Fully Scalable Video Coding with Packed Stream

M.F. López, S.G. Rodríguez, J.P. Ortiz, J.M. Dana, V.G. Ruiz and I. García
Computer Architecture and Electronics Dept.
University of Almería, Almería, Spain

ABSTRACT

Scalable video coding is a technique which allows a compressed video stream to be decoded in several different ways. This ability allows a user to adaptively recover a specific version of a video depending on its own requirements. Video sequences have temporal, spatial and quality scalabilities. In this work we introduce a novel fully scalable video codec. It is based on a motion-compensated temporal filtering (MCTF) of the video sequences and it uses some of the basic elements of JPEG 2000. This paper describes several specific proposals for video on demand and video-conferencing applications over non-reliable packet-switching data networks.

Keywords: Video, compression, scalability, JPEG 2000, video on demand, video-conferencing.

1. INTRODUCTION

The digitalization of video signals can generate very high data-rates if an inefficient coding system is used, typically more than 1 Gbps. This huge bit-rate can be reduced by removing the visual and data redundancies which frequently are found in an image sequence. Thus, after the digitalization, the resulting PCM (Pulse Code Modulation) video data must be compressed using a video compressor. Usually, the bit-rate associated to the video stream managing systems is 100 times smaller than the required for working with their original PCM representations. It means that the use of video compression techniques helps to reduce the transmission and storage resources.

The compression of an image (or of a sequence of images) always implies some increment in the cost of the access to its visual content, specially in terms of computation resources, and in some cases an irreversible loss of visual information. To minimize these problems, a progressive or incremental way of representation of the image sequences known as *scalable video coding* can be used. A scalable video codec allows us to obtain (without transcoding) the following three types of scalability:

1. **Temporal scalability** or image-rate scalability can be used to select the image-rate of the video decoding. Thus, for a given bit-rate data transfer between the coder and the decoder, the decoder can choose between a good perception of the motion in the sequence (high image-rate) or a good perception of the details of a subset of images (low image-rate).

2. **Spatial scalability** allows to decode only a reduced resolution (subsampled) version of the images (**resolution scalability**) or even only a ROI (Region of Interest) of each image (**ROI scalability**). In the former case, a low performance decoder can retrieve only the resolution that it is capable to decompress. In the latter case, under some bit-rate constrains, a user typically increments the quality of a region whose position in the image could change dynamically to follow the movement of an interesting object.

3. **Quality scalability** allows to decompress by continuously increasing the number of bits per pixels of the images. This scalability is widely used for remote decoders when the available bit-rate is variable and obviously unknown at the coding time. Thus, the higher the available band-width is, the better the quality of the displayed images is.

Scalability is an artificial feature in the current standards for video coding (e.g. MPEG-{1,2,4} and H.26{1,3,L}). When scalability is available (most of cases, this happens only partially), it generally produces a significant decrease of the compression ratios.

Standards as MPEG and H.26 are based on the application of differential prediction coding schemes (such as motion compensation) for reducing the temporal redundancy and on the use of the DCT to diminish the

spatial redundancy. In the field of still image coding, a recent standard developed by the ISO/ITU-T has opened new scopes (expectations) for imaging applications due to its excellent possibilities of scalability. This encoding system is called JPEG 2000 and is based on the DWT (Discrete Wavelet Transform).

Motion JPEG 2000 is the direct extension of this standard in the field of video coding. It can compress a sequence of images taking advantage of the scalabilities previously described. Unfortunately, Motion JPEG 2000 does not remove the temporal redundancy of an image sequence and for this reason, the compression ratios are poorer than those reached by the motion compensation-based systems.

Basically, our video codec is based on JPEG 2000 and therefore it takes advantage of all the scalability properties. Additionally, it is able to eliminate the temporal redundancy using a differential prediction-based coding scheme which includes a motion compensation module. The motion in the scene is estimated and compensated at the encoder. The motion information (thorough a set of motion vectors) and the prediction errors are transmitted to the decoder. The decoder only performs the motion compensation and can restore without error the original image sequence.

In order to maintain the decoder's complexity low enough, the motion compensation system uses a block-based approach. For reducing the typical blocking artifacts produced by the non-overlapped division of the images into blocks, the motion compensation is performed in the 2D wavelet domain. The prediction error is a low entropy image sequence in the wavelet domain. The prediction error is compressed using the EBCOT (Embedded Block Coding with Optimized Truncation) algorithm.

Our codec's scope includes (i) Video on Demand (VoD) and Real-Time Video (RTV) transmission (e.g. video-conferencing) using both reliable and error-prone transmission channels (for wireless links and the Internet); (ii) lossless encoding scenarios as medical and satellite imaging; (iii) video capturing and edition, etc. Basically our codec can be applied to the same applications that Motion JPEG 2000 and additionally is useful for those that require lower bit-rates.

2. WAVELET-BASED VIDEO ENCODING TECHNIQUES

The DWT has proved to be an excellent decorrelation tool for images, even better than the DCT.[1] Another advantage of the DWT is the smooth reconstructions obtained when only a portion of the wavelet information (by means of wavelet coefficients) is used. These factors have produced the research community to be interested in the application of the DWT to the field of video compression.

One of the first works[2] in this direction was based on the idea that an image sequence can be processed as a 3D signal. The image sequence is divided into GOPs (Groups Of consecutive Pictures) and each of them is transformed using the 3D-DWT and compressed by an embedded entropy coder as SPIHT. Obviously, the main advantage of this technique is its simplicity. Nevertheless, the compression ratios and the quality of the video reconstructions are not very good. The main reason is that the filters designed for the DWT are not suitable to decorrelate the video in the temporal domain because the temporal domain is not smooth enough. When a small amount of information is used to decompress a sequence of video, unpleasant ghosting artifacts are generated by the movement of the objects (similar to a picture that has been taken using a long exposure time). A way to improve the overall performance of the this technique consists of an alignment of the GOP images as a previous step to its transformation to the wavelet domain. This alignment increases the temporal redundancy[3] and so helps to ameliorate the compression performance. It can be seen that this approach is basically a block-based motion compensation system, although only a large block (as large as the whole image) is used. The compressed stream is built by a single motion vector for each image of the sequence plus the entropy coding of the wavelet coefficients.

A straight forward way to improve this alignment + 3D-DWT technique is the application of a block-based motion compensation differential encoder followed by a 3D-DWT and an entropy codec.[4] The main disadvantage of this kind of codecs is caused by the low performance of the wavelet filters when they are applied to the prediction error. These residual images usually show blocking artifacts where most of the wavelet filters do not work very efficiently. To minimize this problem (clearly visible in reconstructions), a mesh-based motion estimation (ME) algorithm or other more complex algorithms have been proposed.[5] Nevertheless, the compression ratios are not

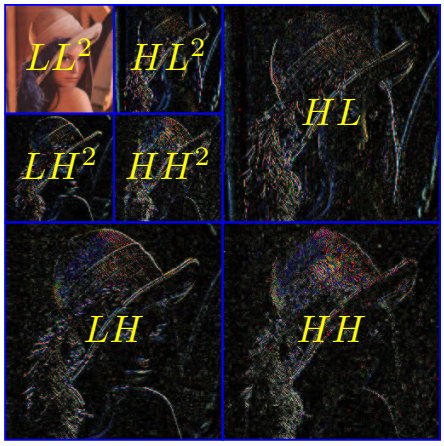

Figure 1. 2D-DWT of *lena* with $r = 2$ levels.

optimal because the DWT is applied to non smooth signals. Additionally, note that the compressed stream must also store the motion fields.

A better way to take advantage of the excellent work that wavelet transforms perform consists in applying the compensation of the motion after the wavelet decomposition.[6] This technique, usually named In-Band Motion Compensation (IBMC in the sequel), computes the residual images on the wavelet domain instead of the image domain. The main advantage of the IBMC video codec is its high visual quality for a partially decoded signal. Although IBMC video coding uses blocks to build the predictions, the blocking visual effect does not appear in the spatial (image) domain. The video codec described in this work is actually an IBMC system.

3. THE WAVELET DOMAIN

The DWT is a transform that allows the representation of digital signals in a domain called the wavelet domain. The wavelet coefficients are obtained by filtering the input signal S using a two-channel filter bank whose outputs are decimated by a factor of two. One of the filters is a low-pass filter and its decimated output is commonly denoted by L. The other is a high-pass filter and its decimated output is indicated by H. These filters are designed in such a way that the information in S that can not be found in L, is stored in H. It means that this pair of filters forms a perfect reconstruction filter bank and the DWT is totally reversible. Note also that, due to the decimation operator, the total number of samples remains constant in the wavelet domain, although the number of samples in each sub-band is half. This is the definition of the 1-level DWT of a 1D signal.

The 2D-DWT is separable. So, for the 2D case, the 1-level 2D-DWT can be computed applying the 1-level DWT to rows and columns of the image. The result is an alternative representation of the image S described by the following four sub-bands: LL, LH, HL and HH (see Figure 1). Thus, the r-level dyadic 2D-DWT is the result of the application of the 1-level 2D-DWT to the LL^{r-1} sub-band. In this new domain (the r-level 2D wavelet domain), $r + 1$ spatial resolution versions of the image can be recovered using the inverse DWT. Notice that the LL^r sub-band is a filtered and subsampled version of the original image with a scale factor of 2^{-r}.

4. MOTION COMPENSATION IN THE WAVELET DOMAIN

Our video codec uses IBMC in order to reduce the temporal redundancy in a GOP. Every image (in the wavelet domain) is motion compensated using a block-based scheme.

The task of a Motion Compensation system (MC) is to produce a prediction image. This prediction image will be subtracted from the current image to obtain a residue image which usually exhibits lower entropy. This prediction image is built with the information taken from two images which have been already decoded at the decompressor. In the sequel, these images will be denoted as reference images. In the following two subsections the prediction step and the method for selecting the reference images will be described.

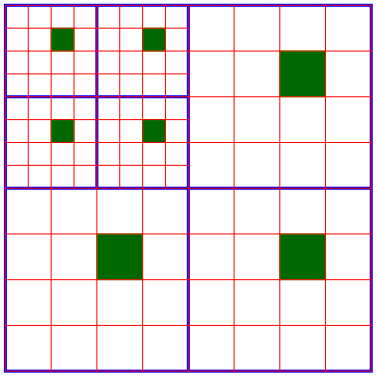

Figure 2. Block partitioning of a 2-level wavelet image used by the MC system.

4.1. Generation of the predictions

The prediction image (which is initially "empty") is divided into square blocks. The size of a block depends on the resolution level it belongs (see Figure 2). More exactly, if $N \times N$ is the block size in the spatial domain, the size of the blocks in the DWT domain at the resolution level r is $N/2^r \times N/2^r$. Notice also that if the image in the spatial domain is divide into $B \times B$ blocks, then the number of blocks in each sub-band is also $B \times B$. The number of motion vectors used for generating an image prediction is $B \times B$. It means that at the DWT domain every motion vector is related to the set of blocks representing a single block of spatial domain (see the fill block of Figure 2).

The wavelet domain described in Section 3 is not shift invariant. This means that a block that is identical in two images in the spatial domain can have a different representation in the wavelet domain when the position of the block is different. To overcome this drawback we can use the *redundant* wavelet domain[7] instead of the *critically sampled* wavelet domain. The redundant domain is the result of removing the subsampling operation from the standard DWT to produce an over-complete representation of the signal. We use the term of "redundant" because the transformed signal is represented several times in this domain. All the representations are equivalent because they store the same information.

The shift invariance of the DWT is guaranteed if the spatial sampling rate is identical for all the resolution levels. The resulting sub-bands have a size equal to the size of the original input signal. Thus, at the resolution level r of the redundant 1D-DWT, there exist 2^r phases. Each phase of a resolution level is the result of choosing even or odd sub-sampling at the previous resolution level. In the 2D case, at the resolution level r, 4^r phases are generated.

For the task of MC, it is necessary to evaluate the phase and location in the reference image related to a block of the current image in the critically sampled wavelet domain. Lets d the motion displacement over any of the two possible directions (vertical or horizontal) that has been calculated by the ME system using the original resolution of the images. The phase ρ and the displacement δ of the reference block are related to the spatial resolution level of the DWT r by:

$$\phi = d \bmod 2^r; \quad \delta = \lfloor d/2^r \rfloor \tag{1}$$

The motion vectors also include information about which reference image has the best match or if any of the reference images are not good enough. In any case, for the image prediction we differentiate three different types of blocks: I (Intra), P (Previous) and N (Next) (see Figure 3). A P-block is used when the best match is located in the previous (past) reference image. A N-block when it is in the next (future) reference image. An I-block is used when this region can not be compensated with any of the reference images.

4.2. Temporal decorrelation of a GOP

Our MCTF scheme is performed for every picture in a GOP, except for the first one which is not compensated (it is an intra-image where all the blocks are I). In order to obtain maximal temporal scalability, pictures are

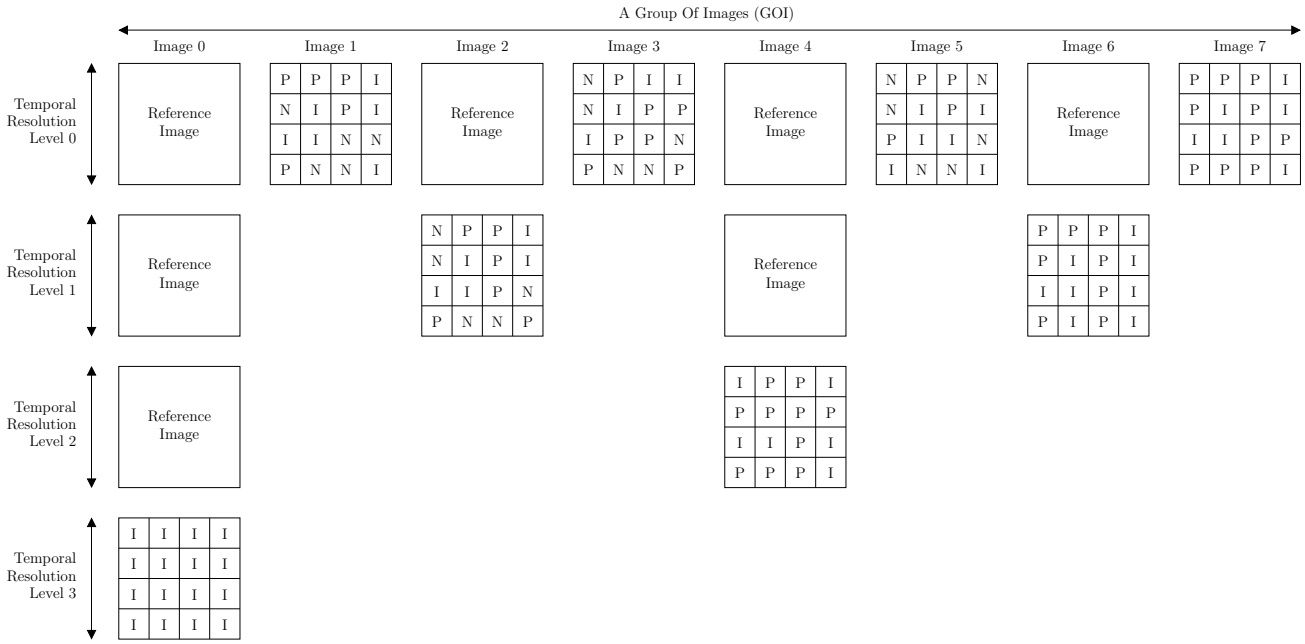

Figure 3. An example of a motion-compensated prediction for a 8-images GOP (only one sub-band is showed).

processed following a pattern that allows the encoder to generate as many temporal resolution levels as possible (see Figure 3). The lowest level (3 in Figure 3) is only composed by a I-picture. The next level includes a picture located in the middle of the GOP and so on. Notice that the last picture in a GOP can only generate I or P blocks. The decompressor receives first the I-picture in the current GOP. I-pictures are self-decoded because they do not depend on any other pictures in the GOP. When more pictures are going to be decoded, the decompressor must choose the amount of data to be taken from each picture.

5. THE EBCOT PARADIGM

EBCOT is the entropy encoder that the JPEG 2000 standard uses to encode images in the wavelet domain.[1] EBCOT is the coder we use to compress the residual images in the wavelet domain. We have selected this algorithm as opposed to other possibilities such as SPIHT,[8] because of three fundamental reasons: (1) its excellent performance has been demonstrated already in the JPEG 2000 and the Motion JPEG 2000 standards; (2) it allows us to process the images in-line, reducing considerably the memory requirements of the video encoding system and (3) it produces a compressed stream that has spatial/ROI and quality scalability, at the same time, without using any transcoding procedures.

In an EBCOT-based encoding system, a wavelet image is first divided into square blocks called *code-blocks* (see Figure 4). Typical sizes of code-blocks are 32×32 or 64×64 coefficients. Then, EBCOT compresses each code-block separately using a progressive coder. Next, neighbor code-blocks are grouped to create the so called *precincts* (see Figure 4). In order to improve some kind of scalabilities, the size of a precinct can be chosen as a function its own decomposition level. For example, the precinct partition shown in Figure 4 is optimal for improving the spatial and quality scalabilities because the wavelet coefficients are transmitted by resolutions and/or bit-planes (layers in the JPEG 2000 slang). On the other hand, if we are interested in a good ROI scalability the precincts should be as small as possible (e.g. one code-block by precinct) for any spatial resolution.

Finally, the bit-stream of the code-blocks of each precinct is divided in l segments of data, where l is the number of quality layers selected for the encoding process. The size of each segment will depend on the contribution of the code-block to the total minimization of the distortion of the reconstruction. The bit-streams of each code-block of each precinct, associated to the same quality layer, are packed into a *packet*. So, by definition,

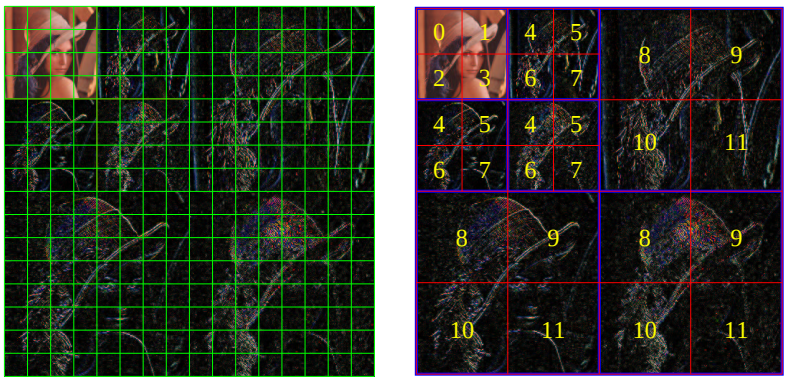

Figure 4. A code-block partition example (left) and a precinct partition example (right) for *lena*.

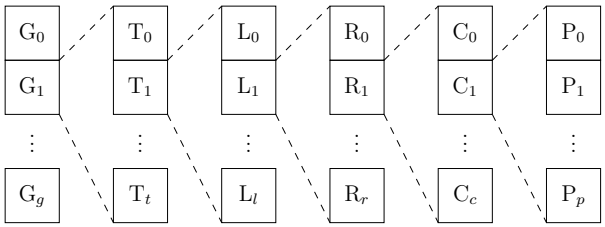

Figure 5. The GTLRCP progression.

a packet is the contribution (that could be zero) of a precinct to a quality layer, for some resolution level. By definition, packets determine a set of optimal truncation points[1] in the stream.

6. SCALABILITY OF THE COMPRESSED STREAM

A major issue in our design has been to optimize the scalability of the compressed stream that is organized as a collection of packets. The order in which packets are placed into the stream depends on the compression parameters, but this order can be changed easily because information about the location of the packets into the stream is also stored in it. As an example, in Figure 5 we have represented the GTLRCP ordering, where G stands for a GOP index, T for temporal resolution, L for quality layer, R for spatial resolution, C for color component and P for precinct. This progression is the most suitable when we are looking for temporal scalability of the video using its original resolution. This is mainly due to packets at every temporal resolution level of the GOP are stored by quality layers. Other useful progressions are: GTRLCP when we prefer first temporal scalability and second spatial scalability, TGLRCP when we prefer temporal scalability at the video level instead of at the GOP level, etc.

In VoD applications, when the band-width of the link forces the truncation of the stream, the decoder can select which packets should be decoded in order to satisfy the requirements of the visualization. The most suitable configuration at this context is a client/server architecture. For example, a client (a simple decoder) can sends to the server (a packet reorganizer) information about the maximum spatial resolution and the maximum number of images per GOP that it needs. The server must only select which packets should be sent (the progression) and the order of importance (the most at the beginning). In other example, the client could select which packets are going to be transmitted using some dynamic algorithm. Thus, the decoder could retrieve a moving ROI that appears in the scene with a better quality than the rest of the picture.

For real-time transmission of video over a link with variable band-width (such as video-conferencing) the encoder should minimize the latency of the visualization at the receiver. The reduction of the GOP size facilitates minimizing the latency. Although this option can decrease the compression ratio, it has the advantage of improving the error resilience, something that is important in this context. For this kind of applications, we

would like to highlight that the selection of a suitable packet progression is necessary. In this sense, any of the GT-subordinated progressions are appropriate.

7. ABOUT ERROR RESILIENCE

In the context of real-time data transmission over non-reliable links some additional properties must be incorporated to the video codec. This is specially true because the use of a reliable protocol, such as the TCP (Transmission Control Protocol), is an inadequate solution. For this reason, our proposal incorporates several error-protecting techniques that are described in this section.

The first kind of techniques are oriented to conceal and restore (if possible) the errors of the received data. The second class of techniques are designed to reduce the impact of the use of erroneous data or even the lost of blocks of data (usually entire packets). This last situation is very frequent in packet-switched networks (such as the Internet) when routers are congested.

All the error-protection techniques used for implementing our video codec have been inherited from JPEG 2000. These techniques include error detection, the conceal of the erroneous data and resynchronization with the encoder.[9] Notice that this is only done at the image level.

These techniques are insufficient for real-time video transmission because the errors in one image can affect to a large number of free error images. Nevertheless, their reconstruction is not possible because it depends on an erroneous image. To overcome this problem, our codec uses other three mechanisms that are specific for video coding: (1) the reduction of the length of the GOP, (2) the temporal interleaving of images and (3) the encoding of the low frequency sub-bands without differential coding. As should be expected, all these techniques reduces slightly the compression ratio.

By definition, every GOP of the compressed sequence can be decoded independently. Therefore, the reduction of the number of images of the GOP helps to minimize the propagation of the errors and also to reduce the latency of the encoding system. Both strategies are necessary in real-time video transmissions.

The dependency between the images of a GOP (see Figure 3) exhibits error resilience because the error propagation in some cases will only affect to a few pictures in that GOP. The number of pictures suffering the error propagation will depend on the temporal resolution level where the error appears.

Finally, we propose to increase the error resilience applying differential MC-based coding only to the high frequency sub-bands. Thus, if errors produced are due to a partial or complete loss of the I-image, the rest of the GOP can be still decoded at low spatial resolution.

8. ENCODING OF THE MOTION FIELDS

The motion compensation scheme used in our proposal generates one motion field for each image of the sequence that is compensated. The number of vectors in each field depends on the size of the images and the blocks. For the case of CIF (Common Interchange Format) video sequences (352×288 pixels/image), and blocks of 16×16 pixels, each field contains 22×18 vectors. Notice that although we can use two reference images to compensate the current one, no bidirectional prediction is performed.

To compress this data we have chosen a static arithmetic code. We have selected a static system because the probabilities of the displacements are a priori known. On the other hand, we have used a non embedded encoding because the amount of data is small enough. The compressed motion fields are multiplexed with the image residuals in order to minimize the delay of the decoding. This can be done sending at first the corresponding motion information for each image.

9. THE CODEC

In this section we present a description of our video codec by means of a block diagram that shows how the different functional blocks are interconnected.

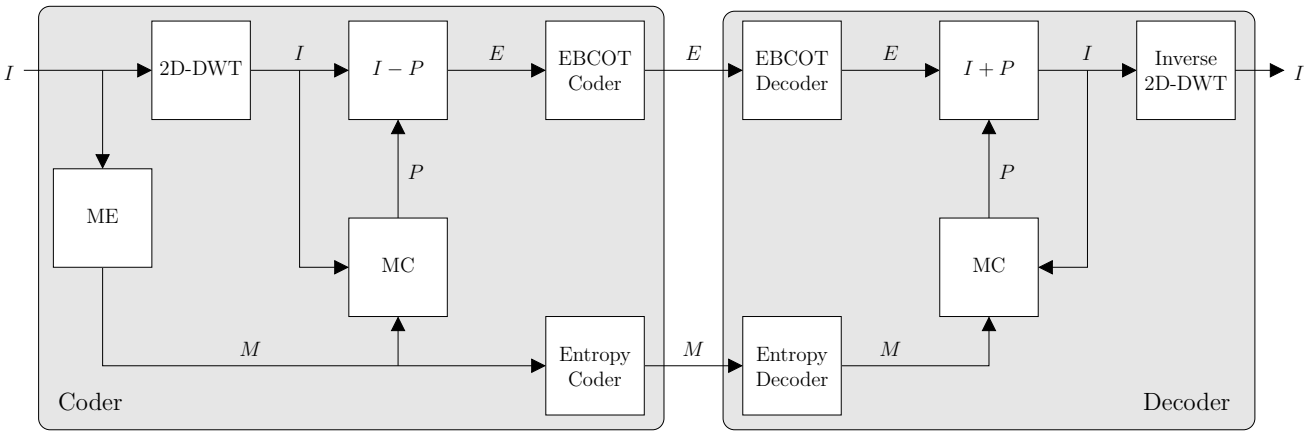

Figure 6. A block diagram of our video codec.

9.1. The coder

It can be seen in Figure 6 that our video coder is composed by six blocks. A sequence of images I is the input of the system and a sequence of packets with the residuals E and motion fields M is the output. These packets are multiplexed in the time as has been described in Section 8.

Before any transformation or coding of the I image sequence, we compute the motion fields using a standard integer-pixel accuracy block-based ME technique on the spatial domain. This task is carried out by the ME module and its output is the sequence of motion fields M (see Figure 6).

In the following step each image $I[\cdot]$ of the image sequence is transformed into the critically-sampled wavelet domain using the 2D-DWT (see Section 3). In the 2D-DWT module, the information contained in I is not modified.

The motion fields M are used by the MC module to generate a sequence of prediction images P. These images are represented in the critically-sampled wavelet domain as it was explained in the Section 4. Next, the P sequence is subtracted to the I sequence generating a sequence of error images or residuals E.

Finally, both sequences E and M are encoded. The first one using EBCOT and the second one using arithmetic coding (see Sections 5 and 8).

9.2. The decoder

The decoding system basically is, as could be expected, the set of the inverse operations performed at the encoder but in the reverse order (see Figure 6). The sequence of images with the prediction errors is decoded. The motion fields are decoded and used to generate the sequence of predictions. The residuals and the predictions are added to generate the image sequence and finally we compute the inverse 2D-DWT.

Obviously, if the band-width or storage resources are limited, the decoder is not able to decode all the compressed stream. In this case, a prediction drift error will appear in a spread way along the time as described in Section 7.

Notice that the complexity of the decoder is smaller than the encoder because the ME is not performed. This is essential for applications where the video sequence is compressed one time and decompressed many times.

10. EXPERIMENTAL RESULTS

The performance of our codec has been compared to two other video coding systems: MPEG-4[10] and the Redundant Wavelet Multihypothesis (RWMH)[4] scheme.

We have used the FFmpeg implementation of MPEG-4.[11] In our experiments, the generated MPEG-4 streams do not provide any scalability (it uses IPPPPPPPPPPP scheme). Temporal, Spatial/ROI and quality

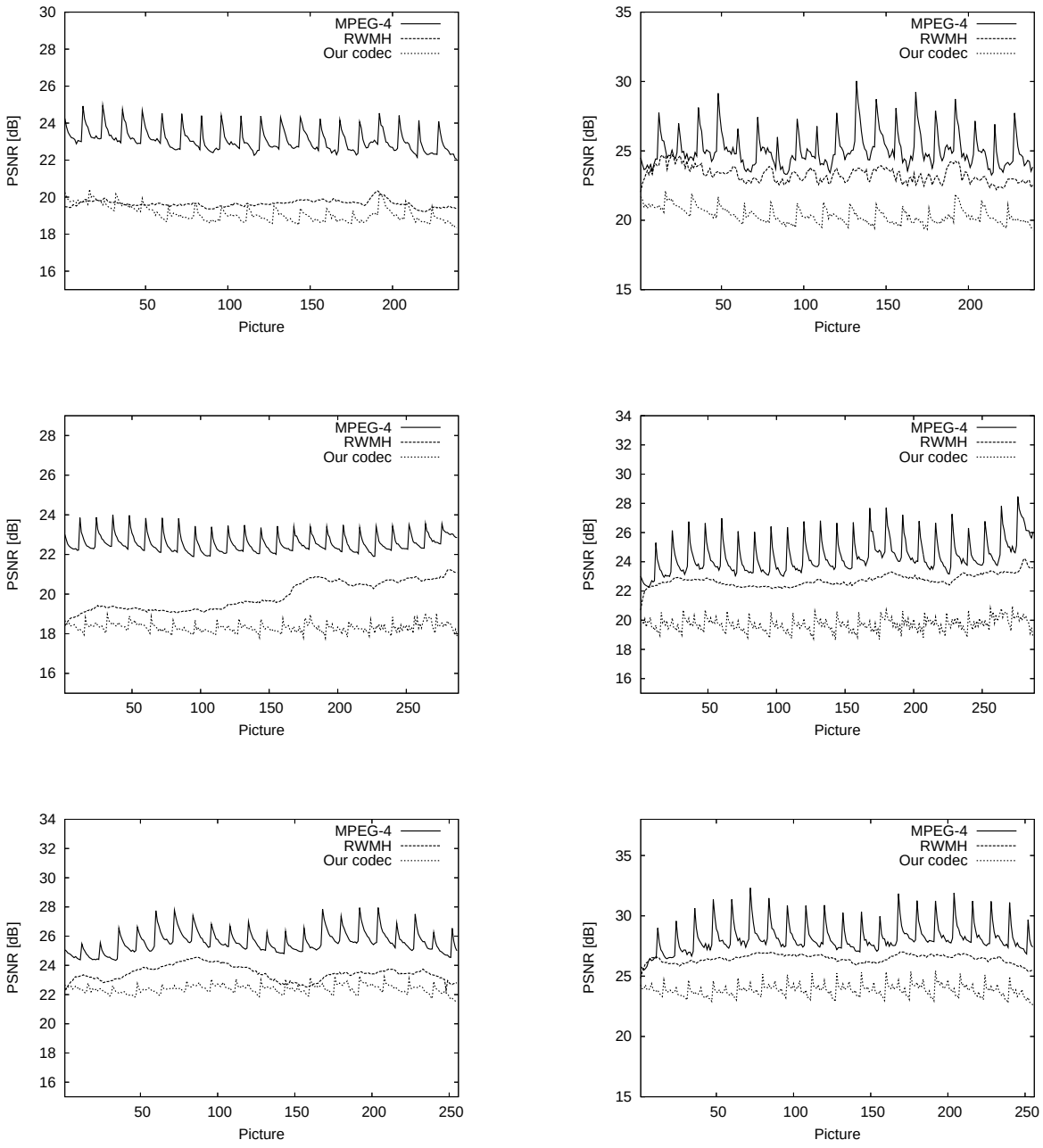

Figure 7. Performance comparison of luminance PSNR for MPEG-4, RWMH and our codec for the *flowergarden*, *mobile* and *tempete* sequences (top to bottom). On the left at 256 Kbps; on the right at 512 Kbps.

scalabilities are unavailable. The RWMH codec is a utility provided by the QccPack project.[12] It provides quality scalability but not temporal (it uses the IPP...P-picture scheme, i.e. the size of the GOP is the size of the sequence) nor spatial. Our codec is fine granularity fully scalable. The compressed stream exhibits temporal, quality and spatial scalabilities.

The performance evaluation of our codec has been carried out using the following parameters: (1) GOP size = 16 pictures; (2) lossless coding; (3) 16 quality layers; (4) the number of bits for every compressed picture is equal for all the pictures in the GOP except for the first one; (5) the decoded pictures have been obtained using the GTLRCP progression; (6) the lowest resolution information of every picture has not been included in the MCTF scheme: so, our codec loses some coding efficiency to achieve error resilience.

Experiments have been performed using three test sequences with CIF format (352×288) at 30 Hz: *flower-garden*, *mobile* and *tempete*. In Figure 7, the luminance PSNR values of the decoded pictures of every image for two bit-rates (256 Kbps and 512 Kbps) have been depicted. The PSNR values obtained by MPEG-4 can be considered as the upper bound for coding efficiency. This is mainly due to its complete lack of scalability features. Compared to RWMH, our codec obtains similar performance at 256 Kbps, but lower at 512 Kbps.

Figure 8 shows the decoded pictures for a subjective comparison. It can be clearly seen that MPEG-4 shows better reconstructions. The scalability features of RWMH and our codec cause of a poorer visual quality of the decoded images. It can be observed that the details of the pictures obtained by our codec are better than those reconstructed by RWMH although the low spatial frequencies information is worse at some places of the picture.

11. CONCLUSIONS

A new fully scalable video codec based on MCTF and JPEG 2000 has been described. In this work some techniques for error resilience has also been designed. Our codec provides fine granularity on temporal, quality and spatial/ROI scalabilities, although it exhibits a slightly lower performance than RWMH and MPEG-4. Thus, we think our codec is able to be improved tinning parameters or changing slightly the coding scheme. Our future work involves the optimization of the parameters of our codec in order to improve the coding efficiency.

REFERENCES

1. D. Taubman and M. Marcellin, *JPEG 2000 Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, 2002.
2. B.-J. Kim and W. Pearlman, "An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees," in *Proceedings of the IEEE Data Compression Conference*, pp. 251–260, 1997.
3. D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Transactions on Image Processing* **3**(5), pp. 572–588, 1994.
4. Y. Wang, S. Cui, and J. Fowler, "3D video coding using redundant-wavelet multihypothesis and motion-compensated temporal filtering," in *Proceedings of the IEEE International Conference in Image Processing (ICIP)*, pp. 775–778, 2003.
5. A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression," *IEEE Transactions on Image Processing* **12**(12), pp. 1530–1542, 2003.
6. Y. Andreopoulos, M. van der Schaar, A. Munteanu, J. Barbarien, P. Schelkens, and J. Cornelis, "Fully-scalable wavelet video coding using in-band motion compensated temporal filtering," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, **3**, pp. 417–420, 2003.
7. M. Shensa, "The discrete wavelet transform: weeding the Á Trous and Mallat algorithms," *IEEE Transactions on Signal Processing* **40**(10), pp. 2464–2482, 1992.
8. B.-J. Kim, Z. Xiong, and W. Pearlman, "Low bit-rate scalable video coding with 3D set partitioning in hierarchical trees (3D SPIHT)," *IEEE Transactions on Circuits and Systems for Video Technology* **10**, pp. 1374–1387, 2000.
9. ISO/IEC, *Information Technology – JPEG 2000 Image Coding System – Part 1: Core Coding System (15444 Standard)*, 2000.
10. S. Battista, F. Casalino, and C. Lande, "MPEG-4: a multimedia standard for the third millennium," *IEEE Multimedia* **7**, pp. 76–84, 2000.

Figure 8. Reconstructed pictures with non-scalable MPEG-4, RWMH and our codec (top to bottom) for the second picture of the *flowergarden* sequence. On the left at 256 Kbps; on the right at 512 Kbps.

11. "FFMEG Multimedia System." http://ffmpeg.sourceforge.net.
12. J. Fowler, "Qccpack: An open-source software library for quantization, compression, and coding," in *Proceedings SPIE 4415*, pp. 294–301, 2000.