\nabla_{X_f} = (\psi_{X_f}, \bar{X}, \sigma_X)$ of X and $\nabla_{Y_f} = (\psi_{Y_f}, \bar{Y}, \sigma_Y))$ of Y we move to calculate the error $err_{x,y} = \psi_X - \psi_Y$.

According to the graph of the error (figure 9) we found that $err_{x,y} \approx 0$.

In addition, the graphical representation of the components of characteristic vectors ψ_{X_f} and ψ_{Y_f} (figure 7 and 8) illustrate that they are the same which verifies the invariant feature vector, which shows that Y is an affine transformation of X .

4. METHOD 3 : RECOGNIZING OF 3D OBJECT BY NEURAL NETWORKS AND PRINCIPAL COMPONENT ANALYSIS

4.1 Principle of the proposed method of recognition

4.1.1 Step 1

Given two 3D objects X (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:

$Y = \alpha \cdot X + \beta : (7)$ using neural Networks as shown in the following figure :

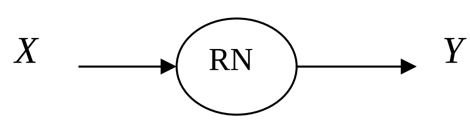

Figure 10: Extraction of parameters α and β

X represents the vector input of points

Y represents the vector output of points

4.1.2 Step 2

In this step we use the parameter α extracted by the neural network in equation (1) from the principal component analysis to calculate the following error:
$err = \lambda^y - (u^y)^t \alpha\, {}^t\alpha (u^y) : (8)$ obtained from equation (6) , 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.

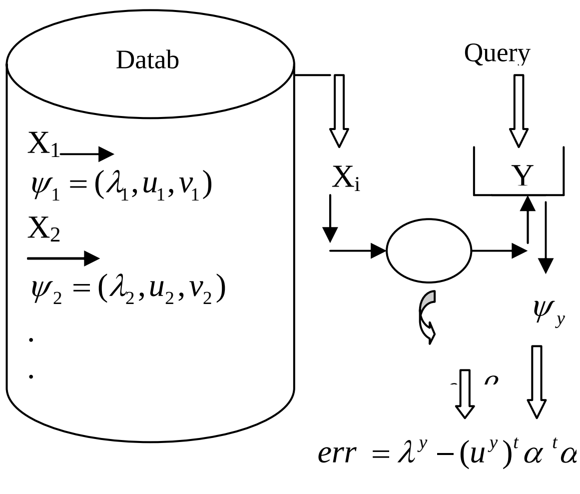

Figure 11 : system of

X_i (resp. Y) corresponds to centered points reduced of the database object i (resp. the query object), so $\mathrm{var}(X_i) = 1$ and $\mathrm{var}(Y) = 1$, and $\psi_i = (\lambda_i, u_i, v_i)$ (resp. $\psi_y = (\lambda_y, u_y, v_y)$) is the characteristic vector associated to X_i (resp. query object Y) and α is the parameter of neural networks.
The recognition is done by measuring the error described in equation (8) , if the error is below a given threshold (usually very close to zero), we say that the object Y is an affine transformation of one of the database X_i .

4.2 Results and evaluation

Consider two 3D objects (object of a database) and (query object) related by an affine transformation (figure 1 and figure 2). After extracting a parameter α by neural networks as shown in figure 1 , we use the latter in equation (7) to calculate the error of equation (8).
The results show the performance of the proposed method. In fact, according to the graph of the error we conclude that $err_i \approx 0$. So Y is an affine transformation of X_i

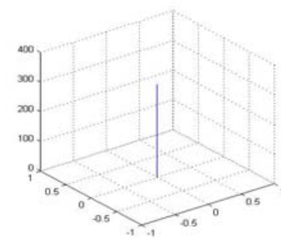

Figure 12: Representation of erreur err_i

5. COMPARISON OF RESULTS

According to figures below we can see that the error[1] - corresponds to the difference between the invariants of the object origin and its transformation by an affine transformation- calculated by our methods (figures 4,9 and 12) is smaller than those obtained by the methods of Fourier transform and Moments (figures 13 and 14).
In addition, the computing time of our method is less than the method of Fourier transform and Moments. This leads us to conclude that our method applied to these objects is more efficient than Fourier transform and Moments.

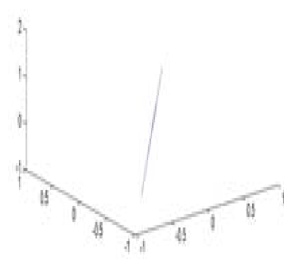

Figure 13: error[1] representation by Fourier

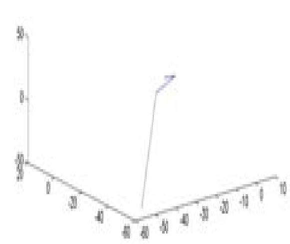

Figure 14 : error[1] representation by Moments

6. CONCLUSION

In this work we presented three methods for description and recognition 3D objects based on neural networks and principal component analysis, their aims are to recognize the object (s) of database (s) similar to a given object (query object).
The recognizing for the first method is done by measuring the similarity between the given 3D object and its transformed using the euclidean distance metric as described in the proposed method.
The second method concerns the description and recognition of 3D objects, the objective of this

method is to determine a vector descriptor extracted from this object using ACP .
The vector extracted is invariant over an affine transformation.
For the third method, it consist of the recognition of 3D object using a features vectors extracted from 3D object that satisfies an equation obtained from the principal component analysis and neural networks.
Experimental results show the validity of these methods.

REFERENCES

[1] Ahmed El Oirrak, Mohamed Daoudi, Driss Aboutajdine: Affine invariant descriptors for color images using Fourier series. Pattern Recognition Letters 24(9-10): 1339- 1348 (2003)

[2] Ahmed El Oirrak, Mohamed Daoudi, Driss Aboutajdine: Estimation of general 2D affine motion using Fourier descriptors. Pattern Recognition 35(1): 223-228 (2002)

[3] Ahmed El Oirrak, Mohamed Daoudi, Driss Aboutajdine: Affine invariant descriptors Using Fourier series. Pattern Recognition . Letters 23(10): 1109-1118 (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 & B. Emile. A Comparative Study of Objects Invariant Descriptor. ORASIS, 2005.

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

[7] M. Elad, A. Tal, S. Ar, "Directed search in a 3D objects database using SVM", Hewlett-Packard Research Report HPL-2000-20R1, 2000.

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

[9] T. Zaharia, F. Prêteux, "3D-shape-based retrieval within the MPEG-7 framework", Proc. SPIE Conf. on Nonlinear Image Processing and Pattern Analysis XII, Vol. 4304, pp. 133-145, San Jose, Etats-Unis, Janv. 2001.

[10] Taubin et D.B. Cooper, "Object recognition based on moment (or algebraic) invariants", J.L. Mundy and A. Zisserman, editors, Geometric Invariants in Computer Vision, pp. 375-397, MIT Press, 1992.

[11] S. J. Dickinson, D. Metaxas, A. Pentland, "The role of model-based segmentation in the recovery of volumetric parts from range data", IEEE Trans. on PAMI, Vol. 19, No. 3, pp. 259-267, Mars 1997.

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

[13] J.W.H. Tangelder, 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 3069 654, 1962.

[15] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," Pattern Recognition, Vol. 13, No. 2, pp. 111-122, 1981.

[16] J. Illingworth et J. Kittler, "A survey of the Hough transform", Computer Vision, Graphics and Image Processing, Vol. 44, pp. 87-116, Oct. 1988.

[17] R. LEPAGE, B. SOLAIMAN, Les réseaux de neurones artificiels et leurs applications en imagerie et en vision par ordinateur, Ecole de technologie supérieure, 2003.

[18] Saporta, G. (2006), Probabilités, analyse des données et statistiques, Techniques, 2ème édition, 622 p, 2006.