Bitplane-by-Bitplane Shift (BbBShift)—A Suggestion for JPEG2000 Region of Interest Image Coding

Zhou Wang, *Student Member, IEEE*, and Alan C. Bovik, *Fellow, IEEE*

***Abstract*—The JPEG2000 image coding standard defines two kinds of region of interest (ROI) coding methods—the general scaling based method and the maximum shift (maxshift) method. The former requires shape coding of the ROIs, which leads to increased complexity of codec implementations and limits the choice of ROI shapes (currently, only rectangle and ellipse shapes are defined). The latter allows for arbitrarily shaped ROI coding without explicitly transmitting any shape information to the decoder, but does not have the flexibility to select an arbitrary scaling value to define the relative importance of the ROI and the background wavelet coefficients. We propose a bitplane-by-bitplane shift (BbBShift) method, which supports both arbitrary ROI shape and arbitrary scaling without shape coding.**

***Index Terms*—Image coding, JPEG2000, region of interest (ROI), bitplane coding, wavelet coding, maximum shift (maxshift), bitplane-by-bitplane shift (BbBShift).**

I. Problems With JPEG2000 ROI Coding

Region of interest (ROI) image coding allows for encoding the ROIs in an image with better quality than the background (BG). ROI coding is one of the requirements in the new JPEG2000 image coding standard [1]–[3]. Two kinds of ROI coding methods are defined—the general scaling based method and the maximum shift (maxshift) method [3]–[6].

In the general scaling based method, the wavelet transform is applied to the image at the encoder and the resulting coefficients not associated with the ROI are scaled down (shifted down) so that the ROI-associated bits are placed in higher bitplanes. During the embedded bitplane coding process, the bits in the higher bitplanes are placed before those in the lower bitplanes. The scaling value and the shape information of the ROIs are also added into the encoded bitstream. At the decoder, the bitplanes are reconstructed and the non-ROI coefficients are scaled up to their original bitplanes before the inverse wavelet transform is applied. If the encoded bitstream is truncated or the encoding/decoding process is terminated before the image is fully encoded/decoded, the ROIs will have a higher quality than the BG. The relative importance of the ROIs and the BG is determined by the scaling value s, which defines the number of bitplanes to be shifted. Fig. 1(a) and (b) illustrates how the bitplanes are shifted in the general scaling based method.

There are three major drawbacks of the general scaling based method shown in Fig. 1(b). First, it is not convenient to deal with different wavelet subbands in different ways, which is sometimes desired by the users. For example, the users may like to treat all the coefficients equally at some low frequency subbands and differently (better quality at ROIs than BG) at the high frequency subbands. Second, it needs to encode and transmit the shape information of the ROIs. This significantly increases the complexity of encoder/decoder implementations. Third, if arbitrary ROI shapes are desired, then shape coding will consume a large number of bits, which significantly decreases the overall coding efficiency. The current standard attempts to avoid this problem and only defines rectangle and ellipse shaped ROIs [2], which can be coded with a small number of bits. However, this limits the application scope of ROI coding because in real-world applications, ROIs are usually associated with certain objects in the image, which generally have arbitrary shapes.

A very effective solution, the maxshift method [1], [3], [7], [8], was proposed for JPEG2000, which does not require any shape coding or any shape information to be explicitly transmitted to the decoder. In the maxshift method, the scaling value, s, must be chosen to satisfy [1]:

$$s \geq \max(M_b), \tag{1}$$

where $\max(M_b)$ is the largest number of magnitude bitplanes for any ROI coefficient. After scaling, all significant bits associated with the ROI will be in higher bitplanes than all the significant bits associated with the background [1]. Fig. 1(c) demonstrates the bitplane shift in the maxshift method. At the decoder, the nonzero ROI and BG coefficients can be identified simply by looking at the coefficients' magnitudes. All coefficients that are found to be lower than the sth bitplane are known to belong to the BG. There is no need to tell the decoder explicitly about the shape information of the ROIs. The BG coefficients are scaled up by s bitplanes before the inverse wavelet transform is applied. With the maxshift method, it is also very easy to treat different wavelet subbands differently. For example, the encoder can include entire low-frequency subbands in the ROI mask and encode a uniform low-resolution version of the image at an early stage of the encoded bitstream. The distinction between ROI and BG is made only at high frequency subbands.

The major limitation of the maxshift method is that it does not have the flexibility to allow for an arbitrary scaling value to de-

Manuscript received January 25, 2002; revised April 9, 2002. This work was supported in part by Texas Instruments, Inc., and by the State of Texas Advanced Technology Program. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Antonio Ortega.

The authors are with the Laboratory for Image and Video Engineering (LIVE), Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712 (e-mail: zhouwang@ieee.org; bovik@ece.utexas.edu).

Publisher Item Identifier S 1070-9908(02)06043-1.

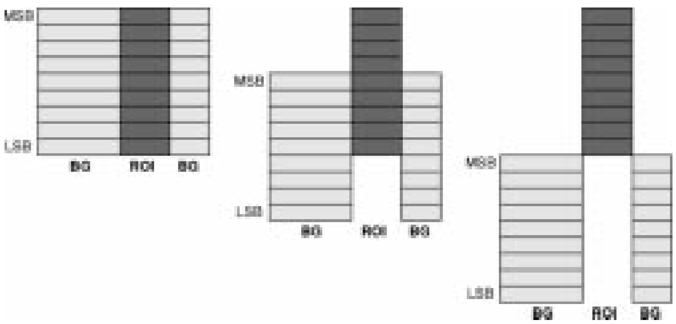

Fig. 1. ROI coding methods in JPEG2000 (bitplanes are represented by the gray bars). (a) No ROI coding, no scaling; (b) scaling based method, $s = 4$; and (c) maxshift method, $s = 9$.

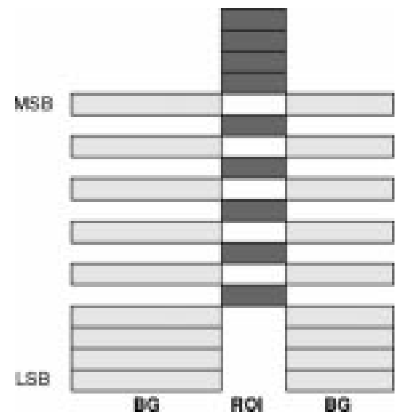

Fig. 2. BbBShift methods with $s_1 = 4$ and $s_2 = 5$ (Bitplanes are represented by the gray bars).

fine the relative importance of the ROI and the BG wavelet coefficients as in the general scaling based method. This means that in all the subbands, where the ROI/BG distinction is applied, no information about the non-ROI coefficients can be received until every detail of the ROI coefficients has been fully decoded, even if the detail is imperceptible random noise (which may happen in reversible coding mode or irreversible coding mode with very small quantization step size). Another limitation of the maxshift method is that when there are multiple ROIs in the same image, an ROI cannot have its own scaling value, and therefore different priority during encoding/transmission of the image.

II. BbBShift Method

A. BbBShift

We propose a bitplane-by-bitplane shift (BbBShift) method. Instead of shifting the bitplanes all at once by the same scaling value s as in maxshift, BbBShift shifts them on a bitplane-by-bitplane basis. An illustration of the BbBShift method is shown in Fig. 2. Two parameters, s_1 and s_2, are used in BbBShift. The sum of s_1 and s_2 must be equal to the largest number of magnitude bitplanes for any ROI coefficient. In this paper, we index the top bitplane as bitplane 1, the next to top as bitplane 2, and so on.

At the encoder, the bitplane shifting scheme is as follows:

1) For any bitplane b of an ROI coefficient:
   if $b \leq s_1$, no shift;
   if $s_1 < b \leq s_1 + s_2$, shift it down to bitplane $s_1 + 2(b - s_1)$.
2) For any bitplane b of a BG coefficient:
   if $b \leq s_2$, shift it down to bitplane $s_1 + 2b - 1$;
   if $b > s_2$, shift it down to bitplane $s_1 + s_2 + b$.

At the decoder, for any given nonzero wavelet coefficient, the first step is to identify whether it is an ROI coefficient or a BG coefficient. This can be done by examining the bitplane level of its most significant bit (MSB). The set of ROI associated bitplanes is given by

$$B_{\mathrm{ROI}} = \{b \,|\, b \leq s_1 \text{ or } b = s_1 + 2k, k = 1, 2, \ldots, s_2\}. \quad (2)$$

If the wavelet coefficient's MSB is at bitplane $b \in B_{\mathrm{ROI}}$, then it must be an ROI coefficient. Otherwise, it is a BG coefficient. The bitplanes are then shifted back to their original levels by reversing the bitplane shifting scheme in the encoder.

B. Comparisons

1) Functionality: In comparison with the general scaling based methods defined in JPEG2000 Part II [2], where only rectangle and ellipse ROI shapes are allowed, the BbBShift method supports arbitrary shaped ROI coding. BbBShift also bears the same advantage of maxshift that different wavelet subbands can have different ROI definitions. In BbBShift, if we set $s_2 = 0$ and choose s_1 according to (1), then BbBShift is equivalent to maxshift. In other words, maxshift is a special case of BbBShift. Compared with maxshfit, BbBShift has the flexibility to have an arbitrary scaling value to adjust the relative importance between ROI and BG coefficients. This flexibility may lead to improved quality of ROI coding, depending on the application. For example, in certain applications, the user may choose an s_1 such that "high enough quality" ROI can be achieved with s_1 bitplanes of the ROI coefficients. After the "high enough quality" of ROI is reached, some significant bits of the BG coefficients begin to be encoded/transmitted, which are treated at higher priorities than the nonsignificant bits of the ROI coefficients. An example is given in Fig. 3, where the 24 bits/pixel (bpp) "Barbara" image is compressed reversibly and decompressed at 0.8 bpp using the maxshift $(s = 12)$ and the BbBShift $(s_1 = 6, s_2 = 6)$ methods, respectively. It can be observed that without visual difference at the ROI, BbBShift coded image provides better quality at the BG. The limitation of the BbBShift method is that if there are multiple ROIs in the same image, the scaling value must be the same for all of them. This limitation also applies to the maxshift method and is a disadvantage in comparison with the general scaling based method, where different ROIs can have different scaling values and therefore different priorities during the encoding/tranmission process.

2) Complexity: It is not necessary for the BbBShift method to have a shape coding component, which is essential in the general scaling based methods. The general scaling based methods also require a complex ROI mask generation procedure, which is different for different ROI shapes and significantly increases the computation and hardware/software implementation expenses. By contrast, the ROI/BG identification process in maxshift and BbBShift is much cheaper. Compared with the maxshift method, a more complicated procedure is needed

(a)

(b)

Fig. 3. The 24 bpp RGB "Barbara" image coded reversibly and decoded at 0.8 bpp using the maxshift method [(a) $s = 12$] and the BbBShift method [(b) $s_1 = 6, s_2 = 6$]. The ROI is at the face area and $1/16$ of the image size.

in the BbBShift method to shift the encoded/decoded bits to the right bitplanes. At the decoder side, this process can be improved by merging the bitplane decoding and the bitplane shifting procedures. In this improved implementation, instead of applying bitplane decoding and bitplane shift as two separate procedures in cascade, right after a bit is decoded, we can immediately determine its correct bitplane (according to the rule of bitplane shift reversing in BbBShift) and write it to its correct position.

3) Coding Efficiency: Similar to the general scaling based method and the maxshift method, the coding efficiency of BbBShift decreases in comparison with JPEG2000 without any ROI coding. The reason is that bitplane shifting increases the dynamic range (or number of bitplanes) of the wavelet coefficients being encoded. It is reported [7], [8] that for lossless coding of images with ROIs, the maxshift method increases the bit rate by 1–8%, compared to lossless coding of an image without ROI (and less compared to the general scaling based method, depending on the scaling value used). If the point of lossless coding is reached, the BbBShift and maxshift methods result in similar bit rates because they have the same number of bitplanes and the information to be coded in each biplane is exactly the same. The difference is that the bitplanes are placed in different order, which may have effect on the entropy coding module. Our current experimental results show that the effect is insignificant. For example, for lossless coding of the 24 bpp "Barbara" image with the ROI ($1/16$ of the image size) selection as in Fig. 3, BbBShift ($s_1 = 6, s_2 = 6$) and maxshift ($s = 12$) spends 12.88 bpp and 12.76 bpp, respectively. More experiments on sixteen 24 bpp RGB images (with size ranging from 512×512 to 800×600) show that BbBShift spends 1.00–3.75% (average 2.37%) less bits than maxshift for ROI size $1/4$ of the image size, and 0.49–1.47% (average 0.97%) more bits than maxshift for ROI size $1/16$ of the image size.

4) Compatibility: It needs to be mentioned that the proposed BbBShift method is not compatible with the current JPEG2000 ROI coding definitions, in which only maxshift and rectangle and ellipse shape scaling based ROI coding are defined. In order to use BbBShift, a new ROI coding mode must be added to the standard.

III. Conclusion

We propose a BbBShift method for JPEG2000 ROI coding and compare it with the current general scaling based method and maxshift method defined in the standard. We believe that BbBShift has many advantages and would complement the existing standard. Another contribution of this work is the idea of using the bitplane level of MSB to distinguish nonzero ROI and BG coefficients. This idea is valuable for future research on ROI image coding.

References

[1] *ISO/IEC JTC 1/SC 29/WG 1 (ITU-T SG8) JPEG 2000 Part I Final Committee Draft Version 1.0*, Mar. 2000.

[2] *ISO/IEC JTC 1/SC 29/WG 1 (ITU-T SG8) JPEG 2000 Part II Final Committee Draft*, Dec. 2000.

[3] C. Christopoulos, A. N. Skodras, and T. Ebrahimi, "JPEG 2000 still image coding system: An overview," *IEEE Trans. Consumer Electron.*, vol. 46, pp. 1103–1127, Nov. 2000.

[4] E. Atsumi and N. Farvardin, "Lossy/lossless region-of-interest image coding based on set partitioning in hierarchical trees," *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 87–91, Oct. 1998.

[5] D. Nister and C. Christopoulos, "Lossless region of interest with a naturally progressive still image coding algorithm," *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 856–860, Oct. 1998.

[6] ——, "Lossless region of interest coding," *Signal Process.*, vol. 78, pp. 1–17, Oct. 1999.

[7] C. Christopoulos, J. Askelf, and M. Larsson, "Efficient methods for encoding regions of interest in the upcoming JPEG2000 still image coding standard," *IEEE Signal Processing Lett.*, vol. 7, pp. 247–249, Sept. 2000.

[8] ——, "Efficient region of interest coding techniques in the upcoming JPEG2000 still image coding standard," *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 41–44, Sept. 2000.