A Parametric Merge Candidate for High Efficiency Video Coding

Michael Tok, Marko Esche, Alexander Glantz, Andreas Krutz, and Thomas Sikora
Technische Universität Berlin Communication Systems Group, Sekr. E-N 1, Einsteinufer 17
D-10587 Berlin, Germany
{tok,esche,glantz,krutz,sikora}@nue.tu-berlin.de

Abstract

Block based motion compensated prediction still is the main technique used for temporal redundancy reduction in modern hybrid video codecs. However, the resulting motion vector fields are highly redundant as well. So, motion vector prediction and difference coding are used to compress such vector fields. A drawback of common motion vector prediction techniques is their inability to predict complex motion such as rotation and zoom in an efficient way. We present a novel Merge candidate for improving already existing vector prediction techniques based on higher order motion models to overcome this issue. To transmit the needed models, an efficient compression scheme is utilized. The improvement results in bit rate savings of 1.7% in average and up to 4% respectively.

1 Introduction

As the increasing spatiotemporal resolution in video content causes higher and higher bandwidth needs for transmission, joint standardization activities between ISO/IEC MPEG and ITU-T VCEG have been started in April 2010 to work towards a new video coding standard for highly efficient video compression. The working title for that standard is HEVC (high efficiency video coding) [1]. The goal is to reduce the average bit rate needed to transmit video data by 50% in comparison to the latest video coding standard H.264/AVC [2]. Until now, the main improvements include larger quadtree based blocks (so called coding units) with a size of up to 64×64 pixel that replace the former macroblocks, larger transform sizes of up to 32×32 pixel, a better motion vector prediction, and compression scheme, better interpolation filters, and an additional sample adaptive offset filter. Until version 5.1 of the HEVC reference software (HM), an additional adaptive inloop filter based on Wiener filtering was used as well.

Even in HEVC, motion compensated inter prediction is the most powerful technique for reducing temporal redundancy as it is in all modern hybrid video codecs. This means that for each Inter block, called prediction unit (PU) in HEVC, a motion vector (MV) is generated by block motion estimation to determine which displacement is necessary to describe at which position a similar block can be found in already decoded frames of the transmitted sequence.

A motion vector field, resulting from such block wise motion estimation is highly redundant as the motion of adjacent blocks is very similar in most cases. Consequently, the MVs of a motion vector field can be predicted from MVs of surrounding prediction units that have already been coded. As there are various

1068-0314/13 $26.00 © 2013 IEEE
DOI 10.1109/DCC.2013.11

ways to derive such so called motion vector predictors, different methods already have been evaluated during the standardization process of HEVC [3]. The first test model (HM 1.0) used 5 different types of motion vector predictors. However, this amount was reduced to 3 in later versions to reduce coding complexity and bits needed for predictor signaling. The advantage of such vector prediction is, that each MV can be represented by a prediction error and a vector predictor index solely. These errors are much smaller in amplitude and thus can be encoded more efficiently. For blocks with no prediction error at all an additional mode called Merge is utilized as well. This mode simply copies the whole motion information such as reference index and motion vector from surrounding blocks.

So far, all MVP schemes used for video coding have one assumption in common. The motion of neighboring blocks has to be very similar. This assumption works well for smooth, translational motion but fails when so called higher order motion as zoom or rotation appears. To describe this kind of motion, parametric motion models can be used. They consist of a set of parameters, describing complex motion between adjacent frames. So, it is obvious that such models can be used to produce additional candidates for motion vector prediction and motion Merge.

During the standardization of H.264/AVC, Sun et al. already presented an MV coding (not prediction) scheme based on parametric motion models [4], but only used corner motion vectors to create bilineарly interpolated MVs for a whole frame. This technique suffers from two drawbacks. By just interpolating bilinearly, the MVs derived in the center of a frame are more imprecise. Also slight variations of the vectors to be encoded lead to not using the model for vector coding. To overcome this issue, Yuan et al. introduced a parametric predictor that is able to describe zoom motion [5]. In [6] the authors present a parametric motion vector predictor for HEVC. This predictor is able to describe all combinations of higher order motion such as zoom, rotation, shearing and perspective deformation. Additionally, by concatenating motion vector predictors for all reference frames of a reference list are generated.

However, the drawback of that parametric predictor is that it is not used for merged Skip and Inter predicted blocks. We propose to extend this prediction technique by including Skip and Inter Merge blocks in the parametric vector prediction process.

The remainder of this paper is organized as follows. Section 2 shortly describes the process of motion vector prediction and coding within the HEVC test model HM 5.1. Section 3 describes the parametric vector predictor as presented in [6] and introduces the novel Merge extension to it. A short overview of the process of obtaining high quality parametric motion models for the parametric motion vector prediction is given in Section 4. An efficient compression scheme needed for transmitting such models is is described in Section 5. Section 6 describes the evaluation process and presents the results in terms of coding gains. Finally, Section 7 summarizes the paper.

2 HEVC Motion Vector Coding

Still motion compensation is the most powerful technique for reducing temporal redundancy in hybrid video codecs. However, the vector fields, representing such motion are highly redundant as well. Thus, predicting such vectors from surrounding ones and, if necessary, just transmitting the motion vector prediction

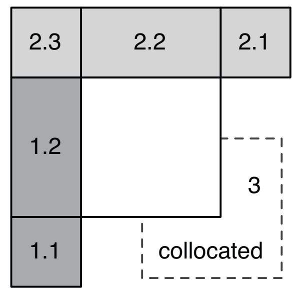


Fig. 1. Motion Vector prediction candidates as used in the HEVC test model 5.1 (The fourth candidate is $(0,0)^T$)

errors is a common technique. In the HEVC test model HM 5.1 motion vectors are encoded by utilizing the so called advanced motion vector prediction (AMVP) and transmitting just the vector prediction error or by signaling to copy the whole motion information from (temporarily) neighboring blocks.

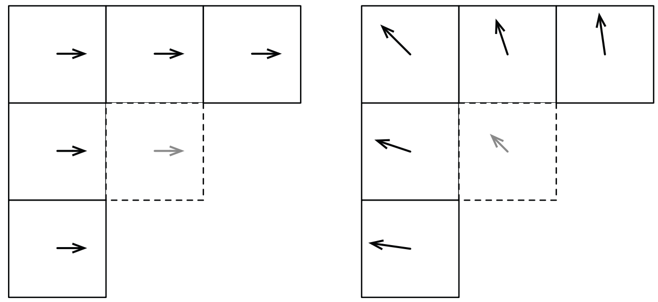
With AMVP, for each coding unit a set of up to four prediction candidates is generated. Figure 1 illustrates where these candidates are derived from. While the first two predictors are taken from spatially neighboring blocks, a third predictor, the so-called collocated one is taken from previously encoded frames. In addition, a zero MVP is added to the set of predictor candidates. Finally, an index signaling the best-fitting vector predictor, the vector prediction error and a reference index, signaling the selected reference frame are encoded and transmitted.

For homogeneously moving regions, the motion information, such as motion vector and reference index, of all coding units belonging to that region are identical. Then for each PU of each CU in that region a motion vector predictor index, a zero vector prediction error $(0,0)^T$ and the same reference index would have to be transmitted when utilizing AMVP. To avoid such unnecessary overhead the so called Merge mode is used. When this mode is selected, one index is transmitted to signalize what predictor and what reference index to use. Also, the transmission of a motion difference is omitted in that case. In contrast to the AMVP candidates, Merge candidates are generated PU-wise, guaranteeing better motion information for merging in some cases. It has to be mentioned, that the Skip mode uses Merge implicitly.

3 Parametric Motion for Motion Vector Prediction

AMVP and the Merge mode are powerful techniques for reducing redundancy in the motion information encoding process. Unfortunately their efficiency depends on the kind of motion to be encoded. As mentioned before, both techniques perform well for smooth translational motion of large regions.

When it comes to more complex motion such as rotation, zoom or even perspective deformations, however, the vector predictors considered by the HEVC coder are not precise enough. Figure 2 illustrates this problem. In the case of translational motion (Figure 2(a)) the candidates for Merge or AMVP for the gray dashed block are ideal. But when it comes to higher order motion such as zoom (Figure 2(b)), the quality of all spatial candidates is inadequate. To partially overcome this issue the collocated predictor is used to just derive a collocated



(a) Spatial Merge candidates for translational motion.

(b) Spatial Merge candidates for Zoom motion.

Fig. 2. Spatial Prediction / Merge candidates (for the gray dashed block) in the case of (a) translatoinal and (b) zoom motion.

predictor from an already encoded frame. The underlying assumption for that predictor is, that motion, no matter if it is complex or not, only changes slowly over time.

However, this assumption does not always hold. Moreover, for many CUs no fitting collocated predictor or Merge candidate exists. For such cases in [6] a parametric motion vector predictor (PMVP) has been presented. For PUs which are assigned to background regions, the motion can be described precisely by perspective eight parameter models. These models $\mathbf{H}$ describe the transformation of pixel or prediction unit positions $\mathbf{p} = (x, y)^T$ of one frame to corresponding positions in adjacent reference frames $\mathbf{p}' = (x', y')^T$ by

$$\begin{pmatrix} x' \cdot w' \\ y' \cdot w' \\ w' \end{pmatrix} = \mathbf{H} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{1}$$

where $\mathbf{H}$ contains the eight perspective transformation parameters

$$\mathbf{H} = \begin{pmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{pmatrix}. \tag{2}$$

For the center $\mathbf{p}$ of each coding unit in a frame, a parametric motion vector $\mathbf{v}_p = \mathbf{p}' - \mathbf{p}$ is calculated. Thus, by adding one such parametric motion model per frame to the video datastream, an additional parametric motion vector predictor $\mathbf{v}_p$ is available which led to bit rate savings of up to 2.42% for the HEVC test model HM 3.2 [6].

Nevertheless, a high amount of CUs is encoded by using the Merge mode. That is why PMVP is only used of very few prediction units. Further savings can be achieved by introducing an additional parametric Merge (PMERGE) candidate based on these models. The advantage of such a parametric Merge candidate is that it is generated for each PU and thus is more precise than a CU wide vector predictor. As the index used by the Merge mode signalizes vector predictor and reference index at once, only one Merge candidate for reference index 0 is generated and added to the list of available Merge candidates.

4 Parametric Motion Model Estimation

For estimating the needed parametric motion models (PMMs) that describe the complex deformations resulting from camera motion, the method presented in [7] is used: For each frame 400 features are selected and tracked by KLT-feature-tracking. Subsequently, a modified RANSAC is applied on these features for robustly estimating an eight parameter perspective motion model from background feature correspondences only. To reduce the amount of iterations needed by RANSAC for finding a reliable subset, in each iteration k only a four parameter model $\mathbf{H}_k$ is derived for correspondence classification

$$\mathbf{H}_k = \begin{pmatrix} \tilde{m}_{0,k} & \tilde{m}_{1,k} & \tilde{m}_{2,k} \\ -\tilde{m}_{1,k} & \tilde{m}_{0,k} & \tilde{m}_{3,k} \\ 0 & 0 & 1 \end{pmatrix}. \tag{3}$$

This model is used to identify whether a feature correspondence is an inlier or not. The largest set of n inliers $(x_{i,k}, y_{i,k})^T$ and their tracked correspondences $(\check{x}_{i,k}, \check{y}_{i,k})^T$ are taken to calculate a final perspective motion model by Least Squares via Pseudo Inverse

$$\mathbf{h} = \left(\mathbf{A}_k^T \mathbf{A}_k\right)^{-1} \check{\mathbf{x}}_k, \tag{4}$$

where $\mathbf{A}_k$ is the perspective design matrix

$$\mathbf{A} = \begin{pmatrix} x_{1,k} & y_{1,k} & 1 & 0 & 0 & 0 & -x_{1,k}\check{x}_{1,k} & -y_{1,k}\check{x}_{1,k} \\ 0 & 0 & 0 & x_{1,k} & y_{1,k} & 1 & -x_{1,k}\check{y}_{1,k} & -y_{1,k}\check{y}_{1,k} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n,k} & y_{n,k} & 1 & 0 & 0 & 0 & -x_{n,k}\check{x}_{n,k} & -y_{n,k}\check{x}_{n,k} \\ 0 & 0 & 0 & x_{n,k} & y_{n,k} & 1 & -x_{n,k}\check{y}_{n,k} & -y_{n,k}\check{y}_{n,k} \end{pmatrix} \tag{5}$$

containing the feature correspondences, $\check{\mathbf{x}}_k = (\check{x}_{1,k}, \check{y}_{1,k}, \ldots, \check{x}_{n,k}, \check{y}_{n,k})^T$ is a column vector consisting of the tracked correpsondences, and $\mathbf{h} = (m_0, \ldots, m_7)^T$ contains the final perspective motion parameters.

5 Parametric Model Compression

For each frame one parametric motion model is transmitted in addition. A single model consits of eight parameters, each represented by a 32 bit single precision floating point value. Thus, for deriving parametric motion vectors for PMVP and PMERGE at decoder side, for each frame additional 256 bit would have to be transmitted. For a common 25 Hz sequence, this would mean a bit rate increase of approximately 6.4 kbit/s. To reduce the overhead caused by model transmission, an efficient compression scheme has to be applied to the model parameters.

Since the parameters $m_0, \ldots, m_7$ of each model are highly correlated and have different ranges of value and as the two perspective parameters m_6 and m_7 are very sensitive to quantization, each model is transformed to a set of four frame-wise corner motion vectors $\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_4$ at the positions $(\pm x_{\text{res}}/2, \pm y_{\text{res}}/2)^T$. This is done by transforming these corner positions with the perspective model

$$\begin{pmatrix} \hat{x}_1 \cdot h_1 & \hat{x}_2 \cdot h_2 & \hat{x}_3 \cdot h_3 & \hat{x}_4 \cdot h_4 \\ \hat{y}_1 \cdot h_1 & \hat{y}_2 \cdot h_2 & \hat{y}_3 \cdot h_3 & \hat{y}_4 \cdot h_4 \\ h_1 & h_2 & h_3 & h_4 \end{pmatrix} = \begin{pmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & 1 \end{pmatrix} \cdot \begin{pmatrix} -\frac{x_{\text{res}}}{2} & \frac{x_{\text{res}}}{2} & -\frac{x_{\text{res}}}{2} & \frac{x_{\text{res}}}{2} \\ -\frac{y_{\text{res}}}{2} & -\frac{y_{\text{res}}}{2} & \frac{y_{\text{res}}}{2} & \frac{y_{\text{res}}}{2} \\ 1 & 1 & 1 & 1 \end{pmatrix} \tag{6}$$

HEVC test software	HM 5.1
Picture order / GOP settings	IBBB (hierarchical QP)
QP	$\in \{22, 27, 32, 37\}$
Asymetric Motion Partitioning	**off** for Low Complexity **on** for High Efficiency
Adaptive Loop Filter	**off** for Low Complexity **on** for High Efficiency
Sample Adaptive Offset Filter (SAO)	**on**
Largest CU size	64×64
Smallest CU size	8×8
Number of reference frames	4
Search Range	64×64
Entropy Coder	CABAC

TABLE 1
Coding conditions used in experimental evaluation

and calculating the difference between the transformed positions and the corner positions

$$\mathbf{V}_{c1} = \begin{pmatrix} \hat{x}_1 + \frac{x_{\text{res}}}{2} \\ \hat{y}_1 + \frac{y_{\text{res}}}{2} \end{pmatrix}; \mathbf{V}_{c2} = \begin{pmatrix} \hat{x}_2 - \frac{x_{\text{res}}}{2} \\ \hat{y}_2 + \frac{y_{\text{res}}}{2} \end{pmatrix}; \mathbf{V}_{c3} = \begin{pmatrix} \hat{x}_3 + \frac{x_{\text{res}}}{2} \\ \hat{y}_3 - \frac{y_{\text{res}}}{2} \end{pmatrix}; \mathbf{V}_{c4} = \begin{pmatrix} \hat{x}_4 - \frac{x_{\text{res}}}{2} \\ \hat{y}_4 - \frac{y_{\text{res}}}{2} \end{pmatrix}. \quad (7)$$

These vectors are more robust to quantization and can easily be transformed back to a perspective model at decoder side. Since the models to be transmitted are used for vector prediction and MERGE in the HEVC which is utilizing quarter pel motion vectors, the corner motion vectors are quantized with quarter pel accuracy as well. By doing so, it is guaranteed, that the maximum displacement error of each parametric model is also not larger than a quarter pel. Finally the resulting corner motion vectors are entropy coded by exponential Golomb coding. At decoder side a parametric model $\tilde{\mathbf{H}}$ is reconstructed by following (4).

6 Experimental Evaluation

For experimental evaluation both, the new parametric MERGE mode and the parametric motion vector predictor presented in [6] have been incorporated into the HEVC test model 5.1. Table 1 presents the encoder settings used for the evaluation. The details of these settings are explained in [8]. Table 2 gives an overview of the used test sequences' properties such as resolution, number of frames and frame rate. Additionally, losses in terms of BD-rates [9] caused by the parametric model overhead are presented for the low complexity and high efficiency coding settings of HM 5.1.

It comes clear, that for the Stefan sequence e.g. the considered techniques have to reduce the average bit rate by at least about 0.3% to overcome the losses induced by the model overhead. The average bit rate savings and losses for the low complexity and high efficiency settings with PMVP, PMERGE and a combination of both techniques are listed in Table 3 and 4.

For both settings an average gain of more than 1% for PMVP and a slight gain of about 0.20% and 0.30% respectively for PMERGE can be achieved. By combining both techniques, the resulting average gain increases to 1.43% and 1.64% respectively. Gains obtained by PMVP are analyzed in [6]. It is shown that

Sequence	Resolution	Frames	fps	Low Complexity BD [%]	Low Complexity BD-PSNR [dB]	High Efficiency BD [%]	High Efficiency BD-PSNR [dB]
Stefan	352 × 240	300	30	0.32	−0.02	0.31	−0.02
BQSquare	416 × 240	600	60	0.49	−0.02	0.44	−0.02
Waterfall	704 × 480	300	25	0.54	−0.02	0.52	0.00
Stanford	720 × 480	304	25	0.25	−0.01	0.24	−0.01
City	1280 × 720	250	25	0.29	−0.01	0.27	−0.01
BlueSky	1920 × 1080	217	25	0.12	0.00	0.12	0.00
BQTerrace	1920 × 1080	600	60	0.11	0.00	0.10	0.00
Station	1920 × 1080	313	25	0.45	−0.01	0.43	−0.01
Average				0.32	−0.01	0.30	−0.01

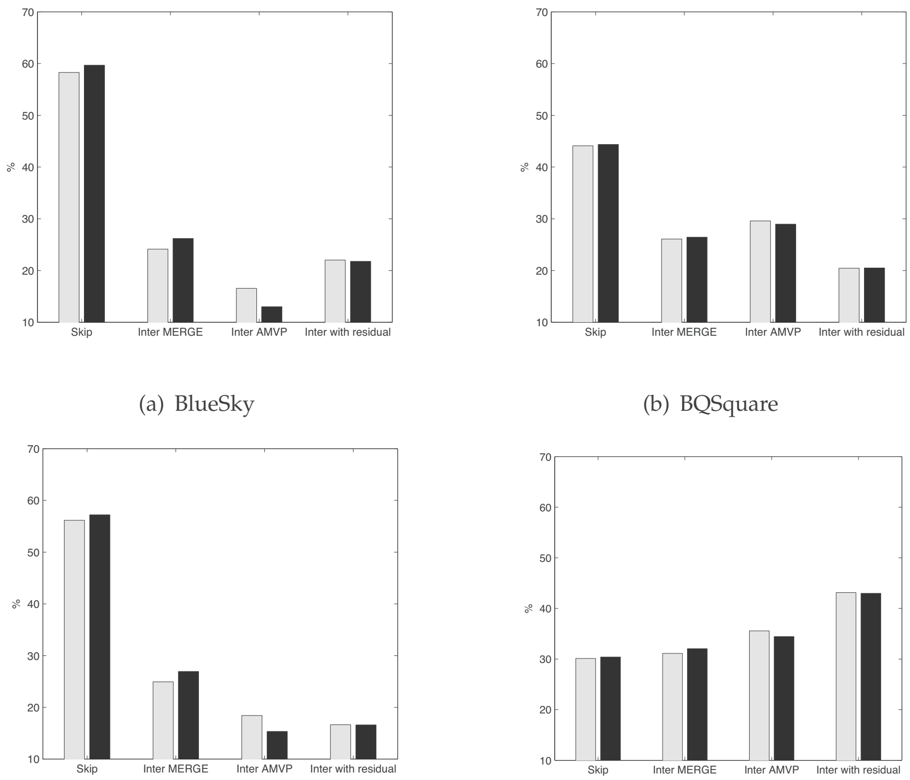
TABLE 2
Properties of used test sequences and their average coding loss through additional parametric motion model transmission

Sequence	PMVP only BD [%]	PMVP only BD-PSNR [dB]	PMERGE only BD [%]	PMERGE only BD-PSNR [dB]	PMERGE + PMVP BD [%]	PMERGE + PMVP BD-PSNR [dB]
Stefan	−1.69	0.09	−0.19	0.01	−1.79	0.09
BQSquare	0.82	−0.03	0.37	−0.01	0.58	−0.02
Waterfall	−2.33	0.07	0.33	−0.01	−2.52	0.07
Stanford	−0.28	0.01	0.11	0.00	−0.33	0.01
City	−3.56	0.10	−0.53	0.01	−3.99	0.11
BlueSky	−1.28	0.04	−0.40	0.01	−1.45	0.05
BQTerrace	0.05	0.00	0.03	0.00	0.03	0.00
Station	−0.70	0.02	−1.32	0.03	−1.96	0.05
Average	−1.12	0.04	−0.20	0.01	−1.43	0.05

TABLE 3
Encoding results for the low delay, low complexity case

Sequence	PMVP only BD [%]	PMVP only BD-PSNR [dB]	PMERGE only BD [%]	PMERGE only BD-PSNR [dB]	PMERGE + PMVP BD [%]	PMERGE + PMVP BD-PSNR [dB]
Stefan	−1.39	0.07	0.01	0.00	−1.62	0.08
BQSquare	0.47	−0.02	0.44	−0.02	0.51	−0.02
Waterfall	−2.58	0.08	−0.07	0.00	−2.79	0.08
Stanford	−0.48	0.01	0.09	0.00	−0.51	0.01
City	−3.71	0.11	−0.45	0.01	−3.99	0.12
BlueSky	−1.50	0.05	−0.71	0.02	−1.96	0.07
BQTerrace	−0.04	0.00	−0.02	0.00	−0.16	0.00
Station	−1.19	0.02	−1.65	0.04	−2.60	0.08
Average	−1.30	0.04	−0.30	0.01	−1.64	0.05

TABLE 4
Encoding results for the low delay, high efficiency case



(a) BlueSky

(b) BQSquare

(c) City

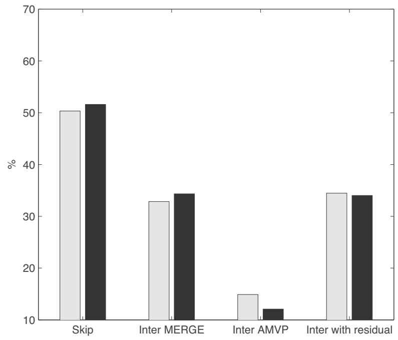
(d) Stefan

Fig. 3. Comparison of mode distributions in terms of frame pixels for QP 27 with high efficiency coding settings (light gray - reference, black - PMERGE)

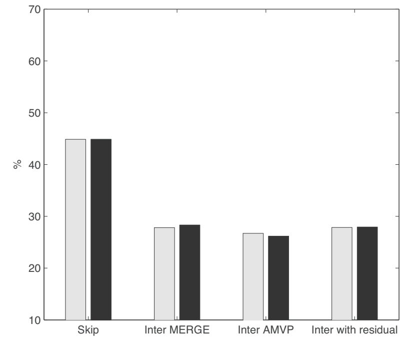
the precice parametric vector predictor leads to a better reference frame selection and thus to a better prediction signal. This results in a decreased number of transform coefficients to be transmitted. To understand the influence of PMERGE on the encoding process, Figure 3 and 4 give an overview of the mode distribution changes when using PMERGE for the high efficiency coding setting. It can be seen that for sequences with gain like BlueSky or City less blocks utilizing AMVP and MERGE are used. The increased number of Skip blocks also indicates that the motion information delivered by the parametric merge candidates has a higher precision than the original ones. A comparison of Inter blocks with residual information shows only small changes.

For BQSquare, the amount of transmitted residual information increases slightly. The losses for BQSquare can be explained by the very slow sub pixel motion in that sequence. Neither PMERGE nor PMVP are needed for that sequence. But a higher amount of predictors and merge candidates increases the costs for indexing which candidate to use. On the other hand, sequences such as Station, and City benefit from these new candidates. Gains of up to 3.99% for the low complexity and high efficiency coding setting demonstrate the potential of PMERGE in combination with PMVP.

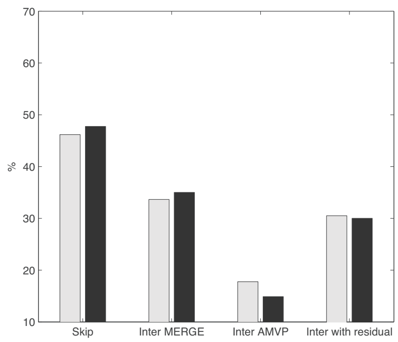
To get a rough estimate of the coding complexity increase through PMERGE and PMVP, the encoding and decoding times for the Stefan sequence have been



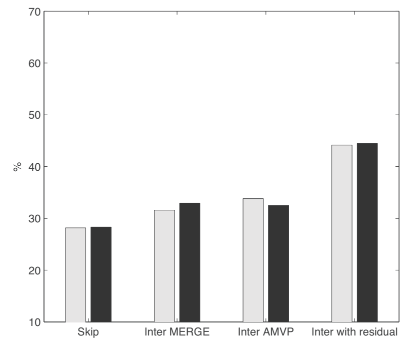
(a) BlueSky



(b) BQSquare



(c) City



(d) Stefan

Fig. 4. Comparison of mode distributions in terms of coding units for QP 27 with high efficiency coding settings (light gray - reference, black - PMERGE)

measured. The encoding time increase of 28% in average mainly arises from the parametric motion estimation and the additional vector prediction and merging candidates testing. The decoding time only increases by 9% in average mainly through the motion model decoding process and the vector candidate calculation.

7 Summary and Conclusion

A novel merge candidate for improving higher order model based motion vector prediction has been presented. The PMERGE, as well as PMVP have been incorporated into the HEVC test model 5.1 as additional candidates for motion merging and vector prediction.

For deriving PMERGE and PMVP candidates an additional parametric motion model is transmitted for each frame. The performance of both methods has been analyzed independently. Effects on the encoding process have been pointed out. PMERGE leads to bit rate reductions of up to 1.65%, PMVP utilization leads to gains of up to 3.71%. By combining both techniques, savings of up to 3.99% are achievble for selected sequences.

However, in cases of simple global motion such as translation, losses of up to 0.51% occur as well. Thus, a dynamic motion complexity based decision for switching PMERGE and PMVP on and off, could lead to even higher bit-rate savings and avoid losses at the same time.

References

[1] K. McCann, T. Wiegand, B. Bross, W.-J. Han, J.-R. Ohm, J. Ridge, S. Sekiguchi, and G. J. Sullivan, "HEVC Draft and Test Model editing," *ITU-T SG16 WP3, ISO/IEC JTC1/SC29/WG11 doc. JCTVC-D500-r1*, Mar 2011.

[2] "Draft ITU-T recommendation and final draft inernational standard of joint video specification (ITU-T Tec. H.264/ICO/IEC 14496-10 AVC)," *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050*, May 2003.

[3] B. Bross, J. Jung, Y.-W. Huang, Y. H. Tan, I.-K. Kim, T. Sugio, M. Zhou, T. K. Tan, E. F. K. Kazui, W.-J. Chien, S. Sekiguchi, S. Park, and W. Wan, "BoG report of CE9: MV Coding and Skip/Merge operations," *ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 document JCTVC-E481-r2*, Mar 2011.

[4] S. Sun and S. Lei, "Motion vector coding with global motion parameters," *ITU-T SG16/Q.6 VCEG document VCEG-N16*, Aug 2001.

[5] H. Yuan, Y. Chang, Z. Lu, and Y. Ma, "Model Based Motion Vector Predictor for Zoom Motion," *Signal Processing Letters, IEEE*, vol. 17, no. 9, pp. 787 –790, Sep 2010.

[6] M. Tok, A. Glantz, A. Krutz, and T. Sikora, "Parametric motion vector prediction for hybrid video coding," in *Picture Coding Symposium*, May 2012, pp. 381 –384.

[7] ——, "Monte-Carlo-based Parametric Motion Estimation using a Hybrid Model Approach," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, p. 1, 2012.

[8] K. McCann, B. Bross, W.-J. Han, S. Sekiguchi, and G. J. Sullivan, "High Efficiency Video Coding (HEVC) Test Model 5 (HM 5) Encoder Description," Jan 2012.

[9] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," *ITU-T SG16/Q.6 VCEG document VCEG-M33*, Mar 2001.