The Statistical Quantized Histogram Texture Features Analysis for Image Retrieval Based on Median and Laplacian Filters in the DCT Domain

Fazal Malik and Baharum Baharudin
Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Malaysia

Abstract: *An effective Content-Based Image Retrieval (CBIR) system is based on efficient feature extraction and accurate retrieval of similar images. Enhanced images by using proper filter methods can also, play an important role in image retrieval in a compressed frequency domain since currently most of the images are represented in the compressed format by using the Discrete Cosine Transformation (DCT) blocks transformation. In compression, some crucial information is lost and perceptual information is left, which has significant energy requirement for retrieval in a compressed domain. In this paper, the statistical texture features are extracted from the enhanced images in the DCT domain using only the DC and first three AC coefficients of the DCT blocks of image having more significant information. We study the effect of filters in image retrieval using texture features. We perform an experimental comparison of the results in terms of accuracy on the basis of median, median with edge extraction and Laplacian filters using quantized histogram texture features in a DCT domain. Experiments using the Corel database, give the improved results on the basis of filters; more specifically, the Laplacian filter with sharpened images gives good performance in retrieval of JPEG format images as compared to the median filter in the DCT frequency domain.*

Keywords: *CBIR, median filter, laplacian filter, statistical texture features, quantized histograms, DCT.*

Received February 25, 2012; accepted May 22, 2012; published online August 5, 2012

1. Introduction

A huge number of images are provided by the Internet and other image capturing devices due to which an efficient and effective retrieval system is needed to retrieve these images using the contents of the images like colour, texture and shape. This system is called Content Based Image Retrieval (CBIR). The CBIR is intensive and is a difficult area of research [8].

CBIR is performed in two steps: feature extraction and searching. In the first step, the features of an image are extracted and stored in the form of a feature vector to create a feature database. In the searching step, a user query image feature vector is constructed and compared with all the feature vectors in the database for similarity in order to retrieve the most similar images to the query image from the database [11, 18].

Availability of the huge number of images due to the rapid development and improvement in Internet, image capture devices and computer hardware causes problems of storage and manipulation of images. To overcome the problems of space and manipulation time, at present almost all of the images are represented in a compressed format like the JPEG and the MPEG [8, 11]. The features of the image can be extracted directly in a compressed domain. To extract the low level features from the compressed images, first the images are decoded from the compressed domain to a pixel domain. After that, image processing and analysis methods are applied to the images in the pixel domain. This process is inefficient because it involves more computations and increases the processing time [9]. Therefore, features can be extracted directly from images in the compressed format by using the Discrete Cosine Transformation (DCT), which is a part of the compression process. In compression using DCT, the significant information is removed from the images and only perceptually important information is left which can play an important role in image retrieval. To get the best retrieval of images using this information, histograms of quantized DCT blocks are optimized by selecting a proper quantization factor and number of DCT blocks [21].

In this paper, our main contribution is to perform the experimental comparative analysis of the statistical quantized histogram texture features for the effective image retrieval in terms of precision and recall in the DCT domain based on the median and Laplacian filters. In our approach, we start with an 8×8 DCT block transformation of the filtered histogram equalized grayscale image. The histograms of the DC and the first three AC coefficients are constructed by quantizing them into 32 bins. Then, the statistical texture features mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness are calculated by using the probability distribution of the intensity levels in the histogram bins of all the blocks. These features construct a feature vector to retrieve the similar images from the database. These vectors are used in a similarity measurement to compare the query image vector with the database vectors. In this paper, we demonstrate the comparison of the results of the quantized histogram texture features based on the median, median with edge extraction and Laplacian filters which give the best performance in terms of image retrieval in the DCT domain.

The rest of the paper is organized as follows: Section 2 explains related works. Section 3 describes, in detail the pre-processing of grayscale images. Section 4 presents the DCT block transformation. Section 5 describes the construction and quantization of histograms in the DCT blocks. The proposed features and their extraction are elaborated in section 6. Section 7 describes the similarity measurement of the image retrieval. Section 8 analyzes the experimental results of the texture histogram features based on filters in terms of precision and recall. Finally, section 9 concludes this paper.

2. Related Works

Many methods and algorithms have been reported in literature for CBIR based on texture histogram features in the DCT domain. The DC and AC coefficients of 8×8 DCT transformed blocks are represented in nine different directions which represent the nine feature vectors of the texture features and grayscale level distribution in the image. This method has high performance in retrieval [3]. In the YUV colour space, the texture features are extracted such that the image is divided into four blocks and only the Y component in each block is transformed into the DCT coefficients to get vertical, horizontal and diagonal features in all the blocks for retrieval [16]. The DC and some of the AC coefficients are used directionally to get energy histograms which are represented as feature vectors to retrieve similar images. In testing with a medium sized database, these features give high level performance in terms of retrieval [7]. The DC vector is combined with a nine AC coefficients distribution vector to get the feature vector of the texture features of the JPEG format images. AC coefficients describe the texture information [14].

The statistical texture features are extracted from the images in the compressed domain by computing the mean and standard deviation moments using the DCT coefficients. The results show good performance in terms of efficiency and effectiveness. This approach has robustness to translation and rotation [5].

The comparative analysis of the features is performed on the basis of the median filter to reduce the noise and the edge extraction method to keep the original edge information, in terms of precision. The feature vector is created by taking average number of pixels in each bin to retrieve the images [20].

The normalized histograms are quantized in 48 bins in each component of the RGB colour image. Thus, for each image a feature vector of a total of 48×3=144 features is created. For similarity measurements, the Euclidean distance is used to calculate the distance between the query image feature vector and the database image feature vectors. The images are ranked by the similarity distance values and the results show good performance in terms of effectiveness [4].

In the JPEG compressed format, the texture features are extracted by computing the central moments of the second and third order using the DCT coefficients. These features are used to form a feature vector to retrieve similar images [17]. The quantized histograms are extracted from the DCT coefficients in [10], such that the DC and the first three AC coefficients are selected in a zigzag order from the transformed 8×8 DCT blocks of the JPEG format images and then the histograms of these coefficients are constructed with a 32 bins quantization. These histograms are used as a feature vector for retrieval. This method is tested using the animal dataset.

3. Pre-processing

Our proposed method starts with the conversion of the input RGB image into a single component grayscale image to reduce the computation cost because the colour image consists of Red, Green and Red components [2]. In the next step the grayscale image is converted into a Histogram Equalized (HE) image, to make the image's intensity levels equal to get a high contrast image as shown in Figure 1. Then the median, edge extraction and Laplacian filters are applied to get a more enhanced image.

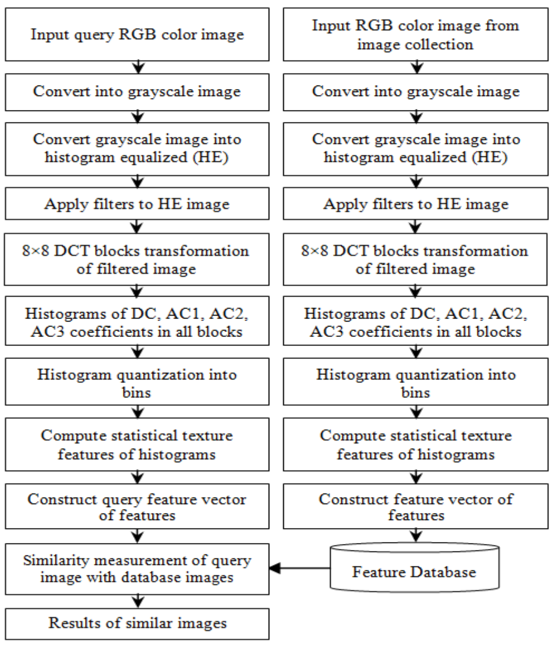


Figure 1. Block diagram of proposed algorithm.

3.1. Median Filter

Images consist of some noise. Before applying the processing techniques on the images to extract the low level features, the images need to be pre-processed to remove unwanted information and to get the enhanced images with the most relevant information. To remove the noise and get an enhanced image, the median filter is applied on the histogram equalized grayscale image [20]. The median filter is based on the neighbourhood operations. It consists of a window which is encompassed over the image to order pixels in the image area and then replace the central pixel with the determined values. The median filter replaces the value of a pixel by the median of the gray levels in the neighbourhood of that pixel [6]. Then, the texture features are extracted from the median filtered image for the retrieval of similar images.

3.2. Edge Extraction

However, the median filtering removes the noise from the images but some black specks are left around the border. These black points on the border are due to the default padding of zeros (0's). Some amount of information in an image, like edge information, is lost [6]. To restore the edge information of the median filtered image, canny edge detection is used to determine the edge information in an image before applying the median filter [20]. This is the most powerful edge detector and it detects two edge points, strong and weak with two threshold values T1 and T2 such that T1<T2. The edge values are strong for the pixel values greater than T2 and the edge values are weak for the pixel values that lie between T1 and T2. Finally, the canny method connects the weak edges to the strong edges by using the 8-connection and detects the edge information before applying the median filter. The edge information of the median filtered image is restored by the already extracted edge information [6]. Then the texture features are extracted from the median filtered image with the edge extraction method using the spatial information and are used for image retrieval. The results are analyzed on the basis of this filter technique.

3.3. Laplacian Filter

The Laplacian filter is a linear filter and it consists of a window or mask having some values which works with the values of image pixels in the neighbourhood. The values in the filter window are called the filter coefficients. The result of this filter is the sum of the products of the filter coefficients and the corresponding image pixel values. This filter gives an image with the strong edges.

Let $f(x,y)$ be an original image and $\nabla^2 f(x,y)$ be a Laplacian image such that:

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \quad (1)$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1,y) + f(x-1,y) - 2f(x,y) \quad (2)$$

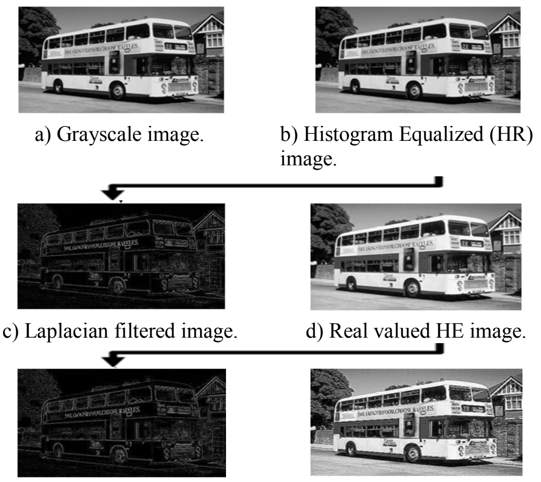
$$\frac{\partial^2 f}{\partial y^2} = f(x,y+1) + f(x,y-1) - 2f(x,y) \quad (3)$$

$$\nabla^2 f = [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)] - 4f(x,y) \quad (4)$$

To get the filtered image, at all points (x, y) in equation 4 can be convoluted with the 3×3 mask in Figure 2.

0	1	0
1	- 4	1
0	1	0

Figure 2. Laplacian filter with 3×3 mask.



a) Grayscale image. b) Histogram Equalized (HR) image.

c) Laplacian filtered image. d) Real valued HE image.

e) Laplacian filtered image. f) Laplacian sharpened image.

Figure 3. Sharpening process using laplacian filter.

Before using the Laplacian filter, the RGB image is converted into grayscale as shown in Figure 3-a, to get a single component of the image and then it is again converted into the histogram equalized image f to get the enhanced image as shown in Figure 3-b. The image f is filtered with the Laplacian filter to get a filtered image, g_1, with the edges of the objects in the image as shown in Figure 3-c. But all the pixel values in g_1 are positive and these values must be negative because of the negative value (-4) at the centre of the mask as shown in Figure 2. For this purpose, the histogram equalized image f is converted into the real valued image f_2 as shown in Figure 3-d. This image f_2, is again filtered with the Laplacian filter to get the image g_2, with the edge information as shown in Figure 3-e. But, during the filter process of the image g_2, some amount of information is lost. To restore this information and get an enhanced and sharpened image g, the Laplacian filtered image g_2 is subtracted from the real valued image as:

$$g = f_2 - g_2 \quad (5)$$

where g is the sharpened and enhanced image with detailed information and it is shown in Figure 3-f. This process is also, called sharpening of the image [6]. The features are extracted from the image g, for retrieval and analysis.

4. DCT Block Transformation

The enhanced filtered image is divided into simple non-overlapping 8×8 blocks. Then all these blocks are converted into the DCT transformed blocks in a frequency domain. Each block is in a 2-dimensional matrix. The 2-dimensional DCT of a block of the size $N \times N$ for $i, j=1,2,\ldots,N$ can be calculated as:

$$F(u,v) = \frac{1}{\sqrt{2N}} c(u)c(v) \sum_{x=1}^{N} \sum_{y=1}^{N} f(x,y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \times \cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad (6)$$

$$c(u) = \begin{cases} \frac{1}{\sqrt{2}} & if u = 0 \\ 1 & if u > 0 \end{cases}$$

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 equation 6, it is also called the DC coefficient and it represents the average intensity value of a block. The DC coefficient is also described as the energy of the block. The other coefficients of the DCT blocks are called the AC coefficients, which correspond to the different frequencies (co sinusoidal).

After 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 Figure 4. All these DC and AC coefficients will be used to construct the histograms.

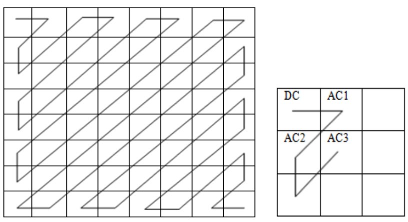


Figure 4. 8×8 DCT block coefficients in zigzag order.

5. Histogram Quantization

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

$$H_{DC}= \{h(b_1)_{DC}, h(b_2)_{DC} \ldots h(b_L)_{DC}\} \quad (7)$$

where $h(b_i)_{DC}$ is the frequency of the DC coefficients in bin b_i and H_{DC} is the histogram of the L bins. In this method, we take L as 32 bins. The histograms of the AC coefficients are also, quantized into L bins such that:

$$H_{AC1}= \{h(b_1)_{AC1}, h(b_2)_{AC1} \ldots h(b_L)_{AC1}\} \quad (8)$$

$$H_{AC2}= \{h(b_1)_{AC2}, h(b_2)_{AC2} \ldots h(b_L)_{AC2}\} \quad (9)$$

$$H_{AC3}= \{h(b_1)_{AC3}, h(b_2)_{AC3} \ldots h(b_L)_{AC3}\} \quad (10)$$

where $h(b_i)_{AC1}$, $h(b_i)_{AC2}$ and $h(b_i)_{AC3}$ are the frequencies of the AC1, AC2 and AC3 coefficients in bin b_i and H_{AC1}, H_{AC2} and H_{AC3} are the histogram of the L bins.

6. Feature Extraction

The statistical texture features are considered useful for classification and retrieval of similar images. These texture features provide information about the properties of the intensity level distribution in the image like uniformity, smoothness, flatness and contrast. The statistical texture features of mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness are calculated by using the probability distribution of the intensity levels in the histogram bins of the histograms H_{DC}, H_{AC1}, H_{AC2}, and H_{AC3}.

Let $P(b)$ is the probability distribution of bin b in each of the four histograms using equations 7 to 10 with L levels; it is defined as:

$$P(b) = \frac{H(b)}{M} \quad (11)$$

where M is the total number of blocks in the image I. The mean is the texture feature that represents something about the brightness of the image. The mean measures the average value of the intensity values. If the mean is high, then it means that the image is bright and if low, then the image is dark. The mean can be defined [1, 12, 13] as:

$$mean = \sum_{b=1}^{L} bP(b) \quad (12)$$

The standard deviation is the second order moment and it shows the contrast of the gray level intensities. The low value of the standard deviation indicates low contrast and the high value shows the high contrast of the image. This can be computed [1, 12, 15] as:

$$stddev = \sqrt{\sum_{b=1}^{L} (b - mean)^2 P(b)} \quad (13)$$

The third order moment is the skewness and it shows the skewness of the intensity values. It is the measurement of the inequality of the intensity level distribution about the mean. The value will be positive or negative of the skewness. The negative value indicates that the large number of intensity values is on the right side of the mean and the skewness of the tail of the intensity values is towards the left side of the distribution or the tail on the left side is longer than the right side. The positive value indicates that the large number of intensity values is on the left side of the mean and the skewness of the tail of the intensity values is towards the right side of the distribution or the tail on right side is longer than the left side. The zero value indicates that the distribution of the intensity values is relatively equal on both sides of the mean. The skewness[1] can be defined [1, 2] as:

$$SKEW = \frac{1}{(stddev)^3}\sum_{b=1}^{L}(b-mean)^3 P(b) \quad (14)$$

The fourth order moment is the kurtosis; it is used to measure the peak of the distribution of the intensity values around the mean. The high value of the kurtosis indicates that the peak of the distribution is sharp and the tail is longer and fat. The low value of the kurtosis indicates that the peak of the distribution is rounded and the tail is shorter and thinner. Kurtosis[2] can be defined [12] as:

$$Kurtosis = \frac{1}{(stddev)^4}\sum_{b=1}^{L}(b-mean)^4 P(b) \quad (15)$$

The energy feature measures the uniformity of the intensity level distribution. If the value is high, then the distribution is to a small number of intensity levels. Energy can be defined [1, 12, 13] as:

$$Energy = \sum_{b=1}^{L}[P(b)]^2 \quad (16)$$

The entropy measures the randomness of the distribution of the coefficients values over the intensity levels. If the value of entropy is high, then the distribution is among more intensity levels in the image. This measurement is the inverse of energy. A simple image has low entropy while a complex image has high entropy. Entropy can be defined [12, 13] as:

$$Entropy = -\sum_{b=1}^{L}P(b)\log_2[P(b)] \quad (17)$$

The smoothness texture is measured by using the standard deviation value. It can be defined [15] as:

$$SM = 1 - \frac{1}{1+(stddev)^2} \quad (18)$$

After the calculation of these texture features, the feature vector *fv* of these values is constructed as:

$$fv = \{mean, stddev, SKEW, kurtosis, Energy, Entropy, SM\} \quad (19)$$

The feature vectors are calculated for all the histograms using equation 19 such as fv_{HDC} is calculated for the DC histograms H_{DC}, fv_{HAC1} for H_{AC1}, fv_{HAC2} for H_{AC2} and fv_{HAC3} for H_{AC3}. The four feature vectors are combined to get a single Feature Vector (*FV*) of the features as:

$$FV = [fv_{HDC}, fv_{HAC1}, fv_{HAC2}, fv_{HAC3}] \quad (20)$$

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 the similarity and retrieval of relevant images. The block diagram of the algorithm is shown in Figure1.

7. Similarity Measurements

Once the feature database of the images is created with the feature vectors using equations 11 to 20 in the first phase of the algorithm as shown in Figure 1, then the user can give an image as a query to retrieve the similar images from the database. The feature vector of the query image is computed by using the same equations 11 to 20 as the second phase of the same algorithm as shown in Figure 1.

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 the distance metric. The small difference between two feature vectors indicates a large similarity and a small distance. The vectors of the images with a small distance are most similar to the query image. The distance metric, which we have included in this work, is the 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. Let the query feature vector be represented by *Q* and the database feature vector by *D* to calculate the difference between the two vectors for similarity using the Euclidean distance as:

$$\Delta d = \sqrt{\sum_{i=1}^{n}\left(\left|Q_i - D_i\right|\right)^2} \quad (21)$$

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

8. Experimental Results

The proposed algorithm is tested by using the Corel database of images provided by Wang *et al*. [19], which is freely available for the researchers. The database consists of 1000 images having 10 image categories and each of which has 100 images. The image categories consist of people, beaches, buildings, buses, dinosaurs, elephants, roses, horses, mountains and food. All these image categories are used in the experiments. All images are in the RGB colour space. They are in the JPEG format with a size of 256×384 and 384×256 pixels.

8.1. Phases of the Algorithm

The algorithm is performed in two phases:

- *Phase 1:* In the first phase, all images are acquired one after another from the collection of images for

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

the feature extraction. The extracted features are stored in a database in the form of feature vectors using equation 20 to create a feature database as shown in Figure 1.

- *Phase 2:* In the second phase, the user is asked to input the query image to retrieve relevant images from the feature database by using the same algorithm. The feature vector is constructed using equation 20 and compared with the feature vectors of the database by computing the similarity using the distance metric in equation 21. The relevant images are displayed to the user according to the query image as shown in Figure 1.

8.2. Results and Discussion

In the experiments the two phases of the algorithm are performed in sequence for the median, median with edge extraction and Laplacian filters using 10 image categories of 1000 images and the average precision and recall are calculated for all these filters. The results in the precision are shown in Table 1 for all three filtering methods using the 10 image categories.

Table 1. Average precision of median, median with edge extraction and Laplacian filters using all the image categories.

Categories	Median Filter	Median with Edge Extraction	Laplacian Filter	Average
People	100	100	100	100
Dinosaurs	100	100	100	100
Horses	80	89	92	87
Buses	79	89	82	83
Beaches	67	78	80	75
Roses	61	62	78	67
Buildings	56	56	57	56
Elephants	51	57	57	55
Mountains	39	34	42	39
Foods	31	30	51	37
Average	66	69	74	70

Table 1 shows the average precision of the 10 categories of images for all the three filter methods and the average precision of the three filter methods for categories using statistical quantized texture histogram features in the DCT domain. It can be seen that the people, dinosaurs, horses and buses give better results as compared to other categories as show in Figure 5. The overall average precision is 70%, which shows good retrieval.

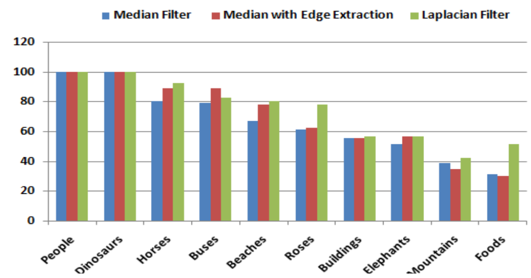


Figure 5. Average precision of image categories using three filter methods.

Figure 5 shows that the Laplacian filter has good precision for almost all the categories except mountains and food. It can be seen in Figure 6 that the performance of the retrieval of images in terms of precision is increasing from the median to the Laplacian filters with average precision of 66%, 69% and 74%, using texture quantized histogram features of the JPEG compressed format images in the frequency DCT domain.

The Laplacian filter with a sharpened and enhanced image using the texture features with 32 bins histogram quantization gives a better result with 74% average precision, especially for people and dinosaurs.

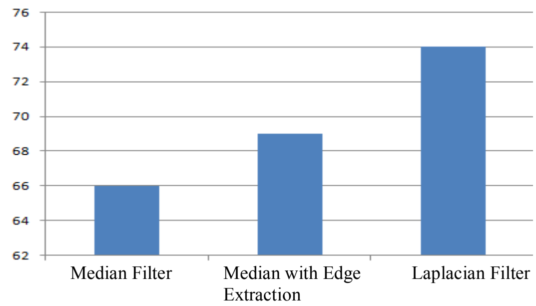


Figure 6. Average precision of three filter methods using all the image categories.

Table 2 shows the average recall of 10 image categories against the three filter methods using the 1000 images. It can be seen that the people, dinosaurs, roses and mountains give better results as compared to other categories as show in Figure 7. The overall average recall is 74%, which shows good retrieval.

Table 2. Average recall of median, median with edge extraction and Laplacian filters using all the image categories.

Categories	Median Filter	Median with Edge Extraction	Laplacian Filter	Average
People	100	100	100	100
Dinosaurs	100	100	95	98
Roses	79	80	88	82
Mountains	80	67	68	72
Elephants	61	69	82	71
Beaches	55	68	73	65
Horses	56	66	72	65
Foods	61	57	73	64
Buses	54	65	70	63
Buildings	57	57	68	61
Average	70	73	79	74

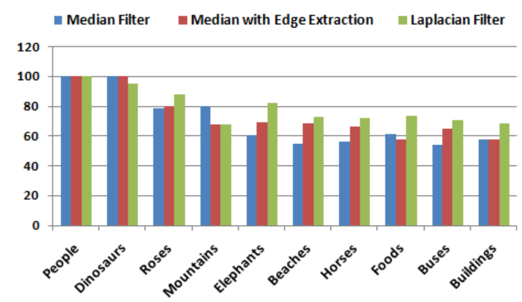


Figure 7. Average recall of the image categories using the three filter methods.

Figure 7 shows that the Laplacian filter has well recall for almost all the image categories. Figure 8 shows the comparison of the three filter methods: median, median with edge extraction and Laplacian filter in terms of recall using all the 1000 images as query images. The average recall of the proposed method based on the median, median with edge extraction and Laplacian filters are 70%, 73% and 79%. This shows that the Laplacian sharpened images have more energy for the retrieval of similar images in the DCT frequency domain using the texture features with 32 bins histogram quantization and give better result with a 79% average recall, especially for people and dinosaurs.

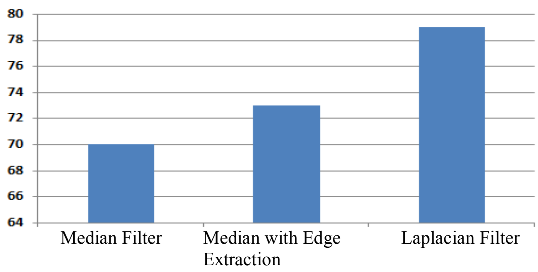


Figure 8. Average recall of the three filter methods using all the image categories.

Figure 9 shows the comparison of precision and recall for image categories using median, median with edge extraction and Laplacian filter methods. It can be seen that six categories including people, dinosaurs, roses, mountains, elephants and beaches give more than a 60% better image in the DCT domain using texture features.

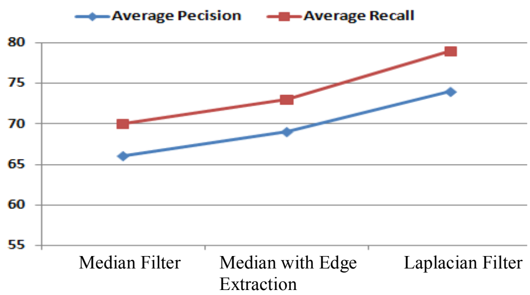


Figure 9. Average precision and recall of all the image categories using the three filter methods

Figure 10 gives the incremental performance of the image retrieval of our proposed algorithm from the median to Laplacian filters. The Laplacian filter with the edge information gives an enhanced image having more significant energy for the best retrieval of similar images.

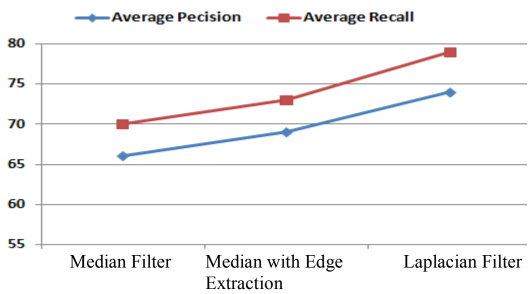


Figure 10. Average precision and recall of the three filter methods using all the image categories.

8.3. Performance Analysis of the Proposed Algorithm

Our approach is compared with the approach of Zhao *et al*. [20], in which they applied a median filter on the histogram equalized grayscale image and then quantized the filtered image into 32 bins in the pixel domain and computed the average of the pixel values for all the bins. These average values of pixels are used as features to construct feature vectors to be used in similarity measurements by taking the absolute difference of the query and the image database feature vectors for retrieval of similar images. They performed a comparison of the retrieval results on the basis of the median and median with edge extraction filter methods. In our approach, we apply the median and Laplacian filters on the histogram equalized grayscale image for further enhancement. This filtered grayscale image is transformed into non-overlapping 8×8 DCT blocks. The DC and the first three AC coefficients of each block are picked up in a zigzag order to construct the histograms. These histograms are quantized into 32 bins and seven statistical texture features are calculated for all the histograms. These texture features construct the feature vectors and are used in the similarity measurements by using the Euclidean distance metric for similar image retrieval. We have also used the same dataset as was used by the algorithm [20]. The results of our approach are better in terms of precision and recall as shown in Figures 11 and 12.

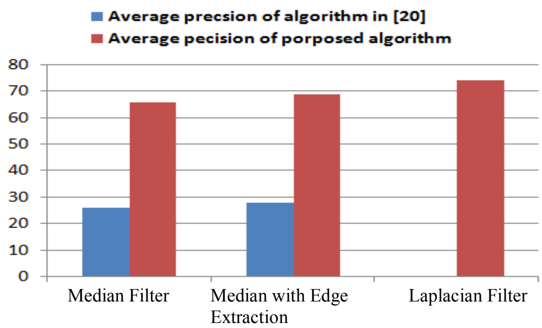


Figure 11. A comparison of average precision of our proposed algorithm and precision of algorithm in [20].

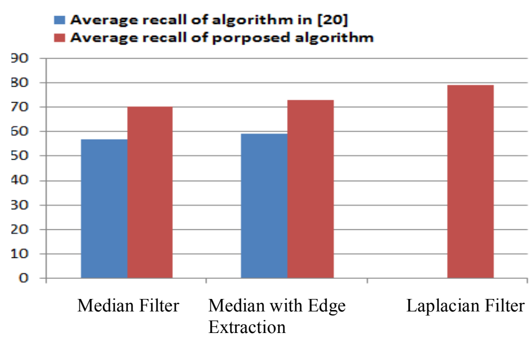
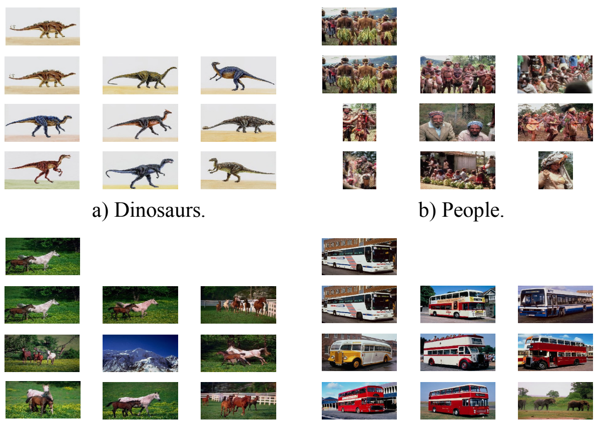


Figure 12. A comparison of average recall of our proposed algorithm and recall of algorithm in [20].

Our proposed algorithm has better results in terms of precision and recall for all the three filter methods in the DCT domain using texture features, especially for the Laplacian filter as shown in Figures 11 and 12. These filter methods provide enhanced images with more significant energy in the DCT blocks and the energy plays an important role in the retrieval of similar images. The energy can be extracted in the form of the statistical texture features which provide near to optimum performance in terms of retrieval, especially for the Laplacian filter as compared to other filter methods. Our proposed algorithm also provides improved performance for the median and median with edge extraction methods as compared to algorithm [20].

Figure 13-a to 13-d shows the results of the user queries. Each result of a figure consists of a query image and the similar retrieved images from the database. The top single image is the query image and the nine below are the relevant images. The results show that the proposed method has good retrieval accuracy.



a) Dinosaurs. b) People.

c) Horses. d) Buses.

Figure 13. Query results.

9. Conclusions

In this paper an algorithm is proposed for the effective image retrieval in which the experimental comparison of the statistical texture features in terms of accuracy of image retrieval based on the median and Laplacian filters is performed in the DCT domain. Only the DC and the first three AC coefficients with more significant energy are selected in each DCT block to get the quantized histogram statistical texture features. These features are extracted from median, median with edge extraction and Laplacian filtered images. The experimental comparison of the results of the three filter methods are analyzed for the 1000 images of 10 categories in terms of the accuracy of image retrieval. We conclude that the enhanced and sharpened Laplacian filtered images using the quantized histogram texture features give good performance in terms of precision and recall in the DCT domain for compressed images as compared to the retrieval of images in the pixel domain.

References

[1] Alnihoud J., "Content-Based Image Retrieval System Based on Self Organizing Map, Fuzzy Color Histogram and Subtractive Fuzzy Clustering," *International Arab Journal of Information Technology*, vol. 9, no. 5, pp. 452-458, 2012.

[2] Anjum M. and Javed M., "Multiresolution and Varying Expressions Analysis of Face Images for Recognition," *Information Technology Journal*, vol. 6, no. 1, pp. 57-65, 2007.

[3] Bae H. and Jung S., "Image Retrieval using Texture Based on DCT," *in Proceedings of International Conference on Information, Communications and Signal Processing*, South Korea, vol. 2, pp. 1065-1068, 1997.

[4] Chakravarti R. and Meng X., "A Study of Color Histogram Based Image Retrieval," *in Proceedings of Information Technology: New Generations*, Las Vegas, pp. 1323-1328, 2009.

[5] Feng G. and Jiang J., "JPEG Compressed Image Retrieval Via Statistical Features," *Pattern Recognition*, vol. 36, no. 4, pp. 977-985, 2003.

[6] Gonzalez R., Woods R., and Eddins S., *Digital Image Processing using MATLAB*, Pearson Prentice Hall, 2004.

[7] Lay J. and Guan L., "Image Retrieval Based on Energy Histograms of the Low Frequency DCT Coefficients," *in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, vol. 6, pp. 3009-3012, 1999.

[8] Liu Y., Zhang D., Lu G., and Ma W., "A Survey of Content-Based Image Retrieval with High-Level Semantics," *Pattern Recognition*, vol. 40, no. 1, pp. 262-282, 2007.

[9] Mandal M., Idris F., and Panchanathan S., "A Critical Evaluation of Image and Video Indexing Techniques in the Compressed Domain," *Image and Vision Computing*, vol. 17, no. 7, pp. 513-529, 1999.

[10] Mohamed A., Khellfi F., Weng Y., Jiang J., and Ipson S., "An Efficient Image Retrieval through DCT Histogram Quantization," *in Proceedings of International Conference on CyberWorlds*, Bradford, pp. 237-240, 2009.

[11] Nezamabadi-Pour H. and Saryazdi S., "Object Based Image Indexing and Retrieval in DCT Domain using Clustering Techniques," *World Academy of Science Engineering and Technology*, vol. 3, no. 1307-6884, pp. 207-210, 2005.

[12] Selvarajah S. and Kodituwakku S., "Analysis and Comparison of Texture Features for Content Based Image Retrieval," *International Journal of Latest Trends in Computing*, vol. 2, no. 1, pp. 108-113, 2011.

[13] Sergy´an S., "Color Histogram Features Based Image Classification in Content-Based Image Retrieval Systems," *in Proceedings of the 6th International Symposium on Applied Machine Intelligence and Informatics*, Herlany, pp. 221-224, 2008.

[14] Shan Z. and Liu Y., "Research on JPEG Image Retrieval," *in Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing*, Beijing, pp. 1-4, 2009.

[15] Thawari P. and Janwe N., "CBIR Based on Color and Texture," *International Journal of Information Technology and Knowledge Management*, vol. 4, no. 1, pp. 129-132, 2011.

[16] Tsai T., Huang Y., and Chiang T., "Dominant Feature Extraction in Block-DCT Domain," *in Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Taipei, vol. 5, pp. 3623-3628, 2006.

[17] Vailaya A., Jain A., and Zhang H., "On Image Classification: City Images vs. Landscapes," *Pattern Recognition*, vol. 31, no. 12, pp. 1921-1935, 1998.

[18] Veltkamp R. and Tanase M., "Content Based Image Retrieval Systems: A Survey," *Technical Report*, Utrecht University, Utrecht the Netherlands, 2002.

[19] Wang J., Li J., and Wiederhold G., "SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture LIbraries," *Presented at the Advances in Visual Information Systems*, vol. 23, no. 9, pp. 171-193, 2000.

[20] Zhao H., Kim P., and Park J., "Feature Analysis Based on Edge Extraction and Median Filtering for CBIR," *in Proceedings of the 11th International Conference on Computer Modelling and Simulation*, Cambridge, pp. 245-249, 2009.

[21] Zhong D. and Defee I., "DCT Histogram Optimization for Image Database Retrieval," *Pattern Recognition Letters*, vol. 26, no. 14, pp. 2272-2281, 2005.

Fazal Malik received his BSc and MSc in computer science degrees from University of Peshawar, Pakistan. He is currently a PhD student at the Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Malaysia. His current research interests include image retrieval and indexing.

Baharum Baharudin received his BSc and MSc of science degrees from Central Michigan University, USA and the PhD degree from the University of Bradford, UK. Currently, he is an associate professor at the Department of Computer and Information Sciences, Universiti Teknologi PETRONAS Malaysia. His research interests include image processing, data mining and knowledge management.