Proceedings of the 5th WSEAS International Conference on Signal Processing, Istanbul, Turkey, May 27-29, 2006 (pp65-69)

An Efficient Method for Region of Interest Coding in JPEG2000

Fayez Idris and Ferat Atef
Department of Computer Engineering
Jordan University of Science and Technology
P.O. Box 3030, Irbid 22110
JORDAN

Abstract: - The JPEG2000 standard defines two methods for region of interest (ROI) coding, the maximum shift and the general scaling. In the maximum shift method, an ROI can have any shape that does not need to be transmitted to the decoder, but does not have the flexibility to select an arbitrary scaling value to compromise the ROI quality compared with the background. The general scaling based method supports any scaling value providing the user/application with the ability to control ROI and background qualities, but it limits the choice of the ROI shape and needs ROI shape coding.
In this paper, we propose a novel ROI coding method named Bitplane Overlap Shift (BOShift). The proposed BOShift is compatible with the JPEG2000 standard and it supports both arbitrary scaling values and arbitrary ROI shapes, without shape coding. Moreover, it encodes images with distinguished performance at low bit rates.

Key-Words: - JPEG2000, Region of Interest, ROI, Maximum Shift, Maxshift, General Scaling,

1 Introduction

Region of interest (ROI) coding feature in JPEG2000 enables a user to define an area of the image to be compressed to have a particular relevance by some defined measure [1]. To encode a region of an image with better quality than the rest of it, i.e. background (BG), the encoder scales down the coefficients that belong to the background below the coefficients that belong to the ROI. This will place the ROI bits in the bit stream before the background bits. Accordingly, the ROI will be decoded before the rest of the image [3].

The background coefficients scaling scheme is defined by the ROI coding methods. The JPEG2000 standard specifies two ROI coding methods: the maximum shift (Maxshift) method [1] and the general scaling based method [2].

General scaling based method defines arbitrary scaling values for coding ROIs giving the user/application control on the quality of the ROIs compared with the background and the ability to define multiple ROIs with different priorities within the encoded image as shown in Fig 1(b). As it is most likely that an ROI can be complex in shape, the general scaling based method has a major drawback when complex shapes are desired to be encoded. The Maxshift method can be realized as a special case of the general scaling based method, since the scaling values are chosen such that there is no overlap between the ROI and background bitplanes. The scaling value, s, must be chosen to satisfy:

$$s \geq max(M_b) \quad (1)$$

where $max(M_b)$ is the largest number of magnitude bitplanes for any ROI coefficient. Forbidding such overlapping is the main advantage of the Maxshift method since there is no need to encode the ROI shape information. After the scaling process, all significant bits associated with the ROI will be in higher bitplanes than all significant bits associated with the background as shown in Fig 1(c). The major drawback of the Maxshift method is that it does not have the flexibility to allow an arbitrary scaling value to define relative importance of the ROI and the background as in the general scaling based method. On the other hand, the scaling value used by the encoder is large thus the total number of magnitude bitplanes for each code block will increase by the maximum number of magnitude bitplanes and this may cause bit overflow. In addition, the user/application cannot define multiple ROIs with relative importance among each other.

To address the Maxshift method limitation methods like the Bitplane by Bitplane Shift (BbBShift) method [5], Generalized Bitplane by Bitplane Shift (GBbBShift) method [6] and Partial Significant Bitplanes Shift (PSBShift) method [7] have been reported. The reported methods are based on defining an arbitrary scaling value to tweak the relative importance between the ROI and background.

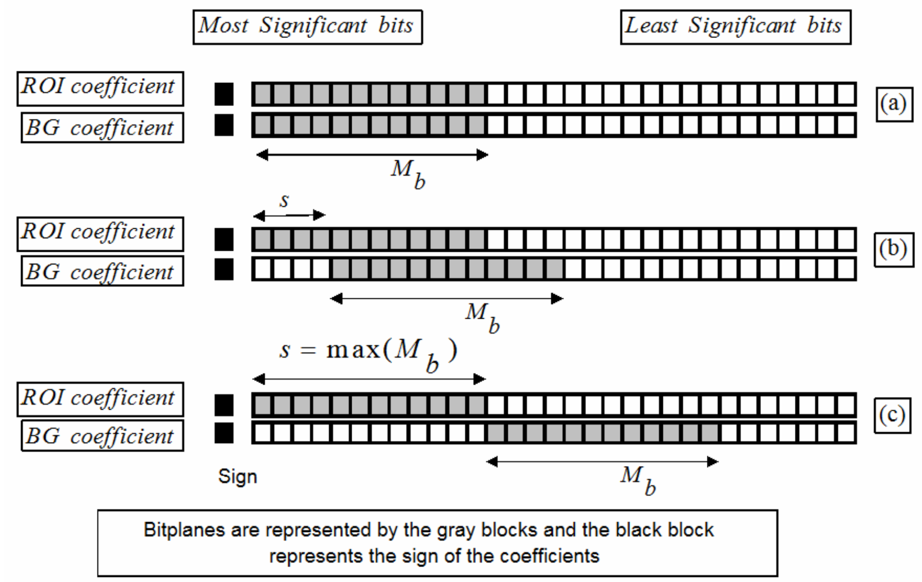

Fig. 1: The standard ROI coding methods. (a) No scaling. (b) General scaling based method (c) Maxshift method

However, the main challenge when proposing a new method is to make it compatible with the standards, such that there is no need to modify the decoder implementation. Introducing a change in the decoder means that all decoder providers have to change their decoder implementations [3]. PSBShift is compatible with the JPEG2000 standard while BbBShift and GBbBShift methods are not [7]. However, PSBShift method encodes more bitplanes than the Maxshift method does. In consequence, it increases the total number of bitplanes encoded in the encoded image bit stream reducing the image coding efficiency.

There is a need for new ROI coding method that is compatible with JPEG2000 image standard, and provides the user/application the flexibility to have control on the relative importance between ROI and background. In addition, arbitrary ROI shapes must be supported without ROI shape coding overhead. Moreover, the new method must give the user/application the ability to define multiple ROIs with different priorities.

The rest of this paper is organized as follows. The proposed method for ROI coding is detailed in Section 2. Experimental results are presented in Section 3. Finally, the conclusions are presented in Section 4.

2 Bitplane Overlap Shift Method

We propose a novel ROI coding method, which is a variation of the Maxshift method. The new method introduces a new parameter named overlap value (o). This value is used to adjust the scaling value used to scale the background coefficients giving the user/application the flexibility to define the relative importance of background compared with the ROI. Thus, the method is named Bitplane Overlap Shift (BOShift) method.

The scaling value used by BOShift method depends on the number of magnitude bitplanes (M_b),as in the Maxshift method, but it is tweaked by the overlap value (o) in order to identify the relative significance of the quantized transform coefficients, as in the general scaling based method. The BOShift scaling value is given by:

$$s = max(M_b) - o \quad (2)$$

The Maxshift method constrains the selection of the scaling value such that it must be sufficiently large to make the smallest non-zero ROI coefficient larger than the largest background coefficient. This constraint, on the selection of the scaling value forbids any overlap between the ROI coefficients bitplanes and the background coefficients bitplanes. In consequence, the decoder differentiates whether the decoded coefficient belongs to the background or not by comparing the number of decoded bits of the current coefficient with the scaling value used by the encoder, which is easily retrieved from the header of the image code stream.

BOShift coding technique forces ROI coefficients that have zero bits in the most (s) bitplanes to be demoted to the background. Consequently, the decoder has the ability to utilize the Maxshift decoder although there are (o) bitplanes that overlap between ROI bitplanes and

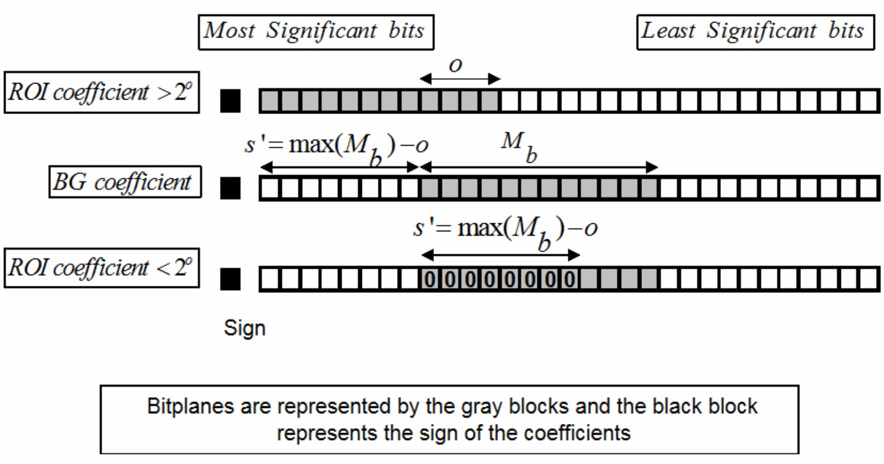

Fig. 2: The BOShift method with o= 4

background bitplanes. That is, although there is (o) bitplanes that overlap between ROI and bitplanes but still the smallest non-zero ROI coefficient larger than the largest background coefficient. As it is demonstrated in Fig. 2 BOShift method reallocates all non-zero ROI coefficients that are less than 2^o to be in the background and therefore the BOShift method is compatible with the JPEG2000 image standard

The following steps describe the procedure for encoding ROI using BOShift:

1. Generate the ROI mask, $M_i(u,v)$
2. Find the scaling value, s , such that

   $$s = \max(M_b) - o \qquad (3)$$

   where o denotes number of bitplanes to overlap between the ROI and the background
3. Scale down the transform coefficient $q(x,y)$ such that

   if $q(x,y) \in BG$ **then**

   $$\textit{scale down } q(x,y) \textit{ by } 2^s \qquad (4)$$

   else if $q(x,y) \in ROI$ **then**

   $$\begin{Bmatrix} \textit{if } q(x,y) < 2^o, \textit{scale } q(x,y) \textit{ by } 2^s \\ \textit{if } q(x,y) \geq 2^o, \textit{do not scale} \end{Bmatrix} \qquad (5)$$

   where ROI and BG denote the region of interest and background respectively given by $M(x,y)$
4. Write the scaling value, s , into code stream

3 Experimental Results

Simulations were carried out using the Lena image. The Lena image was compressed using Maxshift and BOShift with different overlap (o) values. The PSNR of the reconstructed image at different bit rates is shown in Fig. 3. It can be seen from Fig. 3 that BOShift outperforms Maxshift for bit rates below or equal to 2 bpp, and it has similar performance for rates above 2 bpp. It is important to note that the performance of the BOShift is not achieved at the expense of increasing the bit rate of the compressed image. Table 1 shows the effect of using BOShift method with different overlap values on the image coding rates. As it can be noticed, as the overlap value increases the images are coded at lower bit rates. As the overlap value (o) increases, the number of bitplanes encoded decreases but also the number of ROI coefficients that have been demoted to background also increases. This means that, at low bit rates the probability of truncating coefficients that belong to ROI increases, which in turn leads to encoding the ROI with relatively lower quality. Our simulations showed that the BOShift method enhances the image coding rates in the range of 1% to about 8% in comparison with the Maxshift method. This is due to the total number of bitplanes coded by the encoder. Table 2 shows the number of bitplanes encoded for an ROI coefficient for Maxshift method, PSBShift method, and, BOShift method. The BOShift method encodes the images with the lowest number of bitplanes.

Fig. 4 shows the decompressed Lena images using Maxshift and BOShift methods at 0.1 bpp and 0.08 bpp. It can be observed that without visual difference at the ROI, the BOShift coded images have better quality at the background, especially at low bit rates.

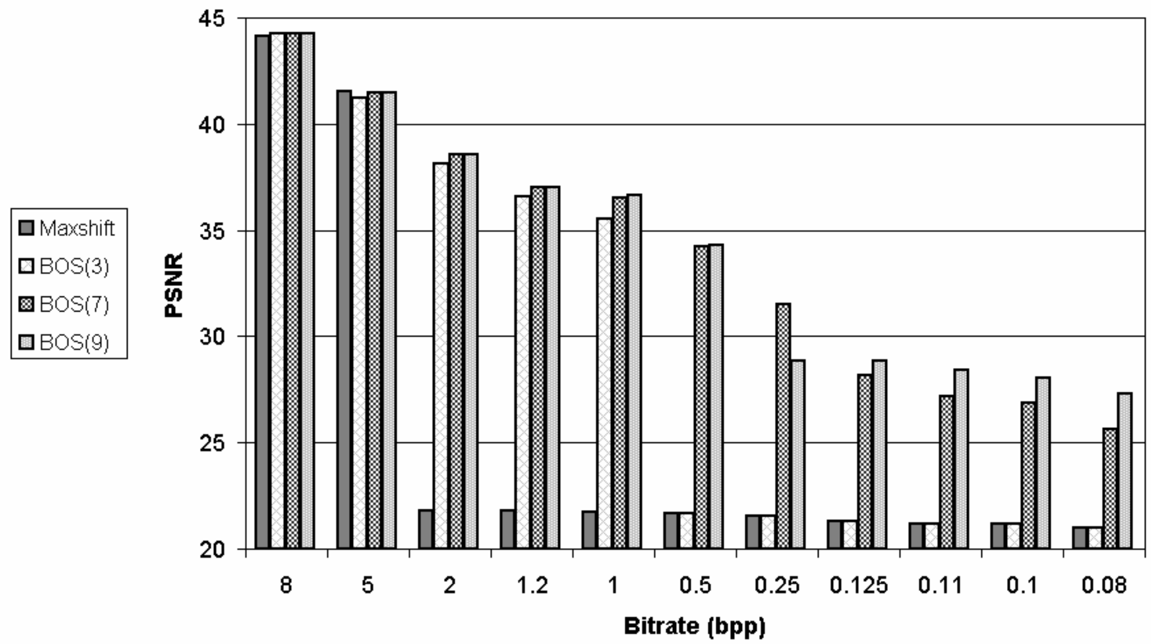

Fig. 3: PSNR values versus image reconstruction bit rates for Lena image encoded using BOShift method (BOS) and Maxshift.

Table 1: Lena image coding rates for Maxshift and BOShift methods using different overlap values

BOShift Overlap value (*o*)	Image coding rates for Maxshift (bpp)	Image coding rates for BOShift (bpp)	BOShift improvement over image coding rates (%)
1	7.271	7.187	1.16%
3		6.916	4.88%
5		6.780	6.75%
7		6.724	7.52%
8		6.717	7.62%
9		6.712	7.69%

Table 2: Number of bitplanes coded for ROI coefficient in Maxshift, PSBShift, and, BOShift

ROI coding method	Number of ROI bitplanes
Maxshift	$max\ (M_b)$
PSBShift	$max\ (M_b) + s$
BOShift	$max\ (M_b) - o$

In comparison with BbBShift and GBbBShift methods, using BOShift method is more feasible since it is compatible with the JPEG 2000 image standard. In addition, BOShift encoder does not have to change the order of the bitplanes in order to adjust the relative importance of ROI compared with the background as BbBShift and GBbBShift methods propose. Both PSBShift and BOShift are compatible with the JPEG 2000 standard. However, BOShift excels by its efficiency when reconstructing images at low bit rates, since the total number of bitplanes encoded using BOShift are less than that for PSBShift method as shown in Table 2.

4 Conclusion

We have proposed a novel JPEG2000 ROI coding method – BOShift method. BOShift method combines the major advantages of the previously proposed ROI coding methods: (1) it supports arbitrarily shaped ROI coding without coding the shape; (2) it can control the relative importance between ROIs and background by using appropriate overlap values. Moreover, it codes images at lower bit rates and reconstruct images with higher fidelity at low bitrates since using BOShift method reduces the total number of bitplanes encoded in the image code stream.

References:

[1] ISO/IEC JTC 1/SC 29/WG 1 (ITU-T SG8) JPEG2000 Part I Final Committee Draft Version 1.0, March 2000.

[2] ISO/IEC JTC 1/SC 29/WG 1 (ITU-T SG8) JPEG2000 Part II Final Committee Draft, December 2000.

[3] D. Chai and A. Bouzerdoum, JPEG2000 Image Compression: An Overview, Australian and New Zealand Intelligent Information Systems Conference (ANZIIS'2001), Perth, Australia, November 2001, pp. 237-241.

[4] C. Christopoulos, J. Askelof, and M Larsson, Efficient Methods for Encoding Regions of Interest in the Upcoming JPEG2000 Still Image Coding Standard; IEEE Signal Processing Letters, Vol. 7, No. 9, September 2000, pp. 247-249.

[5] Z. Wang and A. C. Bovik, Bitplane-by-Bitplane Shift (BbBShift) – A Suggestion for JPEG2000 Region of Interest Coding, *IEEE Signal Processing Letters*, Vol. 9, No. 5, May 2002, pp. 160-162.

[6] Z. Wang, S. Banerjee, B. L. Evans, and A. C. Bovik, Generalized Bitplane-by-Bitplane Shift Method for JPEG2000 ROI Coding, *Proc. IEEE Int. Conf. on Image Processing*, Vol. III, September 2002, pp. 81-84.

[7] L. Liu and G. Fan, A New JPEG2000 Region-of-Interest Image Coding Method: Partial Significant Bitplanes Shift, *IEEE Signal Processing Letters*, Vol. 10, No. 2, February 2003, pp. 35-38.

a) Maxshift, .0.1 bpp

b) BOShift, .0.1 bpp

c) Maxshift, .0.1 bpp

d) BOShift, .0.08 bpp

Fig. 4. “Lena” decoded at 0.1 bpp, and, 0.08bp using the Maxshift method (s = 14) and the BOShift method (o = 9), respectively. The ROI is 1/2 of the image size