28th Picture Coding Symposium, PCS2010, December 8-10, 2010, Nagoya, Japan

DECODER-SIDE HIERARCHICAL MOTION ESTIMATION FOR DENSE VECTOR FIELDS

Sven Klomp, Marco Munderloh, Jörn Ostermann

Institut für Informationsverarbeitung
Leibniz Universität Hannover, Appelstr. 9A, 30167 Hannover, Germany
{klomp, munderl, ostermann}@tnt.uni-hannover.de

ABSTRACT

Current video coding standards perform motion estimation at the encoder to predict frames prior to coding them. Since the decoder does not possess the source frames, the estimated motion vectors have to be transmitted as additional side information.

Recent research revealed that the data rate can be reduced by performing an additional motion estimation at the decoder. As only already decoded data is used, no additional data has to be transmitted.

This paper addresses an improved hierarchical motion estimation algorithm to be used in a decoder-side motion estimation system. A special motion vector latching is used to be more robust for very small block sizes and to better adapt to object borders. With this technique, a dense motion vector field is estimated which reduces the rate by 6.9% in average compared to H.264 / AVC at the same quality.

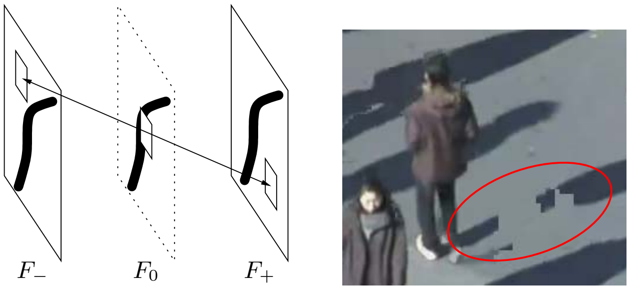
Index Terms— Video coding, motion compensation, dense vector field, block matching

1. INTRODUCTION

In current video coding standards like H.264 / AVC, the encoder performs motion estimation to exploit temporal dependencies in the video sequence. These motion vectors are transmitted to the decoder as side information and used for predicting the current frame to be coded. Thus, the decoder can be kept simple. However, increasing computational power allows to implement more sophisticated algorithms at the decoder. Recent studies have shown that motion estimation algorithms at the decoder can significantly improve the compression efficiency ([1], [2]).

Decoder-side motion vector derivation (DMVD, [1]) estimates the motion at the decoder by matching a template of already decoded pixels neighbouring to the current block in the reference frames. Thus, no motion vectors have to be transmitted and the data rate is reduced. Another approach to save the rate for the motion vectors is decoder-side motion estimation (DSME, [2]), in which a prediction of the current frame is computed by bidirectional interpolation of previously transmitted frames. Since the whole frame is predicted at once, no restrictions (e.g. to the block size) apply and any arbitrary motion estimation algorithm like block-based [3] or mesh-based [4] might be used to create a dense motion field. Therefore, problematic areas like object borders can be handled efficiently.

In [2], a rough motion vector field is estimated using a previous frame F_- and a future frame F_+. Thereafter, the accuracy is improved by performing bidirectional motion estimation with decreasing block sizes. As small blocks are sensitive to noise and can lead to a distorted motion vector field, the minimum block size was set to 4×4 to not impair the prediction quality. However, to handle motion at object borders accurately and fully utilise the advantages of the DSME architecture, a dense motion field is desirable. Furthermore, the bidirectional motion estimation can fail if the search range is too large as illustrated in Figure 1a. Homogeneous regions can match while the resulting motion vector does not represent the true motion. As a result, the actual content of the block to be predicted is lost. A practical example of this problem is shown in Figure 1b, in which parts of the shadow disappear.



(a) (b)

Fig. 1. Bidirectional motion estimation (a) can fail as shown for a detail of the predicted frame F_0 of the PeopleOnStreet sequence (b).

To approach this problem, an improved estimation method

978-1-4244-7133-1/10/$26.00 ©2010 IEEE

is proposed in Section 2.1 allowing to estimate a dense motion field to handle object borders but still retaining the true content of the frame. The experimental results of the proposed algorithm are presented in Section 3. This paper finishes with conclusions in Section 4.

2. DSME ARCHITECTURE

In DSME as proposed in [2], the current frame is predicted by performing motion estimation on adjacent frames at the decoder. This so called DSME frame is then fed into the reference picture buffer of a conventional coder to be used as an additional reference frame.

In contrast to the conventional motion estimation schemes, where the motion vectors are chosen by minimising the prediction error, DSME needs the true motion to compute an accurate prediction. Therefore, a hierarchical motion estimation is chosen for this kind of architecture: it allows very small block sizes on the one hand and prevents wrong local minima on the other hand. The following section describes the motion estimation algorithm in detail.

2.1. Bidirectional true motion estimation

The motion is hierarchically estimated using decreasing block sizes in each level to lock to the global motion but also get an accurate motion field at object borders. The hierarchical motion estimation starts with a block size of 64 pixels and is halved in six iterations until the final dense motion field is reached. For smaller blocks the matching window of the motion estimation is slightly larger than the block size during motion compensation to be more robust against image noise.

At each hierarchy level, the motion vectors between the previous and the next frame (F_-, F_+) are estimated using a conventional block matching algorithm by minimising the mean squared differences (MSD). The search area of the current hierarchy level H_n is thereby dependent on the motion vectors of the previous hierarchy level H_{n-1} as depicted in Figure 2.

The nine neighbouring motion vectors of the previous level H_{n-1} are applied to the current block and define a set of starting points for the motion refinement. The search range around each starting point is decreased with each level. Thus, the current block is able to follow the motion of every neighbouring block.

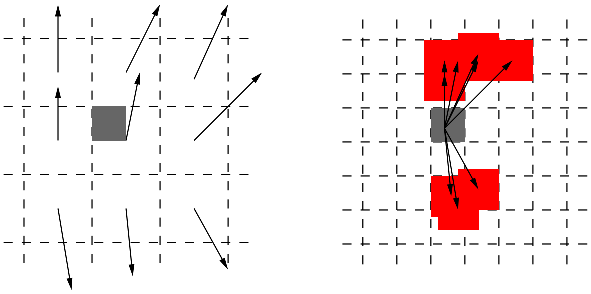

(a) Previous level H_{n-1} (b) Current level H_n

Fig. 2. Search area (red) derived from previous hierarchy level

To reduce noise in the motion vector field caused by small block sizes, the motion search is switched to a candidate based approach at a block size of smaller than 8 × 8. The motion vector for the current block is hereby set to one of the surrounding motion vector candidates without further refinement. This forces small blocks to decide to which motion object they belong and is achieved by setting the search range for those hierarchical levels to zero: the blocks 'latch' to motion vector candidates of the previous level. The resulting motion vector field now follows object borders very accurately as shown in Figure 3.

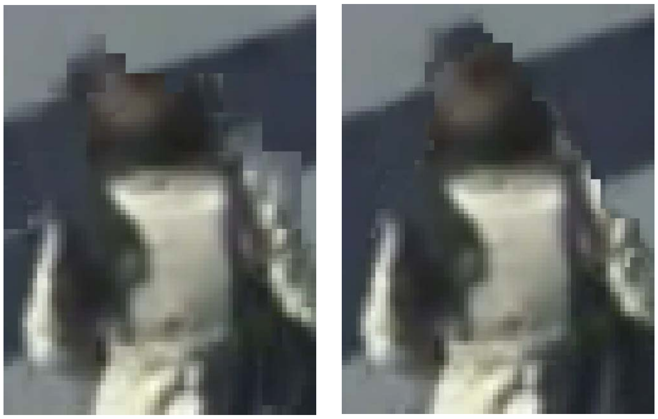

(a) old ME (b) new ME with latching

Fig. 3. Detail of the predicted frame for the PeopleOnStreet sequence

If the last iteration of forward motion search is finished, the motion vectors are aligned to the block grid of the current frame F_0. Thereafter, the vector field is smoothed using a vector median filter weighted by the mean absolute differences of the displaced blocks [5] to eliminate outliers.

2.2. Motion Compensated Prediction

After the motion vector field is estimated, the intermediate frame is predicted by assuming linear motion between the previous (F_-) and the next frame (F_+). This is done by halving the motion vectors and averaging the displaced pixels from F_- and F_+.

In Table 1 the average quality of the predicted frames are given for the motion estimation algorithm used in [2] and the new proposed algorithm. The quality of the predicted frame increases by about 0.6 dB for the four test sequences. Since this prediction is used as an additional reference for the video coder, it directly affects the rate-distortion performance as it is presented in Section 3.

Sequence	old ME	new ME
BasketballDrive	25.81 dB	26.57 dB
BQTerrace	31.66 dB	32.16 dB
Kimono	30.93 dB	31.66 dB
PeopleOnStreet	24.64 dB	24.99 dB

Table 1. Interpolation quality in PSNR of [2] and the proposed motion estimation algorithm

3. EXPERIMENTAL RESULTS

The tools have been implemented into the H.264 / AVC reference software JM 16.2 [6]. The performance of the proposed algorithms are evaluated with several test sequences also used by the JCT-VC group [7]. The sequences are coded with the GOP structure set to I-b-B-b-P using hierarchical B frames. In Figure 4a-d the operational rate-distortion curves of the B frames for the reference implementation (JM 16.2), the algorithm from [2] (DSME_old) and the proposed method (DSME_new) are plotted. To obtain an objective measure of the average PSNR and rate gain, the Bjøntegaard Delta [8] is calculated for each sequence (Table 2). Furthermore, the resulting BD-Rate gain for all frames (I, P, B) is also calculated and referred as overall BD-Rate gain.

Sequence	DSME_old	DSME_new
BasketballDrive	0.24 dB -7.12 %	0.41 dB -12.33 %
BQTerrace	0.11 dB -5.72 %	0.14 dB -7.04 %
Kimono	0.47 dB -13.83 %	0.62 dB -18.64 %
PeopleOnStreet	1.22 dB -23.02 %	1.42 dB -26.61 %

Table 2. BD-PSNR (dB) and BD-Rate (%) of the B frames for several high-definition sequences

The BasketballDrive sequence (Figure 4a) contains fast motion and significant motion blur. Nevertheless, the rate is decreased by 7.12% and 12.33% compared to JM for the old and proposed methods, respectively. Due to the improved robustness and accuracy, the overall BD-Rate gain (I, P, B) between the old and new motion estimation is improved by 2.18%.

Figure 4b shows that the gain for the BQTerrace sequence using the newly proposed algorithm is almost negligible compared to the previous DSME approach. A large part of this sequence contains only global motion due to a camera pan, which can be handled very well by the conventional H.264 / AVC algorithms using the direct mode. Furthermore, the non rigid motion of the water surface is hard to estimate and thus, the overall BD-Rate gain vanishes.

For the Kimono sequence (Figure 4c) with a smooth pan, the rate reduction increases from 13.83% BD-Rate to 18.64%. This results in an overall gain of 1.59% rate reduction while taking also the I and P frames into account.

Additionally, the proposed method works well also for sequences with very high resolution as shown for the PeopleOnStreet sequence in Figure 4d. The average rate reduction increases by 3.50% to 26.61% for the B frames. The resulting BD-Rate gain of all frames is 1.65% compared to the old motion estimation.

All rate-distortion curves have in common that the gain decreases towards higher rates. Two reasons account for this behaviour: First, the improved quality of the key frames F_- and F_+ has only marginal influence on the DSME frame prediction. Second, the reduced rate for the motion vectors has smaller influence at high rate points, since the rate for the residual increases significantly.

4. CONCLUSIONS

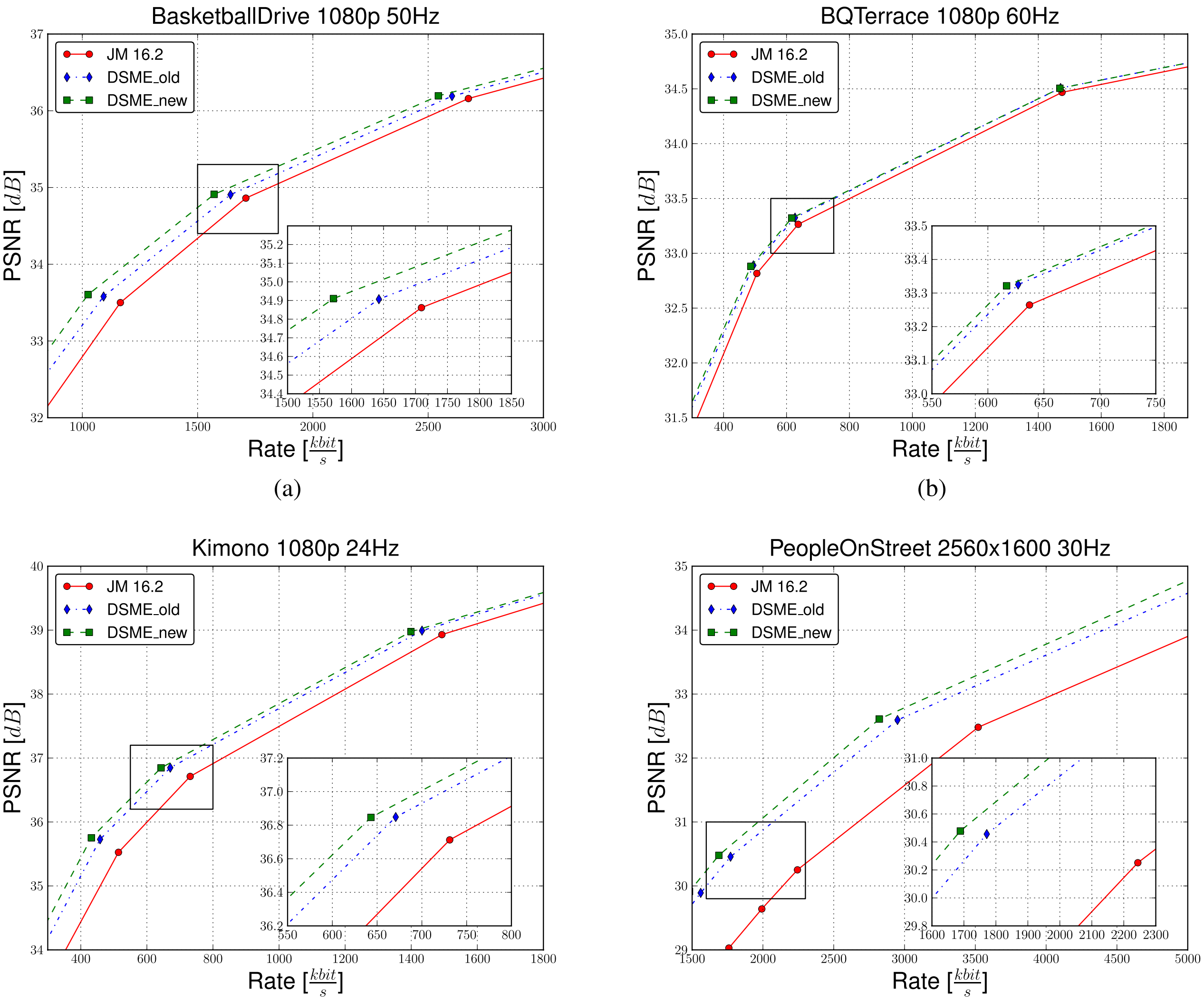
In this paper an improved motion estimation algorithm is proposed that fits the needs of decoder-side motion estimation. It allows to estimate a dense vector field for accurate motion compensation at object borders. The hierarchical approach efficiently reduces wrong motion vectors since local minima are efficiently avoided.

The quality of the predicted frame is enhanced by up to 0.76 dB. This reduces the overall bit rate of the coded sequence by 2.18% compared to [2] without increasing the complexity of the motion estimation algorithm. The introduced latching technique reduces the rate especially in case of complex local motion.

5. REFERENCES

[1] Steffen Kamp and Mathias Wien, "Decoder-side motion vector derivation for hybrid video inter coding," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, Singapore, July 2010, pp. 1277 – 1280, IEEE, Piscataway.

[2] Sven Klomp, Marco Munderloh, Yuri Vatis, and Jörn Ostermann, "Decoder-side block motion estimation for H.264 / MPEG-4 AVC based video coding," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, Taipei, Taiwan, May 2009, pp. 1641–1644.



(a)

(b)

(c)

(d)

Fig. 4. Rate-Distortion performance of B frames for several high-definition sequences

[3] João Ascenso, Catarina Brites, and Fernando Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding," in *5th EURASIP*, Slovak Republic, July 2005.

[4] Marco Munderloh, Sven Klomp, and Jörn Ostermann, "Mesh-based decoder-side motion estimation," in *Proceedings of the IEEE International Conference on Image Processing*, Hong Kong, September 2010, pp. 2049–2052.

[5] Luciano Alparone, Mauro Barni, Franco Bartolini, and Vito Cappellini, "Adaptive weighted vector-median filters for motion fields smoothing," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Georgia, USA, May 1996.

[6] "H.264 / MPEG-4 AVC joint model reference software," Available online at `http://iphome.hhi.de/suehring/tml/`, Version 16.2.

[7] ISO/IEC JTC1/SC29/WG11 MPEG, "Joint call for proposals on video compression technology," in *ISO/IEC JTC1/SC29/WG11 MPEG Output Document N11113*, Kyoto, January 2010.

[8] Gisle Bjøntegaard, "Calculation of average psnr differences between rd curves," in *ITU-T SG16/Q6 Output Document VCEG-M33*, Austin, Texas, April 2001.