

CBIR System using Color Moment and Color Auto-Correlogram with Block Truncation Coding

Vandana Vinayak
Computer Science and Engineering
SBSSTC, Ferozepur

Sonika Jindal
Professor
Computer Science and Engineering
SBSSTC, Ferozepur

ABSTRACT

In content-based Image Retrieval (CBIR) application, a large amount of data is processed. Among various low-level features like color, shape and texture, color is an important feature and represented in the form of histogram. It is essential that features required to be coded in such a way that the storage space requirement is low and processing speed is high. In this paper, we propose a method for indexing of images in the large database with lossy compression technique known as Block Truncation Coding (BTC) along with two different color feature extraction methods - Color Moment and Color Auto-correlogram. Block truncation coding divided the original image into multiple non-overlapping blocks and then retrieve the required features. The proposed method performs better.

Keywords

CBIR, Color Moment, Color Auto-Correlogram, Block Truncation Coding

1. INTRODUCTION

From the last few decades, Content based Image Retrieval is a very vast research area in the field of Image processing. In CBIR, visual features such as shape, color and texture are extracted to characterize images. Each of the features is represented using one or more feature descriptors. It is essential that features required to be coded in such a way that the storage space requirement is low and processing speed is high [13]. During the retrieval, features and descriptors of the query are compared to those of the images in the database in order to rank each indexed image according to its distance to the query.

Due to the rapid development of imaging technology, more and more images are available and stored in large databases. Searching the related images by the querying image is becoming very difficult. Most of the images on the web are compressed. This paper presents an efficient content-based image indexing technique for searching similar images using block truncation coding along with Color moment and correlogram. Experimental results demonstrate its superiority with the existing techniques.

The color indexing work of Swain and Ballard [16], which is based on color histograms, has demonstrated the potential of using color for indexing. Stricker and Orengo [15] have shown that moment based color distribution features can be matched more

robustly than color histograms as histograms do not capture spatial relationship of color regions and thus, they have limited discriminating power. The system presented in [8] integrates mostly the algorithms introduced in into a database environment. Reasonable results can be achieved using the above mentioned algorithms, but it is clear that the false positives which are retrieved result from the lack of spatial information in the index. The simplest way to store spatial information in the index is to divide the image into sub-images and then extract the color features for each sub-image.

Color correlogram and color moments perform better than traditional color histograms when used for content-based image retrieval. However, they require very expensive computation. Color moments have been successfully used in content based image retrieval systems. It has been shown that characterizing one dimensional color distributions with the first three moments is more robust and runs faster than the histogram based methods. Hence, in our proposed method, Color moments are used for extraction of color features.

A color retrieval method based on the primitives of color moments is proposed in [14]. After dividing an image into several blocks, the color moments of all blocks are extracted and clustered into several classes based on a fast non-iterative clustering algorithm. The mean vector of each class is considered as a primitive of the image and all primitives are used as feature vectors. Two test databases from Corel were used and compared the performances of the proposed method with other existing ones. The experimental results showed that the proposed method is usually better than others.

Choras [11] proposed an integrated color, texture and shape feature extraction method in which Gabor filtration is used for determining the number of regions of interest (ROIs). They calculated texture and color features from the ROIs based on threshold Gabor features and histograms, color moments in YUV space, and shape features based on Zernike moments. The features presented proved to be efficient in determining similarity between images.

Xue and Wanjun [17] proposed a method in which the feature extraction methods of color histograms and color moments are integrated to extract the color feature. They reported that the recall and precision had been improved and the index sorting was better.

A method based on color and texture features is proposed in [18]. As color feature, they use color moments of the Hue, Saturation and Value (HSV) of the image and Gabor descriptors are adopted as texture features. They assigned weights to each feature and calculated the similarity with combined features of color and texture using normalized Euclidean distance. They reported that the proposed method has higher retrieval accuracy than the conventional method using color and texture features. The feature dimension is also lower than the conventional methods.

2. FEATURE EXTRACTION

2.1 Color Feature Extraction

The goal of color indexing is to retrieve all the images whose color compositions are similar to the color composition of the query image. Histograms are useful because they are relatively insensitive to position and orientation changes and they are sufficiently accurate [16]. However, they do not capture spatial relationship of color regions and thus, they have limited discriminating power. Color Moment and Color Correlogram techniques performs better than traditional color histograms when used for content-based image retrieval. Color moments have been successfully used in content based image retrieval systems. It has been shown [15] that characterizing one dimensional color distributions with the first three moments is more robust and runs faster than the histogram based methods. Hence, in our proposed method, Color moments are used along with color correlogram for extraction of color features.

2.1.1 Color Moment. Color moments are mainly used for color indexing purposes as features in image retrieval applications in order to compare how similar two images are based on color. Usually one image is compared to a database of digital images with pre-computed features in order to find and retrieve a similar image. Each comparison between images results in a similarity score, and the lower this score is the more identical the two images are supposed to be. Color moments are scaling and rotation invariant. It is usually the case that only the first three color moments are used as features in image retrieval applications as most of the color distribution information is contained in the low-order moments [4] Since color moments encode both shape and color information they are a good feature to use under changing lighting conditions. Color moments can be computed for any color model. Three color moments are computed per channel (e.g. 9 moments if the color model is RGB and 12 moments if the color model is CMYK). Computing color moments is done in the same way as computing moments of a probability distribution.

Mean

The first color moment can be interpreted as the average color in the image, and it can be calculated by using the following formula:

$$E_i = \sum_{j=1}^N \frac{1}{N} p_{ij} \quad (1)$$

where N is the number of pixels in the image and p_{ij} is the value of the jth pixel of the image at the ith color channel.

Standard Deviation

The second color moment is the standard deviation, which is obtained by taking the square root of the variance of the color distribution. it can be calculate as:

$$\sigma_i = \sqrt{\left(\frac{1}{N} \sum_{j=1}^N (p_{ij} - E_i)^2\right)} \quad (2)$$

where E_i is the mean value, or first color moment, for the ith color channel of the image.

Skewness

The third color moment is the skewness. It measures how asymmetric the color distribution is, and thus it gives information about the shape of the color distribution. Skewness can be computed with the following formula:

$$s_i = 3\sqrt{\left(\frac{1}{N} \sum_{j=1}^N (p_{ij} - E_i)^3\right)} \quad (3)$$

Color Indexing

A function of the similarity between two image distributions is defined as the sum of the weighted differences between the moments of the two distributions [7]. It can be calculated as:

$$d_{moment}(H, I) = \sum_{i=1}^r w_{i1} |E_i^1 - E_i^2| + w_{i2} |\sigma_i^1 - \sigma_i^2| + w_{i3} |s_i^1 - s_i^2| \quad (4)$$

Where:

(H, I) : are two image color distributions being compared

i : is the current channel index (e.g. 1 = R, 2 = G, 3 = B)

r : is the number of channels (e.g. 3)

E_i^1, E_i^2 : are the first moments (mean) of the two image distributions

σ_i^1, σ_i^2 : are the second moments (std) of the two image distributions

s_i^1, s_i^2 : are the third moments (skewness) of the two image distributions

w_i : are the weights for each moment (described below)

2.1.2 Color Auto-Correlogram. Correlogram can be stored as a table indexed by pairs of colors (i,j) where d-th entry shows the probability of finding a pixel j from pixel i at distance d. Whereas an auto-correlogram can be stored as a table indexed by color i where d-th entry shows the probability of finding a pixel i from the same pixel at distance d. Hence auto-correlogram shows the spatial correlation between identical colors only [3].

Let [D] denote a set of D fixed distances $d \in D$. Then the correlogram of the image I is defined for level pair g_i, g_j at a distance d is given by -

$$\gamma_{g_i, g_j}^d(I) \equiv Pr_{p_1 \in I_{g_i}, p_2 \in I_{g_j}} [p_2 \in I_{g_j} || p_1 - p_2 = d] \quad (5)$$

Which gives the probability that given any pixel p1 of level g_i , a pixel p2 at a distance d in certain direction from the given pixel p1 is of level g_j .

Autocorrelogram captures the spatial correlation of identical levels only [1]:

$$\alpha_g^d(I) = \gamma_{g_i, g_j}^d(I) \quad (6)$$

It gives the probability that pixels p1 and p2, d away from each other, are of the same level g_i

2.2 Texture Feature Extraction

Texture is an important feature of natural images. A variety of techniques have been developed for measuring texture similarity. Most techniques rely on comparing values of what are known as second-order statistics calculated from query and stored images. These methods calculate measures of image texture such as the degree

of contrast, coarseness, directionality and regularity [6]; or periodicity, directionality and randomness. Alternative methods of texture analysis for image retrieval include the use of Gabor filters [5] and fractals. Gabor filter (or Gabor wavelet) is widely adopted to extract texture features from the images for image retrieval, and has been shown to be very efficient. Manjunath and Ma [5] have shown that image retrieval using Gabor features outperforms that using Pyramid-structured wavelet transform (PWT) features, tree-structured wavelet transform (TWT) features and multiresolution simultaneous autoregressive model (MRSAR) features. Hence, in our proposed method, Gabor filter is used for extraction of texture features.

2.2.1 Gabor Filter. A Gabor filter is a linear filter whose impulse response is defined by a harmonic function multiplied by a Gaussian function. Because of the multiplication-convolution property (Convolution theorem), the Fourier transform of a Gabor filter's impulse response is the convolution of the Fourier transform of the harmonic function and the Fourier transform of the Gaussian function. Gabor filters are directly related to Gabor wavelets, since they can be designed for number of dilations and rotations. Gabor filters have been used in many applications, such as texture segmentation, target detection, fractal dimension management, document analysis, edge detection, retina identification, image coding and image representation. A Gabor filter can be viewed as a sinusoidal plane of particular frequency and orientation, modulated by a Gaussian envelope.

$$h(x, y) = s(x, y)g(x, y) \quad (7)$$

where

$s(x, y)$: Complex sinusoid

$g(x, y)$: 2-D Gaussian shaped function, known as envelope

$$s(x, y) = e^{-j2\pi(u_0x + v_0y)} \quad (8)$$

$$g(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)} \quad (9)$$

The Spatial (2-D) Gabor Filter

Here is the formula of a complex Gabor function in space domain

$$g(x, y) = s(x, y)w_r(x, y) \quad (10)$$

where $s(x, y)$ is a complex sinusoid, known as the carrier, and $w_r(x, y)$ is a 2-D Gaussian-shaped function, known as the envelope.

The complex sinusoid is defined as follows,

$$s(x, y) = \exp(j(2\pi(u_0x + v_0y) + P)) \quad (11)$$

where u_0, v_0 and P define the spatial frequency and the phase of the sinusoid respectively.

This sinusoid can be represented as two separate real functions, conventionally allocated in the real and imaginary part of a complex function. The real part and imaginary part of this sinusoid are -

$$Re(s(x, y)) = \cos(2\pi(u_0x + v_0y) + P) \quad (12)$$

$$Im(s(x, y)) = \sin(2\pi(u_0x + v_0y) + P) \quad (13)$$

The parameters u_0 and v_0 define the spatial frequency of the sinusoid in Cartesian coordinates.

3. BLOCK TRUNCATION CODING

Block truncation coding (BTC) is a simple and fast lossy compression technique for digitized gray scale images [9]. The block truncation coding algorithm is a simple, block-based and spatial domain. This compression technique is developed by Delp and Mitchell. As the amount of image data increase day by day, large storage and bandwidth are needed to store and transmit the images, which is quite costly [12]. The key idea of BTC is to perform moment preserving (MP) quantization for blocks of pixels so that the quality of the image will remain acceptable and at the same time the demand for the storage space will decrease. The truncated block of the BTC is the one bit output of the quantizer for every pixel in the block. Block Truncation Coding divides the image into small non-overlapping blocks of equal size and process these blocks independently [10]. It is a reversible and linear transform used to map each sub image into transform coefficients, which are later quantized and coded. Consider an input image of size $M * N$, it is first divided in sub-images of size $n * n$, which are then transformed to generate $MN = n^2$ subimage transform array. The goal of transformation process is to decorrelate the pixel of each sub-image. The quantizer then eliminates the coefficients that carry least amount of information. The decoder implements the inverse sequence of steps of the encoder and decompress the image.

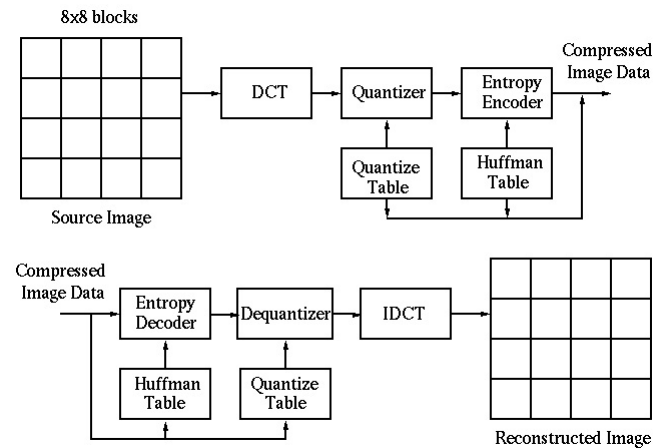


Fig. 1. Block Truncation Coding

3.1 Transform Selection

Block Truncation Coding system based on variety of transforms that have been constructed. The choice of a particular transform in an application depends on amount of reconstruction error that can be tolerated and computational resources available. Compression is achieved during the quantization of transformed coefficients. A simple transformation that is useful in transform coding is Walsh Hadamard Transform (WHT). Other useful transformations are Discrete Cosine Transform (DCT) and Discrete Fourier Transform (DFT).

3.2 Sub-image Size Selection

Second factor affecting transform coding error and computational complexity is sub-image size. Images are subdivided so that redundancy between adjacent sub-image is reduced to some level. As the sub-image size increases, level of compression and computational complexity also increases. The regular sub-image size of an image is 8×8 and 16×16 .

3.3 Bit Allocation

The reconstruction error is a function of the number and relative importance of transform coefficients that are discarded, as well as precision that is used to represent the retained coefficient. The retained coefficients are selected on the basis of maximum variance called zonal coding and on basis of maximum magnitude is called threshold coding. The overall Process of truncating, quantizing and coding the coefficients of transformed sub-image is known as bit allocation.

Block Truncation Coding example– The image was compressed from 8 bits per pixel to 2 bits per pixel (bpp). This is due to the fact that BTC requires 16 bits for the bit map, 8 bits for the sample mean and 8 bits for the sample standard deviation [2]. Thus, the entire 4×4 block requires 32 bits, and hence the data rate is 2 bpp. From this example it is easy to understand how a smaller data rate can be achieved by selecting a bigger block size, or by allocating less bits for the sample mean or the sample standard deviation [5,7].

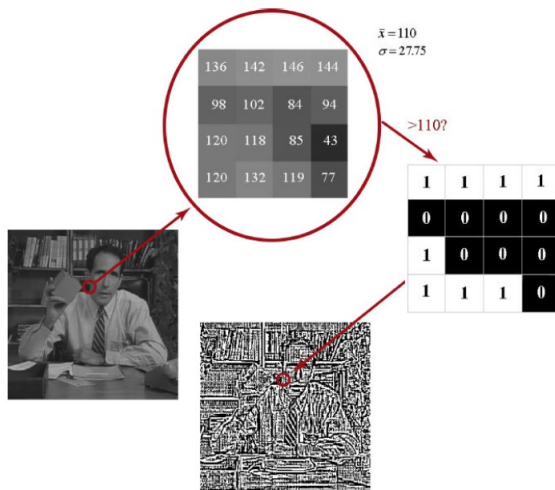


Fig. 2. Illustration of BTC compression process

4. PROBLEM METHODOLOGY

In this paper, our focus is on image compression and indexing technique, so in proposed method we implement block truncation coding technique for better image compression along with color and texture feature extraction. Block truncation coding divides the whole image into non-overlapping blocks and process them independently. This technique increase the functionality of the system. Color feature extraction is done by Color Moments and Color auto-correlogram and the texture extraction is done by Gabor filter which also help to extract better retrieval results. Proposed

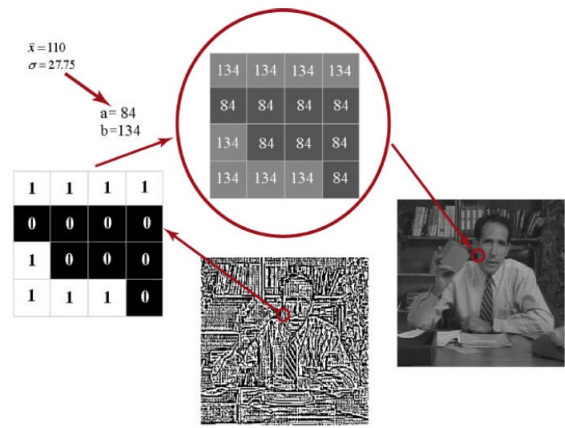


Fig. 3. Illustration of BTC decompression process

Method:-

- Here, we use Corel dataset having several category of images and extract the features from the images.
- Color feature is taken out by Color Moment and color auto-correlogram and texture feature is taken out by Gabor filter.
- Image Compression is done by block truncation coding. Here an image is divided into multiple non-overlapping blocks. Each block is process independently. The quantizer then eliminates the coefficients that carry least amount of information. The decoder implements the inverse sequence of steps of the encoder and decompress the image. With this, the quality of the image will remain acceptable and at the same time the demand for the storage space will decrease.
- Query image is fed into system and process is repeated same for other images and feature database is generated.
- Similarity matching is done by Manhattan Distance measurement and similar images are retrieved from the database.

5. EXPERIMENTAL RESULTS

5.1 Image Database

For evaluation of the proposed method, it has been implemented using Matlab and tested on a general purpose WANG database containing 1000 Corel images in JPEG format. The image set comprises 100 images in each of 10 categories [12]. In our experiment, we have selected 100 images randomly, containing 10 images in each category of animals, buses, flowers, natural scenery and waterfall etc. Within this database, it is known whether any two images are of the same category. In particular, a retrieved image is considered a match if and only if it is in the same category as the query.

5.2 Similarity Measure

Manhattan similarity measure is taken to match the image database features. As it is considered more accurate for natural images or scenery. The Manhattan distance function computes the distance that would be traveled to get from one data point to the other if a grid-like path is followed. The Manhattan distance between two items is the sum of the differences of their corresponding components.

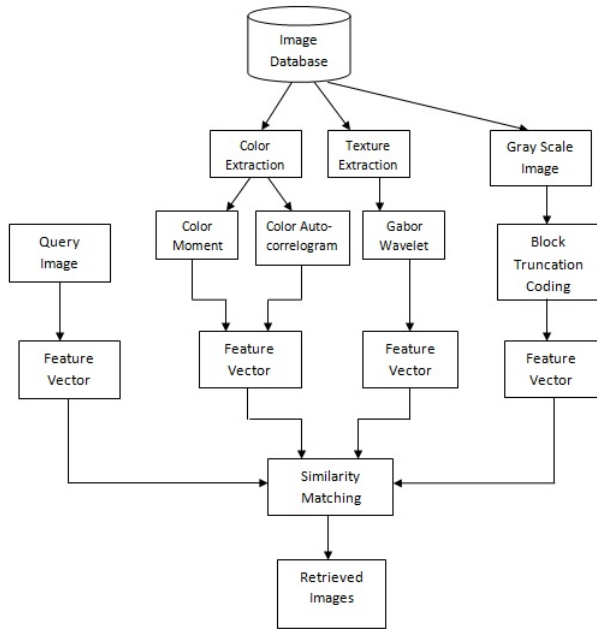


Fig. 4. Proposed Method

The formula for this distance between a point $X=(X_1, X_2, \dots, X_n)$ and a point $Y=(Y_1, Y_2, \dots, Y_n)$ is:

$$d = \sum_{i=1}^n |X_i - Y_i| \quad (14)$$

Where n is the number of variables, and X_i and Y_i are the values of the i th variable, at points X and Y respectively.

5.3 Precision and Recall

PRECISION is the ratio of the number of relevant images retrieved to the total number of irrelevant and relevant images retrieved. It is usually expressed as a percentage.

$$Precision = \frac{[relevantimages] \cap (retrievedimages)}{(retrievedimages)} \quad (15)$$

RECALL is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. It is usually expressed as a percentage.

$$Recall = \frac{[relevantimages] \cap (retrievedimages)}{(relevantimages)} \quad (16)$$

5.4 Results and Discussion

The proposed method is implemented in MATLAB and experiments conducted on the Corel 1000 dataset. The 10 classes are used for relevance estimation, given a query image and it is assumed that the user is searching for images from the same class, and therefore the remaining 99 images from the same class are considered relevant and the images from all other classes are considered irrelevant. After attempting the experiments on various images we calculate the Precision and Recall which represent in the form of graphs.- Precision is the percentage of documents in the retrieved set



Fig. 5. Retrieved Images of Class Dinosaur



Fig. 6. Retrieved Images of Class Flower

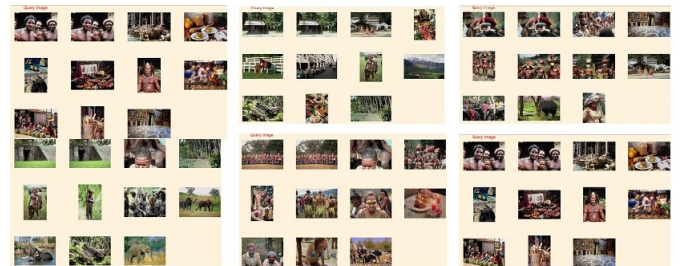


Fig. 7. Retrieved Images of Class Africa

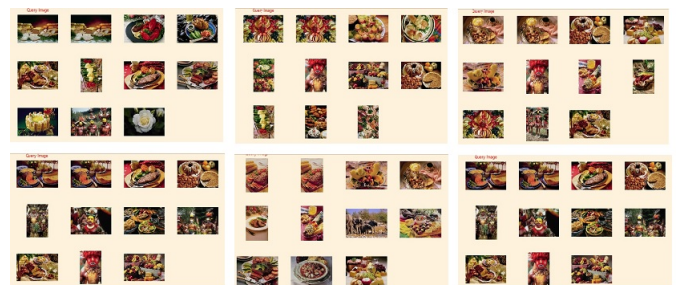


Fig. 8. Retrieved Images of Class Food

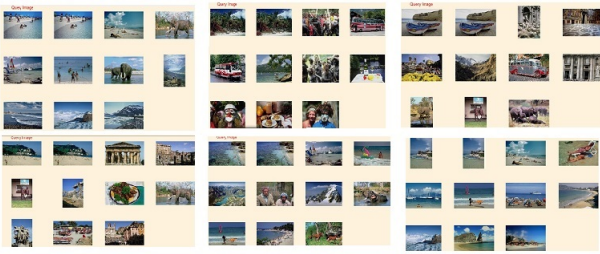


Fig. 9. Retrieved Images of Class Beach

which are actually relevant. It is a measure of exactness or quality. Precision means that an algorithm returned most of the relevant results. More is the value of precision, better is the prediction made for the code which explains the reduction of semantic gap. Recall is the percentage of all relevant documents which show up in the retrieved set. Recall is a measure of completeness or quantity. High recall means that an algorithm returned most of the relevant results from the results of recall of our system. The performance of the system is evaluated in the form of confusion matrix and after analyzing the retrieved results we concluded that proposed method is effective to reduce the semantic gap and indexed the retrieved images.

Classes	Africa	Beach	Monuments	Bus	Dinosaur	Elephant	Flower	Horse	Mountain	Food
Africa	75	5	4	0	0	5	0	3	1	7
Beach	2	72	6	0	0	8	0	0	12	0
Monuments	4	7	64	6	0	0	0	6	7	6
Bus	5	3	10	75	0	0	0	1	4	2
Dinosaur	0	0	0	0	100	0	0	0	0	0
Elephant	10	3	3	0	0	75	0	1	2	6
Flower	0	0	1	0	0	0	93	0	0	6
Horse	1	3	0	0	0	5	0	87	0	4
Mountain	3	6	3	6	0	5	0	0	73	4
Food	8	0	0	0	0	2	9	0	0	81

Fig. 10. Confusion Matrix Table

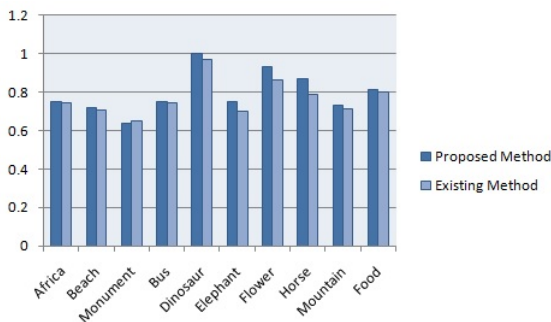


Fig. 11. Average Precision

6. CONCLUSION

A novel approach for CBIR system by combining the low-level features: color and texture feature extraction with block truncation coding algorithm is presented. The retrieved results show more precise value than the previous method and more correctly when

targets are clear. The proposed method is very effective for feature extraction and for better image indexing. The retrieval results provide a view about algorithm efficiency. At the same time, fusion of Color moments and Color auto-correlogram works better for the color feature extraction. For the color image indexing block truncation coding performs better with other methods, so that the semantic gap is less and image indexing is better.

7. REFERENCES

- [1] Meenakshi Garg Amit Singla. Cbir approach based on combined hsv, auto correlogram, color moments and gabor wavelet. volume 3, pages 9007–9012. International Journal Of Engineering And Computer Science, ISSN:2319-7242, 10 October 2014.
- [2] Martha Saenz Edward J. Delp and Paul Salama. *Video and Image Processing Laboratory*, chapter 5.
- [3] Dipankar Hazra. Retrieval of color image using color correlogram and wavelet filters. Proc. of International Conference on Advances in Computer Engineering, 2011.
- [4] Hong-Jiang Zhang Jufu Feng Hui Yu, Mingjing Li. Color texture moments for content-based image retrieval. pages 929–932, September 2002.
- [5] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. volume 18, pages 837–842, 1996.
- [6] W. Niblack. "the qbic project: Querying images by content using color, texture and shape. volume 1908, pages 173–187, 1993.
- [7] Dr. Bob Fisher Noah Keen. *Color Moment*. 2005.
- [8] V.E. Ogle and M. Stonebraker. Retrieval from a relational database of images. pages 40–48, 1995.
- [9] Olli Nevalainen Pasi Franti and Timo Kaukoranta. Compression of digital images by block truncation coding: a survey. Number 37(4), pages 308–332. The Computer Journal, 1994.
- [10] Richard E. Woods Rafael C. Gonzalez. *Digital Image Processing*. Pearson Education, third edition, 2014.
- [11] M. Choras R.S. Choras, T. Andrzyiak. Integrated color, texture and shape information for content-based image retrieval. 10, pages 333–343, 2007.
- [12] K. Hemachandran S. Mangijao Singh. Content-based image retrieval using color moment and gabor texture feature. volume 9 of *Issue 5*. IJCSI International Journal of Computer Science Issues ISSN (Online): 1694-0814, September 2012.
- [13] Vaidel A Shaila S G. Block encoding of color histogram for content based image retrieval applications. volume 6, pages 526–533, 2012.
- [14] J.L. Shih and L.H. Chen. Color image retrieval based on primitives of color moments. IEEE Proceedings online no. 20020614, 2002.
- [15] M. Stricker and M. Orengo. Similarity of color images. volume 2420, pages 381–392. SPIE Conference on Storage and Retrieval for Image and Video Databases, 1995.
- [16] M.Z. Swain and D.H. Ballard. Color indexing. volume 7 (1), pages 11–32. International Journal of Computer Vision, 1991.
- [17] B. Xue and L. Wanjun. Research of image retrieval based on color. IEEE International Forum on Computer Science Technology and Applications, 2009.
- [18] W.W.Y. Ng D.S. Yeung Z.C. Huang, P.P.K. Chan. Content-based image retrieval using color moment and gabor texture

feature. pages 719–724. Poceedings of the IEEE Ninth International Conference on Machine Learning and Cybernetics, Qingdao., 2010.