IMPROVED SAMPLE ADAPTIVE OFFSET FOR HEVC

Woo-Shik Kim, Do-Kyoung Kwon

Systems and Applications R&D Center
Texas Instruments Inc.
12500 TI Blvd., Dallas, TX 75243

ABSTRACT

Sample adaptive offset (SAO) is the new in-loop filter in High Efficiency Video Coding (HEVC) standard. In this paper, the problems in the early version of SAO technique is discussed, and it is shown how the proposed methods improve its performance. It is proposed to restrict edge offset sign according to edge shape to reduce visual artifact caused by edge offset. It is also proposed to reduce number of band offset to facilitate implementation. Experimental results shows the proposed methods effectively reduce the artifact caused by edge offset, and reduce number of band offset by half without compromising coding efficiency.

***Index Terms*—** video compression, image coding, in-loop filtering, HEVC

1. INTRODUCTION

Video post-processing has been widely used to improve video quality. Especially, there are filters to remove artifacts caused by lossy compression such as blocking, ringing, blurring, etc. The filtering process can also be included in the coding loop. This is called in-loop filtering, where the filtered results are used for motion compensation. This requires the same filtering procedure both at the encoder side and the decoder side [1, 2]. In H.264/MPEG-4 AVC [3], which has been the most popular video coding standard nowadays, deblock filter is defined as in-loop filter, so that deblocked frames are used for motion compensation to improve rate-distortion performance [4].

Now, the new video coding standard, High Efficiency Video Coding (HEVC), has been developed by the Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T WP3/16 and ISO/IEC JTC 1/SC 29/WG 11 [5]. HEVC is expected to provide around 40% improvement in coding efficiency compared with H.264/AVC. Its high coding efficiency is brought by several coding tools such as quad-tree based coding with variable sizes of prediction and transform unit that can go up to 64×64 and 32×32, respectively, improved interpolation filter, advanced motion vector prediction (AMVP), residual quad tree (RQT), and so on.

A new in-loop filtering method, sample adaptive offset (SAO), has been introduced recently and adopted into HEVC, which is applied right after deblock filtering process. While the deblock filter works adaptively according to the coded results without transmitting additional information, SAO requires to send additional information to signal filter type and offset value. SAO involves adding an offset directly to the deblocking filtered reconstructed pixel so as to reduce the distortion between the input picture and the deblocked picture. The output of SAO is sent for displaying on a screen, and also stored as a reference for motion compensation.

Send correspondence to wskim@ti.com

There are two types of filters in SAO – edge offset (EO) and band offset (BO). EO is applied according to edge direction and shape, and BO is applied according to pixel intensity level range. These filters are selectively applied to each region. In the early stage of SAO development, the region was formed using a qaud-tree structure. To improve latency this has been changed to block based structure in the final design, where SAO parameters are signaled for each coding tree unit defined in HEVC.

SAO usually improves coding efficiency and visual quality. However, in its early stage of development it had significant problems that limited practical application of the technique. First, it was observed that EO could introduce visual artifacts, e.g., salt and pepper noise. Second, BO required excessive buffer size to store offset values, which could be burdensome for hardware implementation. In this paper, we propose a new SAO design to address them, which is based on our previous works proposed during the HEVC standardization [6, 7]. Note that the proposed method for EO has been adopted into HEVC, and also the proposed method for BO motivated the current design of BO in HEVC.

This paper is organized as follows. The description of SAO EO design in HM-4.0 is given in Section 2 with problem statement, followed by the proposed solution. In Section 3, the description of SAO BO in HM-4.0 is given with problem statement, followed by the proposed solution. It is also described how the latest SAO BO design has been developed. Experimental results are provided in Section 4, followed by concluding remarks in Section 5.

2. SAO EDGE OFFSET (EO)

2.1. What is EO?

EO classifies pixels based on edge direction and shape, and adds corresponding offset value to each pixel. First, one of four edge directions, 0, 90, 135, and 45 degrees as shown in Fig. 1, is selected for the given region. For the selected direction, the corresponding 1-D pattern shown in Fig. 1 is applied for each pixel to determine its category according to edge shape by comparing the pixel to its neighbor pixels in the selected direction. If the edge shape of the pixel belongs to one of four categories shown in Fig. 2, corresponding offset is added to the pixel. The selected direction and four offsets (one for each category) are coded for each region.

2.2. Artifact due to EO

Although EO can generally enhance visual quality of areas with edges, it can also cause artifacts when a wrong offset value is applied. We have noticed that this artifact is visible when the offset direction is orthogonal to that of edge shape. More specifically, for

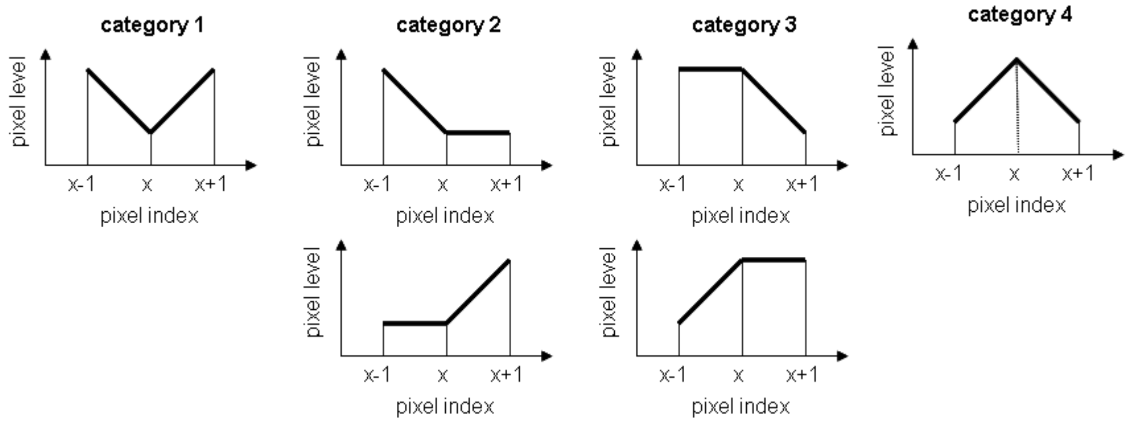


Fig. 2. Edge offset categorization according to edge shape.

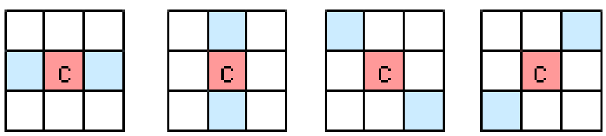


Fig. 1. Edge offset direction according to 1-D patterns; 0, 90, 135 and 45 degrees.

category 1 and 2, if offset value is negative, this will make larger difference between the current pixel and the neighboring pixels, which results in larger contrast. For category 3 and 4, positive offset will do the same.

This can happen when there are more than one part in the region having very different statistics compared to each other. If one part is dominant in the region, the offset value can be determined reflecting the statistics of this part. Then, this offset may cause artifact in the other part in the region.

The artifact usually looks like salt and pepper noise, or sometimes it appears as a line along a false contour. Also, it can cause color distortion when SAO is applied to chroma components. Moreover, this artifact can be emphasized by sharpness enhancement and/or edge strengthening filters, which are widely used as post-processing filters in consumer appliances such as a digital TV. Some examples are given in Section 4.

2.3. Proposed solution - EO sign matching to edge direction

The edge shapes in each category illustrated in Fig. 2 suggest that the sign of offset can be related to the shape of the edge. For example, in category 1 and 2, the reconstructed pixel level is lower than the neighboring pixels. Therefore, the offset would tend to have positive sign. On the other hand, in category 3 and 4, the offset would have negative sign. We have verified this experimentally, and found the offset sign matches the category in more than 97% cases, i.e., positive offsets for categories 1 and 2, and negative offsets for categories 3 and 4. However, as explained in Section 2.2, if a negative offset applies to pixels in category 1 and 2, this will increase contrast between neighboring pixels, and can result in visual artifact. Same will happen when a positive offset applies to pixels in category 3 and 4.

Therefore, we propose to restrict the offset values to be 0 or positive values for category 1 and 2, and 0 or negative values for category 3 and 4. Same scheme is applied for all color components. This consequently makes it unnecessary to code offset sign in entropy coding. Therefore, we propose not to code sign of edge offsets. This not only leads to complexity reduction, but also improves coding efficiency, since less amount of bits is required to code edge offsets. Note that this only applies to EO. In case of BO, no restriction is applied for offset sign value, which is coded explicitly.

The proposed method was first proposed in [6], which has been adopted into HEVC after core experiments performed by JCT-VC.

3. SAO BAND OFFSET (BO)

3.1. What is BO?

While EO adds offset to each pixel according to its edge shape, BO adds offset according to each pixel's intensity value. In HM-4.0, the entire pixel level range (i.e., 0 to 255 for 8-bit video) is equally divided into 32 sections. Each section can have one offset. For each pixel, the offset of the section this pixel belongs to is added. However, signaling 32 offsets for each region would not be efficient as the pixel value range in one region would not span widely. So, in HM-4.0, 32 sections are divided into two groups – middle band and side band as shown in Fig. 3. This requires signaling 16 offsets for each region along with the selected band.

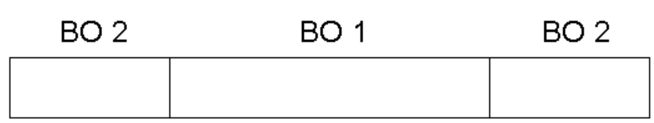


Fig. 3. Band division for band offset in the current SAO design

3.2. Proposed improvements and the final design in HEVC

In HM-4.0, the region which can have its own SAO parameters is determined using a quad-tree structure, and SAO parameters are signaled as part of an adaptation parameter set (APS), which is transmitted separately from slice data containing coded block data. This requires to store SAO parameters for the whole frame at a decoder side until SAO process is completed for each slice. The buffer size is proportional to the number of partitions and size of SAO parameters. In many implementations, the buffer has to be prepared beforehand, and need to provide enough space to store information in the worst case. The worst case happens when a slice or frame is divided into maximum number of partitions, and all the partition uses maximum number of offsets. While EO requires signaling 4 offsets, BO requires to signal 16 offsets, which would increase the buffer size to store SAO parameters significantly. In [7], we pointed out that this can be burdensome. For example, if there are 256 partitions in a picture with all using BO, assuming 6 bits for each offset for 10 bit video case for all three color components, the buffer size becomes 6 bits/offset $\times$ 16 offsets $\times$ 3 color components = 9 Kbyte.

Therefore, it is necessary to decrease number of offsets to reduce required buffer size. However, less number of offset would mean covering wider range of intensity level for each offset, if the band coverage is kept the same. To solve this problem, we propose to divide the whole intensity level range into 4 bands as shown in Fig. 4. Selected band is coded as SAO type. For each band the intensity level range is divided into 8 sub-bands, and one offset is assigned for each sub-band. Therefore, only 8 offsets need to be coded and stored in the buffer, which consequently reduces the maximum buffer size by half.

Note that the two middle bands are assigned as BO type 1 and 2, and the other bands are assigned as BO type 3 and 4, according to the usage frequency in many test sequences. Also note that the same method is applied for all color components.

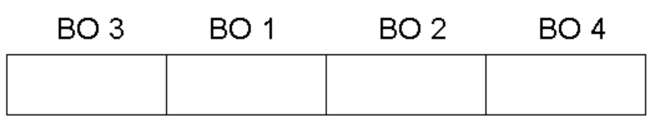


Fig. 4. Band division for band offset in the proposed method.

By increasing number of band from 2 to 4 and decreasing number of sub-band from 16 to 8, one offset can cover the same length of range in the proposed method as in the conventional scheme. However, one of the differences is that the coverage of one band is narrower than before. To resolve this issue, the coverage of offsets of the first and the last sub-band are extended to the pixels outside the band. This is illustrated in Fig. 5. The offset of the first sub-band covers pixels smaller than the minimum bound of the given band in addition to its own sub-band. The offset of the last sub-band covers pixels larger than the maximum bound of the given band in addition to its sub-band.

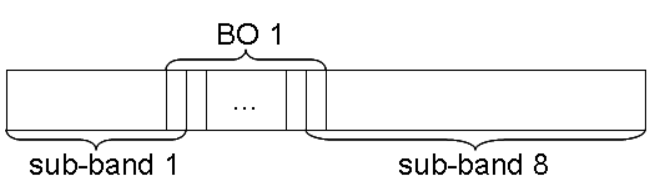


Fig. 5. Extended coverage of the first and last sub-band.

The proposed method had been tested through the core experiment set by JCT-VC. In the later stage of SAO development, the unit or region for SAO parameter signaling is changed from a quad-tree partition to a coding tree unit to improve latency. This means that SAO parameters are interleaved into coded block data in a slice, and SAO parameters for the whole frame no longer need to be stored in a buffer. Also to improve coding efficiency of BO, the number of offset is reduced to 4.

4. EXPERIMENTAL RESULTS

Two methods are proposed to improve the performance of SAO – one for EO scheme, and the other for BO scheme. The proposed methods are implemented separately using HM-4.0 reference code. Total 24 sequences are used for the tests, of which the resolution ranges from WQVGA (400×240) to 4K, compressed using 4 quantization parameter values of 22, 27, 32, and 37. Various test configurations are used such as all intra, low-delay, and random access as defined in the HEVC common test conditions [8].

Table 1 is the summary of the simulation results in terms of the objective quality, where the coding efficiency is represented using BD-bitrate (BDBR) [9]. Note that minus sign means bitrate reduction. For the EO proposal, on average there is 0.0% bitrate reduction in Y, and 0.2% bitrate increment in U and V, respectively. For the BO proposal, on average there is 0.0% bitrate reduction in Y and U, and 0.1% bitrate increment in V, respectively. This implies that both methods have almost no impact on the coding efficiency in terms of the objective quality measure.

Table 1. BD-bitrate results of EO and BO proposals compared to HM-5.0 (%).

Configuration	EO			BO		
	Y	U	V	Y	U	V
All intra	0.0	0.0	0.0	0.0	0.0	0.1
Random access	0.0	0.1	0.1	0.0	0.0	0.1
Low delay B	0.0	0.3	0.4	0.0	-0.1	0.2
Low delay P	0.0	0.2	0.2	0.0	-0.1	0.0

We have performed subjective quality evaluation using various test sequences including those used in [10]. First, it is verified that the claimed SAO artifacts exist in the results of HM-4.0 as shown in the left column images in Fig. 6. It usually appears like salt and pepper noise. Sometimes, it appears as a line or false contour as can be seen in Fig. 6 (c). It is also observed that it can cause color distortion when this artifact occurs in Cb or Cr component (results are available in [6]). Sometimes, it appears once and disappears soon. But, sometimes the artifacts propagates through motion compensation, which causes significant subjective quality degradation. Also when special post-processing filters are used at the display like edge enhancement or sharpness enhancement filters, these artifacts are emphasized and become more visible. In the left column images in Fig. 6, it can be observed that the SAO artifacts are mostly gone by use of the proposed method. We performed extensive subjective quality test and observed there is subjective quality improvement in a few sequences, and no sequence suffers quality degradation due to the proposed method.

The subjective quality of the BO proposal is also examined to verify that it does not affect the quality, while the buffer size to store SAO parameters is reduced by half.

5. CONCLUSION

In this paper, two problems in the conventional SAO design has been analyzed, and solutions are proposed to resolve them. First, it has been observed that EO occasionally causes visual artifact by increasing contrast between neighboring pixels. It is proposed to restrict the offset to have positive sign for edge offset category 1 and 2, and negative sign for category 3 and 4. Experimental results show that the proposed method effectively removes the visual artifact without compromising visual quality, objective quality, and computational complexity.

Secondly, the buffer size to store SAO parameters is analyzed. Then, to reduce the buffer size a new band offset scheme is proposed, in which the number of band is increased from 2 to 4, and the number of sub-band is reduced from 16 to 8, so that the buffer size to store these offsets can be reduced by half. Experimental results show that the proposed BO scheme maintains both subjective and objective quality as before.

6. REFERENCES

[1] K. K. Pang and T. K. Tan, "Optimum loop filter in hybrid coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 2, pp. 158–167, Apr. 1994.

[2] Y.-L. Lee and H. W. Park, "Loop filtering and post-filtering for low-bitrates moving picture coding," in *Proc. of IEEE Int. Conf. Image Proc., ICIP 1999*. Kobe, Japan, Oct. 1999.

[3] ITU-T and ISO/IEC JTC 1, *Advanced video coding for generic audiovisual services*, ITU-T Rec. H.264 and ISO/IEC 14496-10 (AVC), May 2003.

[4] P. List, A. Joch, J. Lainema, G. Bjøntegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 614–619, Jul. 2003.

[5] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, and T. Wiegand, "High efficiency video coding (HEVC) text specification draft 9," Document JCTVC-K1003, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Oct. 2012.

[6] W.-S. Kim and D.-K. Kwon, "Non-CE8: Method of visual coding artifact removal for SAO," Document JCTVC-G680, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Nov. 2011.

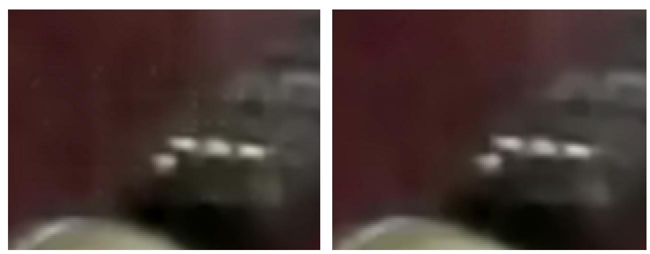
[7] W.-S. Kim and D.-K. Kwon, "Non-CE8: Reduced number of band offsets in SAO," Document JCTVC-G682, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Nov. 2011.

[8] F. Bossen, "Common test conditions and software reference configurations," Document JCTVC-F900, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Jul. 2011.

[9] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," Document VCEG-M33, ITU-T SG.16 Q.6, Apr. 2001.

[10] W.-S. Kim, O. G. Sezer, and M. Budagavi, "Subjective tests on ALF and SAO using HM-4.0," Document JCTVC-G684, ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Nov. 2011.

(a) BQMall



(b) Vidyo1

(c) BasketballDrive

Fig. 6. Reduction of SAO artifact from EO using the proposed method. Left column images: results of HM-4.0, right column images: results by proposed method.