Research Journal of Applied Sciences, Engineering and Technology 4(9): 1215-1224, 2012
ISSN: 2040-7467

Submitted: October 02, 2011 Accepted: December 20, 2011 Published: May 01, 2012

Features Analysis for Content-Based Image Retrieval Based on Color Moments

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

Abstract: In this study an efficient and accurate algorithm is proposed for Content-Based Image Retrieval (CBIR). The CBIR is performed in two steps: features extraction in images and similarity measurement for searching of similar images in image database. For efficient and effective CBIR system the features extraction must be fast and the searching must be accurate. In the proposed algorithm, the effective retrieval of the similar images from the database is based on the efficient extraction of the local statistical color moment features without using the spatial features of images. The basic idea in this algorithm is to convert the color RGB (Red Green and Blue) image into grayscale image to reduce the computations in feature extraction and to increase the efficiency. The grayscale image is divided into non-overlapping blocks of different sizes. The local statistical color moment features are extracted in all blocks. The features are combined into a feature vector. The similarity is measured by using Sum-of-Absolute Difference (SAD) to measure the similarity between query image and database images. In the experiment, the efficiency of feature extraction and accuracy of the image retrieval are measured for different block size methods using the proposed algorithm. The Corel database is used for testing. The results show that the proposed CBIR algorithm provides higher performance in terms of efficiency and accuracy.

Keywords: Color moments, Content-Based Image Retrieval (CBIR), feature vector, Sum-of-Absolute Difference (SAD)

INTRODUCTION

Huge number of digital images is added to the database of images per min due to the advancement in internet, rapid growth in graphic capturing devices, high speedy popularization and improvement in media storage. This huge collection of images has increased the need of different fields of life like medical, crime preventions, education, medicine, art, architecture, engineering etc., to retrieve these images efficiently and accurately. For the retrieval of these relevant images from the database, the first approach is text-based in which traditional keywords are used to search similar images. In this approach the images are stored with labels. This process is called annotation and it is performed manually. But this approach contains two limitations, one is that a lot of human labor is required to annotate the huge number of images and the second is that human has different perception about images. This is very difficult task to store these huge images with multi perceptions. To overcome these disadvantages another automatic approach is introduced called Content Based Image Retrieval (CBIR) (Liu *et al.*, 2007). In CBIR the similar images are retrieved by the visual information of images such as texture, color and shape. These features are extracted from images by using different algorithms and are used in matching of images. The CBIR has become a challenging and hot area of research for the researchers due to the difficulty in matching of the relevancy of images by computed features (Zhao *et al*., 2009).

The CBIR system can be performed in two steps: feature extraction and similarity measurement. In first step different methods are applied on images to extract the visual features of images. Database of the images is created in which these features are stored with images and this database is also called feature database. This process is also called indexing in which searching or matching of images is fast. In the second step of CBIR, when a query image is given by the user to search and retrieve the relevant images from the database, a feature vector of the query image is calculated by using the same procedure as in first step. Then this feature vector is compared with the feature vectors in database. This process is called similarity measurement in which the similarity is measured by computing the distance between the query image feature vector and feature vectors of database. If

Corresponding Author: Fazal Malik, Department of Computer and Information Sciences, University Technology, PETRONAS, Malaysia

the distance is small the images are relevant (Mohamed *et al*., 2009).

For the last decade many different types of CBIR systems have been developed like production systems, demonstration systems, commercial systems and research systems. The examples are: ADL system, Virage, SIMPLIcity, QBIC system, BDLP, AltaVista etc; a detailed survey about CBIR systems is given in (Remco *et al*., 2002) with proper features descriptions.

In Thawari and Janwe (2011) the RGB color image is converted into HSV (Hue Saturation Value) image because the description of color in HSV color space is more close to the human visual perception as compared to RGB color space. Color Histograms are constructed in each color component, H S and V. Each histogram is quantized into 96 bins or blocks. The efficiency of this process is low because of large number of computations. The color moments mean standard deviation and median are calculated in each block of each color component. Total 96×3 = 288 features per component and 288×3 = 864 features per image are calculated.

In this study our proposed CBIR algorithm is also based on statistical color moments. Fast and accurate retrieval of similar images from database is an active area of research. Without efficient feature extraction and proper indexing structures, similar images retrieval is time consuming because query image is compared with all images of database. The computational cost of feature extraction will be increased when the database is large. The problems of image retrieval which have been studied widely in past are given: reduction of the computational cost of local features extraction, proper representation of features and similarity measurement of the most similar images.

To approach these issues and to get the efficient and accurate image retrieval, we proposed a method to extract statistical local features without considering local structural information of image. These features are used and represented to get good performance in terms of efficiency and accuracy. To reduce the computational cost of feature extraction and to increase efficiency, the RGB color image is converted into grayscale single component image. For feature extraction we do not apply histogram method, we just simply divide the image into blocks of different sizes like Whole-Image-as-One-Block, 2-Blocks-Columns-Wise, 2-Blocks-Row-Wise, 2×2, 4×4, 8×8, 16×16, 32×32, and 64×64 blocks. In each block, only two color moments, mean and standard deviation are computed using pixel values. Thus different features for different block methods are extracted, for example for 2×2 method, total 2×2×2 = 8 features, for 4×4 method, 4×4×2 = 32 features, for 8×8 method, 8×8×2 = 128 features and so on. After various experiments, results are analyzed and performance is measured in terms of precision and recall.

LITERATURE REVIEWS

Many methods and algorithms have been developed for CBIR to retrieve similar images from image database.

There are various low-level features which can be used for retrieval but one of them is color information which is widely used in CBIR by researchers. It is very prominent and extensively studied feature. One reason of its importance is that it is invariant to the orientation and scaling of image (Lei *et al*., 1999).

Color information of image can be extracted by different techniques but the mostly used and prominent technique is color histogram. It is extensively used for CBIR.

Color histogram based method is proposed in (Park *et al*., 2008) in which color and shape features are used. Also a new set of features such as size, mean, variance of objects are extracted for retrieval.

A review paper about CBIR having 200 references, comprehensively has discussed the computation of feature extraction, similarity measurement of the features and relevant feedback to enhance the CBIR systems. The semantic gap between low level features and high level concepts was also discussed (Smeulders *et al*., 2000).

Color features are computed in image by using color histogram technique while the texture features are extracted by using Gabor wavelet transform technique. These features are combined together to retrieve similar images (Murala *et al*., 2009).

Color features are extracted by using color histogram. Median filter is applied to reduce the noise, before applying color histogram. But during filtering some edge information is lost. To restore edge information, edge extraction method is applied. Histogram is divided into bins to calculate the average of pixels to create a feature vector for retrieval (Zhao *et al*., 2009).

Color, shape and texture features are combined for CBIR. Gabor filter is used to get Regions of Interest (ROIs). In each ROI, the texture feature are calculated by using Gabor features, the color features are calculated by using histogram and color moments and the shape features are calculated by using Zermikes moments (Choraś *et al*., 2007).

Color moments are prominent representation of the color information in image. Many CBIR systems have successfully used color moments for the retrieval of relevant images, for example QBIC (Remco *et al*., 2002). In color moments, the first moment is the first order moment represents the mean color pixel values in image, the second is the second-order moment represents variance and the third is the third- order moment represents the skewness of the pixel values in image (Balamurugan *et al*., 2010).

To characterize the color image, color moments are used to represents color features image. The three color

moments mean variance and standard deviation represent the color distribution in image (Dubey *et al*., 2010).

RGB color image is the combination of three color components Red, Green and Blue (RGB). The color moments are extracted from each component that is R, G and B of the RGB color image. Mean, Variance and standard deviation are calculated row-wise and columns-wise in R, G and B components. These features are used to extract relevant images (Kekre and Patil, 2009).

PROPOSED ALGORITHM

Block transformation: When the input color RGB image is acquired for processing, in the first step it is converted into grayscale image as shown in Fig. 1. The RGB image consists of three color components Red, Green and Blue. Each component of the image is a two dimensional matrix of pixels values from 0 to 256. To extract the features of this color RGB image the features will be extracted in all these three components separately. For example to extract 64 features in all three components then total features will be 64×3 = 192. Due to these three components the number of computations will also be increased and efficiency will be low for features extraction. Therefore the RGB color image is converted into grayscale which is a single component of 0 to 256 pixels values. This single component will reduce the computations of features extraction and also will increase the efficiency.

In the next step the grayscale image is divided into simple non-overlapping blocks of different sizes like, Whole-Image-as-One-Block, 2-Blocks-Columns-Wise, 2-Blocks-Row-Wise, 2×2, 4×4, 8×8, 16×16, 32×32, and 64×64 blocks. Each block is in 2-dimensional matrix of 0 to 256 values. These values in each block will be used in computations of the color moments to retrieve the similar images from the image database.

Feature extraction: The color moments technique is widely used for extraction of color information since it has invariance to rotation, scaling and translation of image. Color moments also can be used to represent the basic geometrical properties of color image (Kodituwakku and Selvarajah, 2010). Our proposed algorithm is based on color moments which include the first order moment called mean and second order moment called standard deviation. These two statistical features mean and standard deviation are computed in each block by using pixel values. These mean and standard deviation are features to be extracted from blocks and will be used for retrieval of similar images from database without using spatial information of image. Mean value is used as a color feature because it describes somewhat about brightness of image in a block. The high value of mean represents bright image and low value represents dark image. Standard deviation is also used as a feature because it represents contrast of image in a block. The high value of standard deviation shows that the image has high contrast and low value means low contrast (Sergyan, 2008).

These two statistical color moments are extracted from the blocks of grayscale image for all of our proposed different size block methods. These features are extracted by using pixel values of blocks from 0 to 256. The block diagram of algorithm is shown in Fig. 1.

Different number of features is calculated for different block size methods using mean and standard deviations. For example for 4×4 block method, total 4×4×2 = 32 features, for 8×8 block method, total 8×8×2 = 128 features, for 16×16 block method, total 16×16×2 = 512 features and so on, are calculated.

Let denote mean by μ_j and standard deviation by σ_j in a particular block *j*, where *j*=1, 2, 3…, m, are blocks, then they can be calculated (Kodituwakku and Selvarajah, 2010; Bannour *et al*., 2009) as:

$$\mu_j = \frac{1}{N}\sum_{i=1}^{N} x_{ji} \tag{1}$$

$$\sigma_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(x_{ji} - \mu_j\right)^2} \tag{2}$$

where x_{ji} is the pixel value of *ith* pixel in block *j* and *N* is is the total number of pixels in each block *j*.

After the calculation of these color features in all blocks then they are combined together to construct a feature vector *FV* as:

$$FV = \{\mu_1, \mu_2, \mu_3 \ldots\ldots \mu_m, \sigma_1, \sigma_2, \sigma_3 \ldots\ldots \sigma_m\} \tag{3}$$

The feature vectors *FVs* of all images are constructed in first step of our algorithm as shown in Fig. 1 and are stored in database. The feature vector *FV* is also calculated for the user query image by using the same algorithm in the second step as shown in Fig. 1. This query feature vector is compared with all feature vectors in database to retrieve similar images.

Similarity measurement: The statistical color features are used to retrieve similar images to the user query image from the database without using spatial information of images. Simple local statistical features mean and standard deviations of blocks are calculated and are used for similarity measurement.

Once the feature database of the images is created with feature vectors using (1), (2) and (3), 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 also computed by using same (1), (2) and (3).in the second step of same algorithm as shown in Fig. 1.

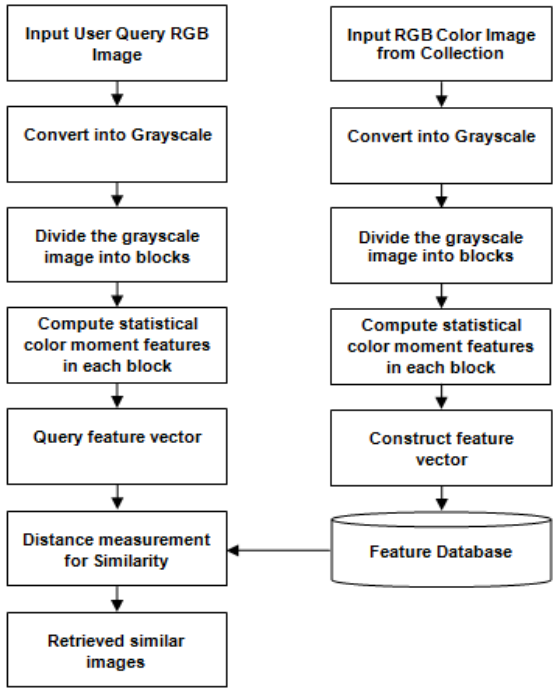

Fig. 1: Block diagram of proposed algorithm

The similarity between the query image and the database images is measured by computing the distance between the query image feature vector and the database image feature vectors. For this purpose the Sum-of-Absolute Differences (SAD) (Kodituwakku and Selvarajah, 2010) is used to calculate the difference between the query image feature vector and database feature vectors for the similarity. Let query feature vector is represented by Q_f and database feature vector by D_f then the distance is calculated as:

$$\Delta d = \sum_{i=1}^{n} \left| \left(Q_f(i) - D_f(i) \right) \right| \quad (4)$$

where n is the number of features, i = 1, 2…, n. Both query and database images are similar for $\Delta d = 0$ and small values of Δd for database images represent relevancy with the query image. The distance values of Δd are calculated for all images with query image. These distance values are arranged in ascending order. The smallest values will be on top which correspond to the most relevant or similar images and at bottom correspond to irrelevant. The top most images are displayed to the users which are the required images.

RESULTS

A database having 1000 images is used to test our proposed CBIR algorithm. This database is provided by (Wang *et al*., 2000; Li and Wang, 2003). This is free for the researchers. The database consists of 10 categories and each has 100 images. The categories are people, beaches, buildings, buses, dinosaurs, elephants, roses, horses, mountains, and foods. All these categories are used for experiments. All images are in RGB color space. They are in JPEG format with size in 256×384 and 384×256.

Phases of algorithm: The algorithm is performed in two phases.

- **Phase-1:** In first phase all images are acquired one after another from the collection of images for feature extraction. The features are extracted using (1) and (2) and these features are stored in database in the form of feature vectors using (3) to create a feature database as shown in Fig. 1.
- **Phase-2:** In the second phase the user is asked to input query image to retrieve relevant images from the feature database by using the same algorithm. The features are extracted in the same way using (1) and (2) and the feature vector is constructed using (3). This feature vector is compared with feature vectors of database by computing the similarity using (4). The similar images are displayed to the user according to the query image as shown in Fig. 1.

Evaluation measurements: The performance of CBIR systems can be evaluated by using two measurements:

Precision and Recall (Thawari and Janwe, 2011). They can be defined as:

- **Precision:** Precision is measured as the fraction of the relevant retrieved images to the total retrieved images in the query:

  $$\text{Precision} = A/B \quad (5)$$

  where *A* is "the relevant retrieved images" and *B* is "the total retrieved images".
- **Recall:** Recall is measured as the fraction of the relevant retrieved images to the total relevant images in the database:

  $$\text{Recall} = A/C \quad (6)$$

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

In experiments the two phases are performed for all proposed block methods separately like Whole-Image-as-One-Block, 2-Blocks-Columns-Wise, 2-Blocks-Row-Wise, 2×2, 4×4, 8×8, 16×16, 32×32, and 64×64 blocks. Query images from all 10 categories are used and the average precisions and recalls are calculated using (5) and (6). The average precision and recall of the results in percentage are as shown in Table 1 and 2.

Table 1 shows the average precision in percentage for 10 categories against 9 different blocks methods by calculating two color moments that is mean and standard deviation. The result in terms of precision of the proposed algorithm is good for all 10 categories and best for dinosaurs, roses and elephants. Similarly result in terms of precision for block methods is also good and best for the 8×8, 16×16, 32×32 and 2-Blocks-Columns-Wise methods.

Table 2 shows the average recall in percentage for 10 categories against 9 different blocks methods by using two statistical color features, mean and standard deviation. Our proposed algorithm gives good result in terms of recall in overall for all 10 categories and best for dinosaurs, buildings and beaches. Result in terms of recall for block methods is also good and best for the 8×8, 16×16, 32×32 and 64×64 methods.

Figure 2 shows the comaprision of precision and recall for 10 categories. The graph shows that average precision and recall for dinosaurs and elephants are almost same and good. While roses have high precision and buildings have good recall.

Table 3 shows the average precision and recall for different block size methods of image. The good precision and recall results are for 8×8 and 16×16 blocks.

Table 1: Average precision in percentage for 10 image categories against different image block methods

Categoies	Whole-Image-as -one-Block	2-Blocks- Columns-Wise	2-Blocks -Row-Wise	2×2 Blocks	4×4 Blocks	8×8 Blocks	16×16 Blocks	32×32 Blocks	64×64 Blocks	Average
People	56	56	56	56	56	56	44	44	56	53
Beaches	56	56	44	44	56	44	44	44	44	48
Buildings	56	44	56	44	44	56	56	56	44	51
Buses	44	67	44	56	44	67	56	44	56	53
Dinosaurs	100	100	100	100	100	100	100	100	100	100
Elephants	33	78	78	67	78	78	78	78	78	72
Roses	100	89	89	89	89	89	100	100	100	94
Horses	33	44	33	44	44	44	44	44	44	42
Mountains	44	44	33	33	44	56	44	44	44	43
Foods	33	33	33	33	44	44	44	44	33	38
Average	56	61	57	57	60	63	61	60	60	59

Table 2: Average recall in percentage for 10 image categories against different image block methods

Categoies	Whole-Image-as -one-Block	2-Blocks- Columns-Wise	2-Blocks -Row-Wise	2×2 Blocks	4×4 Blocks	8×8 Blocks	16×16 Blocks	32×32 Blocks	64×64 Blocks	Average
People	56	71	71	71	71	71	80	67	83	71
Beaches	71	71	67	67	83	80	80	80	80	76
Buildings	100	67	83	67	67	71	71	83	80	77
Buses	57	86	67	83	57	100	83	57	71	74
Dinosaurs	90	100	100	100	100	100	100	100	100	99
Elephants	50	64	64	55	64	70	70	70	70	64
Roses	90	57	57	57	57	57	64	64	64	63
Horses	50	80	60	80	67	80	80	67	67	70
Mountains	80	80	60	60	80	83	67	80	80	74
Foods	60	75	75	60	80	80	80	80	75	74
Average	70	75	70	70	73	79	78	75	77	74

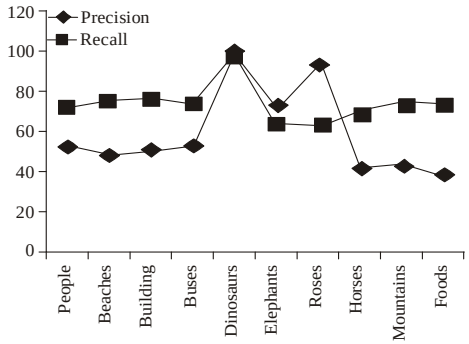

Fig. 2: Comparison of average precision and recall for 10 image categories using different image block methods

Table 3: Average precision and recall of different image block methods.

Blocks methods	Precision	Recall
Whole-image-as-one-block	56	70
2-Blocks- columns-wise	61	75
2-Blocks -row-wise	57	70
2×2 Blocks	57	70
4×4 Blocks	60	73
8×8 Blocks	63	79
16×16 Blocks	61	78
32×32 Blocks	60	75
64×64 Blocks	60	77
Average	59	74

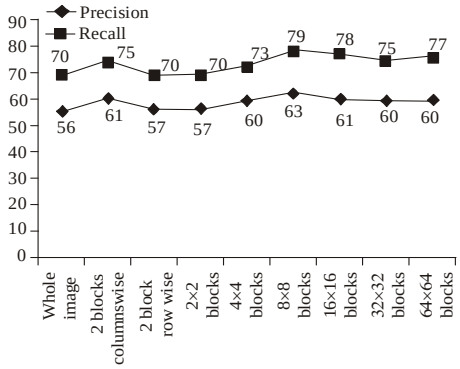

Fig. 3: Comparison of average precision and recall for different image block methods

Figure 3 shows the comaprision of precision and recall for 9 different block size methods. It can be seen that 8×8, 16×16, 32×32 and 2-Blocks-Columns-Wise methods give best performance in terms of precisions and recall.

Table 4 and Fig. 4 show the time taken by 9 different block methods to create feature database. for 10 categories of 1000 images It is clear from the graph that 8×8, 16×16, 32×32 and Whole-Image-as-One-Block methods take less time for the creation of feature database, specially 8×8 and 16×16 blocks take very less time as compared to other methods. Thus 8×8 and 16×16 block methods not only give good results in terms of precision and recall but also efficient in extraction of features.

Table 4: Feature database creation time of 9 block methods for 1000 images

Blocks methods	Time of feature database creation (min)
Whole-image-as-one-block	1:17
2-Blocks- columns-wise	22:12
2-Blocks -row-wise	22:54
2×2 Blocks	6:15
4×4 Blocks	1:24
8×8 Blocks	0:39
16×16 Blocks	1:03
32×32 Blocks	3:23

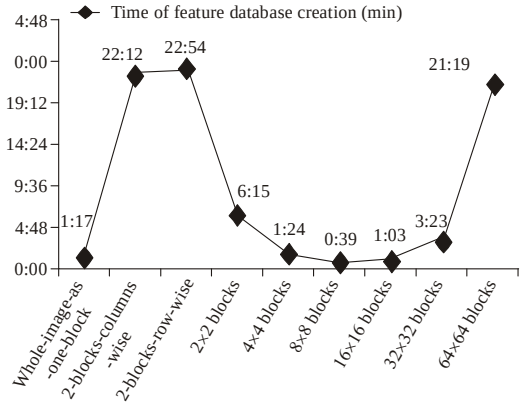

Fig. 4: Feature database creation time for block methods

Performance analysis of proposed algorithm: Our proposed algorithm is compared with algorithm of (Thawari and Janwe, 2011). They used HSV (Hue Saturation and Value) color space for RGB color images while we are converting RGB color images into grayscale images to reduce computation because they used all three color channels H, S and V and each has a dimension of 256×384. We are using only single grayscale image of same dimension. They used color histogram technique to quantize each channel into 96 blocks and each block has a dimension 32×32 pixels. They are calculating statistical moments mean standard deviation and median in each block of each channel. Totally 96×3×3 = 864 features are extracted. Thus this process of feature extraction involves a large number of computations and speed of computation is increasing. In our proposed algorithm we simply divide image into blocks of different sizes like Whole-Image-as-One-Block, 2-Blocks-Columns-Wise, 2-Blocks-Row-Wise, 2×2, 4×4, 8×8, 16×16, 32×32, and 64×64 blocks. In each block two statistical color moments mean and standard deviations are calculated as features. Different number of features is extracted for blocks methods. For example for 2×2 block method totally 2×2×2 = 8, for 4×4 block method totally 4×4×2 = 32 and for 8×8 block method totally 8×8×2 = 128 features are extracted. Thus our algorithm is efficient in computations, for example to create a database of features for 1000 images, the 4×4 block method takes only 1 min and 24 sec, 8×8 method takes 39 sec,16×16 method takes 1 min and 32×32 method takes 3 min and 23 seconds as show in Table 4 and Fig. 4.

Table 5: Average precision and recall of proposed algorithm for 10 image categories against 9 different image block methods

Categories	Precision of proposed algorithm	Recall of proposed algorithm
People	53	71
Beaches	48	76
Buildings	51	77
Buses	53	74
Dinosaurs	100	99
Elephants	72	64
Roses	94	63
Horses	42	70
Mountains	43	74
Foods	38	74
Average	59	74

Table 6: Precision and recall of the algorithm in (Thawari and Janwe, 2011)

Query Image	Precision of algorithm in (Thawari and Janwe, 2011)	Recally of algorithm in (Thawari and Janwe, 2011)
1	30	35
2	30	26
3	25	20
4	26	28
5	80	90
Average	38	40

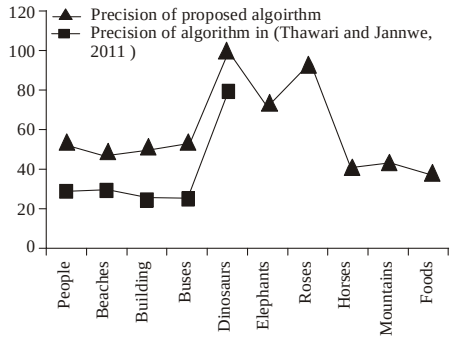

Fig. 5: Comparison of average precision of our proposed algorithm24 and precision of algorithm in (Thawari and Janwe, 2011)

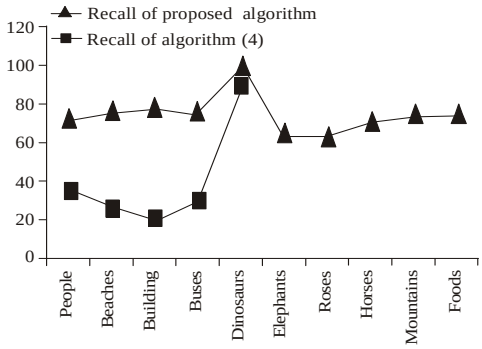

Fig. 6: Comparision of average recall of our proposed algorithm and recall of algorithm in (Thawari and Janwe, 2011)

shown that our proposed algorithm is not only efficient in feature extraction but also gives good accuracy in terms of precision and recall.

We have used 1000 images of the database while the algorithm in (Thawari and Janwe, 2011) has used 500 images of the same database.

The average precision of our proposed algorithm for all block methods is 59 and recall is 74 as shown in Table 5. While the average precision of approach in (Thawari and Janwe, 2011) is 38 and recall is 40 as shown in Table 6.

The comparison of the average precision of our proposed algorithm and the precision of the algorithm in (Thawari and Janwe, 2011) is given in Fig. 5. It can be seen that the precision for the first five categories people, beaches, buildings, buses and dinosaurs of our proposed algorithm is very high than the algorithm in (Thawari and Janwe, 2011).

Similarly the comparison of the average recall of our proposed algorithm and the recall of the algorithm in (Thawari and Janwe, 2011) is given in Fig. 6. The graph shows that the performance of our proposed algorithm in terms of recall for the first five categories people, beaches, buildings, buses and dinosaurs is best as compared to the recall of the algorithm in (Thawari and Janwe, 2011).

Thus the performance of our proposed algorithm is not only efficient in computations of feature extraction but also gives good accuracy in terms of precision and recall.

Figure 7-10 show the results of user queries. Each figure consists of a query image and the retrieved images from the database. The top single image is the query image and below 9 are the relevant images. The results show that proposed algorithm has good retrieval accuracy.

CONCLUSION

In this study a CBIR algorithm is proposed which is based on statistical color moments and these moments are extracted from blocks of images. It is shown in this paper that the statistical features has good retrieval performance without using spatial information of local blocks in images. The grayscale image is used for feature extraction to reduce the computations and increase efficiency. The grayscale image is divided into blocks of different sizes to calculate the local statistical features mean and standard deviation of pixels in each block. We have used 9 different block methods. We have extracted features for all methods and analyzed their individual retrieval performance in terms of accuracy. For similarity measurement Sum-of-Absolute Difference (SAD) is used to measure the similarity of query image with images in database. In experiment the results show that our proposed algorithm is efficient in features extraction for different block methods and gives best performance in term of accuracy, especially for 8×8 and 16×16 block methods. Our proposed algorithm compared with algorithm in (Thawari and Janwe, 2011). It has been

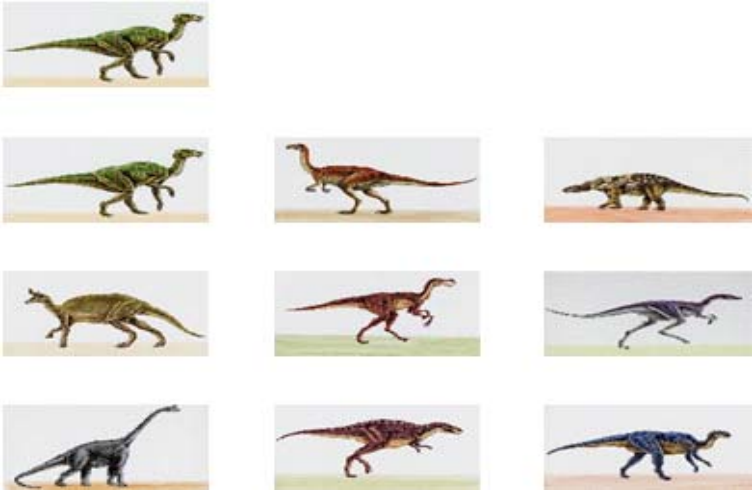

Fig. 7: Query result of dinosaurs

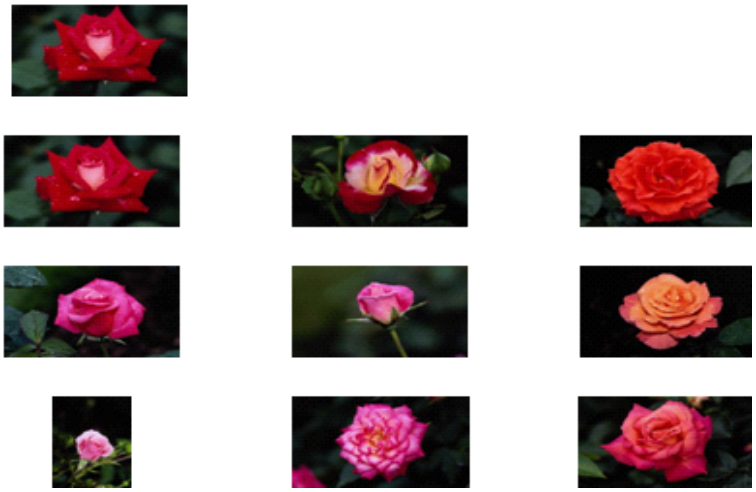

Fig. 8: Query result of roses

Fig. 9: Query result of buses

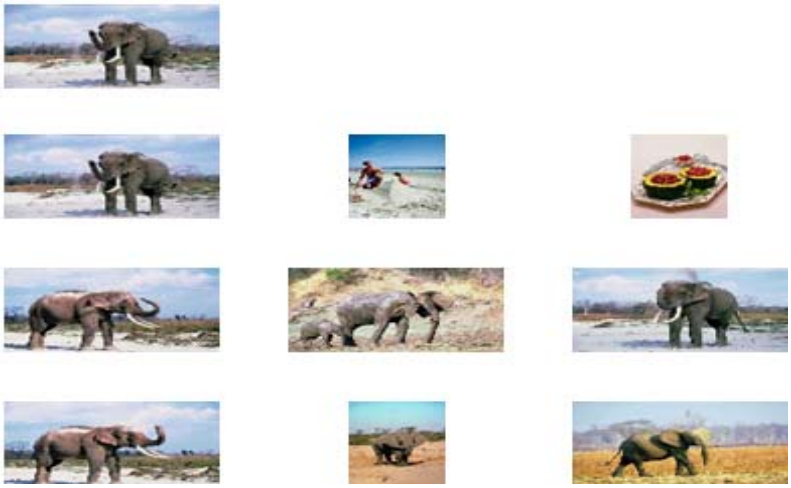

Fig. 10: Query result of elephants

ACKNOWLEDGMENT

The authors would like to thank to the Universiti Teknologi PETRONAS, Malaysia for financial support. We are also acknowledged the efforts of previous researchers in the area of content based image retrieval.

REFERENCES

Balamurugan, V., P. Anandhakumar, S. Nivaneethakrishnan and N. Venkatesan, 2010. An effective color and texture feature based image retrieval technique for medical images using relevance feedback. Engineering, pp: 4-9.

Bannour, H., L. Hlaoua and B. Ayeb, 2009. Survey of the Adequate Descriptor for Content-Based Image Retrieval on the Web: Global versus Local Features. CORIA 2009-Conférence en Recherche d'Information et Applications. Recherche, pp: 445-456.

Choraś, R.S., T. Andrysiak and M. Choraś, 2007. Integrated color, texture and shape information for content-based image retrieval. Pattern Anal. Appl., 10(4): 333-343.

Dubey, R.S., R. Choubey and J. Bhattacharjee, 2010. Multi feature content based image retrieval. Inter. J. Comp. Sci. Eng., 02(06): 2145-2149.

Kekre, H.B. and K. Patil, 2009. Standard deviation of mean and variance of rows and columns of images for CBIR. Inter. J. Comp. Sci. Eng., vol: 1-4.

Kodituwakku, S.R. and S. Selvarajah, 2010. Comparison of color features for image retrieval. Indian J. Comp. Sci. Eng., 1(3): 207-211.

Lei, Z., L. Fuzong and Z. Bo, 1999. A CBIR method based on color-spatial feature. Proceedings of IEEE. IEEE Region 10 Conference. TENCON 99. “Multimedia Technology for Asia-Pacific Information Infrastructure (Cat. No.99CH37030): 166-169.

Li, J. and J.Z. Wang, 2003. Automatic Linguistic Indexing of Pictures by a statistical modeling approach. Pattern Analysis and Machine Intelligence. IEEE Trans., 25: 1075-1088.

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

Mohamed, A., F. Khellfi, Y. Weng, J. Jiang and S. Ipson, 2009. An efficient Image Retrieval through DCT Histogram Quantization. International Conference on Cyber Worlds, pp: 237-240.

Murala, S., A.B. Gonde and R.P. Maheshwari, 2009. Color and Texture Features for Image Indexing and Retrieval. 2009 IEEE International Advance Computing Conference, pp: 1411-1416.

Park, J., Y. An, G. Kang, W. Rasheed, S. Park and G. Kwon, 2008. Defining a New Feature Set for Content-Based Image Analysis Using Histogram Refinement.Wiley Periodicals Inc.

Remco, C.V. and M. Tanase, 2002. Content-Based Image Retrieval Systems: A Survey. October.

Sergyan, S., 2008. Color histogram features based image classification in content-based image retrieval systems. 6th International Symposium on Applied Machine Intelligence and Informatics, pp: 221-224.

Smeulders, A.W.M., M. Worring, S. Santini, A. Gupta and R. Jain, 2000. Content-based image retrieval at the end of the early years. IEEE Trans. Pattern Anal. Mach. Intel., 22(12): 1349-1380.

Thawari, P.B. and N.J. Janwe, 2011. CBIR based on color and texture. Inter. J. Inf. Technol. Knowl. Manage., 4(1): 129-132.

Wang, J.Z., J. Li and G. Wiederhold, 2000. SIMPLIcity: Semantics-Sensitive Integrated Matching Libraries. Advances in Visual for Picture Information Systems: 4th International Conference, VISUAL 2000, Lyon, France,November 2-4: Proceedings, Retrieved from: http: //wang. ist. psu. edu/docs/ related/.

Zhao, H., P. Kim, and J. Park, 2009. Feature analysis based on Edge Extraction and Median Filtering for CBIR, UK Sim, 11th International Conference on Computer Modelling and Simulation, pp: 245-249.