A Versatile Image Retrieval Platform based on a Multi-agent Architecture

Socrates Dimitriadis[1,2]
socrates@ics.forth.gr

Kostas Marias[2]
kmarias@ics.forth.gr

Stelios C. Orphanoudakis[1,2]
orphanou@ics.forth.gr

[1] Department of Computer Science, University of Crete, Heraklion, Hellas
[2] Institute of Computer Science, Foundation for Research and Technology, ICS-FORTH, Hellas

Abstract
Efficient and possibly intelligent image retrieval is an important task, often required in many fields of human activity. While traditional database indexing techniques exhibit a remarkable performance in textual information retrieval current research in content-based image retrieval is focused on developing novel techniques that are biologically motivated and efficient. It is well known that humans have a remarkable ability to process visual information and to handle the volume and complexity of such information quite efficiently. In this paper, we present a content-based image retrieval platform that is based on a multi-agent architecture. Each agent is responsible for assessing the similarity of the query image to each candidate image contained in a collection based on a specific primitive feature and a corresponding similarity criterion. The outputs of various agents are integrated using one of several voting schemes supported by the system. The system's performance has been evaluated using various collections of images, as well as images obtained in specific application domains such as medical imaging. The initial evaluation has yielded very promising results.

INTRODUCTION

Efficient access to information is very important in many application domains. In particular, visual information constitutes one of the most important stimuli for human perception. In recent years, interdisciplinary research has produced significant results in a number of areas in which traditional approaches appeared to have reached their limit. Based on the joint effort of biological neuroscientists, computational neuroscientists, psychologists, and computer scientists, the interdisciplinary field of cognitive science has emerged as the field of research in which brain function is investigated with respect to behavior and mental activity. Visual perception is a major brain activity and has attracted the attention of many cognitive scientists. Understanding the mechanisms of visual perception is important in a number of application areas, including content-based image retrieval (CBIR) [6].

The primary goal of our research is to go beyond traditional approaches to CBIR by implementing a versatile multi-agent architecture, while taking into consideration several aspects of human visual perception [19]. However, there are still many questions unanswered regarding the way humans process, represent, store and retrieve visual information from memory. Humans undoubtedly possess the ability to process visual information efficiently and to identify images as being similar based on their visual content. However, computational approaches are currently falling short of matching this remarkable human ability. The human cerebral cortex consists of many cortical areas, each one specialized and able to execute a specific task. Similarly, the visual cortex consists of a set of specialized modules, which are responsible for processing a specific visual feature and execute their respective tasks independently and in parallel [15]. An image similarity decision is not based on any one of these processes acting in isolation, but it is the result of integrating their individual contributions. This biological paradigm resembles the computational approach adopted in the proposed multi-agent system, since its architecture is an effort towards the modeling of various biological processes and behaviors [11], [12]. More specifically, the adopted interdisciplinary approach to content-based image retrieval is based upon:

1. A biologically inspired multi-agent architecture, which provides a platform for the experimental investigation of a variety of problems arising in CBIR and for the assessment of the perceptual relevance of similarity retrieval schemes involving the cooperation of selected agents. The system's architecture is versatile with respect to adding new agents and modifying their interaction in order to satisfy the requirements of specific application domains (e.g. medical image retrieval).
2. Several agents each one functioning independently in computing the similarity between the query image and each candidate image based on its specialized similarity criterion and matching algorithm. We are currently investigating the implementation of agents that resemble the visual perception of humans.
3. A voting system that supports the integration of partial results obtained by different agents. The user is allowed to select one of several alternative voting schemes [21], while the system loads and handles dynamically the set of available agents and voting methods.

Due to the fact that images represent a particularly large volume of information, the efficient and possibly intelligent browsing of images based on visual content is becoming increasingly important in application fields that make use of large image databases, such as diagnostic medical imaging, remote sensing, entertainment, etc. Surfing on the WWW may also be facilitated and made more intelligent if visual

browsing is exploited. Thus, many research groups have developed different approaches to CBIR and corresponding systems, such as QBIC, Blobworld, ImageRover, PikToSeek, Surfimage, VisualSEEk and others [13]. These systems provide a variety of alternative techniques for both feature extraction and querying, while others, like LPCD and MIR are specialized to specific application domains. The proposed image retrieval platform is described in the next section.

SYSTEM OVERVIEW

As shown in Figure 1, the system consists of four main components: a set of software agents, an image database, a graphical user interface, and a voting system that supports a set of alternative voting schemes. Images can be imported into the database, viewed individually or in galleries, and can be retrieved based on their visual content. Images that are similar to a query image may be retrieved from a specific set of candidate images (gallery) or from all images contained in the database. The query image and the set of candidate images are both selected by the user. During importing of images, each agent extracts the necessary features to be subsequently used for similarity matching and retrieval. This function incurs a heavy computational cost. However, since it is not a real time function, it is executed prior to retrieval and does not affect the system's overall performance.

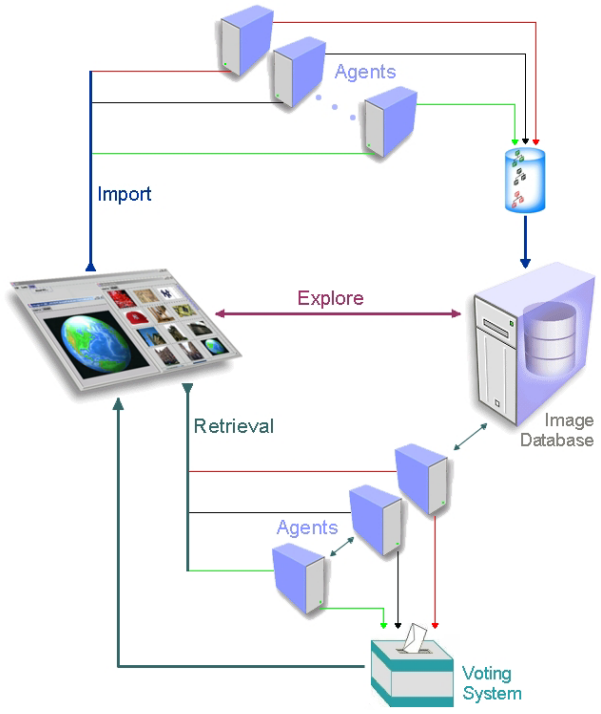


Figure 1: System overview

Image retrieval is a real time function during which each agent uses its preprocessed information and, therefore, consumes limited computational resources. A retrieval process is based on a query image and a set of candidate images (gallery) or all images contained in the database. Each agent works independently, and different agents work in parallel, to compute the similarity between the query image and each candidate image based on its specialized similarity criterion and matching algorithm. Integration of partial results obtained by different agents is achieved through the voting system, using a voting scheme selected by the user. Furthermore, the reasoning on which the final ranking of candidate image similarity is based is exported, thus allowing the user to interpret the response to a specific query and to fine tune relevant parameters in order to improve the response to subsequent queries. This reasoning consists of the contribution of each agent to the final ranking of each candidate image, according to the voting scheme selected by the user. The CBIR system is fully scalable and can be easily extended with additional agents or voting schemes in order to take into account the specific requirements of different experiments and the class of images under consideration. It is developed in Java™ and uses the Java Advanced Imaging package.

Agents

In a multiple agent platform, different agents can collaborate with each other in order to perform a specific image retrieval task. This cooperation is often necessary either because the domain of the problem may be too large to be handled by a single agent or the problem requires agents of different knowledge or skills, to contribute to the final solution [1]. This implies the need for heterogeneity among the agents and this is exactly the orientation of our approach.

In the fully general multi-agent scenario, there may be direct interaction among agents [11]. This interaction however can be indirect through the environmental stimuli. Similarly, the environment's dynamics and the uncertainty that maybe inherent in the domain strongly depend on the agents' actions and goals. In this work, the several agents interact indirectly through the voting system that acts as their environment. However, in our current work most of the agents have a suitable structure that lets them achieve a direct interaction, thus increasing their efficiency.

The use of multiple agents also satisfies the requirement for parallelism and independence. Exploiting parallelism in situations of high complexity is critical for the outcome of an image retrieval task. However, the target is not only to parallelize computations, but also to decompose a difficult task into several simpler ones. According to findings in Cognitive Psychology [5], the solution to a problem does not usually result in an all or nothing decision. Instead, it is very common that a set of factors or attributes, none of which is absolutely critical, contribute to a final decision. Similarly, in this approach, it is entirely possible that no one agent can yield a correct response to a given query by content. However, a correct response can be obtained based on the cooperation and integrated action of several heterogeneous agents. This requires that the design and implementation of the agents be consistent with the principles of simplicity and low computational cost. Nevertheless, the versatile architecture of the platform allows the introduc-

tion of more complex agents depending on the image-retrieval application. We should mention here that although in a more abstract level the general idea resembles a *committee machine* the implementation of the agents differs substantially from the respective neural network *experts*. Some of the *standard* agents used are presented below in greater detail:

Agent colorRGB

This agent deals with the color content of the image in the RGB model. During the import process the agent extracts the three histograms I_R, I_G, I_B, one for each color channel. Formally, each component v of the histogram I for the color channel C of an image E is given by

$$I_{C,v} = \frac{1}{MN}\sum_{i,j}\delta\left(E_{i,j}^{C} - v\right) \rightarrow v \in [1, colors] \quad (1)$$

where δ is the Kronecker function, M, N are the dimensions of the image and *colors* is the number of colors per channel. Dividing the sum by MN normalizes the values of the histogram so that they can be compared among images of different size. In the retrieval process equation (2) is used to compute the dissimilarity of the extracted histograms.

$$d = d_R + d_G + d_B \quad (2)$$

$$d_X = \sqrt{\sum_{i}^{256}\left(I_{X,i}^{a} - I_{X,i}^{b}\right)^2} \quad (3)$$

Agent colorHSV

This agent uses the HSV color model in combination with some statistical measures. HSV is useful for CBIR since it resembles the human perception of color. In order to use this model more efficiently for our purpose, we replace the less important and often redundant components of saturation and value with the mean and the variance of the hue channel. The coordinates of the mean value are normalized with the respective dimensions of the image, while the variance is normalized with the maximum possible variance in the image. The dissimilarity d between two images a, b, is

$$d = \sum_{i}^{360}\left(d_{h,i} * d_{m,i} * d_{v,i}\right) \quad (4)$$

$$d_{h,i} = \left|I_i^a - I_i^b\right| \quad (5)$$

$$d_{m,i} = \left\|M_i^a - M_i^b\right\| \quad (6)$$

$$d_{v,i} = \left|V_i^a - V_i^b\right| \quad (7)$$

where I is the hue histogram and M,V the vectors of the respective mean and variance.

Agent colorLAB

This agent also deals with color and in particular with the Lab color model. Lab was designed to be compatible with the human color perception model and linear in the color spectrum. These properties are suitable for image segmentation and retrieval. The LAB histograms are also calculated from equations (1) and (2). However, the three histograms here differ in their size and the range of values; the L component ranges form 0 to 100, while the a and b channels from –100 to +100.

Agent textureHistogram

This agent processes a combination of texture and color. More specifically, a modified color histogram [10] that favors high frequencies is used. This is done because texture usually presents an intense fluctuation of frequencies. In the modified histogram, equation (8), the square of the laplacian coefficient Δ, equation (9), provides the weights for each histogram element. As a result of applying the kernel k to the image, high frequency structures, like edges and corners, are enhanced.

$$I_{C,v} = \frac{1}{MN}\sum_{i,j} w(i,j)\delta\left(E_{i,j}^{C} - v\right) \rightarrow v \in [1, colors] \quad (8)$$

$$w(i,j) = \Delta^2(i,j) \quad (9)$$

$$k = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \quad (10)$$

The histogram dissimilarity is calculated according to equation (3).

Agent textureGabor

The second agent dealing with texture is based on the Gabor filters. It is believed that gabor filters resemble the responses of simple cells in the primary visual cortex V1. Gabor transformation uses a set of filters (filter bank) that is derived from a wavelet basis with dilations and rotations [2], [3], [18]. In this approach we use four different scales and six orientations that yield to a bank of 24 filters. The wavelet basis, equation (11), is a two dimensional complex function that generates the filters of the bank according to equation (12)

The import process is completed in three stages. First, the image is converted to grayscale mode and we construct the bank of filters. This conversion minimizes the computational cost and yields to minor loss of information since texture is almost independent of color [4]. Then, the image is convolved with the filters and a representative set of parameters is extracted for each one of the exported images. Each one of the convoluted images E_{mn} captures certain information about the texture of the source image in a specific scale and orientation.

$$g(x,y) = \left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left[-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right) + 2\pi jWx\right] \quad (11)$$

$$g_{mn}(x,y) = a^{-m}g(x',y'), \qquad a > 1, \qquad m,n \in \mathrm{Z} \quad (12)$$

$$x' = a^{-m}\left(x\cos\theta + y\sin\theta\right) \quad (13)$$

$$y' = a^{-m}\left(-x\sin\theta + y\cos\theta\right) \quad (14)$$

where $\theta = \frac{n\pi}{K}$ and K the number of orientations.

Finally, the image is segmented and a representative set of parameters for each segment is stored. Segmentation is achieved by dividing the image into non-overlapping tiles of 20x20 pixels and calculating texture measures. First, the mean μ_{mn} and variance σ_{mn} of the energy is computed for each tile and for every scale and orientation according to equations (15) and (16) resulting in a 24D feature vector for each tile.

$$\mu_{mn} = \overline{\sum_{i=k}^{k+20}\sum_{j=l}^{l+20}\left|E_{mn}(i,j)\right|} \tag{15}$$

$$\sigma_{mn} = \overline{\sum_{i=k}^{k+20}\sum_{j=l}^{l+20}\left|E_{mn}(i,j)-\mu_{mn}\right|} \tag{16}$$

Subsequently, the image tiles are classified into three different classes by using the *k means algorithm.* The extraction of the class's representative parameter vector is accomplished by the summation of the mean and variance of the contained tiles excluding border tiles since they usually consist of a mixture of textures. The distance between two images i, j containing c_1 and c_2 classes respectively, is defined as the smallest distance between their clusters:

$$d(i,j) = \min_{c_1 \times c_2} \Delta_{i,j}^{k}, \qquad k \in C_1 \times C_2 \tag{17}$$

$$\Delta_{i,j}^{k} = \sum_{m}\sum_{n}\left(\left|\frac{\mu_{mn}^{i,k}-\mu_{mn}^{j,k}}{a(\mu_{mn})}\right| + \left|\frac{\sigma_{mn}^{i,k}-\sigma_{mn}^{j,k}}{a(\sigma_{mn})}\right|\right) \tag{18}$$

where $\alpha(\mu)$, $\alpha(\sigma)$ are the variances of the respective parameters in the whole image gallery and they are used to normalize the two terms of the summation.

Agent shape

The last agent deals with shape. There have been a lot of proposed methodologies, [14], [16], for shape analysis and interpretation, depending on the application. Our approach is based on the *moment* theory. In 1962, Hu proved that moment-based shape description is information preserving. In addition, moments are mathematically concise, simple to implement and computationally inexpensive. A moment of order $p+q$ of a binary (0/1) image E with dimensions MxN is defined according to (19):

$$m_{pq} = \sum_{x=1}^{M}\sum_{y=1}^{N} x^p y^q E(x,y) \tag{19}$$

In the import process we first segment the image based on a region-growing algorithm [9] according to which we then compute the moments. The initial seeds for the growing are given by using Canny edge detection. To reduce redundancy in the detected seeds we conduct a refinement procedure through an iterative seed-elimination scheme. This method consists of a spatial and a color threshold, which are iteratively increased until an amount of seeds below a threshold is reached. Subsequently, the moments of order 0, 1 and 2 are computed. Moments of higher order are omitted because they do not contribute substantially and they are intolerant to noise. For the retrieval we compare every possible pair between the segmented shapes i, j of the two images as follows:

$$d = \min_{i,j}\left[\sqrt{\sum_{p=0}^{2}\sum_{q=0}^{2}\left(m_{pq}^{i}-m_{pq}^{j}\right)^2}\sqrt{\sum_{b=1}^{bands}\left(C_b^i - C_b^j\right)^2}\right] \tag{20}$$

where $C^i{}_b$ is the mean value of the shape's i pixel colors in band b.

Voting Methods

As mentioned in the introduction our aim is to improve the system's performance by modeling human perception mechanisms. However, the way in which specialized vision areas communicate remains a mystery. Therefore, for image retrieval we propose a fusion procedure based on a voting scheme. Voting is a well-established technique not only in social and political life, but also in computational systems [21]. In our case each agent, according to its specialization, votes for the similarity of the query image to images in a gallery. *Reasoning*, which was mentioned in the system's overview, describes the image scores for each agent with regard to the voting methodology applied. The selected voting plays an important role and has a great influence in the final result. Next, we present three voting methodologies that have been implemented.

Borda count method

The Borda count method belongs to the preferential or positional voting systems. Each agent ranks the candidate images according to their similarity with the query image. The grade of an image, for a specific agent, is equal to its position and its final rank comes up from the summation of the grades. The *reasoning* presents the image rank for each agent apart. There is also a variation of this method according to which a threshold grade is defined. If at least one of the image grades exceeds this threshold, it is excluded from the resulted list regardless of its final position.

Weighted voting method

The proposed weighted voting method is a combination of a utilitarian and a weighted system. The agent-voter does not only rank the candidate images but also gives a similarity-weight to them according to the expected benefit (utilitarian). This weight is computed exclusively by the agent's methodology being independent of the voting scheme. Each agent, moreover, votes with a different gravity-weight to the final result (weighted), which is user dependent.

If P is the amount of agents participating in retrieval and $\vec{a}$ the vector of the computed similarities for a specific image, the final sum S is given by:

$$S = \sum_{i}^{P} w_i \frac{a_i}{\sigma(a_i)} \tag{21}$$

where $\vec{w}$ is the agents' weight vector, the same for the whole image gallery, and $\sigma(\cdot)$ the variance of the respective values in the retrieval set, used to normalize the values. The

reasoning for a specific image presents the weights given to it by the agents.

Hierarchical voting method

The hierarchical voting method belongs to the candidates' successive elimination schemes. In this implementation however, the agents vote in a unique round all together in order to avoid the delay from the serialization of the procedure. The user defines the desired sequence of the eliminations and the system retains a predefined percentage of the images in each round. The *reasoning* presents the image ranking for all successive votes. A basic difference from the other methods is a resulted list that contains a subset of the initial image gallery. It's an easily manipulated method since the parameterization of the voting sequence can alter substantially the final result.

RESULTS

The evaluation of image retrieval systems is still a hard task because of the particularities that image data present [8], [17]. The images are usually copyright protected and thus very difficult for a rich and free collection to be established in the research community. Regarding the relevance judgment subjectivity is an issue because of the absence of real users. As in most cases, we deal with these problems by using a commercial image collection with a rich content and a predefined classification. In this evaluation the experimental results come from the Corel 7 Photo CD collection with 1000 groundtruth images. Regarding the system's overall performance there are a lot of metrics each one focusing on specific aspects of the retrieval including the *rank of the best match*, the *average rank of relevant image*, the *precision* and the *recall* [7]. Precision *P* and recall *R* are standard measures in Information Retrieval and suitable for CBIR as well:

$$P = \frac{\text{Number of relevant images retrieved}}{\text{Total number of images retrieved}}$$

$$R = \frac{\text{Number of relevant images retrieved}}{\text{Total number of relevant images}}$$

However, either value alone contains insufficient information since we can always make *recall* 1 or keep *precision* high. Thus, they should be used together or the number of the retrieved images should be specified. An alternative to this is a *precision versus recall graph*. PR graphs contain a lot of information and their long use makes them easily read by researchers. Next we present some of the experimental results along with the respective PR graphs when possible.

The first of our retrieval experiments is based on the Borda count method. Figure 2(a) presents the query image *windsurfing board* at the left, and next to it the first four retrieval results from a database containing 30 relevant images.



Figure 2: Retrieval with methods *borda* and *weights*

The next two experiments regard the weighted voting method. Figure 2(b,c) presents two different query images, *sunset* and *flower* at the left of the respective rows and next to them the first four results. The *sunset* has 40 relevant images in the database while the *flower* 20. This method is very flexible since the user can customize the weights and the idea of multiple weights adjustable to the desired behavior, is not only promising, but biologically relevant as well.

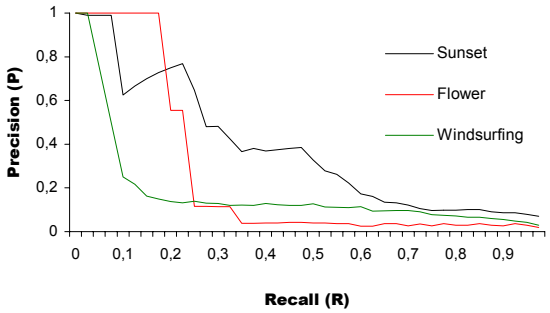


Figure 3: Joint PR graph

Figure 3 depicts a joint PR graph for the overall retrieval of the above experiments. Regarding the *sunset* we see a regular decrease of the system performance. The *precision* is above 0.5 until the *recall* value of 0.3, which means that in the first 24 retrieved images half of them were relevant. Regarding the *flower,* the system got 5 hits in the row in the first retrieved images.

Figure 4 shows the retrieval results for the same query image using three different voting sequences of the hierarchical voting method. By defining the appropriate percentage of elimination for each voting round we can lead to the desired number of results. The fact that only two of the images are present in every result is indicative of the voting sequence significance. The total number of sequence combinations makes the selection of the appropriate one difficult and the evaluation of the overall system performance extremely hard. Furthermore, the inapplicability of a PR graph, since this method returns a subset of the gallery, renders the direct interpretation of the results even harder.



Figure 4: Hierarchical voting retrieval

CONCLUSIONS AND FUTURE WORK

In this paper we examined the issue of content-based image retrieval and we presented the implementation of a corresponding platform based on a biologically inspired multi-agent architecture. The experimental results are indicative of the many possibilities and promising for further work. The proposed architecture enables the system to be versatile and efficient in diverse image retrieval applications. In our current work in content-based medical image retrieval, the system has been extended with the addition of more agents.

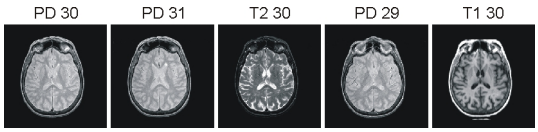


Figure 5: Medical image retrieval

Figure 5 shows a preliminary result; the query image (left) is a slice of an MR PD volume of a normal brain (slice 30) and from a multi-modality database of brain images the system successfully retrieves the closest images of the same brain, regardless of their MR modality. Preliminary results in this application domain are very encouraging and our current aim is to develop an efficient and possibly intelligent content-based medical image retrieval system

REFERENCES

[1] A. Boucher, A. Doisy, X. Ronot and C. Garbay, "A Society of Goal-Oriented Agents for the Analysis of Living Cells", *Artificial Intelligence in Medicine, 14(1,2):183-200, 1998*

[2] B. S. Manjunath and W. Y. Ma, "Texture Features for Browsing and Retrieval of Image Data", *IEEE Trans. on Pat. Analysis and Machine Intelligence, vol.18, No.8, Aug. 1996*

[3] B. S. Manjunath, P. Wu, S. Newsam, H. D. Shin, "A texture descriptor for browsing and similarity retrieval", *Signal Processing: Image Communication 2000*

[4] C. Palm, D. Keysers, T. Lehmann, K. Spitzer, "Gabor Filtering of Complex Hue/Saturation Images for Color Texture Classification"

[5] E. Rosch, "Natural categories", *Cognitive Psychology, 1973, 4:328—350*

[6] E. Y. Chang, Beitao Li and Chen Li, "Toward Perception-Based Image Retrieval", *UCSB Technical Report, Feb. 2000*

[7] H. Muller, W. Muller, D. M. Squire, S. Marchand-Maillet and T. Pun, "Performance evaluation in content-based image retrieval: Overview and proposals", *Pattern Recognition Letters, 2001*

[8] H. Muller, W. Muller, S. Marchand-Maillet, D. M. Squire and T. Pun, "Automated benchmarking in content-based image retrieval", *International Conference on Multimedia and Exposition, ICME 2001, Tokyo, Japan, 2001*

[9] J. Fan, D. K. Y. Yau, A. K. Elmagarmid and W. G. Aref, "Automatic Image Segmentation by Integrating Color-Edge Extraction and Seeded Region Growing", *IEEE Trans. on Image Processing, Vol. 10, No. 10, October 2001*

[10] N. Boujemaa and C. Vertan, "Integrated color texture signature for image retrieval"

[11] Pattie Maes, "Modeling Adaptive Autonomous Agents", *Artificial Life Journal, Vol. 1, pp. 135-162, MIT Press, 1994*

[12] R. Pfeifer and C. Scheier, "Understanding Intelligence", *MIT Press, 1999*

[13] Remco C. Veltkamp, Mirela Tanase and Danielle Sent, "Features in Content Based Image retrieval Systems: A Survey", *Kluwer Academic Publishers, 2001*

[14] Remco C. Veltkamp and Michael Hagedoorn, "State of the Art in Shape Matching", *Technical Report* UU-CS-1999-27, Utrecht University, the Netherlands, 1999

[15] Semir Zeki, "A Vision of the Brain", *Blackwell Scientific Publications, 1994*

[16] Sven Loncaric, "A survey of shape analysis techniques", *Pattern Recognition,* (31)8:983-1001, 1998.

[17] W. Muller, S. Marchand-Maillet, H. Muller and T. Pun, "Towards a fair benchmark for image browsers", *In SPIE Photonics East, Voice, Video and Data Communications, Boston, MA, USA, November 5-8 2000*

[18] Wei-Ying Ma and B. S. Manjunath, "A Texture Thesaurus for Browsing Large Aerial Photographs", *Journal of the American Society for Information Science, 1998*

[19] Xenofon Zabulis, "Perceptually Relevant Mechanisms for the Description and Retrieval of Visual Information", *PhD. Thesis, University of Crete, 2002*

[20] Y. Rubner, J. Puzicha, C. Tomasi and J. M. Buhmann, "Empirical Evaluation of Dissimilarity Measures for Color and Texture", *Comp. Vision and Image Understanding, 2001*

[21] G. Halkiadakis, "Agent Architecture for a Voting System", *MSc Thesis, University of Crete, 1999*