# Parametric Motion Vector Prediction for Hybrid Video Coding

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

*Abstract*— **Motion compensated prediction still is the main technique for redundancy reduction in modern hybrid video codecs. However, the resulting motion vector fields are highly redundant as well. Thus, motion vector prediction and difference coding are used for compressing. One drawback of all common motion vector prediction techniques is, that they are not able to predict complex motion as rotation and zoom efficiently. We present a novel parametric motion vector predictor (PMVP), based on higher-order motion models to overcome this issue. To transmit the needed motion models, an efficient compression scheme is utilized. This scheme is based on transformation, quantization and difference coding. By incorporating this predictor into the HEVC test model HM 3.2 gains of up to 2.42% are achieved.**

## I. Introduction

As the increasing resolution in video content causes higher and higher bandwidth for transmission, joint standardization activities between ISO/IEC MPEG and ITU have been started in April 2010 to work out a new video coding standard for highly efficient video compression. The working title of that standard is HEVC (high efficiency video coding) [1]. The goal is to reduce the average bit rate needed to transmit videos by about 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) that replace the former macroblocks (MB), larger transform sizes, a better motion vector prediction scheme, better interpolation filters, and an optional adaptive loop filter that is based on Wiener filtering.

Even in HEVC, motion compensated inter prediction is the main technique for temporal redundancy reduction as it is in all modern hybrid video codecs. This means that for each Inter block (called prediction unit in HEVC), a motion vector (MV) is generated by block motion estimation to describe at which position a similar block can be found in already decoded frames. A motion vector field, resulting from such blockwise motion estimation is highly redundant as the motion of adjacent blocks is very similar. This means, that these MVs can be predicted from MVs of already coded, surrounding MBs. As there are different ways to derive such motion vector predictors (MVP), different methods already have been evaluated during the standardization process of HEVC [3]. A first test model, called HM 1.0, used 5 different types of MVPs However, this amount was reduced to 3 in later versions for complexity reduction without any significant loss in compression efficiency. The great advantage of motion vector prediction is, that each MV can be represented by a prediction error solely. These errors are much smaller in amplitude and thus can be compressed more efficiently. However, a disadvantage is that the selection of a certain MVP has to be signalized somehow. HEVC for example encodes one MVP index per MV.

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, so called parametric motion models (PMM) can be used. They consist of a set of parameters, describing complex motion between adjacent frames. So, it is obvious that such PMMs can be used to produce additional MVPs for higher-order motion. During the standardization of H.264/AVC, Sun et al. already presented a MV coding scheme based on PMMs [4] , but only used corner motion vectors to create bilinearly interpolated MVs for a whole frame. This technique has two drawbacks. The PMM is more imprecise in the center of each frame due to the linear interpolation. Also slight variations of MVs to the PMM lead to not using it for MV coding. To overcome these issues, Yuan et al. introduced a parametric predictor [5] that is able to describe zoom.

We propose a novel PMVP, based on highly precise 8-parameter perspective motion models. This predictor is able to describe combinations of complex motion as rotation and zoom, induced e.g. by handheld cameras. With concatenating models from differing frames, PMVPs for all reference frames of a reference list are generated. A highly robust estimator is combined with KLT-feature-tracking, to receive precise PMMs. For transmitting the PMMs in an efficient way, a PMM compression scheme based on transformation, quantization and temporal difference coding is used.

The remainder of this paper is organized as follows. Section II shortly describes, how version 3.2 of the HM is deriving MVPs. Section III describes the two proposed changes in the MVP structure and explains how the new PMVP is incorporated into the HEVC. A short overview of how the PMMs, needed for PMVP are estimated at encoder side. An efficient PMM compression method that reduces the bits needed for these PMMs is described in Section V. Section VI describes the evaluation and presents the results in terms

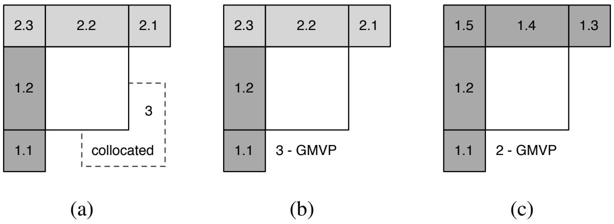


Fig. 1. Methods for deriving MVPs that are inserted in the MVP list of HEVC in HM 3.2. - (a) The HM 3.2 reference software derives three MVPs (left, top, collocated), (b) The proposed Method 1 replaces the collocated predictor by a parametric one., (c) The proposed Method 2 additionally merges the spatial MVPs to reduce bits for indexing.

of coding gains and Section VII summarizes this paper.

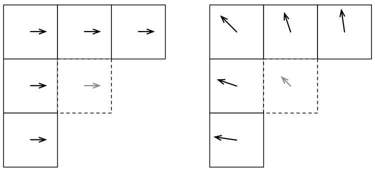
## II. Motion Vector Prediction in HEVC

Although motion compensation is the main technique for temporal redundancy reduction in hybrid video codecs, the motion vector field generated for the compensation is highly redundant itself. Thus, motion vector prediction and difference coding is applied on each motion vector. As explained in [3], the third draft of HEVC utilizes three motion vector predictors. In addition to two spatial predictors, derived from previously coded blocks on top and left side of each block to be coded, a so called collocated predictor is introduced. This predictor derives motion vectors from previously coded frames. Figure 1(a) illustrates how these three MVPs are obtained. To send the real motion vectors of all MBs to the decoder, only the prediction errors and an index, signaling which predictor is used, have to be transmitted.

## III. Parametric Motion Vector Prediction

The spatial predictors of HEVC exploit the smooth change of motion between neighboring blocks. This works well for translational motion, but is suboptimal for zooming, rotation and all kind of mixtures of complex camera motion. The problem of deriving a motion vector field for zoom is illustrated in Figure 2. To overcome this issue the collocated predictor is used in HEVC, which works well as long as motion does not change over time. Such changes occur e.g. when a seuquence is taken by a handheld camera or when zooming or rotation in a sequence changes over time. In these cases all common predictors, considered for HEVC so far are suboptimal. On the other hand, these types of motion, which are often observed in coding units assigned to background regions, can be described very precisely by perspective 8 parameter motion models. These models $\mathbf{H}$ describe the transformation of pixel or coding unit positions $\mathbf{p} = (x, y)^T$ of one frame to corresponding positions in adjacent frames $\mathbf{p}' = (x', y')^T$ by

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



(a) For smooth translation.

(b) For zoom.

Fig. 2. For complex global motion a problem with predicting the gray dashed MVs from neighboring MBs occurs.

where $\mathbf{H}$ contains the 8 perspective transformation parameters

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

That way, for the center of each prediction unit $\mathbf{p}$ in a frame, a parametric motion vector

$$\mathbf{V}_p = \mathbf{p}' - \mathbf{p} \tag{3}$$

is calculated. So, by adding such parametric motion models to a video datastream (one model per frame), an additional parametric MVP is available. By concatenating the parametric models of adjacent frames, MVPs for different reference indices and thus for different reference frames are derived. Thereby, the needed bits for transmitting MV prediction errors can be reduced. Nevertheless, the use of PMVP has to be signalized by an MVP index which increases the amount of bits. That is why, instead of adding PMVP as a fourth predictor, the collocated one is replaced. In the following this MVP derivation scheme is called Method 1. It is illustrated by Figure 1(b). In addition, for further index bit reduction, both spatial predictors are merged. This kind of MVP derivation is called Method 2 and illustrated by Figure 1(c).

## IV. Parametric Motion Estimation

To get a PMM that describes the complex transformations induced by camera motion, the parametric motion estimation method presented in [6] is used. So for each frame 400 features are selected and tracked by KLT-feature-tracking. Then, a robust estimator based on the Helmholtz principle is applied on the set of feature correspondences to reject outliers resulting from foreground motion and mistracking and derive a precise PMM. This estimator takes $m$ randomly selected subsets of two correspondences to generate one simplified four parameter motion model $\mathbf{H}_k$ per subset

$$\mathbf{H}_k = \begin{pmatrix} \tilde{m}_{0,k} & \tilde{m}_{1,k} & \tilde{m}_{2,k} \\ -\tilde{m}_{1,k} & \tilde{m}_{0,k} & \tilde{m}_{3,k} \\ 0 & 0 & 1 \end{pmatrix}. \tag{4}$$

This model is then used to define whether a feature correspondence of the whole set is an inlier or an outlier regarding to $\mathbf{H}_k$. With the number of inliers $N_k$ and the estimated error variance of these inliers $\sigma_k$, a rating per subset is defined by

$$\Phi_k = \frac{N_k}{\sigma_k} \tag{5}$$

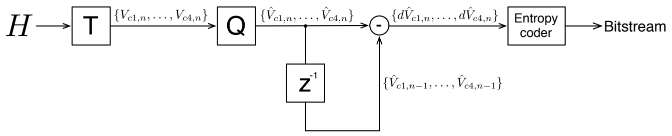

$H$
T
$\{V_{c1,n},\ldots,V_{c4,n}\}$
Q
$\{\hat{V}_{c1,n},\ldots,\hat{V}_{c4,n}\}$
-
$\{d\hat{V}_{c1,n},\ldots,d\hat{V}_{c4,n}\}$
Entropy coder
Bitstream
$z^{-1}$
$\{\hat{V}_{c1,n-1},\ldots,\hat{V}_{c4,n-1}\}$

Fig. 3. Method for lossy perspective motion model compression.

Only for the inlier features $\mathbf{x}_k$ and their tracked correspondences $\check{\mathbf{x}}_k$ with the largest $\Phi_k$, a final perspective PMM is calculated by Least Squares as

$$\mathbf{h} \;=\; \left(\mathbf{A}_k^T\mathbf{A}_k\right)^{-1}\mathbf{A}_k^T\check{\mathbf{x}}_k, \qquad (6)$$

where $\mathbf{A}_k$ is the perspective design matrix for the feature correspondences of the $k$th consensus set and $\mathbf{h} = (m_0, \ldots, m_7)^T$ contains the final motion parameters.

## V. Motion Model Compression

A single PMM consists of 8 parameters, each represented by a 32 bit single precision floating point value. So, for deriving GMVPs at decoder-side, for each frame additional 256 bit have to be transmitted. This would e.g. mean 6.4 kbit/s more for a 25 Hz sequence. Thus, the used PMMs have to be compressed in an efficient way.

Since the parameters $m_0, \ldots, m_7$ 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 PMM is transformed to a set of four frame-wise corner motion vectors at the positions $(\pm x_{\text{res}}/2, \pm y_{\text{res}}/2)^T$, just following (1) and (3). These vectors are more robust to quantization and can easily be transformed back to a perspective model at decoder side. Additionally each vector is highly correlated with its temporal predecessor so that differential coding in combination with exponential Golomb coding is used for redundancy reduction. The whole coding process for the PMMs is illustrated in Figure 3. For each frame $n$, a homography $\mathbf{H}$ is transformed to the four corner motion vectors $\mathbf{V}_{c1,n}$ to $\mathbf{V}_{c4,n}$ and then quantized to the set $\hat{\mathbf{V}}_{c1,n}$ to $\hat{\mathbf{V}}_{c4,n}$. The differrences $d\hat{\mathbf{V}}_{c1,n}$ to $d\hat{\mathbf{V}}_{c4,n}$ of these vectors to their temporal predecessor $\hat{\mathbf{V}}_{c1,n-1}$ to $\hat{\mathbf{V}}_{c4,n-1}$ is taken and entropy coded by exponential Golomb coding, before written into the bitstream. As quantization step size for the corner motion vectors $1/32$ was empirically found to be a good trade-off between bit rate and model quality and thus is used for experimental evaluation.

## VI. Experimental Evaluation

For experimental evaluation, the new motion vector predictor has been incorporated into the HEVC test model HM 3.2 [1] to replace the collocated one (as explained in Section III). As an additional modification for MVP index reduction, the spatial MVPs are combined to one single MVP as explained in Section III. Table I overviews the settings used for the experimental evaluation.

TABLE I
Coding conditions used in experimental evaluation.

| | |
|---|---|
| HEVC test software | HM 3.2 |
| Profile | High efficiency |
| Picture order / GOP settings | IBBB (hierarchical QP) |
| QP-low | $\in \{22, 27, 32, 37\}$ |
| QP-high | $\in \{17, 22, 27, 32\}$ |
| Largest CU size | $64 \times 64$ |
| Smallest CU size | $8 \times 8$ |
| Number of reference frames | 4 |
| Motion search range | $64 \times 64$ |

Table II gives an overview of the used test sequences resolutions and the coding gains in terms of BD-rates [7] for two different QP-ranges. For lower qualities the QP range for the I-Slice of $\{22, 27, 32, 37\}$ has been selected. As a high quality range the QPs $\{17, 22, 27, 32\}$ are evaluated.

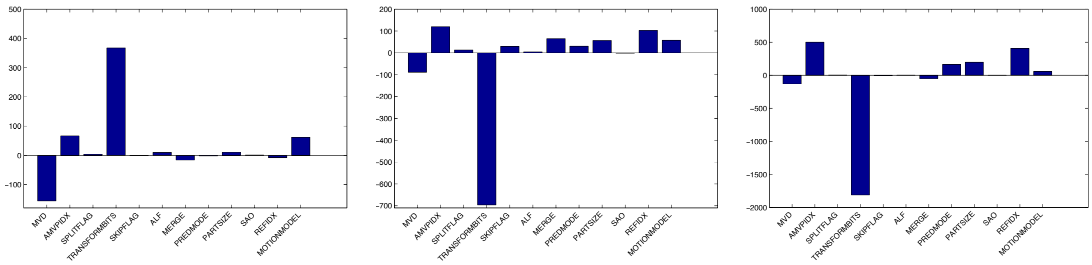
It shows that for sequences as Stefan and Waterfall with lower resolutions and thus few coding units is not as effective as for high resolution sequences as City, Blue Sky or Station. For Stanford even losses of up to 0.19% for Method 1 and 0.44% for Method 2 are observed. To analyze the bit rate differences of the reference encoder and the presented approaches, the distribution of all bins of a stream is counted, before the arithmetic coding by CABAC [8] is done. Figure 4 provides an insight on the effects that occur to the bin distributions when the collocated MVP of HM 3.2 is replaced by the parametric one. The changes are named as follows. MVD stands for motion vector difference bins. AMVPIDX counts the bins needed for signaling the motion vector predictor index. SPLITFLAG bins are needed to encode the quadtree structure of each coding unit. ALF is the count of all bins needed by the adaptive loop filter of HM 3.2. Merging allows the fusion of neighboring coding units. For signaling such a fusion, additional bins are needed. MERGE shows the amount difference of these bins. Prediction mode signaling bins are represented by PREDMODE. PARTSIZE bins are needed to describe the shape of the final Inter prediction units. A tool, called sample adaptive offset is incorporated to the HM 3.2 as an additional in-loop-filter. Bins required by that filter are represented by SAO. REFIDX stands for bins used for selecting certain reference frames for the Inter prediction.

For the Stanford sequence e.g., more bits for transformation coefficients are used (see Figure 4(a)) which leads to a higher bit rate. This results from MVPs that do not fit the motion in this video as accurate as the original HM 3.2 MVP set does. On the other hand, for sequences with complex motion like zoom, which is the main motion in the Station and Waterfall sequence, PMVP delivers highly precise MVPs for all reference frames used for prediction. Thus, by using a few more bits for reference indexing, predictions signals for the Inter mode with much higher quality can be achieved. Therefore, less transformation coefficients have to be transmitted, which leads to bit rate reductions. The City sequence is taken with a hand camera, so the motion of this sequence consists of arbitrary combinations of zoom and rotation. That is why the

TABLE II

BD-RATE FOR ALL SEQUENCES USING GMVP INSTEAD OF COLLOCATED MVP (METHOD 1) AND METHOD 1 WITH SIMPLIFIED SPATIAL MVP (METHOD 2) ON HM 3.2 (BD-LO: QP 22 TO 37, BD-HI: QP 17 TO 32)

| | | Method 1 | | | | Method 2 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | QP-low | | QP-high | | QP-low | | QP-high | |
| Sequence | Resolution | BD [%] | BD-PSNR [dB] | BD [%] | BD-PSNR [dB] | BD [%] | BD-PSNR [dB] | BD [%] | BD-PSNR [dB] |
| Stefan | 352 × 240 | −0.26 | 0.01 | −0.20 | 0.01 | −0.11 | 0.01 | −0.06 | 0.00 |
| Waterfall | 704 × 480 | 0.09 | 0.00 | −0.30 | 0.01 | 0.01 | 0.00 | −0.50 | 0.01 |
| Stanford | 720 × 480 | 0.19 | 0.00 | 0.16 | −0.01 | 0.44 | −0.01 | 0.32 | −0.01 |
| City | 1280 × 720 | −1.77 | 0.05 | −1.54 | 0.04 | −2.42 | 0.07 | −1.99 | 0.06 |
| Blue Sky | 1920 × 1080 | −0.20 | 0.01 | −0.07 | 0.00 | −0.36 | 0.01 | −0.17 | 0.01 |
| Station | 1920 × 1080 | −1.94 | 0.05 | −1.59 | 0.02 | −2.36 | 0.06 | −2.11 | 0.03 |
| Average | | −0.82 | 0.02 | −0.74 | 0.01 | −1.05 | 0.02 | −0.97 | 0.02 |



(a) For Stanford. (b) For Waterfall. (c) For Station.

Fig. 4. Bin distribution changes in per frame bin distribution between HM 3.2 and Method 1 at QP 22 for three selected test sequences.

PMVP leads to up to 2.42% of bit rate reductions.

Method 2 works better for sequences with a high amount of zoom and rotation as with the reduced amount of MVP indices, the bits for MVP signaling decreases. On the other hand, a merged spatial MVP is not that efficient for other kinds of motion as can be seen by the increasing loss for the Stanford sequence when changing from Method 1 to Method 2. Another observation is the reduced gain for lower QP values. This results from an increased amount of Intra blocks and thus a decreased amount of MVs for Inter blocks.

However, by adding MVPs based on PMMs, gains of up to 2.42% for sequences with complex motion are possible. This already shows the potential of this technique. For sequences with motion covered by conventional MVPs, an adaptive decision technique for using or not using PMVP could reduce or avoid bit rate losses.

## VII. SUMMARY

A novel prediction mode for compressing motion vectors in Inter predicted coding units is presented. The PMVP has been incorporated into the HEVC test model HM 3.2 as an additional MVPs besides the common ones. An additional PMM is transmitted for each frame to derive the new PMVP. To get PMVPs for each reference frame that is used for Inter prediction, models of previously decoded frames are combined. For each Inter block, the encoder decides which reference frame and which MVP to use to reduce the bit amount for MV prediction errors.

As PMVP covers highly complex motion but is not useful for simple translational ones, a per frame decision for using or not using PMVP has to be worked out to get even higher gains. This can easily be realized by PMM analysis and signaling if PMVPs should be used for coding a frame or not.

## 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 and ISO/IEC JTC1/SC29/WG11 document JCTVC-D500-r1.doc*, Mar 2011.

[2] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 560 –576, Jul 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, “Feature-Based Global Motion Estimation Using the Helmholtz Principle,” in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing*, Prague, Czech Republic, May 2011.

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

[8] D. Marpe, H. Schwarz, and T. Wiegand, “Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620 – 636, Jul 2003.