Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2016) 24: 4294 – 4304

doi:10.3906/elk-1502-164

Fast image search on a VQ compressed image database

Mehmet YAKUT*
Department of Electronics and Communication Engineering, Kocaeli University, İzmit, Turkey

Received: 19.02.2015 • **Accepted/Published Online:** 31.07.2015 • **Final Version:** 20.06.2016

Abstract: A fast and efficient image search method is developed for a compressed image database using vector quantization (VQ). An image search on an image database requires an exhaustive sequential scan of all the images, given the similarity measure. If compressed images are dealt with, images are decompressed as an initial operation and then the previously mentioned exhaustive search is performed using the predetermined similarity measure. If the images in the database are compressed using VQ, the image search process is reduced to codebook index match tests. A pixel by pixel similarity test of two images computationally costs too much. This bottleneck is overcome by using VQ, where the similarity test of the two image block is performed by a precalculated distortion lookup table. The same is valid for the object search in the image database. The object image is vector quantized first; then the index map of the object image is scanned over the entire index area of the compressed image database. Significant image search speed gains on the VQ image database are obtained. Results show that the VQ compressed image search is faster than a sequential search, and compressed and decompressed JPEG search. Actual speed gain obtained here depends on the application area and required image quality for the database.

Key words: Image search, image compression, vector quantization, compressed recognition

1. Introduction

The aim of multimedia processing is to deal with investigation of image, audio, and video data during their transmission or storage. Typical processing applications are performed on surveillance, teleconferencing, and database images. In this study, vector quantization (VQ) compressed video sequences and still images are considered as the target application. Compression has a very important role since it is required to compress streaming data as fast as possible in an efficient way. Our application has a role in the area of multimedia processing in the compressed domain.

Day by day ascending multimedia data use forces us to compress and process data efficiently. Frequently Internet users need to browse and process image and video data effectively [1]. Obviously, implementing automatic image retrieval solutions is of great potential interest for most users [2]. Feature extraction and content based classification approaches such as curvature scale space (CSS), centroid distance functions (CDF), and null space invariant (NSI) representations can be considered for automated image database retrieval [3]. Both approaches can be used for image query and similarity measurement of test images [1,4]. Compression algorithms transform images into different domains in which image data could be represented by reduced numbers of bits [4].

Content analysis of image data can be performed on either original or compressed images. However,

*Correspondence: myakut@kou.edu.tr

it is computationally advantageous to process image data while they are in compressed state. Sometimes it is possible to analyze image data during the inner steps of the compression scheme, since the data have a better form to investigate. In general, the analysis process takes place at the end of the compression stage due to its advantages [4]. In this way we deal with reduced numbers of bits and uncorrelated components of the compressed images. Processing uncorrelated components of the compressed images enables us to avoid repeated unnecessary steps. Another advantage of processing data in compressed state is eliminating the need for decompression processing, especially when the decompression of image data does not provide additional information [3]. The information about the compressed image remains the same in compressed or decompressed state either in the lossy or lossless compression case. The amount of information in the compressed case cannot be increased by decompressing the image; the only difference is that the decompressed version of the image makes sense for a human viewer. In a scenario of compressed image analysis, a computer analyzes the image database and passes the obtained information to another computer in a network. If the analyzing computer only could perform its operations (i.e. object classification) on the pixel domain, it is obvious that decompression of the inspected image is inevitable. Cost of the analysis process in the pixel domain and compressed domain may be the same, but the compression and decompression process can reduce system resources drastically [5]. Additionally, compression into a specific domain may inherently introduce the properties that we search for in the image [6]. As an example, general transformation based compression algorithms inherently hold frequency information, which could be very useful for a feature extraction algorithm [7].

- Application areas for the fast image search presented in this paper:

  - fast searching for a suspect image in a criminal database,

    Database images are compressed using VQ and kept in server computer, the search image (or crop of image block) is vector quantized, the index map of the search image is sent over a slow communication line immediately, the index map is searched over the entire database, and an index list of similar images found in the database is sent back to the searching client. At the client point candidate VQ compressed images are retrieved using the index list and codebook, and then visually can be interpreted by human operators.

  - Fast search of a special object or symbol carrying person (man wearing eyeglasses etc.) over continuously compressed and recorded (security camera) image data (video or still images).

Efficient use of recording area and fast access to the searched object frame position without operator intervention during playback.

2. VQ compressed image analysis

Rate distortion theory states that VQ is a much more powerful technique for data compression compared to scalar quantization [8]. Vector coding always performs better than scalar coding whether the data source has memory or not [9]. A vector quantizer provides arbitrarily shaped quantization regions in higher dimension. VQ methods have potential of redundancy extraction over linear dependencies by optimal placement of code vectors in the N dimensional space [10]. VQ is an attractive coding technique since the rate distortion bound can be approached as the vector dimension is increased [10,11]. However, the cost of optimized codebook generation increases exponentially with the vector dimension [12]. Most VQ applications are restricted to small vector sizes because of these facts [13].

Multiscale image representation and progressive image transmission is a very important subject when it is required for featured images to scan or a fast image search in a large image database and also when accessing a large criminal image database for recognition of the suspect from a telephone line etc. Database images are kept in different resolution scales and the initial search is performed on coarse quantized lower resolution images [13]. VQ of the lower scale and higher scale representation of the large image database will highly reduce the required transmission bit rate [14].

When long duration image sequences or a large image database are needed to be scanned for finding of an object or a person, it is a very time consuming process to achieve our goal. For example, if we are working on security camera records, we should watch the whole records using fast playback mode. In the remote access case, even the transmission time of the search image could be unbearably long. As a result of investigation of the whole image database, transfer time of the candidate images also may be intolerable. When the number of images is high or the image sequence is very long, we may ignore the search image transmission duration with respect to the database search period. The disadvantages can be altered using the images in the compressed form; however, compressed images are generally needed to be decompressed in order to make similarity tests between the search image and database images [6]. Tests can be done on compressed images with special similarity metrics defined by using the characteristics of the compression technique. If images are compressed using VQ, as a special case of compression, selected target images, patterns, or some special objects can be searched directly over compressed images without using special similarity metrics. VQ of the images with enough visual quality will speed up the image search on the compressed database and determination of the candidate images [15]. In VQ applications arbitrary block shapes can be used depending on the local image properties [15,16]. Block shapes can be defined for the search image or search object properties, and then arbitrary shaped blocks at the object boundaries are related with square code vectors (i.e. combinations of the arbitrary shaped code vectors correspond to some square code vector combinations) [14].

For unstructured VQ applications, the global codebook is trained to represent the whole characteristics of the input images [13]. Most of the time local characteristics are ignored. However, in multiresolution representation global characteristics of the image are contained in the low resolution part and details are contained in the high resolution parts [17]. The low resolution part is the low past filtered and subsampled part of the original image. Since global characteristics are searched on the lower resolution image sequence, time needed for searching is reduced at least by a factor of four assuming half resolution [14].

3. Vector quantization

The vector quantizer finds best representing code vector in the codebook for the vector to be encoded given the distortion measure and sends the index of the code vector to the receiver [9] (Figure 1). The first step in VQ design is to select the initial codebook. Initial values of code vectors can be selected randomly, by pruning, by the nearest neighbor rule, or by splitting [8,9]. The training method for the codebook is known as the generalized Lloyd algorithm (GLA). Given the training set images and distortion measure, the initial codebook is trained by GLA iteration, until training does not increase the quantization performance.

In most VQ applications, squared error $d(x,\hat{x})$ and expected value of $d(x,\hat{x})$ are used as distortion measures.

$$d(x,\hat{x}) = \|x-\hat{x}\|^2 = \sum_{l=0}^{k-1} (x_l - \hat{x}_l)^2 \tag{1}$$

$$D = E\left[d\left(x, \hat{x}\right)\right] = E\left[\left\|x - \hat{x}\right\|^2\right] \tag{2}$$

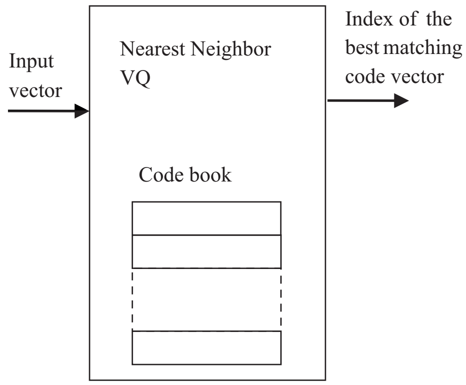

Figure 1. Basic vector quantizer block diagram.

4. Image search throughout image sequence and/or image database

An image search can be performed using different approaches; for example, we may use direct comparisons or we may transform images into another domain in which we can search efficiently. The distance metric provides a measure for testing image similarity in the feature space. L1, L2, and Euclidean metrics are widely used to estimate the similarity of images in different search systems [9].

The general disadvantages of these approaches are complexity and exhaustive calculations. Image feature extraction and sequential image search are challenges for database designers since they are time consuming and computationally hard. Identifying features and faster search techniques can make scanning an image throughout a large set of images easier. On the other hand, researchers are motivated to develop better data compression applications driven by wide use of multimedia and geometrically increasing video, image, and sound records [4]. Another issue is to support progressive transmission of an image in which the image is delivered first at the iconic level and then at the user request, at gradually increased size and resolution [2,6]. Wavelet-based multiresolution compression makes hierarchical on demand accessing image data, which avoids unnecessary data transfer [18].

Compressed data contain the transformed content of the original images but do not provide information for visual inspection. In the compressed search technique, the main purpose is to extract comparable meaningful values from transformed data even though the visual information is hidden [4].

The idea of processing compressed data without decompressing is studied for real-time visual effects like fading and warping. The basic approach for applying image operators to JPEG-compressed images is outlined in [19]. It is shown that large speed gains are obtained for certain operations by avoiding decompression. Digital medical stored data are lossless compressed using Haar wavelet transformation because medical images cannot afford any loss [15]. In this application edge and texture information is used for retrieving medical compressed data for fast access to the patient's diagnostic information securely.

If we use vector quantization for compression and storing the images, some kinds of image searching processes can be performed directly on compressed images even requiring only an addition operator for comparison of two images [14]. Different VQ methods can be adapted to different illumination and contrast conditions. Mean removed and gain/shape VQ can be mentioned here for such cases. Minimum distortion image retrieval based on histogram matching with a training data fitted Gaussian mixture model is used in [4]. In this application original images (256 $\times$ 256) are subsampled (32 $\times$ 32) where images are visually analyzable in the fast decision algorithm. The JPEG2000 based compressed image database search technique is focused on the matching performance by using similar compression ratios in the search image and database [20]. Better matching performance on the wavelet compressed images is achieved by using the same compression ratios.

When an image is vector quantized, compressed image data consist of codebook indexes and the restored image is composed of a code vector array representing the original image with an acceptable image quality loss. This limitation becomes an advantage when we need to compare two compressed or vector quantized images. Similarities of the code vectors or distortion between code vectors can be calculated only once and then placed in a lookup table.

If we use d(x, y) notation for the similarity measure between two comparison images, i.e. X and Y images, d(x, y) can be calculated using Eq. (3), known as Euclidian distance. One can say that two images are similar if the distance between them is small enough given the threshold level for similarity. Calculative intensive comparisons can be executed as shown in Figure 2, which exhaustively consume resources.

$$d(x,y) = \sum_{i=0}^{N}\sum_{j=0}^{M}[x(i,j) - y(i,j)]^{2} \tag{3}$$

If the distance between two images or parts of two images is zero, obviously one can say they are the same. Weighed vector distance criteria can be used to compare images, i.e. regional similarities that we are interested in can be exaggerated [11]. If the distance is less than the predetermined threshold level, that image can be classified as a candidate image for visual inspection.

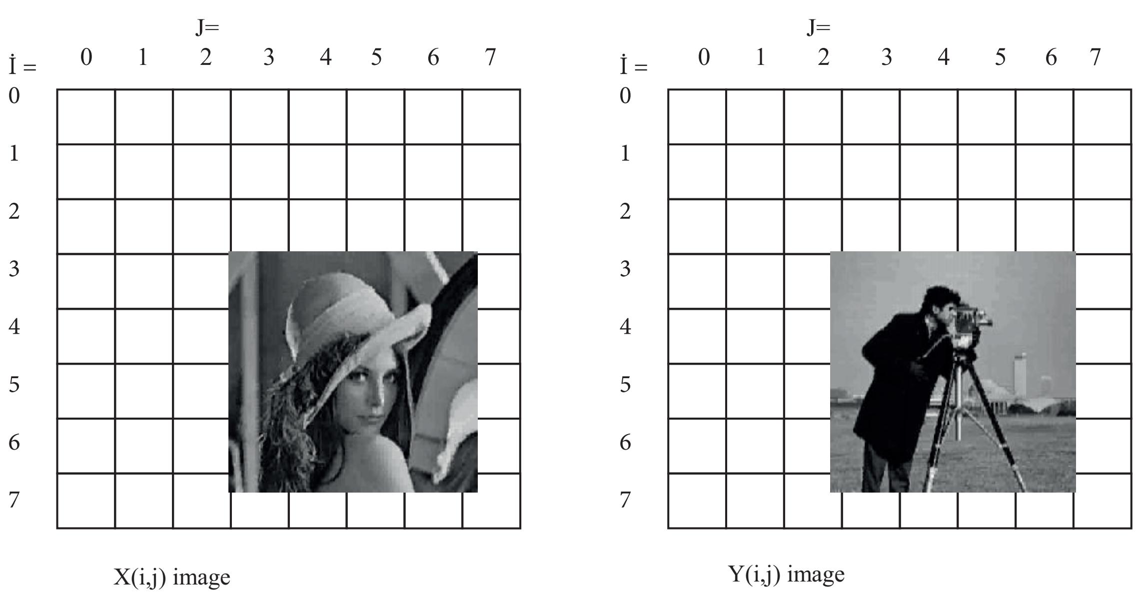

Figure 2. 8 $\times$ 8 blocks from X and Y images to be compared.

Similarity between vector quantized images can be calculated using Eq. (4).

$$\begin{aligned} d(\hat{x},\hat{y}) &= d(Q(x),Q(y)) \\ &= \sum_{i=0}^{N}\sum_{j=0}^{M}\left[\hat{x}(i,j)-\hat{y}(i,j)\right]^2 \\ &= \sum_{l=0}^{L} d(CBx_l[u],CBy_l[v]) \end{aligned} \tag{4}$$

$d(\hat{x},\hat{y})$ is used for quantized distance, $CBx_l[u]$ and $CBy_l[v]$ denote the code vectors used for quantizing image x and y, respectively, and u and v denote codebook indexes for the code vectors, which correspond to x and y vectors, respectively.

Comparison of two images is shown in Figures 2 and 3 by means of Eq. (4). If corresponding vectors of the images are quantized with the same code vectors, then we decide on the correct similarity of two vectors, although the two vectors are not the same. The key of our idea is to use precomputing of the distances between code vectors once and then use them for comparison of two vectors. Simplicity of distortion calculation is available now, only using a lookup distance table between code vectors by means of code vector indexes as shown in Table 1. In our case, exhaustive N $\times$ N subtraction, multiplication, and addition are reduced to only one subtraction for comparison of two vectors. The code vector distance lookup table is calculated offline and once, which results in enormous speed up in the search process and calculation costs go down. Using this approach, we can search even partially occluded objects or image parts.

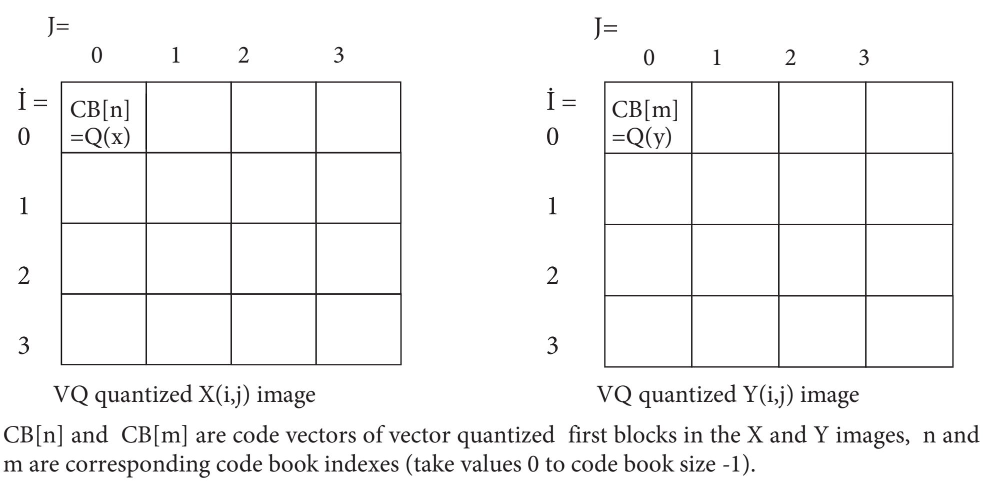

Figure 3. Index map of X and Y images to be compared (compressed X and Y images).

Instead of comparing blocks exhaustively (Figure 4 left side), the codebook distortion lookup table is used for fast comparison (Figure 4 right side). Given the database image codebook index and sample image codebook index we just look up the distortion in Table 2 and add to the total distortion. Then if the total distortion of the whole search image is below the threshold level, we decide that block in the image is similar to the search object or partial image. The decision can be made on the basis of total distortion, number of exact matching blocks, or number of matching blocks below the threshold level. One can determine the occluded blocks or partial match of blocks on the basis of matching neighboring blocks. If some neighboring blocks are exactly or below threshold matched, then we can say that the partial search object is detected. Threshold level can be selected depending on image size, average distortion per vector, or exact distortion level per code vector, considering the average distortion level of VQ compressed images. In this work, almost half of the neighboring matching vectors of the search image have greater distortion than the average distortion level of VQ and second and third best code vector quantization distortion values are used as a similarity measure and decision rule for image match.

Table 1. Distortion lookup table between code vectors (d(CBx_i, CBxj) shows distortion between code vector i. and j.).

Search indexes	di(x, y)	D	#of matches
0	9+0+0+0	9	1
1	0+0+7+7	14	2
2	0+0+0+0	0	4
3	14+11+0+0	25	2
4	11+14+7+0	32	3
5	14+11+7+16	48	0
6	14+11+14+7	46	0
7	11+11+0+11	33	1
8	11+9+14+16	50	0

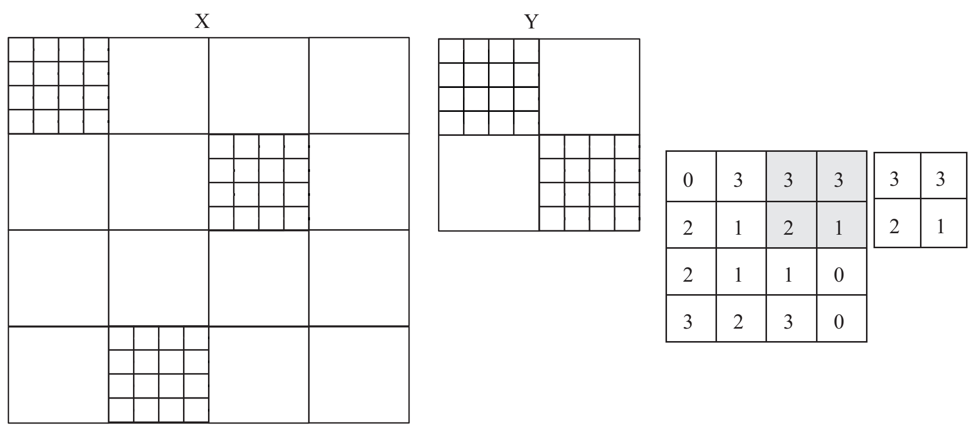

Figure 4. Sample database (X) and search (Y) images (left) and their respective vector quantized index maps (right).

Table 2. Example codebook distortion lookup table and similarity search results of Figure 4.

Codebook indexes	1	2	3
0	16	18	9
1	0	7	11
2	18	0	14
3	9	11	0

In Figures 5 and 6, sample search images and their respective vector quantized codebook index maps are shown. In Figure 7 database images are shown in which the search process is performed. Some test results are shown in Table 3 to give insight into performance level by the detail of search image sizes and matching number of vectors w.r.t. threshold level. As can be seen from Table 3, small images are difficult to detect and require a fine decision rule. If images selected from the database were exactly placed on the vector positions, there would be no mismatch in fact. However, sample search blocks are selected 1, 2, and 3 pixels shifted in both horizontal and vertical directions from the vector position of the original image.

Figure 5. Sample search images.

Figure 6. VQ compressed search images.

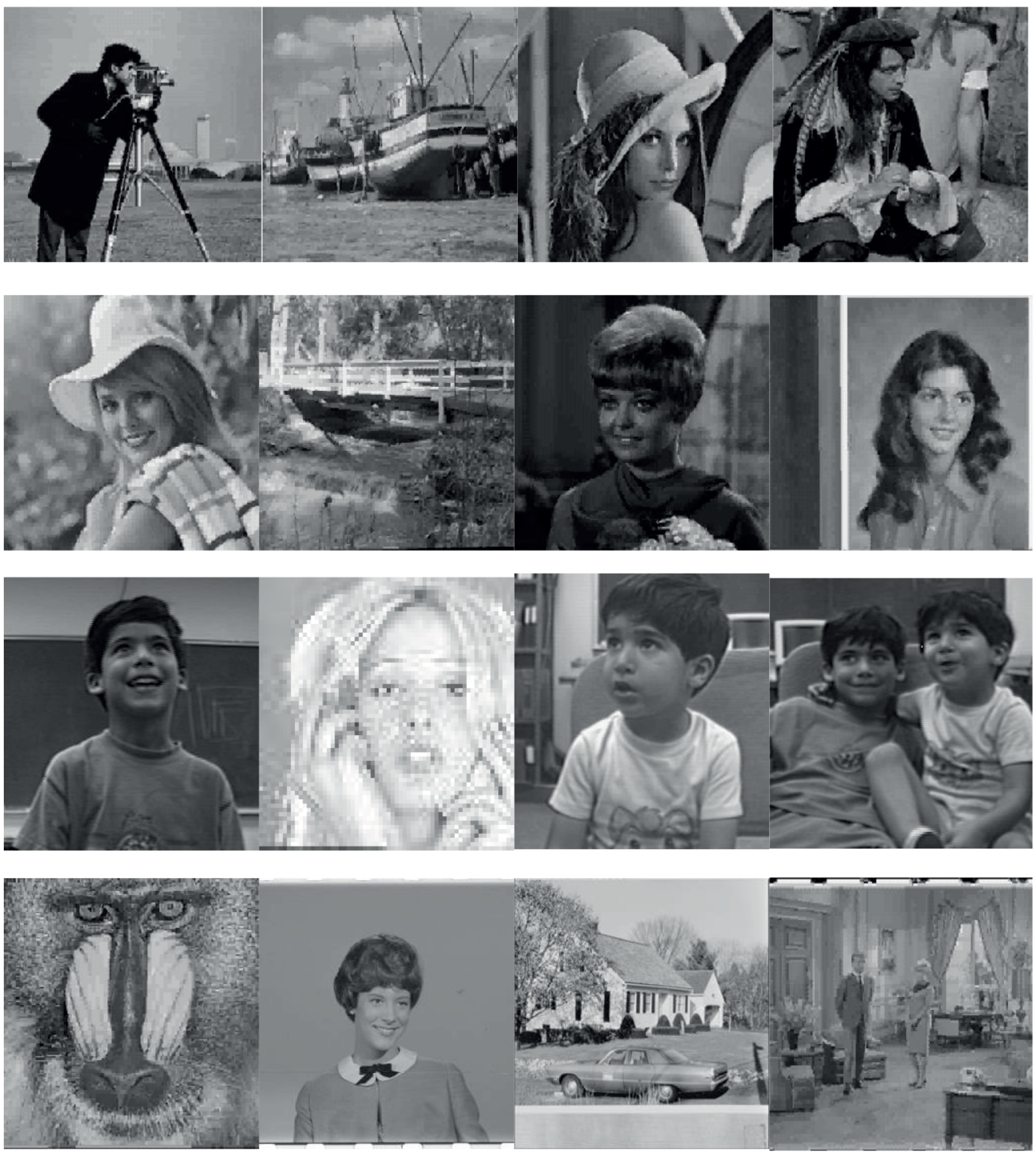

Figure 7. Image database for searching.

Table 3. Code vector matching rates for different search image parts.

	Image size	#of vectors	exact matching #of vectors	#of vectors <Th	#of vectors >Th
Tiffany_ eye	48 × 44	12 × 11	85	26	21
Girl3_ eye	32 × 32	8 × 8	43	9	12
Girl_ eye	64 × 52	16 × 13	133	42	33
Girl3_ nose	24 × 28	6 × 7	26	6	10
Tiffany_ nose	40 × 48	10 × 12	86	12	22
Tiffany_ mouth	60 × 44	15 × 11	118	32	15
Girl3_ mouth	40 × 24	10 × 6	28	14	18
Tire	32 × 28	8 × 7	25	13	18
Vase	24 × 40	6 × 10	22	18	20
Phone	36 × 24	9 × 6	11	16	27

5. Performance comparison between VQ and JPEG compressed search process

In JPEG compression the image is divided into 8 by 8 subblocks and discrete cosine transform (DCT) is applied to each subblock first. Average component and two-dimensional frequency components are transferred separately by using Huffman encoding. In the process of Huffman coding DCT components are zigzag scanned for efficiency of the compression [21]. For the comparison of VQ and JPEG search methods required processing power and similarity costs are given in Table 4. For each 8 by 8 JPEG image subblock there are four VQ codebook indexes since code vectors are 4 by 4 in size. The table is organized to give compressed and decompressed comparison results. Even in compressed comparison of image subblocks, it is needed to decode Huffman encoding and DE quantization of DCT coefficients in JPEG compression. The required numbers of multiplication, addition, and comparison operations are shown for VQ and JPEG compressed images in Table 4. Actually there is an overhead cost for VQ of the search image that is greater than JPEG compression cost. However, the compression of the search image is performed only once and the search process takes place for the entire image database. The overhead costs are not mentioned in the table, since they are ignorable for both JPEG and VQ compression case.

Table 4. Comparison of search methods by their operational cost for an 8 by 8 subblock.

For single 8 by 8 block	JPEG decompressed (Feig and Winograd algorithm for fast DCT)[22]	JPEG compressed (Huffman and zigzag decoding)	VQ compressed code book size 4 × 4 × 256	VQ compressed code book size 4 × 4 × 512
Number of multiplications	94	0	0	0
Number of additions	454	64	4	4
Number of comparisons	1	1	1	1

In Table 5, error performances of the methods are shown. As seen, in JPEG and VQ compression, average error per pixel has approximate magnitude of 3 and 7, respectively. VQ has a greater average error but in similarity tests when the search image is shifted one or two pixels the resulting error is higher than this error level and does not contribute to faulty decisions.

6. Results and conclusion

The results have shown that the proposed search method on vector quantized images is speed efficient and can be used to find different sized objects and part of images throughout the large compressed database images

Table 5. Comparison of search methods by their error performance for an 8 by 8 subblock.

For single 8 by 8 block	JPEG	VQ (code book size $4 \times 4 \times 256$)	VQ (code book size $4 \times 4 \times 512$)
Comparison error (PSNR)	38	29	31
MSE error per pixel	8	48	42

effectively. The location and position of similar images in the image database can be determined very quickly compared to different search methods, given the part of image or sample image objects.

This work shows that the index map of a VQ compressed object image can be searched directly on the compressed domain VQ index map of the image database and when compared to an exhaustive search, which reduces the required number of operations N to one (single comparison for each block). The method described here is considered for use on greater image databases and longer image sequences for the test of applicability in real time and reliability.

References

[1] Brent SW, Yuana CJ, Wei HI, Matthew DC. Object recognition in compressed imagery. Image Vision Comput 1998; 16: 337-352.

[2] Regentova E, Latifi S, Deng S. Image similarity estimation by processing compressed data. Image Vision Comput 2001; 19: 485-500.

[3] Bischof H, Wildenauer H, Leonardis A. Illumination insensitive recognition using eigenspaces. Comput Vis Image Und 2004; 95: 86-104.

[4] Cerra D, Datcu M. A fast compression-based similarity measure with applications to content-based image retrieval. J Vis Com Image Rep 2012; 23: 293-302.

[5] Drew M, Au J. Video key frame production by efficient clustering of compressed chromaticity signatures. ACM Multimedia'00, 30 Oct–3 Nov 2000; Los Angeles, CA, USA: pp. 365-368.

[6] Osadchy M, Keren D. Efficient detection under varying illumination conditions and image plane rotations. Comput Vis Image Und 2004; 93: 245-259.

[7] Park SB, Lee JW, Kim SK. Content-based image classification using a neural network. Pattern Recogn Lett 2004; 25: 287-300.

[8] Jain AK. Image data compression; a review. IEEE 1981; 349-389.

[9] Gersho A, Gray RM. Vector quantization and signal compression. Boston, MA, USA: Kluwer, 1992.

[10] Gray RM. Vector quantization. IEEE ASSP Mag 1984; 1: 4-29.

[11] Kunt M, Ikonomopoulos A, Kocher M. Second generation image coding techniques. IEEE 1985; 73: 549-574.

[12] Barnes CF, Rizvi SA, Nasrabadi NM. Advances in residual vector quantization. IEEE T Image Proces 1996; 5: 226-262.

[13] Wang L, Goldberg M. Progressive image transmission using vector quantization in pyramid form. IEEE T Com 1989; 37: 1339-1349.

[14] Yakut M. Vector quantization using nonrectangular blocks. ISCIS XVI International Symposium of Computer and Information Sciences; 5–7 Nov 2001; Antalya, Turkey: pp. 229-234.

[15] Enireddy V, Reddi KK. A data mining approach for compressed medical image retrieval. Int J Comput App 2012; 52: 975-987.

[16] Sayood K. Introduction to Data Compression. San Francisco, CA, USA: Morgan Kaufman Publishing, 1996.

[17] Lim JS. 2-D Signal and Image Processing. Englewood Cliffs, NJ, USA: Prentice Hall, 1990.

[18] Porter R, Canagarajah N. A robust automatic clustering scheme for image segmentation using wavelets. IEEE T Image Proces 1996; 5: 662-665.

[19] Smith B, Rowe L. Algorithms for manipulating compressed images. IEEE Comput Graph 1993; 13: 34-42.

[20] Amani C, Mounir K, Amel BB. An improved image retrieval algorithm for JPEG2000 compressed images. ISSPIT2012; 12–15 Dec, Vietnam, pp. 453-458.

[21] Pradyumna A, Sara G, Farine A. Retrieval of occluded images using DCT phase and region merging. ICIP 2012; 30 Sep–3 Oct, Orlando, Florida, USA. pp. 2441-2444.

[22] Ephraim F, Shmuel W. Fast algorithms for the discrete cosine transform. IEEE T Sig Proces 1992; 40: 274-2193.