

## PAPER

# Prefiltering and Postfiltering Based on Global Motion Compensation for Improving Coding Efficiency in H.264 and HEVC Codecs

Ho Hyeong RYU<sup>†</sup>, Kwang Yeon CHOI<sup>†</sup>, *Nonmembers*, and Byung Cheol SONG<sup>†a)</sup>, *Member*

**SUMMARY** In this paper, we propose a filtering approach based on global motion estimation (GME) and global motion compensation (GMC) for pre- and postprocessing of video codecs. For preprocessing a video codec, group of pictures (GOP), which is a basic unit for GMC, and reference frames are first defined for an input video sequence. Next, GME and GMC are sequentially performed for every frame in each GOP. Finally, a block-based adaptive temporal filter is applied between the GMC frames before video encoding. For postprocessing a video codec at the decoder end, every decoded frame is inversely motion-compensated using the transmitted global motion information. The holes generated during inverse motion compensation can be filled with the reference frames. The experimental results show that the proposed algorithm provides higher Bjontegaard-delta peak signal-to-noise ratios (BD-PSNRs) of 0.63 and 0.57 dB on an average compared with conventional H.264 and HEVC platforms, respectively.

**key words:** global motion, rotation motion, zoom motion, CODEC, temporal filter

## 1. Introduction

Global motion compensation (GMC) has been adopted as a coding tool in a video coding standard MPEG-4 advanced simple profile (ASP) [1]. However, the GMC was excluded from recent coding standards because of its huge computational complexity. As the state-of-the-art video codecs such as H.264 and HEVC do not consider global motion, their coding efficiency can deteriorate for video sequences having zoom or rotation. While in the real world, there are many kinds of motions, e.g. zoom in/out, rotation, perspective motions and the other irregular motions.

To improve coding efficiency in rate-distortion, several approaches have been developed recently to apply GMC to a coding loop. Lee and Lee proposed a method to generate a new reference frame based on the previous or current frame such that the frames have the same orientations by adopting a specific video stabilization technique [2]. Kim et al. proposed a method for estimating the zoom motion for achieving high compression in video coding with low complexity [3]. However, these algorithms have an inherent drawback: they are not compatible with standard video codecs because they should be implemented inside the coding loop

at an encoder or a decoder. In Joint Exploration Model (JEM), which describes the coding features that are under coordinated test model study by the Joint Video Exploration Team (JVET) of ITU-T VCEG and ISO/IEC MPEG as potential enhanced video coding technology beyond the capabilities of HEVC, a simplified affine transform motion compensation prediction is applied to further improve the coding efficiency [10].

If we employ some image processing tools, such as GMC and temporal filtering, for pre- or postfiltering of standard codecs, we can improve the video coding efficiency while preserving compatibility with video coding standards. For example, an input video sequence can be globally motion-compensated through video stabilization [4] for camera deshaking, and the motion-compensated video sequence can then be encoded. However, such a process has two drawbacks: video stabilization causes data loss and the original global motion of the input video sequence is not preserved. Furthermore, conventional video stabilization algorithms lose the frame boundary.

In this study, we employed GMC and temporal filtering as preprocessing methods for obtaining compatibility with standard video codecs. In addition, we performed inverse GMC (IGMC) as a postprocessing method of a video decoder to improve the coding efficiency while preserving the original global motion of an input video sequence. First, the input video sequence is partitioned into group of pictures (GOPs), which is the basic unit for GMC. Two end frames of each GOP are chosen as key or reference frames. Next, global motion estimation (GME) is performed to determine homography between nonkey and key frames in the GOP by using the Lucas–Kanade method [5]. Simultaneously, we predicted the homography for filling holes during the IGMC at the decoder end. Next, by using the estimated homography and a particular interpolation algorithm, we performed GMC for the nonkey frame. Because some holes can be generated during motion compensation, a hole-filling procedure was adopted based on key frames. Finally, adaptive temporal filtering was applied to GMC frames on a block basis. The output video sequence is encoded with GMC information, which is transmitted within a user data section. The required postprocessing at the decoder is IGMC. As some holes can be produced even in this step, the holes should be filled with key frame(s) according to the transmitted GMC information.

Manuscript received June 21, 2016.

Manuscript revised September 1, 2016.

Manuscript publicized October 7, 2016.

<sup>†</sup>The authors are with Department of Electronic Engineering, Inha University, Incheon, 22212 Korea.

a) E-mail: bcsong@inha.ac.kr

DOI: 10.1587/transinf.2016EDP7273

This paper is organized as follows. Section 2 provides the detailed description of the proposed algorithm. Section 3 shows intensive experimental results. Finally, Sect. 4 provides the concluding remarks.

## 2. Proposed Algorithm

Figure 1 shows the block diagram of the proposed algorithm. Before the detailed description of the algorithm, the following assumptions must be considered:

- As the proposed algorithm does not depend on a specific coding standard, we just employed H.264 and HEVC as the basic platforms.
- An input video sequence does not have any scene change. If necessary, we can employ a proper scene change detection algorithm.
- Global motion like zoom and rotation as well as translation is represented in a single homography. The proposed algorithm can be extended to multiple homography cases.
- No object motion exists at the frame boundary.

### 2.1 GME

GME is the first step of the proposed algorithm, as seen in Fig. 1. Because key frames must exist with a proper interval, we must define GOP (see Fig. 2), which is the basic unit for GME. Two end frames of each GOP are named as the key frames, and are the reference frames for GME. Key frame1  $f_0$  is the reference frame for IGMC at the decoder and for GMC at the encoder, and key frame2  $f_{K+1}$  is the reference frame to recover holes generated after IGMC at the decoder. Here, assume that the number of nonkey frames between two key frames in each GOP is  $K$ .

Figure 2 describes GME of the  $k$ -th frame  $f_k$  for two key frames. First, homography  $H_{k,0}$  is estimated between  $f_k$  and  $f_0$  (see the solid line). Here, the order in subscript indicates the direction of motion estimation. Similarly, two more homographies, that is,  $H_{0,k}$  and  $H_{K+1,k}$ , are estimated simultaneously. For homography estimation, we adopted the famous Lucas–Kanade method [5]. Thus, three homographies are estimated per nonkey frame, and are transmitted to the decoder within the supplemental enhancement information (SEI) section of H.264 and HEVC bit-stream. Assuming 4 bytes per parameter, the additional information for homographies is only 108 bytes per frame.

For motion continuity, the second key frame of each GOP becomes the first key frame of the following GOP, as shown in Fig. 2. Note that the GOP in the proposed scheme is independent of that inside coding loop even though they have the same name. Since the proposed scheme exists outside coding loop, the period of key frames does not affect the GOP structure inside coding loop.

### 2.2 GMC

The GMC procedure consists of an image registration step

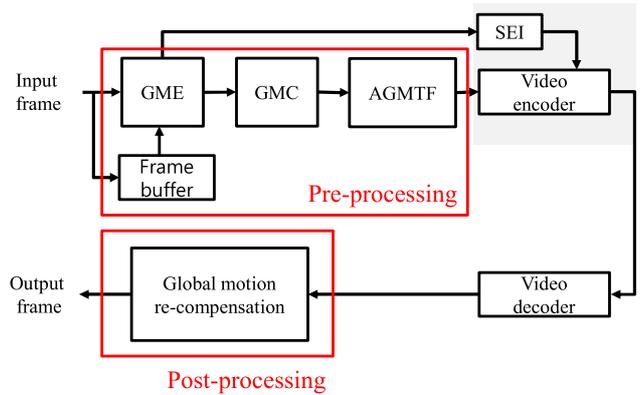


Fig. 1 Block diagram of the proposed algorithm.

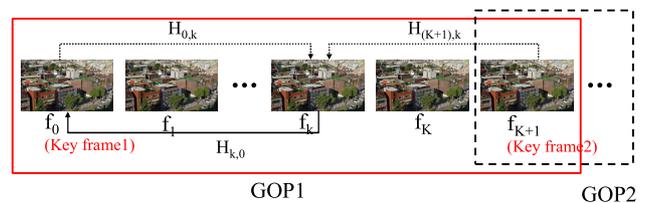


Fig. 2 GOP structure of the proposed method.

with homographies and a hole-filling step.

#### 2.2.1 Image Registration

Assume image registration for  $f_k$  in the GOP. On the basis of  $H_{k,0}$ , the backward warping of map  $f_k$  to  $f_0$  is defined as follows:

$$\begin{aligned} i' &= h_{11} * i + h_{12} * j + h_{13} \\ j' &= h_{12} * i + h_{22} * j + h_{23} \end{aligned} \quad (1)$$

where  $(i, j)$  and  $(i', j')$  indicate a pixel position at  $f_0$  and its corresponding pixel position at  $f_k$ , respectively. In addition, the  $h$  coefficients of Eq. (1) represent the elements of  $H_{k,0}$ . Note that  $(i', j')$  is generally a noninteger pixel position. Therefore, we require a proper interpolation method. In this paper, we employed a 6-tap Lanczos filter. Let  $f_k^G$  denote a GMC frame. Figure 3 shows an exemplar nonkey frame and its GMC frame.

#### 2.2.2 Hole Filling

After GMC, holes can be generated consequentially (see the blue areas in Fig. 3 (b)). Concurrently, lost regions can be produced as a retroaction of holes, depicted in Sect. 2.3. Although holes are not parts of the current frame, they should be filled before encoding because they can deteriorate the overall coding efficiency. The dummy holes to be filled are removed after decoding through specific postprocessing. Holes are filled according to Eq. (2).

$$f_k^G(i, j) = \begin{cases} f_0(i, j) & \text{if } f_k^G(i, j) \in \text{Hole Area} \\ f_k^G(i, j) & \text{otherwise} \end{cases} \quad (2)$$

For instance, since hole pixels in Fig. 3 (b) are equivalent to the co-located pixels in key frame1  $f_0$  of Fig. 3 (a), they can be filled with the co-located pixels. Figure 3 (c) shows the results of hole filling. If the global motion of the scene is represented using a single homography, such holes can be perfectly filled without any artifacts. Thus, the coding efficiency does not deteriorate.

After all the GMC procedures including hole filling are completed, adaptive temporal filtering is performed between the GMC frames in the GOP. For temporal filtering, we employed Krutz's approach [6]. Krutz et al. applied GMC to the coding loops in the decoder and encoder, and then performed block-based temporal filtering. This is called adaptive global motion temporal filter (AGMTF). In this paper, we adopted this AGMTF for preprocessing the video encoder, and applied it to the GMC frames in the GOP. Since AGMTF can remove unnecessary and noise-like high-frequency components in the GMC frames, we can obtain additional coding efficiency. As shown in Fig. 1, the video sequence after AGMTF is encoded.

### 2.3 IGMC

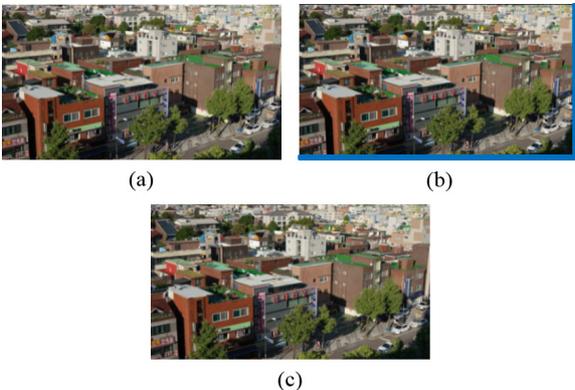
Note that decoded images are the GMC images reprocessed at the encoder. Therefore, the decoded images must be inversely motion-compensated to preserve the original global motion of the input video sequence. The IGMC consists of an image registration step with inverse homography and a hole-filling step.

#### 2.3.1 Image Registration

Assume image registration for  $f_k$  in the GOP. First,  $H_{k,0}$  is extracted from SEI, and its inverse matrix is computed. Then, to map the decoded  $k$ -th frame  $f_k$  to  $f_k^{DG}$ , backward warping is performed as follows:

$$\begin{aligned} i' &= \text{inv}(h_{11}) * i + \text{inv}(h_{12}) * j + \text{inv}(h_{13}) \\ j' &= \text{inv}(h_{21}) * i + \text{inv}(h_{22}) * j + \text{inv}(h_{23}) \end{aligned} \quad (3)$$

where  $(i, j)$  and  $(i', j')$  indicate the pixel position at  $f_k$  and its



**Fig. 3** Example of hole filling in a specific scene. (a)  $f_0$ , (b) GMC  $f_k$ , (c) hole-filled  $f_k$ .

corresponding pixel position at  $f_k^{DG}$ , respectively. Furthermore,  $\text{inv}(h)$  coefficients of Eq. (3) represent the elements of  $\text{inv}(H_{k,0})$ . As  $(i', j')$  is generally a noninteger pixel position, we employed the 6-tap Lanczos filter. At this step, the dummy holes generated earlier (in Sect. 2.2.2) are removed. On the opposite side of the removed dummy data, new holes may be produced. The new holes are equivalent to the lost regions during GMC at the encoder end. These holes are compensated using two key frames, as described in Sect. 2.3.2.

#### 2.3.2 Hole Filling

Without loss of generality, two key frames may contain the new holes. Therefore, hole filling for the  $k$ -th frame after IGMC, that is,  $f_k^{DG'}$ , is performed using Eq. (4).

$$f_k^{DG'}(i, j) = \begin{cases} f_0^{DG}(i, j) & \text{if } f_k^{DG'}(i, j) \in \text{Loss Area1} \\ f_{K+1}^{DG}(i, j) & \text{if } f_k^{DG'}(i, j) \in \text{Loss Area2} \\ f_k^{DG'z}(i, j) & \text{otherwise} \end{cases} \quad (4)$$

where  $f_0^{DG}$  and  $f_{K+1}^{DG}$  indicate the motion-compensated  $f_0$  and  $f_{K+1}$  by  $H_{0,k}$  and  $H_{K+1,k}$ , respectively. Loss areas 1 and 2 represent the lost regions at key frames 1 and 2, respectively.

## 3. Experimental Results

We used four full HD (1920×1080) video sequences, which were acquired from a mirrorless DSLR camera. They are named *test 1*, *2*, *3*, and *4*. Each sequence of 150 frames includes a global motion, such as zoom and rotation. In addition, we employed five 1280×720 MPEG test sequences with global motion (*Big ship*, *City*, *Jets*, *Stockholm*, and *Station*). We used the first 150 frames of each sequence. Table 1 describes the global motion types of test sequences in detail. The GMC GOP size of the proposed algorithm was set to nine, that is,  $K = 7$ . The block size for AGMTF was set to  $32 \times 32$ .

To evaluate the performance of the proposed algorithm, we first adopted H.264 reference software JM 9.0 [7] as the basic platform for the proposed algorithm. Table 2 describes some important encoding parameters for this experiment.

The proposed algorithm was compared with Krutz's algorithm [6] in terms of BD-PSNR and BD-rate [8]. Note

**Table 1** Global motion types of test sequences.

Name	Motion type
<i>Test 1</i>	Strong translation
<i>Test 2</i>	Weak rotation & strong translation
<i>Test 3</i>	Strong zoom
<i>Test 4</i>	Strong zoom
<i>Big ship</i>	Weak translation
<i>City</i>	Strong translation & weak rotation
<i>Jets</i>	Weak zoom
<i>Stockholm</i>	Strong translation
<i>Station</i>	Strong zoom

**Table 2** H.264 encoder setting.

Sequence type	IPPP
Motion estimation scheme	EPZS
Number of reference frames	4
RD optimization	Off
Profile	High
Search range	32
Quantization parameters (QP)	22, 24, 26, 28
Entropy coding	CABAC
Skip intra in inter slices	Off

**Table 3** Performance comparison in terms of BD-PSNR [dB] and BD-rate [%] on the H.264 platform. Left and right values of slash indicates BD-PSNR and BD-rate, respectively.

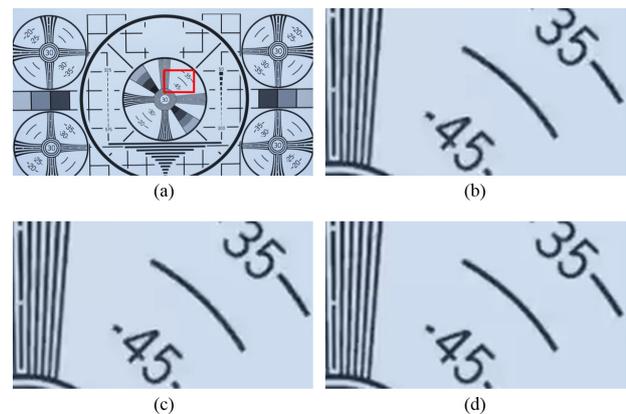
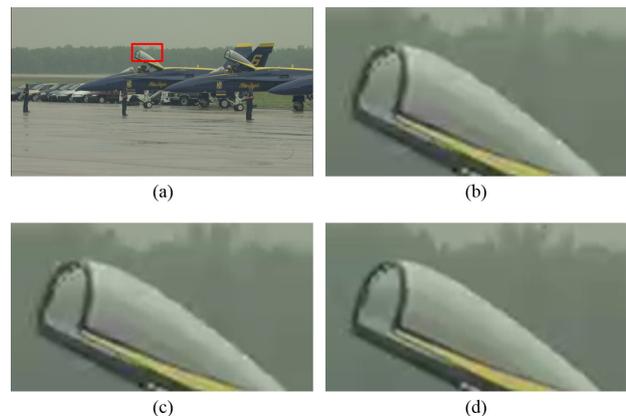
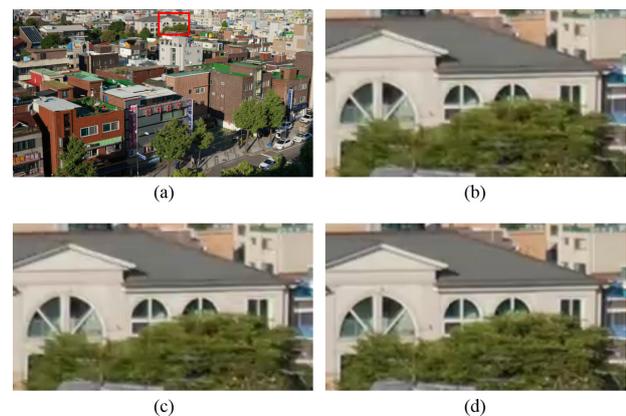
Name	Krutz's [6]	Only GMC	Proposed
<i>Test 1</i>	0.01/-0.9	0.69 / -15.73	1.01 / -24.47
<i>Test 2</i>	0.02/-1.1	0.14 / -3.11	0.32 / -9.45
<i>Test 3</i>	0.21/-7.5	0.81 / -19.76	1.24 / -27.89
<i>Test 4</i>	0.11/-6.3	0.98 / -26.50	1.26 / -31.34
<i>Big ship</i>	0.03/-1.1	0.16 / -5.28	0.20 / -8.43
<i>City</i>	0.09/-4.5	0.35 / -7.46	0.68 / -32.73
<i>Jets</i>	0.12/-4.7	0.10 / -3.71	0.22 / -8.30
<i>Stockholm</i>	0.12/-3.1	0.03 / -1.11	0.13 / -9.19
<b>Average</b>	0.09/-3.7	0.41/ -10.33	0.63 / -18.98

that the reference frames for computing PSNRs are the original input frames. Table 3 shows the results. For fair comparison, hole filling of the proposed algorithm was excluded in Table 3. Krutz's algorithm shows only a small improvement of approximately 0.1 dB over H.264, whereas the proposed algorithm provides a significant BD-PSNR improvement of 0.63 dB on average. The proposed filtering without AGMTF, i.e., 'GMC only' also provides BD-PSNR increase of 0.41dB on average. This indicates that the proposed GMC-based pre-processing creates a synergy effect with temporal filtering such as AGMTF. Especially, it is worth noting that the proposed algorithm gives an outstanding coding efficiency for *test 3* and *test 4* sequences that have strong zoom motion. On the other hand, it is notable that the proposed algorithm provides high BD-rate of about 19% on average.

To evaluate the subjective visual quality, we compared some results for *test 3* sequence as in Fig. 4. *Test 3* image was created by shooting a moving chart. The images were encoded under the same bit-rate condition with a proper quantization parameter (QP) control. With a significant PSNR gap, the proposed algorithm provides outstanding visual quality without any artifacts. Furthermore, Fig. 5 shows the comparison results for the *Jets* sequence. These images were also encoded under the same bit-rate condition with a proper QP control.

Again, with a significant PSNR gap, the proposed algorithm provides outstanding visual quality without any artifacts. Similarly, Fig. 6 shows the comparison results for *test 1* sequence of a city view with camera motion. The proposed algorithm shows better edges and textures than conventional H.264 and the method proposed in [6].

In addition, we evaluated the performance of the proposed algorithm on the HEVC reference software HM 16.7 [9] as the basic platform. Table 4 describes some im-


**Fig. 4** Comparison for *test 3* sequence. (a) The third input frame, (b) conventional H.264 (QP = 36, 33.4 dB), (c) Krutz's method (QP = 36, 33.7 dB), (d) proposed method (QP = 32, 35.4 dB). (b)–(d) represent the zoomed-in image for the red box in (a).

**Fig. 5** Comparison for *Jets* sequence. (a) The second input frame; (b) conventional H.264 (QP = 35, 34.4 dB); (c) Krutz's method (QP = 35, 34.5 dB); (d) proposed method (QP = 32, 35.7 dB). (b)–(d) represent the zoomed-in image for the red box in (a).

**Fig. 6** Comparison results for *test 1* sequence. (a) The third input frame; (b) conventional H.264 (QP = 32, 31.1 dB), (c) Krutz's method (QP = 32, 31.1 dB), (d) proposed method (QP = 28, 32.3 dB). (b)–(d) represent the zoomed-in image for the red box in (a).

**Table 4** HEVC encoder setting.

Sequence type	Low delay mode (no GPB, no QP offset)
Motion estimation scheme	TZ search
Number of reference frame	4
RD optimization	QP factor = 0.3
Search range	32
Quantization parameter (QP)	22, 24, 26, 28
Entropy coding	CABAC

**Table 5** Performance comparison in terms of BD-PSNR [dB] on the HEVC platform.

Name	Krutz's [6]	Only GMC	Proposed
<i>Test 1</i>	0.17	0.43	0.74
<i>Test 2</i>	0.13	0.31	0.51
<i>Test 3</i>	0.15	0.34	0.61
<i>Test 4</i>	0.14	0.30	0.62
<i>Big ship</i>	0.05	0.19	0.39
<i>City</i>	0.09	0.33	0.52
<i>Jets</i>	0.12	0.41	0.64
<i>Stockholm</i>	0.15	0.29	0.61
<i>Station</i>	0.11	0.22	0.50
<b>Average</b>	0.12	0.31	0.57

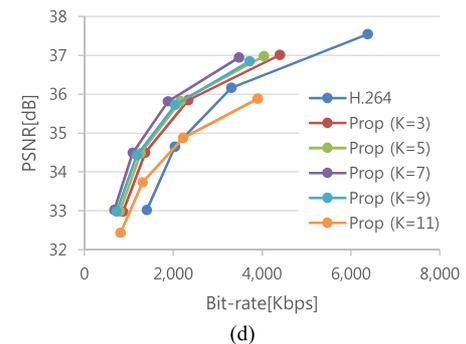
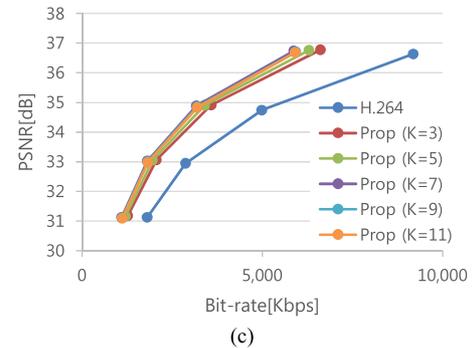
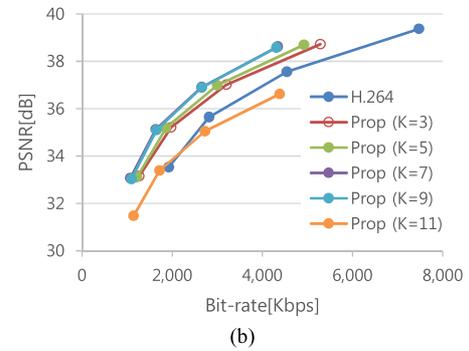
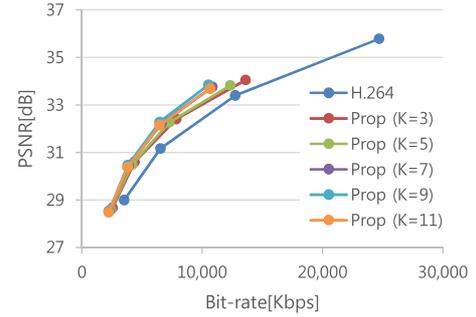
**Table 6** Performance comparison in terms of BD-PSNR on a JEM platform [11].

Name	Affine MCP in JEM [11]	Proposed
<i>Test 1</i>	0.21	0.35
<i>Test 2</i>	0.19	0.22
<i>Test 3</i>	0.17	0.21
<i>Test 4</i>	0.14	0.28
<i>Big ship</i>	0.11	0.29
<i>City</i>	0.13	0.31
<i>Jets</i>	0.15	0.25
<i>Station</i>	0.12	0.20
<b>Average</b>	0.15	0.26

portant encoding parameters for this experiment, which coincide with those of Table 2. Table 5 shows the BD-PSNR results of the proposed algorithm on the HEVC platform for several video sequences. The proposed algorithm still provides a significant BD-PSNR improvement of 0.57 dB on average. Although the stronger inter-prediction of HEVC compared with that of H.264 somewhat mitigates the effect of the proposed algorithm on the coding efficiency, the proposed algorithm still shows satisfactory performance.

All the experiments were performed only for  $K = 7$ . To evaluate the sensitivity of the proposed algorithm to  $K$ , we compared the results for different  $K$  values (see Fig. 7). We observed that the best performance is obtained when  $K = 7$  or 9. For some video sequences, such as *test 3* and *Station*, having large global motion, the largest  $K$  value may deteriorate the coding efficiency. Thus, we set  $K = 7$  for the previous experiments.

In addition, we compared the proposed algorithm with the affine motion compensation prediction of JEM1.0 which is downloadable from [11]. The encoding condition is same as Table 4 in this experiment. Note that affine motion compensation prediction (AMCP) is performed within the coding loop unlike the proposed algorithm. Table 6 shows that the proposed algorithm has slightly better coding efficiency than the AMCP on the same JEM platform.

**Fig. 7** Sensitivity of the proposed algorithm to  $K$ : (a) *test 1*; (b) *test 3*; (c) *test 4*; and (d) *Station*.

Finally, we examined the computational complexity of the proposed algorithm. We implemented the proposed algorithm by using C language on Intel(R) Core(TM) i7-4790 CPU @3.60 Hz, RAM 16 GB. For each 1080 p frame,

the runtime of only pre- and postfiltering amounts to about 25ms. Even though the running time of the Lucas-Kanade algorithm is computationally heavy, many fast algorithms for homography estimation are available nowadays, for instance [13].

#### 4. Concluding Remarks

In this paper, we proposed a filtering approach based on GME and GMC for pre- and postprocessing of video codecs. For the preprocessing of a video codec, GOP, which is the basic unit for GMC, and reference frames were first defined for an input video sequence. Next, GME and GMC were sequentially performed for every frame in each GOP. Finally, a block-based adaptive temporal filter was applied between the GMC frames before video encoding. For post-processing at the decoder end, every decoded frame was inversely motion-compensated using the transmitted global motion information. The holes generated during IGMC can be filled using the reference frames. The experimental results showed that the proposed algorithm provides higher BD-PSNRs of 0.63 and 0.57 dB on an average compared with conventional H.264 and HEVC platforms, respectively.

#### Acknowledgments

This work was supported by INHA UNIVERSITY Research Grant.

#### References

- [1] ISO/IEC 14496-2, Information Technology - Generic Coding of Audio-Visual Objects - Part2: Visual, 2001.
- [2] Y.G. Lee and K.H. Lee, "Improved reference frame by adopting a video stabilization technique," *IEICE Trans. Information and Systems*, vol.E97-D, no.9, pp.2545–2548, 2014.
- [3] H.S. Kim, J.H. Lee, C.K. Kim, and B.G. Kim, "Zoom motion estimation using block-based fast local area scaling," *IEEE Trans. Circuits Syst. Video Technol.*, vol.22, no.9, pp.1280–1291, 2012.
- [4] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.Y. Shum, "Full-frame video stabilization with motion inpainting," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.28, no.7, pp.1150–1163, 2006.
- [5] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: a unifying framework," *Int. J. Comput. Vis.*, vol.56, no.3, pp.221–255, 2004.
- [6] A. Krutz, A. Glantz, M. Tok, M. Esche, and T. Sikora, "Adaptive global motion temporal filtering for high efficiency video coding," *IEEE Trans. Circuits and Systems for Video Technology*, vol.22, no.12, pp.1802–1812, 2012.
- [7] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Reference Software to Committee Draft, JVT-F100 JM9.0.
- [8] G. Bjontegaard, "Calculation of average PSNR difference between RD-curves," 13th VCEG-M33 Meeting, Austin, TX, 2001.
- [9] Joint Collaborative Team on Video Coding, (JCT-VC) of ITU-T Q.6/SG 16 and ISO/IEC JTC 1/SC 29/WG 11, HM-16.7 reference software, 2015. [https://hevc.hhi.fraunhofer.de/svn/svn\\_HEVCSoftware/tags/HM-16.7](https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.7)
- [10] J. Chen, E. Alshina, G.J. Sullivan, J.R. Ohm, and J. Boyce, "Algorithm description of joint exploration test model 1," JVET-B0021, 2016.
- [11] JEM reference software, [https://jvet.hhi.fraunhofer.de/svn/svn\\_HMJEMSoftware/](https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/)

- [12] K. Suehring and X. Li, "JVET common test conditions and software reference configurations," JVET-B1010, 2016.
- [13] P. Márquez-Neila, J. López-Alberca, J.M. Buenaposada, L. Baumela, "Speeding-up homography estimation in mobile devices," *Journal of Real-Time Image Processing*, vol.11, no.1, pp.141–154, 2016.



**Ho Hyeong Ryu** received his B.S. and M.S. degrees in electronic engineering from Inha University, Incheon, Korea in 2014 and 2016. His research interests include image processing and video coding.



**Kwang Yeon Choi** received his B.S. degree in electronic engineering from Inha University, Incheon, Korea in 2015. Currently, he is pursuing a M.S. degree in electronic engineering from Inha University. His research interests include image processing and super-resolution.



**Byung Cheol Song** received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1994, 1996, and 2001, respectively. From 2001 to 2008, he was a senior engineer at Digital Media R&D Center, Samsung Electronics Co., Ltd., Suwon, Korea. In March 2008, he joined the Department of Electronic Engineering, Inha University, Incheon, Korea, and currently is an associate professor. His research interests are in

the general areas of image processing and computer vision.