GENERALIZED BITPLANE-BY-BITPLANE SHIFT METHOD FOR JPEG2000 ROI CODING

Zhou Wang, Serene Banerjee, Brian L. Evans and Alan C. Bovik

Laboratory for Image and Video Engineering (LIVE), Dept. of Electrical and Computer Engineering,
The University of Texas at Austin, Austin, TX 78712-1084, USA
Email: zhouwang@ieee.org, {serene, bevans, bovik}@ece.utexas.edu

ABSTRACT

One interesting feature of the new JPEG2000 image coding standard is support of region of interest (ROI) coding using the maximum shift (Maxshift) method, which allows for arbitrarily shaped ROI image compression without shape coding or explicitly transmitting any shape information to the decoder. The major disadvantage of the Maxshift method is that it cannot adjust the scaling value which determines the degree of relative importance between the ROI and the background wavelet coefficients. The bitplane-by-bitplane shift (BbBShift) method was introduced to support both arbitrary ROI shape and arbitrary scaling without shape coding. In this paper, we propose a generalize BbBShift (GBbBShift) method, which delivers much more flexibility than both Maxshift and BbBShift for "degree-of-interest" adjustment of the ROI with insignificant effect on coding efficiency and computational complexity. Experiments show that it can provide significantly better visual quality than Maxshift at low bit rates. GBbBShift is not compliant with the current JPEG2000 definitions. In order to use it, a new ROI coding mode would need to be added to the standard.

1. JPEG2000 ROI CODING

Region of interest (ROI) image coding allows for encoding the ROIs in an image with better quality than the background (BG). Two kinds of ROI coding methods are defined in the new JPEG2000 image coding standard [1]–[4] — the general scaling based method and the maximum shift (Maxshift) method.

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 associated 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 shows how the bitplanes are shifted in the general scaling based method.

There are three major drawbacks of the general scaling based method. First, it is not convenient to deal with different wavelet subbands in different ways, which is sometimes desired by the users. 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 many real-world applications, ROIs are usually associated with certain objects in the image, which generally have arbitrary shapes.

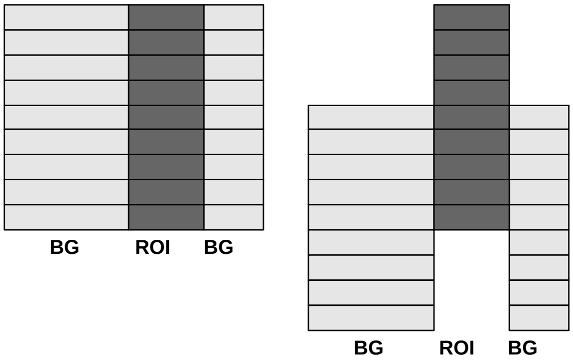
A very effective solution, the Maxshift method [1], [3]–[8], was proposed for JPEG2000, which does not require any shape coding or any shape information to be explicitly transmitted to the decoder. In Maxshift, the scaling value, *s*, must be chosen to satisfy $s \geq \max(M_b)$ [1], where $\max(M_b)$ is the largest number of magnitude bitplanes for any coefficient. After scaling, all significant bits associated with the ROI will be in higher bitplanes than all the significant bits associated with the BG [1]. Fig. 2(a) demonstrates this method. At the decoder, the ROI/BG coefficients can be identified simply by looking at the coefficients' magnitudes. All non-zero coefficients that are found to be lower than the sth bitplane are known to belong to the BG. The non-ROI coefficients are scaled up by s bitplanes before the inverse wavelet transform is applied. With Maxshift, it is also 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 ROI/BG distinction 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 define the degree of relative importance between the ROI and the BG wavelet coefficients. This means that in all the subbands, where the ROI/BG distinction is applied, no information about the BG 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).

2. GENERALIZED BBBSHIFT

2.1. BbBShift Scheme

In [9], we proposed 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 (b). 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



(a) No ROI coding (b) Scaling based method ($s = 4$)

Fig. 1. Scaling-based ROI coding method in JPEG2000 (BG: background; ROI: region of interest).

bitplanes for any coefficient. 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 non-zero wavelet coefficient, the first step is to identify whether it is an ROI coefficient or a non-ROI coefficient. This can be done by examining the bitplane level of its MSB. The set of ROI associated bitplanes is given by:

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

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

2.2. GBbBShift Scheme

In the BbBShift method, after the first s_1 ROI bitplanes are coded, the BG and ROI bitplanes are shifted one by one alternately, and both the ROI and BG of the image are refined gradually. Such a quality refinement pattern of the ROI and BG is similar to that of the general scaling based method. In practice, however, the users may desire more flexible setups. For example, at certain point during the encoding process, the quality of the ROI is already high enough, so that more ROI refinement cannot improve visual quality anymore. At this point, it would be a better choice to focusing on the quality improvement of the BG. An example is shown in Fig. 2 (c), where after the first several ROI bitplanes are coded, more than one BG bitplanes are shifted together. The lower bitplanes of the ROI coefficients are coded only after the BG has achieved good quality. We will show the advantages of this setup method in later examples.

In order for the users to have various kinds of choices for bitplane-shift strategies, we propose a generalized BbBShift (GBbBShift) method. First, we allow the users to arbitrarily divide the bitplanes into ROI bitplanes and BG bitplanes. We represent the bitplane division method using a binary bitplane mask (BP_mask) with one bit for each bitplane. 1 and 0 denote a ROI bitplane and a BG bitplane, respectively. For example, the BP_masks for Maxshift (Fig. 2(a)), BbBShift (Fig. 2(b)) and the new bitplane shift setup (Fig. 2(c)) are 111111111000000000, 111101010101010000, and 111100001111100000, respectively.

At the encoder, the ROI and BG bitplanes are ordered and shifted according to the 1's and 0's of the BP_mask. The BP_mask, which can be coded with several bytes, is transmitted with the encoded bitstream. This is different from the Maxshift and the BbBShift methods, in which the scaling values are transmitted.

At the decoder, a non-zero wavelet coefficient c can be easily classified by checking if $(c \,\&\, BP\_mask) > 0$. If it is true, then c is an ROI coefficient; otherwise, it is a non-ROI coefficient. The bitplanes of c are then shifted back to its original levels by reversing the bitplane shift scheme at the encoder.

3. EXPERIMENTS AND COMPARISONS

3.1. Experiments

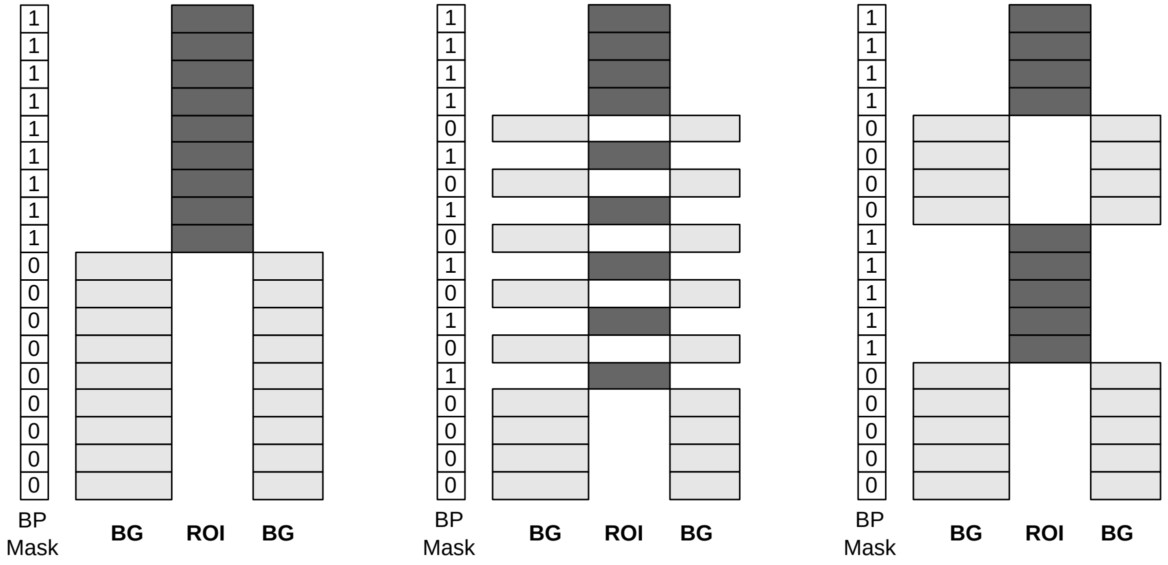
GBbBShift significantly increases the flexibility in selecting the bitplane-shift strategies. In practice, the best way to define BP_mask is highly application dependent. In this paper, we give an example to show that GBbBShift can deliver much better visual quality image than Maxshift at certain bit rates. We compress 24 bits/pixel (bpp) RGB color images using both the Maxshift method and the GBbBShift method, where the parameter setups are s=12 for Maxshift and BP_mask = 111111000000111111000000 for GBbBShift, respectively. In Fig. 3, we show the decompressed "Barbara" images using the two methods at 0.5bpp, 1.0bpp, and 2.0bpp, respectively. It can be observed that without visual difference at the ROI, the GBbBShift coded images provides better quality at the BG, especially at low bit rates such as 0.5bpp and 1.0bpp.

3.2. Comparisons

The major contribution of GBbBShift is the extension of the functionality and flexibility of the current JPEG2000 ROI coding methods. In comparison with the general scaling based methods defined in JPEG2000 Part II [2], where only rectangle and ellipse ROI shapes are allowed, GBbBShift supports arbitrary shaped ROI coding. By adjusting BP_mask, the equivalent implementations of Maxshift and BbBShift can be achieved by GBbBShift. In other words, Maxshift (which is actually a special case of BbBShift with $s_2 = 0$) and BbBShift are both special cases of GBbBShift. Compared with Maxshfit and BbBShift, GBbBShift has more flexibility to adjust the bitplane-shift strategy.

It is not necessary for the GBbBShift 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 implementation expenses. By contrast, Maxshift, BbBShift and GBbBShift do not require any shape coding, and their ROI/BG identification process is much less computationally complex.

Similar to the general scaling based method and the Maxshift method, the coding efficiency of BbBShift and GBbBShift de-



(a) Maxshift ($s = 9$)

(b) BbBShift ($s_1 = 4, s_2 = 5$)

(c) New GBbBShift setup

Fig. 2. ROI coding methods (BP Mask: bitplane mask; BG: background; ROI: region of interest).

creases 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 [3], [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 Maxshift, BbBShift and GBbBShift 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 only difference is that the bitplanes are placed in different order, which may have effect on the entropy coding module. Similar to BbBShift [9], our experiments on GBbBShift show that the bit rate change is insignificant from Maxshift.

It needs to be mentioned that the proposed GBbBShift 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 GBbBShift, a new ROI coding mode would need to be added to the standard.

4. CONCLUSION

We generalize the BbBShift introduced in [9] and propose a GBbBShift method for JPEG2000 ROI coding. The Maxshift and BbBShift methods are special cases of the GbBShift method, while GbBShift provides more flexibility. GBbBShift has many advantages over the current general scaling based method and the Maxshift method defined in the standard. Our experiments also show that it can provide better visual quality images than Maxshift at low bit rates.

5. 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 Electronics*, vol. 46, pp. 1103–1127, Nov. 2000.

[4] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards, and Practice*. Kluwer Academic Publishers, 2001.

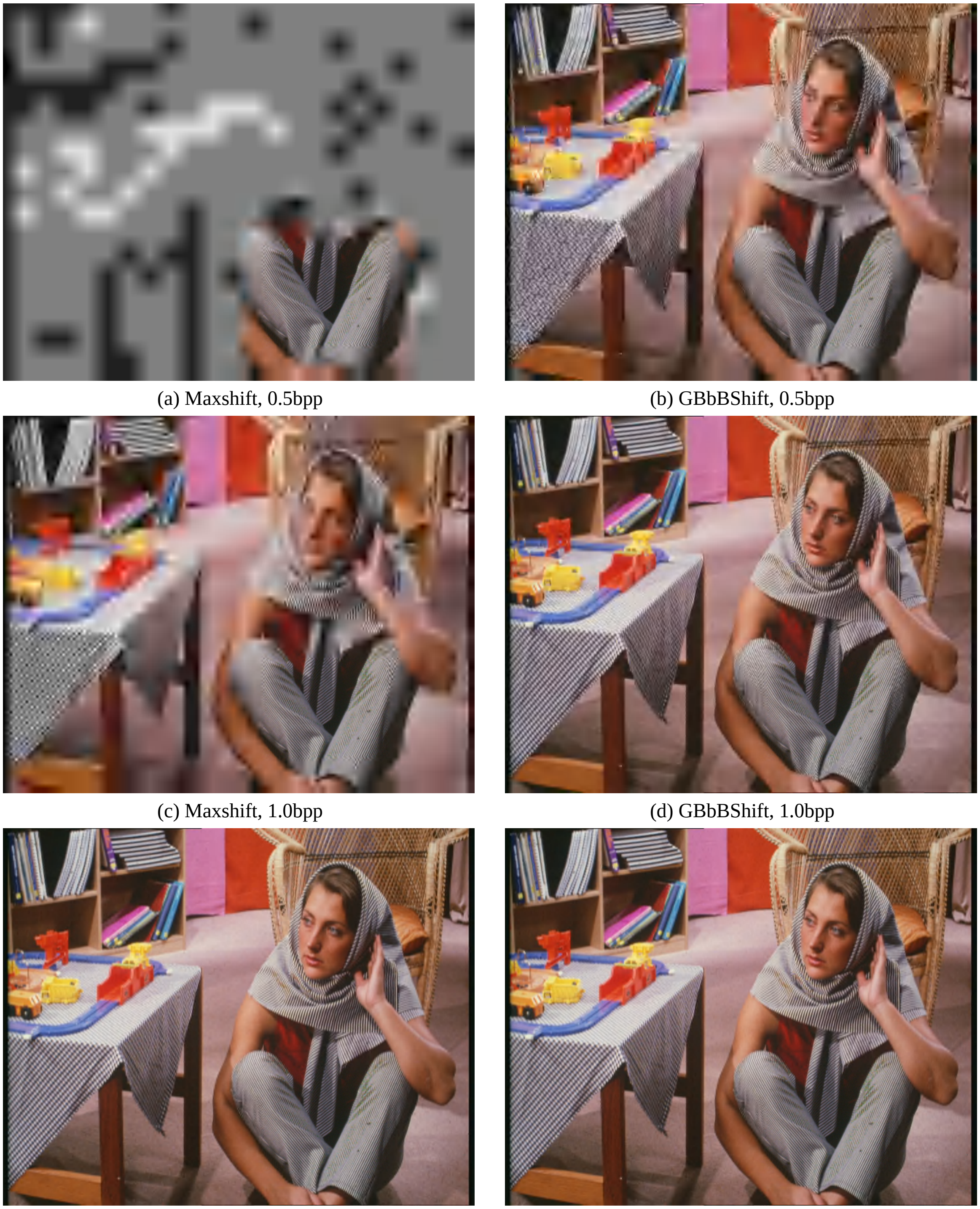
[5] E. Atsumi and N. Farvardin, "Lossy/lossless region-of-interest image coding based on set partitioning in hierarchical trees," in *Proc. IEEE Int. Conf. Image Proc.*, vol. 1, pp. 87–91, Oct. 1998.

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

[7] D. Nister and C. Christopoulos, "Lossless region of interest coding," *Signal Processing*, vol. 78, pp. 1–17, Oct. 1999.

[8] 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 Letters*, vol. 7, pp. 247–249, Sept. 2000.

[9] Z. Wang and A. C. Bovik, "Bitplane-by-bitplane shift (BbBShift) - A suggestion for JPEG 2000 region of interest coding," *IEEE Signal Processing Letters*, May 2002.



(a) Maxshift, 0.5bpp

(b) GBbBShift, 0.5bpp

(c) Maxshift, 1.0bpp

(d) GBbBShift, 1.0bpp

(e) Maxshift, 2.0bpp

(f) GBbBShift, 2.0bpp

Fig. 3. 24bpp RGB “Barbara” image coded reversibly and decoded at 0.5bpp, 1.0bpp, and 2.0bpp using the Maxshift method ($s = 12$) and the GBbBShift method ($BP\_mask = 111111000000111111000000$), respectively. The ROI is 1/16 of the image size.