Distance Scaling of Higher Order Motion Parameters in an Extension of HEVC

Cordula Heithausen, Max Bläser, Mathias Wien
Institut für Nachrichtentechnik
RWTH Aachen University, GERMANY
{heithausen, blaeser, wien}@ient.rwth-aachen.de

***Abstract*—Complex motion in video sequences, such as rotation and scaling, conventionally approximated translationally, can more efficiently be represented by higher order motion models. An important contribution to the efficiency of the higher order motion compensation approach is the prediction and encoding of the additional motion parameters. The additional data rate caused by an increased amount of motion parameters has to be kept small for the higher order motion model to outperform the translational one within blocks of complex motion. Addressing an aspects of motion prediction yet to be adjusted to higher order motion, this paper proposes an advanced method of distance scaling of higher order motion parameters. Just as translational motion vector predictors require scaling whenever the block inheriting them has a different distance to its reference picture than the block predicted from, higher order motion parameter prediction can profit from such scaling as well. However, the distance scaling of higher order motion parameters cannot be applied to the elements of the motion model transformation matrix directly, but is performed on separate higher order motion components provided by a preceding transformation matrix decomposition. Accordingly, the objective of this work is to introduce and evaluate a procedure for higher order motion distance scaling. The resulting rate reduction of about 2% on average and up to over 6% attains a further improvement on the higher order motion compensation in an extension of HEVC.**

I. Introduction

In all prevalent block-based video codecs, motion compensation almost exclusively employs the established translational motion model in combination with a block-matching approach. Though several proposals were made ([1], [2], [3], [4]) over the past years, higher order motion compensation could never quite assert itself due to its demand for a different estimation approach and a non-profitable coding cost increase caused by additional motion parameters. When an increased maximum block size of 64x64 pixels was introduced with the release of the High Efficiency Video Coding (HEVC) standard in 2013, a new opportunity for higher order motion compensation emerged. As outlined by [5], the bigger blocks and new block partition variety make a higher order motion model more lucrative, substituting an approximation by many small translational prediction units with a larger prediction unit modeled by a more complex motion model. Based on that suggestion, a *higher order motion compensation* system (hereafter referred to as *HOMC*) was proposed in [6] and further improved in [7], reaching average rate reductions for higher order motion sequences of over 14% using a maximum *coding tree unit* (CTU) size of 64x64 pixels and over 15% using the quadrupled CTU size of 128x128 pixels. While the parameter estimation of those proposals, adapted from [5], already provides a sufficient level of precision, there is still room for improvement in the parameter prediction. One issue that still has to be exploited is the distance scaling of the additional higher order motion parameters. Distance scaling is deployed when a motion vector predictor references a more or less distant picture than the current *prediction unit* (*PU*) does. Up to this point, the HOMC system [7] only applies distance scaling to the translational motion parameters of a higher order motion model, equal to the conventional motion vector scaling as used in HEVC. Therefore, it is the idea of the research presented in this paper to take HOMC as its basis and enhance it by focusing on the not yet optimized aspect of *higher order motion distance scaling* (hereafter referred to as *HODS*). It can be expected that a reasonable transfer of the existing translational distance scaling to the higher order motion parameter prediction of HOMC will achieve some efficiency gain.

Section II outlines the HOMC system [7] this work is based on. The actual distance scaling of higher order motion parameters is introduced in Section III. The results of testing that distance scaling procedure in the context of HOMC in an extension of HEVC are presented and discussed in Section IV. In Section V finalizing conclusions and outlooks are given.

II. Higher Order Motion Compensation (HOMC) for HEVC

As the work presented in this paper is based on HOMC for HEVC, this section will give a summary of that system to help comprehend the context of this paper. More details can be found in the preceding publications [6] and [7]. The HOMC system requires a motion model beyond the translational one as well as an appropriate parameter estimation algorithm. Both components are outlined in the next two subsections, followed by a description of the integration of HOMC into the HEVC encoding/decoding processes.

A. Selected Higher Order Motion Models

The main demand a higher order motion model has to satisfy is that of achieving a more accurate representation of higher order motion than the translational motion model allows, which merely approximates any complex motion type translationally. But even though the motion type coverage and motion estimation precision grow with the amount of additional motion parameters, choosing a model with many additional parameters does not necessarily contribute more coding efficiency to HOMC. On the contrary, a too sophisticated model mostly aims beyond what is needed. From a certain level of motion model complexity onwards, the PSNR gain per additional parameter cannot justify the coding cost caused by too many additional parameters. A motion model evaluation in [6] revealed that the Zoom&Rotation Motion Model[1] with two additional parameters and the Affine Motion Model with four additional parameters provide a balance between cost and PSNR gain that serves the purpose of HOMC best. Therefore, they are both chosen as the most suitable motion models to compete with the translational motion model in an extension of HEVC. Relating to Figure 1, the Affine Motion Model covers motion forms 1a to 1e, the Zoom&Rotation Motion Model allows for 1a to 1c. Though only these two motion models were implemented for previous and current investigations, HOMC is not limited to them.

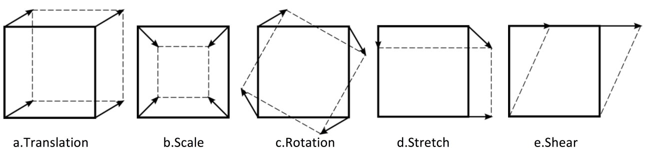

Fig. 1. Examples for affine motion.

B. Estimation of Higher Order Motion Parameters

Although higher order motion estimation is performed block-wise, the established block-matching approach is not advisable for estimating any form of motion beyond translational shifts as it would be too computationally complex. Instead, in order to obtain the parameters of the higher order motion models listed in II-A, a gradient-based estimation method is employed, detailed in [5]. In short, within a PU a temporal as well as a spatial local gradient is determined for each contained pixel. All local gradients together add up to an equation system that is solved for $\Delta \boldsymbol{a}$ in (1), $\Delta \boldsymbol{a}$ being the motion parameters of the PU. As the motion is not linear, the equation system is solved iteratively (2).

$$\boldsymbol{G}_{\mathrm{t}} = \boldsymbol{H} \cdot \Delta \boldsymbol{a} \Rightarrow \Delta \boldsymbol{a} = \boldsymbol{H}^{+} \cdot \boldsymbol{G}_{\mathrm{t}} \tag{1}$$

$$\Delta \boldsymbol{a} = \sum_{i}^{I} \Delta \boldsymbol{a}^{(i)} \tag{2}$$

C. Integration of HOMC into HEVC

The HOMC [6] system is integrated into the HEVC encoder process succeeding the conventional translational motion estimation. That way, the best match resulting from the translational block-matching is available as an initialization for the iterative gradient-based estimation algorithm described in II-B. Once the higher order motion parameters of one of the models listed in II-A are estimated, a rate-distortion comparison with the conventional translational motion parameters is conducted. Whenever the decision is in favor of the higher order motion, the additional parameters are stored in the motion vector of the current block and a corresponding *higher order motion flag* is set. If a chosen motion vector predictor is of higher order, a change of basis transformation is applied adjusting the higher order motion to the coordinate system of the current block (*Block-to-Block Translational Shift Compensation - BBTSC*, detailed in [7]). As the precision of HOMC, being its incentive, has to be preserved, the interpolation and quantization of HEVC is refined in order to satisfy the parameter accuracy of a more complex motion representation. More precise interpolation filters of $1/16th$-pixel precision are introduced, adopted from the scalable version of HEVC (SHVC [8]), and non-translational motion parameters are quantized with an increased quantization factor. For more detailed information on the HOMC integration into HEVC, please refer to [6] and [7].

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

III. Distance Scaling Of Higher Order Motion Parameters (HODS)

In HEVC, to reasonably utilize motion vector predictors independently from Picture Order Count (*POC*) differences, so-called *distance scaling* is applied to the parameters of motion vector predictors whenever the POC differences of the current and prediction PU to their reference pictures are not identical. Thus, the distance scaling can be required by temporal as well as spatial motion vector predictors [9]. Due to the introduction of HOMC into HEVC, the *motion vector predictor* will in the following be called *motion parameter predictor*, as it is of higher dimension than the conventional translational vector. In the HOMC of [6] and [7], the motion parameter predictors undergo the standardized distance scaling of HEVC, scaling only the two translational parameters a_0 and a_1 of the motion model transformation matrix, equal to x and y of the translational motion vector. All higher order motion parameters remain unchanged even when a difference in POC distances occurs. Excluding all parameters a_p with $p > 1$ from the distance scaling process was attributed to the fact that these parameters cannot simply be scaled by mere multiplication with a scaling factor but the scaling would have to be performed on specific higher order motion components representing different forms of higher order motion. While the HOMC system already performs conveniently though neglecting a possible distance scaling of higher order motion parameters, the development and employment of a distance scaling algorithm for the additional parameters introduced by a higher order motion model can be expected to further enhance the HOMC performance. Motivated by the possible improvement of higher order motion prediction accuracy, HODS is introduced and explained in the following subsections.

It has to be pointed out that, when separately scaling the matrices that exclusively represent just one motion component each, it is not taken into account that distance scaled higher order motion parameters indeed influence both each other and the translational parameters. With regard to mathematical precision, these influences would have to be included in the scaling process. However, the HODS presented here considers these inter-dependencies negligible in most cases and provides a suitable approach of reasonable complexity still providing significant compression gain.

A. Algorithm

As higher order motion parameters have different impacts on the overall motion, the HODS aims at scaling each form of higher order motion separately. These different forms of higher order motion are described by specific higher oder motion components, such as a rotation angle ϕ describing the rotation, scaling factors s_x and s_y describing both homogeneous ($s_x = s_y$) and non-homogenous ($s_x \neq s_y$) scaling and a shearing factor k describing the shear within a transform. To attain these components, a motion model transformation matrix decomposition is required. For reasons of simplification, the HODS is illustrated by the example of the Affine Motion Model but can be applied to the less complex Zoom&Rotation Model as well.

$$\begin{pmatrix} a_2 & a_4 & a_0 \\ a_5 & a_3 & a_1 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} a_2 & a_4 \\ a_5 & a_3 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \tag{3}$$

As shown in (3), the transformation of any image pixel $\boldsymbol{x} = (x, y)^T$ via an affine matrix $\boldsymbol{A}_{\text{aff}}$ can be expressed as $\boldsymbol{A}_{\text{aff}}\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$ with $\boldsymbol{A}$ covering all higher order motion components of the transform while $\boldsymbol{b}$ represents the translational component. Since the distance scaling of $\boldsymbol{b}$ is already handled by the conventional distance scaling specified in HEVC, only $\boldsymbol{A}$ is of interest for the HODS. Consequently, the quantized translational elements a_0 and a_1, are neglected by the following scaling algorithm which is exclusively applied to the higher order elements a_2, a_3, a_4 and a_5. The algorithm is executed in six steps per motion parameter predictor candidate:

1) Detection of HODS necessity and determination of HODS factor D.
2) Dequantization of all a_p with $p > 1$.
3) Decomposition of non-translational part of higher order motion transformation matrix
$$\boldsymbol{A} = \begin{pmatrix} a_2 & a_4 \\ a_5 & a_3 \end{pmatrix} \tag{4}$$
into higher order motion components ϕ, s_x, s_y and k.
4) Distance scaling of ϕ, s_x, s_y and k by distance scaling factor D, resulting in $\tilde{\phi}$, $\tilde{s}_x$, $\tilde{s}_y$ and $\tilde{k}$.
5) Re-composition of distance-scaled matrix
$$\tilde{\boldsymbol{A}} = \begin{pmatrix} \tilde{a}_2 & \tilde{a}_4 \\ \tilde{a}_5 & \tilde{a}_3 \end{pmatrix}. \tag{5}$$
6) Quantization of all $\tilde{a}_p$ with $p > 1$.

Some steps of the algorithm, namely the decomposition of $\boldsymbol{A}$ and re-composition of $\tilde{\boldsymbol{A}}$, are ambiguous as they depend on the order in which separate matrices, representing the scale, shear and rotation that is contained in the affine transformation matrix, are multiplied. The multiplication order that was decided upon for this paper is $\boldsymbol{A} = \boldsymbol{A}_{\text{rot}} \cdot \boldsymbol{A}_{\text{shear}} \cdot \boldsymbol{A}_{\text{scale}}$, given in (6). Thus, it is assumed that a block undergoing an affine transformation would first be scaled, then sheared and then rotated, resulting in the cumulative transformation matrix (7). Other multiplication orders are valid as well as long as an order once determined is complied with throughout the entire HODS procedure.

$$\begin{pmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{pmatrix} \cdot \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \tag{6}$$

$$= \begin{pmatrix} s_x \cos(\phi) & s_y k \cos(\phi) - s_y \sin(\phi) \\ s_x \sin(\phi) & s_y k \sin(\phi) + s_y \cos(\phi) \end{pmatrix} \tag{7}$$

B. De-/Quantization of Higher Order Motion Parameters (Steps 2 and 6)

The higher order motion parameters of a motion parameter predictor stored in quantized form are dequantized before undergoing the HODS procedure. Thereby, decomposition, distance scaling and re-composition are performed on double precision values so that the accuracy of the higher order motion transform is retained. Afterwards, the distance-scaled parameters are quantized again. While the specified quantization factor $q = 4$ is applied to the translational parameters of the affine transform, a finer quantization with a bigger factor q is assigned to all a_p with $p > 1$. Further details on the higher order motion quantization are found in [6].

C. POC Distance Comparison and Higher Order Distance Scaling Factor (Step1)

The HODS factor that the individual higher order motion components are scaled with whenever distance scaling is required depends on two different picture distances. The picture distance of the picture containing the current block to the picture containing its reference block is $\Delta\text{POC}_{\text{curr}}$, computed by (8). Similarly, $\Delta\text{POC}_{\text{pred}}$ describes the distance of the picture containing the motion parameter prediction block to the picture containing the reference block of the respective motion parameter predictor, given in (9). The ratio of $\Delta\text{POC}_{\text{curr}}$ divided by $\Delta\text{POC}_{\text{pred}}$ results in the distance scaling factor D, given in (10). Values of D can range from -16 to 16 as specified in the translational distance scaling of HEVC, $D = 1$ implies that $\Delta\text{POC}_{\text{curr}} = \Delta\text{POC}_{\text{pred}}$ and no HODS is needed. A value of D smaller than 1 means the motion parameter predictor is referencing a picture with higher POC distance than the current PU is, i.e. its higher order motion parameters have to be scaled down. Respectively, higher order motion parameters are scaled up when $D > 1$. Furthermore, a negative D compensates a difference in reference picture direction (forward vs. backward). Examples for three different distance scaling scenarios are illustrated in Figures 2, 3 and 4. The white block marks the current PU while the shaded, dash-rimmed block is the PU of the motion parameter predictor. As distance scaling is applied to spatial as well as temporal motion predictor candidates, both cases are depicted, a temporal (collocated) predictor in Figure 2 and a spatial one in Figures 3 and 4.

$$\Delta\text{POC}_{\text{curr}} = \text{POC}_{\text{curr}} - \text{POC}_{\text{currRef}} \tag{8}$$

$$\Delta\mathrm{POC}_{\mathrm{pred}} = \mathrm{POC}_{\mathrm{pred}} - \mathrm{POC}_{\mathrm{predRef}} \tag{9}$$

$$D = \frac{\Delta\mathrm{POC}_{\mathrm{curr}}}{\Delta\mathrm{POC}_{\mathrm{pred}}} \tag{10}$$

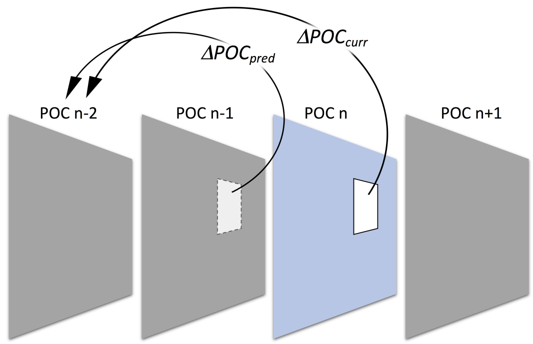

Fig. 2. Example for distance scaling with $D = 2$.

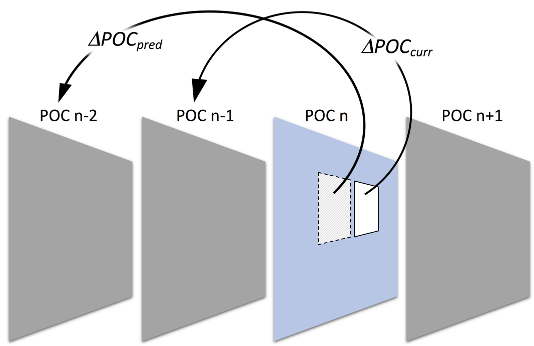

Fig. 3. Example for distance scaling with $D = \frac{1}{2}$.

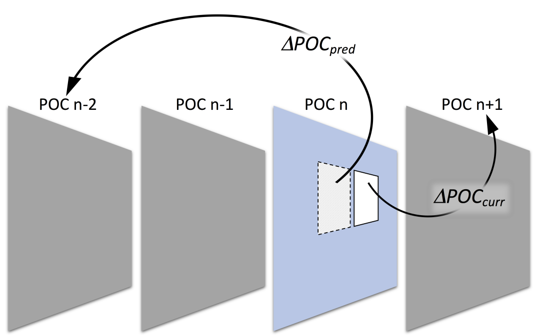

Fig. 4. Example for distance scaling with $D = -\frac{1}{2}$.

D. Transformation Matrix Decomposition (Step 3)

The higher order motion components s_x, s_y, k and ϕ are computed by equations (11), derived from the comparison of transformation matrices (4) and (7).

$$\begin{gathered} s_x = \mathrm{sgn}(a_2)\cdot\sqrt{a_2^2 + a_5^2},\ s_y = \frac{a_2 a_3 - a_4 a_5}{s_x} \\ k = \frac{a_2 a_4 + a_3 a_5}{a_2 a_3 - a_4 a_5},\ \phi = \mathrm{atan2}(a_5, a_2) \end{gathered} \tag{11}$$

E. Distance Scaling of Motion Components (Step 4)

The mathematical operation of applying the HODS factor D to the individual higher order motion components depends on how they each alter with varying POC distance. Their modification is determined by the multiplication of the respective matrices with themselves. It can be observed that s_x, s_y, k and ϕ are either additive or multiplicative when the respective transforms A_{scale}, A_{shear} and A_{rot} are exponentiated.
A scaling matrix multiplied with itself results in the scaling factors being multiplied as in (12), i.e. scaling is multiplicative.The shearing factor on the other hand adds up when the shearing matrix is multiplied with itself as in (13), i.e. shearing is additive. Likewise, rotation is additive, as the product of a rotation matrix multiplied with itself can be expressed by a doubling of the rotation angle[2] as in (14).

$$\boldsymbol{A}_{\mathrm{scale}}^D = \prod_1^D \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} = \begin{pmatrix} s_x^D & 0 \\ 0 & s_y^D \end{pmatrix} =: \begin{pmatrix} \tilde{s}_x & 0 \\ 0 & \tilde{s}_y \end{pmatrix} \tag{12}$$

$$\boldsymbol{A}_{\mathrm{shear}}^D = \prod_1^D \begin{pmatrix} 1 & k \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & D\cdot k \\ 0 & 1 \end{pmatrix} =: \begin{pmatrix} 1 & \tilde{k} \\ 0 & 1 \end{pmatrix} \tag{13}$$

$$\begin{aligned} \boldsymbol{A}_{\mathrm{rot}}^D = \prod_1^D \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} &= \begin{pmatrix} \cos(D\phi) & -\sin(D\phi) \\ \sin(D\phi) & \cos(D\phi) \end{pmatrix} \\ &=: \begin{pmatrix} \cos\tilde{\phi} & -\sin\tilde{\phi} \\ \sin\tilde{\phi} & \cos\tilde{\phi} \end{pmatrix} \end{aligned} \tag{14}$$

F. Distance-scaled Transf. Matrix Re-Composition (Step 5)

In a final step, the sought distance-scaled, non-translational part of the affine transformation matrix $\tilde{\boldsymbol{A}}_{\mathbf{aff}}$, $\tilde{\boldsymbol{A}}$ as defined in (5), is computed from $\tilde{s}_x$, $\tilde{s}_y$, $\tilde{k}$ and $\tilde{\phi}$ by the equations (15).

$$\begin{aligned} \tilde{a}_2 &= \tilde{s}_x\cos(\tilde{\phi}), \quad \tilde{a}_4 = \tilde{s}_y\tilde{k}\cos(\tilde{\phi}) - \tilde{s}_y\sin(\tilde{\phi}) \\ \tilde{a}_5 &= \tilde{s}_x\sin(\tilde{\phi}), \quad \tilde{a}_3 = \tilde{s}_y\tilde{k}\sin(\tilde{\phi}) + \tilde{s}_y\cos(\tilde{\phi}) \end{aligned} \tag{15}$$

G. Implementation of HODS

The HODS is applied to each motion parameter predictor candidate, temporal and spatial, that provides higher order motion parameters. The HODS algorithm is therefore integrated at according sections within the motion parameter predictor candidate list composition of the HM14.0-KTA1.0 software, which is an extension of HEVC providing advanced coding tools beyond the standardized HEVC. Prior to the HODS implementation, the HOMC [6] was integrated as its basis.

IV. Evaluation and Results

A. Test Conditions and Intentions

The test set consists of 100 frames of the ten sequences *Spincalendar*[4,9], *SlideShow*[4,11], *Cactus*[3,8], *Tempete*[6,9], *BQSquare*[5,7], *BlueSky*[3,10], *Station*[3,10], *Jets*[4,10], *TractorPart*[3,10] and *ShieldsPart*[3,8]. The chosen sequences all contain forms of higher order motion as the HOMC system is geared to such. As it was already shown in [6] that HOMC

[2] via trigonometric double-angle formulae

Sequence	CTU 64		CTU 128		CTU 256		CTU 512	
	Aff.	Z&R	Aff.	Z&R	Aff.	Z&R	Aff.	Z&R
Spincal.	31.4(**3.8**)	35.2(**3.1**)	32.3(**2.1**)	33.9(**1.9**)	28.2(**1.3**)	30.2(**1.4**)	26.5(**0.8**)	28.0(**-0.0**)
SlideSh.	19.2(**0.3**)	15.9(**0.4**)	18.4(**-0.1**)	15.3(**-0.0**)	17.1(**0.2**)	13.8(**-0.8**)	16.0(**-0.3**)	13.7(**-0.0**)
Cactus	9.1(**0.2**)	9.6(**0.2**)	8.3(**0.1**)	8.6(**0.2**)	7.6(**0.2**)	8.0(**0.2**)	7.4(**0.1**)	7.9(**0.1**)
Tempete	9.0(**0.8**)	11.1(**0.4**)	8.2(**0.2**)	9.6(**-0.2**)	8.2(**0.2**)	9.6(**-0.2**)	8.2(**0.2**)	9.6(**-0.2**)
BQSq.	5.4(**0.2**)	6.6(**0.1**)	5.5(**0.5**)	6.3(**-0.2**)	5.5(**0.5**)	6.3(**-0.2**)	5.5(**0.5**)	6.3(**-0.2**)
BlueSky	10.7(**1.0**)	12.3(**0.7**)	11.7(**0.5**)	12.2(**0.4**)	10.2(**0.4**)	10.4(**0.3**)	9.3(**0.2**)	9.6(**0.1**)
Station	37.3(**6.4**)	41.8(**3.0**)	40.7(**1.6**)	42.5(**0.9**)	37.1(**0.5**)	39.0(**0.7**)	33.0(**0.7**)	34.8(**0.4**)
Jets	16.8(**1.1**)	21.5(**1.4**)	22.9(**0.7**)	24.8(**1.1**)	20.8(**0.2**)	22.4(**0.4**)	19.2(**0.7**)	20.2(**0.5**)
TractorPart	29.2(**3.7**)	33.2(**1.9**)	32.7(**1.2**)	33.5(**0.5**)	29.4(**0.5**)	30.7(**0.3**)	27.0(**0.3**)	28.7(**0.5**)
ShieldsPart	21.1(**2.4**)	26.5(**1.5**)	26.6(**1.1**)	28.5(**0.5**)	24.5(**1.1**)	25.4(**0.3**)	22.1(**0.8**)	13.7(**0.6**)
average	18.9(**2.0**)	21.4(**1.3**)	20.7(**0.8**)	21.5(**0.5**)	18.9(**0.5**)	19.6(**0.2**)	17.4(**0.4**)	17.3(**0.2**)

TABLE I

RATE REDUCTION [%] OF HOMC+HODS, COMPARED AGAINST KTA (**COMPARED AGAINST HOMC WITHOUT HODS**).

does not cause any significant disadvantage for mainly or exclusively translational sequences, the evaluation of HODS aims at increasing the HOMC efficiency for sequences likely to employ it. All tests are run in *Low Delay P* mode (*LDP*), with a maximum TU size of 64x64 pixels. The number of iterations in (2) is set to $I = 4$ and the higher order quantization factor is set to $q = 256$, as both settings proved successful in [6]. Tested CTU widths are 64, 128, 256 and 512 pixels, as bigger maximum CTU sizes generated additional gains in [7]. Four operating points are measured ($QP = \{28, 32, 36, 40\}$).

B. Test Results and Evaluation

The rate reductions reached through HOMC+HODS in comparison to HM14.0-KTA1.0 without HOMC and also compared to HOMC without HODS are collected in Table II, for each sequence and also averaged over the test set.

HOMC+HODS vs. HM14-KTA: The combination HOMC and HODS achieves an average of over 17% rate reduction for all tested CTU sizes and both higher order motion models. The best averaged performance is reached by a maximum CTU size of 128x128 pixels with a gain of 20.7% for the Affine and 21.5% for the Zoom&Rotation Motion Model.

HOMC+HODS vs. prior HOMC: The averaged rate reductions show that the integration of HODS is profitable for all tested CTU sizes and most profitable for the standard CTU size of 64x64 pixels, resulting in 2% and 1.3% gain over HOMC without HODS for the Affine and the Zoom&Rotation Motion Model, respectively. The maximum gain reached is 6.4% for *Station* and the affine HODS. The affine HODS seems to be more efficient altogether. Its predominance over the HODS of the Zoom&Rotation Motion Model can easily be explained by the higher complexity. Covering not only translation, dilation and rotation, like the Zoom&Rotation Motion Model, but additionally providing non-homogeneous scaling and shearing, it entails two more higher order motion components s_y and k that can profit from HODS. Collectively, the results confirm that HODS is a sensible addition to the HOMC system.

V. CONCLUSION AND OUTLOOK

In this paper a Higher Order Distance Scaling (HODS) of motion components derived from higher order motion parameters has been proposed. Whenever the distances to the reference pictures of the current block and its motion parameter predictor are not identical, the rotation angle, the scaling factors and the shearing factor of a higher order motion transformation matrix are distance-scaled separately and individually. Integrating the corresponding HODS algorithm into the previously introduced HOMC in KTA acquires an averaged rate reduction of around 18% over KTA and, depending on the motion model, 2% or 1.3% over prior HOMC. The HODS improves the HOMC performance for increased CTU sizes as well.

One aspect of the HODS still to be researched is an optimal multiplication order of the matrices representing individual forms of higher order motion as well as the decomposition of their product. Also, further steps of improvement of the HOMC system may include a frame- or even block-wise interchangeability of higher order motion models, a representation of higher order motion through a pixel-wise translational motion vector field and a more efficient signaling of higher order motion usage.

REFERENCES

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

[2] H. Huang, J. 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. Sullivan and R. Baker, “Motion Compensation for Video Compression using Control Grid Interpolation,” in *Proc. of ICASSP ’91*, 1991, pp. 2713–2716.

[4] H. Lakshman, H. Schwarz, and T. Wiegand, “Adaptive motion model selection using a cubic spline based estimation framework,” in *Proc. of ICIP ’10*. IEEE, 2010, pp. 805–808.

[5] M. Narroschke and R. Swoboda, “Extending HEVC by an affine motion model,” in *Proc. of PCS ’13*, 2013, pp. 321–324.

[6] C. Heithausen and J. H. Vorwerk, “Motion compensation with higher order motion models for HEVC,” in *Proc. of ICASSP ’15*. IEEE, 2015.

[7] C. Heithausen, M. Blaeser, M. Wien, and J.-R. Ohm, “Improved Higher Order Motion Compensation in HEVC with Block-to-Block Translational Shift Compensation,” in *Proc. of ICIP ’16*, Phoenix, Arizona, USA, 2016.

[8] J. C. T. on Video Coding, *H.265 : High Efficiency Video Coding*, http://www.itu.int/rec/T-REC-H.265-201504-I/en, Std.

[9] M. Wien, *High Efficiency Video Coding – Coding Tools and Specification*. Berlin, Heidelberg: Springer, 2014.

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

Sequence	CTU 64		CTU 128		CTU 256		CTU 512	
	Aff.	Z&R	Aff.	Z&R	Aff.	Z&R	Aff.	Z&R
Spincal.	31.2(**3.9**)	35.1(**3.2**)	32.3(**2.1**)	33.9(**1.9**)	28.7(**1.5**)	30.2(**1.4**)	26.5(**0.8**)	28.0(**-0.0**)
SlideSh.	19.2(**0.5**)	16.0(**-0.0**)	18.4(**-0.1**)	15.3(**-0.0**)	17.5(**0.0**)	13.8(**-0.8**)	16.0(**-0.3**)	13.7(**-0.0**)
Cactus	9.1(**0.4**)	9.6(**0.3**)	8.3(**0.1**)	8.6(**0.2**)	7.6(**0.1**)	8.0(**0.2**)	7.4(**0.1**)	7.9(**0.1**)
Tempete	9.3(**0.9**)	11.0(**0.6**)	8.2(**0.2**)	9.6(**-0.2**)	8.3(**0.3**)	9.6(**-0.2**)	8.2(**0.2**)	9.6(**-0.2**)
BQSq.	5.3(**0.0**)	6.6(**0.3**)	5.5(**0.5**)	6.3(**-0.2**)	5.3(**-0.4**)	6.3(**-0.2**)	5.5(**0.5**)	6.3(**-0.2**)
BlueSky	10.8(**1.2**)	12.4(**0.6**)	11.7(**0.5**)	12.2(**0.4**)	10.3(**0.4**)	10.4(**0.3**)	9.3(**0.2**)	9.6(**0.1**)
Station	37.1(**6.1**)	41.7(**2.7**)	40.7(**1.6**)	42.5(**0.9**)	37.2(**0.7**)	39.0(**0.7**)	33.0(**0.7**)	34.8(**0.4**)
Jets	16.3(**0.6**)	21.4(**0.6**)	22.9(**0.7**)	24.8(**1.1**)	21.7(**1.1**)	22.4(**0.4**)	19.2(**0.7**)	20.2(**0.5**)
TractorPart	29.2(**3.9**)	33.3(**2.3**)	32.7(**1.2**)	33.5(**0.5**)	29.4(**0.5**)	30.7(**0.3**)	27.0(**0.3**)	28.7(**0.5**)
ShieldsPart	21.4(**3.1**)	26.5(**1.3**)	26.6(**1.1**)	28.5(**0.5**)	24.4(**0.7**)	25.4(**0.3**)	22.1(**0.8**)	13.7(**0.6**)
average	18.9(**2.1**)	21.4(**1.2**)	20.7(**0.8**)	21.5(**0.5**)	19.0(**0.5**)	19.6(**0.2**)	17.4(**0.4**)	17.3(**0.2**)

TABLE II

RATE REDUCTION [%] OF HOMC+HODS, COMPARED AGAINST KTA (**COMPARED AGAINST HOMC WITHOUT HODS**).