2010

Accurate Image Retrieval Based on Compact Composite Descriptors and Relevance Feedback Information

Chatzichristofis, Savvas A.

World Scientific Publishing Company

International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 24, No. 2 (2010) 207–244
© World Scientific Publishing Company
DOI: 10.1142/S0218001410007890

World Scientific
www.worldscientific.com

ACCURATE IMAGE RETRIEVAL BASED ON COMPACT COMPOSITE DESCRIPTORS AND RELEVANCE FEEDBACK INFORMATION

SAVVAS A. CHATZICHRISTOFIS*, KONSTANTINOS ZAGORIS†,
YIANNIS S. BOUTALIS‡ and NIKOS PAPAMARKOS§

Department of Electrical and Computer Engineering
Democritus University of Thrace
12. Vas. Sofias, Xanthi, 67100, Greece
*schatzic@ee.duth.gr
†kzagoris@ee.duth.gr
‡ybout@ee.duth.gr
§papamark@ee.duth.gr

In this paper a new set of descriptors appropriate for image indexing and retrieval is proposed. The proposed descriptors address the tremendously increased need for efficient content-based image retrieval (CBIR) in many application areas such as the Internet, biomedicine, commerce and education. These applications commonly store image information in large image databases where the image information cannot be accessed or used unless the database is organized to allow efficient storage, browsing and retrieval. To be applicable in the design of large image databases, the proposed descriptors are compact, with the smallest requiring only 23 bytes per image. The proposed descriptors' structure combines color and texture information which are extracted using fuzzy approaches. To evaluate the performance of the proposed descriptors, the objective Average Normalized Modified Retrieval Rank (ANMRR) is used. Experiments conducted on five benchmarking image databases demonstrate the effectiveness of the proposed descriptors in outperforming other state-of-the-art descriptors. Also, a Auto Relevance Feedback (ARF) technique is introduced which is based on the proposed descriptors. This technique readjusts the initial retrieval results based on user preferences improving the retrieval score significantly. An online demo of the image retrieval system img(Anaktisi) that implements the proposed descriptors can be found at http://www.anaktisi.net.

Keywords: Image retrieval; image indexing; compact descriptors; low level features; color and texture histogram; relevance feedback; fuzzy techniques.

1. Introduction

The rapid growth of digital images through the widespread popularization of computers and the Internet makes the development of an efficient image retrieval technique imperative. Content-based image retrieval, known as CBIR, extracts several features that describe the content of the image, mapping the visual content of the images into a new space called the feature space. The feature space values for a given image are stored in a descriptor that can be used for retrieving similar images. The key to a successful retrieval system is to choose the right features that represent the images as accurately and uniquely as possible. The features chosen have to be discriminative and sufficient in describing the objects present in the image. To achieve these goals, CBIR systems use three basic types of features: color features, texture features and shape features. It is very difficult to achieve satisfactory retrieval results using only one of these feature types. To date, many proposed retrieval techniques adopt methods in which more than one feature type are involved. For instance, color, texture and shape features are used in both IBM's QBIC[14] and MIT's Photobook.[39] QBIC uses color histograms, a moment-based shape feature, and a texture descriptor. Photobook uses appearance features, texture features and 2D shape features. Other state-of-the-art CBIR systems include SIMBA,[45] CIRES,[22] SIMPLIcity,[49] IRMA,[29] FIRE,[8] MIRROR,[51] and also those in Refs. 21, 28 and 47. A cumulative body of research presents extraction methods for these feature types.

In most retrieval systems that combine two or more feature types, such as color and texture, independent vectors are used to describe each kind of information. It is possible to achieve very good retrieval scores by increasing the size of the descriptors, but this technique has several drawbacks. If the descriptor has hundreds or even thousands of bins, it may be of no practical use because the retrieval procedure is significantly delayed. Also, increasing the size of the descriptor increases the storage requirements which may have a significant penalty for databases that contain millions of images. Many presented methods limit the length of the descriptor to a small number of bins,[10,27] leaving the possible factor values in decimal, non-quantized form.

The Moving Picture Experts Group (MPEG) defines a standard for content-based access to multimedia data in their MPEG-7 standard.[23,35] This standard identifies a set of image descriptors that maintain a balance between the size of the feature and the quality of the retrieval results.[12,13,15,30,51]

In this paper a new set of descriptors is proposed and a method for their implementation in a retrieval system is described. The proposed descriptors have been designed with particular attention to their size and storage requirements, keeping them as small as possible without compromising their discriminating ability. The proposed descriptors incorporate color and texture information into one histogram while keeping their sizes between 23 and 74 bytes per image. The experimental results show that the performance of the proposed descriptors is better than the performance of the similarly-sized MPEG-7 descriptors.

The rest of the paper is organized as follows: Section 2 describes a novel technique for color information extraction. The technique employs a set of fuzzy rules to extract a fuzzy-linking histogram in the HSV color space. A three-input fuzzy system employs 20 rules to generate a ten-bin quantized histogram where each bin corresponds to a preset color. The number of pixels assigned to each bin is stored in a feature vector. In an optional second step, a two-input fuzzy system uses four new rules to transform the ten-bin histogram into a 24-bin histogram, extracting information related to the hue of each color.

In Sec. 3, two novel techniques are proposed for texture information extraction. The first one uses coefficients from the high-frequency bands derived from the Haar Wavelet transform,[49] creating an eight-bin histogram. The second technique employs the five digital filters proposed by the MPEG-7 Edge Histogram Descriptor[6] creating a six-bin histogram. In both methods each bin corresponds to a preset texture form.

Section 4 describes in detail how the systems are combined to produce the proposed descriptors. Section 5 demonstrates the reduction of the proposed descriptors' storage requirements by using the Gustafson Kessel[18] fuzzy classifier to quantize and map the values of the proposed features from the real number space $[0, 1]$ to the integer interval space $[0, 7]$.

Section 6 contains the experimental results of an image retrieval system that uses either the proposed features or the MPEG-7 features on five benchmarking databases. The objective measure ANMRR (Averaged Normalized Modified Retrieval Rank)[35] is used to evaluate the system performance and compare the proposed descriptors to the MPEG-7 standard descriptors.

In Sec. 7, an Auto Relevance Feedback (ARF) technique is introduced which is based on the proposed descriptors. This technique readjusts the initial retrieval results based on user preferences improving the retrieval score significantly. Finally, the conclusions are given in Sec. 8.

2. Color Information Extraction

Color is a low level feature that is widely used in Content Based Image Retrieval systems. Several approaches have been used to describe the color information that appears in the images. In most cases, color histograms are used, which on the one hand are easily extracted from the images and, on the other hand, present independency on some distortions such as rotation and scaling.[37]

An easy way to extract color features from an image is by linking the color space channels. `Linking` is defined as the combination of more than one histogram to a single one. One example is the Scalable Color Descriptor (SCD)[36] demonstrated in MPEG-7.[35] In the SCD implementation, the HSV color space is uniformly quantized into a total of 256 bins defined by 16 levels in H (Hue), four levels in S (Saturation) and four levels in V (Value). The values of H, S and V are calculated for every pixel and are then linearly quantized in the ranges $[0, 15]$, $[0, 3]$ and $[0, 3]$ respectively. Afterwards, the modified histogram is formed using the function:

$$H_{\text{Quantized}} + 16 \times S_{\text{Quantized}} + 64 \times V_{\text{Quantized}} \tag{1}$$

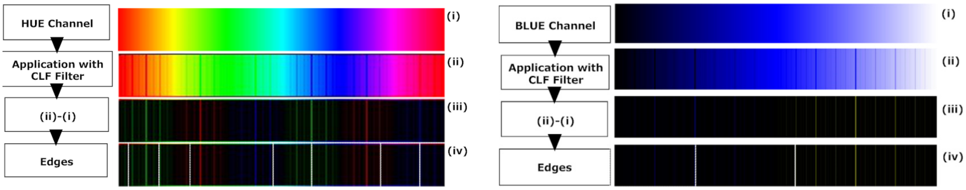
Konstantinidis *et al.*[27] proposed the extraction of a fuzzy-linking histogram based on the color space CIE-L*a*b*. Their three-input fuzzy system uses the L*, a* and b* values from each pixel in an image to classify that pixel into one of ten preset colors, transforming the image into a palette of the ten preset colors.

In this method, the defuzzyfication algorithm classifies the input pixel into one and only one output bin (color) of the system (crisp classification). Additionally, the required conversion of an image from the RGB color space to CIEXYZ and finally to CIE-L*a*b* color space makes the method noticeably time-consuming.

This paper proposes a new two-stage fuzzy-linking system using the HSV color space, which demands smaller computational power than CIELAB because HSV converts directly from the RGB color space. The first stage of the proposed fuzzy system produces a fuzzy-linking histogram that uses the three HSV channels as inputs and forms a ten-bin histogram as output. Each bin represents a preset color: (0) White, (1) Gray, (2) Black, (3) Red, (4) Orange, (5) Yellow, (6) Green, (7) Cyan, (8) Blue, and (9) Magenta.

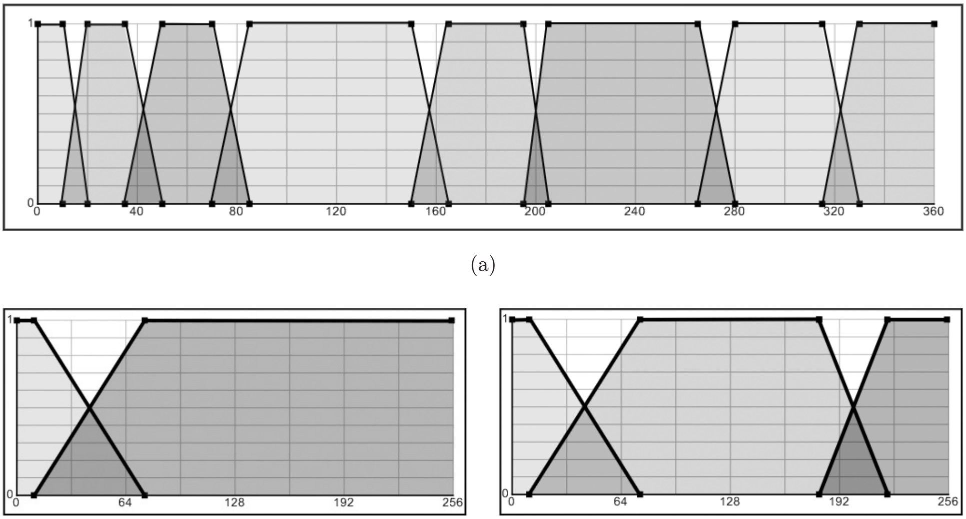
The shaping of the input membership value limits is based on the position of the vertical edges of specially constructed artificial images representing channels H (Hue), S (Saturation) and V (Value). Figure 1(a.iii) illustrates the vertical edges of the image that represents the channel H, which were used for determining the position of membership values of Fig. 2(a). The selected hue regions are stressed by dotted lines in Fig. 1(a.iv). The membership value limits of S and V are identified by the same process.

Coordinate logic filters (CLF)[34] are found to be the most appropriate among edge detection techniques for determining the fine differences and extracting these vertical edges in the specially constructed artificial images representing channels H, S and V. In our procedure, every pixel of the images that represent the channels H, S and V is replaced by the result of the coordinate logic filter "AND" operation on its 3×3 neighborhood. The values of Red, Green and Blue of the nine pixels of every neighborhood are expressed in binary form. The nine binary values of every channel from R, G and B are combined with the use of the logical operator "AND". The result is a binary number for each of the three channels R, G and B. Converting these numbers to byte form produces the value that the neighborhood's central pixel will have. This process is repeated for all the pixels and in the three specially constructed artificial images. The result of this action stresses the edges of the image (Fig. 1(a.ii)). The difference between the initial and the filtered images indicates the total edges. The position of these edges is the boundaries (Limits) of the system's Membership values.



(a) (b)

Fig. 1. Edges extraction with CLF-AND filter.



(a)

(b)

(c)

Fig. 2. Membership functions of (a) Hue, (b) Saturation and (c) Value.

Based on these edges, the inputs of the system are analyzed as follows: channel H is divided into eight fuzzy areas. Their borders are shown in Fig. 2(a) and are defined as: (0) Red to Orange, (1) Orange, (2) Yellow, (3) Green, (4) Cyan, (5) Blue, (6) Magenta and (7) Magenta to Red.

Channel S is divided into two fuzzy areas. The first area, in combination with the fuzzy area activated in channel V, determines whether the input color is clear enough to be ranked in one of the H histogram colors, or if it is simply a shade of white or gray.

The third input, channel V, is divided into three areas. The first area defines whether the input will be black, independently from the other input values. The second fuzzy area combined with the value of channel S defines gray.

A set of 20 TSK-like rules[54] with fuzzy antecedents and crisp consequents is used. These rules are given in Appendix A. The consequent section contains variables that count the number of original pixels mapped to each specific bin of the ten-bin histogram. Four of the rules depend on two only inputs (S and V) and are decided independently of the H value.

For evaluating the consequent variables, two algorithms were compared. First, an LOM (Largest of Maximum) algorithm was used. This method assigns the input to the output bin of the rule with the greatest activation value. Second, a Multi-Participant algorithm was used. This method assigns the input to the output bins which are defined by all the rules that are being activated with a participation rate to each bin proportional to the activation rate of the rule that is activated. Experimental results reveal that the second algorithm performs better.[2–4]

In the second stage of the fuzzy-linking system, a fuzzy system categorizes each color into one of three hues, producing a 24-bin histogram as output. Each bin represents a preset color as follows: (0) White, (1) Gray, (2) Black, (3) Light Red, (4) Red, (5) Dark Red, (6) Light Orange, (7) Orange, (8) Dark Orange, (9) Light Yellow, (10) Yellow, (11) Dark Yellow, (12) Light Green, (13) Green, (14) Dark Green, (15) Light Cyan, (16) Cyan, (17) Dark Cyan, (18) Light Blue, (19) Blue, (20) Dark Blue, (21) Light Magenta, (22) Magenta, (23) Dark Magenta.

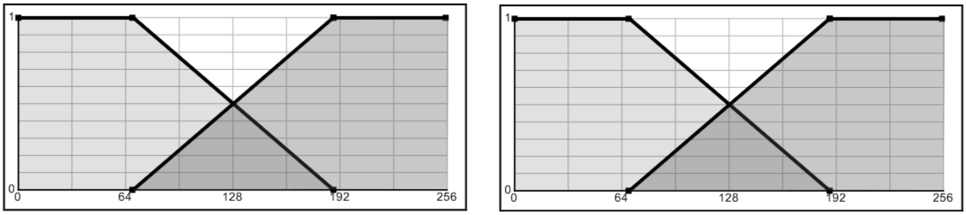
The system developed to assign these shades is based on the determinations of the subtle vertical edges appearing in images with smooth single-color transition from absolute white to absolute black. The use of a coordinate logic filter (CLF) "AND"[34] is also found to be appropriate for determining these vertical edges [Fig. 1(a.iv)].

The values of S and V from each pixel as well as the position number of the bin (or bins) resulting from the previous fuzzy ten-bin stage are the inputs to this 24-bin Fuzzy Linking system. If the previous fuzzy ten-bin stage outputs bin position number three or lower, which defines that pixel as grayscale, the fuzzy system classifies the pixel directly into the corresponding output bin without using the fuzzy rules. If the position number of the bin from the previous fuzzy ten-bin stage is greater than three, the system classifies the input pixel as belonging to one or more of the three hue areas produced by the vertical edge extraction procedure described above. These hues are labeled as follows: Light Color, Color and Dark Color (where Color is the color attribute produced by the first ten-bin stage).

The fuzzy 24-bin linking system inputs are analyzed by dividing channels S and V into two fuzzy regions as depicted in Figs. 3(a) and 3(b) respectively. A set of four TSK-like rules[54] with fuzzy antecedents and crisp consequents are used. These rules are defined in Appendix A. For the evaluation of the consequent variables, the Multi-Participant method is also employed.

3. Texture Information Extraction

Texture is one of the most important attributes used in image analysis and pattern recognition. It provides surface characteristics for the analysis of many types of images including natural scenes, remotely sensed data and biomedical modalities.[20] The present paper focuses on two new methods of texture information extraction



(a)

(b)

Fig. 3. Membership functions for (a) saturation and (b) value for the expansion at 24-bin.

based on fuzzy techniques. The first method creates an eight-bin histogram using the high-frequency bands produced by the Haar Wavelet transform. The second method creates a six-bin histogram using the five digital filters that were proposed in the MPEG-7 Edge Histogram Descriptor. In both methods each bin corresponds to a texture form.

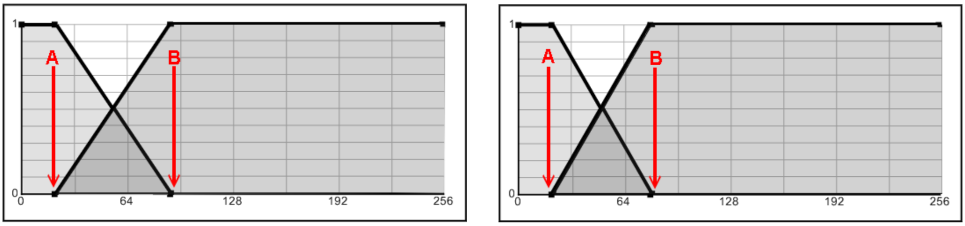
3.1. *Extraction of texture information using high frequency bands of wavelet transforms*

To export texture information from the images, three features that represent energy in high frequency bands of wavelet transforms are used. These elements are the square root of the second order moment of wavelet coefficients in high frequency bands.[7] To obtain these features, the Haar transform is applied to the Y (Luminosity in the YIQ color space) component of an *Image Block*. The choice of *Image Block* size depends on the image dimensions and is discussed in Sec. 4. Suppose, for instance, that the block size is 4×4. After a one-level wavelet transform, each block is decomposed into four frequency bands. Each band contains 2×2 coefficients. The coefficients in the HL band are $\{C_{kl}, C_{k,l+1}, C_{k+1,l}, C_{k+1,l+1}\}$. One feature is then computed as:

$$f = \left(\frac{1}{4} \sum_{i=0}^{1} \sum_{j=0}^{1} C_{k+i,l+j}^{2} \right)^{\frac{1}{2}} \tag{2}$$

The other two features are computed similarly from the LH and HH bands. The motivation for using these features is their relation to texture properties. Moments of wavelet coefficients in various frequency bands are proven effective for discerning texture.[48,49] For example, a large coefficient value on the HL band shows high activity in the horizontal direction. Thus, an image with vertical stripes has high energy in the HL band and low energy in the LH band. Research shows that this texture feature is a good compromise between computational complexity and effectiveness.[49] Elements f_{LH}, f_{HL} and f_{HH} from each image block are normalized and applied as inputs to a three-input fuzzy system that creates an eight-bin (areas) histogram as output. This method classifies the input image block into one or more output bins with the following preset texture form labels: TextuHisto(0) Low Energy Linear area, TextuHisto(1) Low Energy Horizontal activation, TextuHisto(2) Low Energy Vertical activation, TextuHisto(3) Low Energy Horizontal and Vertical activation, TextuHisto(4) High Energy Linear area, TextuHisto(5) High Energy Horizontal activation, TextuHisto(6) High Energy Vertical activation, TextuHisto (7) High Energy Horizontal and Vertical activation.

To shape the domain limits of membership values of the three fuzzy-system inputs over eight texture areas, a simple genetic algorithm is used. A database of 100 images cropped from a set of 80 texture types selected from the Brodatz Album[1] is used.



(a)

(b)

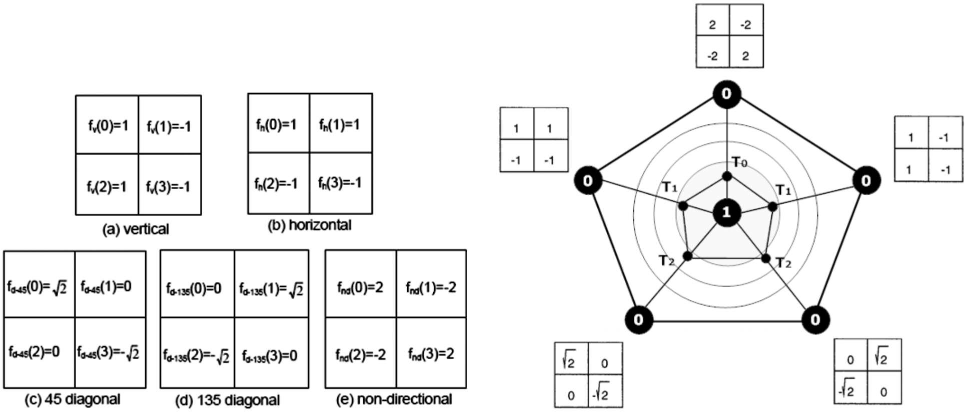
Fig. 4. Membership functions for (a) f_{LH} and f_{HL}, (b) f_{HH}.

For these images the corresponding ideal texture histograms were manually formed. The simple genetic algorithm then determines offline the limits of membership values with an AFT (Auto Fuzzy Tuning) method. Every fuzzy input is separated into two parts with trapezoidal membership functions, as illustrated in Fig. 4. Also, it is assumed that, due to the structure of the information carried by the inputs f_{HL} and f_{LH}, these two can share the same membership value limits. This assumption facilitates the algorithm's implementation. The chromosomes used by the genetic algorithm include four values, allocated in two pairs. The first pair includes the zero points (points A,B of Fig. 4(a)) of the two membership values f_{HL} and f_{LH} while the second pair contains the two zero points (points A,B of Fig. 4(b)) of f_{HH}.

The algorithm begins with a sample of 50 chromosomes. The chromosomes are in an integer and of nonbinary form. An additional control parameter assures that the second number of each pair is always greater than the first and that the number values cannot exceed the limit of their range. The zero point values from all the chromosomes are used by the fuzzy system to determine the texture type for each of the 100 images from the database.

For each image the texture histogram produced by the fuzzy system is compared with the corresponding ideal texture histogram using the Euclidean distance. The fitness function is chosen to be the sum of these Euclidean distances. The chromosomes are then sorted and the best ten are kept for the formation of the next generation. A crossover procedure is applied to the next ten best chromosomes with the algorithm using the point that separates the two pairs as the crossover point. The next best five chromosomes are mutated by increasing or decreasing only one contributor value of the chromosome. Finally, 25 additional chromosomes are randomly inserted. In all cases, the new chromosomes are not allowed to violate the control parameter restrictions. The procedure is repeated until the fitness function is minimized and there is no further improvement. Figure 4(a) shows the f_{HL} and f_{LH} inputs while the f_{HH} input is shown in Fig. 4(b).

A set of eight TSK-like rules[54] with fuzzy antecedents and crisp consequents are used. These rules are defined in Appendix A. For the evaluation of the consequent variables, the Multi-Participant method is also employed.



(a) (b)

Fig. 5. (a) Filter coefficients for edge detection, (b) edge type diagram.

3.2. *Extraction of texture information using the five digital filters proposed by the MPEG-7 EHD*

The five digital filters proposed by the MPEG-7 Edge Histogram Descriptor (EHD), are shown in Fig. 5(a).[6] These filters are used for the extraction of texture information. They are able to characterize the edges present in the applied region as one of the following texture types: vertical, horizontal, 45-degree diagonal, 135-degree diagonal and nondirectional edges. In this section a novel approach is proposed that uses these filters and permits the applied region to participate in more than one texture type.

The proposed texture feature extraction begins by dividing the image into a specified number of *Image Blocks*. Each *Image Block* contains four *Sub Blocks*. The average gray level of each *Sub Block* at (i,j)th *Image Block* is defined as $a_0(i,j)$, $a_1(i,j)$, $a_2(i,j)$, and $a_3(i,j)$. The filter coefficients for vertical, horizontal, 45-degree diagonal, 135-degree diagonal, and nondirectional edges are labeled as $f_v(k)$, $f_h(k)$, $f_{d-45}(k)$, $f_{d-135}(k)$, and $f_{nd}(k)$, respectively, where $k = 0,\dots,3$ represents the location of the Sub Block. The respective edge magnitudes $m_v(i,j)$, $m_h(i,j)$, $m_d - 45(i,j)$, $m_{d-135}(i,j)$, and $m_{nd}(i,j)$ for the (i,j)th *Image Block* can be obtained as follows:

$$m_v(i,j) = \left| \sum_{k=0}^{3} a_k(i,j) \times f_v(k) \right| \tag{3}$$

$$m_h(i,j) = \left| \sum_{k=0}^{3} a_k(i,j) \times f_h(k) \right| \tag{4}$$

$$m_{nd}(i,j) = \left| \sum_{k=0}^{3} a_k(i,j) \times f_{nd}(k) \right| \tag{5}$$

$$m_{d-45}(i,j) = \left| \sum_{k=0}^{3} a_k(i,j) \times f_{d-45}(k) \right| \tag{6}$$

$$m_{d-135}(i,j) = \left| \sum_{k=0}^{3} a_k(i,j) \times f_{d-135}(k) \right| \tag{7}$$

then the max is calculated:

$$\max = MAX(m_v, m_h, m_{nd}, m_{d-45}, m_{d-135}) \tag{8}$$

and all ms normalized

$$m'_v = \frac{m_v}{\max}, \quad m'_d = \frac{m_d}{\max}, \quad m'_{nd} = \frac{m_{nd}}{\max}, \quad m'_{d-45} = \frac{m_{d-45}}{\max}, \quad m'_{d-135} = \frac{m_{d-135}}{\max} \tag{9}$$

The output of the unit that extracts texture information from each *Image Block* is a six-bin (area) histogram. Each bin corresponds to a preset region as follows: EdgeHisto(0) Non Edge, EdgeHisto(1) Non Directional Edge, EdgeHisto(2) Horizontal Edge, EdgeHisto(3) Vertical Edge, EdgeHisto(4) 45-Degree Diagonal and EdgeHisto(5) 135-Degree Diagonal. The system classifies each *Image Block* in a two-step process: first, the system calculates the *max* value. The *max* value must be greater than the defined threshold for the *Image Block* to be classified as a *Texture Block*, otherwise it is classified as a Non Texture Block (Linear). Then, if the *Image Block* is classified as a *Texture Block*, each m value is placed on the pentagonal diagram of Fig. 5(b) along the line corresponding to digital filter from which it was calculated. The diagram's center corresponds to value 1 and the outer edge corresponds to value 0. If any m value is greater than the threshold on the line where it participates, the *Image Block* is classified into the particular type of edge. Thus an *Image Block* can participate in more than one edge type. The following source code describes the process:

```
program SetEdgeType(max, m_nd, m_h, m_v, m_d_45, m_d_135)
{
  if (max < TEdge) then EdgeHisto(0)++
  else
     {
     if (m_nd > T0)    then EdgeHisto(1)++
     if (m_h > T1)     then EdgeHisto(2)++
     if (m_v > T1)     then EdgeHisto(3)++
     if (m_d_45 > T2)  then EdgeHisto(4)++
     if (m_d_135 > T2) then EdgeHisto(5)++
     }
  endif
return(EdgeHisto)
}
```

For the calculation of the thresholds, the genetic algorithm described in Sec. 3.1 is used again. In this case, the chromosome length is only three values that correspond to T_{edge}, T_0 and T_1. For convenience, the implementation assumes that $T_1 = T_2$. To avoid decimal numbers, the values of T_0 and T_1 are transformed into space $[0, 100]$, thereby avoiding modifications to the mutation method. The extra control parameter used by the fuzzy system in Sec. 3.1 is replaced by a new parameter that limits the threshold values to their allowable boundaries. Furthermore, the crossover point is determined to allow a crossover procedure between T_0 and T_1. The threshold values are set as: $T_{\text{Edge}} = 14$, $T_0 = 0.68$, $T_1 = T_2 = 0.98$.

4. Descriptor Implementation

The color and texture features described in the previous sections are combined to produce four descriptors. In order to form the proposed descriptors, the image is initially separated into 1600 *Image Blocks*. This number is chosen as a compromise between the image detail and the computational demand. Considering that the minimum size of each *Image Block* must be 2×2 pixels (a restriction that comes from the Texture units), the proposed descriptors are used for images larger than 80×80 pixels.

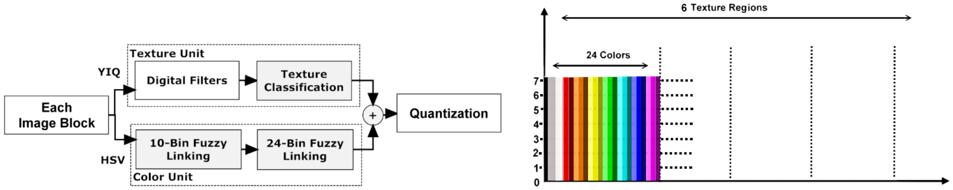
The proposed descriptors are constructed as follows: the unit associated with color information extraction in every descriptor is called the *Color Unit*. Similarly, the *Texture Unit* is the unit associated with texture information extraction. The descriptors' structure has n regions determined by the *Texture Unit*. Each *Texture Unit* region contains m individual regions defined by the *Color Unit*. Overall, each proposed descriptor contains $m \times n$ bins. On the completion of the process, each descriptor's histogram is normalized within the interval $[0, 1]$ and then quantized into three bits/bins. The quantization process and the quantization tables are described in Sec. 5.

4.1. *CEDD — Color and edge directivity descriptor*

The CEDD includes texture information produced by the six-bin histogram of the fuzzy system that uses the five digital filters proposed by the MPEG-7 EHD. Additionally, for color information the CEDD uses the 24-bin color histogram produced by the 24-bin fuzzy-linking system. Overall, the final histogram has $6 \times 24 = 144$ regions.

Each *Image Block* interacts successively with all the fuzzy systems. Defining the bin produced by the texture information fuzzy system as n and the bin produced by the 24-bin fuzzy-linking system as m, then each *Image Block* is placed in the bin position: $n \times 24 + m$.

The process of generating the descriptor is described in the flowchart Fig. 6(a). In the *Texture Unit*, the *Image Block* is separated into four regions called *Sub Blocks*. The value of each *Sub Block* is the mean value of the luminosity of the pixels it contains. The luminosity values are derived from a YIQ color space transformation.



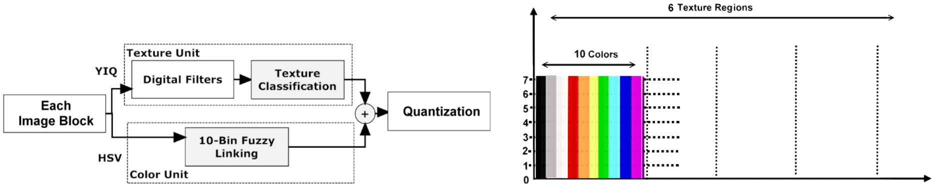
(a) (b)

Fig. 6. CEDD (a) Implementation flowchart, (b) structure.

Each *Image Block* interacts with the five digital filters proposed by MPEG-7 EHD, and with the use of the pentagonal [Fig.5(b)] diagram it is classified in one or more texture categories. For illustration purposes let us assume that the *Texture Unit* classifies a given *Image Block* into the second bin which is defined as NDE (Non Directional Edge). Then, in the *Color Unit*, every *Image Block* is converted to the HSV color space. The mean values of H, S and V are calculated and become inputs to the fuzzy system that produces the fuzzy ten-bin histogram. Let us again assume that the classification resulted in the fourth bin which indicates that the color is red. Then, the second fuzzy system (24-bin Fuzzy Linking System), using the mean values of S and V as well as the position number of the bin (or bins) resulting from the previous fuzzy ten-bin unit, calculates the hue of the color and produces the fuzzy 24-bin histogram. And let us assume that the *Color Unit* system classifies this block in the fourth bin which indicates the color as (3) Light Red. The combination of the three fuzzy systems will finally classify the *Image Block* in the 27th bin $(1 \times 24 + 3)$. The process is repeated for all the image blocks. At the completion of the process, the histogram is normalized within the interval $[0, 1]$ and quantized according to the process described in Sec. 5. Figure 6(b) illustrates the CEDD structure.

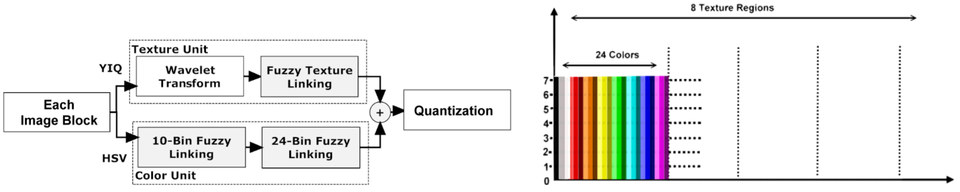
4.2. *C.CEDD — Compact color and edge directivity descriptor*

The method for producing the C.CEDD differs from the CEDD method only in the color unit. The C.CEDD uses the fuzzy ten-bin linking system instead of the fuzzy 24-bin linking system. Overall, the final histogram has only $6 \times 10 = 60$ regions. It is the smallest descriptor of the proposed set. The flowchart in Fig. 7(a) describes the generation of the C.CEDD while Fig. 7(b) shows its structure.



(a) (b)

Fig. 7. C.CEDD (a) Implementation flowchart, (b) structure.



(a) (b)

Fig. 8. FCTH (a) Implementation flowchart, (b) structure.

4.3. *FCTH — Fuzzy color and texture histogram*

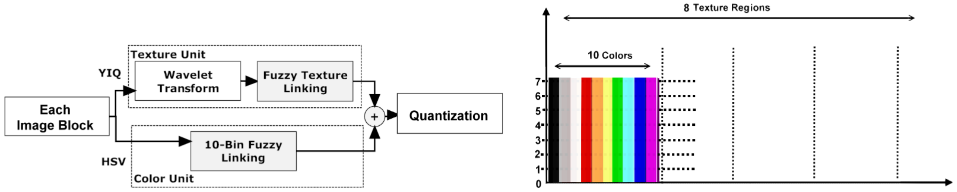
The FCTH descriptor includes the texture information produced in the eight-bin histogram of the fuzzy system that uses the high frequency bands of the Haar wavelet transform. For color information, the descriptor uses the 24-bin color histogram produced by the 24-bin fuzzy-linking system. Overall, the final histogram includes $8 \times 24 = 192$ regions.

Each *Image Block* interacts successively with all the fuzzy systems in the exact manner demonstrated in CEDD production. The FCTH descriptor generation is described in Fig. 8(a) flowchart.

Each *Image Block* is transformed into the YIQ color space and transformed with the Haar Wavelet transform. The f_{LH}, f_{HL} and f_{HH} values are calculated and with the use of the fuzzy system that classifies the f coefficients, this *Image Block* is classified in one of the eight output bins. Suppose, for example, that the classification assigns this block to the second bin defined as Low Energy Horizontal activation. Next, the same *Image Block* is transformed into the HSV color space and the mean H, S and V block values are calculated. These values become inputs to the fuzzy system that forms the ten-bin fuzzy color histogram. Let us assume that this system classifies a given block into the fourth bin defined as color (3) Red. Then, the next fuzzy system uses the mean values of S and V as well as the position number of the bin (or bins) resulting from the previous fuzzy ten-bin unit, to calculate the hue of the color and create the fuzzy 24-bin histogram. Let us assume that the system classifies this *Image Block* in the fourth bin which defines that color as (3) Light Red. The combined three fuzzy systems therefore classify the *Image Block* into the 27th bin $(1 \times 24 + 3)$. The process is repeated for all the blocks of the image. At the completion of the process, the histogram is normalized within the interval $[0, 1]$ and quantized according to the procedures described in Sec. 5. Figure 8(b) illustrates the FCTH descriptor structure.

4.4. *C.FCTH — Compact fuzzy color and texture histogram*

The method for producing C.FCTH differs from the FCTH method only in the color unit. Like its C.CEDD counterpart, this descriptor uses only the fuzzy ten-bin linking system instead of the fuzzy 24-bin linking system. Overall, the final histogram



(a) (b)

Fig. 9. C.FCTH (a) Implementation flowchart, (b) structure.

includes only 8 × 10 = 80 regions. The flowchart in Fig. 9(a) describes the procedure for generating the C.FCTH descriptor while Fig. 9(b) shows the C.FCTH structure.

5. Descriptor Quantization

To restrict the proposed descriptors' length, the normalized bin values of the descriptors are quantized for binary representation in a three bits/bin quantization. For example, the 144-bin CEDD is limited to 144 × 3 = 432 bits. Because most of the values are concentrated within a small range (from 0 to 0.25), they are nonlinearly quantized. Also, the descriptor bins are divided into separate quantization groups with differing quantization values.

In order to calculate the CEDD quantization table, a sample of 10,000 images is used. First, CEDD vectors are calculated for all images. The combined 10,000 × 144 elements constitute inputs into the fuzzy Gustafson Kessel classifier,[18] which separates the volume of the samples into eight regions, mapping the bin values from the decimal area $[0, 1]$ into the integer area $[0, 7]$. The Gustafson Kessel parameters are selected as: clusters = 8, repetitions = 2000, $e = 0.002$, and $m = 2$. The resulting quantization is given in Table 1. The values of the histogram appearing in bins 0−23 are assigned to one of the values $[0, 7]$ according to the minimum distance of each bin value from one of the eight entries in the first row of the table. The same procedure is

Table 1. CEDD quantization table.

CEDD Bin: 0−23/C.CEDD Bin: 0−9							
000	001	010	011	100	101	110	111
0.00018	0.0237	0.0614	0.1139	0.1791	0.2609	0.3417	0.5547
CEDD Bin: 24−47/C.CEDD Bin: 10−19							
000	001	010	011	100	101	110	111
0.00020	0.0224	0.0602	0.1207	0.1811	0.2341	0.3256	0.5207
CEDD Bin: 48−95/C.CEDD Bin: 20−39							
000	001	010	011	100	101	110	111
0.00040	0.0048	0.0108	0.0181	0.0270	0.0381	0.0526	0.0795
CEDD Bin: 96−143/C.CEDD Bin: 40−59							
000	001	010	011	100	101	110	111
0.00096	0.0107	0.0241	0.0415	0.0628	0.0930	0.1369	0.2628

Table 2. FCTH quantization table.

FCTH Bin: 0−47/C.FCTH Bin: 0−19							
000	001	010	011	100	101	110	111
0.00013	0.0093	0.0224	0.0431	0.0831	0.1014	0.1748	0.224
FCTH Bin: 48−143/C.FCTH Bin: 20−59							
000	001	010	011	100	101	110	111
0.00023	0.0173	0.0391	0.0693	0.0791	0.0910	0.1618	0.185
FCTH Bin: 144−191/C.FCTH Bin: 60−79							
000	001	010	011	100	101	110	111
0.00018	0.0273	0.0414	0.0539	0.0691	0.0820	0.0918	0.128

followed for the entries in bins 24−47, 48−95 and 96−143 using the quantization values shown in each of their corresponding rows in the table.

The quantization table for FCTH descriptor is calculated in a similar manner, limiting its total length to $192 \times 3 = 576$ bits. The resulting quantization is presented in Table 2. The values of the histogram appearing in bins 0−47 are assigned to one of the values $[0, 7]$ according to the minimum distance of each bin value from one of the eight entries in the first row of the table. The same procedure is followed for the entries in bins 48−143 and 144−191 using the quantization values shown in each of their corresponding rows.

For convenience, in the implementation of systems that use the proposed descriptors, the quantization tables of the compact versions of the descriptors are the same as the quantization tables of the noncompact versions. The C.CEDD quantization table is the same as the CEDD quantization table. Likewise, the C.FCTH quantization table is the same as the FCTH quantization table. The CEDD length is 54 bytes per image, FCTH length is 72 bytes per image, C.CEDD requires less than 23 bytes per image and C.FCTH uses 30 bytes per image.

6. Experiments

Recently, standard benchmark databases and evaluation campaigns have been created allowing a quantitative comparison of CBIR systems. These benchmarks allow the comparison of image retrieval systems under different aspects: usability and user interfaces, combination with text retrieval, or overall performance of a system.[9] The proposed descriptors are integrated into the retrieval software system img(Rummager)[5] and the online application img(Anaktisi)[52] where they can be quantitatively evaluated.

Img(Rummager) is developed by the authors of this paper in the Automatic Control Systems & Robotics Laboratory[a] at the Democritus University of Thrace-Greece. This system is implemented in C# and operates on an Intel Pentium 3.4 MHz PC (2 GB RAM memory). Img(Rummager) software can connect to a

[a] ACSL: http://www.ee.duth.gr/acsl

database and execute a retrieval procedure, extracting the comparison features in real time. The image database can be stored either on the computer where the retrieval takes place or on a local network. Moreover, this software is capable of executing retrieval procedures among the keyword-based (Tags) results that the Flicker provides. Img(Anaktisi) was also developed by the authors of this paper at the Image Processing and Multimedia Laboratory[b] at the Democritus University of Thrace-Greece. This web program[c] is programmed in C# with the help of Visual Studio 2008 and is based on the Microsoft .NET Framework 3.5. It also employs AJAX, HTML and Javascript technologies for user interaction. Finally, Microsoft SQL Server 2005 is the database used by the web platform to store and retrieve the descriptors for each image.

To evaluate the performance of the descriptors, experiments are performed on five image databases: WANG's database,[31,49] MPEG-7 CCD database, UCID[43] database, img(Rummager) database and Nister database.[38] All the results are available online.[d] Figure 12 illustrates the ANMRR values for the five benchmarking image databases.

6.1. *Similarity measure*

For similarity matching, the distance $D(i, j)$ of two image descriptors x_i and x_j is calculated using the nonbinary Tanimoto coefficient.

$$D(i,j) = T_{ij} = t(x_i, x_j) = \frac{x_i^T x_j}{x_i^T x_i + x_j^T x_j - x_i^T x_j} \tag{10}$$

where x^T is the transpose vector of the descriptor x.

In the absolute congruence of the vectors, the Tanimoto coefficient takes the value 1, while in the maximum deviation the coefficient tends to 0. The Tanimoto Coefficient was found to be preferable than the similarity L1, L2 (Euclidean Distance), Jensen-Shannon[40] and Bhattacharyya because it presented better results. In Sec. 6.6, which describes the experiments carried out on the img(Rummager) database, the ANMRR values for all the different similarity metrics used are outlined in detail.

6.2. *Performance evaluation*

The objective Averaged Normalized Modified Retrieval Rank (ANMRR)[35] is employed to evaluate the performance of the image retrieval system that uses the proposed descriptors in the retrieval procedure.

[b] IPML: http://ipml.ee.duth.gr

[c] http://www.anaktisi.net

[d] http://orpheus.ee.duth.gr/anaktisi/results

The average rank AVR(q) for query q is:

$$\mathrm{AVR}(q) = \sum_{k=1}^{\mathrm{NG}(q)} \frac{\mathrm{Rank}(k)}{\mathrm{NG}(q)} \tag{11}$$

where

- NG(q) is the number of ground truth images for query q. A ground truth is defined as a set of visually similar images.
- $K = \min(X_{NG} \times \mathrm{NG}(q), 2 \times \mathrm{GTM})$.
- $\mathrm{GTM} = \max(\mathrm{NG})$.
- If $\mathrm{NG}(q) > 50$ then $X_{\mathrm{NG}} = 2$ else $X_{\mathrm{NG}} = 4$.
- Rank(k) is the retrieval rank of the ground truth image.

  Consider a query. Assume that as a result of the retrieval, the kth ground truth image for this query q is found at a position R. If this image is in the first K retrievals then $\mathrm{Rank}(k) = R$ else $\mathrm{Rank}(k) = (K + 1)$.

The modified retrieval rank is:

$$\mathrm{MRR}(q) = \mathrm{AVR}(q) - 0.5 \times [1 + \mathrm{NG}(q)] \tag{12}$$

Note that MRR is 0 in case of perfect retrieval. The normalized modified retrieval rank is computed as follows:

$$\mathrm{NMRR}(q) = \frac{\mathrm{MRR}(q)}{1.25 \times K - 0.5 \times [1 + \mathrm{NG}(q)]} \tag{13}$$

Finally the average of NMRR over all queries is defined as:

$$\mathrm{ANMRR} = \frac{1}{Q} \sum_{q=1}^{Q} \mathrm{NMRR}(q) \tag{14}$$

where

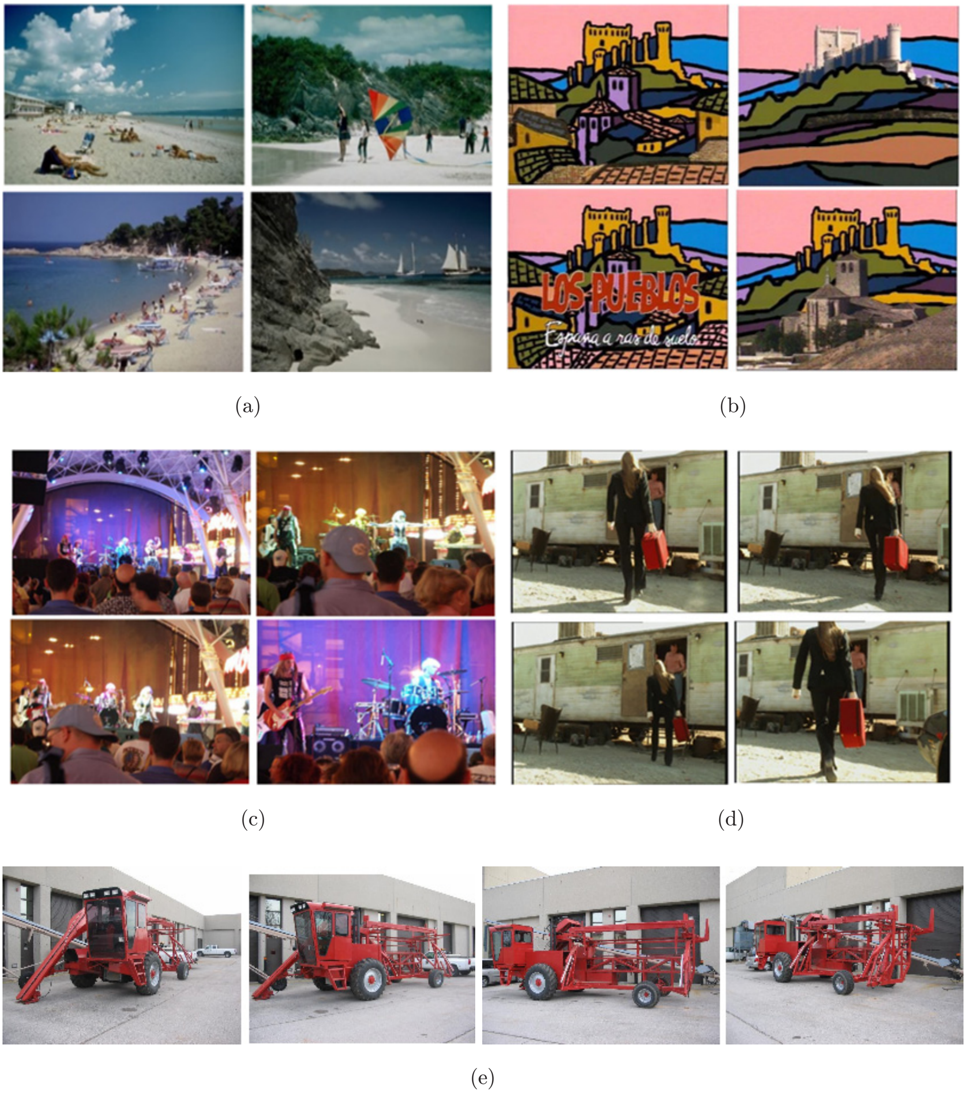
- Q is the total number of queries.

The ANMRR is always in the range of 0 to 1 and the smaller the value of this measure, the better the matching quality of the query. ANMRR is the evaluation criterion used in all of the MPEG-7 color core experiments. Evidence shows that the ANMRR measure approximately coincides linearly with the results of subjective evaluation of search engine retrieval accuracy.[35]

6.3. *Experiments on the WANG database*

The WANG database[31,49] is a subset of 1000 manually-selected images from the Corel stock photo database and forms ten classes of 100 images each. This image database is available online.[e] In particular, queries and ground truths proposed by

[e] http://wang.ist.psu.edu/docs/home.shtml



(a) (b)

(c) (d)

(e)

Fig. 10. Query examples in (a) WANG's database, (b) MPEG-7 CCD database, (c) UCID database and, (d) img(Rummager) database and (e) Nister database. The first image on the top left of each group is also the query image.

the MIRROR[51] image retrieval system are used. MIRROR separates the WANG database into 20 queries. A sample query is illustrated in Fig. 10(a).

The proposed descriptors are used in the retrieval procedure and the results are compared with the corresponding results of the following MPEG-7[35,23,24] descriptors:

`Color Descriptors`: Dominant Color Descriptor (DCD eight colors), Scalable Color Descriptor (SCD-32 colors), Color Layout Descriptor (CLD), Color Structure Descriptor (CSD-32 colors).

Table 3. Results from the WANG image database.

Descriptor	Query					ANMRR
	204	327	522	600	703	
MPEG-7 Descriptors						
DCD MPHSM-8 Colors	0.543	0.407	0.556	0.215	0.306	0.39460
DCD QHDM-8 Colors	0.420	0.469	0.537	0.610	0.781	0.54680
SCD-32 Colors	0.442	0.406	0.508	0.083	0.211	0.35520
CLD	0.616	0.542	0.454	0.454	0.252	0.40000
CSD-32 Colors	0.323	0.348	0.526	0.066	0.146	0.32460
EHD	0.782	0.317	0.690	0.277	0.307	0.50890
HTD	0.887	0.594	0.734	0.445	0.615	0.70540
Other Descriptors						
RGB color histogram	0.618	0.899	0.715	0.569	0.820	0.59134
Tamura directionality	0.889	0.682	0.806	0.690	0.574	0.63622
Correlograms	0.493	0.458	0.674	0.334	0.664	0.50107
Proposed Descriptors						
CEDD	0.314	0.127	0.347	0.059	0.115	**0.25283**
FCTH	**0.235**	**0.114**	**0.323**	0.026	0.092	0.27369
C.CEDD	0.316	0.140	0.452	0.069	**0.088**	0.30637
C.FCTH	0.320	0.224	0.493	**0.013**	0.116	0.31537

`Texture Descriptors`: Edge Histogram Descriptor (EHD), Homogeneous Texture Descriptor (HTD).

The NMRR values for the MPEG-7 descriptors in WANG's database are available at Ref. 51. Table 3 shows indicative examples of query results and the ANMRR scores for all 20 queries. The results of the proposed descriptors are also compared with the results of the RGB Color Histogram, Tamura Directionality Histogram[46] and Auto Color Correlograms.[19]

Color histograms are among the most basic approaches and are widely used in image retrieval. The color space is partitioned and for each partition the pixels with a color within its range are counted, resulting in a representation of the relative frequencies of the occurring colors.[9] We use the RGB color space for the histograms. The distance between the images was measured using L2.

The Tamura Directionality histogram is a graph of local edge probabilities against their directional angle. For the purpose of these experiments, the 16-bin Tamura Directionality Histogram was used, and the distance was calculated using L2.

Color Correlograms distill the spatial correlation of colors, and are both effective and inexpensive for content-based image retrieval. The correlogram robustly tolerates large changes in appearance and shape caused by changes in viewing positions, camera zooms, etc.[19] For the purpose of these experiments, the approach suggested in Ref. 19 with maxdistance $= 16$ was used.

As the results in Table 3 show, on the WANG database the proposed descriptors achieve better retrieval scores than the other descriptors.

Table 4. Mean average precision [%] for each of the features in the WANG image database.

Descriptor	MAP	Descriptor	MAP
CEDD	**50.6**	32 × 32 image	37.6
FCTH	50.1	MPEG7: color layout	41.8
C.CEDD	49.3	X × 32 image	24.3
C.FCTH	47.6	Tamura texture histogram	38.2
Color histogram	50.5	LF SIFT signature	36.7
LF SIFT global search	38.3	Gray value histogram	31.7
LF patches histogram	48.3	LF patches global	30.5
LF SIFT histogram	48.2	MPEG7: edge histogram	40.8
Inv. feature histogram (monomial)	47.6	Inv. feature histogram (relational)	34.9
MPEG7: scalable color	46.7	Gabor vector	23.7
LF patches signature	40.4	Global texture feature	26.3
Gabor histogram	41.3		

In order to be able to compare the results of the proposed descriptors with even more descriptors in the bibliography, the following experiment was carried out. For each image in the Wand database, a search was carried out over the total of 1000 and the AP (Average Precision) was calculated, assuming the Ground Truth to be the remaining 99 images belonging to the same group as the query image. Then the mean of all these average precisions (MAP) was taken. The results are presented in Table 4. The values of the remaining descriptors are taken from Ref. 9. The bigger the MAP, the better the descriptor is. As the results show, the CEDD presents the best results out of all the descriptors.

The deviation that appears between the MAP and the ANMRR is due to the difference between experiments. In the first experiment, only 20 queries were used, with an average ground truth of about 30 images, whereas in the second, 1000 queries (all the images) were used, with 99 images for each ground truth.

6.4. *Experiments on the MPEG-7 CCD database*

The Common Color Dataset (MPEG-7 CCD) contains approximately 5000 images and of a set of 50 common color queries (CCQ). Each query is specified with a set of ground truth images. This is the image database where the MPEG-7 descriptors have been tested. CCD contains images that originated from consecutive frames of television shows, newscasts and sport shows. It also includes a large number of photomaps. MPEG-7 CCD is a database that is clearly designed to be tested with color descriptors, frequently causing texture descriptors to present very low retrieval scores. A query sample is illustrated in Fig. 10(b). The NMRR values for the MPEG-7 descriptors in MPEG-7 CCD database are also available in Ref. 51. Table 5 shows certain indicative query results and the ANMRR values for all 50 queries.

On the MPEG-7 CCD database, the proposed descriptors appear to present the second best scores. The Color Structure Descriptor achieved the best score. The reason that the proposed descriptors failed to satisfactorily retrieve entire ground

Table 5. Results from the MPEG-7 CCD image database.

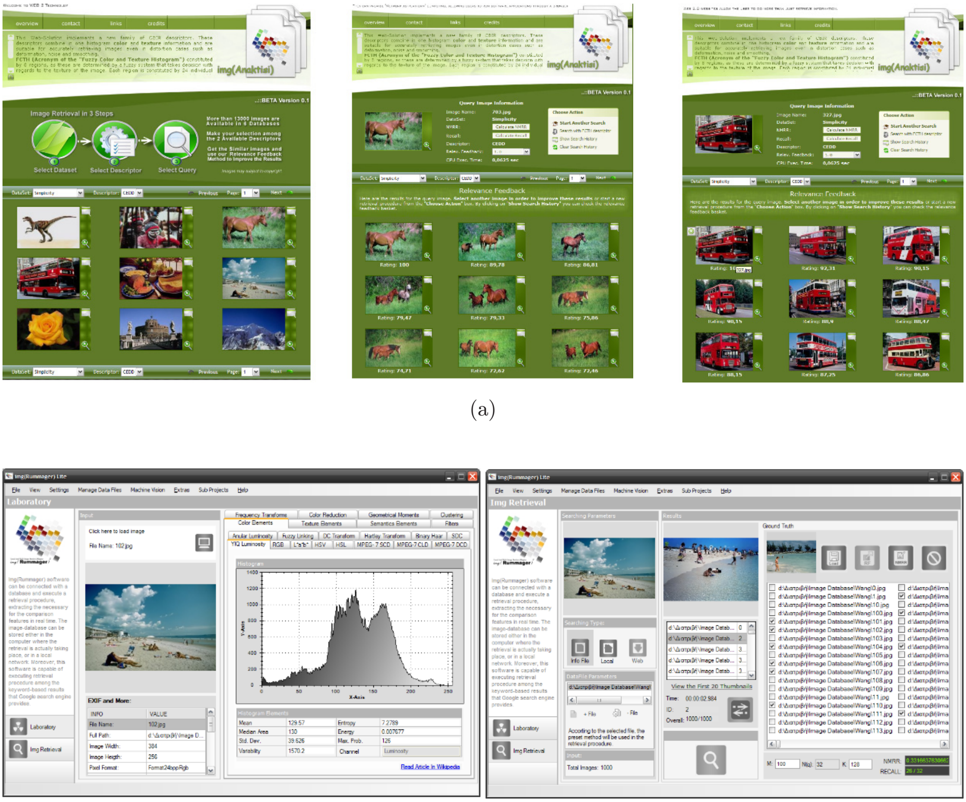
Descriptor	Query			ANMRR
	i0121_add5	img00133_add3	img00438_s3	
MPEG-7 Descriptors				
DCD MPHSM-8 colors	**0**	0.484	0.008	0.2604
DCD QHDM-8 colors	0.057	0.438	0.400	0.2834
SCD-32 Colors	**0**	0.152	**0**	0.1645
CLD	**0**	0.401	0.308	0.2252
CSD-32 Colors	**0**	**0**	**0**	**0.0399**
EHD	**0**	0.406	0.381	0.3217
HTD	0.229	0.401	0.486	0.42498
Other Descriptors				
RGB color histogram	0.229	0.401	0.161	0.42729
Tamura directionality	0.314	0.770	0.714	0.65913
Correlograms	0.000	0.290	0.294	0.28749
Proposed Descriptors				
CEDD	**0**	**0**	0.033	0.08511
FCTH	**0**	0.037	0.003	0.10343
C.CEDD	**0**	0.014	0.167	0.12655
C.FCTH	**0**	0.065	**0**	0.15977

truths for some queries is due to the fact that the MPEG-7 CCD ground truths include images that are directed toward descriptors that mostly control color similarity. The very low scores presented by the MPEG-7 texture descriptors and the Tamura Directionality descriptor confirm this assertion. Another reason for less than perfect recall is the fact that many queries include rotated images in their ground truth. Due to their texture attribute, the proposed descriptors are not suitable for retrieving these images.

6.5. *Experiments on the UCID database*

The UCID database was created as a benchmark database for CBIR and image compression applications.[43] This database currently consists of 1338 uncompressed TIFF images on a variety of topics including natural scenes and man-made objects, both indoors and outdoors. The UCID database is available to fellow researchers.[f] All the UCID images were subjected to manual relevance assessments against 262 selected images, creating 262 ground truth image sets for performance evaluation. In the assessment, only very clearly relevant images are considered to be suitable. This relevance assumption makes the retrieval task easy because the ground truth images are quite similar. On the other hand, it makes the task difficult, because there are images in the database with high visual similarity that are not considered relevant. Hence, it can be difficult to have highly precise results with the given relevance assessment, but because only few images are considered relevant, high recall values

[f] http://vision.cs.aston.ac.uk/datasets/UCID/ucid.html



(a)

(b)

Fig. 11. Screenshots from (a) img(Anaktisi) and (b) img(Rummager).

Table 6. Results from the UCID image database.

Descriptor	Query				ANMRR
	ucid00095	ucid00172	ucid00297	ucid00583	
MPEG-7 Descriptors					
SCD-32 Colors	0.471	0.058	0.471	0.384	0.46665
CLD	0.471	0.471	0.471	0.299	0.43216
EHD	0.471	0.471	**0**	0.477	0.43314
Other Descriptors					
RGB color histogram	0.471	0.471	0.176	0.553	0.52315
Tamura directionality	0.471	0.471	0.471	0.536	0.55682
Correlograms	0.059	0.471	0.294	0.360	0.41386
Proposed Descriptors					
CEDD	**0**	0.471	**0**	**0.147**	**0.28234**
FCTH	0.059	**0**	**0**	0.191	0.28737
C.CEDD	**0**	0.059	**0**	0.241	0.29331
C.FCTH	0.059	**0**	0.059	0.236	0.30871

Table 7. Mean average precision [%] for each of the features for the UCID image database.

Descriptor	MAP	Descriptor	MAP
CEDD	45	32×32 image	22.3
FCTH	44.7	MPEG7: color layout	14
C.CEDD	42.1	$X \times 32$ image	21.7
C.FCTH	40.4	Tamura texture histogram	13.9
Color histogram	43.3	LF SIFT signature	33.2
LF SIFT global search	**62.5**	Gray value histogram	34.1
LF patches histogram	37.5	LF patches global	11.8
LF SIFT histogram	44.7	MPEG7: edge histogram	30.3
Inv. feature histogram (monomial)	41.6	Inv. feature histogram (relational)	25.2
MPEG7: scalable color	37.9	Gabor vector	14.4
LF patches signature	27.6	Global texture feature	4.7
Gabor histogram	6.7		

might be easy to obtain.[10] A query sample is presented in Fig. 10(c). In Table 6, certain indicative results and ANMRR values for all of the 262 queries are demonstrated.

Because the MPEG-7 descriptor results are not available for this database, an implementation of CLD, SCD and EHD in img(Rummager)[g] application is used. The source code is a modification of the implementation that can be found in the LIRE[33] retrieval system. The original version of the descriptors' implementation is written in Java and is available online as open source[h] under the General Public License (GPL). The Img(Rummager) application results match the LIRE results. As shown in Table 6, on the UCID database the proposed descriptors achieve the best retrieval results.

The experiments were also repeated in this database to calculate the MAP. In this case, the ground truth used for every query image was that suggested by the database, but without including the query image. The results are presented in Table 7. The values of the remaining descriptors are taken from Ref. 9.

As can be seen from the results in Table 7, the CEDD presents the second best result, with the best descriptor being the LF SIFT Global Search,[32] which could be expected, because database consists of very close matches that are suitable for SIFT features.

The LF SIFT Global Search descriptor is non-compact and is extracted from the Harris interest points (Local).[11] When comparing the results of the proposed descriptors with the results of the corresponding global compact descriptors, it can be observed that the proposed descriptors have the better MAP.

In the USID case, the deviation appearing between the ANMRR and the MAP is due to the fact that, in the latter case (experiments for MAP measurement), the ground truths did not contain the query image. Given that many ground truths

[g] The prototype is available along with documentation and screenshots at http://www.img-rummager.com

[h] http://sourceforge.net/project/downloading.php?groupname=caliph-emir&filename=Lire-0.5.4.zip&use_mirror=switch

contain only two−three images, the unsuccessful retrieval of any of these would greatly influence the results.

6.6. *Experiments on the img(Rummager) database*

The img(Rummager) database is integrated in the retrieval software system img (Rummager) and includes 22,000 images. The first 4343 images come from the Microsoft Research Cambridge image database,[i] and are used mostly for object detection.[44,50] This database also includes 1000 images from the LabelME image database,[41] 2333 images from the Zubud[j] image database, 1000 Chinese art images, 1000 images of famous paintings, 3000 images from television frames, 224 images from the ICPR 2004 image set, 500 images from the VASC[k] image database and finally a set of images from personal collections. All the images are high quality, multi-object, color photographs that have been chosen according to strict image selection rules.[17] The database includes 100 queries, with an average ground truth size of approximately 15 images. A sample query is illustrated in Fig. 10(d).

In this database, the implementation of CLD, SCD and EHD in img(Rummager) application is also used. As shown in Table 8, on the img(Rummager) database the proposed descriptors achieve the best retrieval results.

Img(Rummager) is the database that was used as the core of the experiments for the shaping of the proposed descriptors. Table 9 shows the ANMRR results for the

Table 8. Results from the img(Rummager) image database.

Descriptor	Query						ANMRR
	286	133	327	400	967	703	
MPEG-7 Descriptors							
SCD-32 Colors	0.012	**0**	0.239	0.256	**0**	0.112	0.29755
CLD	0.124	0.011	0.368	0.256	**0**	**0**	0.31325
EHD	0.786	0.211	0.876	0.498	0.1225	0.2214	0.51214
Other Descriptors							
RGB color histogram	0.224	0.211	0.239	0.256	0.112	0.000	0.30156
Tamura directionality	0.786	0.321	0.239	0.487	0.112	0.275	0.54211
Correlograms	0.000	0.078	0.352	0.256	0.000	0.012	0.25412
Proposed Descriptors							
CEDD	**0**	**0**	**0**	0.110	**0**	**0**	0.20443
FCTH	**0**	0.078	**0**	**0**	**0**	**0**	**0.19239**
C.CEDD	0.010	**0**	0.043	**0**	**0**	0.112	0.24332
C.FCTH	**0**	0.078	**0**	0.144	**0**	0.112	0.24356

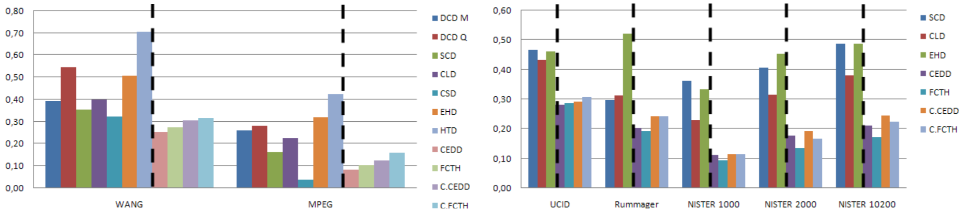
[i] http://research.microsoft.com/vision/cambridge/recognition/default.htm

[j] http://www.vision.ee.ethz.ch/datasets/index.en.html

[k] http://www.ius.cs.cmu.edu/idb/

Table 9. ANMRR results from the img(Rummager) database using several similarity matching techniques.

Descriptor	Tanimoto	L1	L2	Jensen-Shannon	Bhattacharyya
CEDD	0.20443	0.23554	0.20558	0.26554	0.24224
CEDD un-quantized	0.23665	0.25554	0.24013	0.26558	0.23112
FCTH	0.19239	0.21214	0.01125	0.20221	0.21112
FCTH un-quantized	0.18669	0.23325	0.23855	0.23745	0.21556



(a) (b)

Fig. 12. ANMRR results for (a) Wang and MPEG-7 databases and (b) UCID, img(Rummager) and Nister databases.

CEDD and FCTH descriptors using several similarity metric techniques. As the results show, Tanimoto Coefficient presented the best results.

6.7. *Experiments on the Nister image database*

The Nister image database consists of N groups of four images each.[38] All the images are 640 × 480. Each group includes images of a single object. The pictures are taken from different viewpoints and occasionally under different lighting conditions. The first image of every object is used as a query image. Given a query image, only images from the same group are considered relevant.

For the purpose of calculating the efficiency of the proposed descriptors, the database is divided into three subsets. The first subset includes the first 1000 images of the database with 250 queries. The second subset consists of the first 2000 images with 500 queries where half (250 queries) are from the first subset. The third subset includes the entire dataset of 10,200 images with the same 500 queries used in the second subset. A sample query is illustrated in Fig. 10(e).

The Nister database retrieval difficulty is dependent on the chosen subset. Important factors are:

(1) Difficulty of the objects themselves. CD-covers are much easier than flowers.
(2) Sharpness of the images. Many of the indoor images are somewhat blurry and this can affect some algorithms.
(3) Similar or identical objects in different groups.

Table 10. Results on Nister image database.

Descriptor	Query			ANMRR
	ukbench00052	ukbench00352	ukbench00900	
		1000 Images		
MPEG-7 Descriptors				
SCD-32 Colors	0.157	0.129	0.729	0.36365
CLD	0.471	0.486	0.129	0.22920
EHD	0.229	0.143	0.371	0.30060
Proposed Descriptors				
CEDD	**0**	**0**	**0**	0.11297
FCTH	**0**	**0**	**0**	**0.09463**
C.CEDD	**0**	**0**	**0**	0.11537
C.FCTH	**0**	0.071	**0**	0.1152
		2000 Images		
MPEG-7 Descriptors				
SCD-32 Colors	0.171	0.157	0.729	0.40589
CLD	0.471	0.571	0.386	0.3156
EHD	0.243	0.229	0.386	0.4238
Proposed Descriptors				
CEDD	**0**	**0**	**0**	0.17766
FCTH	**0**	**0**	**0**	**0.13494**
C.CEDD	**0**	**0**	**0**	0.19363
C.FCTH	**0**	0.071	**0**	0.16677
		10200 Images		
MPEG-7 Descriptors				
SCD-32 Colors	0.271	0.643	0.729	0.48871
CLD	0.471	0.586	0.471	0.37966
EHD	0.243	0.257	0.471	0.49863
Other Descriptors				
RGB color histogram	0.229	0.500	0.729	0.54437
Tamura directionality	0.729	0.729	0.729	0.70434
Correlograms	**0**	0.471	**0**	0.35711
Proposed Descriptors				
CEDD	**0**	**0**	**0**	0.21220
FCTH	**0**	**0**	**0**	**0.17111**
C.CEDD	**0**	**0**	**0**	0.24509
C.FCTH	**0**	0.271	**0**	0.22403

The subsets and the queries are from various difficulty levels. The images used in every subset as well as the complete results are available online.[1]

As shown in Table 10, the proposed descriptors yield better results on the Nister database as well. In fact, the FCTH descriptor approaches perfect recall (ANMRR = 0.09463) on the first subset. As the number of the images involved in the search procedure increases, the MPEG-7 descriptor's ANMRR value also increases but the proposed descriptor's ANMRR is almost stable.

[1] http://orpheus.ee.duth.gr/anaktisi/results

7. Relevance Feedback Algorithm (RFA)

High retrieval scores in content-based image retrieval systems can be attained by adopting relevance feedback mechanisms. These mechanisms require the user to grade the quality of the query results by marking the retrieved images as being either relevant or not. Then, the search engine uses this grading information in subsequent queries to better satisfy users' needs. It is noted that while relevance feedback mechanisms were first introduced in the information retrieval field,[42] they currently receive considerable attention in the CBIR field. The vast majority of relevance feedback techniques proposed in the literature are based on modifying the values of the search parameters so that they better represent the concept consistent with the user's option. Search parameters are computed as a function of the relevance values assigned by the user to all the images retrieved so far. For instance, relevance feedback is frequently formulated in terms of the modification of the query vector and/or in terms of adaptive similarity metrics. Pattern classification methods such as SVMs have been used[53] in Relevance Feedback (RF) techniques.

Moreover, the user searching for a subset of images using the above descriptors, sometimes does not have a clear and accurate vision of these images. He/she has a general notion of the image in quest but not the exact visual depiction of it. Also, sometimes there is not an appropriate query image to use for retrieval. The proposed Automatic Relevance Feedback (ARF) algorithm attempts to overcome these problems by providing a mechanism to fine tune the retrieval results or to use a group of query images instead of one. The aforementioned is accomplished by manipulating the original query descriptor relying on the subsequent queries' images, while attempting to construct the ideal query descriptor.

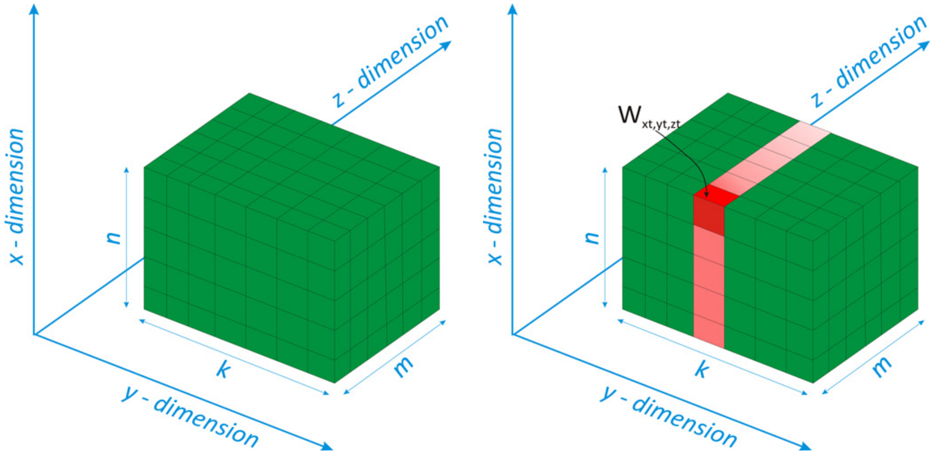
7.1. *The proposed automatic relevance feedback algorithm*

The goal of the proposed Automatic Relevance Feedback (ARF) algorithm is to optimally readjust or even change the initial retrieval results based on user preferences. During this procedure, the user selects from the first round of retrieved images one or more, as being relevant to his/her initial retrieval expectations. Information extracted from these selected images, is used to alter the initial query image descriptor.

Primarily, the initial image query one-dimensional descriptor is transformed to a three-dimensional (x, y, z) vector $W_{x,y,z}$ based on the inner features of the descriptor. The x, $x \in [1, n]$ dimension represents the texture where n is equal to the number of textures that the image descriptor contains. The y, $y \in [1, k]$ dimension corresponds to the dominant colors where k is equal to the number of dominant colors contained in each texture. The z, $z \in [1, m]$ dimension depicts the variation of dominant colors where m is equal to the maximum variation that each color has. Table 11 depicts the values of n, k and m for each proposed descriptor and Fig. 13 illustrates the vector. The advantage of the above transformation is easier access to the inner information of the descriptor through the x, y and z dimensions. For example, the extraction of

Table 11. The n, k, m values for each proposed descriptor.

	CEDD	FCTH	C.CEDD	C.FCTH
n	6	8	6	8
k	8	8	10	10
m	3	3	1	1



(a)

(b)

Fig. 13. (a) The three-dimensional vector $W_{x,y,z}$. (b) The alteration of the values of the vector element $W_{xt,yt,zt}$ and its associated elements.

the bin descriptor of the same variation (z axis) of a dominant color (y axis) for each different texture (x axis) is accomplished by holding the two dimensions (y, z) constant, while x dimension takes all its allowable values in the interval $[1, n]$. The transformation of the descriptor to the three-dimensional vector is based on the following equation:

$$i = (k \times m) \times x + m \times y + z \tag{15}$$

$$x = \left\lfloor \frac{i}{k \times m} \right\rfloor \tag{16}$$

$$y = \left\lfloor \frac{i - \lfloor \frac{i}{k \times m} \rfloor \times (k \times m)}{m} \right\rfloor \tag{17}$$

or

$$y = \left\lfloor \frac{i - x \times (k \times m)}{m} \right\rfloor \tag{18}$$

$$z = i - \left\lfloor \frac{i}{k \times m} \right\rfloor \times (k \times m) - \left\lfloor \frac{i - \left\lfloor \frac{i}{k \times m} \right\rfloor \times (k \times m)}{m} \right\rfloor \times m \tag{19}$$

or

$$z = i - x \times (k \times m) - y \times m \tag{20}$$

where i is the position of the bin inside the descriptor and x,y,z is the position of the same bin inside the three-dimensional vector $W_{x,y,z}$. Initially, the value of each vector element is equal to the value of the corresponding descriptor bin. When the user selects a relevant image from the retrieval results, each bin of that selected image's descriptor X_i updates the corresponding value of the $W_{x,y,z}$ vector in a Kohonen Self Organized Featured Map (KSOFM)[25,26] manner so that it moves closer to the new value emerging from X_i:

$$W_{xt,yt,zt}(t+1) = W_{xt,yt,zt}(t) + L(t) \times \left(X_{xt,yt,zt} - W_{xt,yt,zt}(t)\right) \tag{21}$$

where $X_{xt,yt,zt}$ is the transformed three-dimensional vector of the selected image query descriptor X_i based on Eq. (15).

Each time a user selects another relevant image, an epoch t starts. This epoch ends after all the elements of vector $X_{xt,yt,zt}$ of the selected relevant image are used to update the corresponding values of $W_{xt,yt,zt}$ according to Eq. (21).

$L(t)$ function utilizes the same philosophy as the KSOFM learning rate function and defines the rate of the vector element readjustment. It is not constant; instead decreases each time a new query image descriptor is presented:

$$L(t) = E_{\text{initial}} \times \left(\frac{E_{\text{Final}}}{E_{\text{Initial}}}\right)^{\frac{t}{t_{\max}}} \tag{22}$$

In the present work: $E_{\text{Initial}} = 0.4$, $E_{\text{Final}} = 0.001$, $t \in [0, 30]$ and $t\max = 30$. According to Eq. (22), $L(t)$ is a decreasing function, obtaining values in the interval E_{Initial} to E_{Final}.

Additionally, each one of the other vector elements Wxq, yq, zq (except the $W_{xt,yt,zt}$) also readjust their values based on the following equation:

$$W_{xq,yq,zq}(t+1) = W_{xq,yq,zq}(t) + L(t) \times h(xq, yq, zq) \times \left(X_i - W_{xt,yt,zt}(t)\right) \tag{23}$$

The $h(xq, yq, zq)$ function utilizes the same philosophy as the KSOFM neighborhood function and defines the readjustment rate of the associated descriptor bins:

$$h(xq, yq, zq) = \frac{k \times m}{100} \quad \text{where } yq = yt, \quad zq = zt \tag{24}$$

$$h(xq, yq, zq) = \frac{k}{100 \times |zt - zq|} \quad \text{where } xq = xt, \quad yq = yt \tag{25}$$

$$h(xq, yq, zq) = 0 \quad \text{anything else} \tag{26}$$

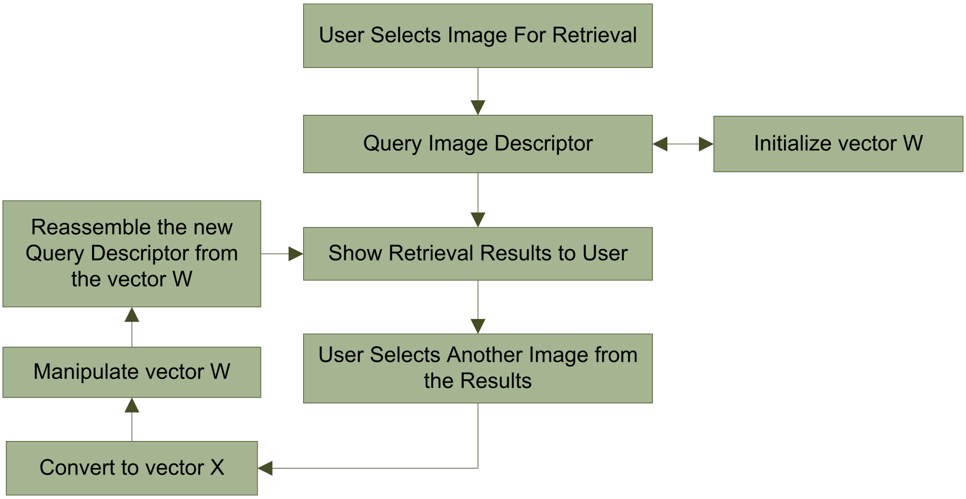

User Selects Image For Retrieval
Query Image Descriptor
Initialize vector W
Reassemble the new Query Descriptor from the vector W
Show Retrieval Results to User
Manipulate vector W
User Selects Another Image from the Results
Convert to vector X

Fig. 14. The flow-chart of the proposed ARF.

Equations (24)−(26) attempts to correct the descriptor errors (for example the quantization) as it readjusts the same color of the corresponding element $W_{xt,yt,zt}$ found within other texture areas (through x axis), and its other color variations found within the same texture area (through z axis) approaching the X_i value.

The readjustment rate of the colors belonging to the other textures is constant and depends on the number $(k \times m)$ of descriptor bins that a texture contains. The readjustment rate of the similar variants of the dominant color is not constant but rather decreases inversely proportional to the distance between the variant colors. Also, the rate depends on the amount of the dominant color (k) that a texture contains.

The final descriptor to query the image database is formed by the values of the three-dimensional vector $W_{x,y,z}$ using Eq. (15). The above procedure is repeated each time the user selects a relevant image. Figure 14 depicts the entire process of the proposed technique.

7.2. *Experimental results*

Table 12 illustrates the improvements achieved by the proposed Automatic Relevance Feedback algorithm for queries on the WANG Database after one, two and three repetitions. Table 13 illustrates the achieved improvements on the MPEG-7 CCD database. As shown from the results, the proposed method improves the retrieval scores significantly.

8. Conclusions and Discussion

In this paper, four descriptors that can be used in indexing and retrieval systems are proposed. The proposed descriptors are compact, varying in size from 23 to 74 bytes

Table 12. Results from the WANG image database.

Descriptor	Query					ANMRR
	204	327	522	600	703	
			Default results			
CEDD	0.314	0.127	0.347	0.059	0.115	0.25283
FCTH	0.235	0.114	0.323	0.026	0.092	0.27369
C.CEDD	0.316	0.140	0.452	0.069	0.088	0.30637
C.FCTH	0.320	0.224	0.493	0.013	0.116	0.31537
			First repetition			
RF Image	285	317	551	609	791	
CEDD	0.303	0.085	0.386	0.046	0.093	0.23332
FCTH	0.204	0.089	0.324	0.019	0.079	0.25443
C.CEDD	0.265	0.109	0.441	0.034	0.084	0.29229
C.FCTH	0.274	0.186	0.324	0.010	0.093	0.30220
			Second repetition			
RF Image	240	346	535	633	796	
CEDD	0.294	0.071	0.293	0.065	0.090	0.21341
FCTH	0.183	0.083	0.308	0.022	0.075	0.23442
C.CEDD	0.251	0.116	0.370	0.104	0.081	0.26887
C.FCTH	0.245	0.171	0.308	0.012	0.090	0.29776
			Third repetition			
RF Image	284	320	503	644	761	
CEDD	0.273	0.045	0.285	0.034	0.081	0.19776
FCTH	0.214	0.049	0.316	0.016	0.067	0.20834
C.CEDD	0.230	0.073	0.365	0.035	0.077	0.25336
C.FCTH	0.241	0.108	0.316	0.012	0.081	0.27557

per image. The descriptors' structure includes color and texture information. The experimental results show that the performance of the proposed descriptors is better than the performance of the similarly-sized MPEG-7 descriptors.

We propose two set of descriptors which leads to similar results. However, the FCTH descriptor and its related C.FCTH descriptor produce more robust results when retrieving images with many texture areas, however, they demand higher computational power and storage space than the CEDD and C.CEDD. On the other hand, the CEDD and its companion C.CEDD satisfactorily retrieve images with a small number of texture areas and their required computational power and storage space is noticeably lower. Therefore, the choice of descriptor depends on the type of images in the search procedure and on the computational requirements of the search.

Additionally, in the present paper an Automatic Relevance Feedback method is proposed. Though extremely simple to implement, the proposed method significantly improves image retrieval scores.

The proposed descriptors are designed for use in Internet image retrieval systems and on databases that frequently store elements for a very large number of images.

Table 13. Results from the MPEG-7 CCD image database.

Descriptor	Query			ANMRR
	i0121_add5	img00133_add3	img00438_s3	
		Default results		
CEDD	0	0	0.033	0.08511
FCTH	0	0.037	0.003	0.10343
C.CEDD	0	0.014	0.167	0.12655
C.FCTH	0	0.065	0	0.15977
		First repetition		
RF Image	i0123_add5	img00134_add3	img00444_s3	
CEDD	0	0	0.022	0.05334
FCTH	0	0.060	0	0.09883
C.CEDD	0	0.009	0.031	0.11341
C.FCTH	0	0.097	0.000	0.14333
		Second repetition		
RF Image	i26e_add1	img00131_add3	img00439_s3	
CEDD	0	0	0.033	0.03445
FCTH	0	0	0	0.08788
C.CEDD	0	0.005	0.008	0.10443
C.FCTH	0	0.009	0.000	0.12221
		Third repetition		
RF Image	0131_add5	img00135_add3	img00440_s3	
CEDD	0	0	0	0.03122
FCTH	0	0.014	0	0.07322
C.CEDD	0	0	0.006	0.10443
C.FCTH	0	0.009	0	0.11322

Such web-based image retrieval engines may need to execute retrieval on a few million images and must therefore use feature descriptors that are as compact as possible. The proposed descriptors meet these requirements.

All the proposed descriptors, img(Rummager) and img(Anaktisi) are programmed in C# and Java and are available as open source projects under the GNU General Public License (GPL).

Acknowledgments

This paper is part of the 03$E\Delta$375 research project, implemented within the framework of the Reinforcement Programme of Human Research Manpower (PENED) and co-financed by National and Community Funds (25%) from the Greek Ministry of Development — General Secretariat of Research and Technology and (75%) from the E.U. — European Social Fund.

Appendix A. The Fuzzy Inference Rules

Table 14. The fuzzy inference rules which bind the fuzzy ten-bin histogram.

IF INPUT *HUE* IS	AND INPUT *S* IS	AND INPUT *V* IS	THEN OUTPUT BIN IS
Any color	0	0	Black
Any color	1	0	Black
Any color	0	2	White
Any color	0	1	Gray
Red to orange	1	1	Red
Red to orange	1	2	Red
Orange	1	1	Orange
Orange	1	2	Orange
Yellow	1	1	Yellow
Yellow	1	2	Yellow
Green	1	1	Green
Green	1	2	Green
Cyan	1	1	Cyan
Cyan	1	2	Cyan
Blue	1	1	Blue
Blue	1	2	Blue
Magenta	1	1	Magenta
Magenta	1	2	Magenta
Magenta to red	1	1	Red
Magenta to red	1	2	Red

Table 15. The fuzzy inference rules which bind the fuzzy 24-bin histogram.

IF INPUT *S* IS	AND INPUT *V* IS	THEN OUTPUT BIN IS
1	1	Color
0	0	Dark color
0	1	Light color
1	0	Dark color

Table 16. The fuzzy inference rules which bind the fuzzy eight-bin texture histogram.

IF INPUT F_{HH} IS	AND INPUT F_{HL} IS	AND INPUT *FLH* IS	THEN OUTPUT BIN IS
0	0	0	Low energy linear
0	0	1	Low energy horizontal
0	1	0	Low energy vertical
0	1	1	Low energy horizontal and vertical
1	0	0	High energy linear
1	0	1	High energy horizontal
1	1	0	High energy vertical
1	1	1	High energy horizontal and vertical

Appendix B. Membership Functions Limits

Table 17. Fuzzy color system.

	Activation value			
Fuzzy ten-bin color system	0	1	1	0
Membership function *HUE*	Limits position			
Red to orange	0	0	5	10
Orange	5	10	35	50
Yellow	35	50	70	85
Green	70	85	150	165
Cyan	150	165	195	205
Blue	195	205	265	280
Magenta	265	280	315	330
Magenta to red	315	330	360	360
SATURATION				
0	0	0	10	75
1	10	75	255	255
VALUE				
0	0	0	10	75
1	10	75	180	200
2	180	200	255	255
	Activation value			
Fuzzy 24-bin color system	0	1	1	0
Membership Function	Limits position			
SATURATION				
0	0	0	68	188
1	68	188	255	255
VALUE				
0	0	0	68	188
1	68	188	255	255

Table 18. Fuzzy six-bin texture system.

	Activation value			
	0	1	1	0
Membership function	Limits position			
F_{LH} and F_{HL}				
0	0	0	20	90
1	20	90	255	255
F_{HH}				
0	0	0	20	80
1	20	80	255	255

References

1. P. Brodatz, *Textures: A Photographic Album for Artists and Designers* (Dover, New York, 1966).
2. S. A. Chatzichristofis and Y. S. Boutalis, CEDD — Color and edge directivity descriptor — A compact descriptor for image indexing and retrieval, *6th Int. Conf. Advanced Research on Computer Vision Systems ICVS 2008*, Santorini, Greece, 2008, pp. 312–322.
3. S. A. Chatzichristofis and Y. S. Boutalis, FCTH — Fuzzy color and texture histogram — A low level feature for accurate image retrieval, *9th Int. Workshop on Image Analysis for Multimedia Interactive Services*, Klagenfurt, Austria, 2008, IEEE Computer Society, pp. 191–196.
4. S. A. Chatzichristofis and Y. S. Boutalis, A hybrid scheme for accurate image retrieval based on color descriptors, *9th Int. Workshop on Image Analysis for IASTED Int. Conf. Artificial Intelligence and Soft Computing* (*ASC*), Palma De Mallorca, Spain, 2007, pp. 280–285.
5. S. A. Chatzichristofis, Y. S. Boutalis and Mathias Lux, IMG(RUMMAGER): An interactive content based image retrieval system, *2nd Int. Workshop on Similarity Search and Applications (SISAP)*, Czech Republic 2009, pp. 151–153.
6. S. W. Chee, K. P. Dong and S.-J. Park, Efficient use of MPEG-7 — Edge histogram descriptor, *ETRI J.* **24** (2002) 23–30.
7. I. Daubechies, Ten lectures on wavelets, *SPIE & IEEE Vis. Commun. Imag. Process.* Philadelphia, USA, 1992.
8. T. Deselaers, D. Keysers and H. Ney, Flexible image retrieval engine, *Multilingual Information Access for Text, Speech and Images, Fifth Workshop of the Cross-Language Evaluation Forum, CLEF 2004* (UK, 2005), pp. 688–689.
9. T. Deselaers, D. Keysers and H. Ney, Features for image retrieval: An experimental comparison, *Inform. Retri.* **11**(2) (2007) 77–107.
10. K. P. Dong, S. J. Yoon, S. W. Chee, S.-J. Park and S.-J. Yoo, A composite histogram for image retrieval, *IEEE Int. Conf. Multimedia and Expo* (New York City, USA, 2000), pp. 355–358.
11. G. Dorko, Selection of discriminative regions and local descriptors for generic object class recognition, Ph.D. thesis, Institut National Polytechnique de Grenoble, 2006.
12. H. Eidenberger, Evaluation of content-based image descriptors by statistical methods, *Multimedia Tools and Applications* **35** (2007) 241–258.
13. H. Eidenberger, How good are the visual MPEG-7 features? *SPIE & IEEE Visual. Communications and Image Processing*, Lugano, Switzerland, 1999, pp. 476–488.
14. C. Faloutsos *et al.*, Efficient and effective querying by image content, *J. Intell. Inform. Syst.* **3** (1994) 231–262.
15. X. Feng and Y.-J. Zhang, Evaluation and comparison of texture descriptors proposed in MPEG-7, *J. Vis. Commun. Imag. Represent.* **17**(4) (2006) 701–716.
16. G. Giacinto and F. Roli, Instance-based relevance feedback in image retrieval using dissimilarity spaces, *Case-Based Reasoning for Signals and Images*, ed. P. Perner (Springer-Verlag, 2007), pp. 419–430.
17. M. Grubinger and C. Leung, A benchmark for performance calibration in visual information search, *Int. Conf. Visual Information Systems VIS 2003*, Miami, FL, USA, 2003, pp. 414–419.
18. E. E. Gustafson and W. C. Kessel, Fuzzy clustering with a fuzzy covariance matrix, *IEEE CDC* (San Diego, California, 1979), pp. 761–766.

19. J. Huang, S. Ravi Kumar, M. Mitra, W.-J. Zhu and R. Zabih, Image indexing using color correlograms, *Conf. Computer Vision and Pattern Recognition* (*CVPR '97*) (IEEE Computer Society, 1997), p. 762.
20. P. W. Huang, S. K. Daia and P. L. Lin, Texture image retrieval and image segmentation using composite sub-band gradient vectors, *J. Vis. Commun. Image Represent.* **17**(5) (2006) 947–957.
21. Y.-P. Huang, T.-W. Chang and F. E. Sandnes, Efficient shape-based image retrieval based on gray relational analysis and association rules, *Int. J. Patt. Recogn. Artif. Intell.* **22**(4) (2008) 711–732.
22. Q. Iqbal and J. Aggarwal, CIRES: A system for content-based retrieval in digital image libraries, *Int. Conf. Control, Automation, Robotics and Vision* (Singapore, 2002), pp. 205–210.
23. ISO/IEC 15938-3, *Information Technology — Multimedia Content Description Interface/Part 3: Visual*, MPEG document, 2002.
24. ISO/IEC/JTC1/SC29/WG11, *Description of Core Experiments for MPEG-7 Color/ Texture Descriptors*, MPEG document N2929, Melbourne, 1999.
25. T. Kohonen, *Self-Organizing Maps*, 2nd edition (Springer-Verlag, Berlin, 1997).
26. T. Kohonen, The self-organizing map, *Proc. IEEE* **78**(9) (1990) 1464–1480.
27. K. Konstantinidis, A. Gasteratos and I. Andreadis, Image retrieval based on fuzzy color histogram processing, *Optics Commun.* **248**(4–6) 15 (2005) 375–386.
28. C. Lai, D. M. J. Tax and R. P. W. Duin, E. Pekalska and P. Paclik, A study on combining image representations for image classification and retrieval, *Int. J. Patt. Recogn. Artif. Intell.* **18**(5) (2004) 867–890.
29. T. M. Lehmann *et al.*, Automatic categorization of medical images for content-based retrieval and data mining, *Comput. Med. Imag. Graph* **29**(2) (2005) 143–155.
30. C. Leszek, MPEG-7 color descriptors and their applications, *Lecture Notes in Computer Science — Computer Analysis of Images and Patterns* (Springer, 2001), 11–20.
31. J. Li and J. Z. Wang, Automatic linguistic indexing of pictures by a statistical modeling approach, *J. Vis. Commun. Imag. IEEE Trans. Patt. Anal. Mach. Intell.* **25**(9) (2003) 1075–1088.
32. D. G. Lowe, Distinctive image features from scale invariant keypoints, *Int. J. Comput. Vis.* **60**(2) (2004) 91–110.
33. M. Lux and S. A. Chatzichristofis, LIRe: Lucene image retrieval — An extensible Java CBIR library, *ACM Int. Conf. Multimedia ACM MM'08* (Vancouver, British Columbia, Canada, 2008), pp. 1085–1087.
34. B. Mertzios and K. Tsirikolias, Logic filters: Theory and applications, *Nonlinear Image Processing*, Chapter 11, eds. S. Mitra and G. Sicuranza (Academic Press, 2004).
35. B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan and A. Yamada, Color and texture descriptors, *IEEE Trans. Circuits Syst. Vid. Technol.* **11**(6) (2001) 703–715.
36. B. S. Manjunath, P. Salembier, T. Sikora and P. Salembier, *Introduction to MPEG 7: Multimedia Content Description Language* (John Wiley and Sons).
37. N. Nikolaou and N. Papamarkos, Color image retrieval using a fractal signature extraction technique, *Eng. Appl. Artif. Intell.* **15** (2002) 81–96.
38. D. Nister and H. Stewenius, Scalable recognition with a vocabulary tree, *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)* Vol. 2, 2006, pp. 2161–2168.
39. A. Pentland, R. Picard and S. Sclaroff, Photobook. Content-based manipulation of image databases, *Int. J. Comput. Vis.* **18**(3) (1996) 233–254.
40. J. Puzicha, Y. Rubner, C. Tomasi and J. Buhmann, Empirical evaluation of dissimilarity measures for color and texture, *Int. Conf. Comput. Vision* (1999), pp. 1165–1173.

41. B. Russell, A. Torralba, K. Murphy and W. T. Freeman, LabelMe: A database and web-based tool for image annotation, *Int. J. Comput. Vis.* (2008), to appear.
42. G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval* (McGraw-Hill, New York, 1988).
43. G. Schaefer and M. Stich, UCID — An uncompressed colour image database, *9th Int. Workshop on Image Analysis for IASTED Int. Conf. Artificial SPIE, Storage and Retrieval Methods and Applications for Multimedia* (San Jose, USA, 2004), pp. 472–480.
44. J. Shotton, J. Winn, C. Rother and A. Criminisi, TextonBoost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation, *European Conf. Computer Vision* (*ECCV*) (Graz, Austria, 2006).
45. S. Siggelkow, M. Schael and H. Burkhardt, Search images by appearance, *DAGM 2001, 23rd DAGM Symp. Pattern Recognition*, Lecture Notes in Computer Science, Vol. 2191 (Munich, Germany, 2001), pp. 9–17.
46. H. Tamura, S. Mori and T. Yamawaki, Textural features corresponding to visual perception, *IEEE Trans. Syst. Man Cybern.* **8**(6) (1978) 460–472.
47. T. Tsai, Y.-P. Huang and T.-W. Chiang, A fast two-stage content-based image retrieval approach in the DCT domain, *Int. J. Patt. Recogn. Artif. Intell.* **22**(4) (2008) 765–781.
48. M. Unser, Texture classification and segmentation using wavelet frames, *IEEE Trans. Image Process.* **4**(11) (1995) 1549–1560.
49. J. Z. Wang, J. Li and G. Wiederhold, SIMPLIcity: Semantics- sensitive integrated matching for picture libraries, *IEEE Trans. Patt. Anal. Mach. Intell.* **23**(9) (2001) 947–963.
50. J. Winn, A. Criminisi and T. Minka, Object categorization by learned universal visual dictionary, *IEEE Int. Conf. Computer Vision (ICCV)* (Beijing, China, 2005).
51. K.-M. Wong, K.-W. Cheung and L.-M. Po, MIRROR: An interactive content based image retrieval system, *IEEE Int. Symp. Circuits and Systems* (Japan, 2005), pp. 1541–1544.
52. K. Zagoris, S. A. Chatzichristofis, N. Papamarkos and Y. S. Boutalis, IMG(ANAKTISI): A web content based image retrieval system, *2nd Int. Workshop on Similarity Search and Applications (SISAP)*, Czech Republic 2009, pp. 154–155.
53. X. S. Zhou and T. S. Huang, Relevance feedback in image retrieval: a comprehensive review, *Multimed. Syst.* **8**(6) (2003) 536–544.
54. H. J. Zimmerman, *Fuzzy Sets, Decision Making and Expert Systems* (Kluwer Academic Publications, Boston MA, 1987).

Savvas A. Chatzichristofis received the diploma in electrical and computer engineering in 2005 from the Democritus University of Thrace (DUTH), Greece. He is currently a research and teaching assistant and is studying towards the Ph.D. degree at the Department of Electrical and Computer Engineering, Democritus University of Thrace.

His research interests include document retrieval, image processing and analysis, document analysis, machine intelligence, neural networks, fuzzy logic and pattern recognition. He is a Member of the Cyprus Scientific and Technical Chamber since 2005, licentiate in the fields of electronics, information science and electrical mechanics.

Konstantinos Zagoris received the Diploma in electrical and computer engineering in 2003 from the Democritus University of Thrace, Greece. He is currently a research and teaching assistant and is studing towards the Ph.D. degree at the Department of Electrical and Computer Engineering, Democritus University of Thrace.

His research interests include document retrieval, color image processing and analysis, document analysis and pattern recognition. He is a member of the Technical Chamber of Greece.

Yiannis S. Boutalis received the diploma in electrical engineering in 1983 from the Democritus University of Thrace (DUTH), Greece and the Ph.D. in electrical and computer engineering (topic image processing) in 1988 from the Computer Science Division of National Technical University of Athens, Greece. Since 1996, he is serving as a faculty member, at the Department of Electrical and Computer Engineering, DUTH, Greece, where he is currently an Associate Professor and director of the Automatic Control Systems Lab.

He served as an assistant visiting professor at the University of Thessaly, Greece, and as a visiting professor in Air Defence Academy of General Staff of Air Forces of Greece.

He also served as a researcher in the Institute of Language and Speech Processing (ILSP), Greece, and as a managing director of the R&D SME Ideatech S.A, Greece, specializing in pattern recognition and signal processing applications.

His current research interests are focused in the development of computational intelligence techniques with applications in control, pattern recognition, signal and image processing problems.

Nikos Papamarkos is currently a Professor in the Department of Electrical and Computer Engineering at the Democritus University of Thrace. During 1987 and 1992, he has also served as a Visiting Research Associate at the Georgia Institute of Technology, USA.

His research interests include digital image processing, document analysis and recognition, computer vision, pattern recognition, neural networks, digital signal processing and optimization algorithm.

He has published more than 150 journal and conference papers. Professor Nikos Papamarkos is a Senior member of IEEE.