Low-Complexity Watermarking into SAO Offsets for HEVC Videos

Xiangjian Wu, Hyun-Ho Jo, and Donggyu Sim

Department of Computer Engineering, Kwangwoon University / Seoul, Rep. of Korea {xjw0118, idjhh, dgsim}@kw.ac.kr

***** Corresponding Author: Donggyu Sim

Received June 28, 2016; Accepted July 27, 2016; Published August 30, 2016

** Regular Paper*

***Abstract*:** This paper proposes a new watermarking algorithm to embed watermarks in thr process of sample adaptive offsets (SAO) for high efficiency video coding (HEVC) compressed videos. The proposed method embeds two-bit watermark into the SAO offsets for each coding tree unit (CTU). To minimize visual quality degradation caused by embedding watermark, watermark bits are embedded into SAO offset depending on the SAO types of block. Furthermore, the embedded watermark can be extracted by simply adding four offsets and checking their least significant bits (LSB) at the decoder side. The experimental results show that the proposed method achieves 0.3% BD-rate increase without much visual quality degradation. Two-bit watermark for each CTU is embedded for more bit watermarking. In addition, the proposed method requires negligible computational load for watermark insertion and extraction.

***Keywords*:** Watermark, HEVC, SAO

1. Introduction

A Digital watermarking is widely used to embed information for identifying the ownership of digital content. In video content, watermark data can be embedded into the media streams. At the client side, the watermarking information can be extracted by a proper and simple detector. The watermarking techniques have been widely employed for media broadcasting and stored media applications such as Blu-ray players. The watermark data should not be detected nor easily altered. For digital watermarking, digital codes are embedded into the spatial frequency domain in general because transform domain coding is used in the most of the video coding standards such as MPEG-4, H.264/AVC, etc.

Digital video watermarking algorithms can be classified into two folds: non-blind and blind algorithms. For non-blind algorithms, original videos are required to detect embedded watermarks at a decoder side. On the other hand, watermarks can be detected without original videos by the blind algorithms. Noorkami et al. proposed the non-blind watermark embedding algorithm to detect watermarks with original videos. The type of non-blind algorithms has not been widely used [1] with limitation of original videos at the client side. Mansouri proposed a blind algorithm [2] to employ syntax elements of bitstream for generating the public key that is used to find the embedding watermark location. However, this conventional algorithm requires high computational complexity in generation of the public key. Xu et al. also proposed a watermarking method to embed watermark codes into DC coefficients; however, it yields higher bitrate increment and visual quality degradation [3, 4]. Motion vectors among video compression syntax elements are also used to embed watermark in compressed domain [5-7]. The MVs for a lager prediction block are selected to embed watermark [5]. However, alternation of motion vectors significantly influences visual quality. On the other hand, watermarking algorithm is also proposed to embed data into motion vectors for low complexity texture areas [6]. This method could reduce visual distortion in reconstructed videos. However, it requires additional computational load to classify texture regions. In addition, watermarking was proposed to add one bit data in compressed videos [14].

Recently, the high efficiency video coding (HEVC) has been standardized by JCT-VC group formed with the moving picture expert group (MPEG) of ISO/IEC and

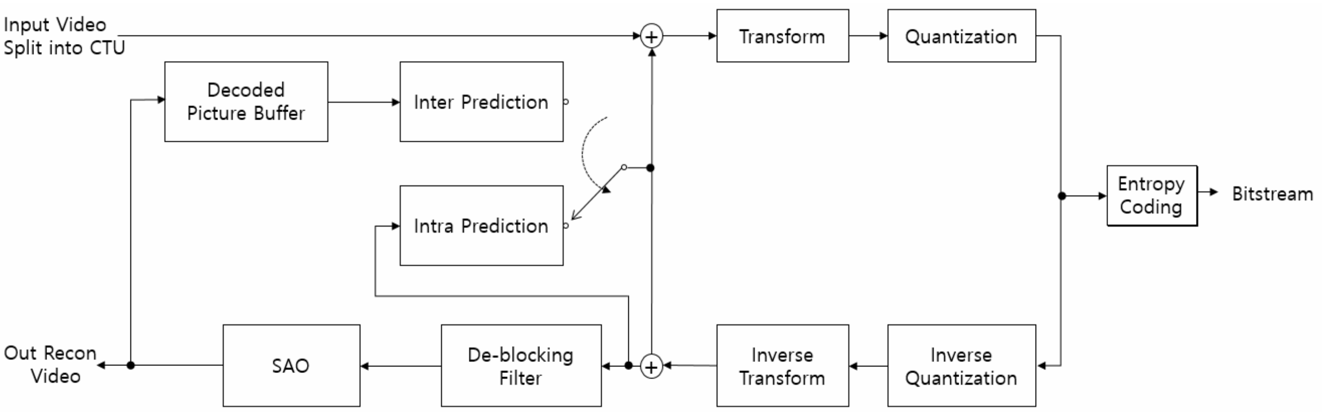

Fig. 1. HEVC block diagram.

video coding expert group (VCEG) of ITU-T as the successor of the H.264/AVC [8]. HEVC shows approximately 50% of bitrate reduction with the same visual quality, against the H.264/AVC high profile. HEVC is expected to replace the H.264/AVC and other prior video codecs for various applications due to its high coding efficiency. Since the HEVC employs the hybrid coding structure as the conventional video coding standards, digital video watermarking methods on frequency domain can be directly applied [9, 10]. However, the sample adaptive offset (SAO) is newly proposed for HEVC as one of the in-loop filters [11].

In this paper, we proposes a novel digital video watermarking algorithm that embeds additional watermarks into the SAO offsets in HEVC along with any frequency domain algorithm and others. The proposed encoder can minimize visual degradation by embedding watermark into SAO offsets. The embedded watermark can be extracted by accumulating four SAO offsets in a CTU and checking the least significant bit (LSB) at the decoder side. One bit watermark is embedded into a CTU for reducing the visual quality degradation and two-bit watermarks for each CTU are embedded for more bit watermarking. We found that the proposed algorithms yields 0.3% BD-rate increase with similar visual quality.

This paper is organized as follow. Section II presents the HEVC and its SAO algorithm in brief. In Section III, the proposed watermarking algorithm for HEVC compressed videos is illustrated in detail. Section IV shows performance evaluation of the proposed method and Section V concludes the presented work and briefly summarizes the further research topics.

2. HEVC and SAO

As mentioned before, digital watermarking is used to protect copyrights about contents [12]. It embeds information into the contents which have the ownership. Most watermarking systems have been developed under several constraints. The hidden information in the video contents should not be perceived. The embedding information process should be simple in order to avoid additional overhead to encoders. For unperceived information hiding, many watermark algorithms embed data in the frequency domain. However, the embedded data could influence visual quality and/or compression performance.

Recently, HEVC has been standardized and it starts to be deployed for many multi-media applications. HEVC is designed based on the block-based hybrid coding structure similar to conventional standards, as shown in Fig. 1. The CTU (Coding Tree Unit) is recursively split from 64x64 into 4x4 by the quad-tree structure into hierarchical CUs (Coding Unit). The large CU can be used to code spatially and temporally sample areas which have no motions or have homogeneous patterns. When a CU in complexity area can be split into multiple smaller CUs, the best CU is selected by RDO (Rate Distortion Optimization) process in the encoder side. PU (Prediction Unit) used in intra or inter prediction should be determined for each CU. TU (transform Unit) partitioning should also be decided by splitting a CU in the quad-tree structure, regardless of PU and TU partitions. For each TU, transform and quantization process are performed. The best TU partition is obtained by adaptively selecting four transform sizes based on RD optimization. HEVC employed two in-loop filters; one is de-blocking filter [13] and the other is SAO. The de-blocking filter has already been adopted in H.264/AVC, however, HEVC de-blocking filter is simpler than H.264/AVC one. The de-blocking filter is applied to each 8x8 and large TU or PU boundaries by modifying 3 pixels around the boundaries. De-blocking filter can remove visible discontinuities at the block boundaries. The discontinuities of block boundaries are caused by block-transform and prediction errors. In addition, HEVC introduces the SAO as another in-loop filters since SAO, a nonlinear amplitude mapping filter that operates on de-blocking filtered samples can effectively remove the ringing artifacts .

In SAO, each CTU is decided to be coded as no-offset, edge offset (EO), or band offset (BO). For the EO type, each CTU is classified into one of four different classes depending on edge direction, and then each pixel in the CTU is categorized into five categories. For the BO type, four bands among 32 bands are selected and then the sample values of those chosen 4 bands will be compensated with offsets. An offset value is added into all

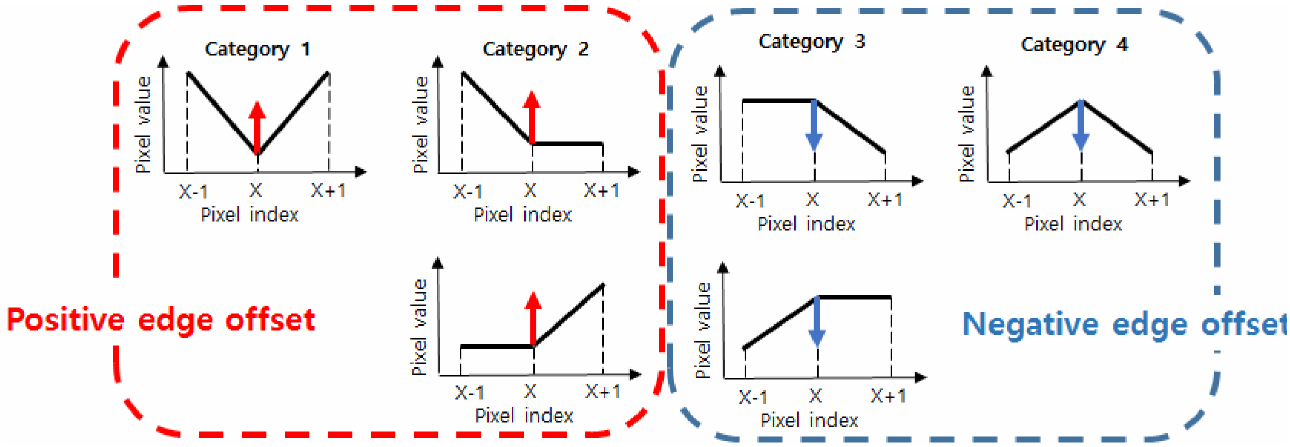

Fig. 2. Categories of SAO offset.

Table 1. Sample categorization rules for edge offset.

Category	Condition
1	c < a && c < b
2	(c < a && c == b) || (c == a && c < b)
3	(c > a && c == b) || (c == a && c > b)
4	c > a && c > b
0	None of the above

the samples in one category or band to minimize sample distortion. SAO is a useful and simple tool to minimize differences between original samples and reconstructed samples.

EO can effectively remove ringing artifacts by categorizing all the pixels and adding offsets to them. At the encoder side, one of four edge directions is selected among horizontal, vertical, 135 degree, and 45 degree sample categories. The selected class is signaled to the decoder side. After selection of a class for each CTU, HEVC encoder categorizes all the pixels inside the CTU into one of five categories depending on value of the current pixel and its two neighboring pixels, as shown in Fig. 2. Table 1 shows the condition to assign a category for each offset. Four offsets are signaled to the decoder. The decoder adds the offsets to pixels considering the class and category information. If a pixel is not included in categories 1~4, then it is classified into category 0 and then the EO is not applied. Note that the category number is derived not only at the decoder side but also at the encoder side. BO classifies all the pixels in a CTU into different bands which have fixed ranges of intensity levels. For the 8-bit depth sample, pixel intensities are divided into 32 bands. Each band has 8 intensity levels. The offsets of consecutive four bands between reconstructed samples and original samples are selected and sent to the decoder to minimize rate distortion (RD) costs. Fig. 3 shows an example that four consecutive bands are selected.

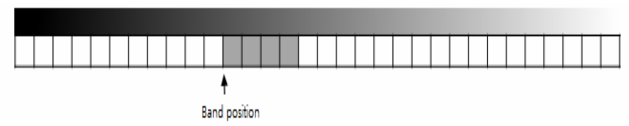

Fig. 3. Example of band offset.

3. The Proposed Watermarking

The proposed watermarking algorithm over SAO offsets is designed based on RDO process. Fig. 4 shows the block diagram of encoder side with the proposed with watermarking process embedded . The input of the proposed watermarking module is watermarking information and SAO offset values. The output of the module is SAO offsets with embedded watermarking information. When the SAO type is BO or EO at the CTU, watermark codes are embedded into the categories that are applied to the minimum number of pixels in the SAO filtering process inside each CTU. We proposed one bit watermarking algorithm into the CTU level [14]. This paper proposes an extended algorithm of two-bit watermark by minimizing the visual quality degradation and bitrate increment based on the one bit watermarking.

3.1 One-bit Watermark Embedding

In the paper [14], we proposed the algorithm of embedding one bit watermarking into the compressed videos. One bit watermark (W_b) is embedded into a CTU in which BO or EO is enabled at the encoder side. Four offsets are accumulated and the least significant bit (LSB) of the accumulation is found. If the one-bit watermark is equal to the LSB of the CTU, no process will be conducted. Otherwise, an offset to be added into the minimum number of pixels is computed during the SAO RDO process. Then, the offset is set to be near zero by adding or subtracting one. Note that C (O_m) denotes the number of pixels where the offset (O_m) is applied in a CTU. The pseudo code of this algorithm is shown as following.

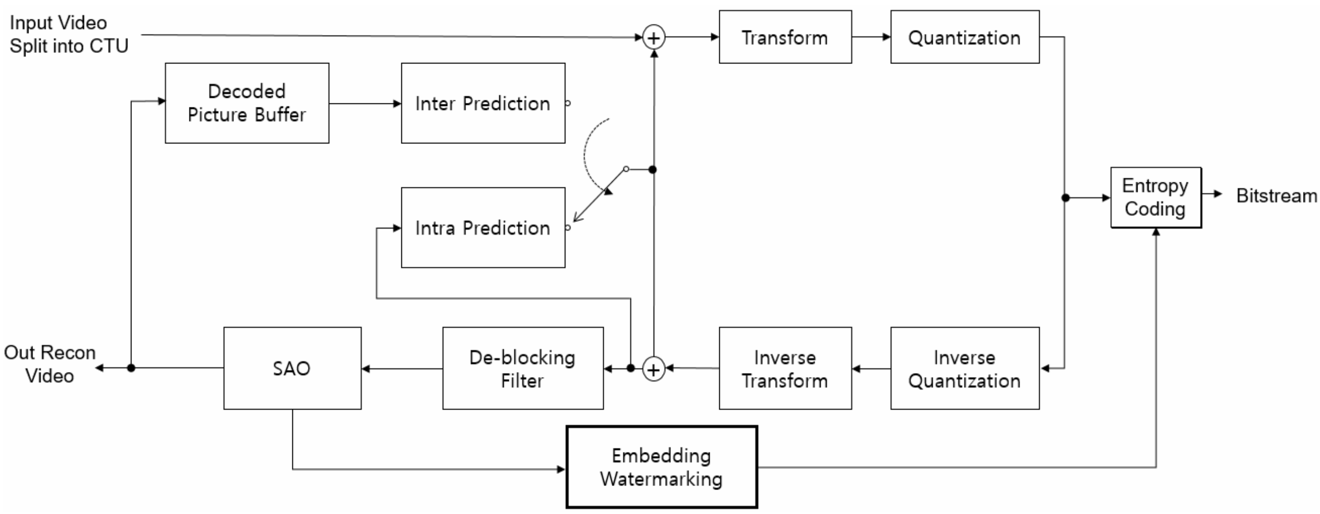

Fig. 1. Block diagram of the proposed watermarking process.

```
for all CTUs in a frame
 if BO or EO is applied then
      B = LSB(O0+O1+O2+O3)
      if B ≠ Wb then
        mc= ArgMin( C(O0), C(O1), C(O2), C(O3) )
        if Omc < 0 then
          Omc = Omc +1
      else
          Omc = Omc -1
        end if
      end if
    end if
  end for
```

At the decoder side, the one bit watermark (W_b) is extracted at the CTUs in which EO or BO is applied. The watermark bit is extracted by finding the LSB of the accumulation of the four offsets denoted by

$$W_b = LSB(\sum_{i=0}^{3} O_i) \quad (1)$$

where O_i represents the offsets of four values in a CTU.

3.2 Two-bit Watermark Embedding

The two-bit watermarks embedding algorithm embeds two -bit watermarks into a CTU in which BO or EO is enabled similar to the one-bit watermark embedding. For embedding two-bit watermarks, the encoder finds two offsets that are minimally applied in the SAO filtering process. Then, the encoder embeds one-bit watermark into the LSB of each offset. The pseudo code of two-bit watermarks embedding algorithm is shown below. Note that W_{b1} and W_{b2} are watermark bits. O_{m1} and O_{m2} are the offsets that will be applied minimally for a CTU.

```
for all CTUs in a frame
 if BO or EO is applied then
  mc1= ArgMin( C(O0), C(O1), C(O2), C(O3) )
  mc2= ArgMin( C(O0), C(O1), C(O2), C(O3) )
     if LSB(Om1) ≠ Wb1 then
    if Om1 < 0 then
            Om1 = Om1 +1
    else
            Om1 = Om1 -1
    end if
     end if
   if LSB(Om2) ≠ Wb2 then
     if Om2 < 0 then
            Om2 = Om2 +1
     else
            Om2 = Om2 -1
   end if
     end if
   end if
 end for
```

At the decoder side, the four offsets are sorted in the number of pixels that each offset is applied when SAO type is EO or BO for a CTU. Then, the decoder regards the LSB of the first and second, in the ascending order, offsets as to be watermark bits respectively.

The two-bit watermark embedding algorithm can embed more bits, compared to the one-bit watermark embedding algorithm by a factor of two. However, insertion and extraction process of the watermark will be more complicated. In addition, the pixel reconstruction should be prior to extraction of the watermark because reconstructed samples are required to sort the four offsets at the decoder side. On the other hand, the one-bit watermark embedding algorithm can extract the watermark bit only with entropy decoding for the syntax elements of the SAO offsets. One-bit embedding algorithm rarely gives

Table 2. BD-rate increment for embedding one bit watermarking.

		AI	RA	LD
Class A	Traffic	0.4	0.4	0.5
	PeopleOnStreet	0.3	0.3	0.3
Class B	Kimono	0.4	0.4	0.4
	ParkScene	0.2	0.5	0.3
	Cactus	0.5	0.5	0.5
	BasketballDrive	0.6	0.4	0.4
	BQTerrace	0.1	0.5	0.5
Average		0.35	0.42	0.35

Table 3. Average number of bits which are embedded for one bit watermarking with random access condition.

		22	27	32	37
Class A	Traffic	160	72	33	8
	PeopleOnStreet	404	124	47	14
Class B	Kimono	114	30	8	3
	ParkScene	106	43	20	6
	Cactus	162	47	20	9
	BasketballDrive	183	60	22	8
	BQTerrace	203	62	25	11

Table 4. BD-Rate increment for embedding two bits watermarking.

		AI	RA	LD
Class A	Traffic	0.6	1.0	1.1
	PeopleOnStreet	0.4	0.9	0.6
Class B	Kimono	0.6	0.8	0.5
	ParkScene	0.4	0.7	0.6
	Cactus	0.6	1.1	08
	BasketballDrive	0.8	0.9	1.0
	BQTerrace	0.4	0.8	0.6
Average		0.5	0.8	0.7

Table 5. Average number of bits which are embedded for two bits watermarking with random access condition.

		22	27	32	37
Class A	Traffic	304	170	80	20
	PeopleOnStreet	756	285	109	31
Class B	Kimono	185	67	26	4
	ParkScene	205	99	49	13
	Cactus	418	122	51	19
	BasketballDrive	368	125	51	18
	BQTerrace	467	143	56	24

visual quality degradation than two-bit embedding algorithm. Two algorithms can be selected with the application's requirements. The proposed algorithm can embed additional data into SAO domain instead of frequency domain or other syntax elements such as motion vector. The proposed algorithm does not influence other watermark algorithms.

4. Performance Evaluation

To evaluate the proposed two-bit algorithm, the one-bit and two-bit watermark were implemented at the HM 12.0 reference software. Two sequences from Class A and five sequences from Class B under the common test condition (CTC) were employed in the experiments. The bitrate increment, visual quality distortion, and the number of watermark bits of the proposed algorithms are evaluated on the CTC conditions [15]. The embedded information is a bit sequence consisted of 900 bits. The bits are sequentially embedded into compressed bitstream. The RD performance and average embedded bits are evaluated..

Table 2 shows the BD-rate performance for Class A and Class B sequences with the one-one bit watermark embedding algorithms. As shown in Table 2, the one-bit watermark embedding algorithm achieves about 0.37% bitrate increment. At Table 3, the average number of bits were embedded into a frame of a sequence. The bitrate increment is negligible for practical applications.

Table 4 shows the BD-rate performance of the proposed two-bit watermark embedding algorithm. The average bit-rate increment is 0.82% and the average embedded watermark bits is shown Table 5. It shows that only a few number of watermark bits are encoded at the high quantization parameter (QP) range. because The reason is that the enabled SAO ratio decreases at the high QP ranges. Two-bit watermark algorithm shows more bitrate increment against to the one-bit watermark algorithm.

To compare the visual quality, two parts of 'BQTerrace' sequence which are cropped are showed with the proposed watermark and no-watermark algorithms. Fig. 6 shows cropped parts of the 3th frame from 'BQTerrace' sequence with one bit watermark in each CTU and without watermark for comparison of the subjective quality. Average 203 bits watermarks per frame are embedded for (a) using the one-bit watermark algorithm. It shows a quite similar visual quality with Fig. 6. (b), even though the watermarks are embedded into the SAO offset.

Fig. 7 shows a part of the first frame which is coded by the random access mode. Fig. 7(a) shows the reconstructed image having two-bit watermark for each CTU. Fig. 7(b) shows the one without any watermark bits. Even though 467 bits are embedded for the frame, we cannot recognize any distortion for both frames.

Regarding of the evaluation of SAO on/off performance [11], 2.5% degradation in BD-rate was observed for SAO off under the random access condition. The proposed algorithm changes one or two bits that minimize visual quality degradation for embedding watermark bits. We found that as small as 0.3% degradation is observed.

(a) (b)

Fig. 6. Part of BQTerrace for the 3th frame (a) with one-bit watermark in each CTU, (b) without any watermarks.

(a) (b)

Fig. 7. Parts of BasketBall Drive for the 8th frame (a) with two-bit watermark in each CTU, (b) without watermark.

5. Conclusion

This paper proposes two-bit watermarking algorithm in SAO domain for the HEVC compressed videos. The proposed method embeds two bits into a CTU in which SAO is applied. The proposed algorithm does not require much computation both in the encoder and decoder when embedding and extracting the watermarks. Furthermore, the proposed watermark can be extracted without the pixel decoding at the decoder side. The experimental results show that the proposed method has a small amount of bitrate increase while does not lead to any visual degradation. For the further work, we will focus on a watermark encoder into SAO offset with rate distortion optimization (RDO) process.

Acknowledgement

This research was partly supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(NRF-2014R1A2A1A11052210) and the MSIP(Ministry of Science, ICT&Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2016-H8501-16-1005) supervised by the IITP(Institute for Information&communications Technology Promotion)

References

[1] M. Noorkami, et al, "A framework for robust watermarking of H.264-encoded video with controllable detection performance," IEEE trans. On Information Forensics Security 2007, pp 14-23 Article (CrossRef Link)
[2] A. Mansouri, et al, "A low complexity video watermarking in H.264 compressed domain," IEEE Trans. On Information Forensics and Security 2010, pp. 649-657 Article (CrossRef Link)
[3] D. Xu, et al, "Blind digital watermarking of low bitrate advanced H.264/AVC compressed video," Digital watermarking 2009, pp. 96-109 Article (CrossRef Link)
[4] S. Joo, et al, "A new robust watermark embedding into wavelet dc components," ETRI Journal 2002, pp. 401- 404 Article (CrossRef Link)
[5] N. Mohaghegh, et al, "H.264 copyright protection with motion vector watermarking," Proceedings of the International Conference on Audio, Language and Image Processing 2008 Article (CrossRef Link)
[6] Z. liu, et al, "A robust video watermarking in motion vectors," Proceedings of the International Conference on Signal Processing 2004 Article (CrossRef Link)
[7] S. Swati, et al, "A watermarking scheme for high efficiency video coding," Plos 2014 Article (CrossRef Link)
[8] G. Sullivan, at al, "Overview of the high efficiency video coding(HEVC) standard," IEEE trans. On Circuits and Systems for Video Technology 2012, pp 1649- 1668 Article (CrossRef Link)
[9] K. Kim, et al, "Practical real time, and robust watermarking on the spatial domain for high-definition video contents," IEICE trans, INF. &System 2008 Article (CrossRef Link)
[10] D. Xu, et al, "Data hiding in encrypted H.264/AVC video streams by code word substitution," IEEE Trans. On Information Forensics and Security 2014, pp. 596-606 Article (CrossRef Link)
[11] C. Fu, et al, "Sample adaptive offset in the HEVC standard," IEEE Trans. On Circuits and Systems for Video Technology 2012, pp. 1755-1764 Article (CrossRef Link)
[12] J. Ingemar, et al, "Digital Watermarking and Steganography," 2007 Article (CrossRef Link)
[13] N. Andrey, et al, "HEVC Deblocking Filter," IEEE Trans. On Circuits and System for Video Technology 2012 Article (CrossRef Link)
[14] X. Wu, H. Jo. D. Sim, "Embedding one bit watermarking information in SAO offsets for HEVC compressed video," International Workshop on Advanced Image Technology(IWAIT) 2016 Article (CrossRef Link)
[15] F. Bossen, et al, "Common HM Test Conditions and Software Reference Con-figurations," ITU-T/ISO/ICE Joint Collaborative Team on Video Coding(JCT-VC) document JCTVC-J1100 2012 Article (CrossRef Link)

Xiangjian Wu received the B.S. degree in Computer Engineering from Kwangwoon University (KWU), Seoul, Korea, in 2014, respectively. His research interests are next-generation video coding, parallel processing for video coding.

Hyun-Ho Jo was born in Jeonnam Province, Korea, in 1982. He received the B.S., M.S., and Ph. D. degree in Computer Engineering from Kwang-woon University (KWU), Seoul, Korea, in 2008, 2010, and 2014, respectively. His current research interests are codec optimization, parallel processing, and video coding.

Donggyu Sim received the B.S. and M.S. degrees in Electronic Engi-neering from Sogang University, Korea, in 1993 and 1995, respectively. He also received the Ph.D. degree at the same University, in 1999. He was with the Hyundai Electronics Co., Ltd. From 1999 to 2000, being involved in MPEG-7 standardization. He was a senior research engineer at Varo Vision Co., Ltd, working on MPEG-4 wireless applications from 2000 to 2002. He worked for the Image Computing Systems Lab. (ICSL) at the University of Washington as a senior research engineer from 2002 to 2005. He researched on ultrasound image analysis and parametric video coding. Since 2005, he has been with the Department of Computer Engineering at Kwangwoon University (KWU), Seoul, Korea. In 2011, he joined the Simon Frasier University, as a visiting scholar. He was elevated to an IEEE Senior Member in 2004. He is one of main inventors in many essential patents licensed to MPEG-LA for HEVC standard. His current research interests are video coding, video processing, computer vision, and video communication.