COMPRESSED DOMAIN JPEG2000 IMAGE INDEXING METHOD
EMPLOYING FULL PACKET HEADER INFORMATION

Farzad Zargari¹, Ali Mosleh², Mohammad Ghanbari³

¹ Multimedia Systems Research Group, IT Faculty, Iran Telecom Research Center, Tehran, Iran
² Department of Computer Engineering, Islamic Azad University, Science and Research Campus, Tehran, Iran
³ Department of Computing and Electronic Systems, University of Essex, Colchester, United Kingdom

Email: zargari@itrc.ac.ir, mosleh.a@srbiau.ac.ir, ghan@essex.ac.uk

ABSTRACT

In this paper we propose a new indexing method for comparing image contents in the JPEG2000 compressed domain. Comparing image contents of the JPEG2000 coded images in the pixel domain requires image decompression, which imposes intensive computational processes of the inverse discrete wavelet transform and the arithmetic decoding. On the other hand, the first decoding stage of the JPEG2000 standard is packet header decoding, which is a simple process, but provides valuable information about the code blocks in the packet. In this paper we use packet header information for JPEG2000 image indexing. The proposed method exploits full packet header information including the number of non-zero bit-planes, the number of coding passes and the code block length for indexing the JPEG2000 compressed images. Experimental results show that the proposed method provides a better performance compared to other JPEG2000 compressed domain indexing techniques and even outperforms the pixel-based image indexing techniques such as the Gabor filter.

1. INTRODUCTION

Visual information is being extensively used in various applications, such as digital libraries, remote sensing, medical imaging and multimedia systems. Hence, there are large image databases and visual management of these databases is a challenging problem. Content Based Image Retrieval (CBIR) and indexing is introduced for effective search and access to the huge amount of available data in image libraries. In CBIR the visual information of the image is used for indexing and retrieval of visual contents. Shape, color and texture are among the typical visual features that are used for image indexing and CBIR.

Since digital images have relatively large file sizes, they are stored mostly in the compressed form. Hence, in pixel-based image indexing techniques that require uncompressed data, the decompression process and time will be added to the retrieval computations and time. This is an annoying problem especially in real time image retrieval applications. Hence, it is desirable to develop CBIR methods which can be applied directly to the compressed domain.

The JPEG2000 [1] is the latest still image compression standard, which uses the Discrete Wavelet Transform (DWT). The DWT coefficients can be used to extract image features such as texture. Therefore most of the proposed JPEG2000 compressed domain image retrieval methods use DWT coefficients to provide image indexing. The proposed method in [2] uses the variance of the DWT coefficients in each sub-band as the feature vector for the JPEG2000 images. In [3] a retrieval technique is developed based on the region of interest (ROI) of JPEG2000 coded images. In this method the color and texture features of ROI are extracted using the ROI wavelet coefficients and in order to access the shape information, the ROI is decompressed. The method in [4] extracts color specifications of the image by creating a color histogram of LL0 sub-bands in the DWT of Y, Cb and Cr color components. They also use Gaussian Density (GD) of the wavelet coefficients to capture texture feature.

The DWT based JPEG2000 compressed domain image indexing methods are slow, because they require the most complex decompression stage of arithmetic decoding, to compute the wavelet coefficients. As a result, efforts have been directed towards JPEG2000 compressed domain image retrieval without arithmetic decoding of the coded image stream. In [5] code stream length of each sub-band is used for image content analysis. Using only sub-bands code stream lengths do not provide satisfactory image retrieval results, particularly for images which are compressed in lower bit rates. In highly compressed images, most of sub-bands code stream are omitted by the JPEG2000 rate-distortion mechanism. The proposed compressed domain image indexing technique in [6] uses mean and variance of maximum number of available bit-planes of code blocks in
each sub-band to index compressed images. Though the performance of [6] is better than [5], it fails in two grounds, i) in highly compressed images where the number of available code blocks is reduced, and ii) in sub-bands with variety of the number of coefficients in the code blocks, hence the indexing cannot be global.

In this paper we propose a JPEG2000 compressed domain image retrieval method which uses all the embedded information in the packet headers of the compressed image for indexing and the similarity measure. In our method only packet header decoding is required and the other time consuming decompression procedures of the JPEG2000 coded images are avoided. Simulation results indicate that the proposed method has superior performance to other packet header based, compressed domain JPEG2000 image retrieval methods such as [6], especially for indexing highly compressed images. This is due to the use of an efficient combination of full packet header information for indexing, so that the indexing can bring about satisfactory results even in the highly compressed images. Moreover, the proposed method outperforms the pixel-based image retrieval methods such as the Gabor filter.

The rest of this paper is organized as follows. In section 2 we provide a brief overview of the JPEG2000 standard. The proposed indexing method is explained in section 3. In section 4 simulation results are presented. Finally the paper concludes in section 5.

2. REVIEW OF THE JPEG2000 STANDARD

JPEG2000 is a recent international standard for image compression. Besides its excellent compression performance it has special error resilience and scalability features that make it a suitable image coding standard for a wide range of applications. In this section we provide a brief explanation about JPEG2000 standard which is relevant to this paper. More detailed explanation about this standard can be found at [1, 7, 8]. A block diagram of the encoding stages of a JPEG2000 encoder is illustrated in Fig. 1. The first stage which is optional divides the original input image into non overlapping rectangular tiles. In the next stage, the RGB color components are converted to YUV; then the multi-resolution DWT is applied to the image components resulting the DWT coefficients in different resolution levels and sub-bands. In this standard, the wavelet coefficients are partitioned into non overlapping blocks called the code blocks. After quantization, each code block is encoded independently in the Tier-1 stage. In this stage, the code blocks are divided into separate bit-planes and each bit-plane is entropy coded by a binary arithmetic coder. Bit-plane coding includes three coding passes: significance propagation, magnitude refinement and clean up. In each code block there might be a number of leading bit-planes that include only zero values (zero bit-planes). The number of these bit-planes is saved by a data structure, called zero tag tree in the packet header and no coding passes is carried out on these bit-planes. Finally to achieve the desired rate-distortion optimization, the Tier-2 Encoder generates the final bit stream from the selected coding passes in the bit stream. It is clear that in lossless compression, none of the coding passes are omitted and the entire generated bit stream at Tier-1 stage is used in the final bit stream.

Fig. 2 shows the structure of the JPEG2000 code stream. In this structure a set of code blocks, based on their spatial positions, form a larger unit called precinct. The precinct bit stream is placed in a packet. Each packet has a header (packet header), containing the necessary information for decoding its code blocks. The packet header includes the following information about the code blocks: the number of leading zero bit-planes of the code block, the number of included coding passes of the code block, the length (in bytes) of the code block bit stream in the packet data part and inclusion and zero length indicators of the code block. It is worth noting that the packet header does not provide any further information for not included packets and zero length packets. Packets of each tile make the corresponding tile stream and the collection of tile streams forms the final code stream.

3. THE PROPOSED METHOD

The most complex decoding stages of a JPEG2000 coded image are the entropy decoding and the inverse discrete wavelet transform. Our proposed retrieval technique
requires none of these complicated processes for indexing the JPEG2000 coded images. In the first step of retrieval process, the packet headers are decoded to extract the required features for indexing. Then, the extracted features for the coded images are used for calculating the similarity metrics. The calculated similarity values provide the necessary information for comparing images. In the following subsections various stages of the proposed method are described in more detail.

3.1. Features Extraction

The main problem with the low computational cost compressed domain JPEG2000 indexing methods such as [5, 6] is that they do not use the available information in the packet header efficiently and completely. In the proposed indexing method in this paper we use full packet header information to make three indexing parameters. For the first indexing parameter the maximum number of bit-planes in a code block is used to make the maximum bit histogram at each sub-band. In a lossy coding the number of coding passes of a code block that are included in the coded image represents the code block importance in the coded image. Hence we define another indexing parameter named importance histogram as the second feature. We take the achieved compression rate of each sub-band, as the third feature. All of these features can be extracted directly from the code block information embedded at the packet header. The detailed procedure for calculating these parameters is as follows.

3.1.1 Maximum Bit Histogram

In this histogram the maximum number of bit-planes in a code block is derived. It is defined as:

\[ BP = MBP - ZBP \] (1)

where \( MBP \) is the maximum number of possible bit-planes in a sub-band and is saved in the QCD and QCC marker segments of the JPEG2000 coded image bit stream [1]. \( ZBP \) is the number of leading zero bit-planes of a code block and is saved in the corresponding packet header using Tag Tree structure. In order to take into account the code block size for this feature we consider \( BP \) as a representative for each coefficient in the code block and make a histogram for the maximum bit-planes in each sub-band. The maximum bit histogram for a sub-band is defined as:

\[ HMB_{sb}[BP] = \frac{1}{S_{sb}} \sum_{j=1}^{N_{cb}} K.Scb_j \] (2)

where \( HMB_{sb}[BP] \) is the normalized histogram value for the maximum number of bit-planes equal to \( BP \). \( S_{sb} \) and \( Scb_j \) are respectively the number of coefficients in sub-band \( sb \) and the number of coefficients in code block \( j \) of this sub-band.

\( N_{cb} \) is the total number of code blocks in sub-band \( sb \). \( K \) is equal to 1 when the maximum number of bit-planes in code block \( j \) is equal to \( BP \) and is 0 otherwise. \( BP \) ranges from zero to the maximum number of possible bit-planes (\( MBP \)) in a sub-band. It is worth noting that in the lossy coding and even in the lossless coding there might be some code blocks which include only zero value coefficients. These bit-planes are not coded and will be signaled as zero code blocks. The \( BP \) value in the maximum bit histogram for such code blocks is set to zero.

3.1.2. Importance Histogram

In lossy compression some or all of coding passes of a code block may be included in the final coded image bit stream using the rate-distortion optimization techniques. Hence, the number of coding passes of a code block that are included in a coded image represents the importance of that code block in the quality of the coded image.

We denote the number of coding passes in a code block by \( CP \); this number is coded in the packet header as a particular codeword segment [1]. Using this number the importance histogram for a sub-band is calculated as:

\[ HC_{sb}[CP] = \frac{1}{S_{sb}} \sum_{j=1}^{N_{cb}} K.Scb_j \] (3)

where \( HC_{sb}[CP] \) is the normalized histogram value for the number of coding passes equal to \( CP \), which we refer to it as importance histogram. \( S_{sb} \), \( Scb_j \) and \( N_{cb} \) are the same parameters as Eq. (2). \( K \) is equal to 1 when the number of coding passes for code block \( j \) is equal to \( CP \) and is 0 otherwise. \( CP \) ranges from zero to the maximum number of possible coding passes (\( MBP \times 3-2 \)) in a sub-band.

3.1.3 Compression Rate Vector

We can define the compression rate factor for a sub-band in the same way as the compression rate is defined for a compressed image. Compression rate factor for each sub-band is defined as the division of the sub-band bit stream length to the total number of sub-band coefficients. The sub-band bit stream length (in byte) is the sum of the code block bit streams length in the sub-band and the code block bit stream length can be decoded from the corresponding packet header. We denote the \( j^{th} \) code block bit stream length in sub-band \( sb \) as \( Li_{sb} \). For each sub-band \( sb \) the compression rate can be defined as:

\[ CR_{sb} = \frac{\sum_{i=1}^{N_{cb}} Li_{sb}}{S_{sb}} \] (4)

where \( CR_{sb} \) is the compression rate of the sub-band \( sb \) and \( S_{sb} \) is the total number of coefficients in the sub-band \( sb \). In
the compression rate vector we use the normalized compression rate, which is defined as:

\[ NCR_{sb} = \frac{CR_{sb}}{\sum_{i=1}^{N_{sb}} CR_i} \]  

(5)

where \( NCR_{sb} \) is the normalized compression rate of the sub-band \( sb \) and \( N_{sb} \) is the total number of sub-bands in the compressed image. The sub-band compression rate vector \( (VCR) \) has \( N_{sb} \) elements and the \( i^{th} \) element is the \( NCR \) of sub-band \( i \).

3.2. Similarity Metrics

Similarity metrics are used as a measure for comparing the feature vectors of the images. We define the following similarity metric to measure the similarity between two images based on the feature vector presented as histogram \( L \).

\[ S_k = \frac{\left( \sum_{sb=1}^{N_k} \sum_{i=1}^{N_{sb}-1} \min(HK_{sb}[i],HK'[sb][i]) \right)}{N_k} \]  

(6)

where \( HK \) and \( HK' \) are corresponding histograms in two images, \( N_k \) is the number of sub-bands in the images and \( N_L \) is the total number of bins in the histogram. This similarity metric is used for the maximum bit and the importance histograms.

For the third feature vector, which is in the form of a vector the similarity metric is calculated as:

\[ S_v = \sum_{sb=1}^{N_k} \min(VCR(sb),VCR(sb)) \]  

(7)

To calculate the final similarity between two images \( (S) \) we use Eq. (8) to combine the three similarity measures of maximum bit histogram \( (S_M) \), importance histogram \( (S_C) \) and compression rate vector \( (S_i) \) as the overall similarity measure between two compressed images:

\[ S = \frac{w_m S_m + w_c S_c + w_v S_v}{w_m + w_c + w_v} \]  

(8)

where \( S \) is the overall similarity value and \( w_m, w_c \) and \( w_v \) are the weighting factors and in our simulations they are equally weighted, i.e. \( w_m=w_c=w_v=1 \). It is worth noting that in color images there are three overall similarity values \( (S) \) each for one of the color components and we use the average of them as the ultimate measure for the similarity of two color images.

4. EXPERIMENTAL EVALUATIONS

We used the query by example method to compare the performance of the proposed compressed domain image retrieval method with the pixel-based Gabor image retrieval method. In these experiments we used VisTex [9] image database, which includes 227 color images of the size 512x512. All the images in the database were coded with the Kakadu JPEG2000 encoder [10]. The coding was lossless including 6 resolution levels with a code block size of 64×64. The proposed feature vectors in Sec. 3.1 were calculated directly from compressed images.

In order to evaluate the performance of the proposed technique in a query by example image retrieval method, we used a sample query image set of 20 randomly selected images from the VisTex database. The similarity of all the database images with a query image was calculated using Eq. (8). Then six images with the highest similarity to the query image were retrieved from the database. Fig. 3 shows the ranked ordered six retrieved images for three sample query images. As expected, the first retrieved image in all cases is the query image. We also implemented the pixel-based Gabor filter [11] of 13×13 mask size, with 6 orientations and 4 scales. These parameters are the optimal parameter set for the Gabor filter [12]. Hence, for each image there will be 4x6=24 output filtered images and average \( (\mu_{mn}) \) and standard deviation \( (\sigma_{mn}) \) of these images make a feature vector of 48 elements. We performed the retrieval using this Gabor filter method on the same query set. For each query image we also found relevant images by inspecting the whole VisTex database. There were at most six and at least 2 relevant images for each image in the sample query image set. Each of the retrieved images, by the three retrieval methods marked as relevant or irrelevant using the images found in this step. The Precision (9) and Recall (10) parameters [13] were used for evaluating the performance of the indexing methods:

\[ \text{Precision} = \frac{\text{No. of retrieved relevant images}}{\text{No. of retrieved images}} \]  

(9)

\[ \text{Recall} = \frac{\text{No. of retrieved relevant images}}{\text{Total No. of retrieved images}} \]  

(10)

We used the marked images as relevant in the previous step to calculate the Precision and Recall values for the retrieved images for each query image using the two realized image retrieval methods. In this way, we calculated the Precision-Recall graphs for the first six retrieved images of each query and then by averaging [13] all the graphs, the averaged Precision-Recall graph (Fig. 4) is used to evaluate the performance of the image retrieval methods. The ideal Precision-Recall graph for retrieval of relevant images which is produced by inspecting the database is also included in Fig. 4. In the Precision-Recall for the proposed method the precision is very high (near to one) at the beginning of retrieved list, which means the most relevant images are retrieved first. Moreover, the recall value at the end of the graph is near to one implying that, most of the relevant images are in the retrieved lists. On the other hand,
the Precision-Recall graph of the proposed method is higher than the Precision-Recall graph of the Gabor filter; hence the proposed method has a superior performance to the Gabor filter. In addition to the superior performance, the proposed method has two other main advantages over to the pixel-based Gabor method. First, the proposed retrieval method has lower computational cost and second, it is in compressed domain.

We have also conducted experiments to compare the performance of our proposed compressed domain image retrieval method with the compressed domain image retrieval method of [6]. The method in [6] (referred as PH hereafter) uses the mean and variance of the maximum number of available bit-planes of the code blocks (BP) in a sub-band as the feature vector. Since, the VisTex database includes a small collection of images; we used a larger database with general images for this part of simulations. In these simulations we used a collection of more than 1300 color images with 768×512 resolution from the Benchathlon image database [14]. All the images in this collection were coded as JPEG2000 images using Kakadu software. We used lossless coding with six resolution levels and the code block size of 64×64 as the coding parameters. Then the features of all images were extracted based on the proposed method and the PH method. We made a sample query image set, including 25 randomly selected images from the new database. We also marked the relevant images to each query image, by inspecting the whole database. There were at least two and at most seven relevant images for the images in the query set. The similarity between each query set image and all of the database images have been calculated in the same way as the pervious experiment and the ranked ordered list of the first seven retrieved images for each query was constructed. Fig. 5 shows the first seven retrieved images list of three sample query images of the new database using the proposed method.

Fig. 6 shows the average Precision-Recall graphs of the first seven retrieved images for the query images set using the proposed method and the PH method. These graphs indicate that the proposed method provides a better performance than the PH method on lossless compressed images.

To compare the performance of the proposed method with the PH method in various compression rates, we used the Kakadu JPEG2000 encoder to encode the images in the database and also the query images set with four different compression rates of 4 b/p, 2 b/p, 1 b/p and 0.5 b/p. The other coding parameters were the same as the lossless case. We used the coded images in each bit rate to derive the indexing parameters based on the proposed method and PH. The retrieved images for the query images sets were found in different bit rates using the proposed image retrieval method and the PH method. The resulted Precision-Recall graphs are shown in Fig. 6.
The top seven retrieved images for three query images in Benchathlon image database. The first image in each row is query image.

Precision-Recall graphs using the proposed method and the PH indexing method in the Benchathlon lossless coded images. These graphs indicate that the proposed method has acceptable performance even at highly compressed images (0.5 b/p) and outperforms the PH method in all the compression rates.

5. CONCLUSION

In this work, a compressed domain image indexing technique is proposed for JPEG2000 coded images. The proposed method decodes only packet header information for indexing and avoids the time consuming stages of the arithmetic decoding and the inverse wavelet transform. Hence, it is very fast and desirable for large databases. Simulation results indicate that the proposed compressed domain indexing method outperforms the pixel-based Gabor filter indexing method. Simulation results indicate that the proposed method has good performance in the highly compressed databases, where the other compressed domain JPEG2000 image indexing methods such as [5,6] do not provide satisfactory results. This is due to the fact that the proposed method employs full packet header information; hence, it has adequate information for image retrieval in the highly compressed images, which have a large number of zero packets. Thus, the proposed method is an efficient indexing method for compressed JPEG2000 images even at low bit rates.

REFERENCES


Fig. 7. Precision-Recall graphs for the proposed method and the PH method in the Benchathlon coded images at four different compression bit rates: a) 4 b/p b) 2 b/p c) 1 b/p d) 0.5 b/p


