Diego Santa Cruz, Touradj Ebrahimi, Mathias Larsson, Joel Askelöf and Charilaos Cristopoulos. **Region of Interest Coding in JPEG2000 for interactive client/server applications**. In *Proc. of the IEEE Third Workshop on Multimedia Signal Processing (MMSP)*, pages 389-394, September 13-15, 1999.

REGION OF INTEREST CODING IN JPEG2000 FOR INTERACTIVE CLIENT/SERVER APPLICATIONS

D. Santa Cruz, T. Ebrahimi
Signal Processing Laboratory
Swiss Federal Institute of Technology
CH-1015 Lausanne, Switzerland

M. Larsson, J. Askelöf, C. Cristopoulos
Compression Lab, TX/ETX/PN/XML
Ericsson Telecom AB
S-126 25 Stockholm, Sweden

Abstract - This paper presents an approach which allows an efficient on-the-fly decoding of regions of interest of an already encoded image without need for a complete decoding/encoding process. The basic method is compatible with the wavelet based compression scheme adopted for the JPEG2000 standard and has been already adopted in a previous verification model. This technique is specially of value in interactive client/server applications linked through narrowband networks, in which the client can request the server to tailor more efficiently the transmission of the desired information at little processing cost. Simulation results show significant improvement in reduction of transmission time and enhanced flexibility at the expense of a very small complexity and bitrate overhead.

INTRODUCTION

The Joint Photographic Experts Group (JPEG) is currently working towards the definition of the next still image compression standard called JPEG2000. Besides the use of a more efficient state-of-the-art technology to address the problem of compression as in the JPEG algorithm, care has been taken to provide in an efficient way multiple other functionalities insufficiently or not covered in JPEG, useful for today's and tomorrow's applications. The funtionality of interest in this paper is region of interest coding, which is introduced below. The approach presented here is applicable to client/server applications, as well as others.

OVERVIEW OF JPEG2000

JPEG2000 has adopted a wavelet based technology in its compression scheme [1]. This means that the first step in the algorithm is to decompose the input image into a set of subbands via a wavelet transform. Each subband will then contain different frequency components of the information in the original image with the appropriate subsampling. The most common decomposition (i.e. dyadic) is performed by recursively applying a low-pass filter and a high-pass filter to the image in both spatial directions, although other decompositions are also possible. The wavelet transform used can be either a floating- or a fixed-point wavelet, which implies lossy coding due to limited precision, or a reversible integer wavelet, which enables lossless coding.

The wavelet coefficients are explicitly quantized if needed and the quantizer indices, or the integer wavelet coefficients, are losslessly entropy encoded using arithmetic coding of its bitplanes. Each subband is independently encoded, starting at the most significant bitplane in the subband down to the less significant one. Each bitplane is coded in three passes: first the predicted significant (*PS*) bits, then the refinement (*REF*) bits and then the predicted non-significant (*PN*) bits. The coefficients are coded by sign and magnitude, the sign of each coefficient being coded when the coefficient becomes significant. The bits are coded with the CJ arithmetic coder [8], the contexts being determined by the significance of its neighboring coefficients. Each group of bits (PS, REF or PN) in each bitplane forms what is called a *coding unit* (*CU*).

The entropy encoded subbands can be output to a bitstream in several possible forms: one after the other, providing progressive by resolution scalability, or interleaved at the CU level, providing progressive by accuracy scalability. An arbitrary ordering is also possible, where the different CUs are ordered in an application dependant order, providing different types of scalability.

This entropy coding algorithm corresponds to the Verification Model (VM) 3.0B, which is the one on which this approach has been implemented. However, it has changed since the last VM.

REGION OF INTEREST CODING

In many applications where digital images are involved (medical, remote sensing, etc.), there exists within the image, one or more regions of greater interest than the rest. By allowing the background (BG) to be coded with lower fidelity than the regions of interest (ROIs), significant gains can be achieved in terms of compression and hence in storage space and transmission times [6,7].

Region of interest coding happens in two steps: the wavelet coefficients that affect the ROIs must be identified, and then coded with a higher quality by some special means. For the first step, an ROI mask is generated, which is explained below. The approach to use in the second step depends on the application. One possibility is to separately encode ROI and BG coefficients in each subband, using a coarser quantization for the BG coefficients [1]. Another approach [1,7] is to shift the quantization indexes, or integer wavelet coefficients, in the ROIs upwards in bitplanes so that they are coded first.

In the above methods, the ROIs must be defined before the encoding process and cannot be changed afterwards (unless a complete decoding and re-encoding is performed). A third method (proposed in this paper) allows for the definition of new ROIs during the encoding process, or even during a progressive transmission (by using a transcoder), as explained further.

ROI mask creation

When an image is coded with an ROI, it should be possible to reconstruct the entire ROI with better fidelity (at a higher bitrate). It is therefore necessary to identify the wavelet coefficients needed for the reconstruction of the ROIs, so that they can be coded at a higher quality. For this purpose an *ROI mask* is created. This mask indicates which wavelet coefficients belong to an ROI, and to which one if multiple ROIs are defined.

In fact, in order to reconstruct a pixel inside an ROI at a higher quality, all the wavelet coefficients that contribute through the inverse wavelet transform are needed at a higher quality, and thus become part of the ROI mask. The exact pixels that are needed are, of course, transform dependent. If all coefficients in the ROI mask are losslessly encoded, then the ROI can be losslessly reconstructed. This is performed for lines and columns at each decomposition level. The process is then repeated for the remaining levels, until the entire wavelet tree is processed. This can be seen in Figure 1.

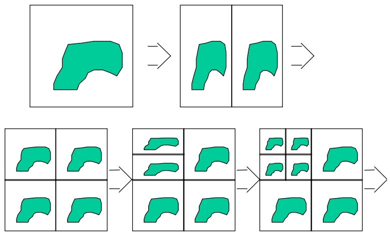
Figure 1: The generation of the ROI mask.

When multiple ROIs are defined, one additional issue arises. Since the ROI mask expands as it is created, there is a possibility for two ROIs that are disjoint in the spatial domain to have some coefficients in common in the wavelet domain. The way to handle these coefficients depends on the application, but the most common solutions are [4,5]: to redundantly encode such coefficients or to them coefficients in the highest quality ROI only.

CLIENT/SERVER APPLICATION

As explained above, there are many situations in which a transmission with ROI coding is desired to reduce delays. However, the ROI geometry and position is not always known until the transmission has begun. This is particularly true in client/server applications where the user defines the ROI once a low quality or low resolution version of the image is available. In order to avoid having to restart the image transmission from the beginning if an ROI with better quality is requested by the user, it is necessary to be able to start the ROI coding at an arbitrary moment during the transmission. This requirement is equivalent to being able to start ROI coding at an arbitrary point in the bitstream. The point at which ROI coding starts is called the *switching bitrate*. The following paragraphs explains how.

First, the wavelet transform is performed, the transformed coefficients are eventually quantized, and the resulting bitplanes arithmetically encoded, as if no ROI was defined. The bitstream header is written to the bitstream and the entropy encoded data starts to being output, in whatever progressive mode has been chosen. When the switching bitrate is reached, a marker (the *ROI switching marker*), the ROI index and the ROI geometry are output to the bitstream. After that the corresponding ROI mask is generated. The ROI switching marker is a byte-aligned code that cannot be emulated by the arithmetic coder, and that signals the switching to ROI mode. The ROI index helps identify the ROI when multiple ROIs, and multiple switches between them, are requested.

Bitplanes that have not been sent in their entirety, in each subband, are arithmetically encoded again skipping all coefficients that do not belong to the ROI. Skipping the non-ROI coefficients is what provides a more efficient transmission. All the contexts of the arithmetic encoder are reset to a uniform distribution before this operation. Note that no requantization or wavelet transform takes place here, as the same quantization indices, or integer wavelet coefficients, are re-encoded.

The re-encoded coefficients are output to the bitstream in the same progressive mode as in the beginning. This continues until the switching bitrate of another ROI is reached, or until all the data for the ROI has been output. In the first case, the same process begins again using a different ROI index. In the second case, another ROI switching marker is output to the bitstream, with a special ROI index that signals non-ROI data. At this point all the coefficients that have not yet been sent are entropy encoded (in much the same way as for ROIs) and output to the bitstream in the same progressive mode, thus switching to non-ROI mode.

When multiple ROIs are used, the coefficients that belong to more than one ROI are redundantly encoded, but only partially. In fact only the bitplanes in each subband that were not sent in their entirety before the first ROI switch are redundantly encoded. As said earlier, a more efficient encoding process can also be implemented at the expense of an increasing complexity on both encoder and decoder sides.

This approach supports decoding the image with a uniform quality, which could even be up to lossless. The operation described here can also be carried out by a transcoder, instead of the encoder. Such a transcoder would only need to entropy decode the bitstream and to re-encode it, according to the ROIs and the switching bitrates. This approach also supports a very flexible functionality for the user, which can request several ROIs, and switch between them. It also supports switching back and forth between non-ROI and ROI mode.

EXPERIMENTAL RESULTS

The scheme described above was implemented in the Verification Model (*VM*), version 2.1, of JPEG2000 [2]. For the sake of simplicity, the ROI shapes supported by this implementation are only rectangular or circular, although the exact same scheme also supports any arbitrary shape for the ROI.

For the examples provided here, one bitstream, in progressive by accuracy mode, is generated for each image, using the ROI coding scheme explained above. The bitstream is decoded at several bitrates and the PSNR values obtained at each one of these, for the ROIs, the BG and the entire image. The images used for the examples are from the JPEG2000 test suite, as well as from the medical testing images provided by DICOM to JPEG2000.

Figure 2 shows the PSNR values for the ‘aerial2’ image, which is a gray-level 2048x2048 8 bit deep aerial picture, using one rectangular ROI covering 28.85% of the area. The image is quantized to achieve a compressed image at 2.0 bpp; the switching bitrate is 0.05 bpp. One can see that before the switching bitrate, both

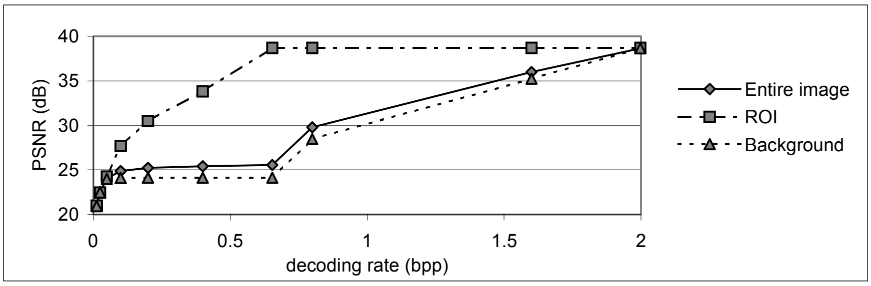

Figure 2: The PSNR values at different decoding rates for the ‘aerial2’ image, using 1 ROI and lossy coding to 2.0 bpp.

BG and ROI have very similar qualities. At the switching bitrate the ROI quality improves much faster, while the BG quality remains constant. Once all the data for the ROI has been coded the BG starts improving, much more slowly than the ROI, until it reaches the same quality as the ROI. The ROI is therefore coded to 2.0 bpp in less than 1/4th the space needed for the whole image, which means that a user would receive the requested ROI at a very good quality in less than 1/4th the time needed if no ROI was used. Since JPEG2000 has far better visual quality than JPEG at very high compression, the switching bitrate of 0.05 bpp provides already an “understandable” image for a user to choose the desired ROI.

Figure 3 shows the PSNR values for an experiment using the medical image ‘MG1’, which is a gray-level 3064x4664 12 bit deep mammogram, using a rectangular ROI, covering 18.68% of the area, and a switching bitrate of 0.05 bpp. Lossless coding is used in this case, which is often a requirement in medical applications. If the example is applied to a telemedicine application using a 64kbit wireless transmission channel, the user would select the ROI after 10.9 sec., receiving a lossless ROI after 5 min. (and a “visually lossless” one long before). In contrast, without ROI the user would have to wait more than 24 min. for the lossless image, which is roughly 5 times the time with ROI coding.

It is worth noting that in the above examples the overheads incurred by the ROI coding are only 1.5% and 0.4% of the compressed bitstream length, respectively.

CONCLUSIONS

As it has been explained above, ROI coding for interactive client/server applications is a very desirable feature in some applications, such as telemedicine, PDAs, etc. The proposed scheme is highly flexible. It allows the user to request an ROI at any moment, new ROIs at any other moment, to switch between different ROIs, and to switch between ROI and non-ROI transmission. It also provides support for lossless coding of ROIs only or of the entire image. These features leave room for server side optimization. In fact, the server can optimize the switching bitrate, as well as the exact geometry of the ROI.

Another very important point is that the complexity of the scheme is very low at the decoder, only the ROI mask has to be generated and some arithmetic coder contexts reinitialized.

At the encoder or the transcoder, complexity is relatively low, the only additional complexity when compared to the decoder is what is necessary to arithmetic decode and re-encode.

Last but not least, the proposed scheme has been accepted for inclusion in JPEG2000's Verification Model 3.0 (B) [1].

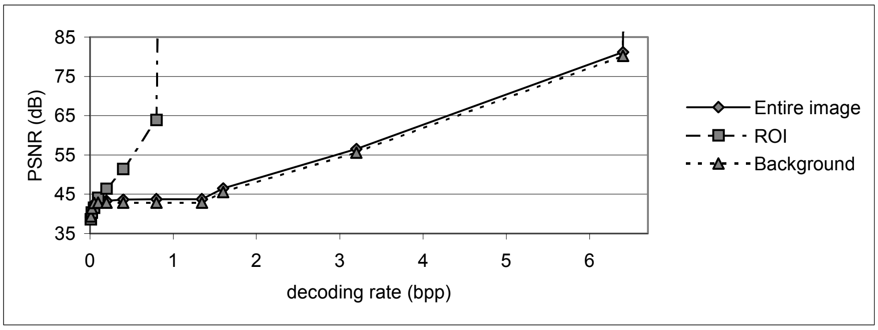

Figure 3: The PSNR values at different decoding rates for the 'MG1' image, using 1 ROI and lossless coding. Note that PSNR values for lossless are infinite (∞).

References

[1] C. Christopoulos (editor), "JPEG 2000 Verification Model 3.0(B)", *ISO/IEC JTC1/SC29/WG1 N1135*, Dec. 01, 1998.

[2] D. Santa Cruz, M. Larsson, T. Ebrahimi, and C. Christopoulos**,** "Report on Core Experiment ROI 1: ROI refinement"**,** *ISO/IEC JTC1/SC29/WG1 N990*, Los Angeles, USA, 02-06 Nov. 1998.

[3] E. Atsumi, N. Farvardin and F. Ono, "Report on ROI CE1: ROI coding from either the beginning or the middle of the encoding process", *ISO/IEC JTC1/SC29/WG1 N1031*, Los Angeles, USA, 02-06 Nov. 1998.

[4] D. Santa Cruz, T. Flohr, M. Larsson and T. Ebrahimi, "Report on Core Experiment ROI 4: Evaluation of ROI coding with disjoint regions", *ISO/IEC JTC1/SC29/WG1 N992*, Los Angeles, USA, 02-06 Nov. 1998.

[5] W. Li, Jim K. and M. Boliek, "JPEG200 ROI Proposal", *ISO/IEC JTC1/SC29/WG1 N990*, Copenhagen, Denmark, Jul. 1998.

[6] D. Nister and C. Christopoulos, "Lossless Region of Interest with a naturally progressive still image coding algorithm", *IEEE International Conference on Image Processing (ICIP-98)*, Oct. 4-7, 1998 Chicago, Illinois, USA.

[7] E. Atsumi and N. Farvardin, "Lossy/lossless region-of-interest image coding based on set partitioning in hierarchical trees", *IEEE International Conference on Image Processing (ICIP-98)*, Oct. 4-7, 1998 Chicago, Illinois, USA.

[8] D. Speck, "Binary arithmetic coder", *ISO/IEC JTC1/SC29/WG1 N299*.