A new JPEG2000 region-of-interest coding method:  
generalized partial bitplanes shift  
Y an Liang*, W enyao Liu  
College of Precision Instruments and Opto-electronic Engineering,  
Tianjin University, Tianjin 300072, China

ABSTRACT

Region-of-interest (ROI) image coding is an interesting feature in JPEG2000, which allows for encoding the ROIs in 
an image with better quality than the background (BG). Considering the limitations of two standard ROI coding methods, 
a new ROI coding method called generalized partial bitplanes shift (GPBShift) is presented in this paper. To control the 
relative importance between ROI and BG, the method divides the bitplanes of ROI and BG coefficients into two parts by 
using scaling values $S_1$ and $S_2$, respectively. Instead of shifting the bitplanes all at once by the same scaling value $S$ in 
the standard methods, GPBShift shifts part of them on the basis of the bitplane shifting scheme. The GPBShift not only can 
code arbitrarily shaped ROI without explicitly transmitting any shape information to the decoder, but also flexibly select 
the scaling values to adjust relative compression quality in ROI and BG. Additionally, the method can efficiently code 
multiple ROI with different priorities in an image. Experimental results show that the GPBShift method can provide 
significantly better visual quality than the Maxshift method at low bit rates, and higher coding efficiency than the general 
scaling based method.

Keywords: region-of-interest (ROI), generalized partial bitplanes shift (GPBShift), image coding, JPEG2000

1. INTRODUCTION

Region-of-interest (ROI) image coding technique means to compress interesting regions in an image without loss or 
with little loss, and to compress uninteresting regions with much loss. Based on this idea a high compression ratio can be 
obtained and the important information can be preserved losslessly or near-losslessly[1][2].

ROI image coding is a new feature in the JPEG2000[3][4] still image coding standard. Two kinds of ROI coding 
methods are defined in the standard: the maximum shift (Maxshift) method and the general scaling based method. These 
two methods are realized by down-shifting the bits of BG coefficients from most significant bitplane (MSB) to least 
significant bitplane (LSB). Considering the limitations of two standard ROI coding methods at the aspects of complexity 
and flexibility of coding implementation, some authors proposed several improved methods for JPEG2000 ROI image 
coding. Wang et al. proposed a bitplane-by-bitplane shift (BbBShift) method[5], which shifts the bitplanes on a 
bitplane-by-bitplane basis instead of shifting them all at once in standard methods. Subsequently, wang et al. extended 
the above method, and proposed a generalized bitplane-by-bitplane shift (GBbBShift) method[6]. Similar to the Maxshift 
method, these two methods shift part of bitplanes so that there is no overlapping between ROI and BG bitplanes. Liu et 
al. discarded the old idea of Maxshift method, and proposed a new method called partial significant bitplanes shift
that isolates a certain number of bitplanes of ROI in the MSBs and only shifts part of the MSBs of ROI coefficients. These three methods mentioned above not only allow for arbitrarily shaped ROI coding without explicitly transmitting any shape information to the decoder, but also can more flexibly adjust the relative importance of ROI and BG than Maxshift method. However, the BbBShift and GBbBShift methods are not compatible with the current JPEG2000 ROI coding definitions, and a new mode is needed. Moreover, it is difficult for the two methods to code multiple ROIs with different priority during encoding/transmission of one image. The PSBShift method supports multiple ROIs coding with different degrees of interest in an image, but has worse performance to control the relative quality of ROI and BG than the BbBShift and GBbBShift methods.

In this paper, a new and flexible ROI coding method called generalized partial bitplanes shift (GPBShift) is proposed, which can combine the advantages of the two standard methods and efficiently compress multiple ROIs with different priority. Additionally, the GPBShift method includes the Maxshift, GBbBShift and PSBShift methods, while provides more flexibility for “degree-of-interest” adjustment of the ROI.

2. GPBSHIFT METHOD

2.1 JPEG2000 ROI coding

ROI image coding is one of the requirements in the new JPEG2000 image coding standard, which allows for encoding the ROIs in an image with better quality than the BG. These two methods place ROI-associated bits in the higher bitplanes by down-shifting the bits of BG coefficients from MSB to LSB, so that ROI coefficients can be coded firstly in the following embedded bitplane coding. At the decoder, the bitplanes are reconstructed and the non-ROI coefficients are shifted 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. Fig.1 illustrates how the bitplanes are shifted.

![Bitplanes Diagram](https://via.placeholder.com/150)

- (a) No scaling
- (b) General scaling based method, \( S = 4 \)
- (c) Maxshift method, \( S = 9 \)
- (d) Proposed GPBShift method, \( S_1 = 4, S_2 = 6 \)

Figure 1: Standard ROI coding methods in JPEG2000 and the proposed GPBShift method

(bitplanes are represented by the gray bars)

An illustration of the general scaling based method is shown in Fig.1 (b), where the scaling value of bitplanes is represented as \( S \). In the general scaling based method, the scaling value and the shape information of the ROIs are also added into the encoded bitstream. 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, which can be coded with a small number of bits. However, this limits the application scope of ROI coding.

A very effective solution, the Maxshift method, was proposed for JPEG2000. It may be considered as a particular case of the general scaling based method when the scaling value is so large that there is no overlapping between BG and ROI bitplanes, i.e., the scaling value $S$ must satisfy

$$S \geq \max(M_b)$$

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 BG. Fig. 1(c) demonstrates the bitplane shift in the Maxshift method. Therefore, ROI shape is implicit for the decoder in the Maxshift method, and arbitrarily shaped ROI coding can be supported. Moreover, it can flexibly treat wavelet subbands differently. The major limitation of the Maxshift method is that it cannot flexibly control the relative importance between ROIs and BG by adjusting the scaling value. 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.

### 2.2 GPBShift method

Instead of shifting the bitplanes all at once by the same scaling value $S$ in the standard methods, the GPBShift method divides the bitplanes of ROI and BG coefficients into two parts by using scaling values $S_1$ and $S_2$, and shifts part of them on the basis of the bitplane shifting scheme. Thereby, we name this method as GPBShift. An illustration of the GPBShift method is shown in Fig.1 (d) (where $S_1=4$, $S_2=6$). The number of the MSBs of ROI coefficients is the same as the scaling value $S_1$, and the number of the MSBs of BG is represented as $S_2$. 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 $b > S_1$, shift it down to bitplane $b + S_1$.

2) For any bitplane $b$ of a BG coefficient:
   - if $b \leq S_2$, shift it down to bitplane $b + S_1$;
   - if $b > S_2$, shift it down to bitplane $b + S_1 + (\max(M_b) - S_2)$.

At the decoder, ROI coefficients can be identified in the same way as Maxshift. The bitplane reconstructing scheme is as follows:

For any bitplane $b$ of any wavelet coefficient (ROI or BG):
   - if $b \leq S_1$, no shift;
   - if $S_1 < b \leq S_1 + \max(M_b)$, shift it up to bitplane $b + S_1$.
   - if $b > S_1 + \max(M_b)$, shift it up to bitplane $b + S_1 + (\max(M_b) - S_2)$.
Then three parts of bitplanes combine to form an original ROI or BG coefficient.

The Maxshift, GBbBShift and PSBShift methods are special cases of the GPBShift method, while GPBShift provides more flexibility for “degree-of-interest” adjustment of the ROI. If \( S_1=S_2=0 \) or \( S_1=S_2=\max(M_b) \), the Maxshift method is obtained. If \( S_1=S_2 \), the GBbBShift method is received. Similarly, the PSBShift method is obtained if \( S_2=\max(M_b) \). Therefore, the encoder of the GPBShift can generate the code stream that is compatible with these three methods, and decoder of the GPBShift can accurately reconstruct the bitstream generated by anyone of these three methods. Especially, it is wonderful for practical applications that the GPBShift and the Maxshift are compatible.

Although the scaling values \( S_1 \) and \( S_2 \) can reach any integer within 0 to \( \max(M_b) \) in theory, it yet would result in idle bitplanes when \( S_1>S_2 \) happens. As illustrated in Fig.2, it would significantly reduce the efficiency of the following entropy coding by increasing the dynamic range of wavelet coefficients. Therefore, \( S_1 \leq S_2 \) is required in the GPBShift method.

![Diagram of idle bitplane](image1)

![GPBShift method for multiple ROI coding](image2)

By selecting different value \( S_1 \) for different ROI, The GPBShift method can code multiple ROIs with different quality according to their priorities in an image. The general scaling based method can support multiple ROI coding. But it needs to code ROI shape and restrict it. The Maxshift method may support multiple ROI coding, while it would significantly reduce the compression efficiency by largely increasing the dynamic range of wavelet coefficients. The dynamic range at least is \((n+1)\times\max(M_b)\), where \( n \) represents the number of ROI. The GPBShift method can support efficient multiple ROI coding with arbitrary ROI shape, as illustrated in Fig. 3. For example, one image with three defined ROIs, i.e., ROI1, ROI2, and ROI3, is compressed reversibly. The priority order of these ROIs is ROI1>ROI2>ROI3. The scaling values of three ROIs should be chosen as \( S_{1,ROI1}>S_{1,ROI2}>S_{1,ROI3} \), e.g., \( S_{1,ROI1}=8 \), \( S_{1,ROI2}=6 \), \( S_{1,ROI3}=4 \), shown in Fig. 3. The scaling value stored in the code stream is \( S_1=\max(S_{1,ROI1}, S_{1,ROI2}, S_{1,ROI3})=8 \). The code process of multiple ROI coding is similar to the one of single ROI coding. Moreover, \( S_2 \geq S_{1,ROI3} \) is also required in order to avoid the problem of idle bitplane.

3. EXPERIMENTS AND COMPARISONS

3.1 Functionality

Compared with the general scaling based method defined in JPEG2000 Part II, which only supports rectangle and ellipse ROI shapes, the GPBShift method can code arbitrarily shaped ROI without coding the shape. The new method
can also allow different ROI definitions in different wavelet subbands as the Maxshift method does. Different from the Maxshift method, the GPBShift method can also flexibly adjust the relative importance between ROI and BG by using different scaling values. This flexibility may lead to improved quality of ROI coding when decode rate is same. An example is given in Fig. 4, where the “Barbara” image (512 × 512, 8bpp) is compressed reversibly and decompressed at 0.5 bpp using the Maxshift method (S=11) and the GPBShift method (S_1=8, S_2=9), respectively. The ROI is at the face region and 1/16 of the image size. It can be observed that without visual difference at the ROI, the GPBShift method provides better quality at the BG than the Maxshift method. The PSNR (peak signal-to-noise ratio) of reconstructed image is 17.50dB in the Maxshift method [Fig.4 (a)], while the one is 26.74dB in the GPBShift method [Fig.4 (b)]. If ROI needs to have higher quality than BG, the scaling values S_1 and S_2 can be closer to 0 or max(M_b) at the same time, then the results would be closer to the Maxshift method.

![Figure 4](image)

(a)                                                     (b)

Figure 4: The “Barbara” image coded reversibly and decoded at 0.5bpp using the Maxshift method [(a), S=11] and the GPBShift method [(b), S_1=8, S_2=9]. The ROI is at the face region and 1/16 of the image size.

In comparison with the two standard methods defined in JPEG2000, another advantage of the GPBShift method is that it can efficiently code multiple arbitrary shaped ROIs in an image without explicitly transmitting any shape information to the decoder. As shown in Fig.5 (a), the “Couple” image (512×512, 8bpp) includes three defined ROIs. The priority order of these ROIs is ROI1>ROI2>ROI3. Thereby, the scaling values of the ROI are set as S_1,ROI1=8, S_1,ROI2=6, S_1,ROI3=5, and the scaling value of the BG is set as S_2=5. The reconstructed image at 0.25 bpp using the GPBShift is given in Fig.5 (b).

3.2 Complexity and compatibility

Similar to the two standard methods, the coding efficiency of GPBShift decreases in comparison with JPEG2000 without any ROI coding. The reason is that bitplane shifting increases the dynamic range of the wavelet coefficients being encoded. It is reported [8], [9] 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. The increased bit rate of the GPBShift is less than Maxshift, depending on the scaling values used. Loss compression has to be used when channel bandwidth is narrow in
communication systems. In loss compression with ROI, the greater part of code bits is distributed to the ROIs, thus it would significantly increase the PSNR of the ROIs and solve the difference between the image quality and the compression rate.

![Figure 5](image.png)

Figure 5: The “Couple” image [(a)] and its result of multiple ROI coding at 0.25bpp using the GPBShift method [(b)]

The general scaling based method requires 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 the GPBShift is much cheaper, when the same scaling values are used. Compared with the Maxshift method, a little more complicated procedure is needed in the GPBShift method in order to shift the encoded/decoded bits to the original bitplanes. If the point of lossless coding is reached, the bit rate produced by the GPBShift method is less or approximately the same as the Maxshift method, depending on the scaling values used and ROI sizes. Experiments on lossless coding of 12 images (8bpp, images of size $176 \times 144$ up to $512 \times 512$, ROI is $1/4$ and $1/16$ of the image size) show that compared with the Maxshift method, the GPBShift method spends $0\%$-$38.57\%$ (depending on the scaling values used) less bits for ROI size $1/4$ of the image size, and for ROI size $1/16$ of the image

![Figure 6](image.png)

Figure 6: Relative difference of bit rate generated by the Maxshift and GPBShift methods at lossless “Lena” image
size, the GPBShift costs 0.20%~16.98% more bits if \( S_2 = 8, 9 \) or 10, while 0%~18.06% less bits if other scaling values are used. The increased bit rate is due to the reason that the GPBShift affects the following entropy coding by reordering the bitplanes. It is still true that the GPBShift method codes no more bitplanes than the Maxshift method. When the point of lossless coding on “Lena” image is reached, the relative difference of bit rate generated by the Maxshift and the GPBShift is shown in Fig. 6.

Moreover, the code stream generated by the GPBShift method has the same syntax format as the current JPEG2000. While it still needs a minor modification of the JPEG2000 Part I decoder as described above in order to decode the bitstream of the GPBShift, the modified decoder can also handle the bitstream produced by the Maxshift method.

4. CONCLUSION

We have proposed a GPBShift method for JPEG2000 ROI coding and compared it with the Maxshift method and general scaling based method defined in the standard. The new method has four primary advantages: 1) it supports arbitrarily shaped ROI coding without coding the shape; 2) it can control the relative importance between ROIs and BG; 3) it allows different wavelet subbands to have different ROI definitions; and 4) the new method can efficiently code multiple ROIs with different priorities in an image at the low bit rates. The GPBShift method is valuable for future research in ROI image coding and its applications.

REFERENCES


*contact. yan_liang@eyou.com; phone 86022-27890667; fax 86022-27406726; College of Precision Instruments and Opto-electronic Engineering, P.O.Box 235, Tianjin University, Tianjin 300072, China