Content-Based Image Retrieval using Cellular Automata

Konstantinos Konstantinidis, Georgios Ch. Sirakoulis and Ioannis Andreadis

Laboratory of Electronics
Dept. of Electrical and Computer Engineering,
Democritus University of Thrace
Xanthi 67100, Greece

{konkonst, gsirak, iandread}@ee.duth.gr

Abstract **– Content-based Image Retrieval (CBIR) is generally known as a collection of techniques for retrieving images on the basis of features, such as color, texture and shape. An efficient tool in CBIR is that of image histograms. In this paper a new image retrieval method is proposed with the use of histograms in conjunction with cellular automata (CAs). The main thrust of this paper is the classification of the images in the database by CAs and the retrieval of the desired images by a simple histogram extracted from the hue component of the HSV color space. Moreover, because of the CAs local rule simplicity, the VLSI implementation of the proposed CA algorithm is straightforward.**

Index Terms – Image Retrieval, Histograms, Cellular Automata.

I. INTRODUCTION

A. *Image Retrieval*

Color is one of the most important features in human vision. It allows the performance of complex tasks such as the discrimination between objects with similar shape characteristics but different color features, the tracking of moving objects, as well as the scene property analysis. In artificial vision the use of color information has been increased significantly during the past years. A very useful tool in color image analysis is the histogram [1]: a global statistical feature which describes the color distribution for a given image. A histogram can be created by firstly dividing a color space into a number of bins and then by counting the number of pixels of the image that belong to each bin. The wide use of histograms in color image analysis is due to their rotation and scaling invariance and their relatively moderate computation cost. On the other hand their downfall is that they are also quite unreliable as they are sensitive to even small changes in the scene of the image. In most color image processing methods, histograms consist of three components in respect to the three components of the color space used. [2]

Among the applications of color histograms is that of image retrieval. Research in color imaging, from sensors to databases, has recently emerged to a number of different applications, including military, industrial and civilian which generate gigabytes of color images per day. Thus, the necessity of organizing this huge information has grown strong [2], which means that in order to allow efficient browsing, appropriate indexing should be available as in keyword searches of text databases. One of the most popular methods used for retrieval is that of query-by-example (QbE), which means that the user has to present an image to the system and the latter searches for others alike by extracting features from the query image and comparing them to the ones stored in the image database. The extraction of meaningful features is critical in Content-based Image Retrieval (CBIR) and therefore an open and active field of research. The features mostly used by researchers for indexing and retrieval are color [3, 4, 5, 6, 7], texture [3, 4, 8], and shape [3, 4, 9]. The principal components of a histogram based retrieval system are: (i) An appropriate color space, such as HSV, L*a*b* or L*u*v*, (ii) a histogram representation, such as straightforward [1, 5, 7] or fuzzy [6] histograms and (iii) a similarity metric, like the Euclidean, the histogram intersection method [7] or the Bhattacharyya distance [10]. In order for the system to perform effectively, the number of regions that the color space is divided into is quite large and thus the colors represented by neighboring regions have relatively small differences. As a result, images which are similar to each other but have small differences in scene or contain noise will produce very dissimilar histograms and vice versa. In order to present a solution to this problem a new method for color image retrieval is introduced for the first time, which classifies the images from the database a priori using cellular automata on the a* and b* components of the L*a*b* color space, and then a single straightforward histogram extracted from the hue component of the HSV color space is used to produce the number of images requested by the user. The inherent parallelism of the proposed CA methodology and its easy VLSI implementation make it suitable for real-time applications in the fields of remote sensing images, computer vision applications, industrial engineering, pattern recognition, etc.

B. *Cellular Automata*

CAs were first introduced by von Neumann [11] in 1948, in an ambitious project: to show that complex phenomena can in principle be reduced to the dynamics of many identical, very simple primitives, capable of interacting and maintaining their identity. Following a suggestion by Ulam, von Neumann adopted a fully discrete approach, in which space, time, and even the dynamical variables were defined to be discrete. CAs are models of physical systems, where space and time are discrete and interactions are local [11]. In this subsection a more formal definition of a CA will be presented [12]. In general, a CA requires:

(i) a regular lattice of cells covering a portion of a *d*-dimensional space;

(ii) a set $\mathbf{C}(\vec{r},t)=\{C_1(\vec{r},t),C_2(\vec{r},t),\ldots,C_m(\vec{r},t)\}$ of variables attached to each site $\vec{r}$ of the lattice giving the local state of each cell at the time $t = 0, 1, 2, \ldots$;

(iii) a rule $R=\{R1, R2, \ldots, Rm\}$ which specifies the time evolution of the states $\boldsymbol{C}(\vec{r},t)$ in the following way:

$$C_j(\vec{r},t+1)=R_j\left(\boldsymbol{C}(\vec{r},t),\boldsymbol{C}(\vec{r}+\vec{\delta}_1,t),\boldsymbol{C}(\vec{r}+\vec{\delta}_2,t),\ldots,\boldsymbol{C}(\vec{r}+\vec{\delta}_q,t)\right) \quad (1)$$

where $\vec{r}+\overrightarrow{\delta_k}$ designate the cells belonging to a given neighborhood of cell $\vec{r}$. In the proposed method the cells take the form of pixels.

In the above definition, the rule *R* is identical for all sites, and is applied simultaneously to each of them, leading to synchronous dynamics. It is important to notice that the rule is homogeneous, i.e. it does not depend explicitly on the cell position $\vec{r}$. However, spatial (or even temporal) inhomogeneities can be introduced by having some $C_j(\vec{r})$ systematically at 1, in some given locations of the lattice, to mark particular cells for which a different rule applies. Furthermore, in the above definition, the new state at time t+1 is only a function of the previous state at time *t*. It is sometimes necessary to have a longer memory and introduce a dependence on the states at time t-1, t-2,…, t-k. Such a situation is already included in the definition, if one keeps a copy of the previous state in the current state.

The neighborhood of cell $\vec{r}$ is the spatial region in which a cell needs to search in its vicinity. In principle, there is no restriction on the size of the neighborhood, except that it is the same for all cells. However, in practice, it is often made up of adjacent cells only. For two-dimensional CA, two neighborhoods are often considered: The von Neumann neighborhood, which consists of a central cell (the one which is to be updated) and its four geographical neighbors north, west, south and east. The Moore neighborhood contains, in addition, second nearest neighbors northeast, northwest, southeast and southwest, which is a total of nine cells. The von Neumann and Moore neighborhoods are shown on Figs. 1(a) and 1(b) respectively. In practice, when simulating a given CA rule, it is impossible to deal with an infinite lattice. The system must be finite and have boundaries. Clearly, a site belonging to the lattice boundary does not have the same neighborhood as other internal sites. In order to define the behavior of these sites, the neighborhood is extending for the boundary sites. Neighborhood extension leads to various types of boundary conditions such as periodic (or cyclic), fixed, adiabatic or reflection [12].

CAs have sufficient expressive dynamics to represent phenomena of arbitrary complexity, and at the same time can be simulated exactly by digital computers, because of their intrinsic discreteness, i.e. the topology of the simulated object is reproduced in the simulating device [13]. The CA approach is consistent with the modern notion of unified space-time. In computer science, space corresponds to memory and time to processing unit. In CAs, memory (CA cell state) and processing unit (CA local rule) are inseparably related to a CA cell. Furthermore, CAs are an alternative to partial differential equations [12, 14-16], and they can easily handle complicated boundary and initial conditions, inhomogeneities and anisotropies. In addition, algorithms based on CAs run quickly on digital computers [17]. Models based on CAs lead to algorithms which are fast when implemented on serial computers, because they exploit the inherent parallelism of the CA structure [18-20]. Moreover, due to the local interconnections and the discreteness in space and time, synchronous very large scale integration (VLSI) circuits have been used as an implementation medium of algorithms based on CAs [21].

CAs have been extensively used as a VLSI architecture, since the CA architecture offers a number of advantages and beneficial features such as simplicity, regularity, and locality of interconnections. More specifically, there are four main factors that determine the cost/performance ratio of an integrated circuit; namely, circuit design and layout, ease of mask generation, silicon-area utilization and maximization of achievable clock speed. For a given technology, the latter is inversely proportional to the maximum length of critical signal paths. In terms of these four parameters, CAs are perhaps the computational structures best suited for a VLSI realization. CAs as a VLSI architecture have been applied, among others, to image processing [22-24], byte error correcting codes [25], classification [26] and as pseudorandom number generators [27]. Special computing machines have also been developed based on the CA architecture [18-19] and, furthermore, special Cellular Automata algorithms have been implemented on massively parallel computers, such as the Cellular Automaton Machine (CAM) [18-19, 28].

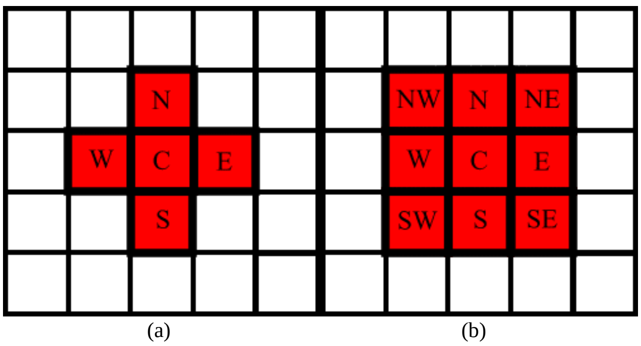

Fig. 1 The von Neumann (a) and Moore (b) neighborhoods used in CAs.

II. THE CELLULAR-BASED COLOR IMAGE RETRIEVAL SYSTEM

The use of global histograms for image retrieval has proven to be an efficient and robust retrieval method [5, 6, 7], as it describes the overall statistics of the color in the images and is insensitive to rotation and scaling of the images themselves. However, such techniques lack in cases where the images have similar colors, but are spatially distributed differently. This problem can be overleaped with the use of local histograms (Splitting each image into smaller regions) [5]. These on the other hand suffer a severe lack of speed due to the rapid increase in computational burden which results from the repetitiveness needed to produce local histograms. This leads to the need of adoption of global histograms with embedded local characteristics, such as the CA histogram. In a CA the future value of the central pixel always depends on the current values of the surrounding pixels in its immediate (von Neumann) or second (Moore) neighborhood. The proposed histogram creation method has a straightforward algorithm in which the a* and b* components, from the L*a*b* color space, and hue, from the HSV color space, are used. One of the reasons why the L*a*b* color space was selected is that it is a perceptually uniform color space which approximates the way that humans perceive color. However, the main reason is that L*a*b* was found to perform better than other color spaces in various retrieval tests performed in the laboratory for this exact purpose [5]. In L*a*b*, L* stands for Luminance and has a range of [0,100], a* represents relative greenness-redness and b* represents relative blueness-yellowness; both having a range of [-128, 127]. All colors and grey levels can be expressed throughout a combination of the three components. On the other hand, the HSV color space was selected as it also reflects human vision quite accurately. Nevertheless, the major reason for its selection is because it mainly uses only one of its components (hue) to describe color in an image. The other two components, i.e. saturation and value are significant mostly when describing black, white, gray and various shades of colors. The HSV color space can be represented as a round cone in which Hue is the angle, ranging from [0^o, 360^o], where 0^o and 360^o degrees feature the color red.

In the first step of the proposed method, a Moore type CA is applied to each of the a* and b* color space components of all the images in the database. The pixels from the second (Moore) neighborhood are all added to the occasional central pixel for five epochs or until all the pixels in the image assume either of the two absolute highest values of the a*b* components (|-128| and |127|). The number of five epochs was selected through several, accuracy versus time, tests, after which the value above was found to be the optimal one to ensure great accuracy but also great speed. At the end of each of these epochs, every pixel's new value is compared to its old one in order to count the number of pixels whose value is still changing. This number is stored in a vector whose final size depends on the number of epochs ran for the CA to come to a stop and is considered to be the CA histogram featuring the number of shifting pixels during each epoch; ergo, each bin of the histogram contains the number of changing pixels for each epoch. If the number of pixels changing is large this means that most neighborhoods of pixels in the occasional image have small values.

The final stage of the system is that of the actual retrieval of the images. Following the CA histogram extraction of all the images in the database, each of these double histograms, since two color space component are considered, are compared to the ones extracted from the query image itself with the use of the Bhattacharyya [10] distance and only the first 100 most similar images are taken into account. Thus it is fair to say that the method described above acts as a pre-classifier for the final part to follow. Hence, from each of these indexed images, another histogram of 32 bins, is extracted, using the hue (H) component of the HSV color space which is compared to the respective one extracted from the query image again with Bhattacharyya distance through which the most similar images (the number is selected by the user) are presented. The Bhattacharyya distance which was selected subsequent to extensive tests and simulations, measures the statistical separability of spectral classes; giving an estimate of the probability of an accurate classification:

$$B(H_Q, H_C) = -\ln \sum_i \sqrt{H_Q(i) \times H_C(i)} \qquad (2)$$

where H_Q and H_C are the histograms of the sought image and the query image, respectively.

The Bhattacharyya distance does not take into account zero histogram entries. For highly structured histograms (i.e. those, that are not uniformly populated), this can lead to the selection of matches in which there is a strong congruence between the structure of query and data–base histogram.

III. EXPERIMENTAL RESULTS

The performance of the proposed method was tested through the retrieval of various images sets from a collection of 1040 images (Image sets 1 and 5 in Figs. 2 and 3 respectively are representative of the database), some selected from different sites on the internet, others scanned from personal photographs and a large amount of images taken with several different digital cameras. The images are online, available at the following URL: http://utopia.duth.gr/~konkonst. The images in the collection are representative for the general requirements of an image retrieval system. The range of topics presented is quite wide and varies from natural scenes, such as landscapes, sport events, concerts, etc; to artificial computer graphics, that tend to confuse image retrieval systems. The experiments were all ran on Mathworks' Matlab R14 on a Pentium 4 processor with 1 GB of RAM. All the images were scaled to a 50x50 pixel size using the nearest-neighbor method in order to make the algorithm faster and to avoid later normalization of the histograms resulting in loss of color quantity information.

The retrieval outcome is presented through a query session which produces 20 images (the figures of the image sets 1 and 5 are shown below in Figs. 2 and 3) ranked in similarity according to the value produced by the metric. The smaller the number that the metric produces, the higher the similarity of that specific image is. Based on the diversity which exists in the image database the ranking of the Bhattacharyya distance value can be considered as quite an objective criterion to compare the query image to a random image in the database. In the first image set (Fig. 2) the general concentration of red is very high; nonetheless the system does not confuse the images due to the difference in the spatial concentration distribution of the particular color as well as the variations of the rest of the colors in the two images thus proving its robustness.

In addition to the system described above, another system was implemented for means of comparison. Straightforward histograms of 32 bins resulting from the hue component of the HSV color space were created for each image in the database and the Bhattacharyya distance was used in order to compare them to the one extracted from the query image.

The measurement used in order to present the performance of the method is the retrieval performance percentage, which is the percentage of actual similar images produced in the 20 first most similar images retrieved by the system. The numerical comparison of six image sets retrieved by the proposed method and the simple hue histogram system is presented through the retrieval precision on Table 1. The proposed method accuracy spans from a low precision percentage of 80 percent to the highest 100 percent stating the clear improvement in precision of the proposed method over the simple hue one which ranges from 60 to 85 percent as shown in Table 1. In addition to the precision perspective, another aspect of retrieval performance is presented by the graph in Fig. 4: precision versus recall [29].

Precision is the proportion of relevant images retrieved R (similar to the query image) in respect to the total retrieved A, whereas recall is the proportion of similar images retrieved in respect to the similar images that exist.

Precision=Similar Retrieved/Total Retrieved= $|A \cap R| / A$

Recall = Similar Retrieved / Similar Exist = $|A \cap R| / R$

Generally, precision and recall are used together in order to point out the change of the precision in respect to recall. In most typical systems the precision drops as recall increases, hence, in order for an image retrieval system to be considered effective the precision values must be higher than the same recall ones, which is the case in the proposed method. For example, when retrieving the second image set the system failed only after having produced 12 correct images and still managed to produce similar images from that particular fail point to the end.

In order to further test the robustness of the compared systems four more tasks were executed. For the first task salt and pepper noise of density 0.15 was inserted to the query images, then the lightness of the images was initially increased and then decreased, moreover the images were blurred by a filter which approximates the linear motion of a camera by *len* (31) pixels, with an angle of *theta* (11^o) degrees in a counter clockwise direction, and last the images were rotated by 45^o clockwise. The outcome of the tests proved that although the query images were severely altered, the precision percentages of the proposed method were not decreased at the slightest and the images were retrieved successfully; thus demonstrating that the algorithm presented is robust even to extreme changes in the images in contradiction to the simple HSV histogram whose results of retrieval percentage were decreased even more.

TABLE I
FILTERING PERFORMANCE (PRECISION) WITH TIME

Image Sets	1	2	3	4	5	6	Time (seconds)
Proposed Method Performance (%)	100	95	95	80	100	95	256
Simple Hue Histogram Performance (%)	85	75	80	60	85	80	65

Fig. 2 The 20 retrieved images from data set 1. All the images are rightfully retrieved. The first image is also the query image; the images are presented in descending score from left to right and from top to bottom.

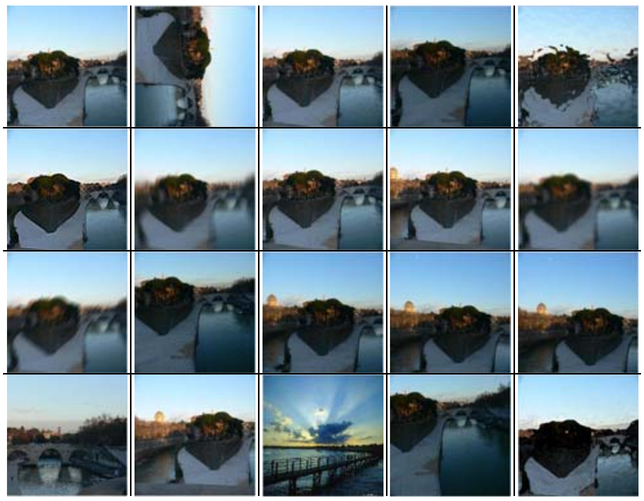

Fig. 3 The 20 retrieved images from data set 5. All the images are rightfully retrieved. The first image is also the query image; the images are presented in descending score from left to right and from top to bottom.

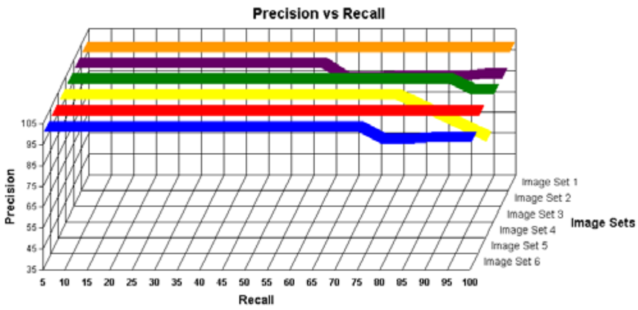

Fig. 4 Comparison of the 6 image sets. Precision against Recall

IV. Conclusions

A new color histogram creation method was proposed for the first time. This method produces histograms whose class contains information, not only on the global color distribution over the image, but on its local concentration as well. The histograms are created based on the a* and b* components of the L*a*b* color space and are acquired through CAs. The resultant CA histograms are suitable for content based image retrieval utilizing color characteristics. A CBIR system employing the proposed histograms was implemented and assessed. The system in hand was compared to a simple HSV histogram-based image retrieval system proving to be much more accurate and robust through several image retrieval tests. Target applications of the proposed system include internet queries, retrieval of remote sensing images and relative computer vision applications. Finally, on account of the assumed CA's local rule simplicity, its VLSI implementation is straightforward. This VLSI implementation lead to dedicated CA processors that execute the CA algorithms, and can be designed using commercially available VLSI CAD tool systems.

References

[1] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Reading, Massachusetts: Addison-Wesley, 1992.

[2] I. Gagliardi and R. Schettini, "A method for the automatic indexing of color images for effective image retrieval," *The New Review of Hypermedia and Multimedia*, vol. 3, pp. 201-224, 1997.

[3] V. Castelli and L. D. Bergman, *Image Databases: Search and Retrieval of Digital Imagery*, John Wiley & Sons, Inc., 2001.

[4] A. Del Bimbo, *Visual Information Retrieval*, San Francisco, California: Morgan Kaufman Publ., 1999.

[5] K. Konstantinidis and I. Andreadis, "Performance and computational burden of histogram based color image retrieval techniques," *Journal of computational methods in sciences and eng.*, in press.

[6] K. Konstantinidis, A. Gasteratos and I. Andreadis, "Image retrieval based on fuzzy color histogram processing," *Optics Communications*, vol. 248, pp. 375-386, 2005.

[7] M.J. Swain and D.H. Ballard, "Color Indexing, International," *Journal of Computer Vision*, vol. 7, pp. 11-32, 1991.

[8] A. Gasteratos, P. Zafeiridis and I. Andreadis, "An Intelligent System for Aerial Image Retrieval and Classification," in Lecture Notes in Artificial Intelligence, vol. 3025, G.A. Vouros and T. Panayiotopoulos, Eds., Berlin Heidelberg: Springer-Verlag, 2004, pp. 63-71..

[9] A.K. Jain and A. Vailaya, "Image Retrieval Using Color and Shape," *Pattern Recognition*, vol. 29, pp. 1233-1244, 1996.

[10] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed., Academic Press, Inc, 1990.

[11] J. von Neumann, *Theory of Self-Reproducing Automata*, University of Illinois, Urbana, 1966.

[12] B. Chopard and M. Droz, *Cellular Automata Modeling of Physical Systems*, Cambridge University Press, Cambridge, 1998.

[13] S. Wolfram, *Theory and Applications of Cellular Automata*, World Scientific, Singapore, 1986.

[14] T. Toffoli, "Cellular automata as an alternative to (rather than an approximation of) differential equations in Modeling Physics," *Physica D*, vol. 10, no. 1-2, pp. 117-127, 1984.

[15] I. Bialynicki-Birula, "Weyl, Dirac, and Maxwell equations on a lattice as unitary cellular automata," *Physical Review D*, vol. 49, pp. 6920-6927, 1994.

[16] M.G. Danikas, I. Karafyllidis, A. Thanailakis and A.M. Bruning, "Simulation of electrical tree growth in solid dielectrics containing voids of arbitrary shape," *Modelling Simulation Materials Sci. Eng.*, vol. 4, pp. 535-552, 1996.

[17] G.Ch. Sirakoulis, I. Karafyllidis, A. Thanailakis and V. Mardiris, "A methodology for VLSI implementation of Cellular Automata algorithms using VHDL," *Adv. Eng. Soft.*, vol 32, pp. 189-202, 2001.

[18] T. Toffoli and N. Margolus, *Cellular Automata Machines: A New Environment for Modeling*, MIT Press, MA, Boston, 1987.

[19] M. Sipper, *Evolution of Parallel Cellular Machines, The Cellular Programming Approach*, Springer, Berlin, 1997.

[20] G.Ch. Sirakoulis, "A TCAD system for vlsi implementation of the CVD process using VHDL," *Integration, the VLSI Journal*, vol. 37, pp. 63-81, 2004.

[21] G.Ch. Sirakoulis, I. Karafyllidis and A. Thanailakis, "A CAD system for the constructions and VLSI implementation of Cellular Automata algorithms using VHDL," *Microprocessors and Microsystems*, vol. 27, pp. 381-396, 2003.

[22] I. Andreadis, I. Karafyllidis, P. Tzionas, A. Thanailakis and Ph. Tsalides, "A New Hardware Module for Automated Visual Inspection Based on a Cellular Automaton Architecture," *Journal of Intelligent and Robotic Systems*, vol. 16, pp. 89-102, 1996.

[23] P. Tzionas, Ph. Tsalides and A. Thanailakis, "A new cellular automaton-based nearest neighbor pattern classifier and its VLSI implementation," IEEE Trans. VLSI Syst., vol. 2, pp. 343-353 1994).

[24] P. Tzionas,, "A cellular automaton processor for line and corner detection in gray-scale images," *Real-Time Imaging*, vol. 6, no. 6, pp. 461-470, 2000.

[25] D.R. Chowdhury, I.S. Gupta and P.P. Chaudhuri, "Cellular Automata Based Byte Error Correcting Code," *IEEE Trans. Comp.*, vol. 44, pp. 371-382, 1995.

[26] P. Tzionas, P. Tsalides and A. Thanailakis, "A design and VLSI Implementation of a pattern classifier using pseudo-2d cellular Automata," *IEE PROC-G*, vol. 139, no. 6, pp. 661-668, 1992.

[27] Ph. Tsalides, T.A. York and A. Thanailakis, "Pseudo-Random number generators for VLSI Systems based on linear cellular automata," *IEE Proc. E. Comput. Digit. Tech.*, vol. 138, no. 4, pp. 241-249, 1991.

[28] H. de Garis, "CAM-Brain: the genetic programming of an artificial brain which grows/evolves at electronic speeds in a cellular automata machine," in *Proc. of the 1st IEEE Conference on Evolutionary Computation*, vol. 1, pp. 337-339, 1994.

[29] H. Muller, W. Muller, D. McG. Squire and S. Marchand-Maillet, "Performance Evaluation in Content-Based Image Retrieval: Overview and Proposals," *Pattern Recognition Letters*, vol. 22, no. 5, pp. 593-601, 2001.