ABSTRACT

The proposed scheme analyzes the image content and then determines the probable ROI masks by examining the significant states of high frequency subbands generated from embedded block coding with optimized truncation (EBCOT). Additionally, probable ROI masks are constructed in all bit planes of subbands by categorizing subblocks as either interesting or uninteresting, smoothing subblocks of interest, and grouping these subblocks based on an or no initial point. The rate-distortion (RD) pairs corresponding to all probable ROI masks are then estimated from the RD distribution during the Tier-2 coding process of EBCOT [2]. Based on these estimations, the Lagrangian multiplier method is employed in the RD function to obtain the optimized ROI mask from the probable masks by minimizing the distortion of the ROI-encoded image at a given bitrate constraint. ROI-encoded images obtained using the proposed scheme outperform ROI-encoded images obtained via the conventional schemes using fixed-square and object-segmentation masks, as judged by subjective visual perception and objective measurement in terms of peak signal-to-noise ratio. Particularly, the proposed scheme can easily adapt the ROI region with varied sizes and shapes according to the bit-rate constraint whereas the conventional schemes only adopt the fixed-square region and fixed segmented objects. Furthermore, when the proposed scheme is applied to motion JPEG 2000 for video compression, the centroid of the ROI mask in the previous frame can be used as an initial point for merging the subblocks of interest in the current frame to track the ROI masks in a video sequence. Therefore, the proposed scheme can easily be employed to improve the perceptual and objective performance in the ROI coding associated with JPEG 2000 and motion JPEG 2000.

Keywords: subblocks, Lagrangian Multiplier

I. INTRODUCTION

JPEG 2000 is superior to conventional JPEG in having higher compression ratio, embedded bit stream, multiple resolution representations, error resilience, and region of interest (ROI) coding[1]. Among these features, the ROI coding can separate an image into a background region and a region of interest. Two ROI coding methods, scaling-based and Maxshift, are supported in part 1 of JPEG 2000. The scaling-based method has the advantage of allowing partial coding of the background region prior to the coding of the entire ROI, but it must transmit the side information of ROI at an additional coding cost. In the Maxshift method, the ROI bit stream is arranged in front of the background bit stream, so that the bit stream does not need to transmit additional side information to locate the ROI. Bit rates are first applied to encode the region of interest, and the residual bit rates are then used to encode the background region. Consequently, the perceptual quality of the ROI of the image reconstructed with ROI coding is apparently better than that of the image reconstructed without ROI coding, particularly at low bit rates. Conventional studies of ROI coding help to yield ROI-encoded images with good coding efficiency, by carefully shifting the bit planes of ROI and coding the side information. Subedar et al. encoded the side information of ROI by differentially chain coding the ROI boundaries, and then adopted the scaling-based method of JPEG 2000 to transmit the shape information before the texture information of ROI [4], to ensure good coding. These schemes can have the advantage of applying ROI masks in the coding processes that are implemented by fixed hardware platforms but their ROI masks must be either specified by a user or identified by an additional process. Therefore, an innovative ROI scheme capable of automatically determining ROIs of various images with diverse textures is critically demanded to effectively integrate the ROI mask determination and the coding process[3].
II GENERATION OF PROBABLE ROI MOLSKS

The block diagram of the proposed automatically-determined ROI scheme embedded in JPEG 2000 is shown in Fig. 1. In JPEG 2000, an image first undergoes pre-processing for tilling and color transformation. Next, the discrete wavelet transform is performed for every tile image and its transformed coefficients are quantized. The EBCOT process includes the pass scanning (Tier-1 coding) and adaptive binary arithmetic coding (Tier-2 coding), to encode the quantized coefficients of codeblocks [5].

When EBCOT manipulates the pass scanning of each bit plane in a codeblock, a significant state is produced, representing the context characteristics of the coefficients on the bit plane. After the pass scanning has been conducted in a codeblock [9], the significant states are produced in a bit-plane order from the most significant bit (MSB) to the least significant bit (LSB). If a coefficient is already nonzero in the earlier bit planes, then its significant state is “1”; otherwise, it is “0”. Consequently, the proposed automatically-determined ROI scheme embedded in JPEG 2000, as shown in Fig. 1, is to explore the significant states of the bit planes, which are produced through the EBCOT process.
III ROI Mask Determination

The flowchart of the proposed scheme, embedded in the EBCOT process. The determination of ROI involves initially analyzing the significant states of the bit planes of highfrequency subbands at all decomposition levels. Probable ROI masks are then determined from the subblocks of interest after smoothing, eliminating and grouping. Then, RD pairs associated with probable ROI masks are estimated and examined to determine the optimized ROI mask that meets the bit-rate criterion with the least distortion [7].

IV EXTRACTING MASKS FROM SIGNIFICANT STATES

In the DWT process, a Low-Low (LL) subband image generally reflects the low-frequency content of an image with highly compact energy. The high-frequency content of an image along the horizontal, vertical and diagonal are expressed in High-Low (HL), Low-High (LH), and High-High (HH) subband images at each decomposition level, respectively, which also indicate edge or textural information. Based on this concept, the proposed scheme utilizes the significant states of bit planes in the HL, LH, and HH subbands at all decomposition levels, to determine the probable ROI masks in a W*W-pixel subblock of interest; when uninteresting where variations of image contents. Moreover, the subblock size for significant tables at the decomposition levels.

The significant states of the LH, HL, and HH subbands, which are in the same bit plane at the same decomposition level, are then integrated as a significant table using an “OR” operation. Each of these newly formed significant tables corresponds to the bth bit plane at the lth decomposition level, as stated by (2)

Equations (1) presents the significant state of each bit plane in the HH subband at a decomposition level. \( S_{b,j}^{HH}(x,y) \) represents the significant state of the coefficient at position, from the bth bit plane of the HH subband at the lth decomposition level after the pass scanning. The distribution of the significant states of LH, HL, and HH subbands shows that the coefficients are significant along the edges of the object in the image. Consequently, the significant states in each bit plane are used to obtain edge and textural information of high-frequency regions and thus determine the regions of interest in the image [10].

The significant states of the LL, HL, and HH subbands, which are in the same bit plane at the same decomposition level, are then integrated as a significant table using an “OR” operation. Each of these newly formed significant tables corresponds to the bth bit plane at the lth decomposition level, as stated by (2)

Accordingly, \( \bar{S}_{b,j}(x,y) \) denotes the significant table associated with a high-frequency edge and texture variation from the bth bit plane on three subbands at the same decomposition level. The significant table that corresponds to a bit plane at the first level is divided into many \( w*w \)-pixel subblocks to examine texture variations of image contents. Moreover, the subblock size for significant tables at the lth decomposition level is set to \( (w/2^{l-1})^2 \times (w/2^{l-1})) \)-pixel with \( l=1,2,...,L \) to ensure consistent classification resolution among different decomposition levels.

V CLASSIFYING SUBBLOCKS TO INTERESTED OR NON INTERESTED ONES

\[
T_{b,l,p}(i,j) = \begin{cases} 
1, & \text{if } \sum_{x=(w/2^{l-1})i}^{((w/2^{l-1})(i+1)-1)} \sum_{y=(w/2^{l-1})j}^{((w/2^{l-1})(j+1)-1)} \bar{S}_{b,j}(x,y) > p \\
0, & \text{otherwise}
\end{cases}
\]

The above Equation determines that a subblock is classified as one of interest or one that is uninteresting where p ranges from 0 to \( (w/2^{l-1})^2 - 1 \). When \( T_{b,l,p}(i,j) \) is unity, this subblock is regarded as a subblock of interest; when \( T_{b,l,p}(i,j) \) is zero, it is a background subblock. The range of \( i \) and \( j \) is from zero to
\( \left( \frac{W}{2w} \right)^2 - 1 \) for denoting the dimension size of a significant table at a decomposition level, and \( p \) represents the number of significant bits for subblock classification. Herein, \( p \) is not a fixed threshold and is varied from 0 to \( \left( \frac{W}{2^{l-1}} \right)^2 - 1 \), according to the subblock size and the decomposition level for classifying significant tables. If a subblock at the \( b \)th bit plane from the significant table at the \( l \)th decomposition level has over \( p \) high-frequency coefficients that exceed \( 2^{b-1} \), then it is classified as a subblock of interest. Such classification in a subblock is performed for all cases on each significant table, in which the size of a subblock is \( \left( \frac{w}{2^{l-1}} \right)^2 \) pixel to obtain consistent classification resolution at different decomposition levels. Consequently, when subband coefficients from DWT at all decomposition levels are represented by bit planes, the classification could generate \( \sum_{i=1}^{L} \left( \frac{w}{2^{l-1}} \right)^2 \times r_i \) possible results.

VI GROUPING THE SUBBLOCKS

In viewing an image, the interested regions are preferably grouped, rather than dispersed throughout the image frame from perceptual point of view. On viewing the two images (image frame coded with grouped ROI and image with scattered ROI) that had been processed under the same bit-rate criterion reveals that the image with grouped ROI clearly has a better perceptual quality. According, not only must an image be partitioned into an ROI and a background region, to ensure the satisfactory objective picture quality of the ROI-encoded image, but also such an image must be presented with a better subjective perceptual quality[11],[12].

VII CONSTRAINTED RUN–LENGTH ALGORITHM

The subblocks in an object of interest are not all grouped together, but some are scattered by the noisy subblocks. Each initial classified result must first be smoothed using constrained run-length algorithm (CRLA) to group the ROIs, and to eliminate such noisy subblocks. Here, CRLA in two operating modes is employed to connect scattered regions and eliminate noises in the initial classified results. For each initial classified result, CRLA is performed six times in the following steps where three constant constraints, \( c_1 \), \( c_2 \) and \( c_3 \), are utilized[13].

1) CRLA is performed on the grouping mode to the initial classified results row-by-row with the specified constraining constant, \( c_1 \), for row smoothing.
2) CRLA is performed on the grouping mode to the result of step (1) column-by-column using the specified constraining constant, \( c_2 \), for column smoothing.
3) The result from step (2) is operated by CRLA at the grouping mode in a row-by-row manner once again with the specified constraining constant, \( c_3 \), for row smoothing.
4) Perform CRLA at the eliminating mode to the results from step (3) row-by-row with the specified constraining constant, \( c_1 \), for row elimination.
5) Perform CRLA at the eliminating mode to the result of step (4) column-by-column using the specified constraining constant, \( c_2 \), for column elimination.
6) Finally, the result from step (5) is operated by CRLA at the eliminating mode row-by-row again with the specified constraining constant, \( c_3 \), for row elimination.

Steps (1), (2), and (3) are applied to group subblocks of interest and fill the gaps among texture or edge subblocks, by counting the subblocks that are not of interest between two neighboring subblocks of interest row-by-row, column-by-column and row-by-row, respectively. If the counted number does not exceed the specified constraining constant \( c_1 \), \( c_2 \) or \( c_3 \), the subblocks that are not of interest, but which are between subblocks of interest, are reset to be “of interest”. Additionally, Steps (4), (5), and (6) are employed by computing the number of subblocks of interest between two neighboring subblocks that are not of interest, to eliminate noisy subblocks from the background. If this number does not exceed the constraint \( c_1 \), \( c_2 \) or \( c_3 \), then these subblocks of interest are reset to “not of interest”. CRLAs are adopted herein to eliminate the noisy subblocks that are scattered on the background. Notably, the constraining constants for performing CRLAs must be appropriately chosen to yield reasonable regions of interest.
When the subblock is smaller, the classifications provide more textural characteristics of the image. Accordingly, the constraints are determined in relation to the subblock sizes. If larger constraints are applied to smooth the initial classified results for a given subblock size, then ROIs become more grouped. High constraints cause regions to be too grouped, whereas small constraint constants disperse the regions of interest. Additionally, when the subblock is smaller, the constraints must be increased to smooth noisy subblocks effectively. However, the sizes and content of objects in an image also influence the determination of the subblock size. Therefore, in capturing the detailed content of an image, the subblock size and the constraining constants should be moderately small. ROIs are located where the gray-level is “255”, and regions of no interest are where the gray-level is “0”. After CRLAs are applied, ROIs are grouped much more closely and the noisy subblocks scattered in the background are all eliminated to generate a grouped ROI mask. Notably, CRLA is applied individually to the classification results of each bit plane at each decomposition level. The probable ROI masks are generated and then undergo RD optimization to yield an adequate ROI at a given bit-rate constraint.

VII BIT STREAM FORMATION

After the compressed bits for each code-block are generated by Tier-1 coding, the Tier-2 coding engine efficiently represents the layer and block summary information for each code-block. A layer consists of consecutive bit-plane coding passes from each code-block in a tile, including all the sub bands of all the components in the tile. The block summary information consists of length of compressed code words of the code block, the most significant magnitude bit-plane at which any sample in the code-block is nonzero, as well as the truncation point between the bit stream layers, among others. The decoder receives this information in an encoded manner in the form of two tag trees[14]. This encoding helps to represent this information in a very compact form without incurring too much overhead in the final compressed file. The encoding process is popularly known as Tag Tree coding.

VIII CALCULATION OF PSNR AND DISTORTION

The peak signal-to-noise ratio, in decibels, between two images. This ratio is often used as a quality measurement between the original and a compressed image. The higher the PSNR, the better the quality of the compressed, or reconstructed image. The Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR) are the two error metrics used to compare image compression quality[15]. The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. The lower the value of MSE, the lower the error. To compute the PSNR, the block first calculates the mean-squared error using the following equation:

\[
MSE = \frac{\sum_{M,N} (I_1 - I_2)^2}{M \times N}
\]

In the previous equation, \(M\) and \(N\) are the number of rows and columns in the input images, respectively. Then the block computes the PSNR using the following equation:

\[
PSNR = 10 \log_{10} \left( \frac{R^2}{MSE} \right)
\]

In the previous equation, \(R\) is the maximum fluctuation in the input image data type. For example, if the input image has a double-precision floating-point data type, then \(R\) is 1. If it has an 8-bit unsigned integer data type, \(R\) is 255.
VII SIMULATION RESULTS

Fig3.1 Original image
This is the plane image of size used in our project for illustration. The size of the image is 512*512. In this image, no pre-processing is done as the image is of medium size and contains only one component (Gray scale).

Fig: 3.2 DWT of the image
The above image shows the various sub-bands obtained from the 2-D wavelet transform. The wavelet transform used here is “bior3.7”

Fig: 3.3 Quantized image
The quantized image is shown in the figure. It has the mean value rounded of around zero.
Fig 3.4 Bit planes of HL sub-band
The figure shows the various bit-planes of the HL sub-band. These bit-planes undergo the significant propagation pass of the Tier-1 coding of EBCOT algorithm to generate significant bits

Fig 3.5 Significant states of various Sub-Bands
This figure shows the significant states of bit-planes after the SPP. Significant bits are obtained from all the sub-bands except LL. The white pixels shows the significant bits

Fig 3.6 Results of CRLA
The figure shows the probable ROI masks after the CRLA process. These masks are analyzed to get the desired ROI mask
Fig: 3.7 Generated ROI Mask

The figure shows the generated ROI mask after analyzing the bit-rate and distortion. The white or 1 regions are the region of interest and dark regions is background region.

Fig: 6.8 Reconstructed image using circular ROI at 0.1bpp

This is the reconstructed image using a circular ROI at a low bit-rate of .1 bpp. In the image the background information are completely lost and also we can clearly witness the artifacts in the reconstructed image. The visual quality is so much reduced such that the plane is not at all distinguishable.

Fig: 6.9 Reconstructed image using rectangular ROI at 0.1bpp

This image is the reconstructed version of the plane image compressed with a rectangular mask at a bit-rate of .1 bpp. In this image also the background information are completely lost and also large artifacts are visible in the ROI region also. This shows the inability of the other types of ROI schemes failing to produce good results at low bit-rates.
This is the reconstructed version of the plane compressed with our proposed mask. In this image the background details are clearly visible as not in the case of above said techniques. We should note that there exist no artifacts in the ROI region and also a very minimal artifacts and loss of information in the background at an very lower bit-rate of .1 bpp. This shows the ability of the proposed scheme to produce ROI masks that provide not only good quality of ROI but also background with minimal distortion and higher PSNR values. From the above facts we can say that our scheme outperforms all the other ROI schemes.

XI CONCLUSION AND DISCUSSION

The above graph shows the plot of PSNR Vs bit-rate of the various ROI schemes. The graph clearly illustrates that at the very low bitrates <0.5 bpp, the proposed scheme offers a good PSNR value than the other ROI masks generated by the fixed ROI schemes. proposed scheme .Whereas uniform number of bits is used to code the image in the non-ROI coded image.

This is obvious from the table 9.1 which shows the total percentage of bits that are used to code the image is used for the ROI alone.
When the more number of bits are used to code the ROI, the better is the quality of the image. In this case, even though the PSNR value of the Non-ROI coded image is higher than that of the ROI coded image, the visual quality of the image will be better for the same rate and compression ratios. The same case is illustrated in the MSE Vs bit-rate graph as well as the comparison table.

<table>
<thead>
<tr>
<th>No. of bits used to encode the image (Non-ROI)</th>
<th>No. of bits used to encode the ROI*</th>
<th>No. of bits used to encode the image (Proposed ROI)</th>
<th>No. of bits used to encode the ROI in proposed scheme</th>
<th>% of bits used to encode the ROI in proposed scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>79397</td>
<td>45.1200</td>
<td>81920</td>
<td>65503</td>
<td>79.95972</td>
</tr>
<tr>
<td>65365</td>
<td>45.1200</td>
<td>65530</td>
<td>65503</td>
<td>99.9588</td>
</tr>
<tr>
<td>52227</td>
<td>45.1200</td>
<td>56319</td>
<td>56304</td>
<td>99.97337</td>
</tr>
<tr>
<td>39311</td>
<td>45.1200</td>
<td>39219</td>
<td>39217</td>
<td>99.9949</td>
</tr>
<tr>
<td>26178</td>
<td>45.1200</td>
<td>26099</td>
<td>26097</td>
<td>99.99234</td>
</tr>
<tr>
<td>13079</td>
<td>45.1200</td>
<td>13250</td>
<td>13236</td>
<td>99.89434</td>
</tr>
<tr>
<td>6517</td>
<td>45.1200</td>
<td>6484</td>
<td>6482</td>
<td>99.96915</td>
</tr>
<tr>
<td>3266</td>
<td>45.1200</td>
<td>3231</td>
<td>3229</td>
<td>99.9381</td>
</tr>
</tbody>
</table>

*Assuming that 45% of the image is ROI

REFERENCES


