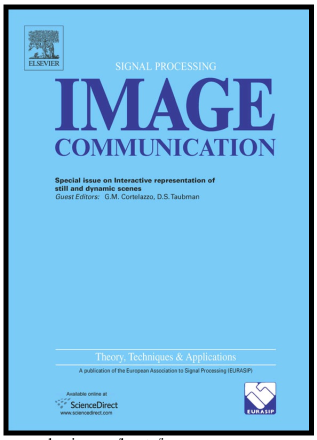
Author's Accepted Manuscript



www.elsevier.com/locate/image

Fast content access and retrieval of JPEG compressed images

M. Mehrabi, F. Zargari, M. Ghanbari, M.A. Shayegan

PII: S0923-5965(16)30062-5
DOI: http://dx.doi.org/10.1016/j.image.2016.05.005
Reference: IMAGE15083

To appear in: *Signal Processing : Image Communication*

Received date: 30 March 2015
Revised date: 14 December 2015
Accepted date: 7 May 2016

Cite this article as: M. Mehrabi, F. Zargari, M. Ghanbari and M.A. Shayegan Fast content access and retrieval of JPEG compressed images, *Signal Processin : Image Communication,* http://dx.doi.org/10.1016/j.image.2016.05.005

Fast Content Access and Retrieval of JPEG Compressed Images

M. Mehrabi[a], F. Zargari[b], M. Ghanbari[c], M. A. Shayegan[d]

[a]Corresponding author : M. Mehrabi is with Department of Computer Engineering, Shiraz Branch, Islamic Azad University, P.O. box 71993-3, Sadra highway, Shiraz, Iran , Tel.: +989395759990 , E-mail: mehrabi@iaushiraz.ac.ir.

[b]F. Zargari is with the Information Technology Research Institute of Iran Telecom Research Center (ITRC), North Kargar St., Tehran, Iran (e-mail: zargari@itrc.ac.ir).

[c]M. Ghanbari is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, CO4 3SQ UK as well as School of Electrical and Computer Engineering, University of Tehran, North Kargar St., Tehran Iran (e-mail: ghan@essex.ac.uk).

[d]M. A. Shayegan is with Department of Computer Engineering, Shiraz Branch, Islamic Azad University, P.O. box 71993-3, Sadra highway, Shiraz, Iran (e-mail: shaygan@iaushiraz.ac.ir).

Abstract — Fast content access and content based retrieval of images are among the common web and signal processing applications; on the other hand, JPEG is the dominant format for image compression in a wide range of applications. In this paper, a simple and fast method for content access and retrieval of JPEG coded images is presented which does not require complete decompression of coded images. The presented method uses DCT coefficients of coded blocks in the JPEG bit stream to extract average values of various size picture blocks namely DC values. The DC values provide an approximation of the coded image, which can be employed to construct a lower resolution picture or color histogram of the JPEG coded image for retrieval and other applications without full decompression of the image.

Index Terms — JPEG images, Fast content access, DC image, Image retrieval, Compressed domain.

1. INTRODUCTION

Nowadays huge amount of visual information like images and videos are created and communicated each day. As a result, management and analysis of visual information is becoming a challenging issue in the multimedia and signal processing research areas and applications. Popularity of various portable devices such as digital cameras or mobile phones, which can capture images or videos easily, and sharing them across the web make image content management and analysis more important than before. On the other hand, in order to tackle with bandwidth and storage problems, most of these visual data are either stored or communicated around the world in compressed form such as JPEG or MPEG. However, compressed visual data increases the processing time.

Since, management, analysis or indexing of visual information is conventionally carried out in the pixel domain; therefore, for compressed data, the decompression process will be added to the complexity of the data processing and analysis. As a result, processing images or videos directly in compressed format is an attractive issue in different areas and many methods in different applications and research areas have been proposed to process and analyze visual data directly in compressed domain. They range from image restoration, or enhancement [1][2] , to face recognition and forgery or tampered detection [3-5], and more widely in content based image and video retrieval [6-9]. On the other hand, several attempts have been made to make a standard framework for content accessing, retrieval and summarizing of compressed visual data say JPSearch for JPEG image format and MPEG-7 for compressed videos [10],[11].

While various transforms are accepted as the coding core of image and video compression methods, transform coefficients are widely used for analyzing and accessing the content of coded visual data. For example, in JPEG image compression, and the MPEG/H.26X video coding standards which employ block based DCT transform [12], processing of the DCT coefficients is widely performed in the compressed domain for different applications [3-5].

Delac et al. [3] use all DCT coefficients of an image as a vector for face recognition in JPEG images. In JPEG domain, the distribution of DCT coefficients as a histogram is also used for forgery localization by

Bianchi et al. [4] and for tampered JPEG image detection by Lin et al. [5]. Furthermore, the combination of DCT coefficients is used as feature vector in content based image and video retrieval. Poursistani et al. [13] used the first 36 coefficients of luminance and the DC terms of chrominance components of an 8×8 transformed block in a JPEG domain to form a code-vector to index JPEG images. The non-zero coefficients are also used to identify dominant color of compressed image and proposed a retrieval method based on the color descriptor [14]. Feng et al [15] used statistics of the DCT coefficients say a set of moments, to form a feature vector to retrieve compressed images. Eom et al. [16] defined an edge histogram based on AC coefficients of DCT transform. Another feature vector, reported in the literature is rough shape descriptor based on DCT moment to retrieve the shape by Ngo et al. [17].

Another important technique for fast and easy access and browsing of compressed visual data is to construct a low-resolution picture, known as DC image [18-21], instead of performing inverse DCT transform and full decompression. The DC coefficient of the DCT transformed block, which is the average of pixels in the block, can reconstruct an approximated picture for fast access to the content of the original image. This picture is a lower resolution resemblance of the original picture. The resolution of DC image depends on the size of the averaged block. The smaller is the block, the higher are the resolution and the quality of the DC images [21].

The DC images play an important role in visual content indexing especially for compressed videos . Reconstruction of these images can be fast and can be used to compare the contents of compressed visual data. Zhuo et al. [22] use the DC image to construct scalable color descriptor (SCD), the MPEG-7 color descriptor of JPEG images. SCD is then used for pornographic image recognition. Wong et al. form a histogram based on DC values to compare image similarities. Then a genetic algorithm is used to search similar images [23]. The DC terms of 4×4 blocks are also used to construct color histogram feature vector by Lu et al. [24] for retrieval of JPEG images. Another example for DC image application is shot or scene change detection of video, which mainly deals with the global changes between pictures or group of pictures in a video sequence. Since DC images can represent global pixel changes in successive frames better than the original pictures, these approximated pictures are widely used in video Shot and scene change detection in compressed domain [25-33].

One important feature of DC images is that for some applications such as image browsing, these approximated pictures are more suitable than the original ones [20][31][32]. Visual information retrieval and indexing are among other applications that DC-images are better suited; because, they are based on grouping pixels with similar values [25][33].

Due to wide applications of DC images, a method for extracting DC values from the DCT transformed block has been proposed by Jiang et al. [34], and they extended it to JPEG coded pictures [31]. It has been claimed that this method is the fastest and has the best performance among all JPEG compressed domain retrieval techniques [9][35]. In this method, the basis vectors are used to extract DC coefficients of small sub-blocks from the DCT coefficients of larger blocks.

In the present paper, a novel method of extracting the average (DC value) of 4×4 and 2×2 sub-blocks of an 8×8 DCT coded block directly from its JPEG bit stream is presented. These average values can be used to construct color histogram or an approximated picture (DC image) of original image for use in image retrieval and other applications. Performance evaluation results indicate that the proposed method can considerably reduce the operation counts compared to inverse DCT and the method by Jiang et al [31], while preserving the quality of the output DC images to high extent. Moreover, the proposed method can be used for fast content access of JPEG images in image retrieval and browsing applications. Simulation results indicate that the proposed method achieves retrieval performance similar to the previous method, while, reducing the processing time.

The rest of the paper is organized as follows. In section II, the proposed method is introduced. The performance and computational cost of the proposed method are evaluated in section III, and the concluding remarks are given in section IV.

2. The Proposed Method

The core of the JPEG coding standard is the DCT transform performed on 8×8 blocks of the pixels [36]. The output of DCT transform is 64 coefficients grouped in an 8×8 matrix. The decoder uses inverse DCT transform to reconstruct pixels of the 8×8 transformed block as (1):

$$X = C^T Y C \tag{1}$$

where Y is the DCT coefficients matrix embedded in JPEG bitstream, C and C^T are DCT inverse transformation matrix and its transpose respectively, and X is the decoded block of pixels. Matrix C is defined as (2).

$$C = \begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\ 0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & -0.4157 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\ 0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975 \end{bmatrix} \tag{2}$$

If X represents an 8×8 decoded block from a JPEG coded block by (3), matrix M_2 in (4) is proposed to extract the averages of 2×2 sub-blocks of X through operation given in (5).

$$X = \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} & x_{04} & x_{05} & x_{06} & x_{07} \\ x_{10} & x_{11} & x_{12} & x_{13} & x_{14} & x_{15} & x_{16} & x_{17} \\ x_{20} & x_{21} & x_{22} & x_{23} & x_{24} & x_{25} & x_{26} & x_{27} \\ x_{30} & x_{31} & x_{32} & x_{33} & x_{34} & x_{35} & x_{36} & x_{37} \\ x_{40} & x_{41} & x_{42} & x_{43} & x_{44} & x_{45} & x_{46} & x_{47} \\ x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} & x_{56} & x_{57} \\ x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} & x_{67} \\ x_{70} & x_{71} & x_{72} & x_{73} & x_{74} & x_{75} & x_{76} & x_{77} \end{bmatrix} \tag{3}$$

$$M_2 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

$$DC_{2\times2} = \frac{1}{4}(MXM^T) = \begin{bmatrix} dc_{00} & 0 & \mathrm{dc}_{02} & 0 & dc_{04} & 0 & dc_{06} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ dc_{20} & 0 & dc_{22} & 0 & dc_{24} & 0 & dc_{26} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ dc_{40} & 0 & dc_{42} & 0 & dc_{44} & 0 & dc_{46} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ dc_{60} & 0 & dc_{62} & 0 & dc_{64} & 0 & dc_{66} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5}$$

Referring to (5), the non-zero elements of matrix $DC_{2\times2}$ are calculated according to (6). Here x_{ij} is an element at the i-th row and j-th column of matrix X which is the reconstructed image pixel.

$$\begin{aligned} dc_{00} &= \frac{1}{4}(x_{00} + x_{01} + x_{10} + x_{11}) \\ dc_{20} &= \frac{1}{4}(x_{20} + x_{21} + x_{30} + x_{31}) \\ dc_{02} &= \frac{1}{4}(x_{02} + x_{03} + x_{12} + x_{13}) \\ dc_{mn} &= \frac{1}{4}\sum_{i=m}^{m+1}\sum_{j=n}^{n+1} x_{ij} \end{aligned} \tag{6}$$

According to (6), the elements of matrix $DC_{2\times2}$ are the averages of 2×2 blocks in the reconstructed block of X in (5). Now matrix X in (5) can be replaced with the right hand side of (1) say C^TYC, which results $DC_{2\times2}$ as in (7).

$$DC_{2\times2} = \frac{1}{4}(M_2 X M_2^T) = \frac{1}{4}(M_2(C^T Y C) M_2^T) \tag{7}$$

where in (7), $DC_{2\times2}$ is the matrix which its elements are average of 2×2 block from 8×8 block of original pixels. Based on associative property of matrix multiplication, (7) concludes (8):

$$DC_{2\times2} = \frac{1}{4}(M_2 C^T (Y) C M_2^T) \tag{8}$$

Considering $N_2 = M_2 C^T$ then $DC_{2\times2}$ in (7) can be calculated as (9):

$$DC_{2\times2} = \frac{1}{4} N_2 (Y) N_2^T \tag{9}$$

Where matrix N_2 is derived based on (2) and (4) as (10):

$$N_2 = \begin{bmatrix} 0.7072 & 0.9601 & 0.6532 & 0.3182 & 0 & -0.2126 & -0.2706 & 0.1803 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.7072 & 0.3753 & -0.6532 & -0.7682 & 0 & 0.5132 & 0.2706 & -0.0747 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.7072 & -0.3753 & -0.6532 & 0.7682 & 0 & -0.5132 & 0.2706 & 0.0747 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.7072 & -0.9061 & 0.6532 & -0.3182 & 0 & 0.2126 & -0.2706 & 0.1803 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{10}$$

Since in (5), the elements of $DC_{2\times 2}$ are averages of 2×2 sub-blocks of matrix X, given the coefficients of an 8×8 JPEG coded block say Y, matrix N_2 in (10) can be used to generate matrix $DC_{2\times 2}$ directly from DCT coefficients matrix Y using (9). As mentioned already, the nonzero elements of $DC_{2\times 2}$ are the averages of 2×2 sub-blocks in the reconstructed pixels say matrix X in (1).

The same approach can be used to generate $DC_{4\times 4}$ which its elements are 4×4 sub-block average. In this case, X is an 8×8 decoded block in (1) and M_4 is given as (11):

$$M_4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (11)$$

Replacing M_2 in (7) by M_4 will yield matrix $DC_{4\times 4}$ whose elements are the averages of 4×4 sub-blocks of an 8×8 matrix X. Once more, the right hand side of the equality in (1) can be used instead of X that results $DC_{4\times 4}$ in (12).

$$DC_{4\times 4} = \frac{1}{16}(M_4(C^T Y C)M_4^T) = \frac{1}{16} N_4 Y N_4^T \quad (12)$$

where $N_4 = M_4 C^T$, and N_4 is calculated as (13).

$$N_4 = \begin{bmatrix} 0.1414 & 0.1281 & 0 & -0.45 & 0 & 0.3006 & 0 & -0.255 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1414 & -0.1281 & 0 & 0.45 & 0 & -0.3006 & 0 & 0.255 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (13)$$

The non-zero elements of $DC_{2\times 2}$ and $DC_{4\times 4}$ matrices are averages of 2×2 and 4×4 blocks of decoded JPEG picture (DC values), and can be used to reconstruct DC images, where their resolutions are quarter and one sixteenth of the original picture size, respectively. Comparing the matrices N_2 and N_4 in (10) and (13) with matrix C in (2) reveal that they have more zeros and identical elements than C, and as a result, calculating $DC_{2\times 2}$ and $DC_{4\times 4}$ (extracting DC images) have less computational cost than the normal inverse DCT in (1). It is worth noting that aforementioned calculations can be performed in any color space and for any color component of image pixels.

On the other hand, without any further processing, the average of blocks can be grouped in a histogram to generate the color histogram of the decoded picture. These histograms are used as feature vectors for content based image retrieval, and hence JPEG coded pictures can be retrieved without full decompression and normal inverse DCT. Since the frequency of color histogram bins varies by changing block size that its average is used, we employed the normalized color histogram. The normalized color histogram feature vector for a color image including three color components A, B and C is calculated as (14):

$$H(x_k, y_m, z_n) = \sum_i^W \sum_j^H f(P_{ij}) / (W \times H) \quad (14)$$

where P_{ij} is the pixel or here block average at the i-th row and j-th column of the image, H and W are the picture height and width, respectively, and the function f(Pij) is defined in (15). Furthermore, x_k, y_m, and z_n are the decision boundaries of A, B and C color components.

$$f(P_{ij}) \begin{cases} = 1 & \\ & if: \quad x_k \le A(P_{ij}) \le x_{(k+1)}, \\ & and \quad y_m \le B(P_{ij}) \le y_{(m+1)}, \\ & and \quad z_n \le C(P_{ij}) \le z_{(n+1)}. \\ = 0 & \\ & otherwise. \end{cases} \quad (15)$$

where $A(P_{ij})$, $B(P_{ij})$ and $C(P_{ij})$ are three color components of pixel P_{ij}, respectively, and x_k, y_m, and z_n are the decision boundaries of three color components, the same as in (14).

The number of decision boundaries defines the number of histogram bins. Therefore, to reduce histogram size, eight bins are chosen for each histogram. This means each bin for a color component is defined according to the following decision boundaries:

$$x_k = k \times (255/8),\ y_m = m \times (255/8),\ \ z_n = n \times (255/8) \\ k, m, n = 1, 2, ..., 8. \quad (16)$$

The histogram intersection S, is selected as similarity metric of two histograms. The intersection of two histograms H1 and H2 of images is calculated as (17):

$$s = \sum_{k=1}^{8} \sum_{m=1}^{8} \sum_{n=1}^{8} (Min\{H1(x_k, y_m, z_n), H2(x_k, y_m, z_n)\}) \quad (17)$$

Since H1 and H2 in (17) are normalized by W×H, then S is a number in the range of [0, 1] and is a measure to represent the similarity between two images with color histograms H1 and H2.

3. Performance Evaluation

Performance of the proposed method is evaluated in various domains including the quality of extracted DC image, retrieval efficiency, and the computational complexity. They are also compared with the previous method of DC image extraction technique in the JPEG compressed domain [31]. The uncompressed color image database UCID is used as the benchmark dataset [37]. The image database consists of 1338 uncompressed images from various scenes. Each image was coded by the JPEG encoder using standard quantization matrix with nominal quality factor equal to 50 (Q=50). Each image in the database is decoded with JPEG decoder (full decompression); this provides a benchmark database that other methods can be compared with. Furthermore, for each JPEG coded image an approximated picture was extracted in four methods without full decompression. At first, methods by Jiang et al [31] are used, which construct approximated pictures or DC images based on calculating the averages of 4×4 and 2×2 blocks (instead of inverse DCT transform in JPEG decoder). Here the approximated pictures are reconstructed by replacing each pixel of the original image with its average block that forms an up-sampled image from the

DC image. The next two methods extract averages of 4×4 and 2×2 blocks based on the proposed method and generate the approximated pictures as the previous method. As a result, there are five new image databases extracted from JPEG coded images of UCID database: JPEG fully decoded image database and four approximated image (up-sampled from DC images) databases using proposed method and the previous method based on average of 4×4 and 2×2 blocks. These decoded databases that contain approximated images are used in different evaluation steps.

In different evaluation steps, we compare the efficiency of the proposed method for generating approximated image or DC-image in terms of complexity analysis and quality of approximated images and their performance in content based retrieval experiments. Here, the complexity of the proposed method is calculated as the number of operations needed to extract DC images. On the other hand, the visual qualities of approximated images generated by aforementioned methods are compared for various images. We used average peak signal to noise ratio (APSNR) as an objective measure [38] for evaluating the quality of the approximated images in comparison with original images. Finally, the approximated images were employed in an image retrieval experiment to evaluate the accuracy of the proposed method for direct retrieval of JPEG coded images without full decompression.

In first step of evaluation, Table 1 compares the quality of the five methods say five decoded databases, compared to the original uncompressed database in APSNR. The APSNRs in the table are computed based on the average PSNRs of three color components and for the entire images in the database.

Referring to table 1, the objective qualities of the approximated images are acceptable and are at most 4 dB worse than the fully decoded images. Fig. 1 shows samples of approximated pictures constructed by four methods and fully decoded images to compare their subjective quality. It indicates that the extracted pictures by the proposed method have satisfactory subjective quality.

TABLE 1
QUALITY OF EXTRACTED IMAGES FROM JPEG CODED IMAGES IN TERM OF APSNR FOR VARIOUS METHODS IN dB

Method	APSNR(dB)
Fully decoded images	30.47
Approximated images by using average of 2×2 block with previous method [31]	26.28
Approximated images by using average of 4×4 block with previous method [31]	23.07
Approximated images by using average of 2×2 blocks with proposed method	26.39
Approximated images by using average of 4×4 blocks with proposed method	23.06

In the next step of evaluation, the computational load of the proposed method was compared with the normal inverse transform in the JPEG standard (benchmark) and the other methods, as well. The number of operations for producing the averages of 2×2 blocks and 4×4 blocks for an 8×8 transformed block in the four tested methods are tabulated in Table 2. Considering the total number of operations, Table 2 indicates substantial reduction in the computational cost of the proposed method compared to the normal inverse DCT and the previous method [31]. In fact, by using the proposed method instead of the normal inverse DCT, the savings in the total number of operations for 4×4 and 2×2 blocks are more than 98% and 96%, respectively. Whereas for the previous method by Jiang et al [31] these values are 93% and 92%, respectively.

Higher number of operations for extracting average of 2×2 block with the previous method is due to the fact that according to Jiang et al [31], averages of 2×2 blocks are calculated in two steps. At first, the 4×4

DCT coefficients are generated from 8×8 DCT coefficients, and then the 4×4 DCT coefficients are used to produce the averages of 2×2 blocks.

TABLE 2
NUMBER OF OPERATIONS FOR AN 8×8 TRANSFORMED BLOCK

METHOD	MULTIPLICATIONS	ADDITIONS	TOTAL
Full inverse transform	4096	4032	8128
Extracting average of 2×2 block with previous method [31]	288	368	656
Extracting average of 4×4 block with previous method [31]	256	320	576
Extracting average of 2×2 block with proposed method	144	184	328
Extracting average of 4×4 block with proposed	72	92	164

In another phase of the performance evaluation, the proposed method was employed in an image retrieval experiment to evaluate its accuracy for direct retrieval of JPEG coded images without full decompression. At this stage, the five aforementioned methods were compared to retrieve JPEG coded images. Hence, a query by example retrieval experiment was performed based on color histogram derived from five aforementioned extracted image databases. For the first extracted image database (fully decoded JPEG images), the color histogram for each image was constructed by using (14) and the pixels of the decoded image (P_{ij}). Nevertheless, for the four other approximated image databases, the averages of blocks from the corresponding algorithms were used directly instead of P_{ij} in (14). The retrieval experiment was performed for a query set including 64 images with highest number of similar images in the original databases. For each image in the query set, the retrieved images were rank ordered in a list based on similarity metric in (17).

In the retrieval performance analysis, average normalized modified retrieval rank (ANMRR) [39] is calculated for query set in different retrieval methods. ANMRR results are in the range of [0-1], where ANMR=0 indicates complete similarity was found as the highest ranking retrieved images and ANMR=1 indicates that none of them were found within K retrieved images in the rank ordered retrieved list. In these retrieval experiments, K was equal to 10. Table 3 reports the ANMRR values averaged over all images in the query set for different methods. Table 3 indicates that the retrieval efficiency of the entire methods based on approximated images is close to the fully decoded one. This is duo to the fact that the low reduction in the quality of DC images does not affect the retrieval performance. However, the proposed method has considerable reduction in computational load and processing time.

TABLE 3
RETRIEVAL PERFORMANCE OF PROPOSED METHOD COMPARED WITH FULLY DECODED JPEG IMAGES AND PREVIOUS METHOD

Method	ANMRR
Fully decoded images	0.1917
Approximated images by using average of 2×2 block with method in [31]	0.1919
Approximated images by using average of 4×4 block with method in [31]	0.1929
Approximated images by using average of 2×2 blocks with proposed method	0.1914
Approximated images by using average of 4×4 blocks with proposed method	0.1936

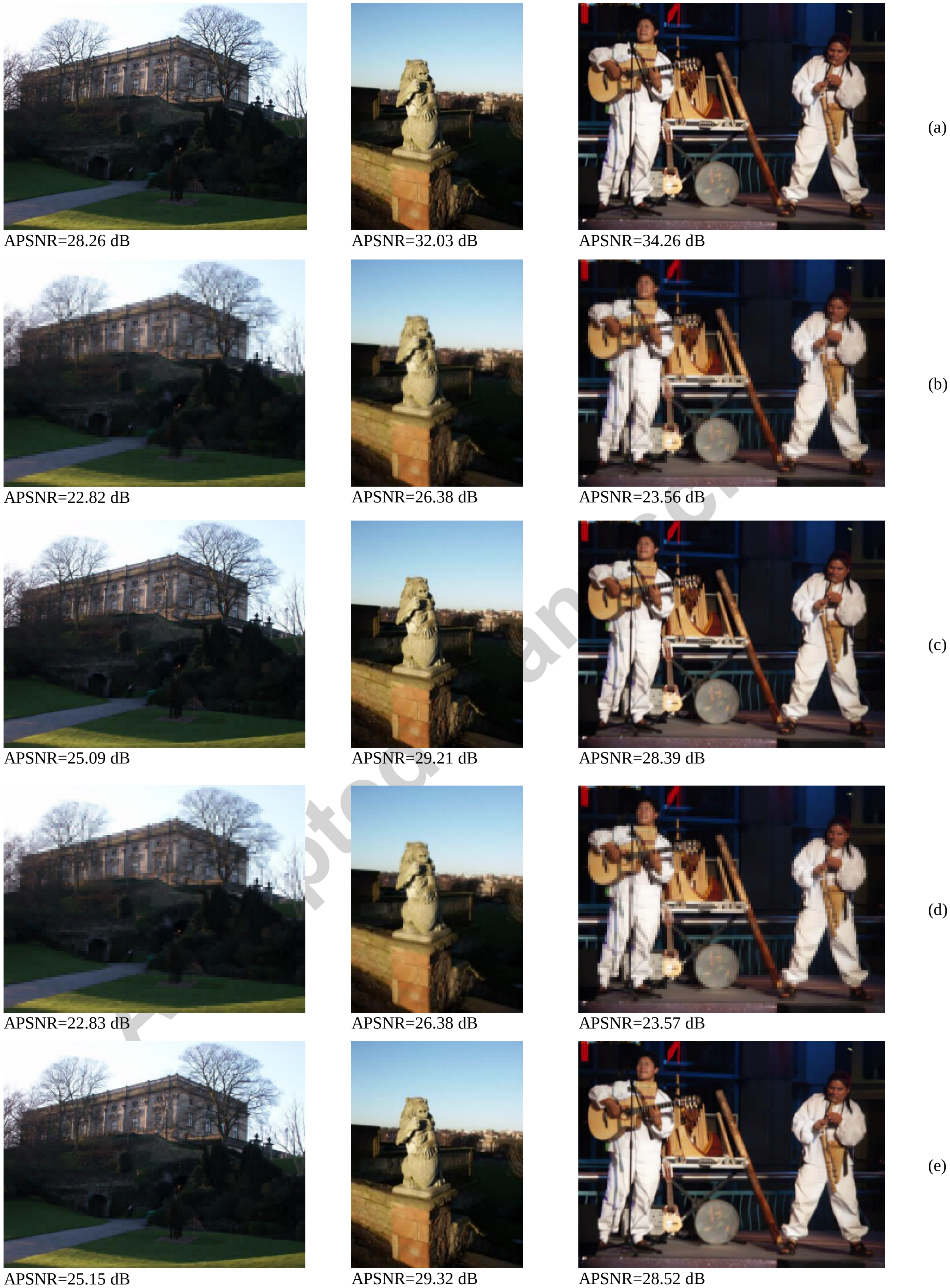


Fig. 1: Samples of extracted pictures from different methods and their APSNRs (a) Normal JPEG decoding (b) Approximated pictures (DC-image) of 4x4 block using previous method [31] (c) Approximated pictures (DC-image) of 2×2 blocks using previous method [31] (d) and (e) Proposed approximated pictures (DC-image) of 4×4 blocks and 2×2 blocks respectively.

4. Conclusion

In this paper, we proposed a novel approach to simplify the inverse DCT transform in the JPEG standard in order to extract averages of 4×4 and 2×2 image blocks from the JPEG coded images at a lower computational cost. In this way, approximated pictures (DC images) are produced that can be used for different applications such as browsing which does not need high quality images. Moreover, the extracted block averages can be directly used for color indexing of JPEG images. The proposed method generates block averages from compressed JPEG bitstream with the same accuracy as earlier works but it is faster. Experimental results indicate that the APSNR of approximated pictures generated by the proposed method are at most 4 dB worse than the fully decoded images (Table 1). However, the reduction ratios in the number of total operations for 4×4 and 2×2 are 98% and 96%, respectively. The proposed method is also compared with the method [31] for extracting the averages of 2×2 and 4×4 blocks, which has been claimed to be the best method for retrieving JPEG images in compressed domain [9][35]. The simulation results indicate that in addition to the slight improvement in the picture quality, the proposed method is about twice faster than using the method by Jiang et al [31].

The low complexity of the proposed method and slight reduction in the quality of the resulted DC images indicate that the proposed method can be used as a fast and simple method for extracting content-preserved DC images of JPEG coded pictures without full decompression. These DC images may also be used in different applications such as compressed domain analysis, browsing and indexing of JPEG images.

5. Akcnowlegment

The work presented in this paper is supported under the research project by Shiraz branch of Islamic Azad University, Shiraz, Iran.

References

[1] L. Ma, D. Zhao, W. Gao, Learning-based image restoration for compressed images, Signal Process. Image Commun. 27 (2012) 54–65. doi:10.1016/j.image.2011.05.004.

[2] C.K. Kuo, C. M., Yang, N. C., Liu, C. S., Tseng, P. Y., Chang, An effective and flexible image enhancement algorithm in compressed domain, Multimed. Tools Appl. (2014) 1–24.

[3] K. Delac, M. Grgic, S. Grgic, Face recognition in JPEG and JPEG2000 compressed domain, Image Vis. Comput. 27 (2009) 1108–1120. doi:10.1016/j.imavis.2008.10.007.

[4] T. Bianchi, A. De Rosa, A. Piva, Improved DCT coefficient analysis for forgery localization in JPEG images, ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc. (2011) 2444–2447. doi:10.1109/ICASSP.2011.5946978.

[5] Z. Lin, J. He, X. Tang, C.K. Tang, Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis, Pattern Recognit. 42 (2009) 2492–2501. doi:10.1016/j.patcog.2009.03.019.

[6] Babu, R. Venkatesh, Manu Tom, P. Wadekar , A survey on compressed domain video analysis techniques, Multimed. Tools Appl. (2014) 1–36.

[7] H. Wang, A. Divakaran, A. Vetro, S.F. Chang, H. Sun, Survey of compressed-domain features used in audio-visual indexing and analysis, J. Vis. Commun. Image Represent. 14 (2003) 150–183. doi:10.1016/S1047-3203(03)00019-1.

[8] S.J. Hou, S.B. Zhou, M.A. Siddique, A compressed sensing approach for query by example video retrieval, Multimed. Tools Appl. 72 (2014) 3031–3044. doi:DOI 10.1007/s11042-013-1573-y.

[9] D. Edmundson, G. Schaefer, Performance comparison of JPEG compressed domain image retrieval techniques, Signal Process. Commun. Comput. (ICSPCC), 2012 IEEE Int. Conf. (2012) 587–592. doi:10.1109/ICSPCC.2012.6335725.

[10] K. Yoon, Y. Kim, J.H. Park, J. Delgado, A. Yamada, F. Dufaux, et al., JPSearch: New international standard providing interoperable framework for image search and sharing, Signal Process. Image Commun. 27 (2012) 709–721. doi:10.1016/j.image.2012.05.001.

[11] R. Koenen, F. Pereira, MPEG-7: A standardized description of audiovisual content, Signal Process. Image Commun. 16 (2000) 5–13. doi:10.1016/S0923-5965(00)00014-X.

[12] M. Ghanbari, Standard codecs: Image compression to advanced video coding, 3rd edition, IET Press (2011).

[13] P. Poursistani, H. Nezamabadi-pour, R. Askari Moghadam, M. Saeed, Image indexing and retrieval in JPEG compressed domain based on vector quantization, Math. Comput. Model. 57 (2013) 1005–1017. doi:10.1016/j.mcm.2011.11.064.

[14] J. Jiang, Y. Weng, P. Li, Dominant colour extraction in DCT domain, Image Vis. Comput. 24 (2006) 1269–1277. doi:10.1016/j.imavis.2006.04.009.
[15] G. Feng, J. Jiang, JPEG compressed image retrieval via statistical festures, Pattern Recognit. 36 (2002) 977–985. doi:10.1016/S0031-3203(02)00114-0.
[16] M. Eom, Y. Choe, Fast extraction of edge histogram in dct domain based on mpeg7, in: World Acad. Sci. Eng. Technol., 2005: pp. 209–212.
[17] C.W. Ngo, T.C. Pong, R.T. Chin, Exploiting image indexing techniques in DCT domain, Pattern Recognit. 34 (2001) 1841–1851. doi:10.1016/S0031-3203(00)00111-4.
[18] B. L. Yeo, B. Liu, Rapid scene analysis on compressed video, IEEE Trans. Circuits Syst. Video Technol. 5 (1995) 533–544. doi:10.1109/76.475896.
[19] F. Shi, X. Guo, Keyframe extraction based on kmeas results to adjacent DC images similarity, ICSPS 2010 - Proc. 2010 2nd Int. Conf. Signal Process. Syst. 1 (2010) 611–613. doi:10.1109/ICSPS.2010.5555457.
[20] M. Mehrabi, F. Zargari, M. Ghanbari, Fast and low complexity method for content accessing and extracting DC-pictures from H.264 coded videos, Consum. Electron. IEEE Trans. 56 (2010) 1801–1808. doi:10.1109/TCE.2010.5606329.
[21] B. L. Yeo, Bede Liu, On the extraction of DC sequence from MPEG compressed video, in: Int. Conf. Image Process., 1995: pp. 260–263. doi:10.1109/ICIP.1995.537464.
[22] L. Zhuo, J. Zhang, Y. Zhao, S. Zhao, Compressed domain based pornographic image recognition using multi-cost sensitive decision trees, Signal Processing. 93 (2013) 2126–2139. doi:10.1016/j.sigpro.2012.07.003.
[23] H.-S. Wong, H.H.S. Ip, L.P.L. Iu, K.K.T. Cheung, L. Guan, Transformation of compressed domain features for content-based image indexing and retrieval, Multimed. Tools Appl. 26 (2005) 5–26. doi:10.1007/s11042-005-6847-6.
[24] Z.-M. Lu, S.-Z. Li, H. Burkhardt, A content-based image retrieval scheme in JPEG compressed domain, Int. J. Innov. Comput. Inf. Control. 2 (2006) 831–839.
[25] S. Bekhet, A. Ahmed, A. Hunter, Video matching using DC-image and local features, Lect. Notes Eng. Comput. Sci. 3 (2013) 2209–2214.
[26] J.R. Kim, S. Suh, S. Sull, Fast Scene Change Detection for Personal Video Recorder, IEEE Trans. Consum. Electron. 49 (2003) 683–688. doi:10.1109/TCE.2003.1233802.
[27] K. Seo, S. Park, S. Jung, Wipe scene-change detector based on visual rhythm spectrum, IEEE Trans. on Consumer Electron., 55, No. 2, (2009), 831-838.
[28] Z. Huang, W.W.Y. Ng, P.P.K. Chan, J. Li, D.S. Yeung, Video shot boundary detection using RBFNN minimizing the L-GEM, IEEE Int. Conf . in Mach. Learn. Cybern. (ICMLC), (2010) 2156–2160.
[29] J.L. Zheng, M.J. Li, M.X. Zhang, J. Zhou, Z.D. Liu, An effective framework of shot segmentation based on I-frame in compressed-domain videos, IEEE Int. Conf. in: Wavelet Anal. Pattern Recognit. (ICWAPR), (2012) 84–90.
[30] J. Almeida, N.J. Leite, R.D.S. Torres, VISON: VIdeo Summarization for ONline applications, Pattern Recognit. Lett. 33 (2012) 397–409. doi:10.1016/j.patrec.2011.08.007.
[31] J. Jiang, A. Armstrong, G.C. Feng, Direct content access and extraction from JPEG compressed images, Pattern Recognit. 35 (2002) 2511–2519. doi:10.1016/S0031-3203(01)00217-5.
[32] S. Soman, M. Ghorpade, V. Sonone, S. Chavan, Content based image retrieval using advanced color and texture features, in: Int. Conf. Comput. Intell., 3 (2012).
[33] J. Almeida, N.J. Leite, R.D.S. Torres, Online video summarization on compressed domain, J. Vis. Commun. Image Represent. 24 (2013) 729–738. doi:10.1016/j.jvcir.2012.01.009.
[34] J. Jiang, G. Feng, The spatial relationship of DCT coefficients between a block and its sub-blocks, IEEE Trans. Signal Process. 50 (2002) 1160–1169. doi:10.1109/78.995072.
[35] D. Edmundson, G. Schaefer, An overview and evaluation of {JPEG} compressed domain retrieval techniques, in: IEEE ELMAR Proc., (2012) 75–78.
[36] G.K. Wallace, The JPEG still picture compression standard, IEEE Trans. Consum. Electron. 38 (1992) 1–17. doi:10.1109/30.125072.
[37] G. Schaefer, M. Stich, UCID - An Uncompressed Colour Image Database, in: SPIE, Storage Retr. Methods Appl. Multimed., 2003: pp. 472–480. doi:10.1117/12.525375.
[38] Q. Huynh-Thu, M. Ghanbari, Scope of validity of PSNR in image/video quality assessment, Electron. Lett. 44 (2008) 800. doi:10.1049/el:20080522.
[39] Moving Picture Experts Group, Description of core experiments for MPEG-7 color/texture descriptors, Tech. Rep. ISO/IEC JTC1/SC29/WG11/ N2929, 1999.

HIGHLIGHTS

- A novel fast algorithm to access the content of JPEG compressed images is presented.
- Extracting 2×2 and 4×4 sub-block averages from JPEG bitstream without full decompression.
- The extracted sub-block averages can directly construct color histogram or an approximated picture.
- The presented method can be efficiently used for fast exploring or retrieval of JPEG coded images.
- Finding block averages from JPEG bitstream at the same accuracy as earlier works but faster.