ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 5 May, 2013 Page No. 1474-1478

CONTENT-BASED IMAGE RETRIEVAL VIA VECTOR QUANTIZATION

Tejas P. Kokate, Sanket S. Kulkarni, Virendra S. Ravalji, Deepak B. Waghchaure
Department of Computer Engineering, K. K. Wagh Institute of Engineering Education and Research, Nashik- 422 003
Maharashtra, India.

Abstract-
Image retrieval and image compression are each areas that have received considerable attention in the past. In this work, we present an approach for content-based image retrieval (CBIR) using vector quantization (VQ). Using VQ allows us to retain the image database in compressed form without any need to store additional features for image retrieval. The hope is that encoding an image with a codebook of a similar image will yield a better representation than when a codebook of a dissimilar image is used. Experiments performed on a color image database over a range of codebook sizes support this hypothesis by considering spatial relationships as well and retrieval based on this method compares well with previous work.

To effectively utilize information stored in a digital image library, effective image indexing and retrieval techniques are essential. This paper proposes an image indexing and retrieval technique based on the compressed image data using vector quantization (VQ). By harnessing the characteristics of VQ, the proposed technique is able to capture the spatial relationships of pixels when indexing the image. Experimental results illustrate the robustness of the proposed technique and also show that its retrieval performance is higher compared with existing color-based techniques. One of the key roles of Vector Quantization (VQ) is to compress the data about the image database. In the past years, many improved algorithms of VQ codebook generation approaches have been developed. In this paper, we present a snapshot of the recent developed schemes. The discussed schemes include LBG, enhance LBG (ELBG).

***Keywords*: CBIR, Vector Quantization (VQ), LBG, ELBG, Image Indexing, Image Retrieval**

I. INTRODUCTION

Due to the need to effectively access the huge quantity of different type of images stored in digital image libraries image retrieval is a growing field of research which has evolved. Former image retrieval systems used the conventional database management approach to index and retrieve images, based on simple attributes such as the image number, image texture and text description [2]. Systems similar to this have their own limitations. Firstly, limited number of query types. These systems cannot take Content Based Image as input. Secondly, because simple attributes can not describe image completely and accurately which results into low retrieval performance. To overcome these limitations,

content-based image retrieval (CBIR) techniques have been actively pursued. CBIR techniques use low-level image features, such as color, shape and texture for indexing and retrieval [2–3]. To date, color-based image retrieval techniques are still popular and are commonly implemented in many CBIR applications [3]. It is popular because of two reasons. Firstly, compared to shape and texture, it is normally much easier to remember the color elements in the images. Secondly, unlike the shape and the texture based techniques, color-based techniques do not assume images have been correctly segmented into individual objects.

A main problem with image databases and digital libraries is that of Content-Based Image Retrieval (CBIR), to which image is given as input are very large in size. With the ever increasing amounts of machine generated imagery (digital cameras, scientific and medical instruments, remote sensors, etc.), the need has never been greater for efficient and effective systems to manage visual data and understand the content within these media. The need to "google" imagery will become increasingly important in the future. Here we are using query-by-example model [1],where the user give image as input and system returns the similar matching images as output. In much of the past work on CBIR, images are first represented by

feature vectors which capture certain contents of the image. Perhaps the most important discriminating feature for image classification and retrieval is color [1, 2]. Intensities of pixels can be accurately represent the color distribution in image. A typical color image, however, is capable of representing 224 different colors and building a histogram of this size is infeasible. The color space is thus first quantized into a fixed number of colors (bins) and images are compared by comparing their color histograms using some vector norm of the difference between the histogram vectors. Examples of schemes for image retrieval based on color historgrams are and an analysis of color histograms can be found in [1]. Since global color based retrieval considers only single pixels, it don't consider the spatial correlation between pixels. To consider the spatial properties of images based on the histogram approach, color coherence vectors (CCVs) along with geometric histograms have been proposed. CCVs measure the spatial coherence of the pixel for a given color, i.e. a large coherent region of a color corresponds to a high coherence value, where as the geometric histograms consider the distribution of certain fixed geometric image blocks instead of single pixels. VQ based image classification and retrieval has been proposed in the past. Idris and Panchanathan [4] use VQ to index compressed images and video data. Initially images are compressed using global codebook, then for each code vector in the codebook, a histogram of image block is generated. Thus a linear combination of the code vector histograms weighted by the frequency of occurrence of a code vector approximates the per pixel (color) histogram of the entire image. Their results compare favorably to methods based on color histograms. They also propose another method that compares images based on which of the code vectors is used in an image. Even in this case, the reduced complexity similarity evaluation performs as well as color histograms [5]. Lu and Teng [6] also employ a universal codebook to represent each image and use the histogram intersection (HI) between code vector usage histograms to compute similarity. The VQ based methods described so far make use of a universal codebook where if images with different statistical properties (classes) were to be later added to the database, then the codebook may need to be updated to maintain retrieval accuracy. One way to solve this is to design separate codebooks for each image. In this work as well, we make use of the VQ codebooks, but instead comparing codebooks, we evaluate the similarity between a query image and a database image by evaluating the encoding distortion when the query image is compressed with the database image codebook. This idea is similar to that of maximum likelihood classification where data is classified as belonging to one of several class models (codebooks) depending upon which model results in the maximum probability (minimum distortion).Section 2 briefly describes vector quantization, Section 3 explains our method for image similarity using VQ along with the encoding and decoding. Section 4 concludes.

II. VECTOR QUANTIZATION

Vector Quantization (VQ) is an efficient and simple approach for data compression. Since it is simple and easy to implement, VQ has been widely used in different applications, such as pattern recognition, image compression, speech recognition, face detection and so on [3].

For the purpose of image compression, the operations of VQ include dividing an image into several vectors (or blocks) and each vector is mapped to the codewords of a codebook to find its reproduction vector [3]. In other words, the objective of VQ is the representation of vectors $X \subseteq R^k$ by a set of reference vectors $CB = \{C1, C2 \ldots\ldots, CN\}$ in R^k in which R^k is the k-dimension Euclidean space. CB is a codebook which has a set of reproduction codewords and $Cj = \{c1, c2, \ldots.., ck\}$ is the j-th codeword. The total number of codewords in CB is N and the number of dimensions of each codeword is k.

There are three major procedures in VQ, namely codebook generation, encoding procedure and decoding procedure. In the codebook generation process, various images configurations over the entire image. In this work, spatial information is incorporated implicitly when we perform quantization on

are divided into several k-dimension training vectors. The representative codebook is generated from these training vectors by the clustering techniques. In the encoding procedure, an original image is divided into several k-dimension vectors and each vector is encoded by the index of codeword by a table look-up method. The encoded results are called an index table. During the decoding procedure, the receiver uses the same codebook to translate the index back to its corresponding codeword for reconstructing the image[3].

Fig. 1 shows an example of encoding an image by VQ. In Fig. 1, an original image is divided into four blocks sized 2 × 2. Each block is then translated to a vector with four dimensions. The first vector $X_1 = (7, 10, 14, 6)$ is mapped to the 7^{th} codeword of the codebook. The index of the 7^{th} codeword replaces the vector to represent the block. When the receiver receives the index table, he uses the 7^{th} codeword $c_7 = (9, 6, 9, 9)$ to reconstruct the first block. One of the key points of VQ is to generate a good codebook such that the distortion between the original image and the reconstructed image is the minimum [3]. However in our proposed system, we are not getting into the overhead of reconstructing the image, instead, we can directly use the index table as in Fig. 1 to retrieve the image from the database. This eliminates the distortion problem and increases the accuracy of the search results. Moreover, since the codebook generation procedure is a time consuming process, how to reduce the computation time is another important issue for the VQ codebook generation.

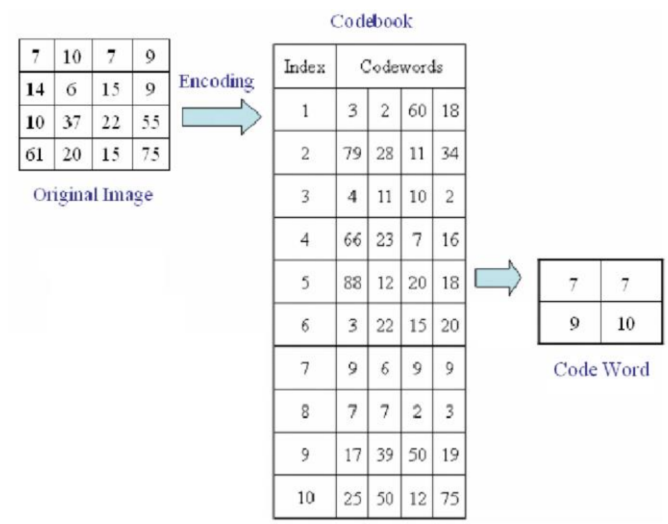

Fig.1 VQ Encoding Process

VQ Codebook Generation Process

Vector Quantization is an efficient and technique used for Data Compression, and is widely used in many different applications like pattern recognition, image compression, speech recognition and so on. The process of Vector Quantization involves three major processes namely Codebook Generation, Encoding process and Decoding process. Above mentioned three processes are described in detail.

Codebook Generation:

A vector quantizer can be defined as a mapping Q of k-dimensional Euclidean space R^K into a finite subset Y of R^K, that is

$$Q: R^k \rightarrow Y$$

Where Y=$(x'_i = 1,2,\ldots..N)$, *and* x'_I *is the* i^{th} vector in Y. Y is also called as codebook.

Codebook generation process in image compression is done using Linde-Buzo-Gray Algorithm as in [2,3]. For image compression purpose, image is divided into vectors, which are mapped to the codewords in the codebook. Mapping of the codewords is based on Euclidean distances. Index of the codewords is used instead of using whole blocks. Thus for eg. if an image is divided into say 64 blocks, each block of size say 8*8, then the compressed image is represented using just 64 values instead for all pixel values.

For efficient image indexing and retrieval we are using the LBG algorithm with some modification for ease of implementation as follows:

1) Firslty we select the training images suitably so that the codebook generation process if effective in generating a codebook which can represent the test database with adequate accuracy.
2) Training images are divided into training vectors of size $n \times n$ pixels.
3) Training vectors are grouped into clusters based on Euclidean distances. Euclidean distance is calculated using following formula:

$$d(X,Y) = \sum_{i=0}^{H-1} \sqrt{(X_i^R - Y_i^R)^2 + (X_i^G - Y_i^G)^2 + (X_i^B - Y_i^B)^2},$$

4) Training vectors having mutual Euclidean distances less than a certain threshold are grouped into same cluster. For eg. in fig.2 below blocks(vectors) having distances less than 10 are grouped together.

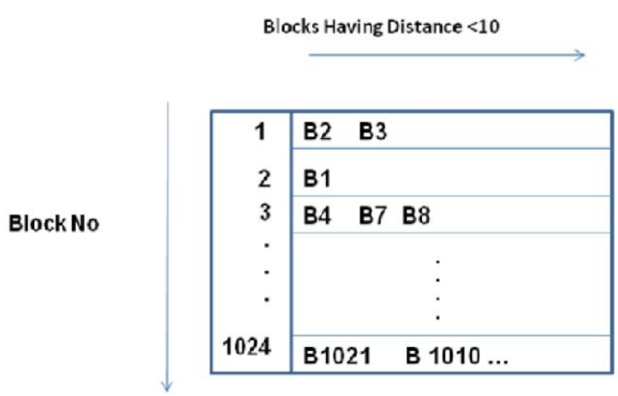

Fig.2(a) Block Clustering Process

5) Average of all training vectors is calculated to form centroid vectors for those clusters which are actually forming the codewords in the codebook for the current iteration.

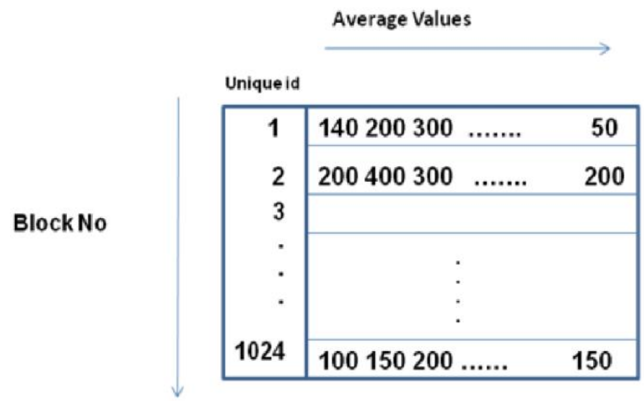

Fig.2(b) Calculating Centroids for Clusters

6) Further, calculating the mutual Euclidean distances between the above generated codewords can reduce the no. of codewords and carrying on further iteration produces an optimized codebook.

The codebook thus formed can be used for efficient retrieval but retrieval accuracy depends upon the threshold value set during codebook generation.

III. VQ BASED IMAGE SIMILARITY

This section describes our approach for Image Similarity using Vector Quantization. It can be seen that because we quantize image blocks rather than single pixels, VQ captures some amount of interpixel (intrablock) spatial correlation along with some textural attributes. The VQ codebooks along with the usage frequencies of code vectors form a generative model for the image and code vectors represent the principle regions or modes in the images. Thus we expect that encoding an image with a codebook of a similar image will yield a better representation than when a codebook of a dissimilar image is used.[10] Image similarity is assessed by matching the Histogram of Query image with the histogram of the images present in the database.

- **Image indexing using code vectors and retrieval based on VQ compressed data**

After VQ compression, each block of pixels is represented by a code vector index number. We can extract image features and carry out image indexing and retrieval based on these index numbers. For a given image, we can calculate the number of occurrence of each index to obtain an index histogram $H(v1, v2, \ldots, vi, \ldots, vn)$, where vi is the number of times code vector i is used by the image, and n is the total number of code vectors in the codebook. During image retrieval, an index histogram is calculated for the query image. Let us call the histogram of it as $H(q1, q2, \ldots, qi, \ldots, qn)$. Then the distance between the query image Q and an image V can be calculated as follows:

$$d(Q, V) = \sum_{i=1}^{n} |H(q_i) - H(v_i)|.$$

Images can be ranked in an ascending order of calculated distances. The larger the calculated distance between the two images, the more different the two images. Finally, the list of top N target images with the smallest distance differences relative to the query image will be retrieved. The list of retrieval images is also ranked from the smallest to the largest, according to their distances. The above image indexing and retrieval process is very similar to the basic color-based image retrieval technique described in Refs. [5,6,10]. The main difference is that in VQ-based technique, the histogram represents the numbers of image vectors with the same code vector index while the histogram in the basic color-based technique represents the numbers of pixels with the same color.

In comparison, image retrieval based on VQ compressed data has the following advantages over the basic color histogram based technique:

1. During the indexing process, no decompression is required.

2. The number of blocks is significantly less than that of pixels, leading to more efficient indexing and retrieval.

3. The VQ-based technique should provide higher image retrieval effectiveness. This is because the similarity is based on blocks of pixels instead of individual pixels. Thus the VQ-based technique can overcome the problem in cases where two blocks of pixels have the same or similar histograms based on the basic color-based technique, but overall appearance of these two blocks are totally different.

For example, two 1-dimensional blocks [1, 255, 1, 255, 1, 255] and [1, 1, 1, 255, 255, 255] have the same histogram in the basic color-based technique, leading to the wrong conclusion that they are the same. But in VQ-based technique, these two blocks will be matched to the different code vectors and placed in different bins of the histogram, leading to proper distinction between these two blocks.

- **Choice of training images**

The selection of training images for codebook generation for the purpose of image indexing and retrieval is similar to that for compression. Thus, the database administrator needs to group similar images into similar categories and pick one or more images from each category to make up the training image set. Another way of choosing the training image set is to randomly pick a suitable percentage of the database images and use them as the training images. It is important to note that the set of training images should be representative of the entire set of database images because only one codebook is generated for that database. The purpose of having only one codebook for a database is to facilitate the comparison of image histograms during retrieval.

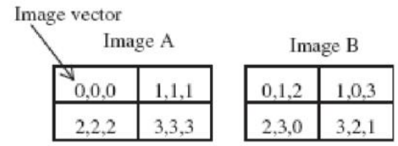

Fig. 3 Two perceptually different images having same histograms. The three numbers in each image vector represent the codeword indices of the red-channel codebook, green-channel codebook and blue-channel codebook, respectively.

- **Codebook structure suitable for image retrieval**

In the general VQ image compression, the main aim is to achieve highest compression ratio while maintaining acceptable reconstructed image quality. This means the number of codewords generated must be kept at the minimum while they still capture all the main features of the images to be compressed. This is achieved traditionally by generating a codebook separately for each of the red, green and blue color channels using the training images (assuming *RGB* color space is used). Each codebook consists of the codewords which best represent that particular color channel. To compress an image, the image vectors in their red, green and blue color channels are separately compared with their respective codebooks to find their best match codewords. In this way, if each codebook size is 256, a total of 224 different image blocks can be represented by the combination of the three codebooks. If each codeword is of 16 pixels, the total number of bytes to store the three codebooks is 12 288 bytes. Though this scheme allows high compression ratio to be achieved, it is not suitable for image indexing and retrieval. If three codebooks are used to index an image, one histogram is built for each color channel. Thus, three histograms are needed to index an image. This causes problems in comparing similarity between images. For image retrieval based on

VQ to be successful, two conditions must be true:

1. Perceptually similar images must have similar histograms
2. Images with similar histograms should reflect that the images are more likely to be perceptually similar.

However, when the codebook is generated using the traditional technique, these conditions may not be met. Here, although perceptually similar images have similar histograms, similar histograms may not reflect that the images are perceptually similar. An example of such images is shown in Fig. 3. Images A and B in Fig. 3 are each made up of four image vectors. If the traditional codebook generation technique is used, three codebooks will be generated. If the color space used is *RGB*, a codebook is generated for each of the red channel, green channel and blue channel of images. To encode images A and B using our proposed technique, each color channel of the images will be encoded using the codebook of the corresponding color channel. As a result, each image vector is represented by three indices, where each index represents the best match codeword in their respective color-channel codebook. For an example, as shown in Fig. 3, the top-left image vector of Image B is represented by codeword index 0 of the red-channel codebook, codeword index 1 of the green-channel codebook and codeword index 2 of the blue-channel codebook. Since each image vector is encoded by three indices of three different codebooks, three histograms are built to index each image. In Fig. 3, only one image vector in image A is encoded by codeword index 0 of the red-channel codebook. The same number of image vector in Image A is also encoded codeword index 1, 2 and 3, respectively. Therefore, the red-channel histogram of Image A isH_R(1,1,1,1). The green and blue channel histograms of Image A are H_G(1,1,1,1) and H_B(1,1,1,1), respectively. As for Image B, its red, green and blue channel histograms are H_R(1,1,1,1), H_G(1,1,1,1) and H_B(1,1,1,1), respectively. By comparing their histograms, we would think that the two images look alike.

However, they look perceptually completely different. This is because the occurrences in the bins of the histograms may be contributed from different image blocks of the image. As the color channels in the two images are combined together differently, they are different perceptually. To facilitate the similarity comparison between images, we propose to generate one codebook instead of three codebooks of the traditional technique. This means each codeword represents a block of image pixels which can be found perceptually from the training images. Each codeword needs 24-bits to represent, instead of 8-bits. Only one histogram is used to index an image in this way. Thus each bin in the histogram counts the occurrences of each codeword. The distance between the histograms of two images shows the total number of different physical image blocks of pixels (codewords) between the images. Perceptual similarity between the images is reflected in similarity of histograms. However, in this proposed technique, to achieve similar image quality as the traditional technique, more codebook storage space will be needed. For example, to achieve image quality of the traditional technique in above example, 224 codewords must be generated and the storage space needed is 49 152 Kbytes, instead of the 12 288 bytes of the traditional technique. That is 4096 times greater. So, since having a codebook that needs such a huge amount of storage space is not feasible, we propose to use a codebook size which is enough to capture the main features of the database images. This is feasible because for image retrieval, we are retrieving relevant images from the database, thus an exact match in the features of the retrieved images to the query image is not needed. As long as the database images have similar features to the query image, they should be retrieved.

IV. Conclusion

In this paper, we have presented a CBIR technique which harnesses the characteristics of VQ to capture the spatial information of the images during image indexing. We have presented the experiments carried out to investigate the robustness of our proposed technique and how it measure-up in terms of retrieval effectiveness to three other existing color-based image retrieval techniques.

A simple image retrieval method based on VQ compression has been introduced. By considering VQ codebooks as image descriptors, a query can be assigned to class whose codebook best encodes the query. The similarity measure has been shown to be robust with respect to codebook sizes and compares favorably with current VQ based systems.

Acknowledgment

The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] “Content-Based Image Retrieval via Vector Quantization” Ajay H. Daptardar and James A. Storer Computer Science Department, Brandeis University, Waltham, MA 02454, USA

[2] “Image indexing and retrieval based on vector quantization” ShyhWei Teng∗, Guojun Lu *Gippsland School of Information Technology, Monash University, Gippsland Campus, Churchill, Vic. 3842, Australia*

[3] “A Survey of VQ Codebook Generation” Tzu-Chuen Lu Department of Information Management Chaoyang University of Technology Taichung,Taiwan, Ching-Yun Chang Computer Laboratory University of Cambridge Trinity Lane, Cambridge CB2 1TN,UK Received March 2010; revised April 2010

[4] K Idris, F., Panchanathan, S.: Image and Video Indexing using Vector Quantization. Machine Vision and Applications 10 (1997) 43–50

[5] Idris, F., Panchanathan, S.: Storage and Retrieval of Compressed Images. IEEE Transactions on Consumer Electronics 43 (1995) 937–941

[6] Lu, G., Teng, S.: A Novel Image Retrieval Technique based on Vector Quantization. In: Proceedings of International Conference on Computational Intelligence for Modeling, Control and Automation. (1999) 36–41.

[7] A. Gersho, R.M. Gray, Vector Quantization and Signal Compression,Kluwer Academic Publishers, Dordrecht, 1992.

[8] H. Abut (Ed.), Vector Quantization, IEEE Press, New York, 1990.

[9] K. Sayood, Introduction to Data Compression, second ed.,Morgan Kaufmann Publishers Inc., San Francisco, CA, 2000.

[10] Schaefer, G.: Compressed Domain Image Retrieval by Comparing Vector Quantization Codebooks. In: Proceedings ofthe SPIE Visual Communications and Image Processing. Volume 4671. (2002) 959–966.

[11] W. Niblack, et al., QBIC Project: querying images by content, using color, texture, and shape, Proceedings of Conference on Storage and Retrieval for Image and Video Databases, 1–3 February 1993, San Jose, CA, US, SPIE, vol. 1908, 1993, pp. 1908–1920.

[12] M.J. Swain, D.H. Ballard, Color indexing, Int. J. Comput. Vision 7 (1991) 11–32.

[13] G. Lu, Image retrieval based on color, Proceedings of the Conference on Storage and Retrieval for Image and Video Databases IV, San Jose, CA, 1–2 February 1996, SPIE Proceedings Series, vol. 2670.