ISSN 2229-5518

Review and Comparison of BTC Based Feature Extraction Techniques in CBIR

Aditi Mehta
M.Tech Student,
Computer Engineering Department, MPSTME,
SVKM's NMIMS (Deemed-to-be University), Mumbai, India

Prof. Paulami shah
Assistant Professor,
Computer Engineering Department, MPSTME,
SVKM's NMIMS (Deemed-to-be University), Mumbai, India

Abstract— in fields such as medical, art galleries, museums, archaeology, medical imaging, trademark databases, criminal investigations, images especially the digital images grow in quantities of thousands and sometimes even lakhs every year. Content based image retrieval is required for such large databases. This paper compares various CBIR-BTC techniques based on difference in generating feature vectors in the non-transform domain. In this paper, Euclidean distance is used as the similarity measure. Two performance evaluation parameters are used; namely precision and recall.

Index Terms— Content Based Image Retrieval(CBIR); Block Truncation Coding (BTC), Red Green Blue (RGB); Hue Saturation Intensity (HSI).Euclidean distance; Precision; Recall**.**

1 Introduction

The very large numbers of images are being generated from a variety of sources (digital camera, digital video, scanner, the internet etc.) which have posed technical challenges to computer systems to store/transmit and index/manage image data effectively to make such collections easily accessible. Image compression deals with the challenge of storage and transmission, where significant advancements have been made [1,4,5]. The challenge to image indexing is studied in the context of image database [2,6,7], which has become one of the promising and important research area for researchers from a wide range of disciplines like computer vision, image processing , image database and recognition systems.

The thirst of better and faster image retrieval techniques is increasing day by day. The ambiguity in text based retrieval emphasizes the need of a better and faster retrieval system. That is why CBIR becomes more important. Some of important applications for CBIR technology could be identified as art galleries, museums, archaeology [3], architecture design geographic information systems [5], weather forecast [5], medical imaging [5], trademark databases, criminal investigations, image search on the Internet.

2 Content Based Image Retrieval

In literature the term content based image retrieval (CBIR) has been used for the first time by Kato et.al. [4], to describe his experiments into automatic retrieval of images from a database by color and shape feature. The typical CBIR system performs two major tasks. The first one is extraction of feature vector which consists of various feature components. It is generated to represent the content of each image in the database with accuracy and uniqueness. The second task is similarity measurement (SM), where a distance between the feature vector of the query image and the feature vector of each image in the database is measured, compared and this is used to retrieve the top "closest" images.

For feature extraction in CBIR there are mainly two approaches [5] feature extraction in spatial domain and feature extraction in transform domain. The feature extraction in spatial domain includes the CBIR techniques based on histograms [5], BTC [1, 2]. The transform domain methods are widely used to extract image features. Many current CBIR systems use Euclidean distance [1-3] on the extracted feature set as a similarity measure. The Direct Euclidian Distance between image P and query image Q can be given as equation 1, where Vpi and Vqi are the feature vectors of image P and Query image Q respectively with size 'n'.

$$ED = \sqrt{\sum_{i=1}^{n} (V_{pi} - V_{qi})2} \qquad (1)$$

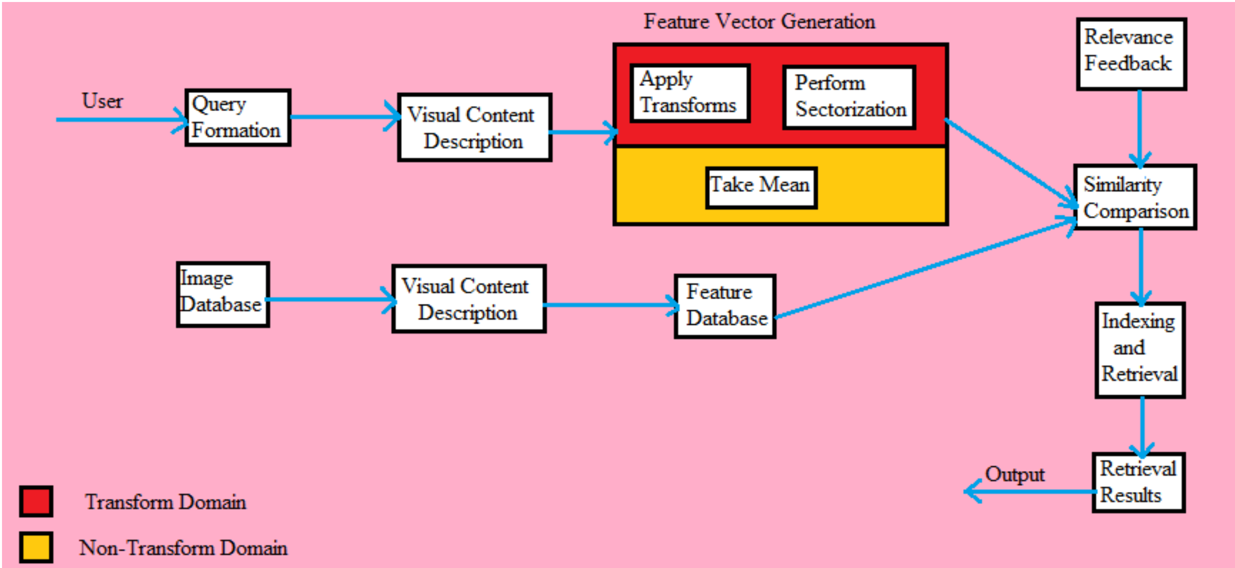

Figure 1: CBIR System

In a typical content-based image retrieval system as shown in figure above, the visual contents of the images in the database are extracted and described by feature vectors. The feature vectors of the images in the database form a feature database. To retrieve images, users provide the retrieval system with example images. The system then changes these examples into its internal representation of feature vectors. The similarities /distances between the feature vectors of the query example and those of the images in the database are then calculated and retrieval is performed. This provides an efficient way to search for the image database. Recent retrieval systems have also begun with taking the feedback from users for making further improvements in the retrieval results.

3 Block Truncation Coding

As Block truncation coding (BTC) is a simple image coding technique developed in the early years of digital imaging more than 29 years ago. BTC has played an important role in the history of digital image coding in the sense that many advanced coding techniques have been developed based on BTC or inspired by the success of BTC.

Block Truncation Coding (BTC) was first developed in 1979 for gray scale image coding.

Step 1: A 256x256 pixel image is divided into blocks of typically 4x4 pixels.

Step 2: For each block the Mean and Standard Deviation are calculated, these values change from block to block. These two values define what values the reconstructed or new block will have, in other words the blocks of the BTC compressed image will all have the same mean and standard deviation of the original image.

Step 3: A two level quantization on the block is where we gain the compression, it is performed as follows:

$$Y(i,j) = \begin{cases} a, & x(i,j) > x' \\ b, & x(i,j) < x' \end{cases} \quad (2)$$

Where x(i,j) are the pixel elements of the original block and y(i,j) are elements of the compressed block. In words this can be explained as: If a pixel value is greater than the mean it is assigned the value "1", otherwise "0". Values equal to the mean can have either a "1" or a "0" depending on the preference of the person or organisation implementing the algorithm.

Step 4: This 16 bit block is stored or transmitted along with the values of Mean and Standard Deviation. Reconstruction is made with two values "a" and "b" which preserve the mean and the standard deviation. The values of "a" and "b" can be computed as follows:

$$a = \bar{x} - \sigma\sqrt{\frac{q}{m-q}} \quad (3)$$

$$b = \bar{x} + \sigma\sqrt{\frac{m-q}{q}} \quad (4)$$

Where σ is the standard deviation, m is the total number of pixels in the block and q is the number of pixels greater than the mean ($\bar{x}$).

Step 5: To reconstruct the image, or create its approximation, elements assigned a 0 are replaced with the "a" value and elements assigned a 1 are replaced with the "b" value.

$$X(i,j) = \begin{cases} a, & y(i,j) = 0 \\ b, & y(i,j) = 1 \end{cases} \quad (5)$$

This demonstrates that the algorithm is asymmetric in that the encoder has much more work to do than the decoder. This is because the decoder is simply replacing 1's and 0's with the estimated value whereas the encoder is also required to calculate the mean, standard deviation and the two values to use.

ISSN 2229-5518

4 COMPARISON OF TECHNIQUES

Various CBIR systems were studied and all of these systems discuss different techniques of feature vector generation based on BTC. All of these techniques are discussed briefly in this section.

4.1. Simple BTC [8]

In simple BTC, the feature vector is calculated using the simple BTC technique.Euclidean distance is used as the similarity measure and the performance is evaluated using precision and recall.

4.2. BTC based CBIR [8]

In this method, Feature Vector is generated by computing interband average image (IBAI) which is the average of all components (R, G, and B) and mean of IBAI is taken as threshold. Two mean colors; one for the pixels greater than or equal to the threshold and other for the pixels smaller than the threshold are also calculated for every component. This gives one upmean color and one low mean color for each component, making it 6 feature vectors per image. Euclidean distance is used as the similarity measure and the performance is evaluated using precision and recall.

4.3. CBIR using BTC-RGB / BTC RGB Level-1(BTC-6)[8]

In this method, Feature Vector is generated by using three independent R, G and B components of color images to calculate three different thresholds and then apply BTC to each individual R, G and B planes. Rest is same as mentioned in section above. Euclidean distance is used as the similarity measure and the performance is evaluated using precision and recall.

4.4. CBIR using Spatial BTC-RGB[8]

In this method, Feature Vector is generated by dividing the image into four equal sized, non-overlapping quadrants. For every quadrant the BTC-RGB feature extraction is applied and features are stored. The size of feature vector in Spatial BTC-RGB is four times to that of BTC-RGB. Euclidean distance is used as the similarity measure and the performance is evaluated using precision and recall.

4.5. CBIR using BTC-LUV[9]

In this method, we first convert RGB components to Kekre's LUV and then the Feature Vector is generated byusing three independent L, U and V components of color images to calculate three different thresholds and then apply BTC to each individual L, U and V planes. Rest is same as mentioned in section *A*. Euclidean distance is used as the similarity measure and the performance is evaluated using precision and recall.

4.6. CBIR using Spatial BTC-LUV[9]

In this method, Feature Vector is generated by dividing the image into four equal sized, non-overlapping quadrants. For every quadrant the BTC-LUV feature extraction is applied and features are stored. The size of feature vector in Spatial BTC -LUV is four times to that of BTC-LUV. Euclidean distance is used as the similarity measure and the performance is evaluated using precision and recall.

4.7. CBIR using BTC RGB Level-2(BTC-12)[10]

In this method, Feature Vector is generated by dividing the image data into 12 parts using the six means obtained in BTC-RGB-Level 1. Euclidean distance is used as the similarity measure and the performance is evaluated using precision and recall.

4.8. CBIR using BTC RGB Level-3(BTC-24)[10]

In this method, Feature Vector is generated by dividing the image data into 24 parts using the twelve means obtained in BTC-RGB-Level 2. Euclidean distance is used as the similarity measure and the performance is evaluated using precision and recall.

4.9. CBIR using BTC RGB Level-4(BTC-48)[11]

In this method, Feature Vector is generated by dividing the image data into 48 parts using the twenty four means obtained in BTC-RGB-Level 3. Euclidean distance is used as the similarity measure and the performance is evaluated using precision and recall.

4.10. Flipping Techniques[11]

A flipped image is a static or moving image that is generated by a mirror reversal of an original across a horizontal (X) axis or vertical (Y) axis. While flipping along the X-axis; the pixel values of the images are interchanged wherein the 1st element of the 1st row is interchanged with the last element of the 1st row, the 2ndelement of the 1st row is interchanged with the second last element of the 1st row and so on till all the pixel values of all the rows are interchanged. Similarly, is done column wise when flipping the image along Y axis.

Even image = Original Image + Flipped Image

Odd Image = Original Image - Flipped Image

i. Even BTC [11]

In this method, the feature vectors are generated by using the 6 feature vectors consisting of only the even part.Euclidean distance is used as the similarity measure and the performance is evaluated using precision and recall.

ii. Odd BTC[11]

In this method, the feature vectors are generated by using the 6 feature vectors consisting of only the odd part.Euclidean distance is used as the similarity measure and the performance is evaluated using precision and recall.

iii. Even and Simple BTC[11]

In this method, the feature vectors are generated by using a combination of both the even part as well as the simple BTC thus increasing the feature vector to size 12. Euclidean distance is used as the similarity measure and the performance is evaluated using precision and recall.

iv. Odd and Simple BTC[11]

In this method, the feature vectors are generated by using a combination of both the odd part as well as the simple BTC thus increasing the feature vector to size 12. Euclidean distance is used as the similarity measure and the performance is evaluated using precision and recall.

v. Even and Odd BTC[11]

In this method, the feature vectors are generated by using a combination of both the even part as well as the odd BTC thus increasing the feature vector to size 12. Euclidean distance is used as the similarity measure and the performance is evaluated using precision and recall.

vi. Even, Odd and Simple BTC [10]

In this method, the feature vectors are generated by using a combination of the even, odd as well as the simple BTC thus increasing the feature vector to size 18. Euclidean distance is used as the similarity measure and the performance is evaluated using precision and recall.

5 IMPLEMENTATION

5.1. Database [8-12]

The CBIR techniques (*4.1-4.10*) are tested on a Generic Image Database which contains 1000 variable size images spread across 11 categories of human being, animals, natural scenery and manmade things.

5.2. Performance Evaluation Parameters

To assess the retrieval effectiveness, precision and recall are used as statistical comparison parameters for the proposed CBIR techniques. The standard definitions for these measures are given by following equations:

$$\text{Precision} = \frac{\text{Number of Relevant Images Retrieved}}{\text{Total Number of Images Retrieved}} \quad (6)$$

$$\text{Recall} = \frac{\text{Number of Relevant Images Retrieved}}{\text{Total Number of Relevant Images in Database}} \quad (7)$$

5.3. Results

Figure 2 compares techniques (4.1-4.10) discussed in section IV. It is seen that the flipping technique having the combination of odd & simple in YUV color space gives the best result amongst all the CBIR-BTC techniques.

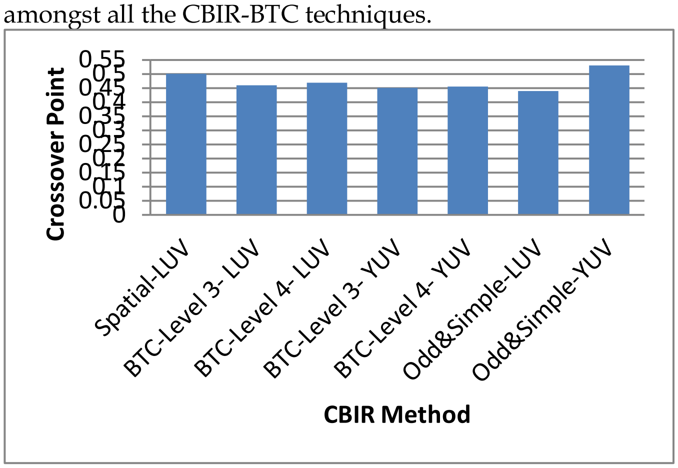

Figure 2: Graph of CBIR-BTC techniques with highest precision & recall crossover with different methods for feature extraction

6 CONCLUSION

6.1 Complexity

Because As far as complexity is concerned, it can be seen that multilevel BTC towards the levels of 2, 3 and so on has maximum complexity as compared to level 1 and flipping techniques. But, the best result is given by flipping techniques (odd and original) followed by level 3 and 4. Performance improves with increased level of BTC. Up to level -3 gradual increases in performance with increasing level is observed while the difference of performance in level-3 and level-4 is negligible due to voids being created at higher levels. Due to this negligible value, level 3 is preferred since it has lesser complexity. Thus, it can be concluded that better results are achieved at the cost of complexity in terms of multilevel BTC. Flipping techniques give the best results, also with an added advantage of lesser complexity (maximum of 18 feature vectors only).

6.2 Space

Number As far as space is concerned, it can be seen that multilevel BTC towards the levels of 2, 3 and so on requires maximum space as compared to level 1 and flipping techniques. But, the best result is given by flipping techniques followed by level 3 and 4. Performance improves with increased level of BTC. Up to level -3 gradual increases in performance with increasing level is observed while the difference of performance in level-3 and level-4 is negligible due to voids being created at higher levels. Due to this negligible value, level 3 is preferred since it has requires lesser space. Thus, it can be concluded that better results are achieved at the cost of larger space usage complexity in terms of multilevel BTC. However, Flipping techniques give the best results, also with an added advantage of lesser space usage (maximum of 18 feature vectors only). Above can be explained in tabular form as follows:

In the table below, * can refer to any of the following color spaces: YC_bC_r. $YCgC_b$, YIQ, YUV, LUV, HSV, HSI, RGB, rgb, and XYZ.

TABLE 1
SIZE OF FEATURE VECTORS FOR CBIR-BTC TECHNIQUES

	CBIR-BTC Method	No of Feature Vectors
Multilevel BTC	BTC-* / BTC-*-Level 1	6
	BTC-*-Level 2	12
	BTC-*-Level 3	24
	BTC-*-Level 4	48
Flipping Techniques	Even BTC	6
	Odd BTC	6
	Simple BTC	6
	Even and Odd BTC	12
	Odd and Simple BTC	12
	Even and Simple BTC	12
	Even Simple and Odd BTC	18
	Spatial-*	24

6.3 Colour Spaces

The results have shown that, the performance improvement is seen in Luminous & Chrominance Color spaces (LUV, YUV, YC_gC_b, YC_bC_r and YIQ) with BTC-CBIR methods compared to

Non-Luminous &Non Chrominance Color spaces(RGB, rgb, etc.).

REFERENCES

[1] H.B.Kekre, Sudeep D. Thepade, "Boosting Block Truncation Codingusing-Kekre's LUV Color Space for Image Retrieval", WASETInternational Journal of Electrical, Computer and System Engineering (IJECSE), Volume 2, Number 3, pp. 172-180, Summer 2008.

[2] H.B.Kekre, Sudeep D. Thepade, "Image Retrieval using AugmentedBlock Truncation Coding Techniques", ACM International Conferenceon Advances in Computing, Communication and Control (ICAC3-2009), pp. 384-390, 23-24 Jan 2009, Fr. ConceicaoRodrigousCollegeofEngg., Mumbai. Is uploaded on online ACM portal.

[3] H.B.Kekre, Sudeep D. Thepade, "Scaling Invariant Fusion of ImagePieces in Panorama Making and Novel Image Blending Technique",International Journal on Imaging (IJI), www.ceser.res.in/iji.html,Volume 1, No. A08, pp. 31-46, Autumn 2008.

[4] Hirata K. and Kato T. "Query by visual example – content-based imageretrieval", In Proc. of Third International Conference on H ExtendingDatabase Technology, EDBT'92, 1992, pp 56-71

[5] H.B.Kekre, Sudeep D. Thepade, "Rendering Futuristic Image RetrievalSystem", National Conference on Enhancements in Computer,Communication and Information Technology, EC2IT-2009, 20-21 Mar2009, K.J.Somaiya College of Engineering, Vidyavihar, Mumbai-77

[6] Minh N. Do, Martin Vetterli, "Wavelet-Based Texture Retrieval Using Generalized Gaussian Density and Kullback-Leibler Distance", IEEE Transactions On Image Processing, Volume 11, Number 2, pp.146-158, February 2002.

[7] B.G.Prasad, K.K. Biswas, and S. K. Gupta, "Region –based image retrieval using integrated color, shape, and location index", International Journal on Computer Vision and Image Understanding Special Issue: Colour for Image Indexing and Retrieval, Volume 94, Issues 1-3, April- June 2004, pp.193-233.

[8] Dr.H.B.Kekre, Sudeep D. Thepade, "Image Retrieval using Augmented Block Truncation Coding Techniques International Conference on Advances in Computing, Communication and Control (ICAC3), November 2009

[9] Dr.H.B.Kekre, Sudeep D. Thepade, "Boosting Block Truncation Coding with Kekre's LUV Colour Space for Image Retrieval", International Journal of Electrical, Computer, and Systems Engineering, Summer 2008.

[10] Dr.H.B.Kekre, Sudeep D. Thepade, Shrikant P. Sanas "Improved CBIR using Multileveled Block Truncation Coding",International Journal on Computer Science and Engineering (IJCSE), Volume 2 (7) 2010, ISSN 2471-2476.

[11] Dr.H.B.Kekre, Sudeep D. Thepade,Varun Lodha, Pooja Luthra, Ajoy Joseph, Chitrangada Nemani, "Augmentation of Block Truncation Coding based Image Retrieval by using Even and Odd Images with Sundry Colour Spaces", International Journal on Computer Science and Engineering (IJCSE), Volume 2 (8) 2010, ISSN 2535-2544.

[12] Dr.H.B.Kekre, Sudeep D. Thepade, Shrikant P. Sanas "Improving Performance of Multileveled BTC Based CBIR Using Sundry Colour Spaces", Volume 4 (6).