

# MOTION COMPENSATION WITH HIGHER ORDER MOTION MODELS FOR HEVC

*Cordula Heithausen and Jan Hendrik Vorwerk*

Institute of Communications Engineering, RWTH Aachen University, 52056 Aachen, Germany

## ABSTRACT

In recent video compression standards only translational motion is accurately compensated, constraining any higher order motion to be approximated by being split into smaller translational units. The objective of this paper is to improve the coding efficiency for video sequences containing complex motion. Various higher order motion models are considered and evaluated to this end. Motivated by the higher maximum block size introduced by the High Efficiency Video Coding Standard (HEVC) and a suitable combination of motion parameter estimation, interpolation and coding, a higher order motion compensation framework for the HEVC is proposed. Through this an average data reduction of 2.9% as well as an increase in average block size is achieved.

*Index Terms*— higher order motion model, motion model evaluation, higher order motion compensation, HEVC motion prediction, gradient-based parameter estimation

## 1. INTRODUCTION

Current video coding standards almost exclusively employ the translational motion model, due to its computational simplicity and the established block-matching approach utilized for its displacement vector estimation. Although some standards allow the use of higher order motion compensation, it has not found widespread application. Several approaches using continuous or partial triangular grids have been proposed in [1, 2, 3]. Lakshman, Schwarz, and Wiegand [4] propose an estimation framework based on cubic splines. But for a block based prediction, the data rate caused by additional motion parameters tends to outgrow the rate saved by the prediction error reduction. Narroschke [5] 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. His work provides the underlying idea for this paper. Here, the gradient-based parameter estimation approach he uses for computing an affine transformation is extended to further higher order motion models. These include a reduced motion model with just four parameters, achieving a higher rate reduction than the affine motion model it is derived from. Furthermore, a high precision interpolation and an expanded quantization scheme are employed. Merging these aspects into a broader realization of an existing idea, this work describes the implementation of a higher order motion compensation framework for exchangeable motion models in the HEVC reference model.

In Section 2 several higher order motion models are shortly discussed, a gradient-based motion parameter estimation approach is explained and adapted to these. Section 3

outlines the integration of the higher order motion compensation approach into the HEVC framework. The evaluation of the motion models as well as the test results of the proposed HEVC extension are presented in Section 4. Section 5 concludes this paper.

## 2. HIGHER ORDER MOTION REPRESENTATION

The different kinds of motion occurring in video sequences may range from simple translation over rotation, scaling and sheering to even more complex types including motion of non-rigid objects or motion in transparent image areas. Higher order motion representation demands both a suitable higher order motion model as well as an algorithm to estimate the corresponding motion parameters.

### 2.1. Higher Order Motion Models

The common translational motion model with its two degrees of freedom can merely represent translation but has to approximate more complex motion. In order to have more than two degrees of freedom available, an alternative motion model is required. Suitable higher order motion models again vary in their motion description ability and accuracy, interrelated with their mathematic complexity. The more degrees of freedom such a model covers, the more motion parameters it will add to the two translational ones, resulting in additional coding cost.

Table 1 lists four higher order motion models alongside the translational motion model, ordered by increasing amount of required motion parameters  $P$ . The **affine motion model** can describe full motion of planar surfaces under orthographic projection and approximates the perspective camera model for high camera distances [6]. Its reduced version, the **zoom&rotation motion model**, only covering translation, uniform dilation (zoom) and rotation, is derived from the affine motion model by defining  $a_2 = a_3$  and  $a_4 = -a_5$ . The **bilinear motion model** adds perspective motion to the affine motion model and also allows more complex deformations. The **quadratic motion model** enables exact motion description of parabolic surfaces for parallel projection [6], provides central projection for high camera distances and can model some deformation of non-rigid objects.

### 2.2. Motion Parameter Estimation

The block-matching approach proved itself effective for the translational motion estimation, yet it presents no efficient

approach for higher order motion estimation, as search algorithms tend to be too computationally expensive when more than two motion parameters are sought [7]. Therefore, the proposed higher order motion compensation system employs an alternative estimation method based on temporal and spacial local gradients,  $g_t$  and  $\mathbf{g}_s = (g_x, g_y)^T$ , of the image signal. By means of this method, more explicitly presented by Narroschke [5], the sought motion parameters  $\mathbf{a}$  are calculated setting up an equation system (1) for  $N$  available pixels  $\mathbf{p}_n$  inside a block.

$$\underbrace{\begin{pmatrix} g_t(\mathbf{p}_0) \\ g_t(\mathbf{p}_1) \\ \vdots \\ g_t(\mathbf{p}_N) \end{pmatrix}}_{\mathbf{G}_t} = \underbrace{\begin{pmatrix} \mathbf{h}^T(\mathbf{p}_0) \\ \mathbf{h}^T(\mathbf{p}_1) \\ \vdots \\ \mathbf{h}^T(\mathbf{p}_N) \end{pmatrix}}_{\mathbf{H}} \cdot \underbrace{\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{pmatrix}}_{\Delta \mathbf{a}} \quad (1)$$

Applying the least-squares method, this yields

$$\Delta \mathbf{a} = \mathbf{H}^+ \cdot \mathbf{G}_t \quad (2)$$

with  $\mathbf{G}_t$  representing a matrix of gradients in time and  $\mathbf{H}^+$  being the pseudo-inverse of an  $N$ -dimensional linear system  $\mathbf{H}$  derived from the transformation equations of the respective motion model. For the individual higher order motion models, the following vectors  $\mathbf{h}^T(\mathbf{p}_n)$  are obtained:

$$\mathbf{h}_{\text{aff}}^T = (g_x, g_y, g_x x_k, g_y y_k, g_x y_k, g_y x_k) \quad (3)$$

$$\mathbf{h}_{z\&r}^T = (g_x, g_y, g_x x_k + g_y y_k, g_x y_k - g_y x_k) \quad (4)$$

$$\mathbf{h}_{\text{bil}}^T = (g_x, g_y, g_x x_k, g_y y_k, g_x y_k, g_y x_k, g_x x_k y_k, g_y x_k y_k) \quad (5)$$

$$\mathbf{h}_{\text{quad}}^T = (g_x, g_y, g_x x_k, g_y y_k, g_x y_k, g_y x_k, g_x x_k y_k, g_y x_k y_k, g_x x_k^2, g_y y_k^2, g_x y_k^2, g_y x_k^2) \quad (6)$$

Since the approach assumes a linear signal, extending this procedure to non-linear signals requires an iterative estimation approach [5] with  $\Delta \mathbf{a}$  resulting from the summation of partial shifts  $\Delta \mathbf{a}^{(i)}$ , for which approximately four iteration steps were found to be sufficient.

### 3. INTEGRATION OF HIGHER ORDER MOTION COMPENSATION IN HEVC

The higher order motion models and the corresponding gradient-based motion parameter estimation method are integrated into the HEVC Test Model, specifically version HM-12.1 of the reference software [8], in order to allow an equitable comparison between conventional (translational) motion prediction and conventional motion prediction extended by the option of a higher order motion representation.

#### 3.1. Implementation in the HM Reference Software

The proposed higher order motion compensation is not designed to replace the motion prediction system currently employed in HEVC but rather to offer an additional option for the encoder to choose from. Subsequent to each block-matching execution, the gradient-based higher order motion estimation approach described in Section 2.2 is applied in order to examine if the rate-distortion cost can hereby be reduced. While the conventional block-matching implementation remains unaltered, its result is sensibly utilized as an initial guess for the gradient-based higher order motion parameter estimation. Here, the estimation is performed on the luma component, as a first step, and the corresponding motion compensation is applied to the chroma components as well.

The procedure for each prediction block can be summarized in four steps: 1. Block-Matching, 2. Gradient-based parameter estimation, 3. Quantization of motion parameters, 4. Comparison of both motion models (translational and higher order) based on rate and distortion.

#### 3.2. Parameter Quantization

Handling the motion parameters as floating point values during the iterative part of the estimation maximizes their precision. Not until their conclusive comparison to the translational block-matching results, quantization is required. The chosen quantization scheme is the simple uniform quantization already utilized in HEVC. However, the parameters are quantized in separate groups according to their function within the respective motion model, e.g. translation, scaling or rotation. While the quarter-pixel quantization scheme of HEVC remains unaltered for the first group covering the two

Motion model	$P$	Camera model	Object surface	Possible motion	Transformation rule
Translational	2	any	flat	translation	$x_{k-1} = a_0 + x_k, y_{k-1} = a_1 + y_k$
Zoom & rotation	4	parallel	flat	translation, zoom, rotation	$x_{k-1} = a_0 + a_2 x_k + a_3 y_k,$ $y_{k-1} = a_1 + a_2' y_k - a_3' x_k$
Affine	6	parallel	flat	any	$x_{k-1} = a_0 + a_2 x_k + a_4 y_k, y_{k-1} = a_1 + a_3 y_k + a_5 x_k$ $x_{k-1} = a_0 + a_2 x_k + a_4 y_k + a_6 x_k y_k$
Bilinear	8	perspective	flat	any	$y_{k-1} = a_1 + a_3 y_k + a_5 x_k + a_7 x_k y_k$
Quadratic	12	perspective	parabolic	any	$x_{k-1} = a_0 + a_2 x_k + a_4 y_k + a_6 x_k y_k + a_8 x_k^2 + a_{10} y_k^2$ $y_{k-1} = a_1 + a_3 y_k + a_5 x_k + a_7 x_k y_k + a_9 y_k^2 + a_{11} x_k^2$

**Table 1:** Motion models ordered by increasing complexity (amount of parameters  $P$ ).

translational motion parameters, different quantization factors are applied for higher order motion parameters. When choosing these additional quantization factors, the premise is to obtain a quantization induced quality degradation less or equal to that introduced by the quantization of the translational motion parameters.

### 3.3. Interpolation filters

As the quarter-pixel interpolation used for motion compensation in HEVC cannot provide a precision high enough to ensure the full benefit of the higher order motion models, an interpolation method which also uses separable eight-tap integer filters but achieves a precision of up to a 1/16th pixel is employed. Its interpolation filters are already known from the scalable extension of HEVC [9], only in this reimplementation of the filtering no upscaling of the entire block or frame is performed but the interpolation filter values are used for obtaining the pixels of the motion compensated block.

### 3.4. Signaling of Additional Motion Parameters

The signaling of the higher order motion parameters is mostly inherited from the translational ones. An additional flag is transmitted, indicating if the original translational or the higher order motion model is applied. In case the translational model is chosen, no additional motion parameters will be transmitted. The flag uses a different context depending on whether or not the PU above or to the left is using the higher order motion model as well. Provided a neighbor is not available, the translational model is assumed. The transmission of the higher order motion parameters is identical to that of the translational parameters using separate context models.

## 4. EVALUATION AND RESULTS

This section is divided into two parts, the first one evaluating the higher order motion models and the second one presenting and analysing the experimental results of the proposed higher order motion compensation system.

### 4.1. Motion Model Evaluation

Preliminary to the implementation of the higher order motion compensation in the HEVC Test Model, the above mentioned motion models were considered regarding their ability to improve the prediction, i.e. reduce distortion, in relation to the amount of additional parameters required, as compared to the commonly used translational motion model.

The motion model evaluation was implemented outside of HEVC in a reduced test-environment. The evaluation process was as follows: Each frame was split into blocks of a consistent size of 64x64 pixels. Forward motion compensation was performed by a full search block-matching with full pixel precision over a given range, a subsequent block-matching refinement with quarter-pixel precision and finally by means of the gradient-based method explained in Section 2.2. The evaluation process concluded with a calculation of the PSNR per block. The test was run on short extracts of the *Cactus* as well as the *Foreman* sequence.

Motion model	PSNR [dB] ( $\Delta$ PSNR/ $\Delta$ P)			
	Cactus		Foreman	
Translational	32.68		32.10	
Zoom&rotation	33.32	(0.32)	33.21	(0.56)
Affine	33.69	(0.25)	34.45	(0.59)
Bilinear	33.84	(0.19)	34.78	(0.45)
Quadratic	34.14	(0.15)	35.48	(0.34)

**Table 2:** Average PSNR and PSNR improvement over the translational motion model per additional parameter.

Table 2 shows the averaged PSNR values of the prediction error for the tested motion models. The highest improvements over the translational motion model per additional motion parameter were observed when applying the affine as well as the zoom&rotation motion model. The bilinear and quadratic motion models, however, could not provide enough additional PSNR enhancement to justify the coding cost caused by their high complexity and are therefore neglected hereafter.

### 4.2. Performance of Proposed HEVC Extension

The evaluation of the proposed higher order motion compensation scheme is conducted in *random access* and *low delay* mode from JCTVC CTC on a set of 15 sequences of different resolutions, containing different amounts of non-translational motion, listed in Table 3. The proposed higher order motion compensation extension is able to employ arbitrary higher order motion models. Here, the results for the affine and the zoom&rotation motion model are discussed as they were evaluated best. (Section 4.1).

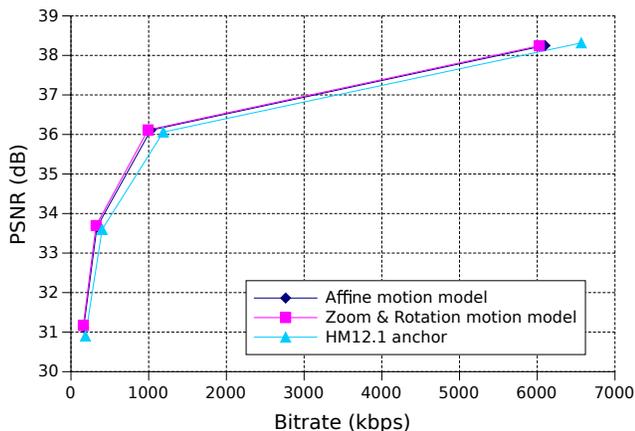
**Quantization factors:** For both motion models there are three groups of parameters to be quantized (Section 3.2), namely translation ( $a_0, a_1$ ), scaling ( $a_2, a_3$  for affine and  $a_2'$  for zoom&rotation) and rotation ( $a_4, a_5$  for affine and  $a_3'$  for zoom&rotation). Selected higher order motion parameter quantization factor sets  $q_i$  were tested on a subset of sequences, with  $(q_{i,\text{scaling}}, q_{i,\text{rotation}}) \in \{ (32, 64), (64, 128), (128, 256), (256, 256), (256, 512), (512, 512) \}$ . The quantization factors  $q_3 = (256, 256)^T$ , leading to the highest rate reduction, are chosen for the following evaluation of the two higher order motion models in the HEVC implementation.

**Coding efficiency:** The coding efficiency is measured at four operating points ( $QP = 22, 27, 32, 37$ ), calculating the average rate reduction in comparison to the HM12.1 anchor following the method described in [10]. Evaluation results can be found in Table 3. In *low delay* mode the data rate is reduced by an average of 2.5% using the affine motion model and 2.9% using the zoom&rotation model. When using the *random access* mode, the data rate reductions are decreased to 0.9% for both motion models as the assumption of small motion needed by the gradient-based estimation does not hold over a high temporal distance. The most rate reduction is achieved on sequences that contain rotation like *SpinCalendar* (highest increase in coding efficiency of 18.8%) and *Cactus*, as rotation approximated by translational motion causes very small partitioning. But also sequences featuring camera zoom are improved. For few sequences containing only very small amounts of non-translational motion such as *City* the

Sequence		BD rate [%]			
		Random access		Low delay	
		affine	z&r	affine	z&r
non-transl.	Cactus <sup>1,8</sup>	-2.9	-3.4	-4.3	-5.0
	BQSquare <sup>4,7</sup>	0.2	0.0	-0.6	-1.2
	Spincalendar <sup>2,9</sup>	-4.8	-5.4	-15.5	-18.8
	Tempete <sup>6,10</sup>	-0.8	-1.2	-5.2	-7.1
	ChinaSpeed <sup>5,9</sup>	-0.4	-0.5	-0.5	-0.6
mixed	SlideShow <sup>2,10</sup>	-6.4	-4.5	-9.2	-6.1
	BigShips <sup>2,8</sup>	-0.6	-0.7	-2.9	-3.6
	Bl.-Bubbles <sup>4,8</sup>	0.2	0.3	-0.1	-0.3
	PartyScene <sup>3,8</sup>	0.2	0.2	0.1	0.0
	City <sup>2,9</sup>	0.9	0.8	0.2	0.3
translational	RaceHorses <sup>4,9</sup>	0.2	0.1	-0.3	-0.3
	BQMall <sup>3,7</sup>	0.4	0.4	0.0	-0.1
	SlideEditing <sup>2,9</sup>	0.0	-0.1	0.4	-0.2
	BQTerrace <sup>1,7</sup>	0.3	0.3	0.2	0.2
	BasketballDT <sup>3,8</sup>	0.3	0.4	0.0	0.0
average		-0.9	-0.9	-2.5	-2.9

**Table 3:** BD-rate of test set, arranged by predominant motion.

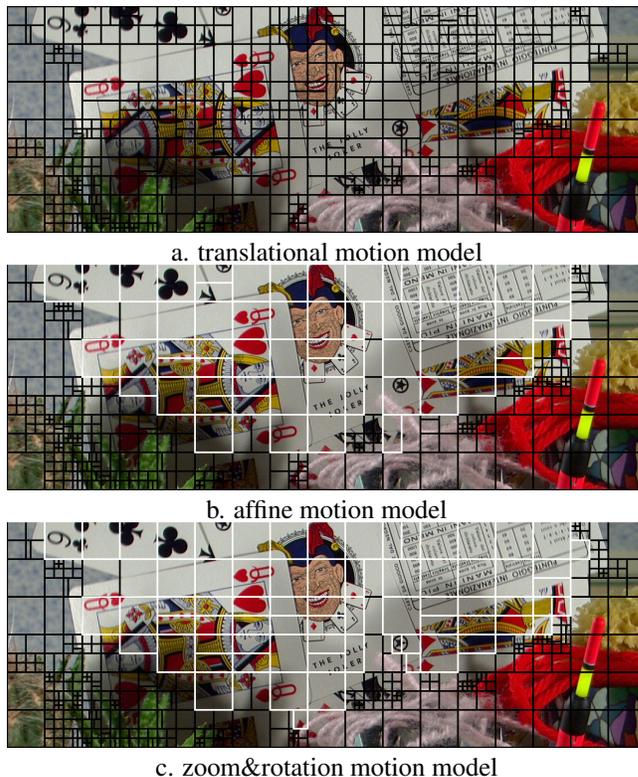
cost of the additional flag may result in an unfortunate data rate increase. Most of the rate reduction is achieved by the high quality (low QP) encodings, while the low quality encodings (high QP) are improved in PSNR (Fig.1). Furthermore, a slight improvement in video quality can be subjectively perceived in certain image details of the higher order motion prediction signal in comparison to the translational one. Rotating lines and edges, for instance, appear more smooth and continuous, e.g. the red horizontal lines in Spincalendar which may show a steplike structure when only motion predicted translationally. Altogether, the zoom&rotation model, though slightly more restricted representing complex motion, is more effective than the affine motion model, with less additional motion parameters to be encoded.



**Fig. 1:** BD-rate ( $QP = 22, 27, 32, 37$ ) of the *Spincalendar* sequence in *low delay* mode.

<sup>1</sup> HD1080 <sup>2</sup> HD720 <sup>3</sup> WVGA <sup>4</sup> WQVGA <sup>5</sup> XGA <sup>6</sup> CIF  
<sup>7</sup> 60fps <sup>8</sup> 50fps <sup>9</sup> 30fps <sup>10</sup> 20fps

**Block Partitioning:** To exemplify how the block partitioning of areas with higher order motion may benefit from the proposed higher order motion compensation approach, an image detail from the *Cactus* sequence is visualized in Figure 2. The clockwise rotating card-deck is split into many small motion segments by the PU partitioning of the HM reference implementation (Fig.2a). This effect is noticeably reduced when the same frame is predicted by the affine or the zoom&rotation motion model (Fig.2b-c). The blocks are bigger and often reach the maximum block size of 64x64 pixels.



**Fig. 2:** PU partitioning of *Cactus*.  $QP=32$ , *low delay*. White blocks use higher order motion models.

## 5. CONCLUSION

This paper proposes a higher order motion compensation extension to HEVC, offering the encoder a block-based choice between the translational and a higher order motion model. An existing gradient-based motion estimation algorithm was adapted to several higher order motion models and implemented in the HM reference software. More accurate interpolation filters are employed, a quantization scheme was developed and evaluated. An average data rate reduction of 2.5% using the affine motion model and 2.9% using the zoom&rotation motion model in *low delay* mode was achieved.

The proposed higher order motion extension to HEVC could be improved by deriving further motion models better adjusted to motion in video sequences, improving the signaling or increasing the maximum block size [5].

## 6. 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] M. Narroschke and R. Swoboda, "Extending hevc by an affine motion model," in *PCS*, 2013, pp. 321–324.
- [6] N. Diehl, "Object-oriented motion estimation and segmentation in image sequences," *Signal Processing: Image Communication*, vol. 3, no. 1, pp. 23–56, 1991.
- [7] Jens-Rainer Ohm, *Multimedia Communication Technology. Representation, Transmission and Identification of Multimedia Signals*, Springer, Heidelberg/Berlin, 2004.
- [8] Joint Collaborative Team on Video Coding, "HM reference software 12.1," [https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVCSoftware](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware).
- [9] Joint Collaborative Team on Video Coding, "SHM reference software 5.0," [https://hevc.hhi.fraunhofer.de/svn/svn\\_SHVCSoftware](https://hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware).
- [10] S. Pateux and J. Jung, "An excel add-in for computing bjotegaard metric and its evolution," VCEG Doc. VCEG-AE07, January 2007.