Published in SPIE's 46th annual meeting, Applications of Digital Image Processing XXIV,
Proc. of SPIE, vol. 4472, pp. 267–275, San Diego, CA, USA, Jul. 29 – Aug. 3, 2001.

New approach to JPEG 2000 compliant Region Of Interest coding

Raphaël Grosbois, Diego Santa-Cruz and Touradj Ebrahimi

Signal Processing Laboratory - Swiss Federal Institute Of Technology, CH-1015, Lausanne, Switzerland

ABSTRACT

Region Of Interest (ROI) coding is one of the innovative functionalities supported by JPEG 2000, the new ISO/ITU-T still image coding standard developed by the Joint Photographic Experts Group (JPEG). It enables a non-uniform distribution of the image quality between a selected region (the ROI) and the rest of the image (background). This feature is obtained by scaling background coefficients in the wavelet domain.

According to the ROI *Maxshift* method defined in JPEG 2000 part 1 (baseline algorithm), the background bit-planes are down-shifted below all ROI coefficients. In addition, the ROI can have any shape as the latter does not need to be transmitted to a JPEG 2000 decoder (no codestream overhead due to the coding of the shape). On the contrary, this method requires decoding of all ROI coefficients before accessing bit-planes of the background. Furthermore, it uses large shifting values that significantly increase the number of total bit-planes to encode.

In JPEG 2000 part 2 (extensions), a generic (Scaling based) ROI coding has been included. This method supports any scaling values. In particular, it allows a rough control on both ROI and background qualities distributions in the codestream, but implies the derivation of a ROI bit-mask at the decoding side.

This paper starts by providing some hints on how to choose an optimal Maxshift scaling value, as well as how to pad the ROI extra bits appearing during the shift operation. Then, it proposes an encoding algorithm that combines advantages of both ROI methods. This algorithm can be used by applications where visually lossless ROI's is acceptable and is based on an extension of the Maxshift method to low scaling values. The generated codestreams remain compliant with the Maxshift decoding algorithm described in JPEG 2000 part 1, and can be consequently handled by any JPEG 2000 decoder.

Keywords: Image coding, wavelet domain, region of interest, ROI, scaling based method, Maxshift method, JPEG 2000

INTRODUCTION

JPEG 2000 is the new wavelet-based still image coding standard developed by the Joint Photographic Experts Group (JPEG). Currently, it is divided into seven distinct parts, where part 1[1] describes the baseline algorithm and part 2[2] its extensions. Together with higher compression performances, JPEG 2000 stands out by its support for many functionalities essential to the evolving communication networks.[3] Among them, one identifies several progressive decoding modes: by resolution, by quality, by position or by region. More specifically, the last progression type refers to the Region Of Interest (ROI)[4] feature, where an ROI is a region of the image that is expected to have a better quality than the rest at any decoding bit-rate. In other words, this implies a non-uniform distribution of the quality inside the image.

This paper is organized as follows. The key points of the JPEG 2000 algorithm are introduced in section 1. Then, section 2 emphasizes on ROI coding methods defined in part 1 and part 2 of the standard. In addition, it gives clues about choosing an optimal Maxshift scaling value, as well as an appropriate bit-padding technique for ROI extra bits. Section 3 proposes a modified version of the ROI scaling based method that remains compliant with JPEG 2000 part 1 whilst supporting any scaling values. Finally conclusions are drawn in the last section.

1. OVERVIEW OF THE JPEG 2000 ALGORITHM

The JPEG 2000 standard only defines a decoding algorithm along with the syntax of a compressed codestream. Nevertheless, a typical encoder at least contains the four following modules: wavelet transform, quantization, entropy encoding and codestream building.

Further author information: (send correspondence to Raphaël Grosbois)
Raphaël Grosbois: E-mail: Raphael.Grosbois@epfl.ch
Diego Santa-Cruz: Email: Diego.SantaCruz@epfl.ch
Touradj Ebrahimi: E-mail: Touradj.Ebrahimi@epfl.ch

Figure 1. Simplified JPEG 2000 compression scheme. The image is decomposed into wavelet subbands. In each subband, the coefficients are quantized, grouped into code-blocks, and their indexes are entropy encoded. At last, the pieces of the resulting bit stream are multiplexed into the final codestream.

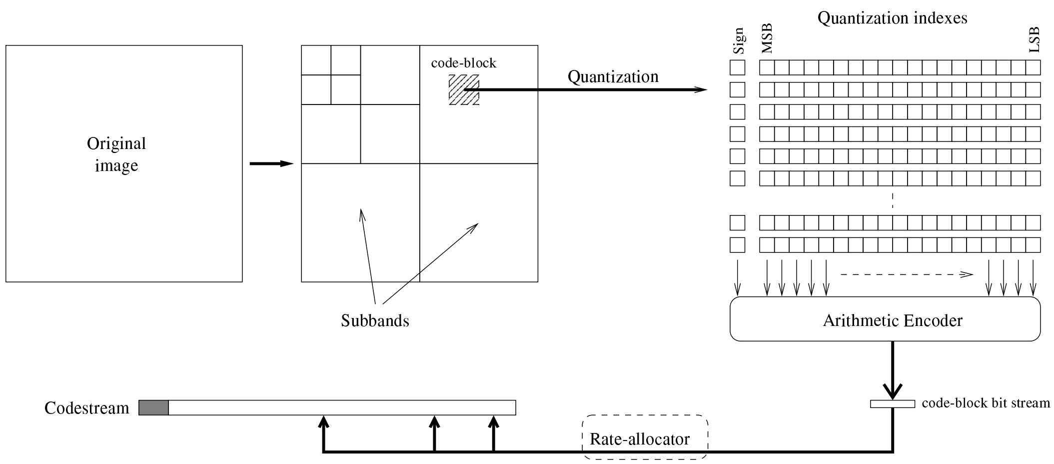
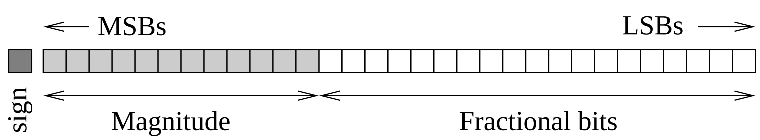


Figure 2. Example of sign-magnitude representation of a quantization index on 32 bits.



As depicted in Figure 1, the original image is first decomposed into wavelet subbands.[6] Then, the wavelet coefficients are quantized (embedded scalar quantization), and they are grouped into rectangular blocks called *code-blocks*. Figure 2 presents the sign-magnitude notation used to represent the quantization indexes in the code-blocks. Note that only the most significant bit-planes (MSB's), containing the sign and the integer part of the magnitude, are encoded. Nevertheless, the fractional bits in the least significant bit-planes (LSB's) are generally kept for a better distortion estimation during the rate-allocation process.[5]

Finally, the quantization indexes are encoded by a bit-plane based arithmetic encoder where each code-block is coded independently. The resulting bit streams are written into the final codestream during the rate allocation step. It is worth pointing out that the entropy encoder works from the most significant bit-plane, which contains at least one non-zero bit, toward the least significant. Note that the encoder skips the MSB's where all bits are equal to zero. In this case, it just indicates in the codestream header the number of skipped bit-planes. Furthermore, each code-block's bit-plane is processed in a maximum of three passes (called *coding passes*), at the end of which, the bit stream may be truncated.[5]

At the decoder side, the code-block coefficients are progressively decoded, bit-plane by bit-plane, from the received bit streams. Then, after the inverse quantization and inverse wavelet transform steps, the decoded image is reconstructed. A more complete description of these operations is given in the standard specification document.[1]

It is straightforward from the above schemes that, depending on the order in which the code-block coding passes are decoded (either because of the progression of the codestream or because of the decoder's parsing capabilities), one can obtain different types of progressive coding modes. Thus, among the several progression modes defined in JPEG 2000, the two most important are:

- Progression by resolution: Code-blocks from lower resolutions (i.e. lower frequencies subbands) are decoded first.
- Progression by quality (or progression by layer): Coding passes codewords from the code-blocks participating the most to the distortion reduction decoded first.

In fact, the ordering of code-blocks coding passes in the codestream is generally managed by a rate-allocation algorithm (such as EBCOT[5]), that tries to optimize the rate-distortion characteristics of the codestream.

Figure 3. ROI scaling operation where s is the scaling value

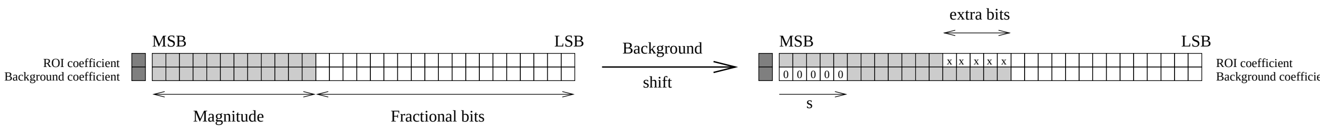
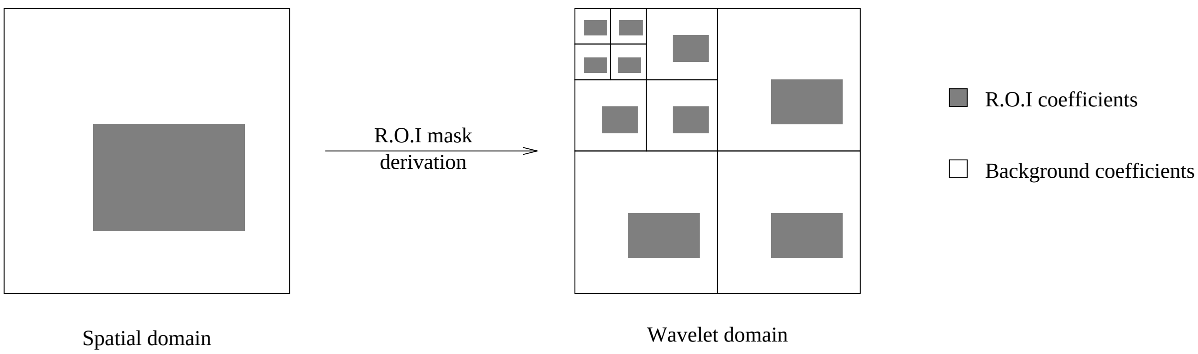


Figure 4. 2D ROI bit-mask derivation in subbands from the ROI in the spatial domain



2. ROI CODING BY SCALING

As mentioned in the introduction, a Region Of Interest (ROI) is an area in the image that is expected to exhibit a better quality than the rest of it at any decoding bit-rate. In the encoding scheme presented earlier, one deduces that such a feature is easily obtained by encoding first the quantization indexes of coefficients that participate to the ROI reconstruction. These coefficients are called ROI coefficients in the remainder of the paper.

2.1. ROI Scaling based method

This method is described in JPEG 2000 part 2.[2] As the quantization indexes are represented in sign-magnitude, and since the entropy encoder processes the code-blocks, bit-plane by bit-plane, the solution defined in the standard aims at isolating ROI bits in the most significant bit-planes. This is done by down-shifting the background bits toward the least significant bit-planes, as illustrated in Figure 3. Note that with this scheme, some ROI extra bits, denoted x in the figure, are encoded by the entropy encoder. The choice of their values (i.e. the bit-padding technique) is discussed in section 2.4.

Any scaling value can be used for the background shift. Moreover, this value allows a rough control on ROI and background qualities during a progressive decoding. Note that in practice, one has to derive first a bit-mask that identifies, in each subband, which are the ROI coefficients (see Figure 4). The mask derivation depends on the length of the synthesis wavelet filters an the number of decomposition levels in the wavelet transform, as described in.[2]

At the decoder side, one has to derive the mask, to shift upward background coefficients, to remove the ROI extra bits, and finally apply the dequantization and inverse wavelet transform. Furthermore, since there is generally no way to distinguish ROI from background coefficients, the bit-mask used during encoding must be transmitted to the decoder. Hence, in order to reduce the cost of the bit-mask encoding, the JPEG 2000 part 2 imposes the ROI's to a combination of rectangular and elliptic regions.

2.2. ROI Maxshift method

This method is defined in JPEG 2000 part 1. It can be understood as a particular case of the previous, when the scaling values are such that, in all code-blocks, there is no overlap between background and ROI bit-planes (see Figure 5). The choice of the ROI extra bits padding is also discussed in section 2.4. Note that, with this method, the ROI bit-mask is not required to

Figure 5. ROI Maxshift method where s is the scaling value

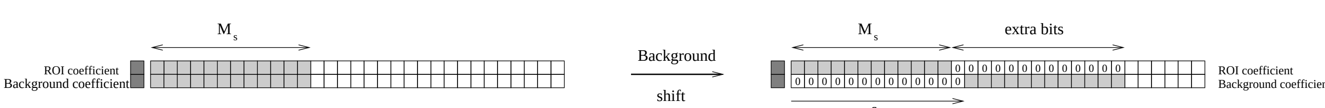
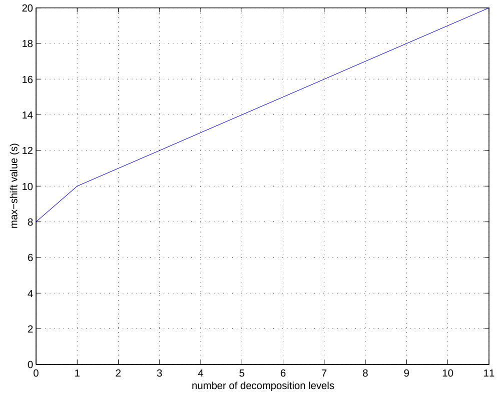


Figure 6. Optimal Maxshift scaling value versus the number of wavelet decomposition levels. This result has been obtained with an ROI located between (200,200) and (400,400) in the GoldHill image.



distinguish ROI from background coefficients at the decoder side. The decision of whether or not a coefficient belongs to the background is taken by comparing the number of decoded bits of the current coefficient with M_s, the nominal maximum number of magnitude bit-planes in subband s. This number is computed from the information contained in the header of the JPEG 2000 codestream and is given by:

$$M_s = G + \epsilon_s - 1 \tag{1}$$

where G is the number of coding guard bits and ϵ_s is the exponent appearing in the definition of the subband's quantization step size:

$$\Delta_s = 2^{R_s - \epsilon_s}\left(1 + \frac{\mu_s}{2^{11}}\right) \tag{2}$$

R_s is the nominal dynamic range of subband s, and μ_s is the mantissa of the quantization step size.

2.3. Maxshift scaling value selection

A valid scaling value for the ROI Maxshift method is a value that ensures no overlap between background and ROI bit-planes. Furthermore, note that if a scaling value s_0 is valid, then any $s_1 \geq s_0$ can also be used for the Maxshift method. A sufficient scaling value is given in the informative part of JPEG 2000 part 1[1]: Provided M_s the nominal maximum number of magnitude bit-planes in each subband s of the image (see Equation 1), a sufficient scaling value is:

$$s_{suf} = \max\left(M_s\right) \tag{3}$$

Here, s_{suf} does not depend on the chosen ROI, but only on the original image bit-depth and the quantization step size in each wavelet subband.

On the other hand, since with the Maxshift method, scaling by s implies dealing with, at least, $2s$ magnitude bits, it could be interesting to find out the optimal (i.e. minimal) scaling value that corresponds to the Maxshift method. This optimal value is simply the maximum number of bits used to represent the magnitude of the background coefficients in all subbands:

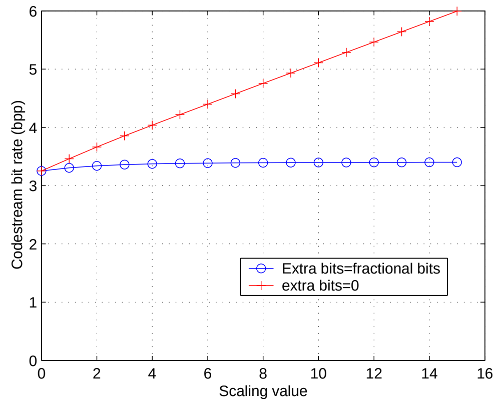
$$s_{opt} = \max_{(u,v,s)\in BG} \lceil \log_2 |w_q(u,v,s)| \rceil \tag{4}$$

where $w_q(u,v,s)$ is the quantization index of the wavelet coefficient in subband s at location (u,v). Note that no background scaling is needed in code-blocks containing no ROI coefficient. Hence, these code-blocks can be excluded from the search.

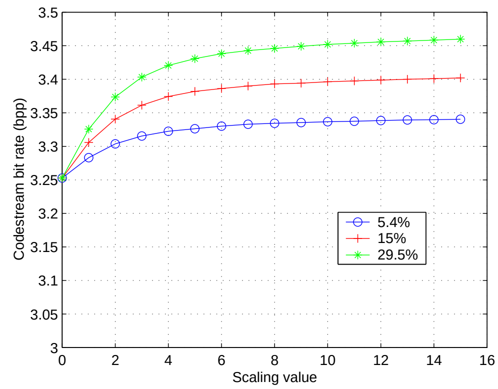
As pointed out earlier, one always has $s_{opt} \leq s_{suf}$. Consequently, s_{opt} is the scaling value that always minimizes the number of bit-planes to encode. On the other hand, this scaling value fully depends on the chosen ROI and can be computed only once the image has been decomposed into wavelet subbands.

Unfortunately, even when using the optimal Maxshift scaling value, the number of bit-planes to encode remains important, especially when the number of wavelet decomposition levels is high. Figure 6 illustrates the evolution of the *optimal Maxshift scaling value* with the increasing number of decomposition levels. We observe that, already at seven decomposition levels, the optimal scaling value is already 16. This means that the entropy encoder has to deal with at least 33 bit-planes. In practice, this

Figure 7. (a) Compressed image bit-rate versus the ROI scaling value for two different types of ROI extra bits padding. (b) Influence of the scaling value on the codestream bit-rate for several ROI sizes: 5.4%, 15% and 29.5% of the original image surface. This last results has been obtained with the zero padding technique for ROI extra bits (i.e *extra bits=0*).



(a)



(b)

implies using structures of 64 bits in order to prevent any bit-overflow, whereas structures of 16 bits would have been enough if no ROI was defined. Additionally, the check of possible bit-overflow can slightly increase the complexity of the encoding algorithm.

2.4. ROI extra bits padding

As said before, the background shift operation implies to encode some ROI extra bits thereafter. These bits, that were originally part of the fractional bits (in the sign magnitude representation) were not originally planned to be encoded. Consequently, and depending on the ROI method, we have to take some precaution when choosing their values. Note also that this choice is only an encoder issue, as JPEG 2000 decoders will generally remove these bits.

With the ROI Maxshift method, it is generally better to set all extra bits to zero. Indeed, if a coefficient has a magnitude of zero (integer part of the magnitude), and one of its extra bit is non-zero, then the decoder will consider it as part of the background, and will wrongly scale-up the magnitude.

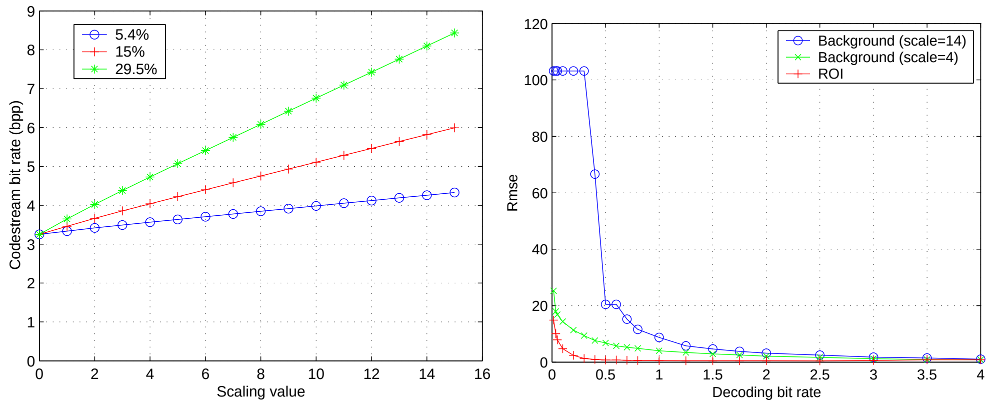
On the other hand, the choice is not that straightforward with the Scaling based method. For instance, Figure 7(a) sketches the compressed image bit-rate versus the scaling value for two different bit-padding techniques. In both cases, the selected ROI is the same and it covers 15% of the original image surface. As one can see, the higher the scaling value and the more relevant becomes the choice of an appropriate bit-padding technique (from a bit-rate overhead point of view). The bit-rate increment is naturally explained by the number of new bit-planes to encode (ROI extra bits and background's LSB's). Here, the curve *extra bits=fractional bits* refers to the case where the fractional bits obtained after scalar quantization are kept, whereas the curve *extra bits=0* is the situation where all extra bits are set to 0. Thus, it appears interesting to construct a codestream with an optimal (i.e. minimum) bit-rate overhead by looking closer at the new bit-planes to encode, and by choosing the extra bits values that are the best compressed by the arithmetic encoder. However, this will significantly increase the complexity of the ROI scaler module. Nevertheless, in most situations, the choice of setting all extra bits to zero appears to be a good trade-off between complexity and coding efficiency.

2.5. Advantages and drawbacks

Figure 7(b) illustrates the influence of the ROI size on the compressed image bit-rate. As noticed earlier, the codestream bit-rate increases with the number of shifted bit-planes because there are more bit-planes to encode. Likewise, this effect is more important for large ROI's. Furthermore, this effect becomes even more significant if a bad bit-padding technique is employed (see Figure 8(a)): for an ROI that covers 15% of the image and with a scaling value of 14 bits, the codestream is around 80% larger than that without ROI*. Thus, if a bad padding technique is chosen, the bad performance can preclude the use of the Maxshift method. On the other hand, with a more optimal padding technique the bit-rate overhead can become negligible.

*This can be seen in Figure 8(a): The bit-rate for a scaling value of 14 (*extra bits=fractional bits*) equal 5.82 bpp, whereas without ROI (i.e. $scale = 0$) the codestream bit rate equal 3.25 bpp

Figure 8. (a) Influence of the scaling value on the codestream bit-rate for several ROI sizes (5.4%, 15% and 29.5% of the original image) with a bad ROI extra bits padding technique. (b) ROI and background RMSE's at various decoding bit-rates and for two different scaling values (4 and 14).



(a) (b)

In the meantime, Figure 8(b) sketches the different types of ROI and background distortion reductions depending on the scaling values, for the Scaling based method: With low scaling values ($scale = 4$), ROI and background distortions decrease together. Whereas for higher scaling values ($scale = 18$), the background distortion starts decreasing once the ROI has reached its full quality (Maxshift method). Here, the choice of the bits padding technique does not influence the ROI and background distortions since the ROI extra bits are discarded by the decoder. The only influence appears on the codestream bit-rates at which the ROI and the background reach their full qualities.

As a conclusion to this section, the following table sums up the main advantages and drawbacks of the two ROI methods supported by JPEG 2000.

	Advantages	**Drawbacks**
Maxshift method (JPEG 2000 part 1)	Handled by all JPEG 2000 decoders ROI shape is implicit Any ROI shape is supported	Large scaling values (higher bit-rates) Possible bit-overflow No background before the whole ROI is decoded
Scaling based method (JPEG 2000 part 2)	Lower bit-rate overhead Finer control on ROI and background distortion reductions	Needs a JPEG 2000 part 2 decoder ROI shape must be explicitly transmitted to the decoder Decoder must derive a ROI bit-mask

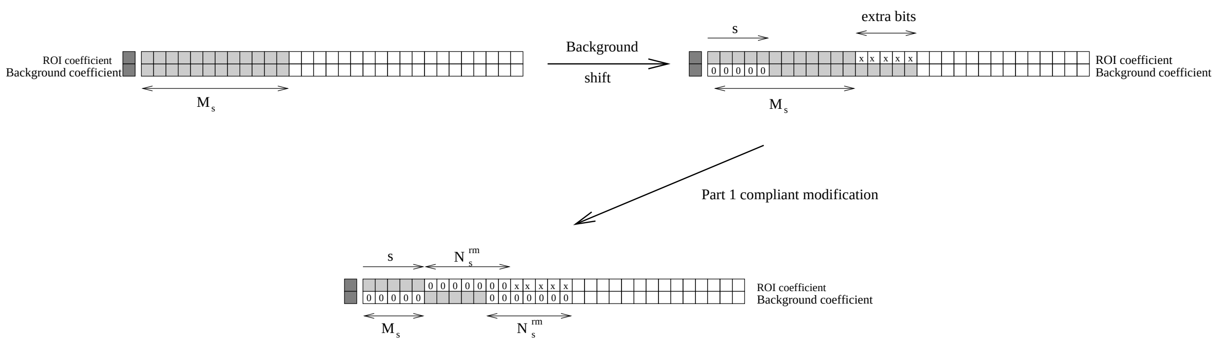
These observations lead us to consider another approach that tries to combine as many advantages as possible from both ROI coding methods.

3. MAXSHIFT-LIKE METHOD WITH LOW SCALING VALUES

In this section, we study the transposition of the ROI Scaling based method in the JPEG 2000 part 1 framework. The goal is to allow any JPEG 2000 decoder to be able to decode some background information before the ROI reaches its full quality. In other words, one wants to find a way of using low ROI scaling values with JPEG 2000 part 1 decoders.

First, one notes that when trying to directly decode a codestream containing an ROI with a low scaling value (i.e. lower than s_{opt}), a JPEG 2000 part 1 decoder handles correctly the ROI coefficients but not always those of the background. In fact, with the ROI de-scaling algorithm presented in part 1,[1] the background coefficients cannot be recognized as such in two cases: The first case is when the background coefficient has some non-zero bits above M_s, and all its bits below M_s are equal to zero. The second case is when the number of decoded bit-planes in the current code-block does not allow the reconstruction of any bit-plane below the last ROI bits.

Figure 9. Maxshift-like ROI scaling method with low scaling values



It is important to note here that, whenever a background coefficient is incorrectly recognized as an ROI coefficient, this one is not shifted upward and the operation is equivalent to a truncation of its last bits. However, truncating the k least significant bits of a coefficient is also equivalent to re-quantizing this same coefficient with a step size multiplied by 2^k. In summary, a JPEG 2000 part 1 decoder perfectly decodes the ROI coefficients of a codestream using a scaling value $s < s_{opt}$, but the decoded background coefficients may have a step size 2^s larger than the expected values.

Since we want to have an algorithm that remains compliant with JPEG 2000 part 1, it is not possible to modify the dequantization algorithm. Hence, a sufficient solution is to modify the quantization step size of all coefficients at the encoder side. Of course, such a modification implies reduction of the final quality of the ROI and can only be considered for applications where visually lossless (not lossless) ROI and background are acceptable.

To take advantage of both ROI methods, we propose the following algorithm that allows a *Maxshift-like* method with low scaling values. Note that this algorithm does not actually re-quantize the wavelet coefficients, but instead it modifies the values of the subbands quantization step sizes in the codestream header.

3.1. Algorithm

Let $M_{b,s}^{BG}$, $M_{b,s}^{ROI}$ and $N_{b,s}^{skip}$ be the maximum number of background and ROI magnitude bit-planes and the number of most significant skipped bit-planes in code-block b of subband s, before any shift operation. The following relation is straightforward:

$$N_{b,s}^{skip} = M_s - \max(M_{s,b}^{ROI}, M_{s,b}^{BG}) \tag{5}$$

Let s be the (low) ROI scaling value considered for the application. After the down-shift of background coefficients in the code-block, the new value for the number of skipped bit-planes ($\tilde{N}_{b,s}^{skip}$) is given by:

$$\tilde{N}_{b,s}^{skip} = M_s - \max(M_{s,b}^{ROI}, M_{s,b}^{BG} - s) \tag{6}$$

Then, to be compliant with the Maxshift method, one has first to remove (i.e. set to zero) all ROI bits overlapping with background bit-planes. Second, one has to modify the quantization step size accordingly. Since the steps are defined at the subband level, the number of removed bit-planes and the quantization step size modification must be shared by all the code-blocks of a same subband.

Hence, the number (N_s^{rm}) of ROI least significant bit-planes to be removed in subband s is:

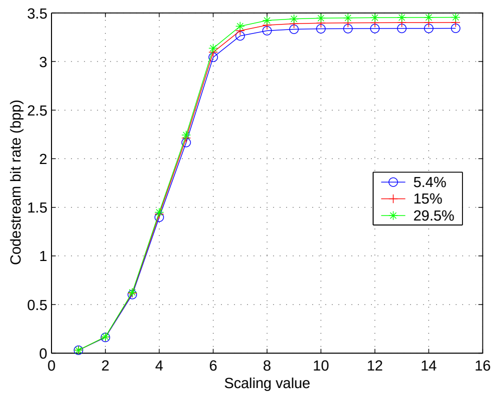
$$N_s^{rm} = \max(\max_{b \in s}(M_{s,b}^{BG} - s), 0) \tag{7}$$

Furthermore, to avoid decoding a background with an higher quality than the ROI, it is worth removing the same number of background least significant bit-planes (i.e. N_s^{rm}). Finally, the new quantization step sizes of subband s is defined by its new exponent $\tilde{\epsilon}_s$:

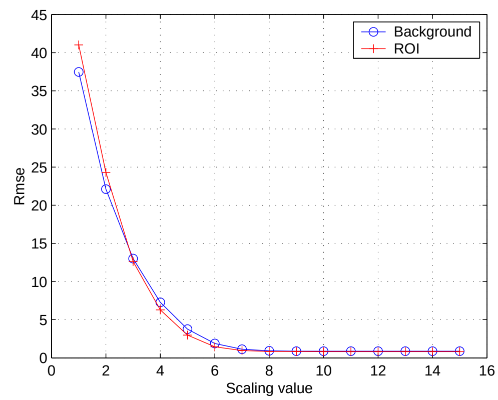
$$\tilde{\epsilon}_s = \epsilon_s + N_s^{rm} \tag{8}$$

Figure 9 illustrates these operations applied on ROI and background coefficients.

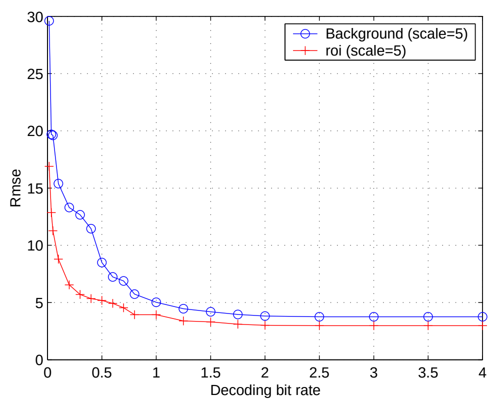
Figure 10. (a) Codestream bit-rate versus scaling value in Maxshift-like method. (b) ROI and background RMSEs after decoding the whole codestream. Background and ROI RMSEs during a progressive decoding: (c) scaling value=5, (d) scaling value=11. (e) Decoded image at 0.4bpp (scaling value=5). (f) Image after decoding the whole codestream (scaling value=5)



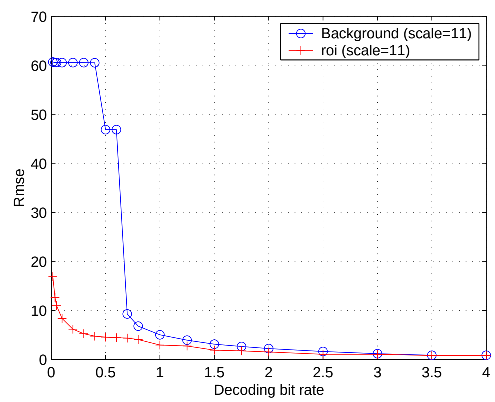
(a)



(b)



(c)



(d)

(e)

(f)

3.2. Results

The algorithm presented above has been applied on the image GoldHill (720x576) where a rectangular ROI has been defined. On the other end, an unmodified JPEG 2000 part 1 decoder has been used to decode the generated codestreams.

Figure 10(a) sketches the bit-rate of the generated codestreams versus the Maxshift-like scaling value for several ROI sizes (5.4%, 15% and 29.5%). As seen previously, the larger the ROI size, the higher the codestream bit-rate. However, since all ROI extra bits are set to zero, this overhead is not very important. On the other hand, compared with Figure 7(b), one notes that the curves behave almost identically for high scaling values (i.e. real Maxshift method), but they are completely different for low scaling values: In the Maxshift-like method, the bit-rate decreases with the scaling value since there is less information to encode (i.e. some ROI and background bit-planes have been removed).

In Figure 10(b), one observes the final ROI and background distortions when varying the scaling value. As expected for low scaling values, the distortions increase (because of removed bit-planes). We deduce from these curves that scaling values under a value of four will certainly not fit in practical applications because too much distortion remains in the final decoded image. Nevertheless, the use of values above six seems to be interesting.

Figure 10(c) and Figure 10(d) illustrate the background and ROI distortion reductions for scaling values of 5 and 11 respectively. In the first case, one sees that both background and ROI distortions decrease at the same time, although the ROI always exhibits a better quality. In the latter case, one already recognizes a behavior similar to the Maxshift method, where the background distortion starts decreasing when the ROI has already reached a good quality. Nevertheless, both ROI and background distortions continue to decrease afterward, proving the fact that the considered scaling value is lower than s_{opt}.

Finally, Figure 10(e) and Figure 10(f) show the resulting image after decoding 0.4 bpp or all the bytes from the whole codestream respectively (scaling value=5). We can see that at 0.4 bpp the ROI has already an acceptable quality, whereas some background information has also been decoded. Furthermore, when decoding the whole codestream, the ROI and background have visually the same quality.

CONCLUSIONS

In this paper, we have revisited the ROI feature in the JPEG 2000 framework. After a brief review of the algorithm defined in the standard, we have given first some indications on how to choose the scaling value in the ROI Maxshift method. We have also studied the opportunity of using an appropriate bit-padding technique after down-scaling the background coefficients. Finally, we have presented an encoding algorithm that allows the use of low ROI scaling values while being compliant with the JPEG 2000 part 1. This algorithm reduces the codestream bit-rate with the counterpart of a slightly decreased final ROI and background qualities. Although very low scaling values are certainly not usable for practical applications (too much distortion), middle ones seem to exhibit very interesting features.

ACKNOWLEDGMENTS

We would like to thank the authors of the Verification Model software and JJ2000 (JPEG 2000 reference software). These codecs have been used to generate the results for ROI methods of part 2 and part 1 respectively. The latter has also been used as the basis for the implementation of the proposed ROI method. Special thanks go to Mr. Ricardo Zannini for fruitful discussions and for the work he did in the framework of JPEG 2000 Region Of Interest, during his diploma project at the Signal Processing Laboratory of the Swiss Federal Institute of Technology.

REFERENCES

1. *JPEG 2000 part 1 Final Draft International Standard*, ISO/IEC JTC1/SC29 WG1 N1890R, August 2000.
2. *JPEG 2000 part 2 Final Committee Draft*, ISO/IEC JTC1/SC20 WG1 N2000, December 2000.
3. D. Santa-Cruz, T. Ebrahimi, J. Askelöf, M. Larsson and C. A. Christopoulos, *JPEG 2000 still image coding versus other standards*, Proceedings of the SPIE's 45th annual meeting, Applications of Digital Image Processing XXIII , vol. 4115, pages 446-454, San Diego, California, Jul. 30-Aug. 4, 2000
4. C. A. Christopoulos, J. Askelöf, M. Larsson, *Efficient methods for Encoding Regions Of Interest in the upcoming JPEG 2000 still image coding standard*, IEEE Signal Processing Letters, Vol. 7, No. 9, Pages: 247-249, September 2000.
5. D. Taubman, *High Performance Scalable Image Compression with EBCOT*, IEEE Transaction on Image Processing, Vol. 9, No. 7, Pages: 1158-1170, July 2000.
6. I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, PA, 1992.