

A Review on Motion Estimation and Compensation for Versatile Video Coding Technology (VVC)

Young-Ju Choi[†], Byung-Gyu Kim^{**}

ABSTRACT

Video coding technologies are progressively becoming more efficient and complex. The Versatile Video Coding (VVC) is a new state-of-the-art video compression standard that is going to be a standard, as the next generation of High Efficiency Video Coding (HEVC) standard. To explore the future video coding technologies beyond the HEVC, numerous efficient methods have been adopted by the Joint Video Exploration Team (JVET). Since then, the next generation video coding standard named as VVC and its software model called VVC Test Model (VTM) have emerged. In this paper, several important coding features for motion estimation and motion compensation in the VVC standard is introduced and analyzed in terms of the performance. Improved coding tools introduced for ME and MC in VVC, can achieve much better and good balance between coding efficiency and coding complexity compared with the HEVC.

Key words: VVC, FVC, Video Coding, Motion Estimation, Motion Compensation

1. INTRODUCTION

Video coding technologies are progressively becoming more efficient and complex. With continuous development of display resolution and type along with enormous demand for high quality video contents, video coding also plays a key role in display and content industries. The Versatile Video Coding (VVC) [1] has been standardized by the Joint Video Exploration Team (JVET) of ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) after standardizing H.264/AVC [2] and H.265/HEVC [3] successfully. Obviously, HEVC is a reliable video compression standard. Nevertheless, more efficient video coding scheme is required for higher-resolution and the newest media services. To explore

the future video coding technologies beyond HEVC, numerous efficient methods have been adopted by JVET and put into the reference software model called Joint Exploration Model (JEM) [4],[5]. Since then, JVET decided to start to standardize the next generation video coding standard in 2017 [6]. Since April 2018, the next generation video coding standard named as VVC and its software model called VVC Test Model (VTM) [7] have emerged, which was released up to version 4.0. The JEM and VTM were developed based on the HEVC Test Model (HM) [8]. Consequently, the basic framework of encoding and decoding is the same with the HEVC, however, the internal coding tools of modules of block structure, intra and inter prediction and transform, loop filter and entropy coding are added and modified.

* Corresponding Author: Byung-Gyu Kim, Address: (04310) Sookmyung Women's Univ., 100, Cheongpa-ro 47-gil, Yongsan-gu, Seoul, Republic of Korea, TEL: +82-2-2077-7293, E-mail: bg.kim@sookmyung.ac.kr
Receipt date: Apr. 1, 2019, Revision date: May 2, 2019
Approval date: June 25, 2019

[†] Dept. of IT Eng., School of Engineering, Sookmyung Women's University
(E-mail: yj.choi@ivpl.sookmyung.ac.kr)

^{**} Dept. of IT Eng., School of Engineering, Sookmyung Women's University

* This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2016R1D1A1B04934750).

Inter prediction, which aims to obtain the similar block in the reference frames in order to reduce the temporal redundancy, is an essential part in the video coding. Main tools for inter prediction are motion estimation (ME) and motion compensation (MC). In ME and MC, finding precise correlation between consecutive frames is important to make better coding performance. Block matching based ME and MC have been implemented in the reference software model of previous video compression standards such as H.264/AVC [31] and H.265/HEVC. The fundamental technique of the conventional block based MC is using translational motion model with integer-pel accuracy. In the early researches, a translational motion model-based MC cannot address complex motions in natural videos such as rotation and zooming. Furthermore, fractional pixel accuracy motion vectors usually give better motion compensated prediction than the integer pixel motion vectors. During the development of the video coding standards, many efforts have been made to find accurate motion. In this paper, we introduce new coding tools for motion estimation and compensation to achieve a good balance between coding efficiency and coding complexity in VVC.

This paper is organized as follows. In section 2, an overview of inter prediction coding structure in HEVC is presented. Section 3 introduces the advanced motion estimation and compensation algorithms for VVC. The experimental results are shown in Section 4. Finally, Section 5 concludes this paper.

2. INTER PREDICTION IN HEVC

2.1 Inter Prediction in HEVC Video Encoder

In the HEVC standard [9],[10],[11],[12],[30], video coding layer is designed to achieve multiple goals. Fig. 1 illustrates the block diagram of a hybrid video encoder. In this process, each picture is split into block-based unit, coding tree units (CTUs), which is the basic processing unit of the HEVC video standard. CTU can be optionally divided into smaller sized blocks. The division can be recursively continued until the maximum depth is reached. The leaf nodes of the quadtree are called coding units (CUs). The encoding process for inter prediction consists of motion estimation (ME) and motion compensation (MC). ME is achieved by finding matching block between a block in the current frame and a similar block in the reference

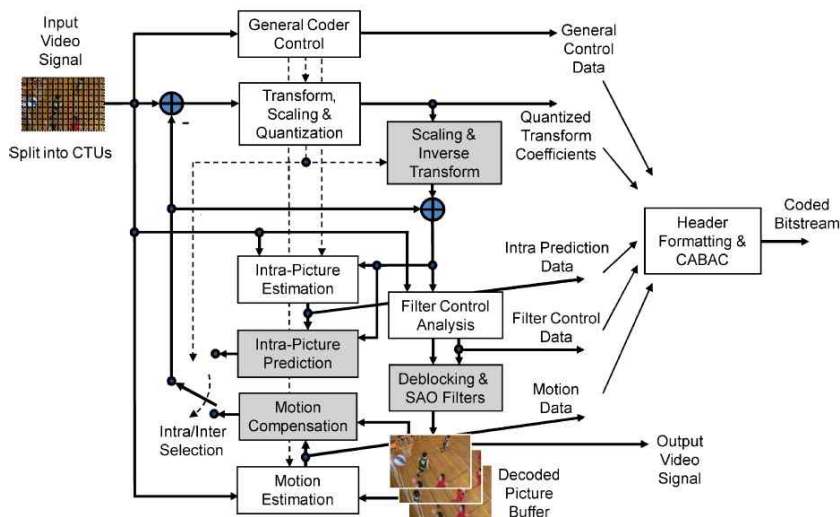


Fig. 1. Illustration of HEVC Video Encoder [9].

frame and generating a residual block through the subtracting two blocks. MC is conducted by generating a reconstructed block using previous block, residual block and motion vector (MV).

2.2 Motion Estimation in HEVC

ME is the procedure of finding optimal matched block in a search window of the reference frame for each coding block in the current frame. Fig. 2 illustrates the ME process. Every current coding block, ME algorithm uses Lagrangian cost function to find best reference block. The function illustrated in (1),

$$J_{MV} = SAD(MV) + \lambda_M \cdot R(MV - PMV), \quad (1)$$

where SAD is the matching function and λ_M is the Lagrangian multiplier. PMV is the predicted motion vector obtained from motion vector prediction process which used to calculate the motion vector difference (MVD). R denotes rate required to encode this MVD.

The full search algorithm for motion estimation has a major problem of significant computational load. As a solution, many researchers proposed the fast motion estimation algorithms. In [13],[14],[15],[16], some of the search pattern based fast ME algorithms are proposed in H.264 and HEVC, e.g. Three Step Search (TSS), Four Step Search (FSS) and Diamond Search (DS). In addition, some fast inter mode decision schemes have been examined in [17],[18],[19],[20]. This fast mode decision works have focused on block partitioning and early termi-

nation mechanisms to speed up inter mode decision process.

To reduce the computational complexity in inter prediction, sufficient works have been proposed and hardware approach is also possible. The research for method to improve encoding efficiency is another important side. In this paper, we analyze some coding methods for ME and MC in inter prediction in the VVC standard based on JVET standard documents.

3. MOTION ESTIMATION AND COMPENSATION FOR VVC

In the VVC standard, techniques that are more complex and more complicated than conventional block-based algorithms have been emerged to improve accuracy. In this paper, we introduce and analyze motion compensation techniques applied in JEM reference. The representative algorithms to introduce are Affine Motion Compensation (AMC), Local Illumination Compensation (LIC) and Overlapped Block Motion Compensation (OBMC).

3.1 Affine Motion Compensation (AMC)

In HEVC, only translational motion model is applied for MC. In the real world, however, there are many kinds of motion, e.g. scale, rotation and irregular motions. To overcome the limitation of translational motion model and maintain low computational complexity, in the JEM, a simplified affine motion compensation prediction [21],[22] is applied. The combination equation of scale, rotation and translation for each pixel can be described as

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \rho \cos \theta & -\rho \sin \theta \\ \rho \sin \theta & \rho \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a \\ b \end{pmatrix}, \quad (2)$$

where ρ is the zooming factor, and θ is the rotation angle. For this motion model, four parameters will be needed.

Instead of these four parameters, it can be represented two MVs because using MVs is more consistent with video coding framework. Two MVs

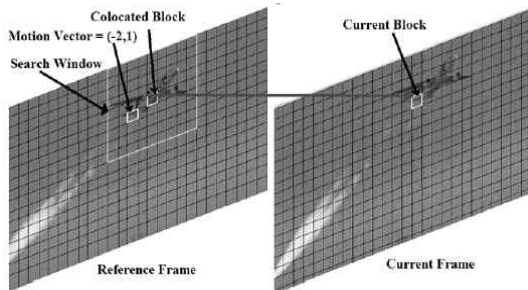


Fig. 2. Illustration of ME Process [16].

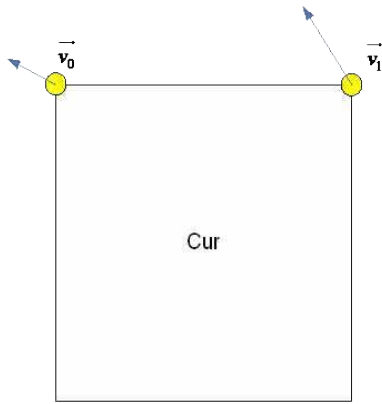


Fig. 3. Control Points of the Affine Motion Model [4].

are called two control point motion vectors. As shown in Fig. 3, the top left and top right location of the current block are selected as two control point motion vectors. The motion vector field (MVF) of a block is described by the following equation:

$$\begin{cases} v_x = \frac{(v_{1x} - v_{0x})}{w}x - \frac{(v_{1y} - v_{0y})}{w}y + v_{0x}, \\ v_y = \frac{(v_{1y} - v_{0y})}{w}x - \frac{(v_{1x} - v_{0x})}{w}y + v_{0y}, \end{cases} \quad (3)$$

where (v_{0x}, v_{0y}) is motion vector of the top left corner control point, and (v_{1x}, v_{1y}) is motion vector of top right corner control point. In order to further simplify, sub-block based affine motion compensation is applied instead of the pixel based motion compensation. The motion vector of the center position of each sub-block, as shown in Fig. 4, is calculated according to equation (3).

In JEM, two affine motion compensation modes are used: AF_INTER mode and AF_MERGE mode. When the current CU is applied in AF_INTER mode, a candidate list consist of the neighbor blocks' MV pairs is constructed. An RD cost check process is used to decide which MV pair candidate is selected as the control point motion vector prediction (CPMVP) of the current CU. After affine motion estimation is conducted and the control point motion vector (CPMV) is found, the difference of the CPMV and the CPMVP is signaled in

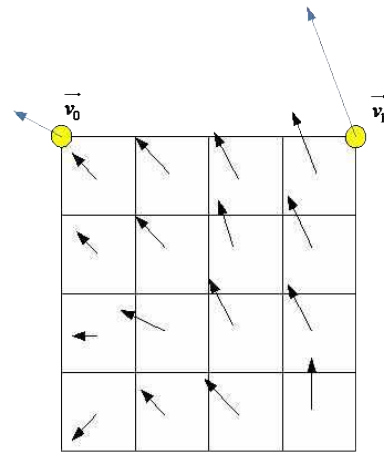


Fig. 4. Affine MVF per Sub-Block [4].

the bitstream. If AF_MERGE mode is used for the current CU, the valid neighbor reconstructed blocks which coded with affine mode are obtained. The selection order of the candidate block is from left, above, above right, left bottom to above left. According to candidate block's CPMV, current block's CPMV can be calculated. After the CPMV of the current CU are derived, through the affine motion model equation (3), the MVF of the current CU is generated.

3.2 Local Illumination Compensation (LIC)

LIC [23] is based on a linear model for illumination changes. When LIC are applied for the current CU, a least square error method is employed to derive the scaling and offset parameter by using the neighboring samples of current CU and their corresponding reference block. As illustrated in Fig. 5, 2:1 subsampled samples in current and reference picture are used. When LIC is enabled for a picture, additional RD cost check process is needed to determine whether LIC is applied or not for a CU.

3.3 Overlapped Block Motion Compensation (OBMC)

OBMC [24] for JEM can be switched on and off at the CU level. When OBMC is used in current

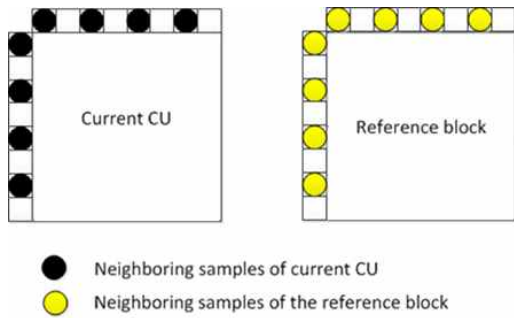


Fig. 5. Neighboring Samples for LIC [4].

CU, OBMC is performed for all block boundaries except the right and bottom boundaries if a CU is coded with sub-CU mode. OBMC is performed at sub-CU level where sub-block size is set equal to 4×4 .

As shown in Fig. 6, MVs of four neighboring sub-blocks of the current sub-block are also used to derive prediction block. Prediction block based on MVs of a neighboring sub-block is denoted as P_N , where N indicates an index for the neighboring above, below, left and right sub-blocks. Prediction block based on MVs of the current sub-block is denoted as P_C . The weighting factors $\{1/4, 1/8, 1/16, 1/32\}$ are used for P_N and weighting factors $\{3/4, 7/8\}$ are used for P_C .

4. EXPERIMENTAL RESULTS

To analyze the performance of three motion compensation tools in JEM, we referenced the

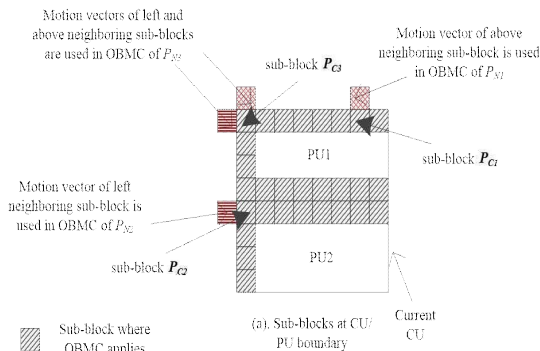


Fig. 6. Sub-Blocks for OBMC [4].

JVET-B0022 [25] document. The performance tests for each tool were based on the JEM 1.0 [26] reference software model. The test conditions were described in the JVET common test conditions (CTC) [27]. Experiments are conducted under random access (RA), low delay B (LDB) and low delay P (LDP) configurations and four base layer quantization parameters (QP) values of 22, 27, 32 and 37. The experimental results were evaluated by the Bjontegaard-Delta Rate (BD-Rate) measurement [28] and encoding and decoding time ratio. The analysis results of each tool consist of tool-on and tool-off tests. In each tool-on test, only one tool was enabled and one tool was disabled in the tool-off test. The tool-on test was measured compared with HM 16.6 [29]. Through two different tests with different characteristics, we evaluated the impact of each tool on JEM.

4.1 Performance of AMC

Table 1 shows the tool-on performance of AMC algorithm. From that results, compared with the HM 16.6 baseline, we can see that AMC tool can bring about 0.9%, 1.6% and 1.9% BD-Rate gain averagely on Y component in RA, LDB and LDP cases respectively. The field of codec standardization is strict for decoder complexity and relatively generous for encoder complexity. Accordingly, AMC is efficient in terms of time complexity.

The tool-off performance of AMC algorithm is shown in Table 2. In LDB and LDP configurations, BD-Rate increased by 1.4% and 1.3% respectively. From that results, it can be seen that the AMC has a great influence on the overall coding gain of JEM. There are only a few encoding and decoding time changes in average, indicating that the computational complexity is negligible.

4.2 Performance of LIC

Table 3 and Table 4 indicate the tool-on and tool-off test results of LIC algorithm respectively. The LIC tool does not show noticeable results

Table 1. The performance of AMC in tool-on test

	Random Access			Low-delay B			Low-delay P		
	Y	U	V	Y	U	V	Y	U	V
Class A	-0.4%	-0.1%	-0.3%						
Class B	-1.4%	-1.2%	-1.2%	-1.5%	-1.5%	-1.8%	-1.7%	-1.1%	-1.0%
Class C	-0.7%	-0.7%	-1.0%	-1.0%	-0.8%	-0.9%	-1.3%	-0.7%	-1.0%
Class D	-1.1%	-1.2%	-1.4%	-2.1%	-2.1%	-1.9%	-2.5%	-2.3%	-1.8%
Class E				-1.8%	-1.0%	-1.8%	-2.1%	-0.3%	-0.8%
Overall	-0.9%	-0.8%	-1.0%	-1.6%	-1.4%	-1.6%	-1.9%	-1.1%	-1.2%
Class F	-1.4%	-1.5%	-1.4%	-1.7%	-1.9%	-1.6%	-1.9%	-1.1%	-1.2%
Enc [%]	118%			118%			110%		
Dec [%]	102%			99%			103%		

Table 2. The performance of AMC in tool-off test

	Random Access			Low-delay B			Low-delay P		
	Y	U	V	Y	U	V	Y	U	V
Class A	0.1%	0.2%	0.0%						
Class B	0.7%	0.7%	0.6%	1.1%	1.3%	1.2%	1.3%	1.4%	1.5%
Class C	0.3%	0.2%	0.2%	0.9%	0.9%	1.2%	0.8%	0.7%	0.6%
Class D	0.6%	0.6%	0.6%	2.0%	2.8%	2.5%	1.7%	2.3%	2.5%
Class E				1.7%	1.3%	0.6%	1.8%	1.2%	1.4%
Overall	0.4%	0.4%	0.4%	1.4%	1.6%	1.4%	1.3%	1.4%	1.5%
Class F	0.8%	0.6%	0.5%	1.1%	1.2%	0.9%	1.3%	1.4%	1.5%
Enc [%]	93%			92%			93%		
Dec [%]	96%			96%			92%		

Table 3. The performance of LIC in tool-on test

	Random Access			Low-delay B			Low-delay P		
	Y	U	V	Y	U	V	Y	U	V
Class A	-0.2%	1.0%	1.5%						
Class B	-0.5%	-0.2%	-0.2%	-0.6%	-0.4%	-0.4%	-1.3%	-0.6%	-0.3%
Class C	-0.5%	-0.4%	-0.6%	-0.6%	-0.5%	-0.4%	-1.1%	-0.5%	-0.5%
Class D	-0.1%	-0.1%	-0.2%	-0.2%	-0.5%	-0.2%	-0.4%	-0.7%	-0.4%
Class E				0.0%	0.6%	0.1%	-0.1%	0.0%	-0.1%
Overall	-0.3%	0.1%	0.1%	-0.4%	-0.3%	-0.3%	-0.8%	-0.5%	-0.3%
Class F	-1.5%	-1.1%	-1.0%	-3.8%	-2.2%	-2.4%	-0.8%	-0.5%	-0.3%
Enc [%]	112%			116%			109%		
Dec [%]	100%			96%			99%		

compared to other inter prediction algorithms. The screen content sequences in class F, which contain graphics, text or animation rather than camera-captured video scenes. Screen content has different characteristics compared with natural video cap-

tured by cameras. Unlike camera-captured content, a screen content may has large uniformly flat areas, repeated patterns, and numerically identical blocks rather than objects with complex textures or motions. In such a case, a scene composed of

Table 4. The performance of LIC in tool-off test

	Random Access			Low-delay B			Low-delay P		
	Y	U	V	Y	U	V	Y	U	V
Class A	0.3%	-0.8%	-1.9%						
Class B	0.6%	0.2%	0.2%	0.8%	0.3%	0.2%	0.8%	0.2%	0.7%
Class C	0.6%	0.5%	0.4%	0.9%	0.7%	1.2%	1.1%	0.5%	0.4%
Class D	0.2%	0.2%	0.3%	0.4%	0.5%	0.7%	0.5%	0.9%	0.9%
Class E				0.1%	0.2%	-0.5%	0.3%	0.5%	0.6%
Overall	0.4%	0.0%	-0.2%	0.6%	0.4%	0.5%	0.7%	0.5%	0.7%
Class F	1.6%	0.9%	0.9%	3.7%	3.4%	3.2%	0.7%	0.5%	0.7%
Enc [%]	91%			85%			89%		
Dec [%]	96%			92%			90%		

some static pictures change occurs not smoothly. Therefore, it is more effective to compensate the motion than to find the motion accurately, in terms of brightness change. For that reason, the results for the F class sequence show better performance. Especially in LDB, which is the encoding config-

uration utilizing the reference frames most dynamically, it achieves 3.8% BD-Rate saving for the Y component.

4.3 Performance of OBMC

Table 5 and Table 6 indicate the tool-on and

Table 5. The performance of OBMC in tool-on test

	Random Access			Low-delay B			Low-delay P		
	Y	U	V	Y	U	V	Y	U	V
Class A	-1.6%	-3.0%	-2.7%						
Class B	-1.6%	-2.5%	-2.4%	-1.8%	-2.6%	-2.2%	-4.7%	-5.3%	-4.9%
Class C	-2.3%	-3.3%	-3.7%	-2.7%	-3.3%	-3.6%	-4.8%	-4.6%	-5.3%
Class D	-2.2%	-3.1%	-3.0%	-2.9%	-3.2%	-2.7%	-5.8%	-5.6%	-4.5%
Class E				-1.8%	-2.5%	-2.4%	-5.4%	-5.1%	-5.1%
Overall	-1.9%	-3.0%	-2.9%	-2.3%	-2.9%	-2.7%	-5.2%	-5.2%	-4.9%
Class F	-0.5%	-1.2%	-1.3%	-0.5%	-1.4%	-1.5%	-5.2%	-5.2%	-4.9%
Enc [%]	110%			105%			103%		
Dec [%]	123%			119%			119%		

Table 6. The performance of OBMC in tool-off test

	Random Access			Low-delay B			Low-delay P		
	Y	U	V	Y	U	V	Y	U	V
Class A	1.2%	3.5%	3.3%						
Class B	1.4%	2.6%	2.4%	1.9%	3.0%	3.1%	3.0%	4.2%	4.3%
Class C	2.3%	3.0%	3.2%	2.8%	3.6%	3.9%	3.9%	4.4%	3.9%
Class D	2.2%	2.5%	2.5%	3.1%	3.3%	3.2%	5.0%	4.5%	4.4%
Class E				3.2%	2.9%	2.3%	4.1%	4.3%	4.4%
Overall	1.5%	2.5%	2.5%	2.2%	2.8%	2.8%	3.3%	3.8%	3.7%
Class F	0.4%	1.0%	0.9%	0.2%	1.0%	1.1%	0.8%	1.9%	1.7%
Enc [%]	85%			89%			95%		
Dec [%]	63%			70%			81%		

tool-off test results of OBMC algorithm respectively. The OBMC can improve the coding gain of JEM remarkably but decoder complexity is also high. The OBMC brings 1.9%, 2.3% and 5.2% average BD-Rate savings on Y component in RA, LDB and LDP configurations respectively. Especially in LDP, shows extremely high gain. The advantage of the OBMC algorithm is that it can obtain the final compensated MV over two or more prediction MVs by using boundary sub-blocks as additional prediction blocks. Compared to the RA and LDB, which allow bi-directional coding schemes so have two or more prediction blocks, LDP has one prediction block. That is why the coding effect is maximized. However, as shown in Table 6, when the OBMC tool is disabled, the decoding time results decrease by 37%, 30% and 19% in RA, LDB and LDP configurations respectively. Therefore, it is not efficient in terms of time complexity.

5. CONCLUSIONS

In this paper, several important coding features for motion estimation and motion compensation in the VVC standard have been introduced and analyzed. To resolve numerous problems of the traditional block-based inter prediction procedure in the HEVC, many approaches have been proposed. With the development of many kinds of video, the research for method to improve encoding efficiency is still important side. In JEM, techniques that are more complex and more complicated been emerged to improve accuracy.

The affine motion compensation (AMC), local illumination compensation (LIC) and overlapped block motion compensation (OBMC) were introduced as the representative algorithms for inter-prediction in this paper. As can be seen from the experimental results, AMC brings high coding gain and is effective in terms of encoding and decoding time complexity. In results of LIC, screen content which has large uniformly flat areas, repeated patterns, and numerically identical blocks rather than

objects with complex textures or motions shows better result rather than the normal natural video. In this case, it is more efficient to compensate in terms of brightness change than to find motion exactly. The results of OBMC show extremely high gain in LDP configuration. Compared to the RA and LDB, which allow bi-directional coding schemes, LDP has only one predication block for each CUs. The advantage of the OBMC algorithm is that it can obtain the final compensated MV over two or more prediction MVs. Therefore, in LDP, the coding effect is maximized. Improved coding tools introduced in this paper for ME and MC in the VVC can achieve much better good balance between coding efficiency and coding complexity compared with the previous HEVC standard.

REFERENCE

- [1] B. Bross, J. Chen, and S. Liu, *Versatile Video Coding (Draft 4)*, Joint Video Exploration Team Document JVET-M1001, 2019.
- [2] *High Efficiency Video Coding*, Recommendation ITU-T H.264, ISO/IEC 14496-10, 2003.
- [3] *High Efficiency Video Coding*, Recommendation ITU-T H.265, ISO/IEC 23008-2, 2013.
- [4] J. Chen, E. Alshina, Gary J. Sullivan, J.-R. Ohm, and J. Boyce, *Algorithm Description of Joint Exploration Test Model 7 (JEM 7)*, Joint Video Exploration Team Document JVET-G1001, 2017.
- [5] Future Video Coding FVC Test Model 7.2 (JEM 7.2), https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/HM-16.6-JEM-7.2/ (accessed Apr., 1, 2019).
- [6] A. Segall, V. Baroncini, J. Boyce, J. Chen, and T. Suzuki, *Joint Call for Proposals on Video Compression With Capability Beyond HEVC*, Joint Video Exploration Team Document JVET-H1002, 2017.
- [7] J. Chen, Y. Ye, and S. Kim, *Algorithm Description for Versatile Video Coding and Test Model 4 (VTM 4)*, Joint Video Exploration

- Team Document JVET-M1002, 2018.
- [8] High Efficiency Video Coding HEVC Test Model 16.10 (HM 16.10), https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.10/ (accessed Apr., 1, 2019).
 - [9] G.J. Sullivan, J.R. Ohm, W.J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 22, No. 12, pp. 1649–1668, 2012.
 - [10] *High Efficiency Video Coding*, Recommendation ITU-T H.265, ISO/IEC 23008-2, 2013.
 - [11] C. Rosewarne, B. Bross, M. Naccari, K. Sharman, and G.J. Sullivan, *High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Update 4 of Encoder Description*, Joint Collaborative Team of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 on Video Coding, JCTVC-V1002, 2015.
 - [12] X. Chen, J. An, and J. Zheng, *EE3: Decoder-Side Motion Vector Refinement Based on Bilateral Template Matching*, Joint Video Exploration Team of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JVET-E0052, 2017.
 - [13] R. Li, B. Zeng, and M.L. Liou, "A New Three-step Search Algorithm for Block Motion Estimation," *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 4, pp. 438–442, 1994.
 - [14] L.M. Po and W.C. Ma, "A Novel Four-step Search Algorithm for Fast Block Motion Estimation," *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 6, pp. 313–317, 1996.
 - [15] S. Zhu and K.K. Ma, "A New Diamond Search Algorithm for Fast Block Matching Motion Estimation," *Proceeding of International Conference Information, Communication, and Signal Processing*, pp. 292–296, 1997.
 - [16] N. Purnachand, L.N. Alves, and A. Navarro, "Fast Motion Estimation Algorithm for HEVC," *Proceeding of IEEE Second International Conference on Consumer Electronics*, pp. 34–37, 2012.
 - [17] J. Yang, J. Kim, K. Won, H. Lee, and B. Jeon, *Early SKIP Detection for HEVC*, Document JCTVC-G543, 2011.
 - [18] R.H. Gweon and Y.L. Lee, *Early Termination of CU Encoding to Reduce HEVC Complexity*, Document JCTVC-F045, 2011.
 - [19] L. Shen, Z. Liu, X. Zhang, W. Zhao, and Z. Zhang, "An Effective CU Size Decision Method for HEVC Encoders," *IEEE Transaction on Multimedia*, Vol. 15, No. 2, pp. 465–470, 2013.
 - [20] H.L. Tan, F. Liu, Y.H. Tan, and C. Yeo, "On Fast Coding Tree Block and Mode Decision for High Efficiency Video Coding (HEVC)," *Proceeding of IEEE International Conference Acoustical, Speech, and Signal Processing*, pp. 825–828, 2012.
 - [21] S. Lin, H. Chen, H. Zhang, S. Maxim, H. Yang, and J. Zhou, *Affine Transform Prediction for Next Generation Video Coding*, MPEG Document m37525 and ITU-T SG16 Document COM16-C1016, 2015.
 - [22] Li. Li, H. Li, D. Liu, H. Yang, S. Lin, H. Chen, et al., "An Efficient Four-parameter Affine Motion Model for Video Coding," *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 28, No. 8, pp. 1934–1948, 2017.
 - [23] H. Liu, Y. Chen, J. Chen, L. Zhang, and M. Karczewicz, *Local Illumination Compensation*, ITU-T SG16/Q6 Document VCEG-AZ06, 2015.
 - [24] M. Karczewicz, J. Chen, W.J. Chien, X. Li, A. Said, L. Zhang, et al., *Study of Coding Efficiency Improvements beyond HEVC*, MPEG Document m37102, 2015.
 - [25] E. Alshina, A. Alshin, K. Choi, and M. Park, *Performance of JEM1.0 Tools Analysis by Samsung*, Joint Video Exploration Team Document JVET-B0022, 2016.
 - [26] Future Video Coding FVC test model 1.0 (JEM

1.0), https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/tags/HM-16.6-JEM-1.0/ (accessed Apr., 1, 2019).

- [27] K. Suehring and X. Li, *JVET Common Test Conditions and Software Reference Configurations*, Joint Video Exploration Team Document JVET-B1010, 2016.
- [28] G. Bjntegaard, *Calculation of Average PSNR Differences between RD-Curves*, ITU-T SG. 16 Q.6, Document VCEG-M33, 2001.
- [29] High Efficiency Video Coding HEVC Test Model 16.6 (HM 16.6), https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.6/ (accessed Apr., 1, 2019).
- [30] K. Goswami, D.Y. Lee, J.H. Kim, S.Y. Jeong, H.Y. Kim, and B.G. Kim, "Two-step Rate Distortion Optimization Algorithm for High Efficiency Video Coding," *Journal of Multimedia Information System*, Vol. 4, No. 4, pp. 311-316, 2017.
- [31] D.S. Lee and Y.M. Kim, "Efficient Coding of Motion Vector and Mode Information for H. 264/AVC," *Journal of Korea Multimedia Society*, Vol. 11, No. 10, pp. 1359-1365, 2008.



Young-Ju Choi

received her BS degrees in the Department of IT Engineering from Sookmyung Women's University, Korea, in 2017. She is a Ph. D. candidate in Department of IT Engineering at Sookmyung Women's University.

Her research interests include image and video coding techniques, deep learning-based video recognition, and intelligent/perceptual video coding algorithms.



Byung-Gyu Kim

has received his BS degree from Pusan National University, Korea, in 1996 and an MS degree from Korea Advanced Institute of Science and Technology (KAIST) in 1998. In 2004, he received a Ph. D. degree in the Department

of Electrical Engineering and Computer Science from Korea Advanced Institute of Science and Technology (KAIST). In March 2004, he joined in the real-time multimedia research team at the Electronics and Telecommunications Research Institute (ETRI), Korea where he was a senior researcher.

From February 2009 to February 2016, he was associate professor in the Division of Computer Science and Engineering at SunMoon University, Korea. In March 2016, he joined the Department of Information Technology (IT) Engineering at Sookmyung Women's University, Korea where he is currently a professor. He is serving as an associate editor of *Circuits, Systems and Signal Processing* (Springer), *The Journal of Supercomputing* (Springer), *The Journal of Real-Time Image Processing* (Springer), and *International Journal of Image Processing and Visual Communication* (IJIPVC). From March 2018, he is serving as the Editor-in-Chief of *The Journal of Multimedia Information System* and associate editor of *IEEE Access Journal*. From 2019, he was appointed as the associate editor of *Heliyon-Computer Science* (Elsevier).

His research interests include software-based image and video object segmentation for the content-based image coding, video coding techniques, 3D video signal processing, wireless multimedia sensor network, embedded multimedia communication, and intelligent information system for image signal processing.