Journal of King Saud University – Computer and Information Sciences (2013) **25**, 207–218

King Saud University
Journal of King Saud University – Computer and Information Sciences
www.ksu.edu.sa
www.sciencedirect.com

ORIGINAL ARTICLE

Analysis of distance metrics in content-based image retrieval using statistical quantized histogram texture features in the DCT domain

Fazal Malik *, **Baharum Baharudin**

Computer and Information Sciences Department, Universiti Teknologi PETRONAS, Malaysia

Received 20 April 2012; revised 22 September 2012; accepted 18 November 2012
Available online 28 November 2012

KEYWORDS

Content-based image retrieval (CBIR);
Statistical texture features;
Distance metrics;
Quantized histogram;
DCT

Abstract The effective content-based image retrieval (CBIR) needs efficient extraction of low level features like color, texture and shapes for indexing and fast query image matching with indexed images for the retrieval of similar images. Features are extracted from images in pixel and compressed domains. However, now most of the existing images are in compressed formats like JPEG using DCT (discrete cosine transformation). In this paper we study the issues of efficient extraction of features and the effective matching of images in the compressed domain. In our method the quantized histogram statistical texture features are extracted from the DCT blocks of the image using the significant energy of the DC and the first three AC coefficients of the blocks. For the effective matching of the image with images, various distance metrics are used to measure similarities using texture features. The analysis of the effective CBIR is performed on the basis of various distance metrics in different number of quantization bins. The proposed method is tested by using Corel image database and the experimental results show that our method has robust image retrieval for various distance metrics with different histogram quantization in a compressed domain.

1. Introduction

A large number of images are available on the internet. Efficient and effective retrieval system is needed to retrieve these images by the contents of the images like color, texture and shape. This system is called content based image retrieval (CBIR). CBIR is an intensive and difficult area of research (Liu et al., 2007).

CBIR is performed in two steps: indexing and searching. In indexing step contents (features) of the image are extracted and are stored in the form of a feature vector in the feature database. In the searching step, user query image feature vector is constructed and compared with all feature vectors in the database for similarity to retrieve the most similar images to the query image from the database (Nezamabadi-pour and Saryazdi, 2005; Remco et al., 2002).

Availability of the huge number of images due to the rapid development and improvement of the internet, image capture devices and computer hardware cause the problem of storage and manipulation of images. To overcome the problem of

* Corresponding author. Tel.: +60 14 9050758.
E-mail address: fazal_i_malik@yahoo.com (F. Malik).
Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

1319-1578 © 2013 Production and hosting by Elsevier B.V. on behalf of King Saud University.
http://dx.doi.org/10.1016/j.jksuci.2012.11.004

space and manipulation time, at present almost all images are represented in compressed formats like JPEG and MPEG (Liu et al., 2007; Nezamabadi-pour and Saryazdi, 2005). The features of image can be extracted directly from the compressed domain. To extract the low level features from the compressed images, first the images are decoded from the compressed domain to the pixel domain. After that, image processing and analysis methods are applied to images in the pixel domain. This process is inefficient because it involves more computations and increases the processing time (Mandal et al., 1999). Therefore features can be extracted from images in the compressed format by using DCT (discrete cosine transformation) which is a part of the compression process. In compression, DCT transformation removes some information from images and important information is left behind which can play an important role in image retrieval. To get the effective retrieval of images using this information, histograms of quantized DCT blocks are optimized by selecting a proper quantization factor and a number of DCT blocks (Zhong and Defee, 2005).

The low level texture features are extracted from 8×8 DCT transformed blocks using DC and AC coefficients in nine different directions which represent nine feature vectors and grayscale level distribution in the image (Bae, 1997). The texture features in different directions are extracted using YUV color space such that the image is divided into four blocks and only the Y component in each block is transformed in DCT coefficients to get vertical, horizontal and diagonal texture features in all blocks for retrieval (Tsai, 2006).

The DC and some AC coefficients are used directionally to get energy histograms which are represented as feature vectors to retrieve similar images. On testing with a medium size database, these features give a high level of performance in terms of retrieval (Jose and Guan, 1999). DC vector is combined with nine AC coefficients distribution vectors to get the feature vector of the texture features of JPEG format images. AC coefficients describe the texture information (Shan and Liu, 2009).

The statistical texture features are extracted from images in compressed domain by computing mean and standard deviation moments using DCT coefficients. This method has robustness to translation and rotation (Feng and Jiang, 2003). In the JPEG compressed format, the texture features are extracted by computing the central moments of the second and third order using DCT coefficients. These features are used to form a feature vector to retrieve similar images (Vailaya, 1998). The quantized histograms are extracted from DCT coefficients in Mohamed et al., (2009) such that DC and first three AC coefficients are selected in a zigzag order from transformed 8×8 DCT blocks of JPEG format images and then the histograms of these coefficients are constructed with 32 bin quantizations. These histograms are used as feature vectors for retrieval. This method is tested by using the animal dataset of Corel database.

The histogram statistical Texture-Pattern is constructed using AC coefficients of each DCT block of image and used for image retrieval by measuring similarity using χ^2 distance metric (BAI et al., 2012). The histogram texture features are extracted directly from the DCT coefficients using the vector quantization technique (Yun and Runsheng, 2002).

To retrieve similar images for the query image from the database, the distance metric is used for matching. To measure the distance for similarity between query and database images the distance metrics like the Manhattan Distance (L1 metric), the Euclidean Distance (L2 metric) and the Vector Cosine Angle Distance (VCAD) are used and the performance of the Manhattan distance is high in terms of precision (Hafner et al., 1995).

The features of the image are represented in the feature vector form which represents the object. For similarity measurement various methods are used to compare the two feature vectors. The comparison methods are classified in two types: distance metric and similarity measure metric. The distance metric measures the difference between the two vectors of images and a small difference means, the two images are mostly similar. The similarity measure metric measures the similarity between the two vectors of images and a large similarity means the two images are closely related (Szabolcs, 2008).

In this paper, our main contribution is to show the performance of the image retrieval by extracting quantized histogram statistical texture features efficiently and matching the query image with database images effectively using various distance metrics of the DCT domain. To the best of our knowledge, such a retrieval of images based on the comparison of distance metrics using statistical texture features of grayscale image quantizing into different number of bins, have not been reported in the literature of the DCT domain. The proposed method is started with the non-overlapping 8×8 DCT block transformation of a grayscale image. The histograms of the DC and the first three AC coefficients are constructed. The statistical texture features like mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness are calculated by using the probability distribution of intensity levels in different quantization bins of histograms of all the blocks. The computed features are used to measure the similarity between the query image and database images by using various distance metrics. The performance is analyzed on the basis of results of various distance metrics in different quantization bins in the DCT domain.

The rest of the paper is organized such that Section 2 presents the DCT block transformation. Section 3 describes the construction and quantization of histograms in the DCT blocks. The proposed features and their extractions are elaborated in Section 4. Section 5 describes the similarity measurement of various distance metrics. Section 6 evaluates and analyzes the performance of distance metrics. Finally, Section 7 concludes this paper.

2. DCT block transformation

In our proposed method, first the input RGB color image is converted into grayscale image as shown in Fig. 1, to reduce the computations because it consists of only a single plane while the RGB image consists of three planes: red, green and blue (Schulz, 2007) and each component is a two dimensional matrix of pixel values from 0 to 256. The extraction of features in these three components will be time consuming, for example to extract 32 features in all three components the total features will be $32 \times 3 = 96$ instead of 32 features. Due to this issue the RGB color image is converted into grayscale which is a single component of 0 to 256 pixel values to reduce the computation cost. The grayscale image is divided into simple non-overlapping 8×8 blocks. Then all these blocks are transformed into DCT blocks in the frequency domain. Each block is in a 2-dimensional matrix. The 2-dimensional DCT of a block of the size $N \times N$ for $i,j = 1,2,\ldots\ldots\ldots\ldots,N$, can be calculated as:

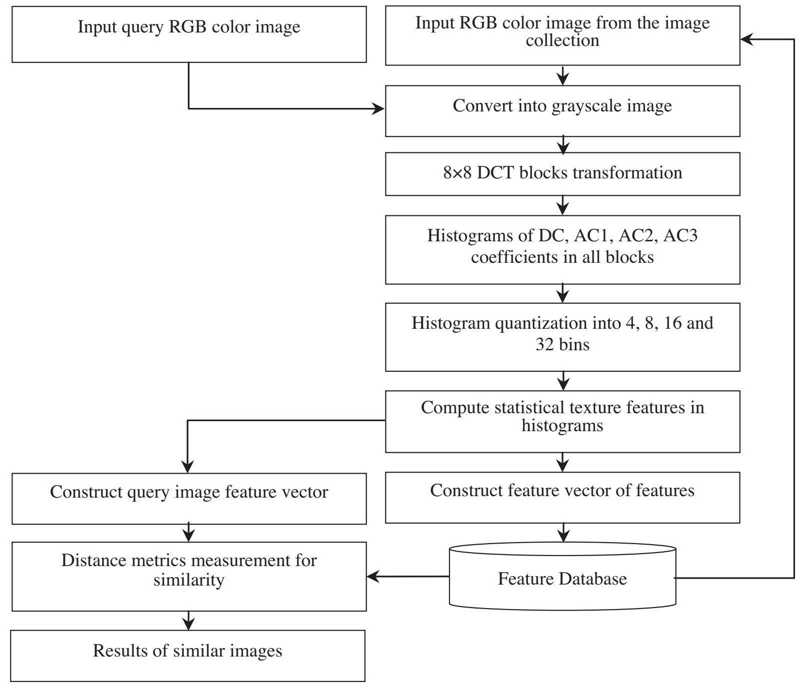

Figure 1 Step wise block diagram of the proposed method.

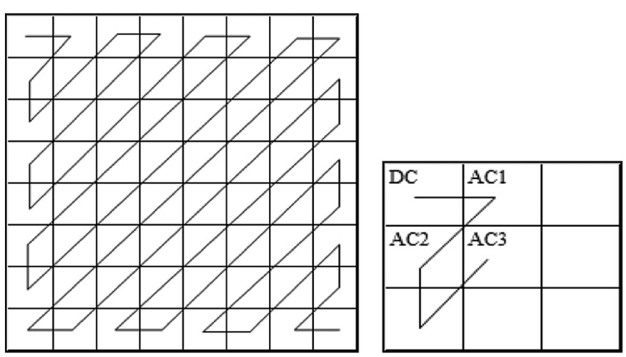

Figure 2 8×8 DCT block coefficients in zigzag order.

$$F(u,v) = \frac{1}{\sqrt{2N}} c(u)c(v) \sum_{x=1}^{N} \sum_{y=1}^{N} f(x,y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \times \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$
$$c(u) = \left\{ \begin{array}{ll} \frac{1}{\sqrt{2}} & if u = 0 \\ 1 & if u > 0 \end{array} \right\} \tag{1}$$

where $F(u, v)$ is the transformed block, $f(x, y)$ is the element of the block and N is the size of the block.

The first uppermost DCT coefficient in the DCT block is $F(0, 0)$ in (1); it is the average intensity value of a block and it is also called the DC coefficient or energy of the block. The other coefficients of the DCT blocks are called AC coefficients, which correspond to the different frequencies (co sinusoidal).

After the DCT transformation, the DC coefficients of all the blocks and the first three AC coefficients (AC1, AC2 and AC3) are selected in a zigzag order as shown in Fig. 2. All these DC and AC coefficients will be used to construct histograms.

3. Histogram quantization

The histogram of the DC is defined as the frequencies of the DC coefficients and the histogram of the AC is the frequencies of the AC coefficients in all the blocks. The DC histogram is then quantized into L bins such that:

$$H_{DC} = \{h(b_1)_{DC}, h(b_2)_{DC} \ldots h(b_L)_{DC}\} \tag{2}$$

where $h(bi)_{DC}$ is the frequency of the DC coefficients in bin bi and H_{DC} is the histogram of the L bins.

The histograms for the AC coefficients are also quantized into L bins such that:

$$H_{AC1} = \{h(b_1)_{AC1}, h(b_2)_{AC1}, \ldots, h(b_L)_{AC1}\} \tag{3}$$
$$H_{AC2} = \{h(b_1)_{AC2}, h(b_2)_{AC2}, \ldots, h(b_L)_{AC2}\} \tag{4}$$
$$H_{AC3} = \{h(b_1)_{AC3}, h(b_2)_{AC3}, \ldots, h(b_L)_{AC3}\} \tag{5}$$

where $h(bi)_{AC1}$, $h(bi)_{AC2}$ and $h(bi)_{AC3}$ are the frequencies and H_{AC1}, H_{AC2} and H_{AC3} are the histograms of AC1, AC2 and AC3 coefficients using the quantization of the L bins.

4. Feature extraction

The first issue in CBIR is to extract the features of the image efficiently and then represent them in a particular form to be used effectively in the matching of images. The statistical texture features are considered useful for the classification and retrieval of similar images. These texture features provide the information about the properties of the intensity level distribution in the image like uniformity, smoothness, flatness, contrast and brightness. The statistical texture features are extracted in the proposed method. The proposed texture features like mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness are calculated by using the probability distribution of intensity levels in the histogram bins of the histograms of DC, AC1, AC2 and AC3 coefficients.

Let $P(b)$ be the probability distribution of bin b in each of the histograms of the four coefficients with the L bins then it is can be calculated as:

$$P(b) = \frac{H(b)}{M} \tag{6}$$

where M is the total number of blocks in the image I.

The mean is the average of intensity values of all bins of the four quantized histograms. It describes the brightness of the image (Szabolcs, 2008; Selvarajah and Kodituwakku, 2011) and can be calculated as:

$$mean = \sum_{b=1}^{L} bP(b) \tag{7}$$

The standard deviation measures the distribution of intensity values about the mean in all blocks of histograms. The calculated value of standard deviation shows low or high contrast of histograms in images with low or high values (Selvarajah and Kodituwakku., 2011; Thawari and Janwe, 2011) and is calculated as:

$$std = \sqrt{\sum_{b=1}^{L} (b - mean)^2 P(b)} \tag{8}$$

The skewness[1] measures the unequal distribution of intensity values of all blocks of histograms about the mean value. The negative value of the skewness shows that most of the distribution of intensity values will be on the right side of the mean and the tail of the intensity values will be longer and skewed on the left side of the mean. The positive value shows that on the right side of the mean value the intensity values distribution will be high compared to the left side and on the left side the values will be skewed with a longer tail. The skewness with zero value is an indication of equal distribution of intensity values on both sides of the mean (Selvarajah and Kodituwakku, 2011; Suresh et al., 2008; Kekre and Sonawan, 2012). The skewness can be calculated as:

$$SKEW = \frac{1}{(std)^3} \sum_{b=1}^{L} (b - mean)^3 P(b) \tag{9}$$

The fourth texture feature is Kurtosis[2] which calculates the peak of the distribution of intensity values about the mean value. Kurtosis with high value shows the sharp peak of distribution with a longer and fat tail and the low value shows the rounded peak of distribution with a shorter and thinner tail (Selvarajah and Kodituwakku, 2011; Suresh et al., 2008; Kekre and Sonawan, 2012) and kurtosis can be calculated as:

$$kurtosis = \frac{1}{(std)^4} \sum_{b=1}^{L} (b - mean)^4 P(b) \tag{10}$$

The energy is measured as a texture feature to calculate the uniformity of the intensity level distribution in all bins of the histograms. The energy with high value shows the distribution of intensity values is for a small number of bins of histograms (Szabolcs, 2008; Selvarajah and Kodituwakku, 2011). Energy can be calculated as:

$$ENERGY = \sum_{b=1}^{L} [P(b)]^2 \tag{11}$$

The entropy measures the randomness of the distribution of intensity levels in bins. If the value of entropy is high then the distribution is among greater intensity levels in the image. This measurement is the inverse of energy. The simple image has low entropy while the complex image has high entropy (Szabolcs, 2008; Selvarajah and Kodituwakku, 2011) and Entropy can be calculated as:

$$ENTROPY = -\sum_{b=1}^{L} P(b) \log_2[P(b)] \tag{12}$$

The smoothness texture is used to measure the surface property of the image by using the standard deviation value of all bins of the histogram (Thawari and Janwe, 2011). It can be calculated as:

$$SM = 1 - \frac{1}{1 + (std)^2} \tag{13}$$

After the calculation of these texture features, these values are combined to get a feature vector fv such as:

$$fv = \{mean, std, SKEW, kurtosis, ENERGY, ENTROPY, SM\} \tag{14}$$

The feature vectors are calculated in all histograms such as fv_{HDC} is calculated in DC histograms H_{DC}, fv_{HAC1} in H_{AC1}, fv_{HAC2} in H_{AC2} *and* fv_{HAC3} in H_{AC3}. The four features of vectors are combined to get a single feature vector (FV) of features as:

$$FV = [fv_{HDC}, fv_{HAC1}, fv_{HAC2}, fv_{HAC3}] \tag{15}$$

The feature vectors (FVs) of all the images are constructed and stored to create a feature database. The feature vector of the user query is also constructed in the same way and compared with the feature vectors of the database for similarity and retrieval of relevant images. The block diagram of the method is shown in Fig. 1.

5. Similarity measurements

Once the feature database of the images is created with feature vectors using (6–15) in the first phase of the method as shown in Fig. 1, then the user can give an image as a query to retrieve similar images from the database. The feature vector of the query image is computed by using (6–15) in a second phase of the same method.

Similarity measurement is the second issue in CBIR in which the query image is compared with other database images. To measure the similarity between the query image and the database images, the difference is calculated between the query feature vector and the database feature vectors by using distance metrics. The small difference between two feature vectors indicates the large similarity and the small distance. The vectors of the images with small distances are most similar to the query images. The distance metrics which are included in this work are the sum of absolute difference (SAD), the sum of squared of absolute differences (SSAD) Selvarajah and Kodituwakku, 2011, Euclidean distance, city block distance, Canberra

[1] http://en.wikipedia.org/wiki/Skewness, last visit on June 1, 2012.
[2] http://en.wikipedia.org/wiki/Kurtosis, last visit on June 1, 2012.

distance, maximum value metric and Minkowski distance (Szabolcs, 2008).

Let the query feature vector be represented by $\boldsymbol{Q}$ and the database feature vector by $\boldsymbol{D}$ for seven distance metrics to calculate the difference between the two vectors for similarity which are:

5.1. Sum of absolute difference(SAD)

The sum of absolute difference (SAD)[3] is a very straightforward distance metric and extensively used for computing the distance between the images in CBIR to get the similarity. In this metric the sum of the differences of the absolute values of the two feature vectors are calculated (Selvarajah and Kodituwakku, 2011). The similarity is decided on the computed value of distance. This distance metric can be calculated as:

$$\Delta d = \sum_{i=1}^{n} (|Q_i| - |D_i|) \tag{16}$$

where n is the number of features, $i = 1, 2\ldots, n$. Both images are the same for $\Delta d = 0$ and the small value of Δd shows the relevant image to the query image.

The distance metric SAD is a simple method to search for similar images in the database to the query image automatically, but it can be sensitive and untrustworthy toward the consequences of background issues of images such as variations in size, color, illumination and direction of light[4].

5.2. Sum of squared absolute Difference(SSAD)

In this metric the sum of the squared differences of absolute values of the two feature vectors are calculated. This distance metric can be calculated (Selvarajah and Kodiyuwakku, 2011) as:

$$\Delta d = \sum_{i=1}^{n} (|Q_i| - |D_i|)^2 \tag{17}$$

It has some computational complexity due to square of differences as compared to SAD. However squaring always gives a positive value but it highlights a big difference. The distance metric SSAD can be used in both pixels and transformed domains but in the transformed domain the calculated value depends upon the quality of compression[5].

5.3. Euclidean distance

This distance metric is most commonly used for similarity measurement in image retrieval because of its efficiency and effectiveness. It measures the distance between two vectors of images by calculating the square root of the sum of the squared absolute differences and it can be calculated (Szabolcs, 2008) as:

$$\Delta d = \sqrt{\sum_{i=1}^{n} (|Q_i - D_i|)^2} \tag{18}$$

5.4. City block distance

This distance metric is also called the Manhattan distance. The city block distance metric has robustness to outliers. This distance metric is computed by the sum of absolute differences between two feature vectors of images and can be calculated[6] (Szabolcs, 2008) as:

$$\Delta d = \sum_{i=1}^{n} |Q_i - D_i| \tag{19}$$

5.5. Canberra distance

The city block distance metric gives a large value for the two similar images which create dissimilarity between similar images. Hence each feature pair difference is normalized by dividing it by the sum of a pair of features. This metric is used for numerical measurement of the distance between the query and database feature vectors and can be calculated (Szabolcs, 2008) as:

$$\Delta d = \sum_{i=1}^{n} \frac{|Q_i - D_i|}{|Q_i| + |D_i|} \tag{20}$$

The value of this method is arranged in ascending order such that the top most shows high similarity. It has similarity with city block distance metric. It has good effect on the data which are spread about the origin (Schulz, 2007).

5.6. Maximum value distance

This distance metric is also called Chebyshev distance. This metric is used to get the largest value of the absolute differences of paired features of feature vectors and can be calculated (Szabolcs, 2008) as:

$$\Delta d = \max\{|Q_1 - D_1|, |Q_2 - D_2|, \ldots, |Q_n - D_n|\} \tag{21}$$

The distance value is the maximum of the difference of the features of the pair of images, which shows the maximum dissimilarity of the two images.

5.7. Minkowski distance

The generalized form of the distance can be defined as:

$$\Delta d = \left[\sum_{i=1}^{n} (|Q_i - D_i|)^p\right]^{1/p} \tag{22}$$

where p is a positive integer.

This generalized form gives other distance metrics for positive values of p, for example $p = 1$ gives the city block and $p = 2$ gives the Euclidean distance. In this work on comparison of distance metrics, we also take $p = 3$ as the Minkowski distance.

[3] http://en.wikipedia.org/wiki/Sum_of_absolute_differences, last visit on September 18, 2012.

[4] http://en.wikipedia.org/wiki/Sum_of_absolute_differences, last visit on September 18, 2012.

[5] http://siddhantahuja.wordpress.com/tag/sum-of-squared-differences/, last visit on September 18, 2012.

[6] http://people.revoledu.com/kardi/tutorial/Similarity/Quantitative-Variables.html, last visit on September 18, 2012.

6. Experimental results

The proposed method is tested by using the Corel database of images, which is freely available for researchers (Wang et al., 2000). The database consists of 1000 images having 10 categories each of which has 100 images. The categories are people, beaches, buildings, buses, dinosaurs, elephants, roses, horses, mountains, and food. All these categories are used for the experiments. All the images are in the RGB color space. They are in the JPEG format with a size of 256×384 and 384×256 pixels.

6.1. Phases of the proposed method

The method is performed in two phases.

6.1.1. Phase-1

In the first phase, all the images are acquired, one after another, from the collection of images for the feature extraction. The extracted features are stored in a database in the form of feature vectors using (15) to create a feature database as shown in Fig. 1.

6.1.2. Phase-2

In the second phase, the user is asked to input the query image to retrieve similar images from the feature database by using the same method. The feature vector is constructed using (15) and compared with the feature vectors of the database by computing the similarities using the distance metrics from (16) to (22). The most similar images are displayed to the user according to the query image as shown in Fig. 1.

6.2. Evaluation measurements

The effectiveness of the image retrieval is based on the performance of the feature extraction and similarity measurement. In this section we describe the performance metrics which have been adopted not only to evaluate the effectiveness of image retrieval but also to make sure of the stability of the results. In order to evaluate the retrieval performance of CBIR, three measurements are used: precision, recall (Thawari and Janwe, 2011) and *F*-Score.

6.2.1. Precision

The precision in image retrieval can be defined as: precision is the measurement of the retrieved relevant images to the query of the total retrieved images[7].

$$Precision = A/B \tag{23}$$

where A is "the relevant retrieved images" and B is "the total retrieved images.

6.2.2. Recall

The recall in image retrieval can be defined as: Recall is the measurement of the retrieved relevant images to the total database images[8].

[7] http://en.wikipedia.org/wiki/Precision_and_recall, last visit on September 19, 2012.

[8] http://en.wikipedia.org/wiki/Precision_and_recall, last visit on September 19, 2012.

Table 1 Average precision of all image categories in histogram bins using Euclidean distance.

Categories	4 Bins	8 Bins	16 Bins	32 Bins	Average
Dinosaurs	100	100	100	100	**100**
Roses	90	91	95	96	**93**
People	85	86	88	88	**87**
Horses	80	82	93	93	**87**
Buses	74	76	84	86	**80**
Elephants	72	76	79	79	**77**
Beaches	75	79	82	82	**80**
Buildings	70	74	75	77	**74**
Mountains	60	61	61	62	**61**
Foods	55	57	57	60	**57**
Average	**76**	**78**	**81**	**82**	80

$$Recall = A/C \tag{24}$$

where A is "the relevant retrieved images" and C is "the total number of relevant images in the database".

For example, a CBIR method for a query image retrieves totally 10 images with 8 relevant images out of totally 30 relevant images in database. Then the precision is $8/10 = 80\%$ and recall is $8/30 = 27\%$. Thus this shows that only recall cannot measure the effectiveness of the CBIR, precision must also be computed.

6.2.3. F-Score[9]

The precision and recall measure the accuracy of image retrieval with relevancy to the query and database images and always two values are computed to show the effectiveness of image retrieval. However these two measurements cannot be considered as complete accuracy for the effective image retrieval. Hence they can be combined to give a single value that describes the accuracy of image retrieval and this combination is called *F*-Score or *F*-measure to measure accuracy. Both precision and recall measurements are combined to compute the score and it is also called as a weighted average or harmonic mean of the precision and recall. *F*-Score can be defined as:

$$F = 2 \times \frac{precision \times recall}{precision + recall} \tag{25}$$

The *F*-score value is a single value that indicates the overall effectiveness of the image retrieval.

In the experiment the two phases of the method are performed for all quantizations of the histograms into 4, 8, 16 and 32 bins separately. For each quantization of histograms, each distance metric is used separately in the experiment to get the results in terms of precision and recall to calculate the *F*-Score. Consequently, each distance metric will be tested in all quantization bins and the results are analyzed for all distance metrics against all quantization bins. The results, in precision, recall and *F*-score of all histogram bins using all image categories and only the Euclidean distance metric, are shown in Tables 1–3.

Table 1 shows the average precision in percentage of 10 image categories in different histogram quantization bins using the Euclidean distance for the matching of query image with database images in search of similar images. It can be seen that

[9] http://aimotion.blogspot.com/2011/05/evaluating-recommender-systems.html, last visit on September 19, 2012.

Table 2 Average recall of all image categories in histogram bins using Euclidean distance.

Categories	4 Bins	8 Bins	16 Bins	32 Bins	Average
Dinosaurs	100	100	100	100	**100**
Roses	89	90	91	93	**91**
People	86	86	94	95	**90**
Horses	82	85	89	92	**87**
Buses	71	75	76	80	**75**
Elephants	75	77	78	80	**77**
Beaches	73	74	75	76	**74**
Buildings	74	76	77	79	**76**
Mountains	64	67	69	73	**68**
Foods	63	66	67	70	**67**
Average	**77**	**79**	**81**	**84**	**81**

Table 3 Average F-Score of all image categories in histogram bins using Euclidean distance.

Categories	4 Bins	8 Bins	16 Bins	32 Bins	Average
Dinosaurs	100	100	100	100	**100**
Roses	89	90	93	94	**92**
People	85	86	91	91	**88**
Horses	81	83	91	93	**87**
Buses	72	75	80	83	**78**
Elephants	73	76	78	79	**77**
Beaches	74	76	78	79	**77**
Buildings	72	75	76	78	**75**
Mountains	62	64	65	67	**64**
Foods	59	61	62	65	**62**
Average	**77**	**79**	**81**	**83**	**80**

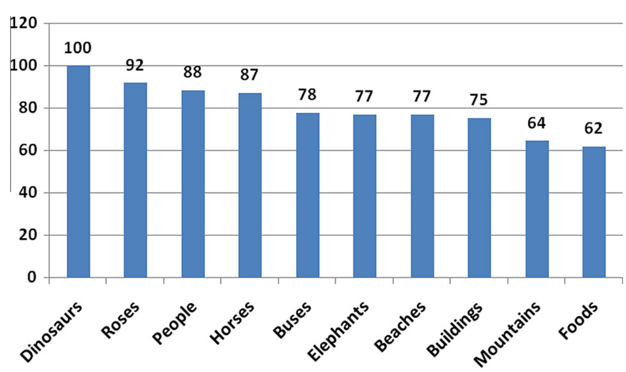

Figure 3 Average *F*-Score of all image categories in histogram bins using Euclidean distance in percentage.

the dinosaurs, roses, people and horses give better results as compared to the other categories. All the images of each category are used as query images. The average precision is improving incrementally from 4 bins quantization to 32 bins. The average precision of 32 bins is 82% whereas the overall average precision is 80%, which shows good retrieval.

Table 2 shows the average recall in percentage of 10 image categories in different histogram quantization bins using Euclidean distance for the matching of the query image with database images in search of similar images. The results show

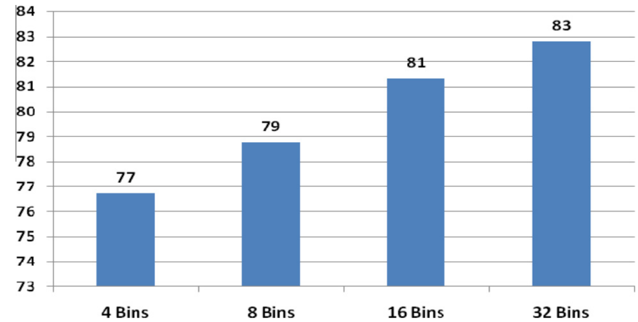

Figure 4 Average *F*-Score of histogram bins using Euclidean distance against all image categories in percentage.

Table 4 Average precision of distance metrics using all image categories in different histogram bins.

Distance Metrics	4 Bins	8 Bins	16 Bins	32 Bins	Average
Euclidean distance	76	78	81	**82**	**80**
City block distance	68	75	79	78	**75**
Sum of squared of absolute differences (SSAD)	65	72	75	79	**73**
Canberra Distance	64	69	75	78	**71**
Maximum value distance	60	66	75	79	**70**
Sum of absolute difference (SAD)	50	59	63	70	**61**
Minkowski distance (with $p = 3$)	55	58	60	70	**61**
Average	**63**	**68**	**73**	**77**	**70**

Table 5 Average recall of distance metrics using all image categories in different histogram bins.

Distance Metrics	4 Bins	8 Bins	16 Bins	32 Bins	Average
Euclidean distance	77	79	81	**84**	**81**
City Block Distance	75	76	80	82	**78**
Canberra distance	75	78	79	80	**78**
Sum of squared of absolute differences (SSAD)	76	77	74	79	**77**
Maximum value distance	79	74	74	78	**76**
Minkowski distance (with $p = 3$)	69	75	76	79	**75**
Sum of absolute difference (SAD)	68	69	65	79	**70**
Average	**74**	**75**	**76**	**80**	**76**

Table 6 Average F-Score of distance metrics using all image categories in different histogram bins.

Distance Metrics	4 Bins	8 Bins	16 Bins	32 Bins	Average
Euclidean distance	77	79	81	**83**	**80**
City block distance	71	75	79	80	**77**
Canberra distance	69	73	77	79	**75**
Sum of Squared of Absolute Differences (SSAD)	70	74	74	79	**74**
Maximum value distance	68	70	74	78	**73**
Minkowski distance (with $p = 3$)	61	65	67	**74**	**67**
Sum of absolute difference (SAD)	58	64	64	74	**65**
Average	**68**	**71**	**74**	**78**	**73**

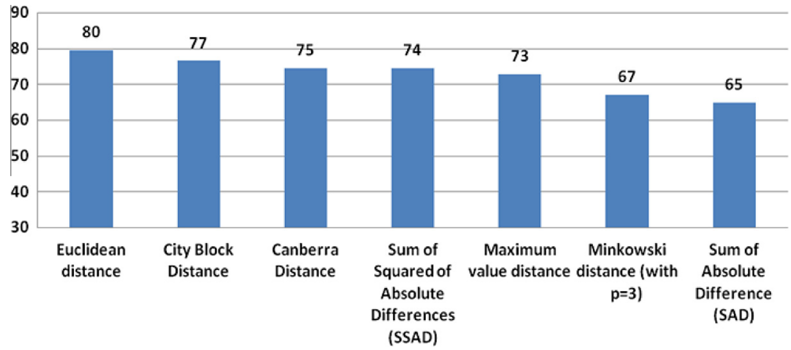

Figure 5 Average F-Score of the proposed distance metrics using different histogram bins.

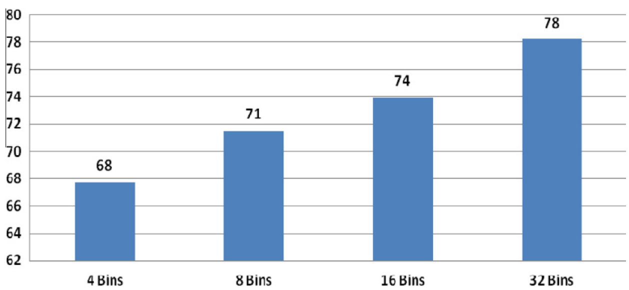

Figure 6 Average *F*-Score of the proposed distance metrics histogram bin wise.

that the performance in recall also is better for dinosaurs, roses, people and horses as compared to the other categories. The average recall is also improving incrementally from 4 bins quantization to 32 bins. The average recall of 32 bins is 84% and 81% is the overall average recall.

Table 3, Figs. 3 and 4 show the average *F*-Score which describes the overall performance of the retrieval of similar images category wise in various histogram quantization bins using the Euclidean distance. The retrieval performance of the proposed method using the Euclidean distance for categories such as dinosaurs, roses, people and horses is better as shown in Fig. 3. It can be seen in Fig. 4 that the *F*-Score retrieval is increasing from the 4 bin quantizations toward 32 bins. The average *F*-Score of 32 bins quantization is 83% while the overall average retrieval is 80% which shows that 32 bins provide more energy in DCT blocks for retrieval.

Table 4 shows the average precision of proposed distance metrics which are used in the matching of images to retrieve similar images using different histogram quantization bins for the extraction of statistical texture features in DCT blocks. It can be seen that the best retrieval average precision is 82% of the Euclidean distance using 32 bins quantization. The city block, sum of squared of absolute differences (SSAD) and Canberra distance also give good performance in terms of precision. The results show that using the top four distance metrics as shown in Table 4, for matching of images for the retrieval of similar images gives the best results especially in 32 bin quantizations in the DCT domain.

Table 5 shows the average recall of proposed distance metrics which are used in the matching of images to retrieve similar images using different histogram quantization bins for the extraction of statistical texture features in DCT blocks. The best retrieval recall is 84% of Euclidean distance using 32 bins quantization. The city block, sum of squared of absolute differences (SSAD) and Canberra distance also give good performance in terms of recall. The overall average recall is 76%.

Table 6 shows the average *F*-Score of the proposed distance metrics using quantized histogram texture features of DCT blocks. The *F*-Score results show that when using histogram statistical texture features of different quantization bins the Euclidean distance gives good retrieval performance, especially in 32 bins in the DCT domain. The city block, Canberra distance and sum of squared of absolute differences (SSAD) also give good performance in terms of retrieval as shown in Figs. 5 and 6, shows the *F*-Score results histogram bin wise for the proposed distance metrics. It is clear that the 32 bins quantization provides good energy in the DC and in the first three AC coefficients of the DCT blocks in the frequency domain for the best retrieval of the JPEG images in terms of an average 73% *F*-Score.

Table 7 shows the average computation time taken by the proposed distance metrics for the matching of a query image with database images to retrieve similar images. It can be seen in Fig. 7 that the Euclidean distance, city block distance and

Table 7 Average computation time (minutes) of proposed distance metrics for matching of query image with database images.

Distance Metrics	4 Bins	8 Bins	16 Bins	32 Bins	Average
Euclidean distance	0.0183	0.0185	0.0186	0.0188	**0.0186**
City block distance	0.0187	0.0188	0.0190	0.0190	**0.0189**
Sum of absolute difference (SAD)	0.0187	0.0188	0.0190	0.0193	**0.0190**
Maximum value distance	0.0187	0.0188	0.0195	0.0192	**0.0191**
Sum of squared of absolute differences (SSAD)	0.0189	0.0189	0.0192	0.0194	**0.0191**
Minkowski distance (with $p = 3$)	0.0190	0.0192	0.0192	0.0193	**0.0192**
Canberra distance	0.0192	0.0193	0.0194	0.0193	**0.0193**
Average	**0.0188**	**0.0189**	**0.0191**	**0.0192**	**0.0190**

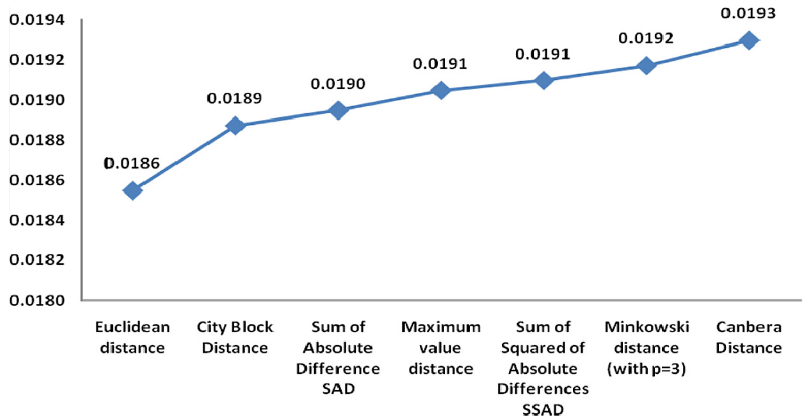

Figure 7 Average computation time (minutes) of proposed distance metrics for matching of the query image with database images.

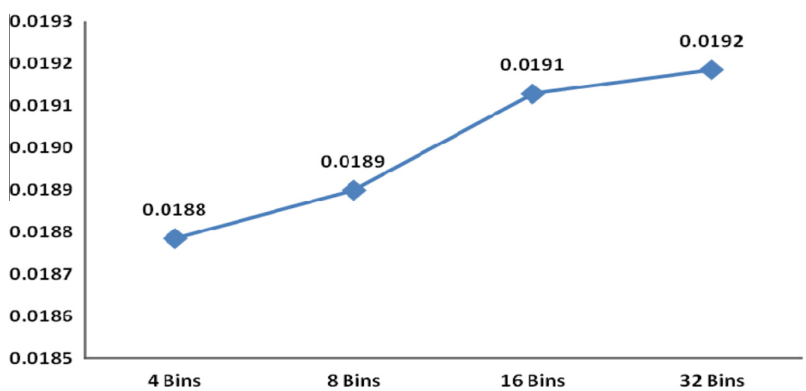

Figure 8 Average computation time (minutes) of the proposed distance metrics for matching of the query image with database images in different quantization bins.

Table 8 Comparison of proposed method with other methods based on precision, category wise.

Categories	Alnihoud J. Liu et blank;al., (2007)	Mohamed A. et al. Vailaya (1998)	Murala S. et al. Mohamed and Khellif (2009)	P.B. Thawari et al. Thawari and Janwe (2011)	P.S. Hiremath, et al. Hiremath and Pujari (2007)	Proposed method
Dinosaurs	100	99	100	90	95	**100**
Roses	97	N/A	80	N/A	61	**96**
Horses	85	40	91	N/A	74	**93**
People	87	N/A	76	50	48	**88**
Buses	84	N/A	52	50	61	**86**
Beaches	68	N/A	50	40	34	**82**
Elephants	50	70	62	N/A	48	**79**
Buildings	70	N/A	47	35	36	**77**
Mountains	32	N/A	28	N/A	42	**62**
Foods	63	N/A	63	N/A	50	**60**
Average	**74**	**70**	**65**	**53**	**55**	**82**

sum of absolute difference (SAD) are taking less computational cost for similarity measurement of the query image with database images. The results show that the Euclidean distance is not only effective in retrieval but also efficient in computations. Fig. 8 shows that the computation using 32 bins quantization takes slightly more time as compared to other quantizations but for retrieval performance is better than other bins.

6.3. Performance analysis of the proposed method

The proposed method is compared with the methods of Liu et al. (2007), Vailaya (1998), Mohamed and Khellfi. (2009), Thawari and Janwe (2011), Hiremath and Pujari (2007) based on precision as shown in Table 7. In the method of Liu et al. (2007) color and shape features are extracted based on SOM

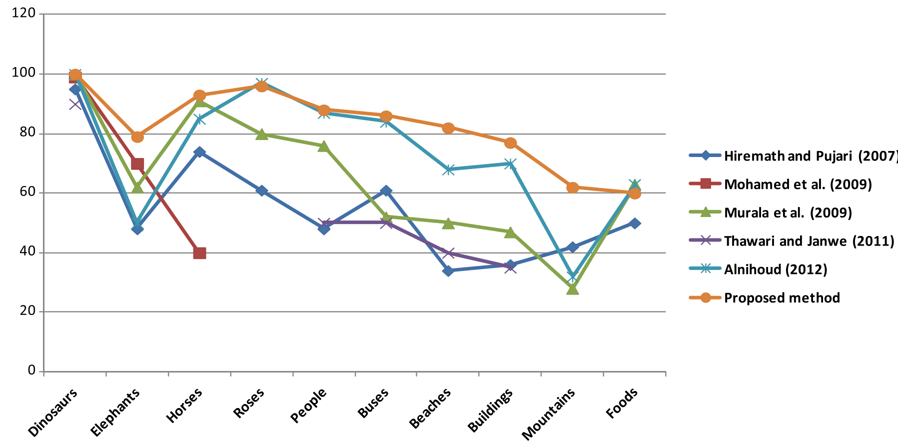

Figure 9 Comparison of the proposed method with other methods based on precision, category wise.

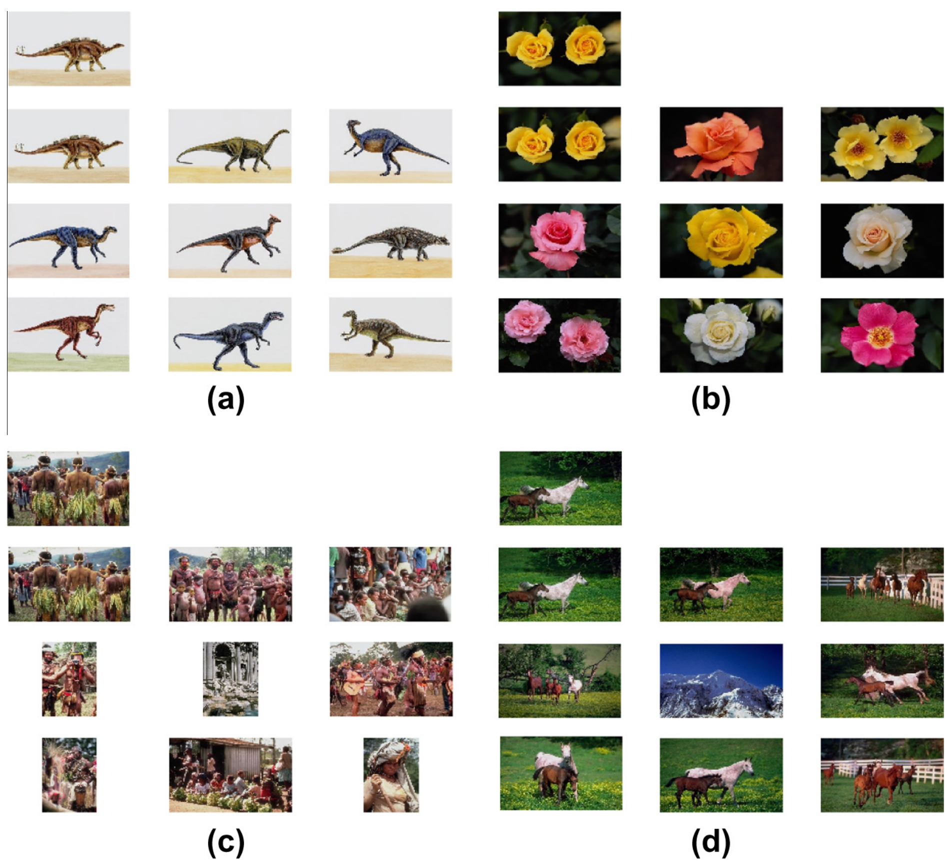

Figure 10 Query results of (a) dinosaurs (b) roses (c) people and (d) horses, using histograms of 32 bins and the Euclidean distance for similarity measurement.

(self-organizing map). Fuzzy Color Histogram (FCH) and subtractive fuzzy clustering algorithms are used to get color features and the object model algorithm is used to get an edge of objects and then shape features like area, centroid, major axis length, minor axis length, eccentricity and orientation which are computed to get performance in terms of average precision of 74%. In method of Vailaya (1998), an image is divided into non-overlapping 8×8 blocks and then these blocks are transformed into the DCT domain. The DC and the first three AC coefficients of each block are picked up in zigzag order. These coefficients are used to construct quantized histograms of 32 bins. These histograms are used to construct a feature vector for retrieval. They have tested their method with animal dataset only and get average precision of 70%. In method of Mohamed and Khellfi. (2009), color and texture features are combined to retrieve similar images. For color, mean and standard deviation are computed in histograms of 32 bins in each channel of the RGB color image, to get totally 192 features. For texture features, mean and standard deviation are computed in sub bands of Gabor Wavelet Transform image with three scales and four orientations to get a feature vector of 48 features. The performance of this method is measured in terms of average precision of 65%. In (Thawari and Janwe, 2011) HSV color space is used with three color channels H, S and V. Histogram of each channel is quantized into 96 blocks and each block has a dimension of 32×32 pixels. The statistical texture moments like mean standard deviation, skewness, kurtosis, energy; entropy and smoothness are calculated in each bin of the histogram. Totally $96 \times 7 \times 3 = 2016$ features are computed. Thus this process of feature extraction involves a large number of computations which increase computation cost. The method has used 500 images of the Corel database for testing. The average precision of the method in Thawari and Janwe (2011) is 53%. In the method of Hiremath and Pujari (2007), color, texture and shape features are fused together and are extracted in a non-overlapping portioned image. Texture features are extracted by using Gabor filter, statistical color moments are used to calculate color features. The shape of objects is extracted by using gradient vector flow fields and the shape features are depicted by using invariant moments. The method is tested by using Corel image database and overall average precision is 55%.

In our proposed method we start with the conversion of RGB color images into grayscale images to reduce the computational cost. This grayscale image is transformed into non-overlapping 8×8 DCT blocks. The DC and first three AC coefficients with significant energy, of each block are picked up in zigzag order to construct the histograms. The histograms are quantized into different number of bins to calculate statistical texture features of histogram bins. For similarity measurement various distances metrics are used. The performance is analyzed on the basis of distance metrics using different quantized histogram bins. Our proposed method is tested by using the same Corel image database as used by other methods. The results of our method are effective not only in retrieval but also in efficiency. The overall average precision of our method is 82% which is higher than other methods using 32 bins histograms. Our proposed method has good performance in terms of precision using quantized texture features and the Euclidean distance for similarity measurement in the DCT domain for compressed images as shown in Table 8 and Fig. 9.

Fig. 10(a–d) shows the results of user queries. Each figure consists of a query image and the similar retrieved images from the database. The top single image is the query image and below nine are the relevant images. The results show that proposed method has good retrieval accuracy.

7. Conclusion

In this paper a CBIR method is proposed which is based on the performance analysis of various distance metrics using the quantized histogram statistical texture features in the DCT domain. Only the DC and the first three AC coefficients having more significant energy are selected in each DCT block to get the quantized histogram statistical texture features. The similarity measurement is performed by using seven distance metrics. The experimental results are analyzed on the basis of seven distance metrics separately using different quantized histogram bins such that the Euclidean distance has better efficiency in computation and effective retrieval in 32 bins quantization. We conclude that the, Euclidean distance, city block distance, and sum of absolute difference (SAD) metrics give good performance in terms of precision using quantized histogram texture features in the DCT domain for compressed images. Our method is compared with other methods and our method has good performance in terms of precision and *F*-Score.

References

Hee-Jung Bae, Sung-Hwan Jung, 1997. Image retrieval using texture based on dct. international conference on information, communications and signal processing ICICS 97, Singapore, 9–12.

BAI, C., et al., 2012. Color Textured Image Retrieval By Combining Texture And Color Features," European Signal Processing Conference (EUSIPCO-2012), Bucharest: Romania.

Feng, Guocan, Jiang, Jianmin., 2003. JPEG compressed image retrieval via statistical features. Pattern Recognit. 36 (4), 977–985.

Hafner, J., Sawhney, H., Equitz, W., Flickner, M., Niblack, W., 1995. Efficient color histogram indexing for quadratic form distance functions. IEEE Trans. Pattern Anal. Mach. Int. 17 (7), 729–736.

Hiremath, P.S., Pujari, J., Content 2007. Based Image Retrieval Using Color, Texture and Shape Features, presented at the Proceedings of the 15th International Conference on Advanced Computing and, Communications.

Kekre, H., Sonawan, K., 2012. Statistical parameters based feature extraction using bins with polynomial transform of histogram. Int. J. Adv. Eng. Technol. 4 (1), 510–524.

Jose, A., Lay, Ling Guan, 1999. Image retrieval based on energy histograms of the low frequency DCT coefficients. IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258) vol. 6 3009–3012.

Liu, Y., Zhang, D., Lu, G., Ma, W., 2007. A survey of content-based image retrieval with high level semantics. Pattern Recognit. 40 (1), 262–282.

Mandal, M.K., Idris, F., Panchanathan, S., 1999. A critical evaluation of image and video indexing techniques in the compressed domain. Image Vision Comput. 17 (7), 513–529.

Aamer Mohamed, F., Khellfi, Ying Weng, Jianmin Jiang, Stan Ipson. 2009.An efficient image retrieval through DCT histogram quantization. International Conference on CyberWorlds 237–240.

Nezamabadi-pour, Hossein, Saryazdi, Saeid, 2005. Object-based image indexing and retrieval in DCT domain using clustering techniques. World Acad. Sci. Eng. Technol., 3–6.

Remco, C., Veltkamp, Mirela Tanase, 2002. content-based image retrieval systems: a survey.

Jan Schulz, Algorithms – Similarity, September 2007,. http://www.code10.info/index.php?option=com_content&view=article&id=49:article_canberra-distance&catid=38:cat_coding_algorithms_data-similarity&Itemid=57.

Selvarajah, S., Kodituwakku. S.R., 2011. Analysis and Comparison of Texture Features for Content Based Image Retrieval. International Journal of Latest Trends in Computing (E-ISSN: 2045–5364), vol. 2 (1), 108–113.

Shan, Zhao, Liu, Yan-Xia, 2009. Research on JPEG Image Retrieval. 5th International Conference on Wireless Communications, Networking and Mobile. Computing 16 (1), 1–4.

Suresh, P., Sundaram, R.M.D., Arumugam, A., 2008. Feature Extraction in Compressed Domain for Content Based Image Retrieval. IEEE International Conference on Advanced Computer Theory and Engineering (ICACTE), pp. 190–194..

Sergyan Szabolcs, 2008. Color histogram features based image classification in content-based image retrieval systems. 6th

International Symposium on Applied Machine Intelligence and Informatics 221–224.
Thawari, P.B., Janwe, N.J., 2011. Cbir based on color and texture. Int. J. Inf. Technol. Knowl. Manage. 4 (1), 129–132.
Tienwei Tsai, Yo-Ping Huang, Te-Wei Chiang, 2006. Dominant feature extraction in block-DCT domain. IEEE International Conference on Systems, Man and Cybernetics 3623–3628.
Aditya Vailaya, Anil Jain, Hong Jiang Zhang, 1998. On image classification: city images vs. Landscapes, In: Pattern Recognition 31(12)1921–1935.
Wang, J.Z., Li, J., Wiederhold, G., 2000. SIMPLIcity: Semantics-Sensitive Integrated Matching Libraries''. Advances in Visual for Picture Information Systems: 4th International Conference, VISUAL 2000, Lyon, France, November 2-4, 2000: Proceedings, http://wang.ist.psu.edu/docs/related/.
Yun, F., Runsheng, W., 2002. An Image retrieval method using DCT features. J. Comput. Sci. Technol. 17 (6).
Zhong, Daidi., Defee, I., 2005. DCT histogram optimization for image database retrieval. Pattern Recognit. Lett. 26 (14), 2272–2281.