Efficient ACC using Binary Coding Stream for Color Descriptor

Anucha Tungkasthan

Department of Computer Technology, Faculty of Science and Technology, RMUTT, Thailand

anucha_t@rmutt.ac.th

Abstract

Auto color correlation (ACC) technique is proposed to enhance the availability of image content to capture local spatial correlation among different colors rather than using color correlogram technique. An ACC technique can reduce the size of color correlogram from $O(m2d)$ to $O(3md)$. However, it is still not applicable for query purposes in a large image database and especially in a real time image processing. This paper presents an application of a well studied image coding technique, namely block truncation coding (BTC), to reduce the storage space required and to increase the speed of retrieval for ACC algorithm. When an ACC is represented by using a binary matrix (BTC), it does not reduce the number of bins. Therefore, a decimal conversion of the binary stream in each row of the matrix is presented. The experimental results obtained from the two different techniques, Binary Coding Stream Conversion (BCSC) and ACC techniques, are investigated.

Keywords: Retrieval, Auto Color Correlation, Block Truncation Coding, Image Processing, Color Descriptor

1. Introduction

Content-Based Image Retrieval (CBIR) or Content-Based Visual Information Retrieval (CBVIR) is a computer vision application that automatically retrieves images based on their visual content. The feature extraction is a pre-processing step for image indexing in CBIR systems. There are various visual descriptors used to extract a low-level feature vector of an image. However, in this paper, we focus on color descriptors for retrieving images. Among the basic low-level image content description features used to enhance the availability of image and video contents, color is the most effective feature for conducting a visual multimedia analysis because it is an identifying feature that is local and independent of view and resolution [1], [2]. In addition, there are many examples from nature where color is used by animals and plants to send clear message of image indexing and retrieval [3]. Although,

several color description techniques have been proposed [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], they can be grouped into two classes based on whether or not they encode information related to the color spatial distribution. Examples of descriptors that do not include spatial color distribution include Color Histogram and Color Moments. Swian and Ballard were among the first who purposed the color histogram for object identification [4]. It is the most commonly used descriptor in image retrieval. Given a discrete color space defined by some color axes (e.g. red, green, blue), the color histogram is obtained by counting the number of times each color occurs in the image array. Examples of color descriptors that incorporate color spatial distribution methods include Color Coherence Vector (CCV) [1], Border/Interior Pixel Classification (BIC) [6], and Color Correlogram [3]. The CCV method uses a histogram-based method for comparing images that incorporate spatial information. The approach classifies each pixel in a given color bucket as coherent or incoherent, based on whether or not it is part of a large similarly-colored region and then uses a color coherence vector (CCV) to store the number of coherent versus incoherent pixels with each color by separating coherent pixels from incoherent pixels. CCV's provides finer distinctions than color histograms. Yi Tao and William I. Grosky [7] purposed a novel approach for spatial color indexing by using Delaunay triangulation, called a color anglogram. They used the HSI color model for image representation. In order to compute the color anglogram of an image, they first divide the image evenly into a number of M*N non-overlapping blocks. The feature point is labeled with its spatial location, dominant hue, and dominant saturation. For each set of feature points labeled with the same hue or saturation, they construct a Delaunay triangulation and then compute the feature point histogram by discretizing and counting the two largest angles produced by this triangulation.

Base on the literature, one of the most successful approaches to integrate the spatial information with the color histograms is the Color Correlogram (CC) [6]. The main aspects of this feature are: 1) it includes the spatial correlation of colors; 2) it can be used to describe the global distribution of local spatial correlation of colors; and 3) it is simple to implement. Even through many researchers proposed other spatial color descriptors nearly at the same period of time but they weren't extensively used when compared to color correlogram. For example, Lee el al [1], proposed spatial color descriptor based on whether or not smooth or edge pixel. If the center pixel in a 3x3 window

is an edge pixel, the global distribution of color pairs around the edges is represented by color adjacency histogram. Conversely, if the center pixel in a 3x3 window is smooth pixel, the color distribution is represented by a color vector angle histogram. Qiu [14] exploited the block truncation coding (*BTC*) for pre-processing before computing the color correlogram. Based on the literature, the correlogram techniques are still so useful today. For example, Young et al [14] presented combination of color and texture to increase the retrieval performance. As its color features, color autocorrelogram [6] (subset of color correlogram) of the hue and saturation component images in HSV color space are used. In addition, the correlogram technique not only has widely been used and applied for the color image retrieval but also can be applied to other applications such as object detection [15], object tracking [16], cut detection [1], [3]. However, the color correlogram is an expensive computation resulting in a computational time of $O(m^2d)$. We denoted m is the number of color in the image and the distance between two pixels $d \in [\min\{M, N\}]$ is fixed a priori. If we consider all the possible combinations of color pairs, the size of the color correlogram will be very large [17]. Therefore a simplified version of the feature called autocorrelogram with computation time of $O(md)$ is often used instead [6] but the efficiency in image retrieval using autocorrelogram is less than the efficiency of image retrieval in color correlogram. Based on a drawback of color correlogram and autocorrelogram techniques, the Auto Color Correlogram and Correlation (ACCC) technique is proposed [18] to overcome the disadvantage of traditional correlogram technique. It is the integration of Autocorrelogram (AC) and Auto Color Correlation (ACC) techniques [18]. It does not only capture the spatial correlation between the identical color, but also compute the local spatial correlation between color, while the size of ACCC is still $O(md)$. However, when an auto color correlation is constructed, the feature size of the RGB values of color bins in any distance k is still very large. Since the speed of retrieval and required storage space for the auto color correlation feature vector of database images depend on the size of the color bins and k distances, the proposed auto color correlation is still very inefficient when applied to a large database. Accordingly, a simple representation of an auto color correlation is clearly needed to increase the speed of retrieval and decrease the storage space required for the auto color correlation values of database images.

This paper is organized as follows. Section 2 presents the basic concept of traditional auto color correlation technique.

Section 3 proposes binary coding stream technique for decreasing the storage space required and increasing the speed of image retrieval. Section 4 presents the similarity measure metric for ACC binary coding stream. Section 5 discusses experimental results. Section 6 provides a summary and conclusions.

2. Auto Color Correlation Algorithm

This section describes a basic concept of traditional auto color correlation (ACC) technique before the proposed technique is demonstrated. An ACC expresses how to extract the important color information of all pixels of color C_j at a distance ***k-th*** from a pixel of color C_i in the image where color $C_i \neq C_j$. Formally, the ACC of image $\{I(x,y),x=1,2,...,M,y=1,2,...,N\}$ is defined as

$$ACC(i,j,k) = MC_j\gamma_{c_ic_j}^{(k)}(I) = \left\{ r_{mcj}\gamma_{c_ic_j}^{(k)}(I), g_{mcj}\gamma_{c_ic_j}^{(k)}(I), b_{mcj}\gamma_{c_ic_j}^{(k)}(I) \mid c_i \neq c_j \right\} \quad (1)$$

Where the original image $I(x,y)$ is quantized to m colors $C_1,C_2,...,C_m$ and the distance between two pixels $d \in [min\{M,N\}]$ is fixed a priori. Let MC_j is the color mean of the total number of color C_j from color C_i at distance ***k-th*** in an image I. The arithmetic mean colors are computed as follows:

$$r_{mc_j}\gamma_{c_ic_j}^{(k)}(I) = \frac{\Gamma_{c_i,rc_j}^{(k)}(I)}{\Gamma_{c_i,c_j}^{(k)}(I)} \mid c_i \neq c_j$$

$$g_{mc_j}\gamma_{c_ic_j}^{(k)}(I) = \frac{\Gamma_{c_i,gc_j}^{(k)}(I)}{\Gamma_{c_i,c_j}^{(k)}(I)} \mid c_i \neq c_j$$

$$b_{mc_j}\gamma_{c_ic_j}^{(k)}(I) = \frac{\Gamma_{c_i,bc_j}^{(k)}(I)}{\Gamma_{c_i,c_j}^{(k)}(I)} \mid c_i \neq c_j \quad (2)$$

Where denominator $\Gamma_{c_i,xc_j}^{(k)}(I)$ is the total of pixels values of color C_j at distance k from any pixel of color C_i when x is RGB color space and denoted $C_j \neq 0$. N is the number of accounting color C_j from color C_i at distance k, defined by:

$$N = \Gamma_{c_i,c_j}^{k}(I) = \left\{ \begin{array}{l} P(x_1,y_1) \in C_i \mid P(x_2,y_2) \in C_j; \\ k = \min\left\{ |x_1 - x_2|, |y_1 - y_2| \right\} \end{array} \right\} \quad (3)$$

In order to gain a deeper understanding of the ACC's computational procedure, it is described as follows.

Algorithm: Auto Color Correlation

```
For every K distance {
  For every X position
  For every Y position {
    Ci ← current pixel
    While (Cj ← Get neighbors pixel of Ci at distance K)
      For every color Cm {
        If (Cm = Ci and Ci ≠ Cj){
          countColor++
          colorR[Cm ] = colorR[Cm ] + colorRCj
          colorG[Cm ] = colorG[Cm ]+ colorGCj
          colorB[Cm ] = colorB[Cm ] + colorBCj
          }
      }
    }
}
meanColorR = sum ( colorR[Cm ])/countColor
meanColorG = sum ( colorG[Cm ])/countColor
```

meanColorB = sum (colorB[C_m])/countColor
}

Although ACC is able to find the local spatial correlation between color by reducing the size of color correlogram from $O(m^2d)$ to $O(3md)$, it is not still applicable for query purposes in a large image database and especially in a real time image processing. In the next section we will propose the technique to decrease the storage space required and increase the speed of retrieval. It can reduce the size of ACC from $O(3md)$ to $O(m)$.

3. The Proposed Binary Coding Stream Technique using BTC

This section presents the proposed binary coding stream technique, which is used for decreasing the storage space required and increasing the speed of image retrieval from an access database based on using the auto color correlation descriptor to extract the image characteristics. A constructed auto color correlation normally includes the mean values in RGB color space, a color correlation of color C_i and color C_j at distance k, when color C_i is not equal to color C_j. However, we only capture the dominant RGB peaks values in any color bins. Therefore, the proposed technique only compares the dominant elements, thereby significantly reducing the feature storage amount and speed of retrieval while process the similarity calculation of the two images. To capture dominant RGB peaks values of ACC feature vector, they are converted to binary, which the Block Truncation Coding (BTC) technique was applied in this process. Block truncation coding (BTC) is a relatively simple image coding technique developed in the early years of digital imaging more than 20 years ago[14]. Although it is a simple technique, BTC has played an important role in the history of digital image coding in the sense that many advanced coding techniques have been developed based on BTC or inspired by the success of BTC [8].The technique first computes the mean pixel value of the whole small block (4x4) and then each pixel in that block is compared to the block mean. If a pixel is greater than or equal to the block mean, the corresponding pixel position of the bitmap will have a value of 1, otherwise it will have a value of 0. The basic concept of BTC technique is shows in fig. 1. To applied with the ACC feature vector, When the RGB values in a color bin of ACC exceed a given threshold, that bin is classified as effective, otherwise, noneffective. If a bin is effective, a binary “1” is assigned, otherwise, a binary “0.” Accordingly, the threshold auto color correlation can be expressed as the following steps.

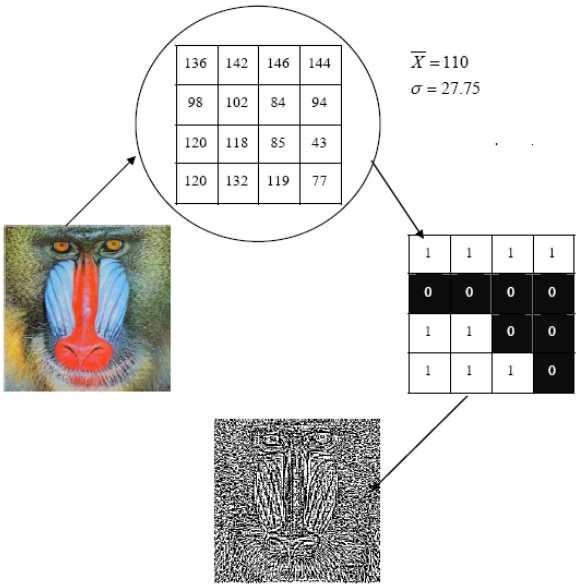


Fig. 1 The basic concept of BTC in Image compression.

Formally, Let $B=\{r(i, j), g(i, j), b(i, j), i=1, 2,\ldots,m, j=1, 2,\ldots,k\}$ be an m x k ACC feature block in RGB color space. An auto color correlation feature vector normally includes many of empty color bins (a color that is not appears in an image). Therefore, the average value will often produce incorrect results. Let $B'=\{r'(i, j), g'(i, j), b'(i, j), i=1, 2,\ldots,m', j=1, 2,\ldots,k'\}$ be an m' x k' ACC feature vector block that contain appears color in an image. Let $Z = \{ib(i, j), i=1, 2,\ldots,m', j=1, 2,\ldots,k'\}$ be the inter-band average ACC block, then we have

$$ib(i, j) = \frac{1}{3}(r'(i, j) + g'(i, j) + b'(i, j)) \quad \forall(i, j) \quad (4)$$

The threshold is computed as the mean of Z, i.e

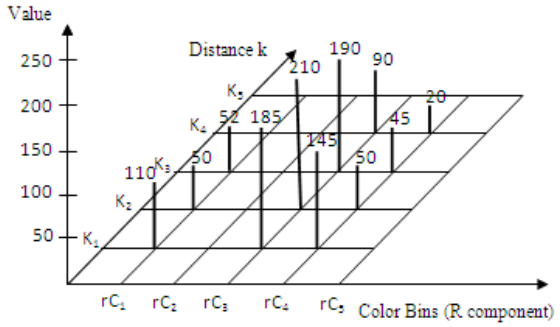
$$T = \frac{1}{m' \times k'} \sum_{i=1}^{m'} \sum_{j=1}^{k'} ib(i, j), \quad (5)$$

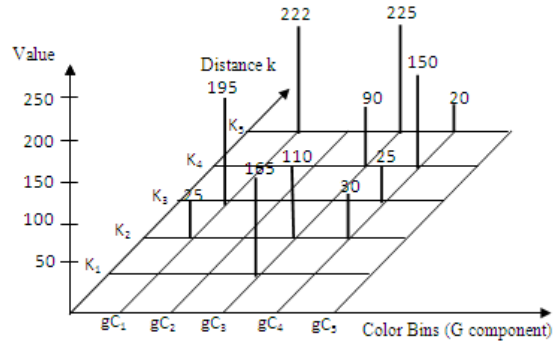
The Binary bit block $\{bk(i, j), i =1, 2,\ldots,m, j=1,2,\ldots,k\}$ is computed as

$$bk(i, j) = \begin{cases} 1 & if \quad ib(i, j) \geq T \\ 0 & if \quad ib(i, j) < T \end{cases} \quad (6)$$

After the creation of the bit block, the following step is to preserve the two sample moments of a block (m x k). They consist of the average as the threshold and standard deviation as shown in equation (5) and (7), respectively:

$$SD : \sigma = \sqrt{\frac{\sum_{i=1}^{m'} \sum_{j=1}^{k'} (ib(i, j) - T)}{m' \times k'}} \quad (7)$$





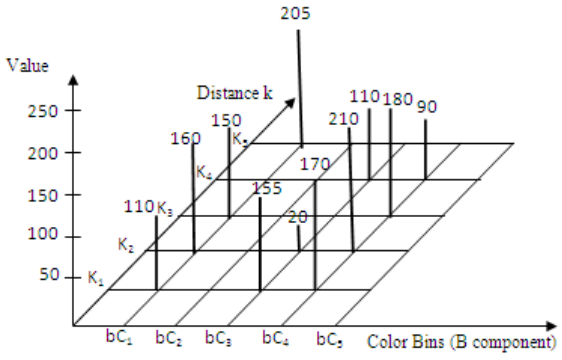


Fig. 2 The ACC feature vector in RGB color space

If we define a 1-bit (2-level quantizer: x+ and x-) with xTh, such that preserve the two moments. The threshold will be the mean value (xave). For a block we transmit bit-level matrix, xsd and xave. The levels x+ and x- then can be determined by setting up the expressions that equate (preserve) the moments before and after quantization according to equation (6). Let $n^+ = (n_r^+, n_g^+, n_b^+)$ and $n^- = (n_r^-, n_g^-, n_b^-)$ are the number of ACC feature vector above and below the threshold (mean). The $x^- = (x_r^-, x_g^-, x_b^-)$ and $x^+ = (x_r^+, x_g^+, x_b^+)$ can be computed as follow:

$$x_r^- = T - \sigma\sqrt{\frac{n_r^+}{n_r^-}},\ x_g^- = T - \sigma\sqrt{\frac{n_g^+}{n_g^-}},\ x_b^- = T - \sigma\sqrt{\frac{n_b^+}{n_b^-}} \quad (8)$$

$$x_r^+ = T + \sigma\sqrt{\frac{n_r^-}{n_r^+}},\ x_g^+ = T + \sigma\sqrt{\frac{n_g^-}{n_g^+}},\ x_b^+ = T + \sigma\sqrt{\frac{n_b^-}{n_b^+}} \quad (9)$$

$$B_r = \begin{bmatrix} 110 & 50 & 52 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 185 & 210 & 190 & 90 & 0 \\ 145 & 50 & 45 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B_r' = \begin{bmatrix} 110 & 50 & 52 & 0 & 0 \\ 185 & 210 & 190 & 90 & 0 \\ 145 & 50 & 45 & 20 & 0 \end{bmatrix}$$

$$B_g = \begin{bmatrix} 0 & 25 & 195 & 0 & 222 \\ 0 & 0 & 0 & 0 & 0 \\ 165 & 110 & 0 & 90 & 225 \\ 0 & 30 & 25 & 150 & 20 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B_g' = \begin{bmatrix} 0 & 25 & 195 & 0 & 222 \\ 165 & 110 & 0 & 90 & 225 \\ 0 & 30 & 25 & 150 & 20 \end{bmatrix}$$

$$B_b = \begin{bmatrix} 110 & 160 & 150 & 0 & 205 \\ 0 & 0 & 0 & 0 & 0 \\ 155 & 20 & 0 & 110 & 0 \\ 170 & 210 & 180 & 90 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B_b' = \begin{bmatrix} 110 & 160 & 150 & 0 & 205 \\ 155 & 20 & 0 & 110 & 0 \\ 170 & 210 & 180 & 90 & 0 \end{bmatrix}$$

(a)

$$bk_r = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad bk_g = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$bk_b = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(b)

Fig. 2. The Matrix of m x k Block represent the ACC feature vector in RGB Color Space (a) m x k to m'x k' in R component and (b) a binary block of ACC in RGB space

It is clear from above descriptions of ACC bit block that the BTC code for each RGB block consists of two color indices and a binary bit Block. Note that this paper uses the x^+ only. From a Fig. 2, the matrix of m x k block in R component and can be represented by Fig. 2(a) where the columns represent the k distance of the neighboring pixels and the rows represent

the colors in an image. A constructed auto color correlation normally includes many of empty color bins. Therefore, the average value will often produce incorrect results. In this case the Color C_2 and C_5 is not contain in an image. The threshold is computed as the mean of ib in equal (5).

Yet, even when an auto color correlation is represented by a binary matrix, this does not reduce the number of bins. Therefore, to reduce the number of bins, a decimal conversion of the binary stream [1] in each row of the matrix is performed. As shown in Fig.. 3, a binary matrix can be converted into a column matrix, where each element is equivalent to a decimal number corresponding to the binary stream of each row in a binary matrix.

$$C_r = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} 16 \\ 0 \\ 30 \\ 16 \\ 0 \end{matrix} \qquad C_r = \begin{bmatrix} 16 & 0 & 30 & 16 & 0 \end{bmatrix}^T$$

$$C_r = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} 16 \\ 0 \\ 30 \\ 16 \\ 0 \end{matrix} \qquad C_r = \begin{bmatrix} 16 & 0 & 30 & 16 & 0 \end{bmatrix}^T$$

$$C_r = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} 16 \\ 0 \\ 30 \\ 16 \\ 0 \end{matrix} \qquad C_r = \begin{bmatrix} 16 & 0 & 30 & 16 & 0 \end{bmatrix}^T$$

Fig. 3. Decimal number conversion.

4. Similarity Measure

The type of similarity measure to be considered depends on the technique used for feature extraction. When measuring the similarity of binary codes for auto color correlation, the intersection technique is used, which measures the similarity of the binary codes for the same color between the query and model images.

Let $Bc_m(I) = b_1^r, b_2^r, ..., b_m^r; b_1^g, b_2^g, ..., b_m^g; b_1^b, b_2^b, ..., b_m^b$ denote the binary code of auto color correlation colors to color Cm in RGB space of query image I, then the intersection result of query image and model image concerning color Cm is calculated as

$$S_{mr}(I,J) = \frac{NC_{mr}^1(I,J)}{N_{mr}^1(I) + N_{mr}^1(J) - NC_{mr}^1(I,J)} \cdot \left(1 - \frac{|x_r^+(I) - x_r^+(J)|}{x_r^+(I) + x_r^+(J)}\right)$$

$$S_{mg}(I,J) = \frac{NC_{mg}^1(I,J)}{N_{mg}^1(I) + N_{mg}^1(J) - NC_{mg}^1(I,J)} \cdot \left(1 - \frac{|x_g^+(I) - x_g^+(J)|}{x_g^+(I) + x_g^+(J)}\right)$$

$$S_{mb}(I,J) = \frac{NC_{mb}^1(I,J)}{N_{mb}^1(I) + N_{mb}^1(J) - NC_{mb}^1(I,J)} \cdot \left(1 - \frac{|x_b^+(I) - x_b^+(J)|}{x_b^+(I) + x_b^+(J)}\right) \quad (10)$$

Where $N_m^1 = \{N_{mr}^1, N_{mg}^1, N_{mb}^1\}$ is the total number of binary "1" in binary string and $NC_m^1 = \{NC_{mr}^1, NC_{mg}^1, NC_{mb}^1\}$ the total number of binary "1"s occurring at the same position in the two binary string. The last term is threshold weighting value of binary "1" in RGB space in order to reduce the bias of a threshold level. For example,

if $B_{C_m}(I) = \{B_{Cr}(I), B_{Cg}(I), B_{C_b}(I)\} = [0 \; 0 \; 0 \; 0 \; 1]$, $[0 \; 0 \; 1 \; 0 \; 1]$, $[0 \; 1 \; 1 \; 0 \; 1]$ and $B_{C_m}(J) = \{B_{Cr}(J), B_{Cg}(J), B_{C_b}(J)\} = [0 \; 1 \; 0 \; 1 \; 1]$,

$[0\ \ 0\ \ 1\ \ 1\ \ 1]$, $[0\ \ 1\ \ 1\ \ 0\ \ 1]$ correspond to the binary codes of color C_m in images I and J, respectively, then $N^1_{mr}(I)=1, N^1_{mg}(I)=2, N^1_{mb}(2)=3$; $N^1_{mr}(J)=3, N^1_{mg}(J)=3, N^1_{mb}(J)=3$ and $NC^1_{mr}(I,J)=1, NC^1_{mg}(I,J)=2, NC^1_{mb}(I,J)=3$ are obtained. Accordingly, the intersection result of $S_{mr}(I,J)$, $S_{mr}(I,J)$, $S_{mr}(I,J)$ are 0.33, 0.66, and 1 , respectively. For all m colors used in the construction of an auto color correlation, the total intersection is computed as

$$S_{\forall m}(I,J)=\frac{1}{3m}\sum_{\forall m}(S_{mr}(I,J)+S_{mg}(I,J)+S_{mb}(I,J)) \quad (11)$$

5. Experimental Results

To evaluate the accuracy of the retrieval process based on using the proposed technique, we measured the image retrieval performance and compared the results obtained from the two techniques, the proposed technique and the traditional ACC technique. The two techniques were tested by using the same database of 3000 color JPEG images with a size of 140 x 100 pixels from Corel stock photographs. Sixty-four colors and a set of spatial distance {1, 3, 5, 7, 9} were used in computation for the two algorithms in this experiment. The metrics that we used for measuring the accuracy of queries are ***r-measure*** and ***p_1-measure***, where ***r-measure*** is the sums up of the rank of correct answer from all queries and average ***r-measure*** is the ***r-measure*** divided by the number of queries ***q***. ***p_1-measure*** is the sum of the precision with the recall equal to 1 and the average ***p_1-measure*** is the ***p_1-measure*** divided by *q*. ***r-measure*** and ***p_1-measure*** are defined as

$$r-measure=\sum_{i=1}^{q}Rank(Q_i) \quad (12)$$

$$Avg\ r-measure=\frac{r-measure}{q} \quad (13)$$

$$P_1-measure=\sum_{i=1}^{q}\frac{1}{Rank(Q_i)} \quad (14)$$

$$Avg\ p_1-measure=\frac{p_1-measure}{q} \quad (15)$$

The experimental results are shown in Table I. In order to compare the retrieving performance of two techniques, we also formulated the hypothesis based on the rank of the correct answer by using the statistical t-test. In order to measure the significance of the rank of correct answer obtained by our proposed technique, we did a t-test on the fifty queries for retrieving images. The rank of correct answer of traditional ACC and proposed technique were 6.28 ±8.4 and 6.5 ±7.3, respectively. Using the t-test to compare the means of two independent queries, the P-values obtained from the t-test of the existing ACC technique versus the proposed technique are 1.698e-25. A statistical test showed that there were no statistically significant results for the rank of the correct answer obtained from the proposed technique and the traditional ACC technique. Using binary coding

stream conversion with an ACC feature vector in RGB color space can decrease the storage space required and instead increase the speed of retrieval. Moreover, the accuracy of retrieval process was not significantly different when comparing with the traditional ACC technique.

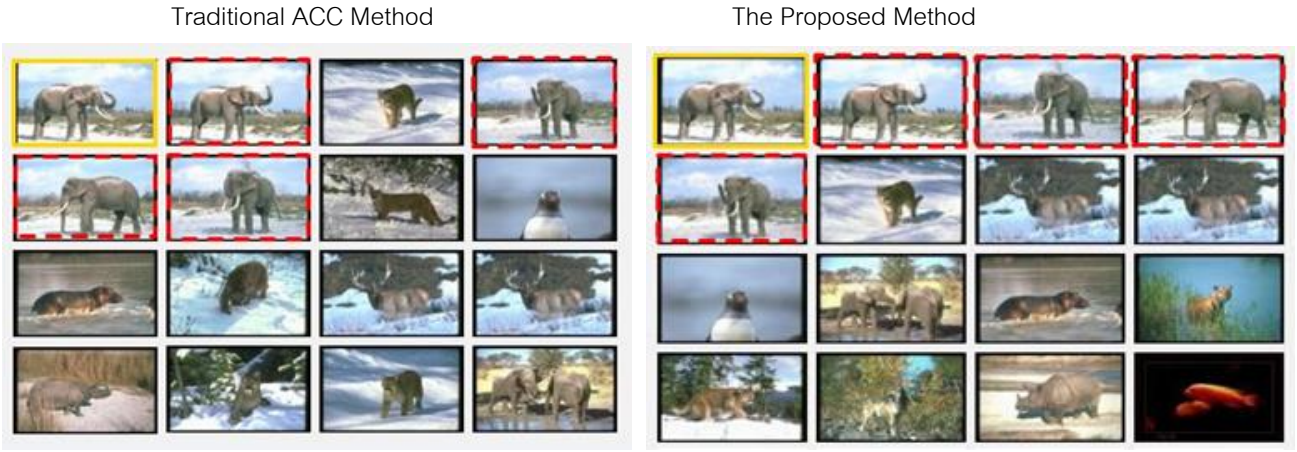


Fig. 3. The retrieving performance on query image with elephant (a) the average rank is 3.25, (b) the average rank is 2.5

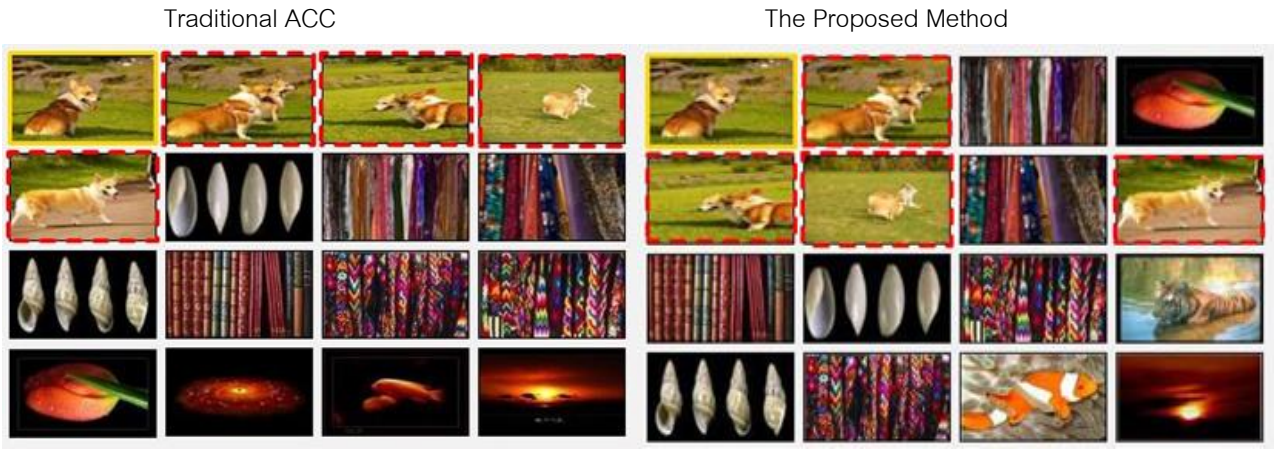


Fig. 4. The retrieving performance on query image with dog (a) the average rank is 2.5, (b) the average rank is 4.25

Fig. 5. The retrieving performance on query image with elephant (a) the average rank is 4, (b) the average rank is 4.5

Table 1. The Average Rank of the Two Methods

Query Image	Average Rank (Rank Qi)	
	Traditional ACC	Proposed Method
1	**3.25**	**2.5**
2	3.5	5.25
3	3.5	3
4	4	5.5
7	4.25	3.5
8	4.67	4.25
9	2.5	4.25
10	**5.67**	**3**
11	2.5	3.5
12	3	2
13	3.67	4.5
14	5.4	5.67
15	5.33	7
16	6.25	6.5
17	**4**	**4.5**
18	8	9.5
19	5.33	3
20	2.5	3
21	7.5	8.3
22	3.5	3
23	3	3
24	3.5	4.25
25	7	6
26	3.5	4
27	2.5	3
28	5	5.25
29	6	7
30	5.5	4.87

Table 2. Performances of the Two Methods

Method	Traditional ACC	Proposed Technique
r-measure	314	320
Avg r- measure	6.28	6.5
p1-measure	74	72
Avg p1-measure	0.76	0.73

5. Conclusion

This paper presented a technique of using a well known image coding technique to achieve the storage space required and to increase the retrieval speed when using an auto color correlation descriptor to extract the image characteristics. We used the BTC coding described in this paper to construct the ACC binary block. After that a decimal conversion of the binary stream in each row of the matrix was performed in order to reduce the number of bins. To measure the similarity of binary codes between the query and model images, the intersection technique was applied. Based on the experimental results, using binary coding with a

feature vector of ACC in RGB color space, the accuracy of retrieval process was not significantly different when comparing with the traditional ACC technique. The proposed technique is suitable for real-time image processing.

References

[1] Y. H. Lee, K. H. Lee, H. Y. Ha, Senior Member IEEE, "Spatial Color Descriptor for Image Retrieval and Video Segmentation," IEEE Trans. Multimedia. 5(3), pp. 358–367, 2003.

[2] R. Schettini , G. CIOCCA , S. Zuffi, "A Survey Of Methods For Colour Image Indexing And Retrieval In Image Databases," Proc. Schettini01 a survey, 2001.

[3] M. Swain, D. Ballard, "Color Indexing. International Journal of Computer Vision," 7(1), pp 11–32, 1991.

[4] M. A. Stricker and M. Orengo, "Similarity of color images," Proc. SPIE, Storage Retrieval Still Image Video Databases IV, vol. 2420, pp. 381–392, 1996.

[5] Greg P., Ramin Z., Justin M, "Comparing Images Using Color Coherence Vectors," Proc. ACM on Multimedia, pp. 65-73, 1997.

[6] J. Huang, S.R. Kumar, M. Mitra, and Z. Wei-Jing, "Spatial Color Indexing and Applications," Proc. Sixth International Conference on Computer Vision, pp. 606 – 607, 1998.

[7] R. Alaoui, S. Ouatik E. Alaoui and M. Meknassi, "Spatial Color Indexing: An Efficient and Robust Technique for Content-Based Image Retrieval," Journal of Computer Science, 5 (2), pp. 109-114, 2009.

[8] G. Qiu, "Color Image Indexing using BTC, "IEEE Trans. Image Processing, 12(1), pp. 93–101, 2003.

[9] O. S. Renato, A. N. Mario, X. F. Alexandre, "A Compact and Efficient Image Retrieval Approach Based on Border/Interior Pixel Classification," Proc Information and Knowledge Management, pp. 102-109, 2002.

[10] Yi T., William I. G. : Spatial Color Indexing: A Novel Approach for Content-Based Image Retrieval. In: Proc. ICMCS. pp. 530 – 535 (1999)

[11] Yi T., William I. G. : Content-based image retrieval using joint correlogram. Springer. Multimedia Tools and Applications, 34(2), pp. 239-248 (2007)

[12] G. Pass, R. Zabih, "Comparing images using joint histograms," Springer. Multimedia Systems, 7(3), pp. 234-240, 1999.

[13] S. chinlek and W. Premchaiswade, "Image retrieval using AC/CDC," Proc. ISCIT 2001, 2001.

[14] Y. D. Chun, N. C. Kim, Member, IEEE, and I. H. Jang, Member, IEEE, "Content-Based Image Retrieval Using Multiresolution Color and Texture Features," IEEE Trans. on Multimedia, 10(6), 2008.

[15] J.B. Luo, D. Crandall, "Color Object Detection Using Spatial-Color Joint Probability Functions," IEEE Transactions on Image Processing, 15(6), pp. 1443-1453, 2006.

[16] Q. Zhao and H. Tao, "Object Tracking using Color Correlogram," Proc.SVPTES, pp. 263 – 270, 2006.

[17] H. Permuter, J. Francos, and I. H. Jermyn, "Gaussian mixture models of texture and colour for image database retrieval" Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing, Hong Kong, vol. 3, pp. 569–572, 2003.

[18] A. Tungkasthan, S. Intarasema, W. Premchaiswadi, "Spatial color indexing using ACC algorithm," 2009 7th International Conference on ICT and Knowledge Engineering, 1-2 Dec. 2009, 113 – 117.