International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 11 Issue 4 –NOVEMBER 2014.

A New Content Based Image Retrieval System Based on Quad-tree Classified Vector Quantization using Edge Oriented Classifier

Sowmya A[#1], Vijay Kumar A[*2], Nagaraju D[#3]

[#] *Department of Computer Science and Engineering, Acharya Nagarjuna University*
Guntur India
[1]asowmya91@gmail.com
[3]dnr.anu@gmail.com

[*] *Department of Information Technology, Nalla Malla Reddy Engineering College*
Hyderabad, India
[2]vijayatmakur@gmail.com

***Abstract*— In large image databases, content based image retrieval is efficient evaluating image contents, making the retrieval system slow indeed. The proposed method exploits visual importance of images in compressed domain for efficient and fast image retrieval. The image is first segmented based on Quad tree structure to classify the image into two patterns, low-detail and high-detail blocks. Then each segmented block is evaluated and further classified if required to recognize patterns. For recognizing high detail blocks, an edge oriented classifier with 32 edge binary templates is employed to extract a variety of visually important regions. Then the blocks of the image are encoded by the predefined QCVQ codebook generated using enhanced LBG algorithm for obtaining indices. Then index histogram is constructed, counting the frequencies of indices. The histogram intersection gives the result. As the indices are obtained from the encoded classified patterns, it acts as the feature for the image retrieval system. The encoding and indexing methodology used extracting visually important details, makes the proposed system efficient and faster than the conventional VQ based methods.**

***Keywords*— Quadtree Segmentation, Content Based Image Retrieval, LBG Algorithm, Vector Quantization, Image Indexing, Histogram Intersection.**

I. Introduction

For large image databases, to efficiently and precisely retrieve the desired images, the development of a content based image retrieval system has become an important research issue [1, 2]. In CBIR, the image retrieval is carried out according to the image contents. It adopts a variety of visual contents like color [3, 4, 5], texture [7, 8, 9], shape [6] ,moments [11],spatial relationship[10, 12, 13],,etc., to help retrieve images. In order to improve the retrieval accuracy of CBIR systems, region based image retrieval methods via image segmentation were introduced. These methods attempt to represent images at object level, which is intended to be close to the perception of human visual system.

Color is one of the most widely used image features in CBIR. The color content of an image is relatively invariant with size, translation and rotation about the viewing axis, relatively robust to background complication [2] and faster to compute. But they ignore spatial correlation in images .To overcome this problem several feature descriptors that consider spatial relationship such as the color histogram [4, 5], the color coherence vector (CCV) [10], the color auto correlogram (CAC) [12], and the chromaticity moment (CM) [11] were proposed.

In recent years, compressed-domain techniques for image retrieval, such as Vector Quantization (VQ) has become an important research issue. VQ is a powerful tool often used for lossy image compression using a codebook [28, 27]. Only the index of the codebook is sent instead of the quantized values. This conserves space and achieves more compression which is an effective means for image indexing and retrieval. However, VQ-based methods ignore visually important regions which are edge intensive, limiting the system retrieval performance.

Traditional block-based image coding algorithms, such as vector quantization, transform coding, and block truncation coding techniques require the partitioning of the original image into a number of, usually square blocks of pixels which are then encoded as separate entities. In all these schemes, the block size is a fundamental design parameter.

In this paper, we propose the QCVQ-based method that uses quadtree segmentation, a simple technique for image representation at different resolution levels, which partitions an image into variable block size regions based on a quadtree structure. Studies have demonstrated that quadtree based image segmentation can be effective and efficient mechanism for isolating blocks of distinct perceptual significance and thereby allowing different coding strategies that are perceptually suited to the individual segment categories. It provides an effective compromise between the accuracy with which the region boundaries are determined and the overhead required to specify the segmentation information.

In this paper, we propose an edge-oriented classifier, [14, 15, 16], which is used for generating visually important features. The proposed method can exploit both the spatial distribution between pixels in the block and can also describe blocks with different levels of visual importance in natural images and can have better image retrieval performance compared to the conventional VQ-based methods.

II. RELATED WORK

In this section, conventional image retrieval methods and the state-of-the-art VQ-based method are introduced. The color histogram [10, 17] is easy to implement for image indexing but the representation is dependent on the color of the object being studied, ignoring its shape and texture. Color histograms can potentially be identical for two images with different object content which happens to share color information. Conversely, without spatial or shape information, similar objects of different color may be indistinguishable based on color histogram comparisons. Hence, two spatially different images may have the same color histograms.

A color correlogram[12] is a technique which computes the distribution of a given color as a function of the distance between two pixels, is quietly efficient even with very coarsely quantized color information. It expresses how the spatial correlation of pairs of colors changes with distance but captures spatial correlation between identical colors only.

In [13], an overview of some efficient color and texture descriptors were provided in the MPEG-7 standard. The dominant color descriptor, the color structure descriptor, the scalable color descriptor, and the color layout descriptor are included as color features in MPEG-7. Several texture descriptors, texture browsers, homogeneous textures, and local edge histogram descriptors were also used for image retrieval applications.

In recent years, many VQ-based retrieval methods were proposed [18, 19, 20, 22, 21]. Indris and Panchanathan [19] described each image in the database as a codevector usage map using a universal codebook and the XOR operation is used to measure the similarity between two images. But the retrieval performance degrades since two different images may have the same codevector usage map. This problem was solved by counting the frequency of the index of each codevector to generate an index histogram [20]. Schaefer [22] constructed a codebook for each individual image and the similarity of images is computed by the modified Hausdorff distance (MHD) between image codebooks. The encoding distortion distance, measured based on MSD measure was used to improve the retrieval performance [18].

Teng and Lu [21] used the modified Linde-Buzo-Gray (LBG) algorithm which uses the splitting technique to build a color codebook with each codevector having the values of the three color channels in the hue-saturation-value (HSV) color space for image indexing and retrieval based on VQ.For a given image, an index histogram of VQ-compressed indices is calculated as a feature vector. The VQ index histogram can describe the spatial distribution between pixels in the block but cannot describe blocks with different levels of visual importance.

In this paper, we propose a method that uses quadtree segmentation to extract low-detail and high-detail regions. Then, an edge-oriented classifier [14, 15, 16], is used for further classifying each high detail region into several classes according to edge orientations and patterns describing textures, which are visually important features. The conventional VQ method [21] considers only the spatial distribution between pixels in the block and ignores visually important patterns. By contrast, the proposed method can exploit both the spatial distribution between pixels in the block and can also describe blocks with different levels of visual importance in natural images and can have better image retrieval performance.

III. PROPOSED METHOD

In this section, we first present an overview of the proposed QCVQ-based image retrieval system. Its diagram is plotted in Fig. 1. Both in the database and query image modules, color RGB images are segmented using the quadtree segmentation method, which depends on the details of blocks. Two basic block classes are generated: low-detail and high-detail blocks. In low-detail regions, there are block sizes of 16×16, 8×8, and 4×4. In high-detail regions, there is only one block size of 4×4. We downsample low-detail regions with 16×16 and 8×8 homogeneous blocks in each RGB plane to 4×4 blocks. Furthermore, the high-detail blocks of size 4×4 are further classified into 5 classes which belong to different edge orientations using the edge oriented thresholding classifier.

Then, image indexing is performed on each class of blocks, i.e., segmented blocks in each class are encoded as an index using the corresponding pre-trained QCVQ codebook and the frequency of all the indices is counted to build an index histogram as a feature for the image. The major components of the proposed image indexing scheme and the design method of codebooks which includes one low-detail subcodebook (Low-detail subCB) as well as five high-detail subcodebooks (C1-C5 subCB) are introduced in detail in the following subsections.

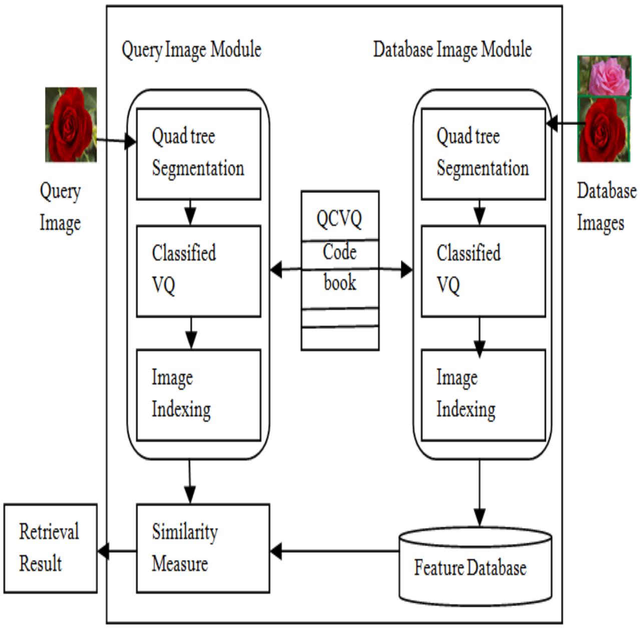


Fig. 1 Diagram of the proposed QCVQ-based image retrieval system

A Quadtree segmentation

Quadtree segmentation [29, 7] is one of the segmentation techniques. A quadtree is a tree data structure in which each internal node has exactly four children. Quad trees are most often used to partition a two-dimensional space by recursively subdividing it into four quadrants or regions. Quad tree segmentation procedure is based on the variances of blocks in gray-level image information and it uses a hierarchical data structure to efficiently address variable block size regions. The whole segmentation technique is summarized into following steps.

a. A color image is first transformed into a gray level image.

b. Gray level image is segmented uniformly into blocks of size 16×16.

c. The splitting process is continued until the smallest block of size 4×4 is acquired or the variance of the block is not greater than a predefined threshold.

Figure 2 shows original image (a) and their overall quadtree segmentation images (b) for two types of regions, blocks with white border for low-detail regions.

A smaller threshold can be used to catch more fine-detail regions, and a larger threshold is used to capture more homogeneous regions. The threshold should be chosen properly such that fine-texture regions and homogeneous regions can be well classified. The threshold of quadtree segmentation is determined as 2.5 based on the variances of blocks .

(a) Original image (b)Overall segmentation image

Fig. 2 Example of quadtree segmentation results for lena image.

B Edge Oriented classifier design

The edge oriented classifier design technique is summarized into following steps.

1) Calculate horizontal and vertical gradients as follows:

$$Edge_h(x,y) = \max(|g(x+1,y) - g(x,y)|, |g(x,y) - g(x-1,y)|)$$
$$Edge_v(x,y) = \max(|g(x,y+1) - g(x,y)|, |g(x,y) - g(x,y-1)|)$$

Where $Edge_h(x,y)$ and $Edge_v(x,y)$ for x, y=1,2,3,4 are vertical and horizontal gradients, respectively. Then the weighing factor w(x, y) is defined as the maximum of $Edge_h(x,y)$ *and* $Edge_v(x,y)$.

$$w(x,y) = \max(Edge_h(x,y), Edge_v(x,y))$$

2) Binarize the block by calculating the threshold T_{local}

$$T_{local} = \sum_{x=1}^{4}\sum_{y=1}^{4}\frac{w(x,y)}{W}g(x,y)$$

Where $W = \sum_{x=1}^{4}\sum_{y=1}^{4} w(x,y)$

and w(x, y) is a weighting factor associated with the gray level value g(x, y),and W is the sum of weighting factors.

3) Then the binarized blocks are compared with predefined 32 binary edge templates (T1 to T32) using the distance measure as

$$d(X,Z) = \sum_{i=0}^{k-1} XOR(x_i, z_i)$$

where X is a binarized block, Z is one of the 32 binary edge templates, K is the number of pixels in the block, i.e., K = 16, and XOR is the exclusive or operation. If the distance d(X, Z) is greater than a threshold θ, the block X is assigned as a mixed class that contains random textures.

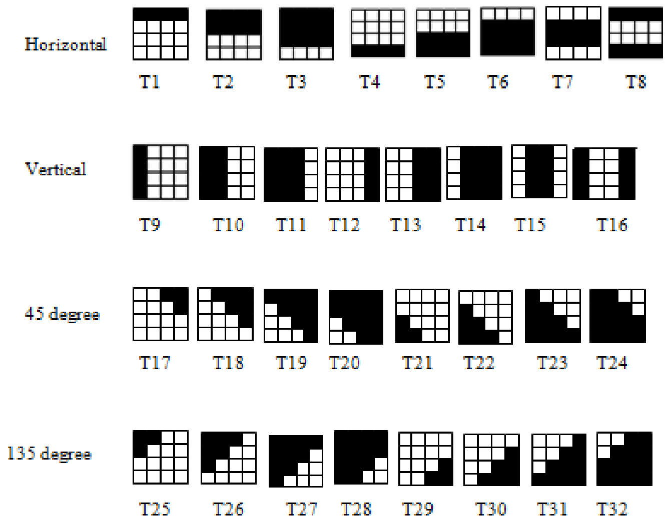


Fig. 3 The 32 binary edge template classes (T1-T32)

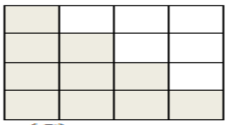
To achieve computational complexity and memory reduction for QCVQ, we downsample the blocks with sizes 16×16 and 8×8 to those of size of 4×4. Low detail regions are joined into one class. Here in this proposed method we use 32 edge templates obtained when the block is considered horizontally, vertically, 45 degree and 135 degree. It is true that the retrieval performance will be improved if other edge patterns are included to describe the image in more details.

149	85	92	89
150	135	87	90
152	126	149	75
116	123	142	149

(a)

1	0	0	0
1	1	0	0
1	1	1	0
1	1	1	1

(b)



(C)

Fig 4 (a) Original edge images block. (b) Bit Pattern (c) Matched Pattern

Then, the 32 edge templates are grouped based on symmetric reflection, rotation, and inversion of edge image blocks [23]. We group T1, T3, T4, T6, T9, T11, T12, and T14 to class C1, group T2, T5, T7, T8, T10, T13, T15 and T16 to class C2,group T17, T20, T21, T24, T25, T28, T29,and T32 to class C3, and group T18, T19, T22, T23, T26, T27, T30,and T31 to class C4. The 32 binary edge templates are regrouped into four edge templates. Together with one mixed class, which is denoted by class C5, high-detail blocks are classified into five classes C1-C5. Hence, we train six subcodebooks. One is for low-detail blocks and others are for high-detail blocks. The number of quantized edge patterns is set to 5 because in [23], Hrovje et al. showed that, if five codebooks are adopted, the goals of memory reduction and robustness for image coding can be achieved simultaneously. In this paper, we use codebooks with five classes to balance the computational complexity and the retrieval performance.The edge oriented classifier design is shown as in Fig.5.

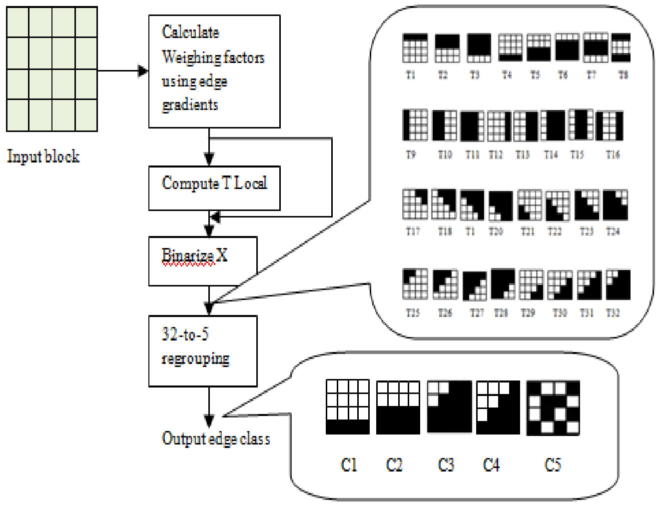


Fig. 5 The edge oriented classifier

C QCVQ codebook design

The most important work in QCVQ compression is to design a versatile codebook. Its basic idea is to establish a codebook consisting of code vectors such that each code vector can represent a group of images. It is clearly not easy to find suitable vectors for a specific image, but this problem can be solved by Lloyd and Max quantization[30]. Lloyd-Max quantization allows an optimal set of vectors to be found for any set of test images. This gives a library of blocks that are easy to fit to the desired image. The method for finding the best library and how to apply this library has been developed in several works.

We build a color codebook that contains six subcodebooks. Each of which describes one of six classes of image blocks obtained by quadtree segmentation and the edge oriented classifier discussed above in the HSV color space using the modified LBG algorithm [21]. For each subcodebook, each codevector represents the values of three color components, hue(H), saturation (S), and value (V), of a pixel.

Linde-Buzo-Gary algorithm[30] requires an initial codebook. There are different methods like Ran-dom Codes and Splitting [32], in which the initial code book can be obtained. First , an image or a set of images is first partitioned into 4×4 blocks using the segmentation and classification methods discussed above which are represented as 4 2-tuple vectors, called training vectors[31] to generate six block classes (i.e., Low-detail, C1~C4, and Mixed classes C5). For each class of blocks, an initial vector is calculated by averaging all of the training vectors in a class to form a codevector for a subcodebook of size 1. Then the splitting technique is used to produce two vectors from the initial codevector by adding and subtracting a predefined value δ. These vectors are served as initial vectors to generate the codebook of the next level. Therefore, if the size of the current codebook is N, then, at the next level, the size of the codebook becomes 2N.

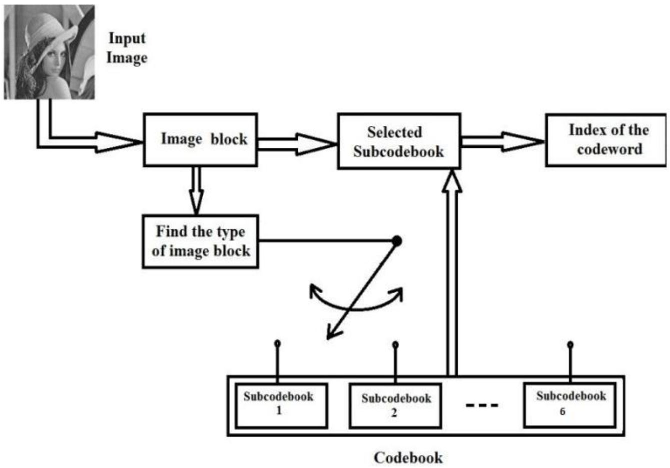


Fig 6 Block diagram of Classified Vector Quantization Scheme

The splitting procedure continues until the desired codebook size is reached and the subcodebook generation process stops when the average distortion is below a predefined threshold, and the generated codevectors are then determined as a subcodebook. When each subcodebook with its corresponding block class is generated, we combine them as the final codebook for QCVQ.

The distortion function *D* is defined as

$$D(C) = \frac{1}{N_b}\sum_{j=1}^{N_c}\sum_{i=1}^{N_b}\mu_{ij}\left\|x_i - c_j\right\|$$

Where μ_{ij} are called the index coefficients which should satisfy the following two conditions:

$$\sum_{j=1}^{N_c}\mu_{ij} = 1, \forall i \in \{1,2,\ldots,N_b\} \text{ and}$$

$$\mu_{ij} = \begin{cases} 1, & \text{if } x_i \text{ is the jth region } (R_i) \\ 0, & \text{else} \end{cases}$$

In order to design codebook optimally, one should consider the following points:

(a) The region *Rj, j=1,2,…,NC*, which represents the input vectors belonging to the codeword *cj* must satisfy

$$R_j = \{x \in X : d(x, c_j) < d(x, c_k), \forall k \neq j\}$$

Where $d(x, c_j)$ is the Euclidean distance between the vectors *x* and c_j in the HSV color space.

(b) The codeword c_j is the centroid of the jth region *(Rj)* which is computed as

$$c_j = \frac{1}{N_j} i \quad \text{for } x_i \in R_i$$

Where, N_j is the number of member vectors of R_j.

The iteration of GLA for a codebook generation is given as follows [33]:

Step 1: Randomly generate an initial codebook *CB0*.

Step 2: *i = 0.*

Step 3**:** Perform the following process for each training vector.

1)The weighted Euclidean distance in the HSV color space is calculated to find the nearest centroid vector for each training vector. Let X be a training vector and Y be a centroid vector. Then their weighted Euclidean distance is defined as follows:

$$d(X,Y) = \sum_{i=0}^{L-1}\sqrt{w_H(X_i^H - Y_i^H)^2 + w_S(X_i^S - Y_i^S)^2 + w_V(X_i^V - Y_i^V)^2}$$

2)Search the nearest codeword among *CBi.*

Step 4: Partition the codebook into *N* cells.

Step 5: Compute the centroid of each cell to obtain the new codebook *CBi+1*.

Step 6: Compute the average distortion for *CBi+1.* If it is changed by a small enough amount since the last iteration, the codebook may converge and the procedure stops. Otherwise, *i = i + 1* and go to Step 3.

After codebook design process, each codeword of the codebook is assigned a unique index value. Then in the encoding process, any arbitrary vector corresponding to a block from the image under consideration is replaced by the index of the most appropriate representative codeword. The matching is done based on the computation of minimum squared Euclidean distance between the input training vector and the codeword from the codebook. Hence an index table is produced. The codebook and the index-table is the compressed form of the input image.

D Image indexing and similarity measure

In the last stage, we perform image indexing using the QCVQ codebook i.e., each block in an image is encoded by the QCVQ codebook to generate an index. Then, we count the frequencies of indices to build an index histogram as the feature for each image in the database.

To compare the similarity of two images, we calculate the distance between their index histograms using histogram intersection (HI). The HI measure is defined as follows [4]:

$$d(Q,D) = \frac{\sum_{i=0}^{N-1} min\{H_Q(i), H_D(i)\}}{\sum_{i=0}^{N-1} H_D(i)}$$

where $H_Q(i)$ and $H_D(i)$ are the index histograms of the query image Q and the database image D and N is the dimension of the index histogram. Here, we set N =1024.

If the overlapping area of $H_Q(i)$ and $H_D(i)$ is large, the query image Q and the database image D are concluded to be similar.

IV. EXPERIMENTAL RESULTS

A Experimental metrics

For the natural-scene image database, the precision vs. recall graph (PR graph) is computed to evaluate the retrieval effectiveness. Precision and recall are defined as follows:

$$\text{precision} = \frac{\text{number of relevant images retreived}}{\text{number of retrieved images}}$$

$$\text{recall} = \frac{\text{number of relevant images retreived}}{\text{total number of relevent images in the database}}$$

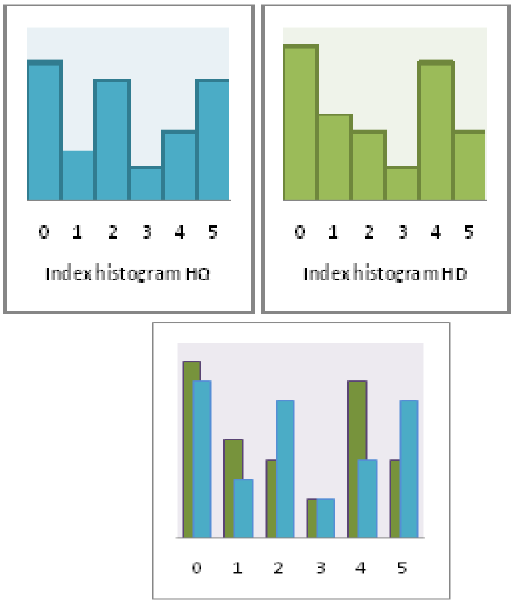


Fig.7 Histogram intersection (HI) for two index histograms.

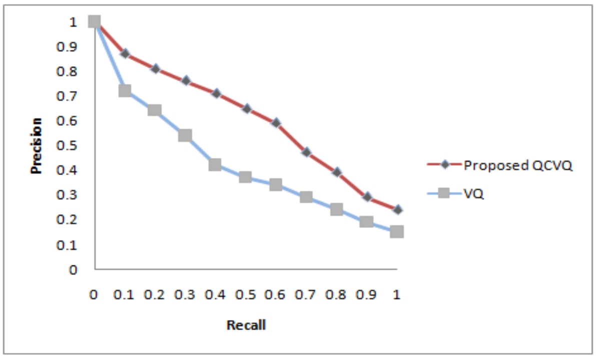


Fig. 8 The average PR graph over all of the 1,000 natural-scene database images

(a)Proposed QCVQ (7 matches out of 8)

(b)VQ (5 matches out of 8)

Fig.9 Retrieval results with the proposed QCVQ-based scheme and the VQ-based method. The query image belongs to class "pink rose". a)Proposed QCVQ (7 matches out of 8). b)VQ (5 matches out of 8)

V. Conclusions

This paper proposes the QCVQ-based image indexing and retrieval scheme, which can significantly enhance the power of indexing color images by using quadtree segmentation to extract the low-detail and the high-detail regions separately. An edge-oriented classifier, is used to further classify visually important edge intensive blocks. The simulation results for two databases show that the proposed scheme is always superior to the well-known VQ-based method.

In future work, we would add more edge templates in the design of the edge oriented classifier inorder to improve retrieval performance. We would like to explore and propose more discriminative and powerful feature descriptors, like the scale-invariant feature transform which deals with scale changes [3, 24, 26, 25], and firefly algorithm for codebook design which always occur in real-world images.

References

[1] Arnold WMS, Marcel W, Simone S, Amarnath G, Ramesh J (2000) Content-based image retrieval at the end of the early years. IEEE Trans Pattern Anal Mach Intell 22(12):1349–1380.

[2] Young R, Thomas SH (1999) Image retrieval: Current techniques, promising, directions, and open issues.J Vis Commun Image Represent 10:39–62

[3] Lu Z-C, Chang C-C (2007) Color image retrieval technique based on color features and image bitmap. Inf Process Manag 43(2):461–472

[4] Michael JS, Dana HB (1991) Color indexing. Int J Comput Vis 7(1):11–32

[5] Neetu SS, Paresh RS, Jaikaran SS (2011) Efficient CBIR using color histogram processing. Signal Image Process 2(1):94–112

[6] Salih, ND, Besar R, Abas FS (2011) Multi-level shape description technique systems. Proceedings of the 5th International Conference on IT & Multimedia at UNITEN

[7] Hamid AM, Taher TK, Amir HR, Mahdi (2005) S-T Wavelet correlogram: a new approach for image indexing and retrieval. Pattern Recogn 38:2506–2518

[8] Kokare M, Biswas PK, Chatterji BN (2007) Texture image retrieval using rotated wavelet filters. Pattern Recogn Lett 28(10):1240–1249

[9] Liapis S, Tziritas G (2004) Color and texture image retrieval using chromaticity histograms and wavelet frames. IEEE Trans Multimed 6(5):676–686

[10] Greg P, Ramin Z, Justin M (1996) Comparing images using color coherence vectors. Proc ACM Multimed 96:65–73

[11] Paschos G, Radev I, Prabakar N (2003) Image content-based retrieval using chromaticity moments. IEEE Trans Knowl Data Eng 15(5):1069–1072

[12] Jing H, Kumar SR, Mandar M, Wei-Jing Z, Ramin Z (1997) Image indexing using color correlogram. Proc Conf Comp Vision Pattern Recog 97:762–768

[13] Manjunath B, Ohm JR, Vasudevan V, Yamada A (2001) Color and texture descriptors. IEEE Trans Circuits Syst Video Technol 11(6):703–715

[14] Hrovje D, Nikola R, Dinko B, Jurica U (2000) Local thresholding classified vector quantization with memory reduction. Image Signal Process Anal

[15] Quweider MK, Salari E (1996) Efficient classification and codebook design for CVQ. IEE Proc Vision Image Signal Process 143(6):344–352

[16] Ramamurthi B, GershoA(1986) Classified vector quantization of images. IEEE TransCommun 34(11):1105–1115

[17] Guoping Q (2003) Color image indexing using BTC. IEEE Trans Image Process 12(1):93–101

[18] Ajay HD, James AS (2005) Content-based image retrieval via vector quantization. Lect Notes Comput Sci3804:502–509

[19] Idris F, Panchanathan S (1995) Storage and retrieval of compressed images. IEEE Trans Comput Electron 41(3):937–941

[20] Idris F, Panchanathan S (1997) Image and video indexing using vector quantization visual computing and communications laboratory

[21] Shyh-Wei T, Guojun L (2007) Image indexing and retrieval based on vector quantization. Pattern Recogn 40(11):3299–3316

[22] Schaefer G (2002) Compressed domain image retrieval by comparing vector quantization codebooks. Visual Commun Image Process 959–966

[23] Hrovje D, Nikola R, Dinko B, Jurica U (2000) Local thresholding classified vector quantization with memory reduction. Image Signal Process Anal

[24] Jing L, Allinson NM (2008) A comprehensive review of current local features for computer vision.Neurocomputing 71(10–12):1771–1787

[25] Yamazaki T, Fujikawa T, Katto J (2012) Improving the performance of SIFT using bilateral filter and its application to generic object recognition. IEEE Int Conf Acoust Speech Signal Process 945–948

[26] Van de Sande KEA, Gevers T, Snoek CGM (2010) Evaluating color descriptors for object and scene recognition. IEEE Trans Pattern Anal Mach Intell 32(9):1582–1596

[27] Yang SH, Yang SS (1996) New classified vector quantization with quadtree segmentation for image coding. Int Conf Signal Process 2:1051–1054

[28] Robert MG (1982) Vector quantization. IEEE Trans Inf Theory 28:157–166

[29] J. Vaisey, A. Cersho,"Image compression with variable block size segmentation", IEEE Trans. on Signal Processing, vol. 40, no. 8, pp. 2040-2060.

[30] Y. Linde, A.Buzo, and R.M Gray," An algorithm for vector quantizer design," IEEE on comm., Vol. 28, No.1, pp.84-95, 1980.

[31] Dong Liu, Xiaoyan Sun, and Feng Wu, "Inpainting with image patches for compression," Journal of Vis-ual Communication and Image Representation, Vol.23, No.1, pp.100-113, Jan. 201

[32] PierangelaCicconi, Emmanuel Reusens, FrédéricDu-faux, IoleMoccagatta, Bruno Rouchouze, TouradjE-brahimi and Murat Kunt, "New trends in image data compression," Computerized Medical Imaging and Graphics, Vol. 18, No.2, pp.107-124, April 1994.