Improved signal reconstruction and return channel suppression in Distributed Video Coding systems

Xavi Artigas [1], Luis Torres [1]
[1] Universitat Politècnica de Catalunya (UPC), Department of Signal Theory & Communications
Campus Nord, D-5. Jordi Girona 1-3, 08034 Barcelona
E-mail: *{xavi, luis}@gps.tsc.upc.edu*

Abstract - *Distributed Video Coding (DVC) is a coding paradigm that gives the decoder the task to exploit the source statistics to achieve efficient compression. Current approaches to DVC rely on motion-compensated interpolation to generate at the decoder an estimation of the frame being decoded. This paper presents an iterative motion-compensated interpolation technique that takes advantage of all available information about the frame being estimated, not only the previous and posterior frames as is common practice. Simulation results show that the addition of this estimation technique to an existing DVC codec produces a 0.15 dB improvement in the PSNR of the rate-distortion plots. Furthermore, a method to avoid using the return channel existing in some DVC implementations is presented that only incurs in a penalty of 10 to 100 kbit/s.*

Keywords – *Distributed Video Coding, Wyner-Ziv Coding, Motion-Compensated Temporal Interpolation, Iterative Motion Compensation, Return Channel Suppression.*

1. INTRODUCTION

Video coding has traditionally allocated most of the computational burden of the coding/decoding process at the encoder, leading to a complexity balance where complex encoders interact with simpler decoders. The Slepian-Wolf [1] and Wyner-Ziv [2] theorems showed that it is feasible to shift part of this computational burden to the decoder, allowing the existence of simpler encoders. Distributed Video Coding aims at exploiting these theorems to build encoders that are more suitable for applications where low complexity encoders are a must because memory, computation, and energy are scarce.

Current approaches to Distributed Video Coding ([3], [4]) are mainly based on the general bloc diagram shown in Fig. 1. Some frames of the video sequence are intra-coded, while the others (called Wyner-Ziv frames, or WZ frames) are coded using the DVC scheme. At the receiver, intra frames are interpolated to generate an estimation of the WZ frames, which is then combined in the distributed decoder with the information sent by the transmitter for the WZ frames, producing the decoded frames. From the distributed decoder view point, the information sent by the transmitter is known as main signal, and the estimation generated at the receiver is known as side information. It is not clear yet what is the best way to generate the main signal, and different codecs use different mechanisms. This paper focuses on the side information, so the exact way in which the main signal is generated is not of importance here.

Research on DVC has been focused until now on the generation of the main signal, since motion-compensated temporal interpolation (MCTI) has been used in all current approaches as a frame interpolator at the receiver, yielding excellent results. However, the quality of the decoded picture depends on both the main signal and the side information, and, as it turns out, in the particular context of DVC, MCTI can still be improved.

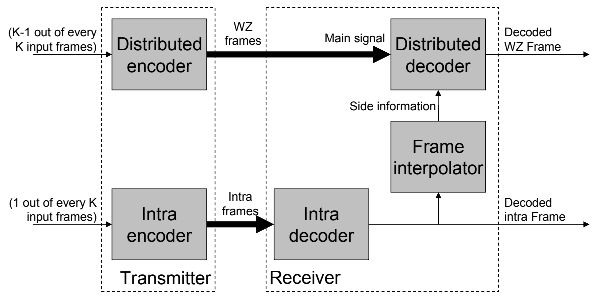

Fig. 1. Scheme for a non-iterative DVC codec.

Common MCTI techniques used so far in a DVC context assume there is no a priori knowledge about the frame being interpolated, so they rely solely on the past and future frames to generate the interpolation. However, once decoded, a WZ frame contains additional information that might not have been present in the reference frames used by the interpolator, and was carried by the main signal. Therefore, by using the already decoded WZ frame, the receiver can generate a second motion-compensated (MC) interpolation, using a priori knowledge about the frame, which should improve the quality of the interpolation.

This paper also presents a mechanism to suppress the need for a return channel that exists in some current DVC codecs, like the one in [4].

The paper is distributed as follows: Section 2 gives some background about MC techniques; Section 3 proposes the new frame interpolator; Section 4 examines a return channel suppression mechanism; Section 5 lists the obtained results, and finally, Section 6 extracts some conclusions and suggests future lines of research.

2. MOTION-COMPENSATED INTERPOLATION

The following subsections introduce motion-compensated temporal interpolation (MCTI) and MCTI with extra information, a special kind of MC interpolation that can be used in DVC contexts.

2.1. Motion-compensated temporal interpolation

The purpose of MCTI is to create an estimate (an interpolation) of a particular frame by using blocks from previous and subsequent frames.

MCTI is mainly being used in systems performing temporal up-sampling and, in contrast to the MC technique used in hybrid codecs, it has no knowledge about the frame being decoded.

MCTI can also be used in a DVC context to generate, at the receiver, an estimate of the frame being decoded that serves as side information. However, in this scenario, plain MCTI can be improved, as shown in the following subsection.

2.2. Motion-compensated temporal interpolation with extra information

Ramchandran *et al* [3] and Girod *et al* [5] presented different motion-based interpolation techniques in a Distributed Video Coding context, which use auxiliary information sent by the encoder to help the block matching process, and both reported improvements over the plain MCTI. In this paper, this auxiliary information is called *extra information*. Fig. 2 represents the general scheme for these systems.

In [3], a Cyclic Redundancy Check (CRC) is calculated and transmitted for every block. The decoder performs motion search and chooses the candidate block that produces the same CRC.

In [5], a small robust hash code is sent for each block. Instead of performing MCTI then, the receiver searches for the block in the previous frame that matches the received hash code.

In both systems, the transmitter sends two signals: the main signal, destined to the DVC decoder, and the extra information, destined to the frame interpolator. The extra information will help the frame interpolator generate the side information.

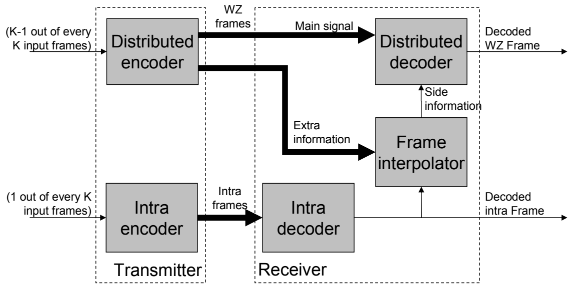

Fig. 2. General scheme for a DVC codec using motion-compensated interpolation with extra information.

3. ITERATIVE MOTION-COMPENSATED INTERPOLATION

Subsection 3.1 explains the concept of iterative MC interpolation, and afterwards, a specific implementation of this concept is introduced.

3.1. General scheme for iterative MC interpolation

This method is the main contribution of this paper. As the methods presented in subsection 2.2, it is based on the concept that plain MCTI estimates the frame being decoded without using any knowledge about it, other than previous and posterior frames. However, when used for DVC, extra knowledge about the frame being decoded might be available, if sent by the transmitter. As stated in the previous section, current methods [3] [5] transmit two signals: the main signal and the extra information. The proposed method only sends the main signal, which is then used equally by the DVC decoder and the frame interpolator.

The concept behind the proposed method is to generate the side information by means of a second estimation based on the first one, but taking into account any information provided by the main signal. Information is extracted from the main signal as follows:

Firstly, standard MCTI is used to estimate the frame being decoded (frame n), and the distributed decoder is run to generate the decoded WZ frame. In this paper, this first outcome of the distributed decoder is called partially decoded picture.

At this point, a second motion-compensated interpolation, called Motion-Compensated Restoration (MCR), is executed. MCR, depicted in Fig. 3, works pretty much like MCTI, but uses the partially decoded picture as well as the previous and posterior frames. Therefore, it is in a much better position than MCTI to generate a good estimate, since it has some information about the frame it is trying to estimate. The inner functioning of MCR will be explained in the next subsection.

Once this second estimate is obtained, the distributed decoder is run again. Since the estimate is better, the results for this second run are expected

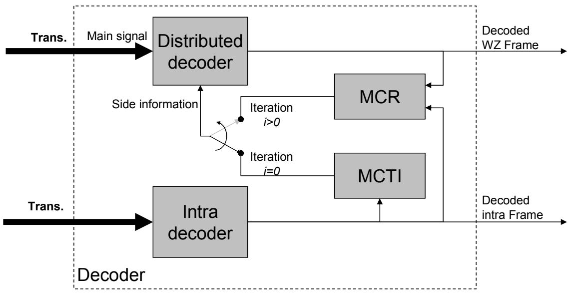

Fig. 3. General scheme for the DVC decoder, using iterative side-information generation.

to be better than the ones obtained in the first run. The partially decoded frame obtained this way can then be fed again into the MCR, and the whole process iterated.

3.2. Motion-Compensated Restoration

This process generates an estimate for the frame being decoded using information from neighboring frames, just like MCTI, but also from the partially decoded picture. Therefore, it has more information available than MCTI, and should produce better estimates. This is how it works:

For each aligned block in the partially decoded picture, the most similar block is searched for, in a number of sources. These sources include the past frame, the future frame, the motion-compensated average of the past and the future frame, and the result from the MCTI previously performed.

Every aligned block in the partially decoded picture is then substituted by its found best match.

This is different with respect to previous methods (subsection 2.2) in the sense that the partially decoded frame is used to hint the MC where the blocks from the reference frames must be placed, instead of using extra information sent by the encoder for this purpose.

The objective of MCR is not to improve the quality of the partially decoded picture, but to generate a motion-compensated estimate that is more accurate than MCTI alone. An image is being built again using blocks from the past and the future frames, but MCR has a better knowledge of where these blocks go than MCTI has.

The previous MCTI is used as one of the sources to guarantee that, in the worse case, MCR will perform exactly like MCTI.

Fig. 4 shows a part of frame 93 from the Foreman sequence, and clarifies how MCR works. Image a) is the result from MCTI, and, among other errors, the mouth is located too high. These artifacts are expected, since MCTI has only information from the reference frames, and can only interpolate the position of the mouth. In this particular frame, the interpolated mouth position is not correct. Image b) shows the partially decoded frame. By using the main signal, the distributed decoder has tried to

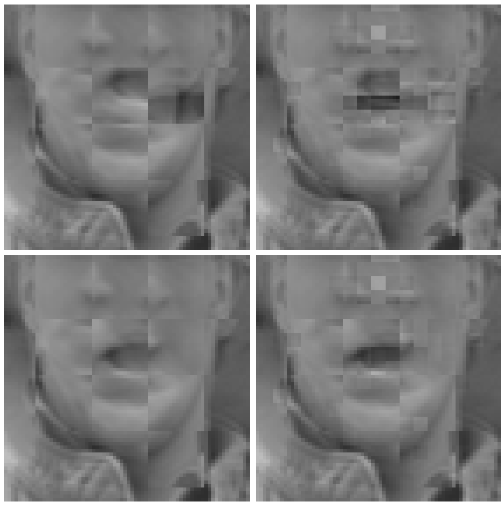

Fig. 4. a) Result from MCTI. b) Decoded image using MCTI as side-information. c) Result from MCR. d) Decoded image using MCR as side information.

correct the mouth position. The “false” mouth has been erased a bit, and a black slot appeared below, suggesting that maybe the real mouth should be located a bit lower. Image c) shows the result from MCR. MCR has taken every block in image b) and tried to find them among its many sources. The block most similar to the black slot has been one containing a mouth, but this time it is located a bit lower. Image d) shows the outcome of the second decoding run. Since the quality of the estimation was higher, the decoder has done a better job. In particular, observe the mouth in its proper place, and more nicely defined than it was in image b).

4. RETURN CHANNEL SUPPRESSION

The codec presented in [4] requires a return channel, since each decoding run requests information in the form of parity bits to the encoder. The proposed MCR scheme performs potentially many decoding runs per frame (at least two) in a way such that subsequent runs use all available parity bits (which have been transmitted in previous runs), and request additional bits if necessary.

In order to suppress the need for the return channel, the encoder needs to estimate the quality of the estimation the decoder will do, so the appropriate amount of information is sent through the main signal. The estimation performed by the encoder needs to be as computationally simple as possible, so motion compensation is not an option.

A very simple strategy is presented here: The encoder averages the previous and the next frame, and the difference with the original frame is calculated and the error probability computed. The required puncturing level for the turbo encoder is then determined from this probability using an

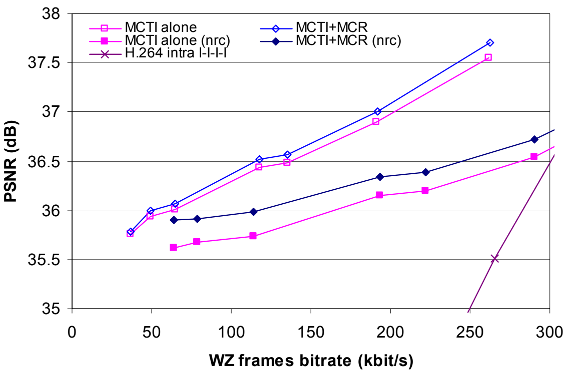

Fig. 5. Rate-distortion plot for an implementation of Girod's codec. 100 frames of the Foreman sequence 30Hz QCIF have been simulated. Bitrate of the WZ frames is shown (one out of every two frames). Plots marked "ncr" use the No Return Channel approach.

algorithm based on empirical results obtained by examining three different standard test sequences.

5. SIMULATION RESULTS

As a baseline for the experiments, a codec based on the transform-domain codec based on turbo codes presented by Girod *et al.* in [4] has been implemented. The transmission of different test sequences has been simulated using MCTI and MCTI+MCR. In this case only one iteration has proved useful, however, MC mechanisms other than MCR may profit from more iterations. The obtained results (Fig. 5) show that MCR improves the PSNR of Girod's codec [4] by 0.15 dB.

It can also be observed that, when not using MCR, the suppression of the return channel incurs in a bitrate penalty ranging from 10 kbit/s (for low bitrates) to 100 kbit/s (for high bitrates).

The plot in Fig. 6 shows that, for every frame, MCR always generates equal or better quality estimates than MCTI alone. Experiments with different sequences (Salesman and Mother & Daughter) have shown similar results, meaning that MCR can only improve the quality of the estimations, and consequently, the final result.

6. CONCLUSIONS

This paper has presented the concept of helping the motion-compensated interpolation process in a Distributed Coding Context using all the available information about the frame being decoded, without requiring any extra information to be sent, thus lowering the bit-rate.

One specific implementation of this concept, called Motion Compensated Restoration (MCR), has been presented. This MCR can be added to existing DVC codecs as a way to improve their side

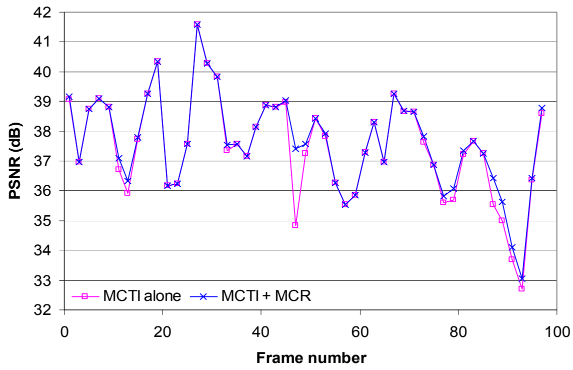

Fig. 6. PSNR of every WZ frame in the simulation of the Foreman sequence 30Hz QCIF, with and without MCR.

information generation, and therefore their final quality. Simulation results show that, when added to current DVC approaches, MCR increases the final quality by about 0.15dB for a variety of sequences.

Furthermore, a method has been presented to suppress the need for a return channel existing in some current DVC approaches. This method incurs in a bitrate penalty ranging from 10 to 100 kbit/s.

Although these progresses may be considered modest, the concepts presented in the paper may pave the way for greater improvements in the future. Current research is on the way to validate this assessment.

ACKNOWLEDGEMENT

The work presented was developed within VISNET, a European Network of Excellence (http://www.visnet-noe.org), funded under the European Commission IST FP6 programme.

REFERENCES

[1] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources", IEEE Trans. Inform. Theory, vol. 19 pp. 471-480, July 1973.
[2] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder", IEEE Trans. Inform. Theory, vol. 22, pp. 1-10, January 1976.
[3] R. Puri and K. Ramchandran. "PRISM: A new robust video coding architecture based on distributed compression principles". *Proc. of 40th Allerton Conf. on Comm., Control, and Computing*, Allerton, IL, Oct. 2002.
[4] B. Girod, A. Aaron, S. Rane and D. Rebollo-Monedero, "Distributed video coding", *Proc. of the IEEE*, Volume 93, Issue 1, January 2005.
[5] A. Aaron, S. Rane, B. Girod, "Wyner-Ziv video coding with hash-based motion compensation at the receiver", *Proc. IEEE Int. Conf. on Image Processing*, Singapore, October 24-27, 2004.