Multiple Description Coding Schemes for the H.264/AVC Coder

O. Campana, A. Cattani, A. De Giusti, S. Milani, N. Zandonà and G. Calvagno

{ottavio.campana,andrea.degiusti,simone.milani,calvagno}@dei.unipd.it,
andrea.cattani@telsey.com, nicola.zandona@st.com

Abstract- The recent development of multimedia communications across unreliable channels has brought the need for robust coding techniques, such that a partial loss of information does not necessarily imply the loss of the whole video sequence. The implementation of Multiple Description Coding (MDC) schemes based on the H.264/AVC coding standard provides an affordable solution to this problem.

This paper presents some MDC schemes that differently exploit the correlation (spatial or temporal) of a video sequence. Several efforts were made in order to develop H.264/AVC standard-compliant MD coders. Finally, the capabilities of the different schemes are evaluated and compared.

Keywords- multiple description coding, video coding, transform coding, H.264, MPEG-4 part 10, rate control, buffer control, coefficients statistics.

I. INTRODUCTION

During the last years, many Multiple Description Coding (MDC) schemes have been proposed [1, 2, 3, 4]. These MDC schemes aim to reduce the effects of transmission errors by coding the video signal in two (or possibly more) correlated descriptions, which are independently transmitted over separate channels. At the receiver, whenever all the descriptions arrive without errors, the encoded sequence is correctly reconstructed, while, in case of transmission errors, the lost information is partially recovered from the received descriptions.

This article provides an insight on the capabilities of some MDC schemes implementing the H.264/AVC coding standard. Where possible, many efforts were done in order to develop H.264/AVC standard-compliant MD coders, with the aim of providing Multiple-Description capabilities without affecting the H.264/AVC bitstream syntax. Finally, a comparison between the proposed MDC solutions is given.

MDC schemes can be grouped into two sets: those exploiting the spatial correlation within each frame of the sequence, and those taking advantage of the temporal correlation between the subsequences obtained by the temporal sampling of the original sequence. In sections II and III two schemes based on spatial correlation are presented, while in sections IV and V two algorithms exploiting temporal correlation are proposed. Results and conclusions are given in VI and VII.

II. SUB-SAMPLING MULTIPLE DESCRIPTION CODING (SMDC)

Due to the high spatial correlation within a frame, each pixel may be estimated according to its neighbors. Consequently, a MDC scheme can be defined by assigning adjacent pixels to separate descriptions.

In the Sub-Sampling Multiple Description Coding (SMDC) [5], each input frame is sampled along its rows and columns with a sampling factor of 2. Let $x(i, j)$ be the luma sample of the current frame at position (i, j), then the four sub-sequences are formed with pixels $x(2i\,,\,2j)$, $x(2i+1\,,\,2j)$, $x(2i\,,\,2j+1)$, and $x(2i+1\,,\,2j+1)$, respectively. In this way, four sub-sequences with halved resolution on both spatial directions, and a quarter of the original size, correspond to each input sequence. Each sub-sequence is then sent to a separate H.264/AVC encoder, and the output bitstreams are sent to four independent channels.

In the case that only one description arrives at the receiver, the end-user is able to reconstruct the coded sequence at a lower resolution without any artifact or channel distortion. When more descriptions arrive, the decoder can estimate the lost information exploiting the correlation among neighboring pixels. In our approach, assuming that pixel $x(2i\,,\,2j)$ belongs to the lost description, the missing pixel is replaced by the mean of the available pixels.

It is worth noticing that, also when no channel errors occur, the full resolution sequence is reconstructed with some small artifacts. This anomalous behavior is mainly due to

---

O. Campana, A. De Giusti, S. Milani and G. Calvagno are with the Dept. of Information Engineering at University of Padova (via Gradenigo 6/B, 35131 Padova - Italy). N. Zandonà is with STMicroelectronics (via C. Olivetti 2, 20041 Agrate (MI) - Italy). A. Cattani is with Telsey (viale dell'Industria 1, 31055 Quinto di Treviso (TV) - Italy). This work was supported by the Italian Ministry of Education, University and Research (MIUR), under the FIRB project "Reconfigurable platforms for wide-band wireless communications," (PRIMO) prot. RBNE018RFY.

small differences of the four encoders compression gain resulting in a non-homogeneous spatial quality, expecially observable at low bitrates. Correlating filters were adopted in order to address this problem and, hopefully, obtain a more pleasant reconstruction of the full resolution sequence. However, the performance of such filters was seriously affected by the data intrinsic correlation, and negligible improvements were obtained.

It is also worth to note that the SMDC scheme is a particular case of the frame-based MDC scheme [6, 7] where additional descriptions are added with a correlating transform. However, relevant drawbacks affect both solutions. In fact, since spatial sampling causes the reduction of the correlation between adjacent pixels, a decrease of the intra and inter coding gains is encountered.

III. MULTIPLE DESCRIPTION MOTION CODING (MDMC)

This Multiple Description Motion Coding (MDMC) scheme [8] splits the block-based motion vector field into two parts using a quincunx sampling. The resulting motion vector subfields are successively transmitted to the decoder over separate channels. Finally, the residual information of each macroblock is multiplexed into two bitstreams.

Some further constraints have been imposed on the design of the MDMC algorithm. First, an independent decoding of a single description has to be feasible so that, in case a whole description is lost, the receiver may decode the received data. Second, the computational overhead introduced by the MDC algorithm should be kept as low as possible in order to reduce the coder computational requirements and permit a real time execution.

Both these goals have been achieved by inserting a new layer, called the “Multiple Description Layer” (MDL), between the Network Abstraction Layer and the Transport Layer in the original H.264/AVC coder. The operations performed by the MDL concern the bipartitioning of the H.264/AVC output bitstream into two descriptions. Each description includes some unique specific information (i.e. not present in the other one). However, partial information has to be necessarily duplicated on both descriptions in order to permit the signal reconstruction. In this scheme, the whole motion estimation is performed once within the Single Description Coder (SDC), while the MDL only splits the coded information and rewrites some syntax elements.

In practice, descriptions are formed copying the initial SPS and PPS headers on both the streams as well as all the SH headers, while slice data are divided into two parts. The motion vector information of each macroblock is partitioned using an enhanced version of the quincunx sampling used by Kim and Lee [3]. This extension is necessary since the H.264/AVC coding standard permits a richer partitioning of the frames into blocks of different size, while the H.263 coding standard supports only 8x8 pixels blocks.

Unfortunately, the splitting process has some drawbacks when decoding the residual information related to the context-based entropy-coding algorithms used in H.264/AVC, namely CABAC and CAVLC. In particular, this happens because the residual information of each macroblock in H.264/AVC is partitioned into sixteen blocks of 4x4 pixels before being coded. The CAVLC algorithm uses the number of quantized coefficients different from zero in the blocks on the left and above the current one. Since coefficients of adjacent macroblocks are included in different descriptions, the merging routines of MDL may not be able to decode them correctly unless a transcoding is performed before transmitting the two bitstreams. In each macroblock, the upper and left 4x4 pixels blocks need to be transcoded, while the other nine blocks can be sent with no changes. The transcoding operation rewrites the `coeff_token` parameter which is the syntax element containing the information about the number of quantized block coefficients different from zero. The bit string used to code the `coeff_token` is selected between several VLC tables, and has to be changed for both upper and left blocks. In fact, during the splitting process, the coefficients of the blocks belonging to adjacent macroblocks are set to zero, and thus the MDL needs to change the actual coding context.

We ran many simulations in order to evaluate the redundancy introduced in the bitstreams and the quality of the reconstructed signal in case of channel errors. The redundancy was estimated measuring the percent increment between the overall bitrate of the two MD streams and the original one. In the proposed implementation, the allocated redundancy can not be controlled by reducing the perceived quality as other MDC schemes do, but it is an intrinsic property of the coded sequence. Simulations showed that the lower is the quality of the encoded sequence, the bigger is the introduced redundancy. For instance, the coding of the “*Foreman*” sequence yielded a redundancy increment between 7.15% and 67.43% for incrementing values of `QP` in the range 0-51.

There are two reasons that are likely to explain the faster growth of the redundancy for static sequences. A first interpretation of this behavior is that, for high-quality and high-motion sequences, most of the bitstream is represented by residual information. In fact, a small `QP` value increases the number of non-null coefficients. When the quality is low or the sequence is almost stationary, the number of coded coefficients decreases, and the percentage of the bitstream related to residual information becomes proportionally smaller. On the other hand, the percentage of coded bits associated with motion vectors and headers increases, yielding more redundancy between the two bitstreams. A second explanation concerns motion vectors. If the sequence is stationary, then all the vectors will be either null or close to zero. Therefore, such coefficients will be represented with a small number of bits per component, and the replaced null motion vector will be close to the real one that has to be coded.

IV. MULTIPLE STATE VIDEO CODER (MSVC)

Depending on the sequence frame rate and given the high temporal redundancy of the input sequence, each frame slightly differs from the previous ones. From this assumption, two (or possibly more) subsequences may be extracted from the original video sequence by temporal sampling. Each temporal subset of frames is then processed in order to create different video streams.

In the simplest implementation, given the input sequence, odd and even frames are multiplexed in two subsequences $x(2n)$ and $x(2n+1)$. Each subsequence is then independently processed by a H.264/AVC coder, and the corresponding output bitstreams are sent over independent channels. This type of MD coder is also known as "Multiple State Video Coder" (MSVC) [9] since it requires the storage of more than one frame, i.e. the state, in order to permit the correct decoding of the whole sequence. Also in this scheme, the allocated redundancy is not a tunable parameter, and therefore the side-decoder achievable PSNR is an intrinsic property of the sequence being coded.

At this point, a note on the interpolation process should be made. Since the state of the decoder cannot be recovered after a loss of information, the interpolation which estimates the missing frame increases the perceived distortion of the reconstructed sequence. This is mainly due to the mismatch between the reference state between encoder and decoder.

Note also that the two streams are perfectly compliant with the syntax defined in the standard. In case the receiver gets both the streams, it can reconstruct the whole sequence at full frame rate. Whenever one of the two streams is lost, a standard H.264/AVC decoder may reproduce anyway the coded sequence at half bitrate. Moreover, if the receiving device implements a MD decoder, the missing information can be estimated by temporal interpolation of the correctly received subsequence.

In the case of losses, the quality of the reconstructed sequences is seriously affected by the specific algorithm used for error concealment. A naive solution simply displays the coded sequence at half bitrate. Unfortunately, this technique, yields a less smooth reconstructed sequence with a decreased visual quality. However, smoothness may be improved estimating the lost state.

In our work, we experimented many types of recovering techniques. In the first state-recovering algorithm, the average between adjacent frames is used as an estimate of the lost state. The "average" state was then inserted in the H.264/AVC-decoder frame buffer permitting the decoding of all the successive frames of the corresponding subsequence.

Quality variation can be indeed smoothed taking into account the information on motion vectors leading to the *in-place motion compensation*. Since motion vectors computed in the encoding process are temporally correlated, then a lost frame could be estimated according to the information available from the succeeding one. Let $x(2n+1)$ be the lost frame (odd sequence) and $x(2n+2)$ be the next even frame which is decoded taking the frame $x(2n)$ as reference for motion compensation. A good estimation of $x(2n+1)$ can be made taking the previous frame as $x(2n)$ reference and using the MVs of $x(2n+2)$ halved for motion compensation.

Experimental results showed that the in-place recovering algorithm improves the quality of the reconstructed sequence achieving a 1 dB PSNR gain with respect to the interpolation by frames average. Obviously, the in-place technique has a greater computational complexity since motion compensation is required to reconstruct the lost state.

V. MOTION-COMPENSATED MULTIPLE DESCRIPTION (MCMD) VIDEO CODING

In the Motion-Compensated Multiple Description (MCMD) scheme [10], the input sequence is subsampled into even and odd frames sequences. The MD encoder is made-up of three dependent coders employing separated frame buffers: a central coder which receives both even and odd frames, and two symmetric side coders which work on even and odd frames, respectively. Coders outputs are merged into two equally-important descriptions, and sent over independent channels.

The central coder works on the full-rate sequence, and predicts the current frame $x(n)$ from the previous two, $x(n-1)$ and $x(n-2)$, by implementing a second-order linear predictor after the block-based motion compensation in the DPCM loop, while side encoders separately work on the even or odd frames at halved frame rate. The side encoders do not behave as regular H.264/AVC encoders. In fact, they do not output a regular prediction error, instead they encode the difference between the estimate of the central and the prediction at side encoders. Therefore, the side information is not a prediction error but a signal that the decoder has to add to its state to be able to recover the real prediction error in case of transmission errors.

It should be noted that, forcing the macroblock partition at side coders increases the additional redundancy. In order to mitigate this effect, we let each side coder to estimate independently its optimal motion vectors and macroblock partitions. Unfortunately, the reconstructed frame at the central coder is not yet available, and therefore, side coders cannot coherently perform motion compensation and provide their mismatch signal. However, under the reasonable assumption that the mismatch signal contains mainly high-frequency components, we may assume that the coarse quantization of side coders retains only low-frequency components, allowing us to perform side motion estimation before the central one.

This consideration leads us to an algorithm that can be summarized into the following three steps:

i) The motion vectors MV_2 and macroblock partitions for side coders are estimated.

ii) The MV_2 set is passed to the central coder, and central motion estimate is performed by computing the MV_1 set. Since this operation may imply a different optimal macroblock layout, the MV_2 vectors need to be adapted to fit it.

iii) The mismatch signal is computed as the difference between the reconstructed frames at central and side coders. No motion estimation is performed at this step.

Following this three-step algorithm, the estimate of the MV_2 motion vectors is demanded to side coders rather than to the central one. Thus, side bitrates significantly decrease at the expense of sub-optimal MV_2 motion vectors used in the central coder. Forcing side coders to compute the MV_2 motion vectors instead of the central coder yields a great decrement of side coders bitrates, at the cost of non-optimal motion vectors used in the central coder. Nevertheless, simulations showed that the resulting overall redundancy is acceptable for MDC applications.

VI. COMPARISON BETWEEN THE PROPOSED SCHEMES

In this paper, four MDC techniques were considered and implemented on the H.264/AVC video coding standard: Subsampling Multiple Description Coding (SMDC), Multiple Description Motion Coding (MDMC), Multiple State Video Coding (MSVC), and Motion Compensated Multi Description (MCMD) coding.

Evaluating a MDC scheme is a complex task, because many aspects should be considered. The Redundancy Rate-Distortion (RRD) region $\rho(D_1, D_0)$, which describes the distortion D_1 achievable by side decoders as a function of the allocated redundancy, and with fixed central encoder distortion D_0, is often used to evaluate the capabilities of MDC schemes. We propose a RRD region comparison of the proposed solutions. However, it should be noted that it does not fit particularly well for the coding schemes where redundancy is not a degree of freedom.

Observe that, in SMDC, MDMC and MSVD the user can not set the amount of introduced redundancy, since it is an inner property of the encoded sequence and not a degree of freedom. In the MCMD coding scheme, instead, redundancy can be varied tuning the side encoders rates. For this reason, the RRD region is represented by a curve for the MCMD coding scheme, while it degenerates to a single point for the first three schemes. In Figure 1, the four RRD regions have been reported. These results correspond to the coding of the “*foreman*” and “*akiyo*” qcif sequences using `QP` = 29, and assuming that one whole frame is lost during the transmission of a description.

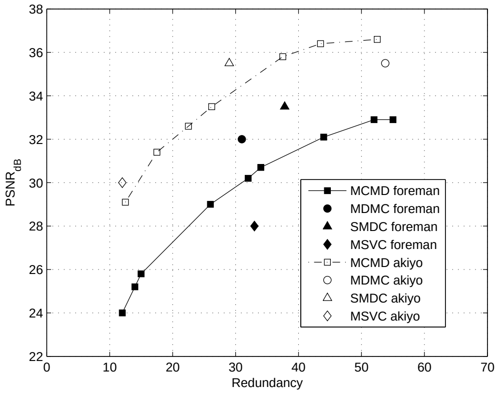

Figure 1. Redundancy Rate-Distortion (RRD) regions for the four MD coding schemes for the “*foreman*” and “*akiyo*” qcif sequences and `QP` = 29, in case of a whole frame loss.

Figure 1 shows the behaviour of the proposed schemes related to the motion complexity of sequences. In this example “*foreman*” is a fast-varying sequence and “*akiyo*” is a quasi-static one. The SMCD and MCMD techniques are substantially invariant to the characteristics of the scene being encoded. On the contrary, the MDMC and MSVC algorithms are strongly affected by the amount of motion: MDMC performs well with non-static sequences because the produced bitstream principally consists of the residual information that is not doubled in the splitting process, while the estimate of the lost state in SMCD works at its best on static scenes.

Since PSNR is not the only relevant aspect, but also computational cost and syntax compliance are important, we propose the qualitative comparison of the considered schemes, which is reported in Table 1.

Spatial MDC schemes usually demand lower computational complexity than temporal ones. In fact, in temporal MDC algorithms the storage of many reference frames is required, and in the particular case of the MCMD algorithm, there is also an extra computational load due to the double motion estimation required to compute two motion vectors fields. In the MCMD, as for the MSVC, the number of operations is roughly the same of a standard H.264 coder.

	SMDC	**MDMC**	**MSVC**	**MCMD**
Exploited correlation	spatial	spatial	temporal	temporal
Efficiency	good	good	quite good	variable
Computational Cost	medium	low	medium	high
Syntax Compliance	yes	yes	yes	no
Tunability	no	no	no	yes

TABLE 1. Qualitative comparison of the considered MDC coding techniques.

However, it requires a double-sized frame buffer.

We must also point out that MSVC, SMDC, MDMC provide fully H.264/AVC-compliant bitstreams which can be, thus, decoded by a standard H.264 decoder. MCMD streams do not correspond to the standard H.264/AVC syntax, and require an ad-hoc multiple-description decoder at the receiver. In addition, we must also observe that the MDMC layer is external to the H.264 core unit, and can be independently optimized since it only depends on the bitstream syntax. In this perspective, we may define a set of working profiles for the MDMC coder using the different coding performance provided by the H.264 core unit, and choose them according to the desired computational complexity.

VII. CONCLUSIONS

In this paper, four MDC techniques were considered and implemented on the H.264/AVC video coding standard: Subsampling Multiple Description Coding (SMDC), Multiple Description Motion Coding (MDMC), Multiple State Video Coding (MSVC), and Motion-Compensated Multi Description (MCMD) coding.

These algorithms showed different capabilities that could be obtained exploiting either the temporal or the spatial redundancy intrinsic to a video sequence. A direct comparison of the MDC schemes performance is not easily definable because of the differences of these algorithms in terms of parameters tunability, working domain, computational complexity, and syntax compliance. A qualitative but comprehensive overview of all these factors showed that there is not an universal best solution. Even if MDC schemes based on temporal correlation provide better results than spatially-based techniques, choosing the optimal multiple description coding scheme is a trade-off between the above factors, and the specific environment where multiple descriptions coding is going to be applied.

REFERENCES

[1] N. Jayant, “Sub-sampling of a DPCM speech channel to provide two “self-contained” half-rate channels,” *Bell Syst. Tech J.*, vol. 60, no. 4, pp. 501–509, Apr. 1981.

[2] Y. Wang and S. Lin, “Error-resilient video coding using multiple description motion compensation,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 6, pp. 438–452, June 2002.

[3] C. Kim and S. Lee, “Multiple description coding of motion fields for robust video transmission,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 9, pp. 999–1010, Sept. 2001.

[4] Y. Wang, A. Reibman, and S. Lin, “Multiple description coding for video delivery,” *Proc. IEEE*, vol. 93, no. 1, pp. 57–70, Jan. 2005.

[5] A. Cattani, “Appunti sulla codifica MD,” 2004, FIRB - University of Padova internal report, (in italian).

[6] R. Bernardini, M. Durigon, R. Rinaldo, L. Celetto, and A. Vitali, “Polyphase spatial subsampling multiple description coding of video streams with H264,” in *Proceedings of ICIP*, vol. 5, Singapore, Oct. 2004, pp. 3213–3216.

[7] R. Bernardini, R. Rinaldo, A. Tonello, and A. Vitali, “Frame based multiple description for multimedia transmission over wireless networks,” in *Proc. of 7th Int. Symp. WPMC*, vol. 2, Abano Terme, Italy, July 2004, pp. 529–532.

[8] O. Campana and S. Milani, “A Multiple Description Coding Scheme For The H.264/AVC Coder,” in *Proc. of the International Conference on Telecommunication and Computer Networks IADAT-tcn2004*, San Sebastian, Spain, Dec. 2004, pp. 191–195.

[9] A. Cattani, “Codifica video tramite descrizioni multiple,” Master’s thesis, University of Padova - Department of Information Engineering, June 2003, (in italian).

[10] N. Zandonà, S. Milani, and A. De Giusti, “Motion-Compensated Multiple Description Video Coding for the H.264/AVC Standard,” in *Proc. of IADAT International Conference on Multimedia, Image Processing and Computer Vision*, Madrid, Spain, Mar. 2005, pp. 290–294.