

Sample Adaptive Offset in the HEVC Standard

Chih-Ming Fu, Elena Alshina, Alexander Alshin, Yu-Wen Huang, Ching-Yeh Chen, and Chia-Yang Tsai, *Member, IEEE*, Chih-Wei Hsu, Shaw-Min Lei, *Fellow, IEEE*, Jeong-Hoon Park, and Woo-Jin Han, *Member, IEEE*

Abstract—This paper provides a technical overview of a newly added in-loop filtering technique, sample adaptive offset (SAO), in High Efficiency Video Coding (HEVC). The key idea of SAO is to reduce sample distortion by first classifying reconstructed samples into different categories, obtaining an offset for each category, and then adding the offset to each sample of the category. The offset of each category is properly calculated at the encoder and explicitly signaled to the decoder for reducing sample distortion effectively, while the classification of each sample is performed at both the encoder and the decoder for saving side information significantly. To achieve low latency of only one coding tree unit (CTU), a CTU-based syntax design is specified to adapt SAO parameters for each CTU. A CTU-based optimization algorithm can be used to derive SAO parameters of each CTU, and the SAO parameters of the CTU are inter-leaved into the slice data. It is reported that SAO achieves on average 3.5% BD-rate reduction and up to 23.5% BD-rate reduction with less than 1% encoding time increase and about 2.5% decoding time increase under common test conditions of HEVC reference software version 8.0.

Index Terms—Advanced video coding (AVC), band offset, edge offset, H.264, High Efficiency Video Coding (HEVC), Joint Collaborative Team on Video Coding (JCT-VC), sample adaptive offset (SAO), video standard.

I. INTRODUCTION

RECENTLY High Efficiency Video Coding (HEVC) [1] is under development by the Joint Collaborative Team on Video Coding (JCT-VC), which is established by ISO/IEC Moving Picture Experts Group (MPEG) and ITU-T Video Coding Experts Group (VCEG). The target of HEVC is 50% bit rate reduction in comparison with advanced video coding (AVC), or namely H.264 [2], under the same visual quality.

Block-based intra/inter prediction and transform coding are still applied in HEVC. Therefore, artifacts that are commonly

Manuscript received April 21, 2012; revised July 28, 2012; accepted August 21, 2012. Date of publication October 5, 2012; date of current version January 8, 2013. This paper was recommended by Associate Editor J. Ridge. (*Corresponding author: W.-J. Han*)

C.-M. Fu, Y.-W. Huang, C.-Y. Chen, C.-Y. Tsai, C.-W. Hsu, and S.-M. Lei are with MediaTek, Hsinchu 30078, Taiwan (e-mail: chihming.fu@mediatek.com; yuwen.huang@mediatek.com; chingyeh.chen@mediatek.com; chiayang.tsai@mediatek.com; cw.hsu@mediatek.com; shawmin.lei@mediatek.com).

E. Alshina, A. Alshin, and J. Park are with the Digital Media and Communication Research and Development Center, Samsung Electronics, Suwon 443-742, Korea (e-mail: elena_a.alshina@samsung.com; alexander_b.alshin@samsung.com; jeonghoon@samsung.com).

W.-J. Han is with Gachon University, Seongnam 461-701, Korea (e-mail: hurumi@gmail.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSVT.2012.2221529

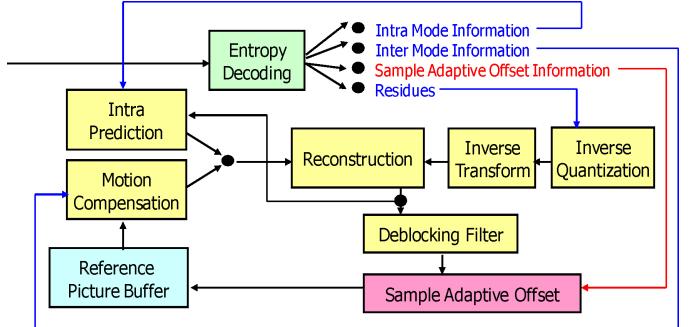


Fig. 1. Block diagram of HEVC decoder.

seen in prior video coding standards at medium and low bit rates, such as blocking artifacts, ringing artifacts, color biases, and blurring artifacts [3], may still exist in HEVC. In order to reduce these artifacts, HEVC also adopts in-loop filtering as in AVC.

HEVC uses various transforms ranging from 4×4 to 32×32 , while AVC uses transforms no larger than 8×8 . A larger transform could introduce more artifacts [4] including ringing artifacts that mainly come from quantization errors of transform coefficients. Besides, HEVC uses 8-tap fractional luma sample interpolation and 4-tap fractional chroma sample interpolation, while AVC uses 6-tap and 2-tap for luma and chroma, respectively. A higher number of interpolation taps can also lead to more serious ringing artifacts. Hence, it is necessary to incorporate new in-loop filtering techniques in HEVC. In addition to developing a new deblocking filter (DF), HEVC further introduces a completely new stage: sample adaptive offset (SAO). As shown in Fig. 1, SAO is located after DF and also belongs to in-loop filtering. The concept of SAO is to reduce mean sample distortion of a region by first classifying the region samples into multiple categories with a selected classifier, obtaining an offset for each category, and then adding the offset to each sample of the category, where the classifier index and the offsets of the region are coded in the bitstream. Please note that a customized SAO encoder does not necessarily attempt to minimize mean sample distortion but can use another criterion to generate SAO parameters. Please also note that SAO not only is useful in HEVC but also can be applied on top of AVC and other prior video coding standards.

The rest of this paper is organized as follows. Section II presents the evolutions, motivations, and challenges of SAO, Section III introduces two sample processing techniques, edge offset (EO) and band offset (BO), and Section IV describes

the SAO syntax design. Next, implementation issues are discussed in Section V, then experimental results are shown in Section VI, and finally conclusions are provided in Section VII.

II. EVOLUTIONS, MOTIVATIONS, AND CHALLENGES

In Windows Media Video 9 (WMV9) [5], a deringing filter can be applied for reducing ringing artifacts. First, samples are classified into two categories: edge and nonedge. Then, edge samples and nonedge samples are filtered differently according to their categories. The deringing filter in WMV9 is a post-processing technique applied to decoded pictures, which might lead to serious flickering artifacts from picture to picture.

In VCEG-AL27 [6], an adaptive loop filtering (ALF) method was proposed. Reconstructed samples are first classified into different categories by using values of sum-modified Laplacian, and then both filtering and offsets are applied to classified samples, where each category corresponds to one filter and one offset. It also shows that in-loop filtering can provide better coding efficiency than post filtering. This method achieves good coding gains, but the complexity of sample classification and filtering might be a concern.

In JCTVC-A124 [7], an effective filter, extreme correction (EXC), was proposed and is located between DF and ALF. The EXC can recognize extreme samples by a fixed classifier comparing each sample with its four neighbors, and thus the decoder can correct those extreme samples without explicitly signaling the extreme sample positions. Another filter, Band Correction (BDC), was also proposed in JCTVC-A124 [7], and it is located after ALF. The BDC can recognize which band each sample belongs to by the sample value, and thus the decoder can correct band samples without explicitly signaling the sample positions. The EXC and BDC both use picture-based approaches, which obtain filter parameters after scanning the entire picture, and are applied as sequential stages, which can lead to unacceptably long latency on the encoder side. Moreover, the coding gains of EXC and BDC are not good enough to justify their complexities.

In JCTVC-B077 [8], more picture-based processing stages were studied. After DF, a combined stage of filtering and offsets with picture quadtree partitioning is used. Next stages are picture-based band offset (PBO), picture-based edge offset (PEO), and picture-based adaptive clipping (PAC), where the PBO improves BDC by allowing both uniform and nonuniform band classifications and the PEO improves EXC by allowing two different classifier patterns of neighboring samples. The results show that the coding gains of adding more stages seem unable to justify the increased encoding latency.

In order to reduce the encoding latency, JCTVC-C147 [9] and JCTVC-D122 [10] proposed combining sequential stages of adaptive offset techniques into one stage and allowing the encoder to select only one classifier for each region adaptively, where the regions can be picture quadtree partitions. However, the processing of each sample was still too complex.

In JCTVC-E049 [11], significant complexity reduction was achieved by further simplifying the sample classification used in JCTVC-D122 [10], and the proposed coding tool was

renamed as sample adaptive offset (SAO). Both the number of classifier options and the number of classifier operations are decreased. The features of the SAO in JCTVC-E049 [11] are summarized as follows. First, it allows local adaptation by using a region-based approach to obtain good coding efficiency. Second, sequential stages are combined into one stage by selecting only one classifier per region to reduce the encoding latency. Next, 2-D edge classification patterns are removed, and only four 1-D edge classification patterns are used. Moreover, only offsets of partial bands instead of all bands are signaled to reduce the side information. Last, fast distortion estimation is applied during rate distortion optimization to reduce memory access and distortion calculations. With these features, SAO could achieve good coding efficiency with reasonable complexity. Hence, in the 5th JCT-VC meeting, the SAO was adopted into the working draft of HEVC.

After JCTVC-E049 [11], SAO kept evolving to its current shape. Changes are briefly described as follows: SAO can be applied to not only luma but also chroma, the picture quadtree-based adaptation is replaced with a coding tree unit (CTU) based adaptation to further reduce the encoding latency to only one CTU, the syntax and binarization are cleaned up to save context coded bins in context-based adaptive binary arithmetic coding (CABAC), and the encoding algorithm is improved especially for small CTU sizes. Please note that one CTU has a luma coding tree block (CTB) and two chroma CTBs. More details of the SAO design will be shown in later sections.

From the previous paragraphs, the motivations of SAO can be restated as follows. The SAO tries to reduce undesirable visual artifacts including ringing artifacts that could become more serious with large transforms and longer tap inter polarizations. The SAO also attempts to reduce the mean distortion between original samples and reconstructed samples by first classifying reconstructed samples into different categories, obtaining an offset for each category, and then adding the offset to each sample of the category without signaling the locations of to-be-corrected samples.

From the complexity point of view, the sample classification has to be very simple because it is required at both the encoder and the decoder. In addition, SAO should be applied only for reconstructed samples where the compression causes systematic distortions due to transform, quantization or prediction artifacts. These are the most challenging parts of the SAO design. During the SAO development, the edges and specific bands of samples were identified to be simple enough for sample classification and likely have nonzero average differences between original samples and reconstructed samples. Moreover, they are crucial for improving visual quality.

III. SAMPLE PROCESSING

SAO may use different offsets sample by sample in a region depending on the sample classification, and SAO parameters are adapted from region to region. Two SAO types that can satisfy the requirements of low complexity are adopted in HEVC: edge offset (EO) and band offset (BO). For EO, the sample classification is based on comparison between current samples and neighboring samples. For BO, the sample

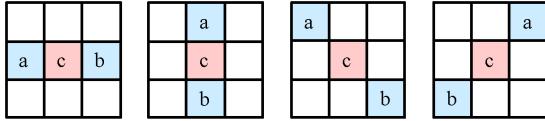


Fig. 2. Four 1-D directional patterns for EO sample classification: horizontal (EO class = 0), vertical (EO class = 1), 135° diagonal (EO class = 2), and 45° diagonal (EO class = 3).

TABLE I
SAMPLE CLASSIFICATION 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

classification is based on sample values. Please note that each color component may have its own SAO parameters [12]. To achieve low encoding latency and to reduce the buffer requirement, the region size is fixed to one CTB. To reduce side information, multiple CTUs can be merged together to share SAO parameters [13].

A. Edge Offset

In order to keep reasonable balance between complexity and coding efficiency, EO uses four 1-D directional patterns for sample classification: horizontal, vertical, 135° diagonal, and 45° diagonal, as shown in Fig. 2, where the label “c” represents a current sample and the labels “a” and “b” represent two neighboring samples. According to these patterns, four EO classes are specified, and each EO class corresponds to one pattern. On the encoder side, only one EO class can be selected for each CTB that enables EO. Based on rate-distortion optimization, the best EO class is sent in the bitstream as side information. Since the patterns are 1-D, the results of the classifier do not exactly correspond to extreme samples.

For a given EO class, each sample inside the CTB is classified into one of five categories. The current sample value, labeled as “c,” is compared with its two neighbors along the selected 1-D pattern. The classification rules for each sample are summarized in Table I. Categories 1 and 4 are associated with a local valley and a local peak along the selected 1-D pattern, respectively. Categories 2 and 3 are associated with concave and convex corners along the selected 1-D pattern, respectively. If the current sample does not belong to EO categories 1–4, then it is category 0 and SAO is not applied.

The meanings of edge offset signs are illustrated in Fig. 3 and explained as follows. Positive offsets for categories 1 and 2 result in smoothing since local valleys and concave corners become smoother, while negative offsets for these categories result in sharpening. On the other hand, the meanings are opposite for categories 3 and 4, where negative offsets result in smoothing and positive offsets result in sharpening. From the statistical analyses [14]–[16], the EO in HEVC disallows sharpening and sends absolute values of offsets, while signs of offsets are implicitly derived according to EO categories.

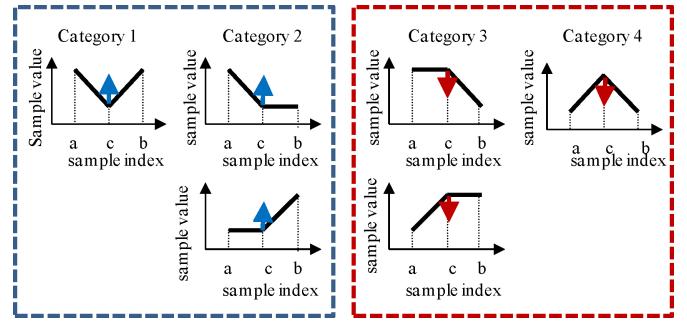


Fig. 3. Positive offsets for EO categories 1 and 2 and negative offsets for EO categories 3 and 4 result in smoothing.

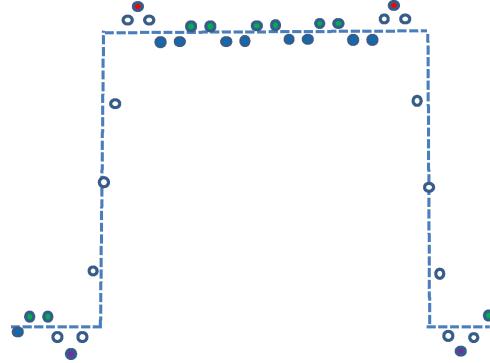


Fig. 4. Gibbs phenomenon where the dotted curve is the original samples and the solid curve is the reconstructed samples.

Fig. 4 shows the well-known Gibbs phenomenon in many signal processing textbooks and can be used to simulate a few video compression artifacts, especially the ringing artifacts. The horizontal axis and the vertical axis are not explicitly shown but are used to denote the sample position along a 1-D path and the sample value, respectively. The dotted curve represents the original samples, while the solid curve represents the reconstructed samples by discarding high frequencies of the original samples. The local peaks, convex corners, concave corners, and local valleys are painted as solid circles, while non-of-the-above samples are painted as empty circles. It can be seen that if we apply negative offsets to local peaks and convex corners, and positive offsets to concave corners and valleys, the distortion can be effectively reduced.

B. Band Offset

BO implies one offset is added to all samples of the same band. The sample value range is equally divided into 32 bands. For 8-bit samples ranging from 0 to 255, the width of a band is 8, and sample values from $8k$ to $8k + 7$ belong to band k , where k ranges from 0 to 31. The average difference between the original samples and reconstructed samples in a band (i.e., offset of a band) is signaled to the decoder. There is no constraint on offset signs.

Only offsets of four consecutive bands and the starting band position are signaled to the decoder [17], [18]. The number of signaled offsets in BO was decided to be reduced from 16 to 4 and is now equal to the number of signaled offsets in EO for reducing the line buffer requirement, which will be described

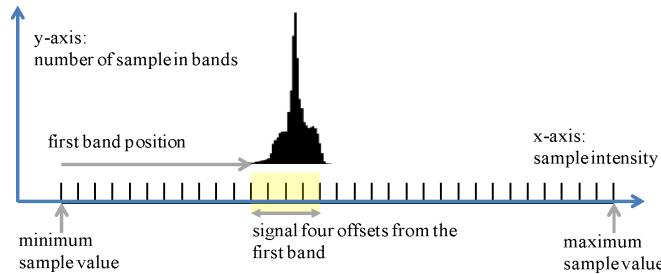


Fig. 5. Example of sample distribution in a CTB, where BO send the offsets of four consecutive bands.

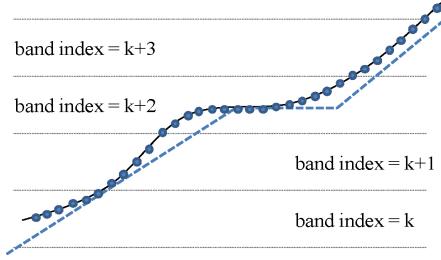


Fig. 6. Example of BO, where the dotted curve is the original samples and the solid curve is the reconstructed samples.

in Section V-C. Another reason of selecting only four bands is that the sample range in a region can be quite limited after the regions are reduced from picture quadtree partitions to CTBs, especially for chroma CTBs, as an example shown in Fig. 5.

Fig. 6 can be used to explain why BO works in a few circumstances. The horizontal axis and the vertical axis are not explicitly shown but are used to denote the sample position and the sample value, respectively. The dotted curve is the original samples, while the solid curve is the reconstructed samples, which might be corrupted by quantization errors of prediction residues and phase shifts due to coded motion vectors deviating from the true motions. In this example, the reconstructed samples are shifted to the left of the original samples, which systematically results in negative errors that can be corrected by BO for bands k , $k + 1$, $k + 2$, and $k + 3$.

IV. SYNTAX DESIGN

In the sequence parameter set (SPS), one syntax element, `sample_adaptive_offset_enabled_flag`, is used to indicate whether SAO is enabled in the current video sequence. In the slice header, two syntax elements, `slice_sao_luma_flag` and `slice_sao_chroma_flag`, indicate whether SAO is enabled for luma and chroma, respectively, in the current slice.

The current SAO encoding algorithm can be configured as CTU-based for low-delay applications. Syntax-wise, the basic unit for adapting SAO parameters is always one CTU. If SAO is enabled in the current slice, SAO parameters of CTUs are inter-leaved into the slice data. The SAO data of one CTU is placed at the beginning of the CTU in the bitstream. The CTU-level SAO parameters contain SAO merging information, SAO type information, and offset information.

A. SAO Merging

For each CTU, there are three options: reusing SAO parameters of the left CTU by setting the syntax element `sao_merge_left_flag` to true, reusing SAO parameters of the above CTU by setting the syntax element `sao_merge_up_flag` to true, or transmitting new SAO parameters. Please note that the SAO merging information is shared by three color components. As shown in Fig. 7, a CTU includes the corresponding luma CTB, Cb CTB, and Cr CTB. When the current CTU selects SAO merge-left or SAO merge-up, all SAO parameters of the left or above CTU are copied for reuse, so no more information is sent. This CTU-based SAO information sharing [19] can reduce side information effectively.

B. SAO Type and Offsets

If a current CTU does not merge with a neighboring CTU, the rest SAO information of the current CTU will be signaled, as shown in Fig. 8. All luma syntax elements are first sent and followed by all Cb syntax elements and then all Cr syntax elements. For each color component, the SAO type is first transmitted (`sao_type_idx_luma` or `sao_type_idx_chroma`) to indicate EO, BO, or OFF. If BO is selected, the starting band position (`sao_band_position`) is signaled; else if EO is selected, the EO class (`sao_eo_class_luma` or `sao_eo_class_chroma`) is signaled. For both BO and EO, four offsets are transmitted. Please note that Cb and Cr share the SAO type (`sao_type_idx_chroma`) and EO class (`sao_eo_class_chroma`) to further reduce the side information and to speed up SAO by achieving more efficient memory access in certain platforms [20], so these syntax elements are coded for Cb only and do not have to be sent for Cr.

C. Context Bins and Bypass Bins

All the CTU-level SAO syntax elements including SAO merging information, SAO type information, and offset information are coded by CABAC. Only the first bin of the SAO type, which specifies SAO on/off in the current CTU, and the SAO merge-left/up flags are context coded. The rest bins are bypass coded for significantly increasing the SAO parsing throughput in CABAC [21]–[25].

V. IMPLEMENTATION

This section discusses six implementation issues. The first three issues are for both encoders and decoders, while the rest are encoder-only issues. Effective solutions are also provided.

A. Fast Edge Offset Sample Classification

Although the sample classification rules of EO in Table I seem nontrivial, Table I is mainly used for easy explanation of the concepts, and the EO sample classification does not need to be implemented in that way. A fast algorithm can be described in the following equations:

$$\text{sign3}(x) = (x > 0)?(+1) : ((x == 0)?(0) : (-1)) \quad (1)$$

$$\text{edgeIdx} = 2 + \text{sign3}(c - a) + \text{sign3}(c - b) \quad (2)$$

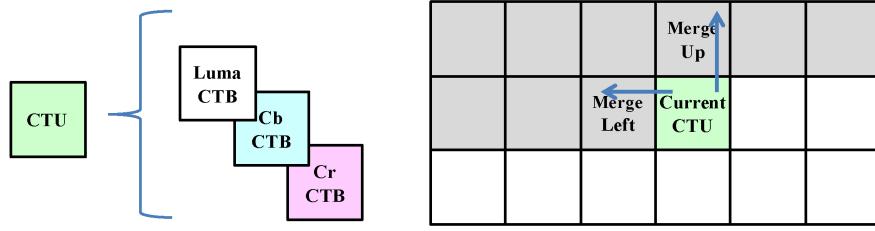


Fig. 7. CTU consists of CTBs of three color components, and the current CTU can reuse SAO parameters of the left or above CTU.

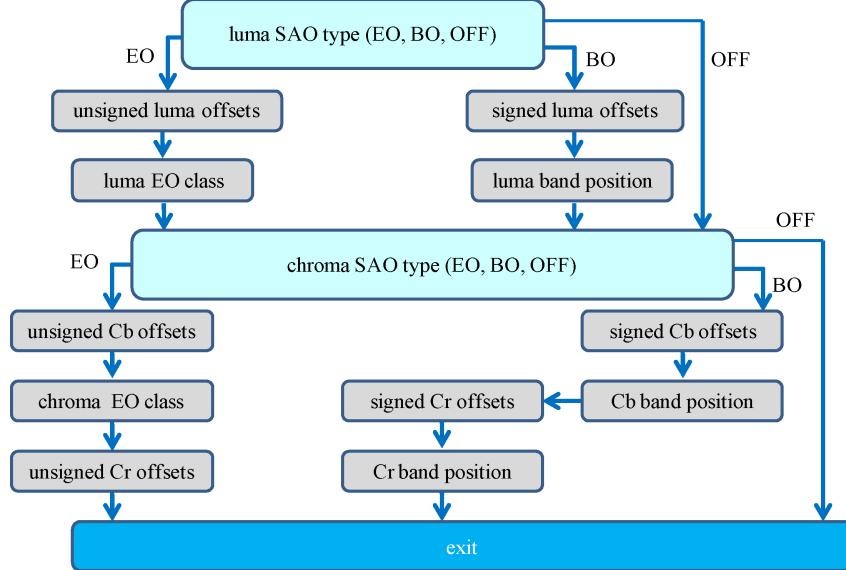


Fig. 8. Illustration of coding the rest CTU-level SAO information when the current CTU is not merged with the left or above CTU.

$$\text{edgeIdx2category}[] = \{1, 2, 0, 3, 4\} \quad (3)$$

$$\text{category} = \text{edgeIdx2category}[\text{edgeIdx}]. \quad (4)$$

In (2), c is the current sample, and a and b are the two neighboring samples, as shown in Fig. 2 and Table I. Besides the fast calculations in (1)–(4), data reuse between samples can be further applied for the next sample classification. For example, assuming the EO class is 0 (i.e., using 1-D horizontal pattern) and the samples in the CTB are processed in the raster scan order, the $\text{sign3}(c-a)$ of the current sample does not have to be calculated and can be directly set to the $-\text{sign3}(c-b)$ of the neighboring sample to the left. Likewise, the $\text{sign3}(c-b)$ of the current sample can be reused by the neighboring sample to the right.

B. Fast Band Offset Sample Classification

The sample range is equally divided into 32 bands in BO. Since 32 is equal to two to the power of five, the BO sample classification can be implemented as using the five most significant bits of each sample as the classification result. In this way, the complexity of BO becomes very low, especially in hardware that only needs wire connections without logic gates to obtain the classification result from the sample value.

C. Line Buffer Requirement

CTU-based processing is commonly adopted in practical implementations. CTUs are encoded or decoded one by one in raster scan order. In DF, vertical filtering across a horizontal boundary requires to read four luma samples, two Cb samples, and two Cr samples and write three luma samples, one Cb sample, and one Cr sample on both sides of the boundary. Hence, when a current CTU is being processed, DF has not finished processing the bottom sample rows of the above CTU. Since SAO is after DF, the bottom sample rows of the above CTU have not been processed by SAO either. In order to apply CTU-based processing, DF and SAO need line buffers.

Let us assume that DF uses N sample line buffers to store horizontally deblocked samples of the bottom N rows in the above CTB, where N is four for luma and two for Cb and Cr. Since the N th row above the horizontal CTB boundary will not be modified by the vertical deblocking, the SAO processes for the $(N+1)$ th row above the horizontal CTB boundary can be done before the current CTB comes. However, the bottom N rows of the above CTB still have to wait for the current CTB for applying DF and SAO. When the current CTB comes and if the above CTB selects EO with the EO class larger than zero, the SAO processes for the N th row above the boundary needs to use the $(N+1)$ th row above the boundary. Intuitively, we can store samples of the $(N+1)$ th row in the SAO line buffer.

Fortunately, with the introduced fast EO sample classification, we can store the “sign3” results of the N th row and the $(N+1)$ th row instead, which needs only two bits per sample.

In addition to the “sign3” line buffer, line buffers to store the SAO type information and offsets of the upper CTB row are also required. Since these pieces of information are at the CTB-level instead of sample level, the line buffer size is acceptable. For 8-bit 4:2:0 videos, the line buffer size per smallest CTU is 99 bits, where the smallest CTU is 8×8 and 4×4 in luma and chroma, respectively. Among the 99 bits, 32 bits are from “sign3” values of eight luma and chroma samples, two bits are from luma SAO type, two bits are from chroma SAO type, 15 bits are from starting band positions or EO classes, and 48 bits are from luma and chroma offsets. The SAO line buffer size in total is smaller than 3K bytes for full HD (i.e., 1920×1080) videos.

D. Fast Distortion Estimation

During the rate-distortion optimization [26] on the encoder side, distortions between original samples and reconstructed samples have to be calculated for many times. A straightforward implementation for SAO would need to add offsets to pre-SAO samples to generate post-SAO samples and then calculate the distortion between original samples and post-SAO samples. To reduce the memory access and operations, a fast distortion estimation method [27] can be implemented as follows. Let k , $s(k)$, and $x(k)$ be sample positions, original samples, and pre-SAO samples, respectively, where k belongs to C and C is the set of samples that are inside a CTB and belong to a specified SAO type (i.e., BO or EO), a specified starting band position or EO class, and a specified band or category. The distortion between original samples and pre-SAO samples can be described in the following equation:

$$D_{\text{pre}} = \sum_{k \in C} (s(k) - x(k))^2. \quad (5)$$

The distortion between original samples and post-SAO samples can be described in the following equation:

$$D_{\text{post}} = \sum_{k \in C} (s(k) - (x(k) + h))^2. \quad (6)$$

In (6), h is the offset for the sample set. The delta distortion is defined in the following equation:

$$\Delta D = D_{\text{post}} - D_{\text{pre}} = \sum_{k \in C} (h^2 - 2h(s(k) - x(k))) = Nh^2 - 2hE. \quad (7)$$

In (7), N is the number of samples in the set, and E is the sum of differences between original samples and pre-SAO samples as defined in the following equation:

$$E = \sum_{k \in C} (s(k) - x(k)). \quad (8)$$

Please note that sample classification and (8) can be calculated right after pre-SAO samples are available during the DF processes. Thus, N and E can be calculated only once and

stored. Next, the delta rate-distortion cost is defined in the following equation:

$$\Delta J = \Delta D + \lambda R \quad (9)$$

In (9), λ is the Lagrange multiplier, and R represents the estimated bits of side information.

For a given CTB with a specified SAO type (i.e., BO or EO), a specified starting band position or EO class, and a specified band or category, a few h values (i.e., offsets) close to the value of E/N are tested, and the offset that minimizes ΔJ will be chosen. After offsets of all bands or categories are chosen, we can add up the ΔJ of each of the 32 bands for BO or each of the five categories for EO to obtain the delta rate-distortion cost of the entire CTB, where the distortions of the BO bands using zero offsets implicitly and the EO category 0 can be precalculated by (5) and stored for reuse. When the delta cost of the entire CTB is negative, SAO can be enabled for the CTB. Similarly, the best SAO type and the best starting position or EO class can be found using the fast distortion estimation.

E. Slice-Level On/Off Control

In the HEVC reference software common test conditions [28], hierarchical quantization parameter (QP) settings for each group of pictures (GOP) are often used. As an example, in the random access condition, the GOP size is eight. Any pictures with picture order count (POC, i.e., display order) equal to $8k$ belongs to depth 0, any picture with POC equal to $(8k + 4)$ belongs to depth 1, any picture with POC equal to $(8k + 2)$ or $(8k + 6)$ belongs to depth 2, and any picture with POC equal to $(8k + 1)$, $(8k + 3)$, $(8k + 5)$, or $(8k + 7)$ belongs to depth 3, where k is a nonnegative integer. A picture with a larger depth will be given a higher QP.

A slice-level on/off decision algorithm [29], [30] is provided as follows. For depth 0 pictures, SAO is always enabled in the slice header. Given a current processing picture with a nonzero depth N , the previous picture is set to the last picture of depth $(N - 1)$ in the decoding order. If the previous picture disables SAO for more than 75% of CTBs, the current picture will early terminate the SAO encoding process and disable SAO in all slice headers. Please note that luma and chroma can be turned on or off independently in the slice header.

F. Considerations for Right and Bottom Samples in the CTU

In the reference software, SAO parameters are estimated for each CTU at the encoder. Since SAO is after DF, the SAO parameters cannot be precisely estimated until the deblocked samples are available. However, the deblocked samples of the right columns and the bottom rows in the current CTU may be unavailable because the right CTU and the below CTU may have not yet been reconstructed in a CTU-based encoder. In order to consider this fact for practical CTU-based encoders, two options are provided. The first option [31] avoids using the bottom three luma sample rows, the bottom one Cb sample row, the bottom one Cr sample row, the rightmost four luma sample columns, the rightmost two Cb sample columns, and the rightmost two Cr sample columns in the current CTU

TABLE II
AVERAGE BD-RATES OF ENABLING SAO VERSUS DISABLING SAO
FOR DIFFERENT CTU SIZES

CTU Size in Luma	Option 1: Skip right and bottom samples in the CTU during parameter estimation			Option 2: Use predeblocked samples near right and bottom boundaries in the CTU during parameter estimation		
	Y	Cb	Cr	Y	Cb	Cr
64 × 64	-3.5%	-4.8%	-5.8%	-3.3%	-5.3%	-6.6%
32 × 32	-2.0%	-1.1%	-1.5%	-2.5%	-2.0%	-2.7%
16 × 16	0.0%	-0.3%	0.3%	-0.8%	0.4%	0.1%

during SAO parameter estimation on the encoder side. It does not suffer noticeable coding efficiency loss when the CTU size is 64×64 in luma. Nevertheless, when the CTU size is smaller, the percentage of unused samples in the CTU becomes higher and might cause considerable coding efficiency loss. Hence, the second option [32] uses predeblocked samples to replace the unavailable deblocked samples in the current CTU during SAO parameter estimation, which can reduce the coding efficiency loss caused by the first option for smaller CTU sizes.

VI. EXPERIMENTAL RESULTS

This section presents experimental data to illustrate the benefits of SAO objectively and subjectively.

A. Test Conditions

All experiments were conducted using the HEVC test model version 8.0 (HM8.0) and the common test conditions [28].

- 1) Four quantization values are used: 22, 27, 32, and 37.
- 2) All coding efficiency tools in the Main profile are enabled.
- 3) Six classes of test sequences representing different use cases and video characteristics are used. There are 24 test sequences in total.
- 4) Four different GOP structures are used. They are all intra (AI), random access (RA), low delay with B pictures (LB), and low delay with P pictures (LP).
- 5) Bjontegaard-Delta bitrate (BD-rate) measure [33] is used to evaluate objective coding efficiency.
- 6) Run times are measured at both encoders and decoders.

B. Objective Results

Table II summarizes the BD-rates of enabling SAO for different CTU sizes, where the anchor is disabling SAO. Please note that a negative BD-rate number means SAO has coding gains. The two options of considerations for boundary samples in the CTU during SAO parameter estimation are shown. For CTU size equal to 64×64 in luma, SAO can provide about 3.5% luma coding gains and more than 5% chroma coding gains on average. Besides, option 2 (i.e., using predeblocked boundary samples) is better than option 1 (i.e., skipping boundary samples) for CTU sizes equal to 32×32 and 16×16 in luma.

Table III reports the sequence by sequence luma BD-rates and the summarized luma BD-rates and run times under different GOP structures for CTU size equal to 64×64 in

TABLE III
SEQUENCE BY SEQUENCE LUMA BD-RATES AND SUMMARIZED LUMA
BD-RATES AND RUN TIMES UNDER DIFFERENT GOP STRUCTURES

Anchor: Disabling SAO Test: Enabling SAO CTU Size in Luma: 64 × 64 CTU Boundary: Option 1	Y BD-rate			
	All Intra (AI)	Random Access (RA)	Low Delay B (LB)	Low Delay P (LP)
Class A Cropped 4K×2K	Traffic	-1.0%	-1.2%	
	PeopleOnStreet	-1.3%	-2.1%	
	Nebuta	-0.1%	-2.9%	
	SteamLocomotive	-0.3%	-2.9%	
Class B 1080p	Kimono	-0.5%	-0.6%	-8.0%
	ParkScene	-0.7%	-0.7%	-9.1%
	Cactus	-0.5%	-2.4%	-10.4%
	BasketballDrive	-0.2%	-1.6%	-9.0%
Class C WVGA	BQTerrace	-0.5%	-5.1%	-19.0%
	BasketballDrill	-1.1%	-1.7%	-6.3%
	BQMall	-0.3%	-0.8%	-8.3%
	PartyScene	-0.2%	0.1%	-4.1%
Class D WQVGA	RaceHorses	-0.6%	-2.1%	-9.8%
	BasketballPass	-0.3%	-0.5%	-4.7%
	BQ Square	-0.5%	-0.0%	-3.8%
	BlowingBubbles	-0.2%	0.6%	-2.7%
Class E 720p	RaceHorses	-0.6%	-1.2%	-6.3%
	FourPeople	-0.7%		-8.8%
	Johnny	-0.4%		-13.2%
	KristenAndSara	-0.6%		-11.1%
Class F	BasketballDrillText	-1.1%	-2.0%	-23.5%
	ChinaSpeed	-1.4%	-3.5%	-8.2%
	SlideEditing	-1.5%	-2.6%	-5.5%
	SlideShow	-1.9%	-2.1%	-12.2%
Class Summary	Class A	-0.6%	-2.3%	
	Class B	-0.5%	-2.1%	-11.1%
	Class C	-0.5%	-1.1%	-7.1%
	Class D	-0.4%	-0.3%	-4.4%
	Class E	-0.6%		-11.0%
	Class F	-1.5%	-2.6%	-12.3%
Overall Summary	All	-0.7%	-1.7%	-9.2%
	Encoding time (%)	101%	100%	100%
	Decoding time (%)	103%	103%	102%

luma using the option 1 algorithm (i.e., skipping boundary samples). For *BasketballDrillText* in the LP condition, the SAO coding gain reaches 23.5%. It is noted that SAO is particularly effective for Class F sequences, which mainly contain computer graphic screen contents instead of natural images. It is also noted that SAO shows higher coding gains in the LP condition, where there is no bidirectional prediction. Regarding the run times, SAO increases less than 1% encoding time and 2–3% decoding time.

C. Subjective Results

One of the reasons that HEVC provides better coding efficiency than prior standards is the extended transform size, up to 32×32 . A large size transform provides better energy compaction; however, it tends to cause undesirable ringing artifacts. Fig. 9 shows an example of the computer-generated *SlideEditing* test sequence to highlight these effects. As shown in the figure, SAO significantly improves the visual quality by suppressing the ringing artifacts near true edges.

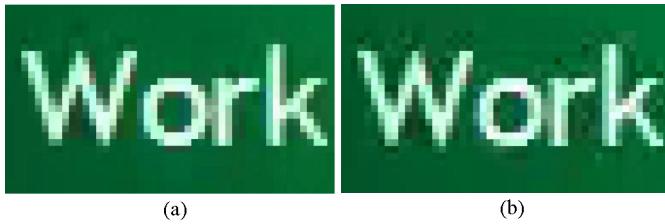


Fig. 9. Example in *SlideEditing* test sequence under LP condition, POC=100, QP=32: SAO is enabled in the left picture whereas SAO is disabled in the right picture.



Fig. 10. Subjective quality comparison of: (a) *RaceHorses* test sequence, POC=20, QP=32, LP condition, (b) *BasketballPass* test sequence, POC=14, QP=32, LP condition. For both test sequences, SAO is enabled in above pictures and disabled in middle pictures, and the below pictures are original pictures.

Fig. 10 shows two natural video examples of *RaceHorses* and *BasketballPass* test sequences. It can be observed easily that the edges of objects become much cleaner when SAO is enabled. According to the blind viewing tests conducted internally on our own, in general SAO can improve subjective quality, which is also reported independently in another subjective testing [34].

VII. CONCLUSION

The sample adaptive offset (SAO) technique has been adopted into the Main profile of the high-efficiency video coding (HEVC) standard. In this paper, SAO is described in detail. SAO locates after deblocking and is a new in-loop filtering technique that reduces the distortion between original

samples and reconstructed samples. It has been observed that SAO can improve video compression in both objective and subjective measures with reasonable complexity.

ACKNOWLEDGMENT

The authors thank the experts of ITU-T VCEG, ISO/IEC MPEG, and JCT-VC for their contributions and comments, which improved SAO from immature works to the current shape adopted in HEVC.

REFERENCES

- [1] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, *High Efficiency Video Coding (HEVC) Text Specification Draft 8*, document JCTVC-J1003, Jul., 2012.
- [2] Joint Video Team of ITU-T VCEG and ISO/IEC MPEG, *ITU-T Rec. H.264, ISO/IEC 14496-10 AVC*, 2005.
- [3] M. Yuen and H. R. Wu, "A survey of hybrid MC/DPCM/DCT video coding distortions," *J. Signal Process.*, vol. 70, no. 3, pp. 247–278, Nov. 1998.
- [4] W.-J. Han, J. Min, I. K. Kim, E. Alshina, A. Alshin, T. Lee, J. Chen, V. Seregin, S. Lee, Y. M. Hong, M. S. Cheon, N. Shlyakhov, K. McCann, T. Davies, and J. H. Park, "Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1709–1720, Dec. 2010.
- [5] S. Srinivasan, P. Hsu, T. Holcomb, K. Mukerjee, S. L. Regunathan, B. Lin, J. Liang, M.-C. Lee, and J. R.-Corber, "Windows Media Video 9: Overview and applications," *Signal Process. Image Commun.*, vol. 19, no. 9, pp. 851–875, Oct. 2004.
- [6] W.-J. Chien and M. Karczewicz, *Adaptive Filter Based on Combination of Sum-Modified Laplacian Filter Indexing and Quadtree Partitioning*, document VCEG-AL27, Jul., 2009.
- [7] K. McCann, W.-J. Han, I.-K. Kim, J.-H. Min, E. Alshina, A. Alshin, T. Lee, J. Chen, V. Seregin, S. Lee, Y.-M. Hong, M.-S. Cheon, and N. Shlyakhov, *Samsung's Response to the Call for Proposals on Video Compression Technology*, document JCTVC-A124, Apr. 2010.
- [8] Y.-W. Huang, C.-M. Fu, C.-Y. Chen, C.-Y. Tsai, Y. Gao, J. An, K. Zhang, and S. Lei, *In-Loop Adaptive Restoration*, document JCTVC-B077, Jul. 2010.
- [9] C.-M. Fu, C.-Y. Chen, Y.-W. Huang, and S. Lei, *TE10 Subtest 3: Quadtree-Based Adaptive Offset*, document JCTVC-C147, Oct. 2010.
- [10] C.-M. Fu, C.-Y. Chen, Y.-W. Huang, and S. Lei, *CE8 Subset 3: Picture Quadtree Adaptive Offset*, document JCTVC-D122, Jan. 2011.
- [11] C.-M. Fu, C.-Y. Chen, C.-Y. Tsai, Y.-W. Huang, and S. Lei, *CE13: Sample Adaptive Offset with LCU-Independent Decoding*, document JCTVC-E049, Mar. 2011.
- [12] C.-M. Fu, C.-Y. Chen, Y.-W. Huang, S. Lei, S. Park, B. Jeon, A. Alshin, and E. Alshina, *Sample Adaptive Offset for Chroma*, document JCTVC-F057, Jul. 2011.
- [13] C.-M. Fu, C.-Y. Chen, C.-Y. Tsai, Y.-W. Huang, S. Lei, I. S. Chong, M. Karczewicz, E. Alshina, and A. Alshin, *E8.a.3: SAO with LCU-Based Syntax*, document JCTVC-H0273, Feb. 2012.
- [14] C.-M. Fu, Y.-W. Huang, S. Lei, I. S. Chong, and M. Karczewicz, *Non-CE8: Offset Coding in SAO*, document JCTVC-G222, Nov. 2011.
- [15] W.-S. Kim and D.-K. Kwon, *Non-CE8: Method of Visual Coding Artifact Removal for SAO*, document JCTVC-G680, Nov. 2011.
- [16] W.-S. Kim and D.-K. Kwon, *CE8 Subset c: Necessity of Sign Bits for SAO Offsets*, JCTVC-H0434, Feb. 2012.
- [17] G. Laroche, T. Poirier, and P. Onno, *On Additional SAO Band Offset Classifications*, JCTVC-G246, Joint Collaborative Team on Video Coding, Nov. 2011.
- [18] E. Maani and O. Nakagami, *Flexible Band Offset Mode in SAO*, document JCTVC-H0406, Feb. 2012.
- [19] K. Minoo and D. Baylon, *AHG6: Coding of SAO Merge Left and Merge up Flags*, JCTVC-J0355, Joint Collaborative Team on Video Coding, Jul. 2012.
- [20] E. Alshina, A. Alshin, J. H. Park, G. Laroche, C. Gisquet, and P. Onno, *AHG6: On SAO Type Sharing Between U and V Components*, JCTVC-J0045, Joint Collaborative Team on Video Coding, Jul. 2012.

- [21] A. Minezawa, K. Sugimoto, and S. Sekiguchi, *Non-CE1: Improved Edge Offset Coding for SAO*, JCTVC-I0066, Joint Collaborative Team on Video Coding, Apr. 2012.
- [22] C.-M. Fu, Y.-W. Huang, and S. Lei, *Non-CE1: Bug-Fix of Offset Coding in SAO Interleaving Mode*, JCTVC-I0168, Joint Collaborative Team on Video Coding, Apr. 2012.
- [23] E. Alshina, A. Alshin, J. H. Park, C.-M. Fu, Y.-W. Huang, and S. Lei, *AHG5/AHG6: On Reducing Context Models for SAO Merge Syntax*, JCTVC-J0041, Joint Collaborative Team on Video Coding, Jul. 2012.
- [24] E. Alshina, A. Alshin, and J. H. Park, *AHG5: On Bypass Coding for SAO Syntax Elements*, JCTVC-J0043, Joint Collaborative Team on Video Coding, Jul. 2012.
- [25] J. Xu and A. Tabatabai, *AHG6: On SAO Signaling*, JCTVC-J0268, Joint Collaborative Team on Video Coding, Jul. 2012.
- [26] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.
- [27] C.-M. Fu, C.-Y. Chen, Y.-W. Huang, and S. Lei, "Sample adaptive offset for HEVC," in *Proc. IEEE 13th Int. Workshop MMSP*, Oct. 2011, pp. 1–5.
- [28] F. Bossen, *Common HM Test Conditions and Software Reference Configurations*, document JCTVC-J1100, Jul. 2012.
- [29] G. Laroche, T. Poirier, and P. Onno, *Non-CE1: Encoder Modification for SAO Interleaving Mode*, JCTVC-I0184, Joint Collaborative Team on Video Coding, Apr. 2012.
- [30] E. Alshina, A. Alshin, and J. H. Park, *Encoder Modification for SAO*, JCTVC-J0044, Joint Collaborative Team on Video Coding, Jul. 2012.
- [31] Y.-W. Huang, E. Alshina, I. S. Chong, W. Wan, and M. Zhou, *Description of Core Experiment 1 (CE1): Sample Adaptive Offset Filtering*, JCTVC-H1101, Joint Collaborative Team on Video Coding, Feb. 2012.
- [32] W.-S. Kim, *AHG6: SAO Parameter Estimation Using Non-Deblocked Pixels*, JCTVC-J0139, Joint Collaborative Team on Video Coding, Jul. 2012.
- [33] G. Bjøntegaard, *Calculation of average PSNR differences between RD curves*, document VCEG-M33, 13th VCEG Meeting, Apr. 2001.
- [34] T. K. Tan, A. Fujibayashi, Y. Suzuki, and J. Takiue, *AHG8: Objective and Subjective Evaluation of HM5.0*, JCTVC-H0116, Joint Collaborative Team on Video Coding, Feb. 2012.



Chih-Ming Fu was born in I-Lan, Taiwan, in 1977. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from National Tsing Hua University (NTNU), Hsinchu, Taiwan, in 1999, 2001, and 2006, respectively.

He joined MediaTek, Inc., Hsinchu, in March 2008 and is currently working with the Multimedia Technology Development Division, Corporate Technology Office, of the MediaTek headquarters as a Technical Manager. His major research interests include image and video coding algorithms, image and video processing algorithms, wireless communication, and pattern recognition.



Elena Alshina was born in Russia in 1972. She received the M.S. degree in physics from Lomonosov Moscow State University (MSU), Moscow, Russian Federation, in 1995. She continued her post-graduate study at the Faculty of Computational Mathematics and Cybernetics, MSU, and received the Ph.D. degree in mathematical modeling in 1998.

In 1998, she was with the Institute for Mathematical Modeling, Russian Academy of Science, Moscow, performing research on high accuracy numerical methods and new finite-difference scheme development. Simultaneously she has also worked as an Associate Professor with the Moscow Institute of Electronic Technology, Moscow. In 2006, she joined Samsung (first Moscow Research Center and then moved to Korea). Currently, she is with the Multimedia Platform Laboratory, Digital Media and Communications Research and Development Center, Samsung Electronics, as a Principle Engineer making high-efficiency video codec.



Alexander Alshin was born in Russia in 1972. He received the M.S. and Ph.D. degrees in mathematical physics from Moscow State University (MSU), Moscow, Russia, in 1995 and 1998, respectively.

From February 1998 to January 2006, he was a Senior Researcher with the Physical Department, MSU. In 2006, he joined Samsung Electronics, Suwon, Korea, and is currently a Principle Engineer with the Multimedia Platform Laboratory, DMC Research and Development Center, Samsung Electronics, Korea. His current research interests include partial differential equations, nonlinear functional analysis, applied mathematics, numerical analysis, and their applications to video compression.



Yu-Wen Huang was born in Kaohsiung, Taiwan, in 1978. He received the B.S. degree in electrical engineering and the Ph.D. degree in electronics engineering from National Taiwan University (NTU), Taipei, Taiwan, in 2000 and 2004, respectively.

He joined MediaTek, Inc., Hsinchu, Taiwan, in December 2004, and is currently a Senior Manager with the Multimedia Technology Development Division, Corporate Technology Office, MediaTek headquarters. His major research interests include image and video coding algorithms, image and video processing algorithms, and associated very large scale integration architectures.



Ching-Yeh Chen was born in Taipei, Taiwan, in 1980. He received the B.S. degree in electrical engineering and the Ph.D. degree in electronics engineering from National Taiwan University (NTU), Taipei, Taiwan, in 2002 and 2006, respectively.

He joined MediaTek, Inc., Hsinchu, Taiwan, in October 2006, and is currently a Technical Manager with the Multimedia Technology Development Division, Corporate Technology Office, MediaTek headquarters. His major research interests include intelligent image and video signal processing, global and local motion estimation, image and video coding algorithms, and associated very large scale integration architectures.



Chia-Yang Tsai (M'09) received the B.S., M.S., and Ph.D. degrees in electronics engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 2001, 2003, and 2010, respectively.

He joined MediaTek, Inc., Hsinchu, in March 2010 and is currently a Senior Engineer with the Multimedia Technology Development Division, Corporate Technology Office, MediaTek headquarters. His research interests include image and video coding algorithms, video rate control, multimedia communication, and image and video signal processing.



Chih-Wei Hsu received the B.S. degree in electrical engineering and the M.S. degree in electronics engineering from National Taiwan University (NTU), Taipei, Taiwan, in 2001 and 2003, respectively.

He joined MediaTek, Inc. in October 2003 and is currently working as a Technical Manager with the Multimedia Technology Development Division, Corporate Technology Office, MediaTek headquarters. His major research interests include global and local motion estimation, image and video coding algorithms, and associated VLSI architectures.



Shaw-Min Lei (S'87–M'88–SM'95–F'06) received the B.S. and M.S. degrees from National Taiwan University (NTU), Taipei, Taiwan, in 1980 and 1982, respectively, and the Ph.D. degree from the University of California, Los Angeles, in 1988, all in electrical engineering.

From August 1988 to October 1995, he was with Bellcore (Bell Communications Research), Red Bank, NJ, where he had worked mostly in video compression and communication areas and for a short period of time in wireless communication areas. From October 1995 to March 2007, he was with Sharp Laboratories of America, Camas, WA, where he was a Manager of the Video Coding and Communication Group. Since March 2007, he has been with MediaTek, Hsinchu, Taiwan, as the Director of Multimedia Technology Division, working in video/image coding and processing areas. His group has made very significant contributions to the under-developing high efficiency video coding (HEVC) standard. He has published more than 60 peer-reviewed technical papers and more than 200 contributions to MPEG4, JPEG2000, H.263+, H.264, and HEVC international standard meetings. He holds more than 40 patents. His current research interests include video/image compression, processing and communication, picture quality enhancement, multimedia communication, and data compression.



Woo-Jin Han (M'02) received the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1997 and 2002, respectively.

He is currently a Professor with the Department of Software Design and Management, Gachon University, Seongnam, Korea. From 2003 to 2011, he was a Principal Engineer with the Multimedia Platform Laboratory, Digital Media and Communication Research and Development Center, Samsung Electronics, Suwon, Korea. Since 2003, he has contributed successfully to the ISO/IEC Moving Pictures Experts Group, Joint Video Team, and Joint Collaborative Team standardization efforts. His current research interests include high-efficiency video compression techniques, scalable video coding, multi-view synthesis, and visual contents understanding.

Dr. Han was an editor of the HEVC video coding standard in 2010.



Jeong-Hoon Park received the B.S. and M.S. degrees in electrical engineering from Hanyang University, Seoul, Korea, in 1991 and 1994, respectively.

He is currently the Head of the Multimedia Platform Laboratory, Digital Media and Communication Research and Development Center, Samsung Electronics, Suwon, Korea. In 1998, he was a Visiting Scholar with the University of California, Los Angeles, where he researched on error resilient video compression and transmission. Since then, he has contributed to the ITU-T Video Coding Experts Group and ISO/IEC Moving Pictures Experts Group. From 2002 to 2004, he was involved in developing mobile broadcasting technology as an active member of WorldDMB forum. His current research interests include audio and video compression technology, medical image processing, and multimedia systems, including video broadcasting and communications.