Motion Blur Compensation in HEVC Using Fixed-length Adaptive Filter

Yiqun Liu, Wei Wu and Jörn Ostermann
Institut für Informationsverarbeitung
Gottfried Wilhelm Leibniz Universität Hannover, Germany
{liuyiqun, office}@tnt.uni-hannover.de, wuwei@iie.ac.cn

***Abstract*—Motion compensation is one of the most important elements in modern hybrid video coders. It utilizes temporal information to predict the current block and reduces thereby the redundancy of a video. The accuracy of prediction depends on the similarity of the content between the reference block and the current block. With the change of velocity of the camera or certain objects in a scene, which is typically expected in action and sports movies, motion blur varies from frame to frame leading to a reduced prediction accuracy. We employ fixed-length filters to compensate varying motion blur in hybrid video coding. While former approaches needed additional signaling for blurring filters, our filter is derived only based on the motion vector. We implemented our approach in the High Efficiency Video Coding (HEVC) reference software HM 13.0. Compared to the reference we gain 2.15% in terms of BD-Rate in average for JCT-VC test sequences and 4.43% for self-recorded sequences containing lots of varying motion blur.**

Keywords: Motion blur compensation, HEVC, Video coding

I. Introduction

Motion compensation together with intra-frame prediction, quantization, transform and entropy coding has been the cornerstone of hybrid video coding system. It is used in the High Efficiency Video Coding (HEVC) [1] standard as well as in predecessors like MPEG-2 [2] and AVC [3]. Motion compensation uses already coded previous or future frames to predict the content of the current coding unit (CU). Instead of the original block content, only a displacement vector called Motion Vector (MV) and the corresponding prediction error are used for coding and quantization, in order to generate the bit stream. The data rate can be vastly reduced by this prediction technique if the similarity between the reference frame and the current frame is high.

The general motion compensation method from video coding standards works well with stationary objects or moving objects of constant velocity. However, its accuracy is limited in case of varying motion blur. Motion blur occurs in the direction of object motion if an object moves during the exposure time. An accelerated object looks blurrier in the current frame than in the previous reference frame while one in deacceleration appears sharper. The changing extent of blur between successive frames generally enlarges the prediction error. Thus it results in reduced compression efficiency and an increased data rate for the residual of inter-predicted CUs.

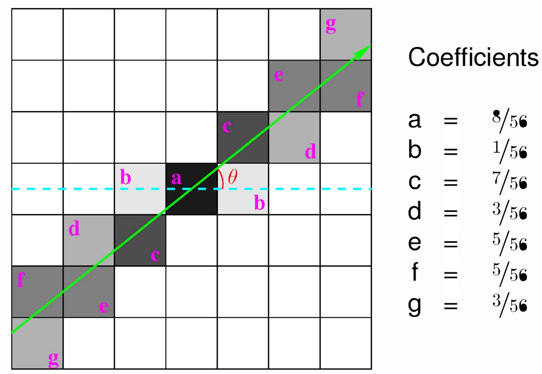


Fig. 1: Discrete version of linear motion blur ($\theta = \arctan \frac{4}{5}$)

Several approaches were proposed to reduce the prediction error by filtering the reference frame for varying motion blur compensation. Some of them use either pre-defined filter [4][5] or adaptive filter [6] for single layer coding. All three approaches need additional signaling for the choice of filter or for the filter coefficients. Another proposal [7] solves the problems of additional signaling by employing the motion information from the base layer of scalable video coding. However, in this method the compensation for blur is severely restricted to only 4 directions, which might lead to insufficient improvement for prediction.

In this paper we propose a blur compensation algorithm for single layer coding that generates the blurring filter *in arbitrary direction* based on the direction of the transmitted MV. At the encoder, a *validation check* ensures that the filter associated with the transmitted MV is useful. Thus no additional signaling for the filter coefficients is necessary, whereas the usage of the filter is signaled by the encoder for each CU.

The remaining paper is organized as follows: Section II analyzes the filter used for blur compensation, and explains our motion blur compensation method in detail. Experimental results are given in Section III before Section IV concludes the paper.

II. Proposed Reference Frame Filtering

Due to the variability of the extent of motion blur which is caused by the change of the velocity of the camera or an object between frames, we suggest to filter the reference frame in

TABLE I: Discrete form of proposed 3×3 blurring filter for different ranges of θ

θ	$\left[0, \arcsin\frac{1}{3}\right)$	$\left[\arcsin\frac{1}{3}, \arccos\frac{1}{3}\right)$	$\left[\arccos\frac{1}{3}, \pi - \arccos\frac{1}{3}\right)$	$\left[\pi - \arccos\frac{1}{3}, \pi - \arcsin\frac{1}{3}\right)$	$\left[\pi - \arcsin\frac{1}{3}, \pi\right)$
a	0			$\frac{1}{2} - \frac{1}{6}\frac{1}{\min(\sin\theta, -\cos\theta)}$	0
b	0	$\max(0, \frac{1}{6}\left(\frac{1}{\cos\theta} - \frac{1}{\sin\theta}\right))$	$\frac{1}{2} - \frac{1}{6}\frac{1}{\sin\theta}$	$\max(0, -\frac{1}{6}\left(\frac{1}{\cos\theta} + \frac{1}{\sin\theta}\right))$	0
c	0	$\frac{1}{2} - \frac{1}{6}\frac{1}{\min(\sin\theta, \cos\theta)}$	0		
d	$\frac{1}{2} - \frac{1}{6}\frac{1}{\cos\theta}$	$\max(0, \frac{1}{6}\left(\frac{1}{\sin\theta} - \frac{1}{\cos\theta}\right))$	0	$\max(0, \frac{1}{6}\left(\frac{1}{\sin\theta} + \frac{1}{\cos\theta}\right))$	$\frac{1}{2} + \frac{1}{6}\frac{1}{\cos\theta}$
e	$\frac{1}{3}\frac{1}{\max(\lvert\cos\theta\rvert, \sin\theta)}$				

order to add blur and hence to increase the similarity between the reference CU and the current CU.

A commonly used linear uniform motion blur point spread function (PSF) in continuous time domain is described with a line segment L and an angle θ with respect to the horizontal axis [8], as given in (1).

$$h(x, y) = \begin{cases} \frac{1}{L}, & \sqrt{x^2 + y^2} \leq \frac{L}{2}, \frac{y}{x} = \tan\theta \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

L is proportional to motion speed and exposure duration, θ indicates the motion direction and (x,y) is the location in Cartesian coordinate system.

The discrete version of Eq. (1) is acquired by considering a bright spot traversing across the sensors covered by the line segment during the exposure time with constant velocity [9]. Each coefficient of the blur kernel is proportional to the time spent on each sensor element. With the assumption of a constant motion, the filter coefficients are given by the normalized length of the intersection of the line segment with each pixel in the grid, as illustrated by Fig. 1.

In case of varying motion blur, a filtered reference may improve the coding performance. We assume that motion blur as well as the change of motion blur can be described by Eq. (1). Motion can be considered constant for the exposure time of a single frame. Since the time interval between two nearby frames is only 0.02 seconds for a 50 fps video sequence, we suggest that the change of motion blur extent is small in most cases. Therefore we employ a fixed extent of 3 pels to describe the phenomenon of variation in blurring, i.e., $L = 3$. Hence the two dimensional blurring filter for the luminance component has a dimension of 3×3.

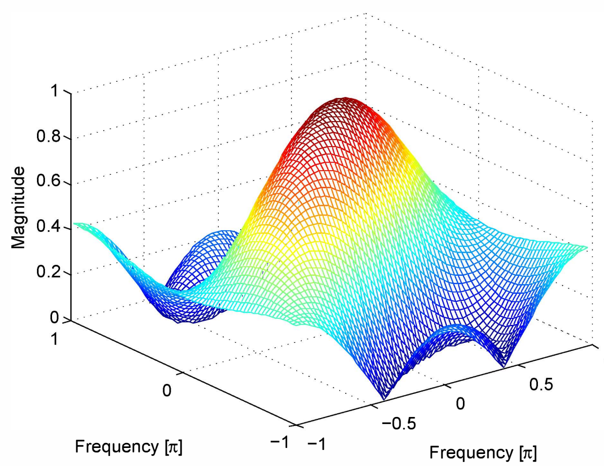
The other degree of freedom for a blurring filter is the angle θ. It is derived from the motion vector used in HEVC. Hence a standard HEVC motion search is executed before the blurring filter is established. The blurred reference frame is generated by using the directions of the MVs.

The general discrete version of our 3×3 filter has 5 different coefficients (a – e) due to symmetry and is calculated according to an arbitrary angle θ using Table I. The suggested filter is a low-pass filter. As an example, filter coefficients for $\theta = 15°$ and $\theta = 60°$ are listed in Table II and their frequency responses are shown in Fig. 2.

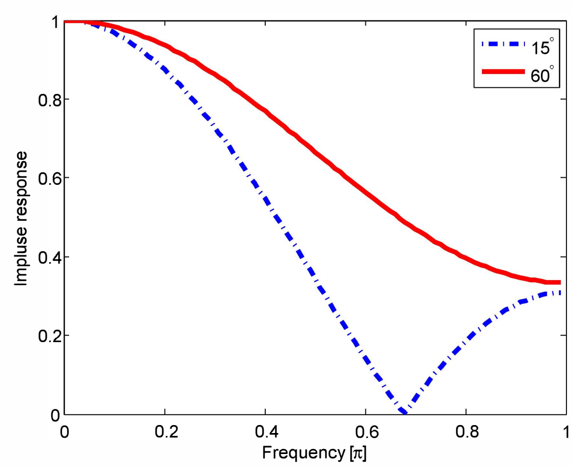
We will only blur the luminance component of the reference by using such a 3×3 filter if a test sequence has the 4:2:0 format. Chrominance pixels lie at every other pixel position of

TABLE II: Filter coefficients

3×3 Filter	θ = 15°	θ = 60°
$\begin{bmatrix} a & b & c \\ d & e & d \\ c & b & a \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0.327 & 0.345 & 0.327 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0.141 & 0.167 \\ 0 & 0.381 & 0 \\ 0.167 & 0.141 & 0 \end{bmatrix}$



(a) 2D frequency response for ($\theta = 60°$)



(b) Frequency responses alongside X-axis ($\theta = 15°, 60°$)

Fig. 2: Frequency responses of blurring filter

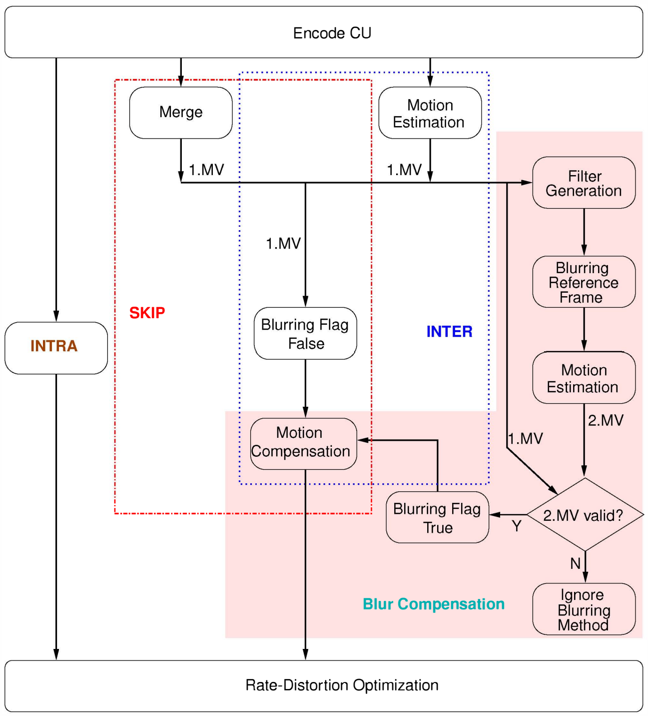


Fig. 3: Flow chart of motion blur compensation

TABLE III: BD-Rate (Y) with different numbers of filters

Sequence	Number of filters 6	9	18	30
JCT-VC	-2.19%	-2.15%	-2.10%	-2.00%
Self-recorded	-4.36%	-4.44%	-4.35%	-4.28%
Average	-2.98%	-2.98%	-2.91%	-2.83%

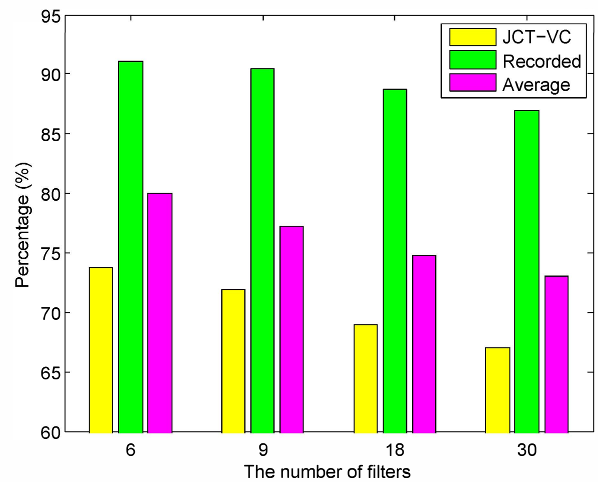


Fig. 4: Ratio of valid 2. MV

the luminance component [10]. With blur extent of only 3 pels, three neighboring luminance pixels but only 1 chrominance pixel have to be considered. A neighbored chrominance pixel is out of the influence range of blurring, hence no filter is required. In case of a sequence of 4:4:4 chroma subsampling, the same filter for luminance component will be applied for chrominance components.

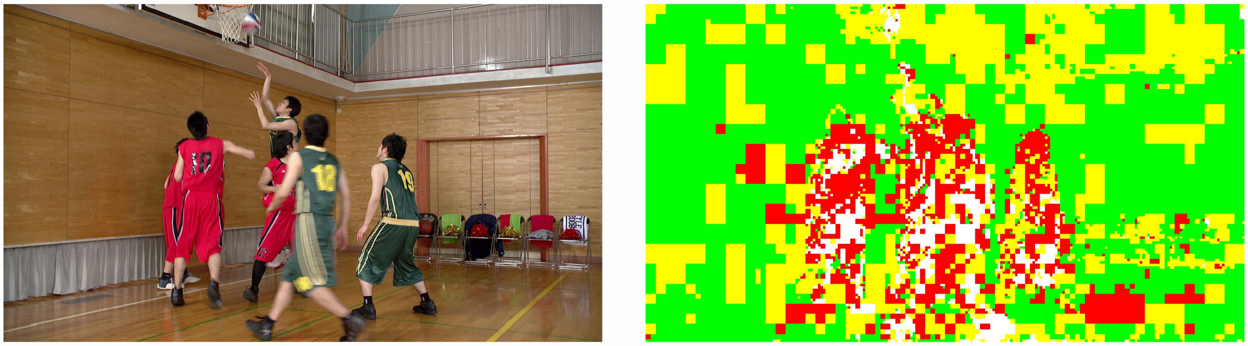
Besides the standard coding methods like Intra-frame and Inter-frame prediction (incl. Skip), we add a motion blur compensation mode to HEVC. This motion blur compensation can be considered as a complement of Inter mode (incl. Skip) of HEVC during the coding of a certain CU, as it is illustrated in Fig. 3. We create temporary reference frames by filtering the reconstructed frame with a filter of angle θ derived from the original motion vector (1. MV) from the motion estimation (ME) of HEVC and use these reference frames as new reference frames for motion blur compensation. Assuming the range of θ is uniformly quantized into N_θ intervals, θ_i is the representative angle in the middle of the i-th interval. θ_i is determined by quantizing $\theta = \arctan \frac{y}{x}$, where y and x are vertical and horizontal value of 1. MV. For each frame in the reference picture list there will be a corresponding frame for motion blur compensation, since the results of ME based on reconstructed frames and blurred frames do not always share the same reference index.

One vital part in the encoding process is the validation check of the estimated motion vector (2. MV), after a second motion estimation is made based on the blurred reference. In case the quantization of the angle of the 2. MV results in the same θ_i as the quantization of the 1. MV, the 2. MV is valid.

The validation check ensures that the object moves in the direction where variation of blur exists. Furthermore, no additional signaling for the filter is needed as long as 2. MV based on the blurred reference has the same representative direction as the 1. MV, since the filter is generated with fixed length and the only variable is θ.

The angle θ depends only on the number of filters N_θ, which can be arbitrarily defined according to the requirement on the accuracy of directions , e.g. each filter will cover an angle of $4°$ if we apply 45 filters. Hence the variation of blur in arbitrary direction can be covered by this motion blur compensation method. Theoretically the prediction of varying blur will be more accurate with the increase of the number of filters. The validation rate is the ratio of valid 2. MVs and the number of MVs. The number of filters has to be identical for encoder and decoder.

A "Blurring Flag" is set within the encoding for each CU which has been coded with Inter mode (incl. Skip) for all partition sizes (2N×2N, 2N×N, 2N×Nu, 2N×Nd…). It is coded directly after the coding of prediction mode in the bitstream, if CU decides to use inter-prediction. The decoder is able to determine whether the reference frame is blurred during reconstruction of a CU and to replicate the blurring of reference frame using the angle θ derived from the encoded MV. The blurring flag is considered within the Rate-Distortion (RD) optimization.



(a) Decoded Frame (b) Predicion Mode

Fig. 5: Coding mode distribution of Basketball Drive. Red (darkest): Blur, Green: Skip , Yellow: Inter and White: Intra.

III. Experimental Results

The implementation of our proposed algorithm is based on the HEVC reference software HM 13.0 [11].

We have applied the JCT-VC common test conditions [12] with the configuration of Low Delay P (LD-P) and Random Access (RA) for the evaluation. Our test set contains only sequences with 4:2:0 chroma subsampling, which include JCT-VC sequences (*Basketball Drive*, *BQ Terrace*, *Cactus*, *Kimono*, *Park Scene*, *People On Street*, *Traffic*) as well as self-recorded sequences (1280 x 720). The latter include *Playground* [13] (filmed with a hand-held camera) and *Bike* [7] (filmed with a camera attached to the helmet of a cyclist). We used a fixed focal length for all self-recorded sequences. Thus no blur contained in the sequences is caused by focus change. Because of the shaking of the camera during the recording, we assume to have a significant amount of varying motion blur in these sequences.

The influence of the number of filters on the coding efficiency based on LD-P configuration is shown in Table III in terms of Bjøntegaard delta (BD)-Rate. Negative numbers mean gain compared to the anchor while positive numbers represent a loss. As an anchor we used HM 13.0 without motion blur compensation.

Theoretically the prediction of a CU becomes more accurate with an increase of the number of filter, since the generated filters are more precise in directions. Meanwhile the range of valid angles for the 2. MV decreases from 30° for 6 filters to 6° for 30 filters, which results in a drop of the validation rate as shown in Fig. 4.

It is reasonable to find out that gain from blur compensation drops for a large number of filters e.g. 30, as the motion blur compensation is ignored more often. A compromise is necessary between accuracy and validation rate. We suggest 9 filters for further simulations, for the results based on 9 filters in Table III are at least as good as those with the other number of filters if not better, especially for our own self-recorded video sequences which have in general more varying blurring.

The usage of motion blur compensation is illustrated in Fig. 5 showing frame No. 481 of Basketball Drive as an example. In Fig. 5b CUs coded with the proposed blurring method are shown in red while yellow, green and white represents inter, skip and intra modes, respectively. Looking at the decoded frame (Fig. 5a), it seems that motion blur compensation is chosen mostly for the areas where the obvious varying motion and related blur should occur: on the bodies of the players.

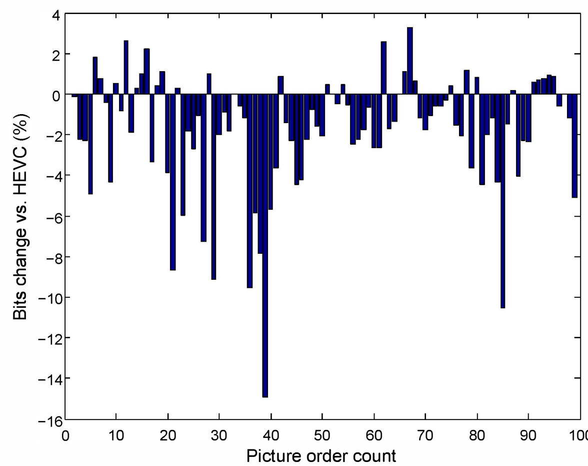


Fig. 6: Bit rate change per picture, *Playground*, RA, QP 24. Reference: HM 13.0

Fig. 6 shows the frame-wise change of bits for the entire *Playground* sequence. The majority of the frames hold negative numbers up to over 14% in the figure, meaning less bits are needed to encode the content of the frame. Some limited positive values related to our increase of peak signal-to-noise ratio (PSNR) can also be found. There is no change for I-frames. It is obvious that motion blur compensation reduces the number of bits for compression. Typically about 120 bits are used for coding of the "Blurring Flag" per frame in case of HDTV.

Our self-recorded sequences contain much more varying motion blur due to the camera motion as well as longer exposure time. Consequently they benefit more from our proposed motion compensation method, as shown in Table IV. The

TABLE IV:
BD-Rate with 9 Filters vs. HM 13.0

Sequence	Low Delay P			Time
	Y	U	V	
Playground	-4.26%	-2.02%	-3.21%	303%
Bike 01	-5.71%	-2.96%	-1.96%	440%
Bike 02	-1.31%	-1.06%	-0.71%	393%
Bike 03	-6.43%	-3.95%	-2.48%	370%
Average (Recorded)	-4.43%	-2.50%	-2.09%	
Basketball Drive	-3.85%	-1.20%	-1.33%	365%
Kimono	-2.89%	1.11%	1.16%	303%
Cactus	-1.94%	-0.78%	-0.50%	306%
Park Scene	-1.16%	-0.02%	-0.86%	314%
BQ Terrace	-1.95%	-1.11%	-0.59%	367%
People On Street	-2.17%	-0.94%	-0.45%	317%
Traffic	-1.07%	-0.41%	-0.11%	406%
Average (JCT-VC)	-2.15%	-0.48%	-0.38%	
Average (Overall)	-2.98%	-1.21%	-1.00%	353%

Sequence	Random Access			Time
	Y	U	V	
Playground	-2.98%	-1.68%	-2.03%	343%
Bike 01	-2.32%	-1.08%	-0.18%	354%
Bike 02	-0.68%	-0.35%	-0.59%	265%
Bike 03	-2.69%	-1.30%	-0.68%	203%
Basketball Drive	-0.98%	-1.10%	-1.16%	358%
Kimono	-0.30%	0.22%	0.10%	320%
Cactus	-0.27%	-0.09%	-0.33%	299%
Park Scene	-0.10%	-0.02%	-0.03%	343%
BQ Terrace	-0.26%	-0.06%	0.02%	337%
People On Street	-0.38%	-0.65%	-0.36%	285%
Traffic	0.05%	0.22%	0.18%	329%
Average (Overall)	-0.99%	-0.53%	-0.46%	312%

average gain compared to the HM 13.0 anchor for $\{Y, U, V\}$ based on these sequences reaches $\{4.43\%, 2.50\%, 2.09\%\}$ for LD-P configuration, majority of which offer over 4% gain for luminance.

Due to the fact that an additional motion estimation and the filtering for reference frames have to be made during the process of encoding, the computational complexity rises as the price for the gains. The motion blur compensation method increases execution time by 253% and 212% compared to HM-13.0 for LD-P and RA respectively, as shown in the last column of Table IV.

As expected, the proposed method works less good on JCT-VC sequences. Nevertheless, we get gain for all those sequences, which is averaged to $\{2.15\%, 0.48\%, 0.38\%\}$ for $\{Y, U, V\}$ respectively. Because of the rapid change in motion blur from the players of *Basketball Drive* and from the leaves on the background of *Kimono*, it is reasonable to observe higher luminance gains compared to the other sequences of this test set.

Overall, the BD-Rate gain of motion blur compensation is $\{2.98\%, 1.21\%, 1.00\%\}$ on average for LD-P and $\{0.99\%, 0.53\%, 0.46\%\}$ for RA. Results based on LD-P are better than that based on RA, since the temporal distance between the reference frame and the current frame from LD-P is shorter and the proposed filter is only 3×3, which is especially designed for small variations of motion blur.

IV. Conclusion

The performance of the general motion compensated prediction in HEVC is limited by varying motion blur. To improve the prediction accuracy in order to reduce the coding data rate we propose 3×3 blurring filters for reference frame filtering. The filter is generated based on the direction of the transmitted MV, which can be an arbitrary angle θ. Hence no extra signaling of filter coefficients is needed. Compared to HM 13.0, our proposed motion blur compensation can provide an average BD-Rate gain of 2.15% for JCT-VC sequences and of 4.43% for consumer recorded sequences.

References

[1] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, "High Efficiency Video Coding (HEVC) text spec. draft 10 (for FDIS&Consent)," in *JCT-VC Doc. JCTVC-L1003*, 12th Meeting: Geneva, Switzerland, Jan. 2013.

[2] ISO/IEC and ITU-T, *Recommendation ITU-T H.262 and ISO/IEC 13818-2 (MPEG-2 Part 2): Information technology - Generic coding of moving pictures and associated audio information: Video*, Mar. 1995.

[3] ISO/IEC and ITU-T, *Recommendation ITU-T H.264 and ISO/IEC 14496-10 (MPEG-4 Part 10): Advanced Video Coding (AVC)-3rd Ed.*, Geneva, Switzerland, Jul. 2004.

[4] M. Budagavi, "Video Compression Using Blur Compensation," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 2, Sept 2005, pp. II–882–5.

[5] J. Jang, H. Lee, T.-Y. Jung, S.-M. Hong, and J. Jeong, "Pre-filtering with Locally Adaptive Filter Set for Reference Frame in Video Compression," in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*, July 2011, pp. 1–6.

[6] Y. Vatis and J. Ostermann, "Adaptive Interpolation Filter for H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 2, pp. 179–192, 2009.

[7] T. Laude, H. Meuel, Y. Liu, and J. Ostermann, "Motion Blur Compensation in Scalable HEVC Hybrid Video Coding," in *Picture Coding Symposium (PCS), 2013*, Dec 2013, pp. 313–316.

[8] H. Lin and C. Chang, "Photo consistent motion blur modeling for realistic image synthesis," in *PSIVT 2006, LNCS 4319*, 2006, pp. 1273–1282.

[9] J. Oliveira, M. Figueiredo, and J. Bioucas-Dias, "Parametric Blur Estimation for Blind Restoration of Natural Images: Linear Motion and Out-of-Focus," *Image Processing, IEEE Transactions on*, vol. 23, no. 1, pp. 466–477, Jan 2014.

[10] M. Ghanbari, *Video Coding, an Introduction to Standard Codecs.* London, United Kingdom: The Institution of Electrical Engineers, 1999.

[11] K. McCann, B. Bross, W. Han, I. Kim, K. Sugimoto, and G. J. Sullivan, "High Efficiency Video Coding (HEVC) Test Model 13 (HM 13) Encoder Description," in *JCT-VC Doc. JCTVC-O1002*, 15th Meeting: Geneva, Switzerland, Oct. 2013.

[12] X. Li, J. Boyce, P. Onno, and Y. Ye, "L1009: Common Test Conditions and Software Reference Configurations for the Scalable Test Model. Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11. 12th Meeting, Geneva, CH, 14-23 Jan," 2013.

[13] K. Cordes, B. Scheuermann, B. Rosenhahn, and J. Ostermann, "Occlusion Handling for the Integration of Virtual Objects into Video," in *The 7th International Conference on Computer Vision Theory and Applications (VISAPP)*, 2012.