Spin Images and Neural Networks for Efficient Content-Based Retrieval in 3D Object Databases

Pedro A. de Alarcón, Alberto D. Pascual-Montano, and José M. Carazo

Biocomputing Unit. Centro Nacional de Biotecnología (CSIC)
Campus Universidad Autónoma de Madrid, 28049 Madrid, Spain
{pedro,pascual,carazo}@cnb.uam.es
http://www.biocomp.cnb.uam.es

Abstract. We describe a system for querying 3D model databases using the *spin image* representation as a shape signature for objects depicted as triangular meshes. The spin image representation facilitates the task of aligning the query object with respect to matched models (coarse-grain registration). The main contribution of this work is the introduction of a three-level indexing schema based on artificial neural networks. The indexing schema improves significantly the efficiency in matching query spin images against those stored in the database. Our results are suitable for content-based retrieval in 3D general object databases. A particular application to molecular databases is also presented.

1 Introduction

Retrieving objects by their content as opposed to keyword indexing or simple browsing has become an important operation, and consequently, an active field of research. After more than a decade of intensive research, *content-based image retrieval* (CBIR) technology moved out of the laboratory and into the marketplace, in the form of commercial products like QBIC [1] and Virage [2]. CBIR draws many of its methods from the field of image processing, computer vision, pattern recognition, and database technology.

As web-based repositories of multi-dimensional scientific data continue to grow, so does the need for content-based retrieval of three-dimensional (3D) scalar data. One of the most prevalent source of volumetric data is medical and biological imaging. During the last decade bioscientists have witnessed an spectacular growth of molecular databases. Molecular databases store in some cases (Protein Data Bank [3]) thousands of molecular complexes described as three dimensional datasets. Public access to these repositories boosts research targeted at the discovery of new drugs and medicines. Certainly, content-based retrieval would provide scientists with a valuable tool that facilitates, for example, the task of finding molecules that are structurally similar to a given one. To do so, different features are used to represent the content of

M. S. Lew, N. Sebe, and J. P. Eakins (Eds.): CIVR 2002, LNCS 2383, pp. 225-234, 2002.

a 3D object. Among them, shape is the most relevant attribute for the task of object recognition. There exist a number of CBIR systems that use different shape features in order to perform similarity retrieval in 3D general model databases. Most of them use the object's surface (e.g: VRML description) to derive a signature that represents its shape. Some examples are Nefertiti [4], the Ferret Project [5], or the 3D search engine depicted in [6]. Other systems are tailored to domain-specific databases of volumetric datasets. In [7] a 3D neuroradiologic image retrieval system is presented. It deals directly with multimodal 3D brain images. They use thematic data and image-based descriptors in order to better characterize pathological versus normal brains. This is a good example of how a CBIR system is perfectly adapted to a particular application domain. Nevertheless, it results hard to figure out how to extend their developments to other 3D model databases. Ankerst *et al.* [8] introduced a novel approach for similarity search in 3D protein databases. Shape histograms and chemical properties of proteins form the objects′ feature vectors. The shape histograms are built from partitioning the 3D space into concentric shells and sectors. They assume proteins to be given as sets of 3D points. However, we still believe that some difficulties may appear from using that shape signature since it is neither scale-invariant, nor translation invariant. Also, it does not provide a description of the protein's shape at the local level. This issue is quite important as many molecular interactions take place at very precise locations on the protein surface (active sites).

In the present manuscript, we describe a system for querying 3D model databases using the *spin image* representation [9]. The spin image representation provides a local description of the surface that is translation/rotation-invariant. A normalization step is used in order to make it invariant to scale changes. Nevertheless, the main contribution of this work is the introduction of a 3-level indexing schema based on artificial neural networks. The indexing schema improves significantly the efficiency in matching query spin images against those stored in the database. Our schema makes the spin image representation potentially suitable for content-based retrieval in large 3D model databases

The rest of the paper is organized as follows. Section 2 is devoted to briefly analyse the representation of 3D free-form objects and to describe the spin image representation. Then, the indexing mechanism is presented (section 3). Our system is tested with some small 3D object databases in section 4. The paper concludes with the main conclussions and some guidelines for future work (section 5).

2 Object Representation and Shape Features

Three-dimensional object recognition uses the true 3D shape of objects in its model representation. 3D data can come in the form of depth-maps, isolated 3D points and lines, or 3D intensity images, depending on the sensor and sensing algorithm. In this work, we will consider the shape as a geometric concept derived from the object's surface. Definitions of free-form surfaces and objects are often intuitive rather than formal. Besl [10] stated, "a free-form surface has a well defined surface normal that is continuous almost everywhere except at vertices, edges and cusps". Many other definitions exist though.

2.1 Object Representations in Computer Vision

Unfortunately, most 3D file formats (VRML, 3Dstudio, etc) have been designed for visualization. They contain only geometric and appearance attributes, and usually lack of semantic information that would facilitate object recognition. In a computer vision context [11], the object representation is designed for use in the specific vision application (such as navigation, recognition, or tracking), and visual fidelity may not be a criterion of interest. On the other hand, efficient execution, saliency and locality are desirable features. Several surface representations in the context of computer vision have been proposed in the literature [12]. Among those that adopt complete mathematical forms, parametric surfaces, algebraic implicit surfaces, superquadrics and generalized cylinders are the most popular ones. They all are global and compact representations but lack of some characteristics. For example, only parametric forms allow local control of the surface, but in contrast, they are difficult to fit to the actual object's contour. Polygonal meshes have become a popular representation for 3D objects. Meshes allow local control of the surface and can faithfully approximate complex free-form objects to any desired accuracy given sufficient space to store the representation. In the past this was their major limitation, but with the decreasing cost of computer memory and advances in hardware for computer graphics, even very dense meshes have become practical. In our system, we will assume 3D objects to be represented as polygonal meshes (also known as polygon soups). A polygonal mesh is defined by two components: a list of 3D vertices and an indexed list of polygons each specified as al list of vertex indices. We restrict polygons to be triangles because triangle meshes are easy to manipulate and efficiently rendered.

2.2 The Spin Image Representation

The problem of determining the similarity of two shapes is a fundamental task in shape-based recognition, retrieval, clustering, and classification [13,14]. In general, matching methods are grouped according to their representation of shape: 2D contours, 3D volumes, 3D surfaces, structural models and statistics. Unfortunately, most of the methods for 2D shape matching are not extensible to the 3D case. Nevertheless, a number of methodologies have been proposed for three-dimensional shape matching [14]. For the present work we have chosen the spin image representation [9]. This model-based representation combines the descriptive nature of global signatures with the robustness to partial views and clutter of local features. The representation allows achieving both object recognition and coarse-grain registration. Also, as no assumptions about the topology or shape are made, arbitrarily shaped objects can be recognized.

Given a polygonal mesh, an oriented point O on the surface is specified by its 3D position p and surface normal n (fig. 1). We have chosen the centroids of triangles in the mesh as the location of oriented points and the corresponding triangle's normals as their associated direction. Thus, a given object has as many oriented points as the number of triangles that form the mesh. Each oriented point defines a local coordinate system using the tangent plane P through p oriented perpendicularly to n and the line L through p parallel to n. The two coordinates of the basis are α, the perpendicular distance to the line L, and β the signed perpendicular distance to the plane P. There is an unique function ("spin map") that maps each 3D point x on the surface into (α, β) coordinates with respect to the oriented point p. The spin map provides a mapping from points on the surface into a two-dimensional space where some of the 3D metric is preserved. A 2D histogram can be built from the spin map by accumulating 2D positions into discrete bins. Bilinear interpolation of the spin-map coordinates is used in order to reduce the effects of possible noise in the data. This histogram is called "spin image" and it is unique for each oriented point in the surface. For each oriented point on the surface of the object, its spin image is computed. The set of all spin images forms the "spin image stack" and it is a global description of the object's shape.

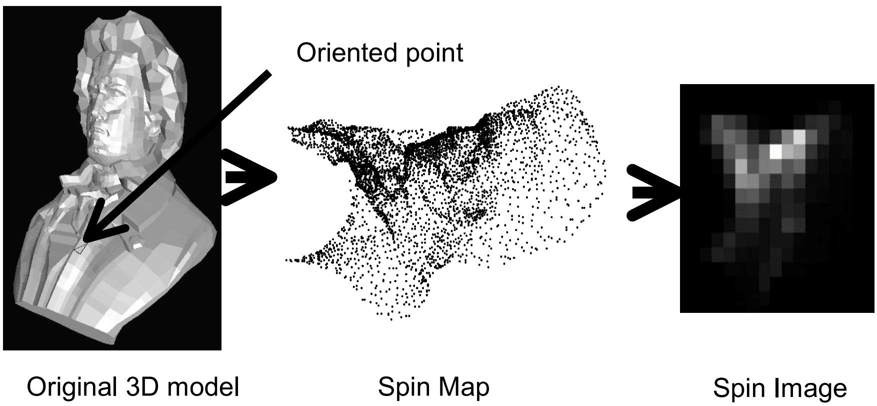

Fig. 1. For each oriented point (triangle center) on the triangular mesh, a spin map can be computed by an unique mapping into a 2D space. The spin map is discretized into bins so that the spin image from that oriented point is obtained

A spin image is an object centered (i.e: pose-independent) encoding of the shape of an object because the spin-map coordinates of a point with respect to a particular oriented point basis are independent of translational and rotational transformations. However they are still scale-dependent since the α,β values depend on the scale of the original 3D space. To avoid this, we normalize each vector (spin image) to unit length. As a consequence, spin images are much more robust against resolution changes. The resolution of a mesh is commonly defined as the average length of its edges for uniformly distributed meshes. Let's suppose two meshes representing the same object but one surface has a richer sampling than the other. Without normalization, their spin images would be different due to the difference in their ranges. Normalization also makes spin images to be scale-invariant and less sensitive to the metric that will be used thereafter for comparison. In [9] a modified version of the linear correlation coefficient is used as metric distance between two images. The metric considers the amount of overlap in the comparison of two spin images, which is important for partial matching purposes. In our case, Euclidean distance produced excellent results quite similar to those obtained with the linear correlation coefficient.

3 Artificial Neural Networks for Efficient Indexing

At recognition time, spin images from points on the model are compared to spin images from points in the scene. When two spin images are similar enough, the object (or a partial view of it) is recognized and a point correspondence between model and scene is established (registration). Models are stored together with their spin images stacks in a database. The user presents a query object to the database and ask for a "top list" of similar objects as well as which surface regions of the query object resemble to those of the retrieved objects. Efficiency is a critical factor in this procedure. The worst case computational cost of the query is $O(N_{db}N_q)$, where N_{db} is the total number of spin images stored in the database and N_q the number of spin images that were selected from the query object. It is essential to provide some indexing mechanism in order to quickly match scene spin images to model ones. Moreover, in [15] the authors pointed out that spin images coming from the same surface can be correlated for two reasons: first, spin images generated from oriented point bases that are close to each other on the surface will be correlated. Second, surface symmetry and the inherit symmetry of spin image generation will cause two oriented point bases on equal but opposite sides of a plane of symmetry to be correlated. Even more, surfaces from different objects can be similar on the local scale. Therefore, it may exist a correlation between spin images of small support generated from different objects. We can benefit from this correlation to make spin image comparisons more efficient through mapping the total number of spin images into a reduced representative set of them. A different approach is proposed in [9]. Spin images are compressed to reduce the amount of correlation among them. Spin images can be considered as p-dimensional vectors. Principal component analysis (PCA) was used in order to map p-dimensional vectors into s-dimensional space where $s<p$.

Artificial neural networks (ANN) have been applied in similar contexts [16,17]. In this work we propose an improved indexing method based on ANN to efficiently access to spin images. Our method achieves both compression and indexing of the original set of spin images. Basically, a self-organized map (SOM) is built from the stack of spin images of a given object. This is a way of "summarizing" the whole stack into a set of representative spin images. Then, a clustering algorithm (kcmeans) is applied in order to group representative views in the SOM map into a reduced set of clusters (fig. 2). At query time, spin images will be first compared with the clusters centers resulting from the kcmeans method and subsequently with the SOM map if finer answer is requested.

3.1 Self-Organizing Feature Maps (SOM)

The Self-Organizing Map is a neural network that simulates the hypothesized self-organization process carried out in the human brain when some input data are presented [18]. The structure of this neural network is composed of two layers: an input layer formed by a set of units (on for each component of the input vector) and an output layer formed by units or neurons arranged in a low dimensional grid (usually two-dimensional). The algorithm maps a set of input vectors (spin images in our case) onto a set of output vectors (neurons), but unlike other mapping algorithms, the output response is ordered according to some characteristic feature of the input vectors. It can be interpreted as a nonlinear projection of the p-dimensional input onto an output array of nodes. Each neuron has a vector of coefficients associated with it ($v_i \in \Re^p$) usually known as "code vectors".

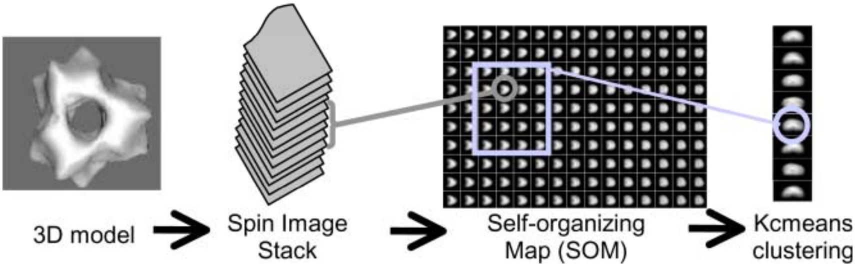

Fig. 2. Three-level indexing mechanism. A self-organized 2D map "summarizes" the information provided by spin images. Each output vector has an attached set of spin images. The third-level of indexing clusters the map into a reduced set of representative images (centers of clusters)

The functionality of the algorithm can be described as follows: when a spin image ($x_i \in \Re^p$) is presented to the net, the neurons in the output layer compete with each other and the winner (whose code vector has the minimum distance from the input spin image) as well as a predefined set of neighbor code vectors update themselves. This process is continued until some stopping criterion is met, usually, when code vectors "stabilize" or when a number of iterations are completed. Details of the algorithm can be found in [18]. The result of applying SOM to the spin image stack is a reduced set of spin images that represent the inherent clustering in the original set. At this level of indexing, comparison with a given object is reduced from thousands of spin images to less than two hundreds.

3.2 Kernel Probability Density Estimator Clustering Algorithm (Kernel C-means)

Kernel c-means clustering algorithm is a process of grouping similar objects into the same class by maximizing an objective function explicitly designed to estimate a set of the cluster's centers whose probability density function resembles as best as possible the probability density of the input data. The theoretical basis of these methods has been reported in detail elsewhere [19,20] and will only be briefly reviewed here. Given n data items of dimension p, $X_i \in \Re^{p \cdot 1}$ with $i=1 \ldots n$, the problem is to find c surrogate "representative" data items (centroids), $V_i \in \Re^{p \cdot 1}$ with $j=1 \ldots c$, such that the estimated probability density:

$$D(\mathbf{X}) = \frac{1}{c}\sum_{j=1}^{c} K\left(\mathbf{X} - \mathbf{V}_j; \alpha\right) \cdot \tag{1}$$

where K is a kernel function, with $\alpha > 0$ the kernel width parameter that controls the smoothness of the estimated density. Intuitively, the kernel estimator can be seen as a sum of "bumps" placed at the observations (data). The kernel function K determines the shape of the bumps while the parameter α determines the width. A commonly used kernel function is the well-known Gaussian kernel. In general, let $D(X;\theta)$ be the probability density for the random variable **X**, where θ is some unknown parameter vector. Let $X_i \in \Re^{p \cdot 1}$, $i=1\ldots n$, denote the data items then:

$$L = \prod_{i=1}^{n} D(\mathbf{X}_i;\theta) \tag{2}$$

is the likelihood function, and the most common statistical estimator for θ is obtained my maximizing eq. 1. Note that in this case the parameter vector is composed by the (code vectors) $V_i \in \Re^{p \cdot 1}$ and the Kernel width α, i.e: $\theta = \{\{Vj\},\alpha\}$. The log-likelihood is:

$$l = \sum_{i=1}^{n} \ln\left(D(\mathbf{X}_i)\right) = \sum_{i=1}^{n} \ln\left(\frac{1}{c}\sum_{j=1}^{c} K\left(\mathbf{X}_i - \mathbf{V}_j;\alpha\right)\right) \tag{3}$$

The Kernel c-means algorithm consists of an iterative optimization of the above objective function (eq. 3). As a result, we add an additional level of indexing following the self-organizing map to further reduce the number of comparisons to be done at query time. The indexing will be defined by two parameters *(d,c)* where *d* is the number of nodes in the self-organizing map (second level) and *c* the number of clusters produced by the kcmeans algorithm (third level).

Efficiency in accessing to stored spin images is thus related to these parameters. As said above, if no indexing mechanism exists worst case computational complexity is $O(N_{db}N_q)$ where N_{db} is the total number of spin images in the database. More particularly:

$$N_{db} = \sum_{i=1}^{N_{obj}} N_{spin\_images}(obj_i) \tag{4}$$

With our methodology, the parameter N_{db} is reformulated as $N_{db} = dN_{obj}$ at the second level, and $N_{db} = cN_{obj}$ at the third one. The reader should note that the number of "original" spin images per object equals to the number of triangles (which is quite often in the range of thousands). On the other hand, we have performed efficient and accurate retrieval with (*d,c*) parameters in a range of 150 and 10 feature vectors respectively.

4 Experiments

Our system was tested against three small databases (fig. 3). The aircraft and molecular database contains very similar objects. It was intentionally done in order to measure the accuracy in the recognition process. The database of macromolecules was built from real datasets depicted as volumetric density images. A method for obtaining a density-oriented mesh representation from volumetric datasets is detailed in [20].

For each query object the complete spin image stack is computed. The accuracy is defined by the number of query spin images that were correctly classified when compared to the third level of indexing. All the experiments were performed with (p,c)=(150,10). Note that the number of comparisons per object is reduced from several thousands to 10.

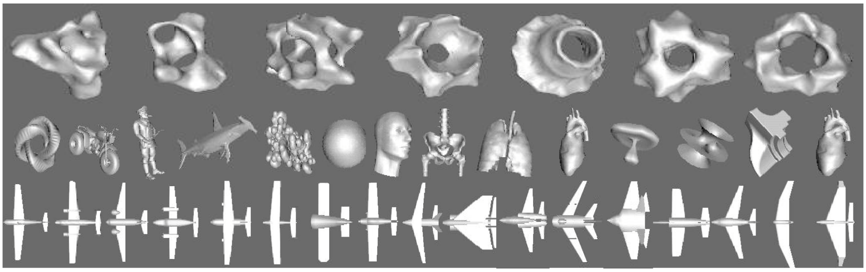

Fig. 3. Three test databases (one per row). Each object was presented to the database as a query input. The query object was retrieved in first place 100% of times. Similarity between the query object and target is measured (average recognition accuracy, a.r.a for short). (a) Molecular database (first row). 7 models, a.r.a: 70% (b) Mixture database (second row). 14 models, a.r.a: 92% (c) Aircraft database, 17 models a.r.a: 73%. Note that recognition accuracy is lower than that for molecular and aircraft databases since they contain very similar objects. Nevertheless, for all query examples the similarity value with the second best matched object is lower than 10%. No ambiguity in the answer is possible

Given an arbitrarily shaped query object, a number of spin images are calculated. Let be $T_{obj}=\{t\}$, the set of triangles t that compose the mesh of such object and $SP_{obj}=\{s\}$ the corresponding set of spin images where spin image s_i corresponds to triangle t_i. The query spin image set QSP_{obj} is a subset of SP_{obj}. This is important as the user might be interested in retrieving those objects in the database that have similarity to a particular surface patch of the query object. Thus, locality is well supported.

At query time, Euclidean distance of spin images in QSP_{obj} and those at the 3rd (or 2nd) level of indexing is computed. A vote is added to the database object with closest spin image. The average recognition accuracy is the number of spin images in QSP_{obj} that were correctly classified assumed that the query object was already stored in the database (fig. 3). Another interesting way of visualizing the query result is by the use of locality. Triangles associated to spin images in QSP_{obj} might be colored according to the color assigned to the three best-matched objects in the database (fig. 4).

5 Conclusions and Future Work

In this work we have presented a shape-based retrieval system that is suitable to both general and domain-specific 3D object databases. It assumes free-form 3D objects to be provided as triangular meshes. The spin image representation has been employed as a robust shape signature. Our main contribution is the use of a self-organizing map and a clustering algorithm as an indexing mechanism so that the overall number of comparisons is significantly reduced. Thus, our method achieves indexing and data compression as only "representative" spin images are chosen. Also, it increases the robustness against noise (non-representative spin images). The experimental results confirm a remarkable improvement in the efficiency of the comparison procedure so that interactive queries are possible. Even more, the recognition accuracy penalty is minimum. An important issue for further work is to incorporate metrics (better than euclidean distance) at the indexing level that would allow to retrieve objects when only partial views are provided. Also, we plan to provide the user not only with a top list of matching objects but with the rigid transformations that would align the query object with the target. Application to large molecular databases is underway.

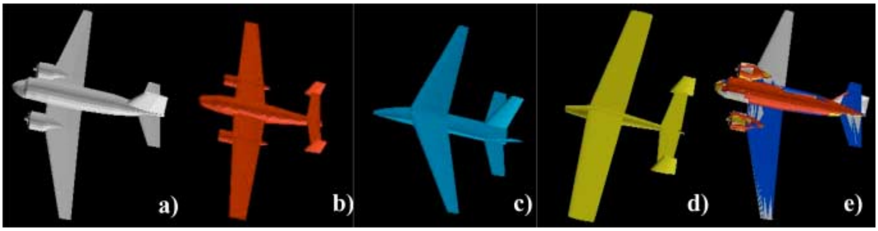

Fig. 4. An additional advantage of our system is that spin images support locality. A new plane is presented to the aircraft database (a). When compared with the third indexing level (kcmeans), the three most similar aircrafts are shown with a.r.a 27%(b) 13%(c) and 12%(d). Triangles in the query object were colored (d) according to the correspondence of their associated spin images with respect to matching items

References

1. Flickner, M. et al: Query by image and video content: the QBIC system. IEEE Computer Vol. 28(9) (1995) 23-32.
2. Gupta, A. et al.: The Virage image search engine: an open framework for image management. Storage and Retrieval for Image and Video Databases IV, Proc SPIE 2670 (1996) 76-87.
3. Sussman, Lin, Jiang, Manning, Prilusky, Ritter and Abola: Protein Data Bank (PDB): database of 3D structural information of biological macromolecules. Acta Crystallogr. Sect D 54 (1998) 1078-1084.
4. Paquet, E., Rioux, M.: Nefertiti: a tool for 3-D shape databases management, Proceedings of the SAE International Conference on Digital Human Modeling for Design and Engineering, The Hague, Netherlands. (1999)
5. Elvins, T., Jain, R.: Web-based volumetric data retrieval. Symposium on Virtual Reality Modeling Language, San Diego USA. (1995) 7-12.
6. Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D.: Matching 3D Models with Shape Distributions. Shape Modeling International , Genova, Italy, May, 2001.
7. Liu, Y., Rothfus, W. E., Kanade, T.: Content-based 3D Neuroradiologic Image Retrieval: Preliminary Results. IEEE International Workshop on Content-based Access of Image and Video Databases. (1998) 91 - 100.
8. Ankerst M., Kastenmüller G., Kriegel H.-P., Seidl T.:Nearest Neighbor Classification in 3D Protein Databases. Proc. 7th Int. Conf. on Intelligent Systems for Molecular Biology (ISMB), Heidelberg, Germany. AAAI Press (1999) 34-43.

9. Johnson, A., Hebert, M.: Recognizing objects by matching oriented points. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '97) (1997) 684–689.
10. Besl, P. J.: The free-form surface matching problem. Machine Vision for Three-dimensional Scenes. Academic Press, San Diego (1990) 25-71.
11. Hebert, M., Ponce, J., Boult and Gross, A.: Object Representation in Computer Vision. Springer-Verlag Eds., Berlin 1995.
12. Campbell, R. J., Flynn, P. J.: A Survey of Free-Form Object Representation and Recognition Techniques. Computer Vision and Image Understanding, Vol 1 (2001) 166-210.
13. Loncaric, S.: A survey of shape analysis techniques. Pattern recognition, Vol 31(8) (1998) 983-1001.
14. Veltkamp, R. C., Hagedoorn, M.: State of the art in shape matching. Technical Report UU-CS-1999-27, Utretch University, the Netherlands, 1999.
15. Johnson, A., Hebert, M.: Efficient multiple model recognition in cluttered 3-D scenes. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '98) (1998) 671 - 677.
16. Böhm C., Kriegel H.-P., Seidl T.: Adaptable Similarity Search Using Vector Quantization. Proc. Int. Conf. on Data Warehousing and Knowledge Discovery (DaWaK 2001), Munich, Germany, (2001).
17. Bernard, S., Boujemaa, N., Vitale, D., Bricot, C.: Fingerprint Classification using Kohonen Topologic Map. The IEEE International Conference On Image Processing, ICIP (2001).
18. Kohonen T.: Self-Organizing maps, Second Edition, Springer-Verlag Eds (1997).
19. Pascual-Montano, A., Donate, L. E., Valle, M., Bárcena, M., Pascual-Marqui, R. D., Carazo, J. M.: A Novel Neural Network Technique for Analysis and Classification of EM Single Particle Images. J. of Struct. Biol. 133 (2/3), (2001) 233-245.
20. De-Alarcón, P. A., Pascual-Montano, A., Gupta, A., Carazo, J. M.:Modeling shape and topology of low-resolution density maps of biological macromolecules. (2002) (Biophysical Journal, in press).