IMPROVED HIGHER ORDER MOTION COMPENSATION IN HEVC WITH BLOCK-TO-BLOCK TRANSLATIONAL SHIFT COMPENSATION

Cordula Heithausen, Max Bläser, Mathias Wien and Jens-Rainer Ohm

Institute of Communications Engineering
RWTH Aachen University
52056 Aachen
Germany
Email: {heithausen, blaeser, wien, ohm}@ient.rwth-aachen.de

ABSTRACT

Conventionally, complex motion in video sequences is approximated by smaller block units in order to be representable by a translational motion model. This approximation results in a fine block partitioning and a high prediction error, both at cost of more data rate than potentially necessary. A worthwhile data reduction has been shown to be achievable by adding a higher order motion model to the most recent video coding standard, High Efficiency Video Coding (HEVC). The benefit of this additional option of inter-frame prediction is due to the more accurate motion compensation as well as the usage of larger block sizes. This paper deals with more efficient encoding of higher order motion parameters in this context. The geometrically accurate prediction of higher order motion parameters from a neighbored block needs to consider the dependency of the block-to-block parameter difference based on the spatial relation between two block centers. An algorithm is introduced for correcting the translational component and reducing the difference between the actual and the predicted motion when determining higher order parameters from neighbored blocks. Additionally, a further increase of the maximum block size up to 512x512 pixels is investigated.

***Index Terms*—** higher order motion compensation, HEVC, motion parameter encoding, CTU size increase, KTA

1. INTRODUCTION

Throughout the history of video coding standards, the blockwise translational motion model has been prevalent. Other concepts employing higher order motion models could not compete against its computational simplicity and the established block-matching approaches for displacement vector estimation. For instance, approaches using continuous or partial triangular grids have been proposed in [1, 2, 3], and an estimation framework based on cubic splines is proposed in [4]. However, the rate saving prediction error reduction or gain in motion accuracy provided by these proposals tends to be ruled out by the data rate caused by additional motion parameters.

In High Efficiency Video Coding [5], larger block size options up to 64x64 pixels were introduced. [6] suggests that due to HEVC allowing a higher maximum block size than its predecessors, higher order motion models can improve the coding efficiency after all.

Motivated by [6], a motion compensation system with various higher order motion models for HEVC was proposed in [7], which provides a data rate reduction of 2.9% on average for a large set of sequences with varying amount of non-translational motion. This approach, subsequently denoted as Higher Order Motion Compensation (*HOMC*), is the starting point of the novel proposal presented in this paper.

Namely, the *Block-to-Block Translational Shift Compensation* (*BBTSC*), an improvement of the motion parameter prediction of [7], is established and evaluated.

Via inter-block change of basis transformations, a block-to-block translational shift caused by the usage of higher order motion transform predictors is compensated. Thus, higher order motion representation is approached more closely, making not only predictive motion vector coding but also merge and skip mode more suitable for HOMC.

In addition to that, the current block size limit (*coding tree unit, CTU*) of HEVC is increased and its effect on the performance of higher order motion compensation is investigated.

In Section 2 the HOMC System of [7] is described. Section 3 forms the main and novel part of this work covering the motivation, development and integration of *BBTSC*. Section 4 motivates an increased CTU size for HOMC with and without the newly introduced BBTSC. Results are presented in Section 5. Finally, conclusions are drawn and an outlook is given in Section 6.

2. HIGHER ORDER MOTION COMPENSATION (HOMC) FOR HEVC

The HOMC system consists of three main components, a higher order motion model beyond the translational one, an estimation algorithm for its parameters and a method of integrating this in the HEVC encoding/decoding processes.

2.1. Selected Higher Order Motion Models

Despite the request for a more precise motion representation than a translational one, a too sophisticated model mostly aims beyond what is needed. An evaluation of various higher order motion models in [7] showed that too many additional parameters result in an extra cost that cannot be justified by the PSNR gain per additional parameter they supply. The Zoom&Rotation Motion Model[1] as well as the Affine Motion Model on the other hand provide a worthwhile gain per additional parameter while requiring only two/four of them. Thus, these two are employed in the current investigation. Their transformation rules are given by $x_{k-1} = a_0 + a_2 x_k + a_4 y_k$ and $y_{k-1} = a_1 + a_3 y_k + a_5 x_k$, with $a_3 = a_2$ and $a_5 = -a_4$ for the Zoom&Rotation Model. Note that the HOMC is not limited to these two motion models.

2.2. Estimation of Higher Order Motion Parameters

The higher order motion parameters for HOMC are obtained by a gradient-based estimation method, detailed in [6]. Inside a PU, the pixel-wise temporal and spatial local gradients, g_t and $\boldsymbol{g}_\text{s} = (g_x, g_y)^\text{T}$, are computed as input to an equation system, which iteratively solves for the motion parameter vector $\Delta\boldsymbol{a}$, as exercised in (1). The spatial local gradients per pixel position within a PU constituting a PU-wise matrix $\boldsymbol{H}$ for the considered motion models are given in (2).

$$\Delta\boldsymbol{a} = \sum_i \Delta\boldsymbol{a}^{(i)}, \boldsymbol{G}_\text{t} = \boldsymbol{H} \cdot \Delta\boldsymbol{a}^{(i)} \Rightarrow \Delta\boldsymbol{a}^{(i)} = \boldsymbol{H}^{+} \cdot \boldsymbol{G}_\text{t} \quad (1)$$

$$\begin{aligned} \boldsymbol{h}_\text{aff}^\text{T} &= (g_x, g_y, g_x x_k, g_y y_k, g_x y_k, g_y x_k) \\ \boldsymbol{h}_\text{z\&r}^\text{T} &= (g_x, g_y, g_x x_k + g_y y_k, g_x y_k - g_y x_k) \end{aligned} \quad (2)$$

2.3. Integration of HOMC in HEVC

The purpose of HOMC is not to replace the conventional motion prediction system of HEVC, but to offer an alternative option to the encoder. After each block-matching operation, the algorithm outlined in 2.2 is performed iteratively, using the vector resulting from the previous translational motion estimation as an initialization. The decision whether to use the option of a higher order motion model or stay with the translational one is made for each block via rate-distortion comparison.

The comparably higher precision of a more complex motion model requires modification of certain HEVC components that would otherwise not satisfy the parameter accuracy. Therefore, the interpolation filters for higher order motion transformation are borrowed from SHVC [8], where rounding by a precision of $1/16$th pixel is performed. The quantization is adapted as well, applying a more precise quantization to parameters representing the higher order motion. For signaling additional motion parameters, a new flag is introduced at CU level, indicating whether a higher order motion model is used. More detailed information can be found in [7].

3. BLOCK-TO-BLOCK TRANSLATIONAL SHIFT COMPENSATION (BBTSC)

The purpose of BBTSC is the correction of translational aberration in block-to-block higher order motion parameter prediction or merging, which is explainable by the fact that the respective block center is commonly used as coordinate reference. In conventional motion vector prediction, the translational motion vector of a PU is encoded using motion vectors from spatially or temporally neighboring PUs as predictors, if available. The best motion vector predictor from a list of candidates is chosen and a motion vector difference between this predictor and the estimated motion of the current PU is calculated. Only this difference and the motion vector predictor index have to be encoded. In merge mode, vectors of candidates are adopted directly and only a residual is encoded while in skip mode not even a residual update is applied to the chosen merge candidate. The data rate increases with the size of the motion vector difference or residual to be encoded.

3.1. Prediction Distortion through Translational Shift

The HOMC system of [7] utilizes motion vector predictors similar to HEVC, which will be called *motion parameter predictors* in the following. Both higher order and merely translational motion parameter predictors can be chosen for the current PU. When the motion parameter predictor as well as the motion transform estimated for the current PU are both of higher order, it can be shown that the provided parameters are not optimal for motion parameter prediction.

In this context, it has to be considered that non-translational motion present in a reference PU will cause an additional translational shift in the current PU to which the motion parameters are extrapolated. To illustrate this effect, in the following it is assumed that a reference PU is rotated about its own center by a known rotation angle φ. This assumption can be made without loss of generality as the motion within a PU is always estimated with reference to its center M_pred.[2] In Figure 1 this simple scenario of motion referencing is depicted, where with increasing distance from the rotation center an additional translation $\Delta x, \Delta y$ occurs.

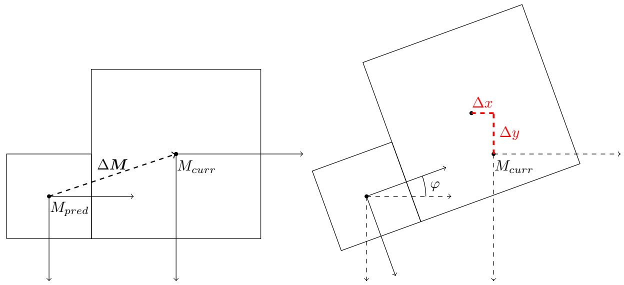

Fig. 1. Motion transform predictor from neighbored PU (right) causing additional translation shift in current PU (left).

3.2. Compensation of Translational Shift

The translational correction components $\Delta\boldsymbol{x} = \{\Delta x, \Delta y\}$ can be determined for any more complex higher order motion parameter predictors as well. The general idea is to obtain a compensated motion transform T_pred for a current block from

---

[1] reduced version of the Affine Motion Model, with four parameters

[2] The actual rotation center of an image region can be neglected when predicting the rotation parameters themselves.

the motion transform T_{ref} (3) of a reference block by describing the conversion of the origin of the coordinate system. In this case, the considered coordinate systems, originating in the centers of the two blocks, only differ by a translational shift $\Delta\boldsymbol{M} = \{\Delta M_x, \Delta M_y\}$. A translational change of basis represented by $B_{\text{curr}\to\text{ref}}$ (4) can describe their relation.

$$T_{\text{ref}} = \begin{pmatrix} a_2 & a_4 & a_0 \\ a_5 & a_3 & a_1 \\ 0 & 0 & 1 \end{pmatrix} \tag{3}$$

$$B_{\text{curr}\to\text{ref}} = \begin{pmatrix} 1 & 0 & \Delta M_x \\ 0 & 1 & \Delta M_y \\ 0 & 0 & 1 \end{pmatrix} \tag{4}$$

A better motion transform predictor for any block-center-related pixel within the current block is obtained by

1. transforming its coordinates from the current coordinate system into the coordinate system of the neighboring prediction block via $B_{\text{curr}\to\text{ref}}$ (4),
2. applying the higher order motion matrix of the neighboring prediction block T_{ref} (3) to the transformed pixel,
3. transforming the result back to the coordinate system of the current block by multiplying it with the inverse change of basis matrix $B^{-1}_{\text{curr}\to\text{ref}} = B_{\text{ref}\to\text{curr}}$.

This procedure is described by equation (5), resulting in (6).

$$T_{\text{pred}} = B^{-1}_{\text{curr}\to\text{ref}} \cdot T_{\text{ref}} \cdot B_{\text{curr}\to\text{ref}} \tag{5}$$

$$= \begin{pmatrix} a_2 & a_4 & \Delta M_x(a_2 - 1) + \Delta M_y a_4 + a_0 \\ a_5 & a_3 & \Delta M_x a_5 + \Delta M_y (a_3 - 1) + a_1 \end{pmatrix} \tag{6}$$

Comparing (3) to (6), it can be noticed that for any higher order transformation matrix the resulting compensated version differs only in the translational components. These translational shifts are given by (7) with $a_3 = a_2$ and $a_5 = -a_4$ for the Zoom&Rotation Motion Model.

$$\begin{aligned} \Delta a_0 &= \Delta \text{M}_x(a_2 - 1) + \Delta \text{M}_y a_4 \\ \Delta a_1 &= \Delta \text{M}_x a_5 + \Delta \text{M}_y(a_3 - 1) \end{aligned} \tag{7}$$

3.3. Integration of BBTSC into HOMC in HEVC

Since its basic version, the HOMC algorithm [7] with BBTSC has been migrated from version HM-12.1 [9] of the HEVC reference software to the KTA 1.0[3] version of HM-14.0 [10]. The BBTSC method described in 3.2 is applied to each higher order motion transform predictor prior to being added to the prediction candidate list that is constructed for the current PU. The resulting compensated higher order motion parameter predictors are added to the list as well and will substitute the non-BBTSC predictors whenever the current PU itself contains non-translational motion.

Also, during collection of the merge candidates, the BBTSC is applied to each candidate with a non-translational motion transform, independently of the motion of the current PU.

No information on the BBTSC results has to be transmitted as the algorithm is executed on both the encoder and the decoder side. Due to the higher accuracy of BBTSC the merge mode is expected to be used more frequently than in HOMC [7].

4. VARIATION OF MAXIMUM BLOCK SIZE

As outlined in 2.3, the former HOMC of [7] entailed adjustments to the quantization and interpolation scheme of the HEVC Test Model essential for preservation of the higher order motion parameter accuracy. The upper limit of the PU size of 64x64 pixels on the other hand was abided. It could be observed that whenever the encoder chose to employ one of the higher order motion models, the PU sizes predominantly increased and were often set to the maximum. It can be concluded that the HEVC encoder favors a larger PU with one set of higher order motion parameters over several smaller PUs with one translational vector each. This leads to the assumption that the maximum benefit of CTU size increase had not been reached yet. Thus, the algorithm presented in Section 3 is investigated in combination with a CTU size increase of up to 512x512 pixels to further improve the coding efficiency of HOMC [7] and/or BBTSC.

5. EVALUATION AND RESULTS

As the basic HOMC had already been tested in [7] on sequences with variable amount of non-translational motion, the tests run for the current proposal and their results found below served to measure the improvement that HOMC enhanced by BBTSC can provide on mainly non-translational motion sequences that sufficiently employ HOMC.

5.1. Test Conditions

The test set consists of 100 frames of the sequences *Spincalendar*[5,11], *SlideShow*[5,13], *Cactus*[4,10], *Tempete*[8,11], *BigShips*[5,10], *ChinaSpeed*[7,11], *BQSquare*[6,9], *BlueSky*[4,12], *Station*[4,12] and *Jets*[5,12]. All tests are run in *low delay P* mode (*LDP*), with a maximum TU size of 64x64 pixels. The number of iterations in (1) is set to $I = 4$ and the quantization factor for higher order motion parameters is set to $q = 256$, as both settings proved successful in [7]. Tested CTU sizes are 64x64, 128x128, 256x256 and 512x512 pixels. Four operating points are measured ($QP = \{28, 32, 36, 40\}$).

5.2. Improvement through BBTSC

The results of testing BBTSC as an extension to HOMC are presented in Table 1. BBTSC increases the rate reduction of HOMC by an average of 5.3% and 5.8% employing a CTU size of 64x64 pixels as well as 3.1% and 2.7% employing a CTU size of 128x128 pixels for the Affine (Aff.) and Zoom&Rotation (Z&R) Motion Model, respectively. Best results were achieved for test sequences *Spincalendar* and *Station*, showing almost exclusively rotation and zoom motion, respectively. Figure 2 shows results for *Station*. Investigation of the coding statistics shows that by making the parameter prediction of HOMC more precise through BBTSC, both merge and skip mode become more worthwhile and thus are chosen more frequently. Exemplarily for this, Table 2 shows the area encoded in merge and skip mode for two sequences.

---

[3] Besides the CTU size increase and a TU size of 64x64, all additional features implemented by the developers of HM14.0 KTA 1.0 were disabled, such that the software otherwise operates like HM 14.0.

[4] HD1080 [5] HD720 [6] WQVGA [7] XGA [8] CIF [9] 60fps [10] 50fps [11] 30fps [12] 25fps [13] 20fps

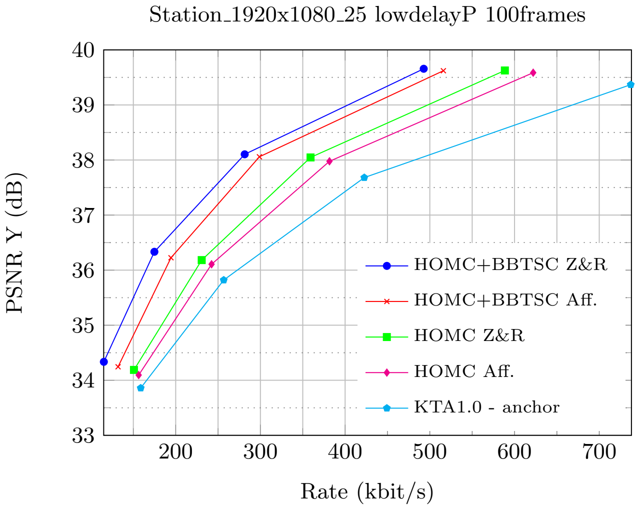

Fig. 2. BD-rate of *Station, LDP, CTU=64, QP=28,32,36,40.*

	area encoded [%] without/with BBTSC			
mode	QP=28	QP=32	QP=36	QP=40
a) MRG	7.4/**16.6**	5.7/**9.3**	5.1/**7.2**	4.6/**5.7**
SKIP	31.7/**67.0**	49.9/**78.9**	64.5/ **83.3**	75.4/**85.8**
b) MRG	13.0/**24.9**	9.9/**21.4**	7.5/**14.1**	6.1/**10.2**
SKIP	21.1/**32.4**	26.8/**45.8**	41.5/**61.8**	60.5/**72.8**

Table 2. Area [%] encoded in merge/skip mode without/with BBTSC for a) *Station*, b) *Spincalendar*, *Z&R, LDP, CTU=64*.

5.3. Results of Maximum Block Size Variation

Figure 3 depicts the coding efficiency improvement of HOMC through CTU size enlargement. The CTU size of 512x512 pixels provides an average gain over the CTU size of 64x64 pixels of 5.3% and 5.6% for the Affine and Zoom&Rotation Motion Model, respectively. The BBTSC extension however shows its best performance employing the smallest CTU size of 64x64 pixels. It reaches an improvement over HOMC without BBTSC when using a CTU size limit of up to 256x256 pixels for both the Affine and the Zoom&Rotation Motion Model. Obviously, the better prediction of parameters becomes less relevant when block sizes are larger, such that less parameters have to be coded.

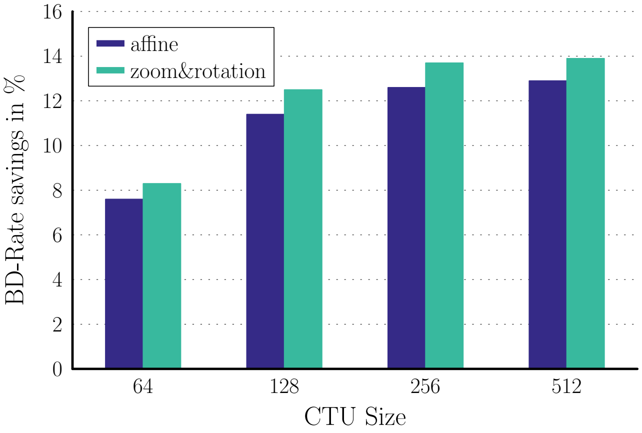

Fig. 3. Rate reduction of HOMC on top of KTA over KTA without HOMC for different CTU sizes.

6. CONCLUSION AND OUTLOOK

In this paper, an improvement to the previously proposed HOMC algorithm was presented. By applying the newly developed BBTSC to non-translational motion parameter predictors and merge candidates, the motion prediction is refined, making AMVP, merge mode and skip mode more suitable for HOMC. The tests showed an increase of bit rate reduction of more than five percent compared to HOMC without BBTSC. The overall improvement over the translational motion compensation with the standardized CTU size reached 14.1%. Supporting larger CTU sizes, the rate reduction of HOMC+BBTSC could further be improved, up to 15.2%. The next steps towards further optimizing HOMC+BBTSC for HEVC will be including a temporal motion transform predictor as well as implementing interchangeable higher order motion models on frame or even CTU level and model-independent predictor choices. Also, an extension of BBTSC to Bi-Prediction will be investigated.

Sequence	CTU 64		CTU 128		CTU 256		CTU 512	
	Aff.	Z&R	Aff.	Z&R	Aff.	Z&R	Aff.	Z&R
Spincal.	28.6(**11.4**)	33.1(**12.6**)	30.8(**5.7**)	32.6(**4.6**)	27.3(**-0.0**)	29.2(**-1.1**)	25.9(**-2.2**)	28.0(**-2.8**)
SlideSh.	18.9(**3.9**)	15.6(**1.6**)	18.5(**1.3**)	15.3(**-0.2**)	16.9(**-0.4**)	14.5(**-1.3**)	16.3(**-0.6**)	13.7(**-1.0**)
Cactus	9.0(**2.1**)	9.4(**1.7**)	8.1(**0.6**)	8.5(**0.5**)	7.4(**-0.2**)	7.9(**-0.2**)	7.3(**-0.4**)	7.8(**-0.4**)
Tempete	8.3(**1.2**)	10.7(**1.8**)	8.1(**0.5**)	9.8(**0.1**)	8.1(**0.5**)	9.8(**0.1**)	8.1(**0.5**)	9.8(**0.1**)
BigShips	0.6(**-0.2**)	0.8(**0.2**)	0.8(**-0.3**)	1.2(**0.0**)	1.0(**-0.2**)	1.1(**-0.4**)	0.7(**-0.5**)	1.0(**-0.4**)
ChinaSp.	0.0(**-0.1**)	0.1(**0.1**)	-0.0(**-0.0**)	-0.0(**0.1**)	0.1(**0.1**)	0.0(**0.0**)	-0.0(**0.0**)	-0.1(**-0.2**)
BQSq.	5.3(**0.9**)	6.5(**1.0**)	5.0(**0.3**)	6.5(**0.3**)	5.0(**0.3**)	6.5(**0.3**)	5.0(**0.3**)	6.5(**0.3**)
BlueSky	9.8(**4.7**)	11.7(**4.4**)	11.3(**1.9**)	11.8(**1.3**)	9.8(**-0.0**)	10.2(**-0.4**)	9.2(**-0.8**)	9.5(**-1.1**)
Station	32.9(**21.5**)	32.9(**24.4**)	39.7(**13.9**)	42.0(**12.9**)	36.8(**5.5**)	38.5(**4.2**)	32.6(**0.3**)	34.6(**-0.6**)
Jets	15.9(**7.5**)	20.3(**9.9**)	22.3(**7.0**)	23.9(**7.0**)	21.0(**1.8**)	22.2(**1.6**)	18.6(**-1.1**)	19.8(**-1.6**)
average	12.9(**5.3**)	14.1(**5.8**)	14.5(**3.1**)	15.2(**2.7**)	13.3(**0.7**)	14.0(**0.3**)	12.4(**-0.5**)	13.1(**-0.8**)

Table 1. Rate reduction [%] of HOMC with BBTSC, compared against KTA (**compared against HOMC without BBTSC**).

7. REFERENCES

[1] Y. Nakaya and H. Harashima, "Iterative motion estimation method using triangular patches for motion compensation," in *Visual Communications,'91, Boston, MA*. International Society for Optics and Photonics, 1991, pp. 546–557.

[2] H. Huang, J.W. Woods, Y. Zhao, and H. Bai, "Control-point representation and differential coding affine-motion compensation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 23, no. 10, pp. 1651–1660, Oct 2013.

[3] G.J. Sullivan and R. Baker, "Motion compensation for video compression using control grid interpolation," in *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*, Apr 1991, vol. 4, pp. 2713–2716.

[4] H. Lakshman, H. Schwarz, and T. Wiegand, "Adaptive motion model selection using a cubic spline based estimation framework," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010, pp. 805–808.

[5] Mathias Wien, *High Efficiency Video Coding – Coding Tools and Specification*, Springer, Berlin, Heidelberg, Sept. 2014.

[6] M. Narroschke and R. Swoboda, "Extending HEVC by an affine motion model," in *PCS*, 2013, pp. 321–324.

[7] Cordula Heithausen and Jan Hendrik Vorwerk, "Motion compensation with higher order motion models for HEVC," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP '15*, Brisbane, Australia, April 2015, IEEE, Piscataway.

[8] Joint Collaborative Team on Video Coding, "H.265 : High efficiency video coding," http://www.itu.int/rec/T-REC-H.265-201504-I/en.

[9] Joint Collaborative Team on Video Coding, "HM reference software 12.1," https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware.

[10] Joint Collaborative Team on Video Coding, "HM 14.0 KTA 1.0 software," https://vceg.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/HM-14.0-KTA-1.0.