Sample Adaptive Offset for HEVC

Chih-Ming Fu, Ching-Yeh Chen, Yu-Wen Huang, Shawmin Lei

Mediatek Inc.
No. 1, Dusing Rd. 1, Hsinchu Science Park, Hsinchu, Taiwan 30078
{chihming.fu, chingyeh.chen, yuwen.huang, shawmin.lei}@mediatek.com

***Abstract*—A new video coding tool, sample adaptive offset (SAO), is introduced in this paper. SAO has been adopted into the Working Draft of the new video coding standard, High-Efficiency Video Coding (HEVC). The SAO is located after deblocking in the video coding loop. The concept of SAO is to classify reconstructed pixels into different categories and then reduce the distortion by simply adding an offset for each category of pixels. The pixel intensity and edge properties are used for pixel classification. To further improve the coding efficiency, a picture can be divided into regions for localization of offset parameters. Simulation results show that SAO can achieve on average 2% bit rate reduction and up to 6% bit rate reduction. The run time increases for encoders and decoders are only 2%.**

***Keywords*— HEVC, SAO, Sample Adaptive Offset**

I. Introduction

ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) formed a Joint Collaborative Team on Video Coding (JCT-VC) in 2010, and the next generation video coding standard, High-Efficiency Video Coding (HEVC), is now being developed with the target of reducing 50% bit rate in comparison with Advanced Video Coding (AVC), or namely H.264, under the same visual quality. In JCT-VC meetings, many loop filtering proposals are extensively studied.

Band-correction (BDC) and extreme-value correction (EXC) are two video coding tools first proposed in JCTVC-A124 [1] for reducing the distortion between original pictures and reconstructed pictures. BDC is located between adaptive loop filtering (ALF) and reference picture buffer, and EXC is located between deblocking filter (DF) and ALF. The concept of BDC and EXC is to perform pixel classification and then derive an offset for each group of pixels. The pixel classification has to be done on the decoder side to avoid pixel-level side information, and the derived offsets are sent in the bitstream. BDC uses pixel intensity to classify pixels into different bands, and EXC uses the relation between a central pixel and its neighbouring pixels to classify pixels into different categories. BDC and EXC can achieve about 0.7% bit rate reduction for luma component. However, the decoding time of BDC and EXC is too high to justify the achieved coding gain. Therefore, they were not adopted in HEVC.

Inspired by BDC and EXC, we proposed sample adaptive offset (SAO) in JCTVC-B077 [2], JCTVC-C147 [3], JCTVC-D122 [4], and JCTVC-E049 [5], which include band offset (BO) and edge offset (EO). BO and EO can be regarded as improved BDC and EXC respectively and are combined into one SAO stage with local adaptation of BO and EO in each picture. The major improvements for SAO over its prior art [1] are the new pixel classification methods, which can further enhance the coding efficiency and decrease the decoding time significantly. By combining BO and EO into one SAO stage, the coding pass can also be significantly reduced without suffering coding efficiency loss. Consequently, the proposed SAO was adopted into HEVC working draft version 3.0 (WD-3.0) and HEVC test model version 3.0 (HM-3.0). We also proposed JCTVC-F056 [7] for chroma SAO, and it was also adopted in WD-4.0 and HM-4.0 in July, 2011. In this paper, the adopted SAO in JCTVC-E049 and JCTVC-F056 will be introduced. The rest of this paper is organized as follows. The details of SAO algorithm will be described in Section II, simulation results will be discussed in Section III, and conclusions will be given in Section IV.

II. Algorithm Description

The block diagram of HM-4.0 encoder is shown in Fig. 1, and the proposed SAO is located between DF and ALF. The side information of SAO is encoded by the entropy coding module and sent to the decoder. The block diagram of HM-4.0 decoder is shown in Fig. 2.

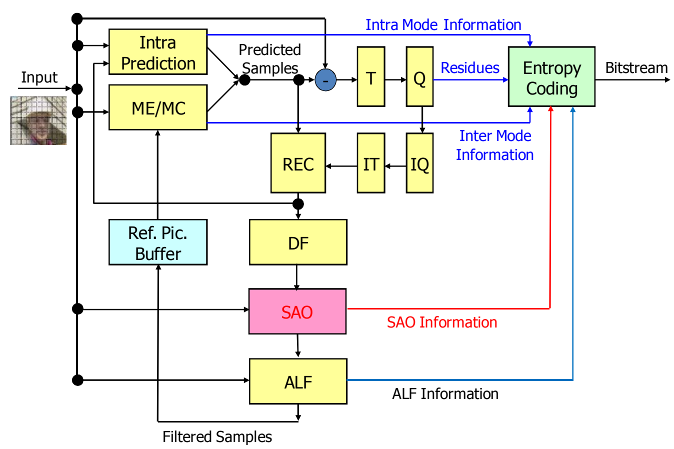

Fig. 1. Block diagram of HM-4.0 encoder

 MMSP 2011

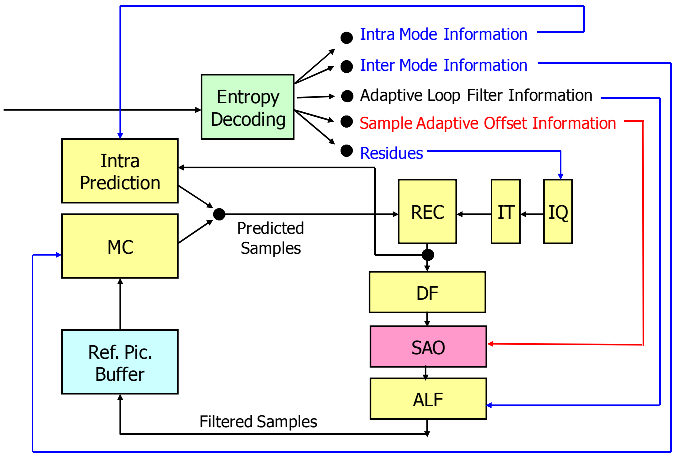

Fig. 2. Block diagram of HM-4.0 decoder

A. Overview

In HEVC, the basic processing unit is called coding unit (CU), and each CU can be further split into four sub-CUs recursively. Each picture consists of non-overlapping largest coding units (LCUs), which are similar to the conventional macroblocks used in prior video coding standards. Since the characteristics of a picture may vary with locations, SAO divides a picture into LCU-aligned regions to obtain local statistical information. Each region can be enhanced by either BO or EO, which is very different from the prior art in JCTVC-A124 where BDC and EXC are both applied for the entire picture as two separate stages. The proposed local adaptation method can lead to 1-pass processing and adapt offsets more rapidly for better coding efficiency. Figure 3 shows an example of dividing one picture into LCU-aligned regions where the dash lines indicate boundaries of LCUs. Note that neither BO nor EO is used for the regions marked by "OFF". During the DF process, statistical information for deriving offsets is collected, and then the decision of SAO parameters can be made by rate-distortion optimization with fast distortion estimation. By using a math equation, the fast distortion estimation can calculate the estimated distortion reduction of SAO and avoid fetching any pixel from the picture buffer to add offsets. On the encoder side, there is only one additional pass of picture buffer access to add offsets on deblocked pixels after all SAO parameters are decided.

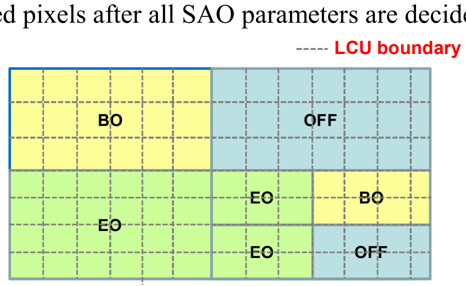

Fig. 3. Example of dividing a picture into LCU-aligned regions

B. Band offset

In JCTVC-A124, BDC always divides the pixel intensity into 16 fixed bands and transmits the offsets of all bands to the decoder. We found that there exists a trade-off between the number of bands and the amount of side information. A larger number of bands results in a smaller intensity interval of each band, which reduces the number of pixels in each band and increases the possibility of obtaining a nonzero offset that can reduce distortion. However, it also increases the amount of side information due to more offsets to be sent. Therefore, a band grouping method is proposed as follows. First, the BO classifies all pixels of a region into multiple bands where each band contains pixels in the same intensity interval. The intensity range is equally divided into 32 intervals from zero to the maximum intensity value (e.g. 255 for 8-bit pixels), and each interval has an offset. Next, it is observed that an offset tends to become zero when the number of pixels of the band is large, especially for those central bands. Hence, the 32 bands are divided into two groups. One group consists of the central 16 bands, while the other group consists of the rest 16 bands, as shown in Fig. 4. Only offsets in one group are transmitted. Regarding the pixel classification operation in BO, the five most significant bits of each pixel can be directly used as the band index, leading to an implementation with very low complexity.

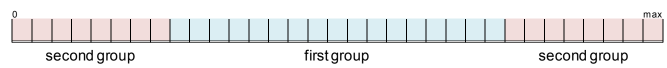

Fig. 4. Band offsets with two groups

C. Edge offset

In JCTVC-A124, EXC uses a 2-D 5-pixel cross pattern for pixel classification without considering any edge directional information. Since the pixel classification has to be performed also by the decoder, EXC suffers much higher decoding time than BDC. Furthermore, it is desirable to simplify the 2-D pattern for lower complexity. In the proposed EO, we use four 1-D 3-pixel patterns for pixel classification with consideration of edge directional information, as shown in Fig. 5. Each region of a picture can select one pattern to classify pixels into multiple categories by comparing each pixel with its two neighbouring pixels. The selection will be sent in bit-stream as side information. Figure 6 shows the pixel classification rule for EO. For example, valley pixels belong to category 1, and peak pixels belong to category 4.

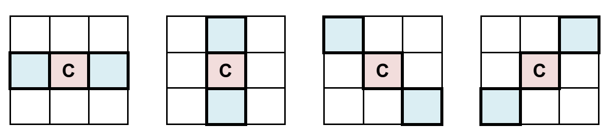

Fig. 5. Four 1-D 3-pixel patterns for the pixel classification in EO; from left to right: 0-degree, 90-degree, 135-degree, 45-degree

Category	Condition
1	c < 2 neighbors
2	c < 1 neighbor && c == 1 neighbor
3	c > 1 neighbor && c == 1 neighbor
4	c > 2 neighbors
0	None of the above

Fig. 6. Pixel classification rule for EO

TABLE I

Experimental results of BO without local adaptation

Anchor : JCTVC-D600		HE-RA	HE-LD	LC-RA	LC-LD
		Y BD-rate (%)	Y BD-rate (%)	Y BD-rate (%)	Y BD-rate (%)
Summary	All	-0.3	-0.5	-0.3	-0.5
	Encoding Time (%)	99	99	100	100
	Decoding Time (%)	100	99	102	101

TABLE II

Experimental results of EO without local adaptation

Anchor : JCTVC-D600		HE-RA	HE-LD	LC-RA	LC-LD
		Y BD-rate (%)	Y BD-rate (%)	Y BD-rate (%)	Y BD-rate (%)
Summary	All	-0.9	-1.3	-1.0	-1.2
	Encoding Time (%)	100	99	100	100
	Decoding Time (%)	104	103	102	105

TABLE III

Experimental results of the proposed SAO

Anchor : JCTVC-D600		HE-RA	HE-LD	LC-RA	LC-LD
		Y BD-rate (%)	Y BD-rate (%)	Y BD-rate (%)	Y BD-rate (%)
Class A 2560x1600	Traffic	-1.4		-1.5	
	PeopleOnStreet	-1.5		-2.3	
	Nebuta	-0.8		-2.8	
	SteamLocomotive	-0.9		-3.2	
Class B 1080p	Kimono	-0.5	-0.9	-0.9	-3.2
	ParkScene	-0.7	-1.6	-1.0	-2.8
	Cactus	-1.6	-2.1	-2.3	-3.4
	BasketballDrive	-1.1	-1.8	-2.1	-3.4
	BQTerrace	-3.7	-4.0	-5.5	-5.4
Class C WVGA	BasketballDrill	-2.4	-3.9	-2.3	-4.1
	BQMall	-1.4	-2.7	-1.4	-3.2
	PartyScene	-0.9	-2.0	-0.3	-2.1
	RaceHorses	-0.8	-1.0	-2.2	-2.7
Class D WQVGA	BasketballPass	-0.5	-1.0	-0.8	-1.9
	BQSquare	-2.7	-3.4	-1.2	-2.2
	BlowingBubbles	-0.6	-1.3	0.1	-0.7
	RaceHorses	-0.6	-1.3	-1.3	-2.1
Class E 720p	Vidyo1		-2.7		-2.7
	Vidyo3		-4.5		-6.2
	Vidyo4		-1.1		-2.6
Summary	All	-1.3	-2.2	-1.8	-3.0
	Encoding Time (%)	100	100	100	99
	Decoding Time (%)	103	103	101	101

D. Encoder Algorithm

In the proposed SAO encoder, one picture will be divided into multi-level quadtree regions. To determine the SAO parameters, the following process is performed:

1. Decide the maximum quadtree level, L, according to the picture size.
2. Divide the picture into smallest quadtree regions of level-L.
3. Collect the statistic data of all SAO types for each smallest region.
4. Set $k = L$
5. Derive the SAO parameters for each SAO type in each region at level-k.
6. Select the best SAO type with the lowest rate-distortion (RD) cost of each region at level-k.
7. For each parent region at level-(k-1), combine the statistic data of its children regions at level-k.
8. For each parent region at level-(k-1), decide the children regions are merged into the parent region according the RD-cost or not.
9. Set $k = k - 1$; if k > 0, run step 5, 6, 7, and 8; otherwise, end. ($k = 0$ represents that a picture is a single region).

The statistic data are collected for SAO encoder quadtree partitioning as follows: $N_{l,i,t,c}$ represents the number of pixels in level-l, region-i, type-t, and category-c. $a_{l,i,t,c}$ represents the corresponding offset. $e_{l,i,t,c}$ represents the corresponding sum of difference between reconstructed signal and original signal. $R_{l,i,t}$ represents the estimated rate of corresponding offsets. By using the collected statistic data, the estimated distortion reduction can be obtained by using a simple equation. The estimated distortion reduction $D_{l,i,t}$ for level-l, region-i, and type-t can be calculated by the following equation:

$$D_{l,i,t} = \sum_{c \in t} \left(N_{l,i,t,c} a^2{}_{l,i,t,c} - 2a_{l,i,t,c} e_{l,i,t,c} \right), \qquad (1)$$

where c is a category in SAO type-t. The detail information of the above equation is described in the Appendix A. After the estimated distortion reduction is obtained, we can calculate the RD-cost by using equation (2), where λ is the Lagrange multiplier.

$$J_{l,i,t} = D_{l,i,t} + \lambda R_{l,i,t} \qquad (2)$$

The encoder chooses the SAO type with the lowest estimated RD-cost (3) in level-l and region-i according to equation (4), where $\mathbf{T}$ is a set of SAO types which can be selected for a current region. Note that the RD-cost should be less than zero; otherwise, the current region will disable the processing of SAO.

$$J_{l,i} = \min_{t \in \mathbf{T}} (J_{l,i,t}) \qquad (3)$$

$$t_{l,i} = \arg\min_{t \in \mathbf{T}} (J_{l,i,t}). \qquad (4)$$

After the minimum cost and the best SAO type for each region are obtained, the encoder tries to merge children regions into one parent region according to equation (5) and (6), where $J'_{l,t}$ and $t'_{l,i}$ represent the new RD-cost and the new selected SAO type after merging. Note that $\Omega_{l,i}$ represents a set of sub-regions in level-l and region-i.

$$J'_{l,i} = \min\Big(\sum_{k \in \Omega_{l,i}} J_{l+1,k}, J_{l,i} \Big), \qquad (5)$$

$$t'_{l,i} = \arg\min_{t \in \mathbf{T}} \Big(\sum_{k \in \Omega_{l,i}} J_{l+1,k}, J_{l,i} \Big). \qquad (6)$$

TABLE IV

Experimental results of the proposed SAO for Cb component

Anchor : JCTVC-E700		HE-RA Cb BD-rate (%)	HE-LD Cb BD-rate (%)	LC-RA Cb BD-rate (%)	LC-LD Cb BD-rate (%)
Class A 2560x1600	Traffic	-1.5		-1.9	
	PeopleOnStreet	-1.3		-0.9	
	Nebuta	-6.2		-9.2	
	SteamLocomotive	-13.2		-7.4	
Class B 1080p	Kimono	-1.1	-1.5	-1.2	-7.1
	ParkScene	-1.1	-2.8	-0.4	-2.1
	Cactus	-4.9	-5.1	-7.0	-6.3
	BasketballDrive	-4.1	-7.7	-3.1	-7.8
	BQTerrace	-4.9	-8.2	-3.9	-9.2
Class C WVGA	BasketballDrill	-4.8	-9.0	-5.2	-11.2
	BQMall	-1.9	-3.7	-1.5	-5.4
	PartyScene	-2.3	-3.9	-1.1	-3.8
	RaceHorses	-1.3	-2.8	-0.7	-2.0
Class D WQVGA	BasketballPass	-2.1	-3.5	-1.0	-3.6
	BQSquare	-1.9	-1.6	-0.4	-2.7
	BlowingBubbles	-2.2	-4.6	-1.6	-6.3
	RaceHorses	-1.0	-1.9	-0.2	-1.0
Class E 720p	Vidyo1		-3.6		-6.0
	Vidyo3		-6.6		-13.6
	Vidyo4		-6.7		-13.5
Summary	All	-3.3	-4.6	-2.7	-6.3
	Encoding Time (%)	98	101	102	100
	Decoding Time (%)	98	100	101	102

TABLE V

Experimental results of the proposed SAO Cr component

Anchor : JCTVC-E700		HE-RA Cr BD-rate (%)	HE-LD Cr BD-rate (%)	LC-RA Cr BD-rate (%)	LC-LD Cr BD-rate (%)
Class A 2560x1600	Traffic	-1.5		-1.1	
	PeopleOnStreet	-2.1		-2.2	
	Nebuta	-3.5		-5.8	
	SteamLocomotive	-18.4		-10.2	
Class B 1080p	Kimono	-1.6	-2.3	-0.9	-5.2
	ParkScene	-2.1	-3.1	-0.5	-4.5
	Cactus	-4.3	-7.1	-3.6	-6.3
	BasketballDrive	-3.7	-7.3	-3.1	-7.6
	BQTerrace	-7.1	-11.6	-4.1	-13.6
Class C WVGA	BasketballDrill	-6.5	-12.5	-7.9	-15.3
	BQMall	-2.0	-4.4	-2.1	-6.7
	PartyScene	-2.7	-4.5	-0.9	-4.2
	RaceHorses	-2.1	-4.4	-1.8	-4.7
Class D WQVGA	BasketballPass	-1.7	-3.4	-0.9	-3.4
	BQSquare	-4.3	-4.8	-1.8	-7.4
	BlowingBubbles	-1.5	-2.6	-0.6	-3.8
	RaceHorses	-1.6	-3.0	-0.5	-3.2
Class E 720p	Vidyo1		-4.5		-9.3
	Vidyo3		-8.2		-8.8
	Vidyo4		-10.8		-17.2
Summary	All	-3.9	-5.9	-2.8	-7.6
	Encoding Time (%)	98	101	102	100
	Decoding Time (%)	98	100	101	102

According to the merge result of children regions, the spilt information of the corresponding parent region is determined by (7).

$$s_{l,i} = \begin{cases} 1, & \text{if } \sum_{k \in \Omega_{l+1,i}} J'_{l+1,k} < J'_{l,i} \\ 0, & \text{otherwise} \end{cases}, \qquad (7)$$

where $s_{l,i}$ equal to 1 represents that the current region will be split into sub-regions; otherwise the current region will be a leaf region without splitting. After all regions in level-k are processed, the encoder will process level-(k-1) until the level-0 is reached. Finally, the decoded picture buffer can be updated by using the determined SAO parameters for each region.

III. Simulation Result

Based on software platform HM-2.0, Common test configurations defined in JCTVC-D600 [6] are used as the anchor in TABLE I, TABLE II, and TABLE III. TABLE I shows the experimental result of the proposed BO without local adaptation. The bit rate reductions of BO are 0.3%, 0.5%, 0.3%, and 0.5% for high efficiency random access (HE-RA), high efficiency low delay (HE-LD), low complexity random access (LC-RA), and low complexity low delay (LC-LD), respectively, while the encoding time and decoding time are almost unchanged. TABLE II shows the experimental result of the proposed EO without local adaptation. The bit rate reductions of EO are 0.9%, 1.3%, 1.0%, and 1.2% for HE-RA, HE-LD, LC-RA, and LC-LD, respectively, while the encoding time is roughly unchanged and the decoding time is increased by 2-5%. As for the proposed SAO, it can achieve 1.3%, 2.2%, 1.8%, and 3.0% bit rate reductions for HE-RA, HE-LD, LC-RA, and LC-LD, respectively, with unnoticeable encoding time increase and 1-3% decoding time increase. Compared with the JCTVC-D600 anchor, the related art, BDC together with EXC, can achieve 0.7% BD-rate reduction with 1% encoding time increase and the 10% decoding time increase. It can be seen that SAO can not only effectively enhance the coding efficiency but also significantly accelerate the decoding process by a factor of 5 in comparison with BDC together with EXC. If the results are examined sequence by sequence, we can see that SAO brings coding gains for almost all cases. The coding gain is up to 6.2% for Vidyo3 in LC-LD. When TABLE I and TABLE II are compared with TABLE III, it can be seen that local adaptation not only combines BO and EO into one stage but also achieves better coding efficiency.

We also apply SAO on chroma component based on the HM-3.0 platform. The experimental results of Cb and Cr component are shown in TABLE IV and TABLE V respectively. The anchor here is JCTVC-E700 [8]. The bit rate reductions of Cb are 3.3%-6.3% and the bit rate reductions of Cr are 3.9%-7.6%, while the encoding time and decoding time are almost unchanged and luma performance is remain the same. It can be seen that SAO is also effective for chroma components. The proposed SAO was adopted in WD-4.0 and HM-4.0 as well.

IV. Conclusion

Sample adaptive offset, namely SAO, was proposed to reduce the distortion between reconstructed pixels and original pixels. The proposed SAO can achieve 1.3%, 2.2%, 1.8%, and 3.0% bit rate reductions for HE-RA, HE-LD, LC-RA, and LC-LD, respectively. The encoding time is roughly unchanged, and the decoding time is increased by 1-3%. Because of its good coding gain and low complexity, SAO was adopted in HEVC WD-3.0 and HM-3.0. Besides, we also show the proposed SAO is effective for chroma. The bit rate reductions of Cb component are 3.3%-6.3% and the bit rate reductions of Cr component are 3.9%-7.6%, while the encoding time and decoding time are almost unchanged. The proposed SAO for chroma was adopted in HEVC WD-4.0 and HM-4.0.

REFERENCES

[1] 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 of Joint Collaborative Team on Video Coding, JCTVC-A124, Apr. 2010.

[2] 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 of Joint Collaborative Team on Video Coding, JCTVC-B077, Jul. 2010.

[3] C.-M. Fu, C.-Y. Chen, Y.-W. Huang, and S. Lei, "TE10 Subtest 3: Quadtree-based Adaptive Offset," Document of Joint Collaborative Team on Video Coding, JCTVC-C147, Oct. 2010.

[4] C.-M. Fu, C.-Y. Chen, Y.-W. Huang, and S. Lei, "TE8 Subtest3: Picture Quadtree-based Adaptive Offset," Document of Joint Collaborative Team on Video Coding, JCTVC-D122, Jan. 2011.

[5] C.-M. Fu, C.-Y. Chen, C.-Y. Tsai, Y.-W. Huang, and S. Lei, "CE13: Sample Adaptive Offset with LCU-Independent Decoding" Document of Joint Collaborative Team on Video Coding, JCTVC-E049, Mar. 2011.

[6] F. Bossen, "Common Test Conditions and Software Reference Configurations," Document of Joint Collaborative Team on Video Coding, JCTVC-D600, Jan. 2011.

[7] C.-M. Fu, C.-Y. Chen, Y.-W. Huang, S. Lei (MediaTek), S. Park, B. Jeon (LGE), A. Alshin, E. Alshina (Samsung) "Sample Adaptive Offset for Chroma" Document of Joint Collaborative Team on Video Coding, JCTVC-F057, July. 2011.

[8] F. Bossen, "Common Test Conditions and Software Reference Configurations," Document of Joint Collaborative Team on Video Coding, JCTVC-E700, Jaly. 2011.

APPENDIX A

The fast distortion calculation of SAO can be derived as follows. Table VI shows the definition of the variables required by the derivation of fast distortion calculation of SAO. The distortion of the deblocked signal will be

$$\varepsilon_{rec} = \sum_{k\in K} \left(x(k) - s(k)\right)^2$$

Based on the above equation, ε_{rec} will be the TRUE distortion, not estimated distortion.

$$\varepsilon_{rec} = \sum_{k\in K} (x(k)^2 - 2\cdot x(k)\cdot s(k) + s(k)^2)$$

The distortion of the SAO processed signal will be

$$\varepsilon_{SAO} = \sum_{c\in t}\sum_{k\in C} \left((x(k) + a_c) - s(k)\right)^2$$

TABLE VI

Notation of Fast Distortion Calculation

$s(k)$ is the original signal.
$x(k)$ is the reconstructed signal.
a_c is the categorized offset
$\varepsilon_{rec}(k)$ is the distortion of the deblocked signal
$\varepsilon_{SAO}(k)$ is the distortion of the SAO signal.
K is a set of pixels to be processed
C is the pixels belonged to one type of SAO category
t is a set of SAO category for one type

$$= \sum_{c\in t}\sum_{k\in C} (x(k)^2 - 2\cdot x(k)\cdot s(k) + s(k)^2 + 2\cdot a_c\cdot x(k) + {a_c}^2 - 2\cdot a_c\cdot s(k))$$

$d\varepsilon_{AO} = \varepsilon_{SAO} - \varepsilon_{rec}$ = distortion reduction of SAO processed signal after the offset values is applied

$$d\varepsilon_{SAO} = \sum_{c\in t}\sum_{k\in C} (2\cdot a_c\cdot (s(k) - e_c) + {a_c}^2 - 2\cdot a_c\cdot s(k))$$

$$= \sum_{c\in t}\sum_{k\in C} (2\cdot a_c\cdot s(k) - 2\cdot a_c\cdot e_c + {a_c}^2 - 2\cdot a_c\cdot s(k))$$

$$= \sum_{c\in t}\sum_{k\in C} ({a_c}^2 - 2\cdot a_c\cdot e_c)$$

$$= \sum_{c\in t} \left(N_c {a_c}^2 - 2\cdot N_c\cdot a_c\cdot e_c\right)$$

where N_c is the number of pixel of current category which is $\sum_{k\in C}(1)$, a_c is the offset value to be added on the pixels belonging to category c, and e_c is summation of the offset value between original signal and reconstructed signal.