3D Objects Retrieval Using Geodesic Distance Based on Eikonal Equation

Driss Naji[1], M. Fakir*[2], O. Bencharef[3]
Faculty of Science and Technology, laboratory of information processing & decision support (TIAD), BeniMellal, Morocco. B.P : 523 . Ph: +212 666 936 396
*Corresponding author, e-mail: naji.drisss@gmail.com[1], fakfad@yahoo.fr[2], bencharef98@gmail.com[3]

Abstract

Recently, shape-based matching and retrieval of 3D polygonal models has become one of the most fundamental problems in computer vision. Dealing with families of objects instead of a single one may impose further challenges on regular geometric algorithms. In this paper we focus on the classification of 3D objects based on their geodesic distance & path calculated on a mesh using an iterative algorithm for solving the Eikonal equation. For the classification process, we use both Multiclass Support Vector Machine (M-SVM) classifier and K-Nearest Neighbors (KNN), Decision Tree (DT) and Artificial Neural Networks (ANN) to better evaluate our descriptors. We illustrate the potential of extracted characteristics by two 3D benchmarks. The recognition rates achieved in all experiments show that a small number of curve between 9 and 12 can correctly categorize a family of 3D objects.

Keywords*: shape retrieval, eikonalequation, M-SVM, ANN, KNN, k-means, C4.5*

1. Introduction

Three-dimensional object recognition is subject to researches of several authors [1-2]. This domain has numerous applications in the real world such as a robotics, artificial intelligence and more [1-3]. Many approaches have been proposed [4-5]. The first one is to use the characteristic views of 3D object for feature extraction. This approach is most often taken in the recognition systems of 3D objects. D. Naji al [4] have proposed a method based on the characterization of 3D objects by a set of 7 characteristic views, A.Theetten& al [5] used the visual hulls to find how many characteristic views are required and what relative positions are optimal.

The second kind involves techniques designed to handle either polyhedral or conic objects [6]. Many approaches based on 3D Shape descriptors were proposed [7]. Extended Gaussian Images [8-9] were frequently used, based on a mapping of normal surface of an object onto the unit sphere (Gaussian sphere.) extended to include areas of each face. The downsides of this approach are the face position information is lost and which faces are connected.

The Shape-Diameter Function (SDF) [10] is a scalar function defined on the mesh surface. It expresses the diameter of the object’s volume in the neighborhood of each point on the surface. The SDF was used for segmentation, skeleton detection and extraction.

Shapira et al [10] proposed to use the normalized histogram of SDF as a geometrical attribute to represent each part of the objects. To find similarity between objects, they measures similarity between the two objects parts using a metric as normalized histogram of the part, using SDF values.

To reinforce their approach Shapira et al [10] use a tree scheme to define the structure of parts inside the object. By using the part context (the path between the node representing the part, and the root of the tree) as a part of the retrieval process, the SDF based system is very useful to detect and compare objects with simple geometry for complex objects we can notice two problems:

a) The level of decomposition: which means that a finger is a hand part or it should be considered as separate part.

b) The number of parts to be detected.

Received April 27, 2014; Revised July 21, 2015; Accepted August 5, 2015

c) The way to distinguish object with different number of parts. Some time we can find similar object with different number of parts.

In the 2D/3D approach a 3D object is represented by a set of 2D view (they use a large dataset of 2D views). Features are extracted among the different single-view [9].

Since every object is a set of 2D view, a linear classifier can be trained in a discriminative manner (or a binary SVM classifier can be trained) to categorize each object (set of view).

In practice, this method has two major disadvantages in the 3D objects recognition process. When the image is seen from different angles of view, different objects might be confused. On the other hand, the presence of shadows and spatial distortions leads to losses of information.

K. Ding et al [9] propose a 3D model descriptor called Sphere Image, which is defined as a collection of view features. The model is an improvement of the characteristic views approaches.

A. Eitz et al [10] propose 3D object retrieval based on sketched feature lines as input, the system turns out to be generally quite fast and useful for simple object.

In my knowledge no single shape descriptor and classification performs best for all 3D objects representation. This field has attracted many researches.

In this paper, we use a system based on the extraction of features by the calculation of the geodesic distance [9] on a mesh using an iterative algorithm for solving the Eikonal equation. For good classification results we adopt M-SVM to achieve excellent recognition rates in experiment. The method has been implemented as a system of 3D object recognition. After outlining the system, the rest of this paper describes the recovery of 3D information by the geodesic distance, the object recognition algorithms (Figure 1) shows an overview of the system. Experimental results are finally shown to demonstrate the effectiveness of the method.

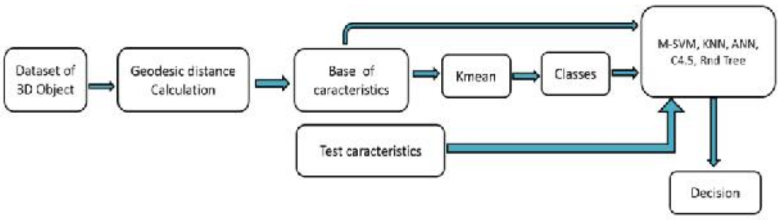


Figure 1. Flow-chart of recognition system

This paper is organized as follows; in section 2 we describe the mathematical background used. Section 3 deals with classification approach. The last section describe the databases and experiments results.

2. Riemannian Manifold

Traditionally, methods [11-14] for calculating the geodesic path are classified into two categories, depending on whether we calculate the distance between a source vertex and all other vertices or the distance between a pair of fixed vertices which will be called starting point and arrival point. The Dijkstra's algorithm [15] which allows exhaustively finding the shortest path between a starting vertex and all other vertices of the mesh.

In this section, we illustrate some basic definitions of the Riemannian manifold [9] which unifies the solution to all the problems discussed in this summary. This concept requires only designing a local metric, which is then integrated over the entire domain to obtain the distance between pairs of points. The important point is that the geodesic distance to a set of starting points satisfies a nonlinear differential equation. The Eikonal equation is solved numerically by calculating the geodesic distance.

A Riemannian manifold is a set equipped with a metric space $\mathcal{M} \in \mathbb{R}^2 \longrightarrow \mathrm{H}(\mathrm{x}) \in \mathbb{R}^{3\times2}$ positive definite.

Using the Riemannian metric, we can calculate the length of the curve γ as follows:

$$\gamma : [0,1] \rightarrow \mathcal{M}$$
$$L(\gamma) = \int_0^1 \sqrt{\gamma'(t)^T H(\gamma(t))\gamma'(t)}dt \tag{1}$$

The Remote $\mathcal{U}_s$ set to $\mathcal{M}$ with respect to the starting set of points $\mathcal{S} = (x_k)_k \subset \mathcal{M}$ is given by:

Let $(\mathcal{M}, H)$ be a Riemannian space $\mathcal{M} \subset \mathbb{R}^2$, the geodesic distance is defined by:

$$\forall(x,y) \in \mathcal{M}^2,\ d_{\mathcal{M}}(x,y) = min_{\gamma \epsilon P(x,y)} L(\gamma) = L(\gamma^*) \text{ and } \gamma^* = argmin_{\gamma \epsilon P(x,y)}\ L(\gamma) \tag{2}$$

With respect to the starting set of points.

The distance function $\mathcal{U}_s$ defined on $\mathcal{M}$ relative to the starting set of points $\mathcal{S} = (x_k)_k \subset \mathcal{M}$ is given by Equation (3) [12]:

$$\forall\ x \in \mathcal{M},\ \mathcal{U}_s(x) = min_k\, d_{\mathcal{M}}(x, x_k) \tag{3}$$

If the metric H is continuous, then for every starter set S⊂M the distance function $\mathcal{U}_s$ is the unique solution of the Hamilton-Jacobi equation.

$$\begin{cases} \|\nabla_x \mathcal{U}_s\|_{H(x)^{-1}} = 1 \\ \mathcal{U}_s(x_k) = 0, \forall\ k \end{cases} \tag{4}$$

The geodesic curve γ that realizes the minimum between x and the points of S is the solution of the following Hamilton-Jacobi equation:

$$\begin{cases} \gamma'(t) = \frac{H(x)^{-1}\nabla_{\gamma(t)}\mathcal{U}_S}{\|H(x)^{-1}\nabla_{\gamma(t)}\mathcal{U}_S\|} \\ \qquad \gamma(0) = x \end{cases} \tag{5}$$

For the discretization of Eikonale Equation we consider a small neighborhood $B(x)$ of each point $x \in \Omega\backslash\mathrm{S}$ such that $B(x) \cap S = \emptyset$, where $\mathcal{U}_s$ is the unique solution of the equation:

$$\begin{cases} \forall x \in \Omega, \mathrm{U}(x) = \min_{y \in \partial B(x)} U(y) + d(y,x), \\ \forall x \in \mathrm{S}, \mathrm{U}(x) = 0 \end{cases} \tag{6}$$

Where $d(y,x)$ is the geodesic distance defined above by the Equation (7):

$$\begin{cases} \quad U = \Gamma(U) \\ U(x) = 0\ if x \in S \end{cases} \tag{7}$$

Where Γ is an operator defined as follows:

$$\mathrm{V} = \Gamma(\mathrm{U}) \Leftrightarrow \mathrm{V}(x) = \min_{y \in \partial B(x)} U(y) + d(y,x) \tag{8}$$

The $\mathcal{U}_s(x)$distance function for $x \in \Omega$ can be approached numerically by a vector $u \in \mathbb{R}^n$ such that each component u_i represents an approximation of the value of $\mathcal{U}_s(x_i)$. Then discrete Eikonal equation is given by (9) [11]:

$$\begin{cases} \forall x_i \in \Omega, u_i = \min_{y \in \partial B(x_i)} u(y) + \|y - x_i\|_{T_i^{-1}}\ , \\ \forall x_i \in \mathrm{S}, u_i = 0. \end{cases} \tag{9}$$

Where T_i is the metric tensor associated with the point x_i.

To calculate numerically the solution of the Eikonal equation [12] which is nonlinear, we can use the method based on iterations of Jacobi, giving it an initial iterate $u^{(0)}$, we can calculate the value of the solution u as follows [11]:

$$u^{k+1} = \Gamma(u^k) \tag{10}$$

In practice we choose three starting points and three arrival points, each distance is the sum of geodesic distance on every mesh from the starting point to the end point.

3. Classification

Support Vector Machines (SVM) [18] are a class of learning algorithms that can be applied to any problem that involves a phenomenon f that produces output y=f(x) from a set of input x and the goal is to find f from the observation of a number of couples input/output.

The solution of binary classification problems using the Support Vector Machines (SVM) method is well developed [19-20].

However, the SVM was originally developed for binary decision problems, and its extension to multi-class problems is not straightforward. Multi-class pattern recognition problems (where one has 2 classes) are typically solved using voting scheme methods based on combining several binary classification decision functions [21, 22].

The popular methods for applying SVMs to multi-class classification problems decompose the multi-class problems into many binary-class problems and incorporate binary-class SVMs (M-SVM). Although the “one-against-one” approach demonstrates superior performance [23-24], it may require prohibitively-expensive computing resources for many real-world problems. The “one-to-the others” approach shows somewhat less accuracy, but still demands heavy computing resources, especially for real-time applications. After doing the following comparison [25], "one-against-one" is a good method whose performance is comparable to "one-against-the rest." We do latter simply because it’s being used widely in the Multiclass Support Vector Machines literature to solve multi-class pattern recognition problems; because it’s training time is shorter as in [21, 22].

4. Results and Analysis

4.1. Databases

We tested our method on two databases: the Princeton benchmark [26] and the Watertight dataset [27].

4.1.1. Princeton

The Princeton Shape Benchmark [26] is a set of 3D polygonal models collected from the web, the benchmark contains 907 models for training, and the test database contains 907. For each 3D model, there is an Object File Format (.off) (Figure 2) present some objects from the same family.

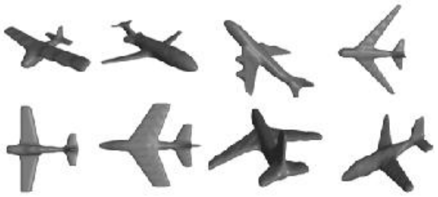


Figure 2. Different object of the family Fly

4.1.2. The Watertight Datasets

Watertight datasets [27] consists of 400 non-rigid 3D models which are classified into 400 categories.Objects are represented by seamless surfaces. This database provides a family of 3D object of different classes as show in the Figure 3.

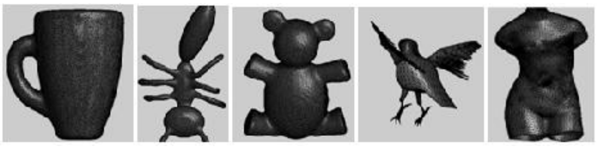


Figure 3. some views of 3D objects

4.2. Experiments

The first goal of this experiment is to detect the minimal number of curves needed to categorize a family of 3D objects. After that we are going to evaluate the selected number using different techniques.

To identify the number of geodesic curve needed to categorize a family, we tested the following combinations:

a) 2 starting points and 3 arrival points, (6 curves).
b) 3 starting points and 3 arrival points, (9 curves).
c) 3 starting points and 4 arrival points, (12 curves).
d) 4 starting points and 4 arrival points, (16 curves).
e) 4 starting points and 5 arrival points, (20 curves).

We choose the SVM classifier to evaluate the proposed combinations. Table 1 present recognition rate using different kind of kernel. We should notice that the runtime and the system performance, dependent essentially of the complexity of the object. In general, it depends on the number of vertices and faces of each mesh.

Table 1. The recognition rate for different number of curve using geodesic distance and M-SVM applied to the Watertight dataset

M-SVM (kernel)	6curves	9curves	12curves	16curves	20curves
Rbf	69,18%	98,46%	96,46%	94,47%	91,46%
Polynomial	41,13%	72,31%	73,31%	71,21%	68,31%
Quadratic	65,20%	98,23%	98,78%	95,13%	87,43%
Linear	68,36%	96,46%	97,16%	94,76%	89,55%

The kernel is effectively a similarity measure, so choosing a kernel according to prior knowledge is a good idea. In the absence of expert knowledge, we choose to perform our test using the four kernels.

Except for the polynomial kernel, all kernels present good results. Concerning the number of curves needed to classify family of 3D objects the best results was found for the two sets composed of 9 and 12 curves. The decreasing of classification rate for objects represented by 20 geodesic curves can be explained by the fact that more the number of curve increase the descriptors become more specific. As a result it can be used to categorize a single object or to separate object from the same family.

In the second test, we try to evaluate the effectiveness of 9 curves in similarity detection. Table 2 show some results obtaining by using the KNN with K=4.

Table 2. 3D objects retrieval using geodesic distance

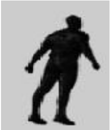
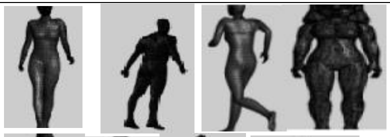
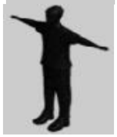
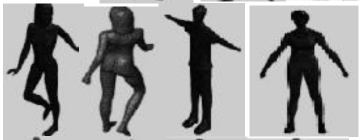
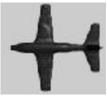
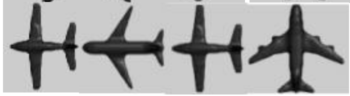
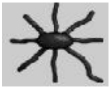
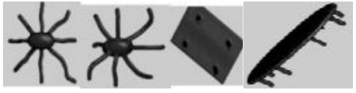
Requested Image	Results using geodesic distance
	
	
	
	

Table 2 show the efficiency of the geodesic distance; all returned object can be considered similar.

In the next part of the experiment we evaluated 9 curve set using different techniques. Firstly by evaluating the intra-connection of objects from the same class and secondly by the quality to separate object from different class using the Princeton dataset.

4.2.1. Evaluation of Classes Intra-Connection

By using the k-means clustering algorithm [28] we found the same repartition of vectors. (Figure 4) present the projection of vectors of each class while (Figure5) present the class centers.

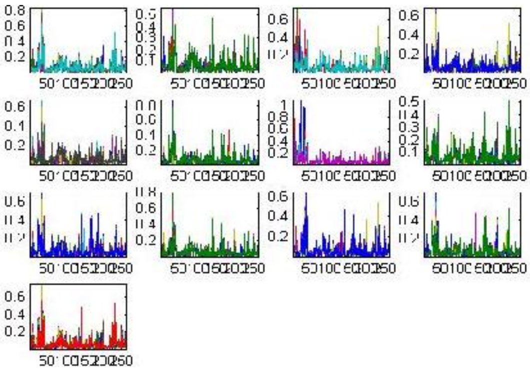


Figure 4. Repartition of Watertight dataset using K-means algorithm

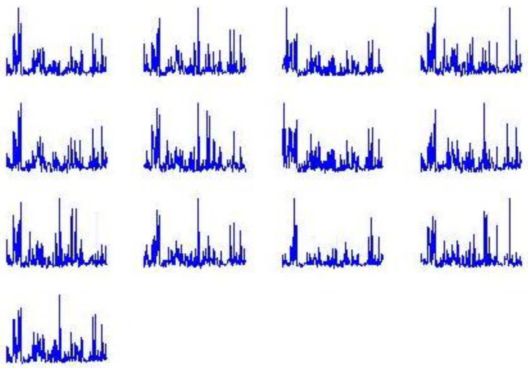


Figure 5. Class centers of Watertight dataset using K-means algorithm

Using K-means clustering more than 97% of vectors was assigned correctly to their class of member ship.

4.2.2. Evaluation of Classes Inter-Connection

Table 3. Geodesic distance evaluation using PCA

Axis	Eigen value	Difference	Proportion (%)	Histogram	Cumulative (%)
1	107.977441	97.909587	59.00 %		59.00 %
2	10.067854	0.340559	5.50 %		64.51 %
3	9.727296	2.395702	5.32 %		69.82 %
4	7.331594	1.973111	4.01 %		73.83 %
5	5.358483	0.359001	2.93 %		76.76 %

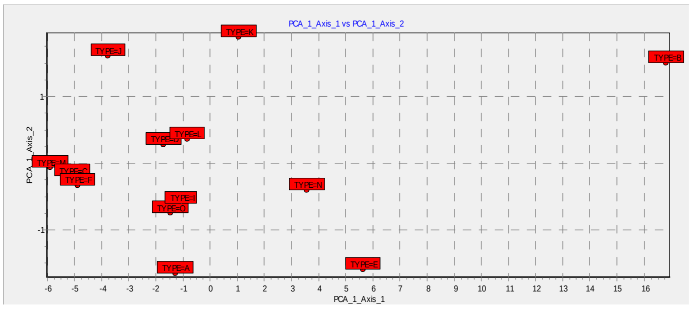


Figure 6. Class projection using two first axis of the ACP

To evaluate class inter connection of Watertight dataset we start by using the Principal Component Analysis (PCA) [29] to analyze the matrix representing the data set. Table 3 shows the first five components.

Just by using two axis we can notice the distinction between the different classes of member ship (Figure 6).

4.2.3. Evaluation Using Data Mining Algorithms

The data set of models has been split into a training and a test Database. Algorithm should be trained on the training database (without influence of the test database). Then, after all exploration has been completed and all algorithmic parameters have been frozen, results should be reported for the test database.

Table 4. Recognition rate of Watertight dataset

Class	KNN	ANN	C4.5	Random Tree
A1	0.3158	0.7895	0.7368	0.99
A2	0.45	0.85	0.7	0.98
A3	0.5	0.15	0.85	0.98
A4	0.25	0.1	0.5	0.97
A5	0.55	0.7	0.6	0.97
A6	0.4	0.8	0.5	0.96
A7	0.5	0.65	0.6	0.96
A8	0.4	0.3	0.6	0.95
A9	0.15	0.8	0.5	0.95
A10	0.2	0.45	0.75	0.94
A11	0.8	0.35	0.85	0.94
A12	0.45	0	0.6	0.93
A13	0.55	0.1	0.5	0.93
Computing times	936ms	4618 ms	1888ms	172ms

We evaluate the proposed descriptors (using 9 curves) using multiclass classification technics like ANN, KNN, C4.5, Random Tree (Table 4).

The classifiers parameters used in the experimental are the following:

a) **KNN parameters**: Neighbors 5 and the HEOMDistance (Heterogeneous Euclidean Overlap Metric).
b) **ANN parameters**: One hidden layer, Neurons in the hidden layer 10, Validation set proportion is equal to 0.2, and Error rate threshold 0.01.
c) **Decision tree (C4.5)**: parameters: Min size of leaves 5, Confidence-level for pessimistic 0.25. Number of nodes 71, Number of leaves 36.
d) **Random Tree**: Number of attribute for split = -1, Number of nodes 251, Number of leaves 126.

We can notice the very good results achieved using Random Tree: more than 93% as classification rate for all class of membership are obtained.

5. Conclusion

In this article, we have presented an approach for 3D objects retrieval based on geodesic distance, applied to an inhomogeneous database. We focus on study of number of geodesic curves needed to categorize a family of 3D objects. Based on various test using clustering, statistical and classification technics. We can conclude that a small number of geodesic curves between 9 and 12 are enough to characterize a family of 3D objects.

References

[1] F Tombari, S Salti, L Di Stefano. *Unique signatures of histograms for local surface description*. In European Conference on Computer Vision. 2010: 356-369.

[2] FM Sukno, JL Waddington, PF Whelan. *Comparing 3D descriptors for local search of craniofacial landmarks*. In Advances in Visual Computing. 2012: 92-103.

[3] Y Guo, J Wan, M Lu, W Niu. A parts-based method for articulated target recognition in laser radar data. *Optik.* 20013; 124(17): 2727-2733.

[4] Driss Naji, M Fakir, O Bencharef, B Bouikhalene, A Razouk. Indexing of Three Dimensions Objects Using GIST, Zernike & PCA Descriptors. *IAES International Journal of Artificial Intelligence (IJ-AI)*. 2013; 2(1): 1-6.

[5] A Theetten, Vandeborre, Jean-Philippe, Daoudi M. *Determining characteristic views of a 3D object by visual hulls and Hausdorff distance*. Fifth International Conference on 3D Digital Imaging and Modeling. 2005.

[6] KC Wong, Y Cheng, J Kittler. *Recognition of polyhedral objects using triangle pair features*. Communications, Speech and Vision, IEE Proceedings I. 1993; 140(1).

[7] JP Tarel, WA Wolovich, DB Cooper. *Covariant-conics decomposition of quartics for 2D object recognition and affine alignment*. International Conference on Image Processing, 1998. ICIP 98, Proceedings. 1998.

[8] Horn BKP. *Extended Gaussian images*. IEEE Proceedings. 2005; 72(12).

[9] Ke Ding, Yun-Hui Liu. Sphere Image for 3-D Model Retrieval. *Multimedia, IEEE Transactions.* 2014; 16(5): 1369-1376.

[10] M Eitz, R Richter, T Boubekeur, K Hildebrand, M Alexa. *Sketch-based shape retrieval*. ACM Trans. Graph (Proc. SIGGRAPH). 2012; 31(4): 1-10.

[11] G Peyré, M Péchaud, R Keriven, LD Cohen. Geodesic Methods. *Computer Vision and Graphics*. 2009; 5(3-4): 197-397.

[12] Shapira L, Shamir A, Cohen-Or D. Consistent mesh partitioning and skeletonisation using the shape diameter function. *Vis. Comput*. 2008; 24: 249-259.

[13] Mount David M. *On Finding Shortest Paths on Convex Polyhedra*. Maryland univ college park center for automation research. Tech. Rep. ada 166246. 1985.

[14] JM Itchell, D Mount, C Papadimitriou. The discrete geodesic problem. *SIAM J. Comput*. 1987; 16(4): 647-668.

[15] T Horup M. *Undirected single source shortest path in linear time*. In FOCS. 1997: 12-21.

[16] JM Itchell. *Geometric shortest paths and network optimization*. In Handbook of Computational Geometry. Elsevier Science, Amsterdam. 1998.

[17] Dijkstra, Edsger W. EWD: A Short Introduction to the Art of Programming. TH Eindhoven, the Netherlands. 1971.

[18] KB Duan, SS Keerthi. *Which is the best multiclass SVM method an empirical study*. Control Division, Department of Mechanical Engineering, National University of Singapore. Technical Report CD-03-12. 2003.

[19] Vladimir N Vapnik. An Overview of Statistical Learning Theory. *IEEE Transactions on Neural Networks.* 1999; 10(5).

[20] B Schölkopf, S Mika, CJC Burges, P Knirsch, KR Müller, G Raetsch, A Smola. Input space vs. feature space in kernel-based methods. *IEEE Transactions on Neural Networks*. 1999; 10(5): 1000-1017.

[21] B Schölkopf, CJC Burges, V Vapnik. Extracting Support Data for a Given Task. In. Fayyad UM, Uthurusamy R. *Editors*. Proceedings, First International Conference on Knowledge Discovery & Data Mining. Menlo Park, CA: AAAI Press;1995.

[22] V Blanz, B Schölkopf, H Bultoff, CJC Burges, V Vapnik, T Vetter. *Comparison of View-Based Object Recognition Algorithms Using Realistic 3D Models.* International Conference on Artificial Neural Networks. 1996.

[23] F Schwenker. *Hierarchical Support Vector Machines for Multi-Class Pattern Recognition*. Fourth International conference on Knowledge-Based Intelligent Engineering Systems & Allied Technologies. Brighton, UK. 2000.

[24] Y Lee, Y Lin, G Wahba. Multicategory Support Vector Machines, Theory, and Application to the Classification of Microarray Data and Satellite Radiance Data. *Journal of the American Statistical Association as the lead paper in the Theory and Methods section.* 2004; 99: 67-81.

[25] CW Hsu, CJ Lin. A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks.* 2002; 13(2): 415-425.

[26] P Shilane, P Min, M Kazhdan, T Funkhouser. *The Princeton Shape Benchmark.* Proceedings of SMI 2004: International Conference on Shape Modeling and Applications. 2004: 167-178.

[27] Daniela Giorgi, Silvia Biasotti, Laura Paraboschi. *Watertight Models Track*. Shape Retrieval Contest 2007: CNR – IMATI Via De Marini 6, Genoa, Italy. 2007.

[28] JA Hartigan, MA Wong A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society*, *Series C (Applied Statistics).* 1979; 28(1): 100-108.

[29] Jean-Paul Benzécri. Data analysis. T2 (lessons on factor analysis and pattern recognition and statistical work of the Laboratory of the University of Paris 6. T. 2: correspondence analysis). Dunod Paris Brussels Montreal. 1973.