ON EXTRAPOLATING SIDE INFORMATION IN DISTRIBUTED VIDEO CODING

S. Borchert [a,b], R. P. Westerlaken [a], R. Klein Gunnewiek [b], R.L. Lagendijk [a,b]

[a] Information and CommunicationTheory Group,
Faculty of Electrical Engineering, Mathematics and Computer Science,
Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands
[b] Philips Research, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands

Keywords: Distributed Source Coding, Video Compression, Motion Estimation/Compensation, 3DRS

Abstract

The ongoing research in Distributed Video Coding (DVC) for low complexity encoding is trying to bridge the substantial performance gap to well known state-of-the-art coders. We introduce our true motion based extrapolation scheme and compare its performance to other state of the art systems in the field of DVC. The results of the extrapolation based approach display nearly the same performance as interpolation based ones. These results are also significantly better than the results from other state of the art extrapolation approaches. Furthermore we study the influence of the motion estimation part by investigating an interpolation approach, based on the same scheme. Finally, we study the losses incurred by using only past frames to represent the motion in the current one.

1. Introduction

To obtain high coding efficiency current video coding systems like H.264 employ an asymmetric complexity partitioning between encoder and decoder. Hence a device, that can easily run a video decoder in real-time, may not be suitable at all to run a video encoder. Complexity differences (in terms of the number of operations) between video encoding and decoding may differ one to two orders of magnitude. This suits video decompression on resource constrained/portable devices, but is unfavorable for video compression on these devices.
One way of dealing with video compression on resource constrained devices is Wyner-Ziv video coding [1],[2]. In this approach, referred to as distributed source coding, the complexities of encoder and decoder are reversed. Hence the encoder becomes fairly simple and leaves all the computationally expensive processing to the decoder. This is done by shifting the complex procedure of motion estimation/compensation from the encoder to the decoder. In contrast to conventional coders the motion estimation is thus only done at the decoder side. It is used to generate a motion compensated prediction Y, called side information, of the input frame X. The coding efficiency of a DVC scheme depends highly on the quality of Y.
Unlike the widely used motion compensated *interpolation* [3], [4], [5], [6], [7], [8], [9], the main contribution of this paper is the investigation of an *extrapolation* based scheme [10], [11], [12].
The discussion about motion compensation is given in Section 2. In Section 3 we present the actual method we are using. Its performance will be compared to other state of the art approaches in Section 4. Finally, Section 5 gives our conclusions.

2. Motion compensation for DVC

As opposed to conventional coding, in a practical DVC system it is not possible to use the motion estimation/compensation on the current frame itself. Thus we need at least one key frame which can be decoded without side information, f.i. a conventionally intra coded frame. As shown in Figure 1 it is only possible to use either key frames or already decoded Wyner-Ziv frames as input for the motion estimation. On the one hand it is possible to interpolate the missing frames between already available ones (f.i. *I-WZ-I-WZ-...-I*) and on the other hand it is possible to extrapolate from past frames

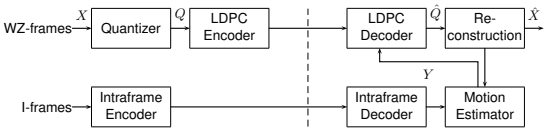

Fig. 1. Distributed Video Coding scheme.

(f.i. *I-WZ-WZ-...-WZ*).
Clearly both approaches have advantages, as well as disadvantages. Current literature mainly focuses on interpolation [3], [4], [5], [8], [9]. The main advantage of interpolation is future information about uncovering areas. However this only holds if the temporal distance is small enough, i.e. the GOP-size (*Group of Pictures*) is small. But keeping the temporal distance small has the drawback of either a high bit rate or a low quality for the only intra coded key frames. For a small GOP-size of two, the comparisons with extrapolation schemes in [6], [7] show noticeably better results for interpolation. But the results in [6], using a slightly larger GOP-size of 4 already suffer a significant quality reduction.
In contrast motion estimation and compensation in the extrapolation case can be done over a temporal distance of only one, regardless of the GOP-size. Nevertheless the main reason for using extrapolation thus far was only the sequential decoding for low latency [10], [11], [12]. Yet the main advantage of extrapolation is given by possible bit rate savings due to less I-frames. In case of successful decoding it is possible to use previously decoded WZ-frames. Since it is not possible to model uncovering areas from past frames only, these areas will increase the necessary bit rate of the WZ-frames. But even this effect can be softened by making use of iterative refinement as f.i. described in [12]. In section 3 we will present our extrapolation based method and compare its result to some other state of the art algorithms for both interpolation [4], [6], [7], [9] and extrapolation [6], [7] in Section 4.

3. Proposed extrapolation scheme

Before we can extrapolate $Y[n]$ from $X[n-1]$, we need to get a good estimate of the motion. The algorithm we consider in this paper is the 3DRS (*3-D Recursive Search*) motion estimation algorithm [13]. The 3DRS algorithm itself constructs a small set of candidate motion vectors, obtained from spatio-temporal predictions. Next to spatial and temporal candidates also update candidates are added to the candidate set. An update candidate is computed by taking a spatial candidate and adding a small random vector update to it. This method yields a smooth and consistent vector field. We also use CARS (*Content-Adaptive Recursive Search*) as an extension [14]. Thus the estimation process starts on large blocks and is only reduced at places in the image where the spatial accuracy must be high, i.e. near the border of moving objects.
Furthermore we use 3 frames for our motion estimation [15]. The motion estimation and compensation is then executed as follows:

(1) Generate three frames motion vector field for position n-2 (estimate motion between n-2 and n-3 as well as btw. n-2 and n-1).
(2) Generate two frames motion vector field for position n-1.
(3) Find areas that are more than once addressed. Check consistency of the vectors by shifting the field generated in (1) to n-1. Compare shifted vectors and available one (2). The vector with the lowest difference is chosen.
(4) Motion compensated extrapolation of the vector field, taking (3) into account. Fill unreferenced areas with temporally previous vectors (temporal hole-filling), i.e. copy the vector at the coinciding position from vector field (2).
(5) Extrapolate current, motion compensated frame $Y[n]$.

This extrapolation scheme has two fundamental problems. On the one hand there are the de-occluding areas and on the other hand the motion estimation itself.
The correct content in de-occluding areas (holes) is not present in the previous frame. But these areas are part of background and can in general be modelled reasonably well by copying neighboring background. This can be done either spatially by using surrounding pixel values or temporally by using motion vectors, as done in our approach.
With regard to motion estimation in conventional video coding this scheme has the following drawback. Since the current frame is not available, the actual motion can only be represented by the motion found for the previous frame (step 2). In addition it is also necessary to extrapolate the vector field itself from [n-1] to [n] (step 4). This is also a lossy step, which introduces additional errors. For these errors to be as low as possible the quality of the motion estimation is essential. But if the motion in the sequence itself is highly inconsistent a perfect estimate at [n-1] can still be far off the actual motion at [n].
The actual PSNR difference between interpolation and extrapolation as well as the conventional coding case, will be investigated in section 4.

sequence	proposed	[7]	[6]	[6], [7]	[9]	[4]	based on proposed		proposed	[6], [7]	proposed
-	**MX**	MX	MX	MI-2	MI-2	MI-4	**MI-2**	**MI-4**	**ME**	ME	**ME-MX**
carphone	**30.9**	29.45	29.03	31.72	-	-	**34.9**	**30.1**	**36.6**	33.80	**+5.7**
stefan	**26.3**	23.73	22.88	23.54	-	21.53	**27.6**	**23.2**	**27.6**	24.84	**+1.3**
foreman	**32.3**	31.57	30.66	32.56	32.2	28.85	**34.9**	**29.2**	**34.6**	33.21	**+2.3**
coastguard	**34.1**	31.32	30.82	32.17	34.2	30.59	**37.5**	**31.9**	**34.9**	32.61	**+0.8**
mother	**41.9**	-	-	-	38.1	-	**44.4**	**39.6**	**42.9**	-	**+1.0**
news	**33.5**	-	-	-	33.0	32.8	**36.4**	**32.7**	**37.3**	-	**+3.8**
hall	**37.4**	-	-	-	36.7	-	**40.0**	**36.2**	**37.7**	-	**+0.3**
silent	**34.0**	34.26	33.62	36.09	-	-	**36.4**	**31.7**	**35.8**	38.11	**+1.8**

TABLE I

AVERAGE PSNRS OF DIFFERENT SIDE INFORMATION SCHEMES

4. Experimental results

To investigate the performance of different interpolation and extrapolation schemes we compare the average PSNR results for several sequences in QCIF (176x144) resolution. This choice is necessary for the comparison with other schemes. However, the motion estimation in use was not designed for this low resolution and motion estimation in general works better on higher resolutions. Furthermore QCIF is an atypical resolution for almost any application. Since interpolation is less error prone with its averaging between past and future frames, this does in fact give extrapolation an extra penalty at this resolution.

We focus only on the PSNR of the side information itself. Therefore all results listed here are generated from original (non-quantized) key frames. Thus possibly varying settings for the key frames, as well as the influence of the overall framework have no influence on the results.

A. State of the art schemes

The schemes we compare are divided into 3 groups. Following the notation in [6] MX represents extrapolation, MI interpolation and ME motion estimation with the original frame available. The ME case is only a theoretical one to evaluate the performance of the motion estimation/compensation scheme compared to the motion prediction in conventional coding. As such it should always be better, since additional motion information is required. The motion vectors add additional information, since they are based on the original frame. However, no penalty has been set for this additional information.

The first approach we list in table I is divided into the quarter pixel extrapolation based results (MX) and the bidirectional multi reference ones (MI-2) in [7]. The results in [6] are similar to the ones in [7] for ME and MI, but different for MX. The last approach with a GOP size of 2 is the temporal side information generation in [9]. To show the impact of a larger GOP-size we also add the MI-4 results from [4].

The observations made in table I are:

MX On average our extrapolation scheme yields the best extrapolation performance. The exception of the silent sequence will be analyzed in more detail in the second part of the experiments (ME).

MI-2 On average our extrapolation scheme comes very close to the performance of the MI-2 schemes from [6], [7], [9]. The main difference between [7] and [9] is that the latter one uses a spatially smooth vector field as described in [8], while the former relies on a normal minimum residue approach. The 3DRS algorithm we use also yields a smooth vector field.

MI-2 The decreased quality of the extrapolated carphone sequence can be explained by the high amount of jerky global motion. With only past frames available and no camera stabilization it is very hard to adapt to this.

MI-2 The proposed interpolation scheme, making use of the same motion estimation as the extrapolation one shows a significantly better PSNR. As the motion estimation part is similar, this shows the higher robustness of interpolation towards errors. This is enhanced by the use of QCIF resolution, which is too small for the employed motion estimation. This is also shown by the fact that in some cases MI-2 outperforms ME.

MI-4 If we look at a GOP size of 4 instead of 2 the quality of the interpolated side information decreases significantly. In all cases it is noticeably below the extrapolation results.

B. Availability of the original frame

If the original frame $X[n]$ would be available, as is the case in a conventional video encoder, the quality of the side information increases because the vectors represent the motion at the correct time. The results of this experiment are given here:

1) The higher the difference between our proposed MX and the corresponding ME, the lower the temporal consistency of the motion in the sequence. In this case the motion

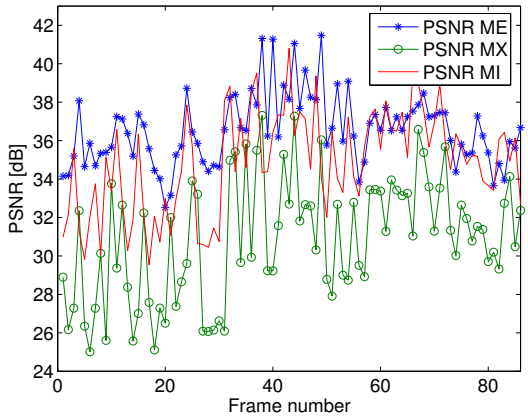

Fig. 2. Comparison of MX, MI-2 and ME for carphone.

from the previous frame is not a good representative for the current one. This is where interpolation approaches have the biggest advantage. An example for this is the carphone sequence. In figure 2, we can observe severe quality differences between MX and ME, with MI in between.

2) For extrapolation purposes a smooth vector field yields a better performance compared to a minimum residue one [7] with one exception. The silent sequence contains a lot of sudden and fast arm/hand movement. These are very hard to model with 3DRS, since they violate the assumption of objects moving gradually. For this case an approach without a smoothness-constraint is better.

5. Conclusions

The main contribution of this paper is to give an overview of the performance of many state of the art algorithms to generate side information. At the same time new methods for extrapolation and for interpolation are introduced. The main focus however is the extrapolation based approach. It is emphasized that extrapolation is indeed a valid alternative to interpolation and not only useful if low latency is required. While the proposed extrapolation algorithm is very close to many of the state of the art interpolation approaches, these differences seem to be based mainly on the motion estimation performance. Using the motion estimation part of the proposed scheme for interpolation shows that the stand alone quality of the side information is higher than the one for extrapolation. This is due to the future information being available, which reduces the errors by for instance averaging between the adjacent frames. But despite future information being available there can still be problems finding the correct motion vectors and correctly combining the information from past and future frames. While this is not crucial for QCIF resolution, where motion tends to be relatively small between neighboring frames, it becomes increasingly difficult with higher GOP sizes. We also point out that interpolation based schemes only perform optimally for small temporal distances. We consider such a system not practical due to the large amount of expensive I-frames. If the amount of I-frames is reduced, f.i. the GOP size increased to 4, the PSNR decreases significantly and falls below the one from extrapolation.

While this paper solely evaluates the quality of the side information generation schemes, future work will be focused on how the PSNR numbers of the side information reflect upon the bit rate of the complete compression scheme. This will allow us to objectively compare the quality gain of the side information with the cost in additional key frames.

References

[1] S.S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (discus): design and construction," in *Proceedings IEEE Data Compression Conference*, pp. 158-167, March 1999.

[2] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proceedings IEEE Data Compression Conference*, pp. 252-261, March 2002.

[3] A. Aaron, D. Varodayan, and B. Girod, "Wyner-Ziv Residual Coding of Video", *Proc. International Picture Coding Symposium*, Beijing, P. R. China, April 2006.

[4] J. Ascenso, C. Brites, F. Pereira, "Content Adaptive Wyner-Ziv Video Coding Driven by Motion Activity", *IEEE International Conference on Image Processing*, Atlanta, USA, October 2006.

[5] Wei-Jung Chien, L.J. Karam, G.P. Abousleman, "Distributed video coding with 3D recursive search block matching", *IEEE International Symposium on Circuits and Systems* , Island of Kos, Greece, May 2006.

[6] M. Tagliasacchi, S. Tubaro, A. Sarti, "On the Modeling of Motion in Wyner-Ziv Video Coding", *IEEE International Conference on Image Processing*, Atlanta, USA, October 2006.

[7] Z. Li, L. Liu, and E. J. Delp, "Rate distortion analysis of motion side estimation in Wyner-Ziv video coding," *IEEE Trans. on Image Processing*, Vol. 16, No. 1, pp. 98 - 113, January 2007.

[8] J. Ascenso, C. Brites, F. Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding" , *5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*, Smolenice, Slovak Republic, July 2005

[9] M. Tagliasacchi, A. Trapanese, S. Tubaro, J. Ascenso, C. Brites, F. Pereira, "Exploiting Spatial Redundancy In Pixel Domain Wyner-Ziv Video Coding", *IEEE International Conference on Image Processing*, Atlanta, USA, October 2006.

[10] A. Aaron and B. Girod, "Wyner-Ziv video coding with low-encoder complexity," *Proc. Picture Coding Symposium*, San Francisco, CA, Dec. 2004.

[11] L. Natrio, C. Brites, J. Ascenso, F. Pereira, "Extrapolating Side Information for Low-Delay Pixel-Domain Distributed Video Coding", *Int. Workshop on Very Low Bitrate Video Coding*, Sardinia, Italy, September 2005.

[12] Adikari, A.B.B.; Fernando, W.A.C.; Arachchi, H.K.; Weerakkody, W.A.R.J., "Sequential motion estimation using luminance and chrominance information for distributed video coding of Wyner-Ziv frames", *Electronics Letters Volume 42*, Issue 7, pp. 398 - 399, March 2006.

[13] Haan, G. de et al, "True-motion estimation using 3-d recursive-search block matching", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 3, Issue 5, pp. 368- 397, October 1993.

[14] Braspenning, R.; Haan, G. de: "Efficient Motion Estimation with Content-Adaptive Resolution.", *Proceedings of ISCE*, pp. 29-34, September 2002.

[15] S. Borchert, R.P. Westerlaken, R. Klein Gunnewiek, R.L. Lagendijk, "Improving Motion Compensated Extrapolation for Distributed Video Coding", *Proceedings of the thirteenth annual conference of the Advanced School for Computing and Imaging*, Heijen, the Netherlands, June 2007.