

Featured next-generation video compression technology HEVC

Deblocking Filter and Adaptive Loop Filter

Y Hae-cheol Choi / Hanbat University

summary

HEVC (High Efficiency Video Coding) is currently being standardized. It is the tentative name for a new video encoding standard. In this standardization To achieve high encoding performance beyond H.264/AVC, various Laws are being discussed, among them deblocking filter and adaptive I would like to explain loop filter technology in this article. technically The deblocking filter and adaptive loop filter perform filtering on the restored image to reduce information lost during quantization and encoding operations. It is a technology to improve subjective image quality by performing

I. Introduction

ISO/IEC JCT1/SC29 WG11 Moving Picture Experts Group(MPEG) and ITU-T Q.6/16 Video Coding Experts Group (VCEG) has come together to create a new

Standardization of video encoding is in progress. This was tentatively named High Efficiency Video Coding (HEVC), and was announced through a Call for Proposal in January 2010. and began formal standardization. These standardization The group actively encourages research and contributions of new video coding algorithms for the HEVC standard. there is.

HEVC standardization at the Guangzhou meeting in October 2010 Seo established the first test model by combining algorithms that had been verified through competition. This It was tentatively named HM (High Efficiency Video Coding Test Model), and a coding tool for HM 1.0 has currently been selected. Technical documentation for HM 1.0 can be found here: As of November 2010, it has not been officially distributed. TMuC(Test Model under Consideration) [1]

contains most of the encoding tools of HM 1.0

If you want to get more detailed technical information, please refer to the TMuC document.

HM 1.0 includes various encoding algorithms.

Oh, there it is. Among them, deblocking filter and adaptive loop that perform filtering on the restored image.

The purpose of this paper is to explain the filter(ALF) technology.

C. This technology uses quantization and computer decimal calculation errors, etc.

This is a technology that adaptively performs filtering on restored images to reduce information loss that occurs due to

The meaning of the word 'adaptive' refers to the entire video.

Instead of applying the same filter, filters are applied differentially to areas where a lot of distortion occurs and areas where a lot of distortion occurs.

It has the meaning of performing turing.

Deblocking filter is an adaptive in-loop filtering method also included in H.264/AVC[2].

am. The deblocking filtering method of HM 1.0 is

The method jointly proposed by Tandberg, Ericsson, and Nokia is a method to solve the shortcomings of the high complexity of the deblocking filtering method of H.264/AVC while showing similar subjective image quality performance.

[3].

ALF inserts the post-processing filter [4] applied after decoding the video into the decoding process.

This is a filtering method within the Kin loop. The biggest feature of ALF is

This means that the optimal filter coefficient for each image is

transmitted for each slice. Therefore, the encoding efficiency is

improved by adaptively applying the optimal filter coefficient for each slice.

The rate can be improved significantly. ALF method in HM 1.0

It is based on Toshiba's ALF method [5][6] and Qualcomm's ALF[7]-[9] method.

II. Deblocking Filter

In this chapter, we comparatively analyze the deblocking filters of H.264/AVC and HM 1.0.

1. H.264 deblocking filter

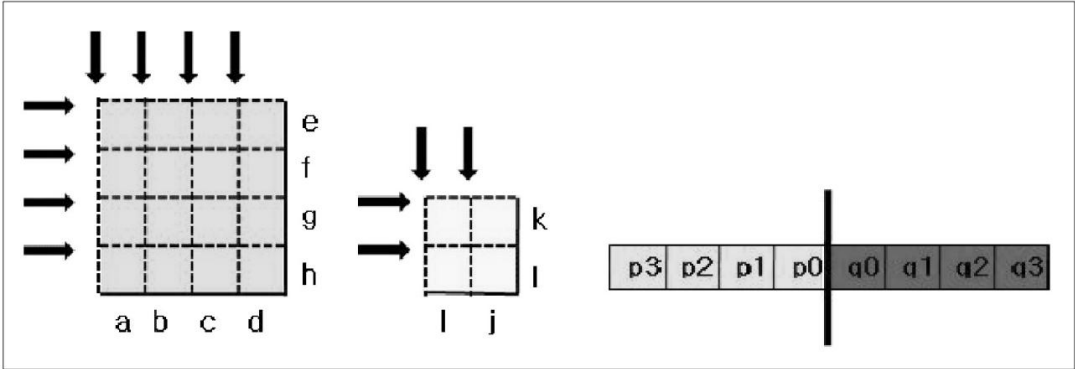
Deblocking Filter was first introduced as a standard in H.264/AVC. was adopted. Previously managed through post-processing filters



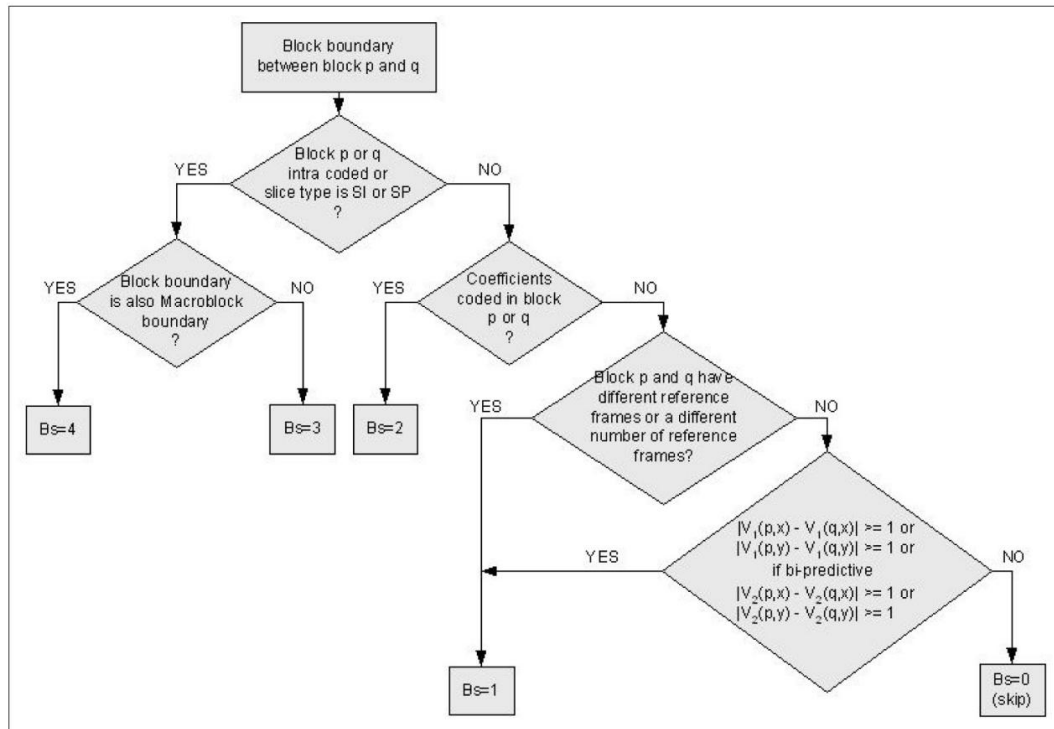
<Figure 1> (a) Image without deblocking filter applied (a), image with applied deblocking filter (b)

There have been various studies to improve image quality, but Filters are applied because they are not included in the gelatinization process. The video was not used as a reference video, but Video with deblocking filter applied in H.264/AVC By using it as a reference image, not only the subjective image quality is improved. It also leads to an improvement in objective image quality of around 0.5dB. paid it The biggest purpose of the deblocking filter is as follows <Figure Effect of blocking artifacts such as 1(a)> The objective is to improve subjective image quality by removing , this This blocking phenomenon is caused by DCT (discrete After applying cosine transform, some high frequency areas are removed. This inevitably occurs. Deblocking filter A low-pass filter is applied to the area where such distortion occurs. Apply it to create boundaries between blocks as shown in <Figure 1(b)>. Make it dull H.264/AVC deblocking filter is adaptive It's an enemy filter. That is, areas where distortion is likely to occur. Differentially apply filters to areas that do not Severe smoothing of the image due to excessive filtering Prevents the (smoothing) phenomenon. Technically, the deblocking filter in H.264/AVC is

Whether to apply a filter to block boundaries in 4x4 units decide Brightness of 16x16 size as shown in <Figure 2> For (luma) blocks and 8x8 color (chroma) blocks What happens to each dotted line in <Figure 2> in 4x4 block units? Boundary whether to apply a deblocking filter It is determined according to the Strength(Bs) value. Vertical in <Figure 2> The pixel index for the border is horizontally upward. Create a vertical border from two adjacent 4x4 blocks. Indicates the index for each of the 4 pixels on the left and right. It's a timeless painting. Bs when 8 pixels are given as shown in <Figure 2> The decision is based on the following <Figure 3>. border macro Is it a block boundary, or is the encoded block within the screen or Is inter-screen encoding, the movement of two adjacent blocks? Are the blocks different, are the pixels within the block highly similar, etc.? Based on the complex process as shown in <Figure 3>, Bs are decided. Ultimately, the Bs value at each boundary Therefore, the filter coefficients and filters are applied to change the image The number of cows is differentially applied to provide adaptive A deblocking filter is applied.



<Figure 2> Application of deblocking filter in 4x4 units and pixel index for vertical border



<Figure 3> How to Determine Boundary Strength(Bs)

2. HM 1.0 deblocking filter

The deblocking filter of HM 1.0 is a method that simplifies the complex process of the deblocking filter of H.264/AVC. The deblocking filter of H.264/AVC is a complex process that requires approximately 15% of the calculation amount of the entire decoding process in the decoder. Also, after H.264/AVC Residual signal due to improved performance of both prediction methods (Residual signal) does not occur much. Therefore, excessive filtering in 4x4 units is rather subjective or objective. This may cause deterioration in visual image quality. These disadvantages In order to overcome this, TENTM has relatively simplified A deblocking filter method was proposed, and this is HM 1.0

was adopted.

Basically, the deblocking filter of HM 1.0 is predictive.

Filtering is performed on the boundary between the prediction unit (PU) and the transformation unit (TU). However, it is the border of the image or has a width of 8 in the vertical or horizontal direction. No filtering is performed on boundaries smaller than a pixel. Complexity has been simplified by limiting

It is decided based on .

The method based on the first PU explains the vertical boundary based on <Figure 4>. For horizontal border The content is the same as the vertical border. <Figure 4> is one It represents the PU of . The rightmost vertical border is

In the next PU, not shown in the figure (to be encoded in the future)

Since the decision is made, filtering can be performed on each of the four boundaries in Figure 4, except as candidate boundaries.

The decision follows the following:

- For the first vertical border, there is a PU on the left
And the horizontal size of the PU to be currently encoded is 8.
Filtering can be performed when there are more than pixels.
•If the PU you want to currently encode is PART_nLx2N, the second vertical boundary is the first boundary and the second boundary.
Filtering is performed when the horizontal distance is more than 8 pixels.
can do. (Currently, asymmetric prediction partition technology has been excluded from HM 1.0,
In the case of PART_nLx2N, it may change later.)
Otherwise, the PU you are currently trying to encode
If PART_Nx2N or PART_NxN, three times
The second vertical boundary is the distance between the first boundary and the third boundary.
Perform filtering when horizontal distance is greater than 8 pixels

can do.

Otherwise, the PU you are currently trying to encode

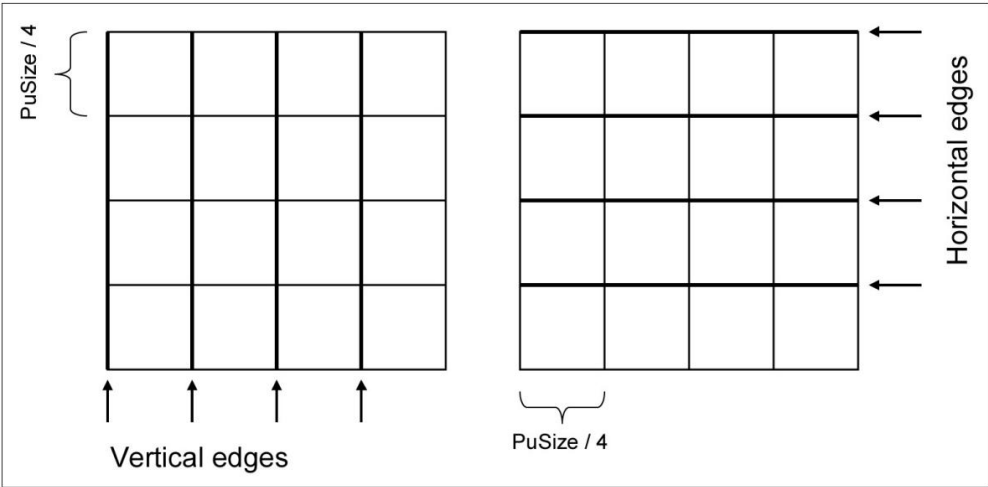
- If PART_nRx2N, the fourth vertical border is
The horizontal distance between the first border and the next PU is 8 pixels.
Filtering can be performed when (Currently, asymmetric prediction partition technology has been excluded from HM 1.0, so in the case of PART_nRx2N
It may change in the future.)

The method based on the second TU is

- The distance in the horizontal or vertical direction is less than 8 pixels.
Only the boundary that is the image is determined as a candidate boundary to apply the deblocking filter to. For example, if the minimum size of a TU is 4
If , filtering can be performed on every other boundary.
there

is. <Figure 4> shows the index of each pixel to explain the actual filtering process. After the filtering described above

Among the beam boundaries, the boundaries on which actual filtering will be performed are as follows:



<Figure 4> Determination of the boundary where filtering will be applied in one prediction mode block

A	p_{3_0}	p_{2_0}	p_{1_0}	p_{0_0}	q_{0_0}	q_{1_0}	q_{2_0}	q_{3_0}	B
	p_{3_1}	p_{2_1}	p_{1_1}	p_{0_1}	q_{0_1}	q_{1_1}	q_{2_1}	q_{3_1}	
	p_{3_2}	p_{2_2}	p_{1_2}	p_{0_2}	q_{0_2}	q_{1_2}	q_{2_2}	q_{3_2}	
	p_{3_3}	p_{2_3}	p_{1_3}	p_{0_3}	q_{0_3}	q_{1_3}	q_{2_3}	q_{3_3}	
	p_{3_4}	p_{2_4}	p_{1_4}	p_{0_4}	q_{0_4}	q_{1_4}	q_{2_4}	q_{3_4}	
	p_{3_5}	p_{2_5}	p_{1_5}	p_{0_5}	q_{0_5}	q_{1_5}	q_{2_5}	q_{3_5}	
	p_{3_6}	p_{2_6}	p_{1_6}	p_{0_6}	q_{0_6}	q_{1_6}	q_{2_6}	q_{3_6}	
	p_{3_7}	p_{2_7}	p_{1_7}	p_{0_7}	q_{0_7}	q_{1_7}	q_{2_7}	q_{3_7}	

〈그림 5〉 HM 1.0 deblocking filter를 위한 화소 색인

같이 결정한다.

밝기 블록에서 다음 조건들 중 하나라도 참인 경우 필터링을 수행한다.

- A 혹은 B 블록의 예측 모드가 화면내(INTRA) 부호화인 경우
- A 혹은 B 블록이 0 이 아닌 변환 계수 값을 갖는 경우
- 1/4 밝기 샘플 단위에서 움직임 벡터가 $|MV_h(A) - MV_h(B)| \geq 4$ or $|MV_v(A) - MV_v(B)| \geq 4$ 인 경우
- A 블록의 참조 영상에 대한 PBIC 값 (icScale 혹은 icOffset)이 B 블록의 값과 다른 경우
- $d = |p_{2_2} - 2p_{1_2} + p_{0_2}| + |q_{2_2} - 2q_{1_2} + q_{0_2}| + |p_{2_5} - 2p_{1_5} + p_{0_5}| + |q_{2_5} - 2q_{1_5} + q_{0_5}| < \beta$ 인 경우

하지만, 밝기 블록에서 다음 조건 중 하나라도 참인 경우 필터링을 수행하지 않는다.

- A 와 B 블록이 동일한 TU와 PU에 속하는 경우
- 경계가 PU의 경계이고 A와 B의 planar_flag 값이

1 인 경우 (현재 HM 1.0에서 planar 모드는 빠져 있으므로 이 경우는 추후에 변경될 수 있다.)

밝기 블록의 deblocking filter는 weak filter와 strong filter 두 개가 존재한다. 상기의 조건에 따라 필터링을 수행하도록 결정된 블록에 대해 다음 조건들이 모두 참인 경우 strong filter가 적용되며, 그렇지 않은 경우 weak filter가 적용된다.

- $d < (\beta \gg 2)$
- $(|p_{3_3} - p_{0_3}| + |q_{0_3} - q_{3_3}|) < (\beta \gg 3)$
- $|p_{0_3} - q_{0_3}| < ((5 \cdot t_c + 1) \gg 1)$

상기 조건에서 t_c 와 β 는 <표 1>과 같이 QP (quantization parameter)에 의해 결정된다. 또한 블록 A 혹은 블록 B가 화면내 부호화 모드인 경우, $t_c = t_c(qp+4)$ 이다.

밝기 블록의 strong filter는 다음의 필터 계수 값을 갖는다.

- $p_{0_0} = \text{Clip}_{-255}((p_{2_2} + 2p_{1_2} + 2p_{0_2} + 2q_{0_0} + q_{1_0} + 4) \gg 3)$

<Table 1> The relationship between QP, and γ IC.

qp	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
t_c	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	7
qp	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
tc	1	1	1	1	1	1	1	1	1	2	2	2	2	3	3	3	3	4
β	8	9	10	11	12	13	14	15	16	17	18	20	22	24	26	28	30	32
qp	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
tc	4	4	5	5	6	6	7	8	9	9	10	10	11	11	12	12	13	13
β	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62	64		
qp	54	55																
tc	14	14																
β																		

$\bullet q_{00} = \text{Clip0-255}((p_{1i} + 2 \cdot p_{0i} + 2 \cdot q_{0i} + 2 \cdot q_{1i} + q_{2i} + 4) \gg 3)$

$\bullet p_{10} = \text{Clip0-255}((p_{2i} + p_{1i} + p_{0i} + q_{0i} + 2) \gg 2)$

$\bullet q_{10} = \text{Clip0-255}((p_{0i} + q_{0i} + q_{1i} + q_{2i} + 2) \gg 2)$

$\bullet p_{20} = \text{Clip0-255}((2 \cdot p_{3i} + 3 \cdot p_{2i} + p_{1i} + p_{0i} + q_{0i} + 4) \gg 3)$

$\bullet q_{20} = \text{Clip0-255}((p_{0i} + q_{0i} + q_{1i} + 3 \cdot q_{2i} + 2 \cdot q_{3i} + 4) \gg 3)$

The week filter of the brightness block uses the following filter coefficient values:
have

$\bullet \ddot{y} = \text{Clip}(-t_c, t_c, (13 \cdot (q_{0i} - p_{0i}) + 4 \cdot (q_{1i} - p_{1i}) - 5 \cdot (q_{2i} - p_{2i}) + 16) \gg 5))$

$\bullet p_{0i} = \text{Clip0-255}(p_{0i} + \ddot{y})$ $\bullet q_{0i} = \text{Clip0-255}(q_{0i} - \ddot{y})$

$\bullet p_{1i} = \text{Clip0-255}(p_{1i} + \ddot{y}/2)$

$q_{1i} = \text{Clip0-255}(q_{1i} - \ddot{y}/2)$

Comparing the two filters above, strongfilter is
Filtering is applied by 3 pixels on the left and right based on the border.
Change the value, week filter has less left and right 2 picks

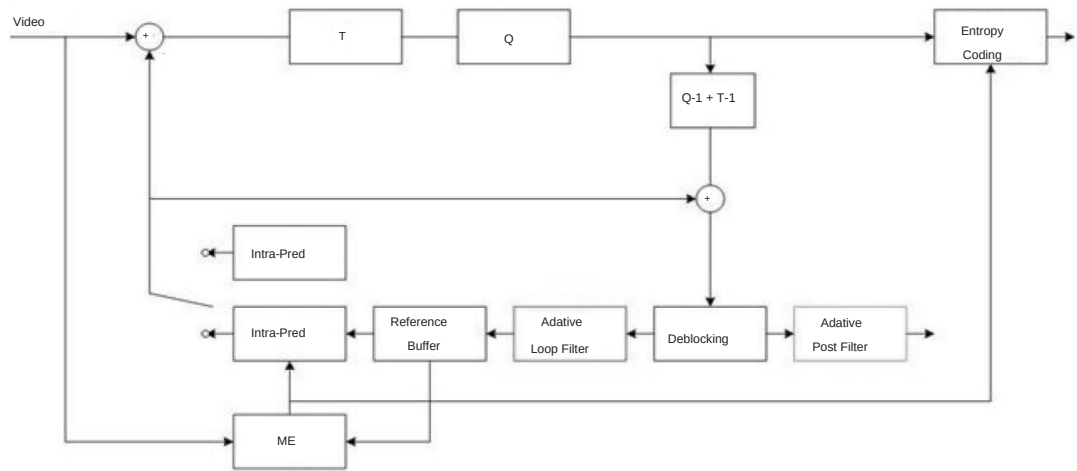
Apply filters cell by cell. Therefore, image quality distortion occurs.
Pixels that are likely to be affected are strongly filtered to reduce blocking,
while blocks that are not affected are subjected to excessive smoothing.
Performs weak filtering while preventing anger.
Color blocks are block A or block B on the border.
Filtering is performed only if the screen is encoded, but the
boundary is within one PU or
If planar_flag has a value of 1, filtering is not performed. The
specific process of the deblocking filter of a color block is as
follows: only one pixel on the left and right of the border is counted.
decide

$\bullet \ddot{y} = \text{Clip}(-t_c, t_c, (((q_{0i} - p_{0i}) < 2) + p_{1i} - q_{1i} + 4) \gg 3))$

$\bullet p_{0i} = \text{Clip0-255}(p_{0i} + \ddot{y})$ $\bullet q_{0i} = \text{Clip0-255}(q_{0i} - \ddot{y})$

$\bullet t_c = t_c(q_p + 4)$

In a simulation experiment on the objective image quality of the HM 1.0
deblocking filter, depending on whether the deblocking filter is used or not,
The difference between PNSR and bit rate is insignificant, approximately 1%.
It was. This is because HM 1.0 supports predictive coding and dabbling.



<Figure 5> Adaptive Post Decoder including Filter

Because high coding efficiency has already been improved to close to BD-bitrate - 30% through lock encoding technology, the margin of objective coding efficiency that can be raised using deblocking filter is small, and the fundamental purpose of deblocking filter is to improve subjective picture quality. am. For reference, since the deblocking filter of the existing H.264/AVC is not implemented in TMuC, there is still no direct connection between the deblocking filter of HM 1.0 and that of H.264/AVC.

phosphorus comparisons were not reported. In conclusion, the current HM The advantage of the 1.0 deblocking filter is the reduction in complexity.

III. Adaptive Loop Filter

<Figure 5> shows ALF and adaptive post filter decoding. It indicates when and what input signals are received and performed in the process. The two filters have in common the reconstructed image after the deblocking filtering process.

An adaptive filter is applied to the picture to output the final decoded picture. These are Optimization that minimizes the difference between the original image and the restored image By designing and applying an adversarial filter, some of the information lost during the encoding process can be recovered. The difference between ALF and adaptive post filter is that the ALF method is included in the decoding process, so the image to which ALF is applied can be used as a reference image in predictive encoding, but the adaptive postfilter method is included in the decoding process. Only filtering is applied and the resulting image is used as a reference image. The point is that it cannot be utilized. The ALF of HM 1.0 is based on Toshiba's BALF method and adopted by Qualcomm in a form developed in accordance with the coding unit structure of HM 1.0. It was. In the following sections, Toshiba's method and Qualcomm's Explain the method.

1. BALF

BALF(Block-based adaptive loop filter)y

It is an ALF method based on Wiener filter. Calculate the filter coefficients
Even considering the amount of additional bits for transmission, it is
difficult to find and apply one optimal filter to the entire video sequence.
Rather than designing the optimal filter differently for each video,
Applying it can increase overall performance. Therefore, in an encoder
including ALF, every slice
Differences between the original slice and the decrypted slice
Find the optimal filter that minimizes and transmit the filter coefficients
by including them in the slice header. In the decoder, filtering is
performed using the transmitted filter coefficients.
Perform the ring.

The method of finding the optimal filter follows the Wiener filter method.
In other words, if the pixel value at the (x,y) position where the filter is
applied is called pALF(x,y) , as shown in Equation 1 below, the difference
between this and the original image pixel value original(x,y) is defined as
the error, and the error is Design filter coefficients to minimize the sum of squares
do.

$$\begin{aligned} p_{ALF}(x,y) &= \sum_i \sum_j h(i,j) \cdot p_{reconstructed}(x+i,y+j) + DCoffset \\ \min_{x,y} \sum_x \sum_y \left(p_{original}(x,y) - p_{ALF}(x,y) \right)^2 \end{aligned} \tag{1}$$

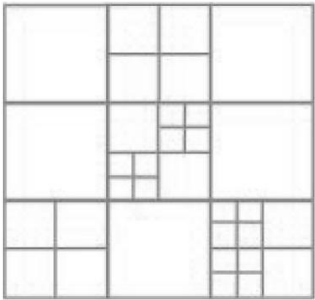
BALF applies a filter to all pixels in the image.
It doesn't turn on. 8x8, 16x16,... ,Filtering can be applied based on
blocks of various sizes, such as 128x128.
Improves performance by adaptively determining (filter on/off)
there is. However, for each block, additional bits are needed to
transmit information about whether or not the filter can be applied. To
reduce this, we proposed a quadtree-based adaptive filter (QALF)
that divides the image hierarchically and informs whether or not a
filter can be applied to it.

2. HM 1.0 Adaptive Loop Filter

The ALF of HM 1.0 is also optimized using Equation 1 above.

The system is designed based on Wiener filter. Basically
It is based on the QALF method, and the differences are the syntax for indicating
whether or not to apply a filter at the block level, the shape of the filter, and the
and the number of filters, etc.
The syntax for indicating whether or not to apply a filter uses the
quad-tree based block division technique of QALF. Using this, filtering
can be done hierarchically as shown in <Figure 6>.
It may or may not apply. Split the block into smaller pieces
As the number increases, additional information for filter application
increases. That is why the Coding Tree Block of HM 1.0
Additional information by designing the syntax according to (CTB)
We tried to reduce the depth of CU (coding uint) that can divide blocks
in ALF.

Separately for each slice the optimal depth value that limits
send. The maximum value of this optimal depth information is equal
to the maximum depth value in CTB, and a smaller value is transmitted.
It can be. For example, in CTB the maximum depth is 3.
However, the maximum depth of AFL can be set to 2. As explained
earlier, the filter performance increases as it is divided into smaller
sizes, but since syntax information about whether or not the filter can
be applied also increases, the depth value that achieves optimal
performance may be different from the maximum depth value of the CTB.
The filter shape for the brightness block of HM1.0ALF is <that.



<Figure 7> Quadtree based ALF's Example of ON/OFF adjustment

It has a diamond shape as shown in Figure 7, and has the numbers written in the picture.

The ruler pairs filter tabs with the same filter coefficients.

It is. Set up a symmetrical filter around the center point.

The number of unique filter coefficients was reduced by calculating

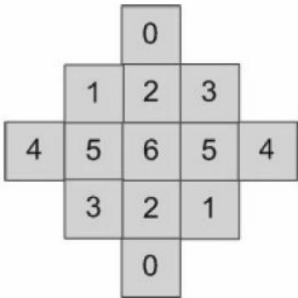
Also, when encoding filter coefficients, the filters of previous images

Filter system by performing predictive coding using coefficients

The amount of bits for transmitting numbers has been reduced. HM 1.0

ALF has up to M filter sets per slice.

The maximum value of M is 16.



<For picture brightness block

Filter shape and filter coefficient symmetry in ALF

Regarding ALF for color blocks, alf_

A syntax called chroma_idc is encoded. This is the next 4

It has branch values.

- 0 = Do not apply ALF,
- 1 = Apply ALF only to Cb
- 2 = Apply ALF only to Cr

- 3 = Apply ALF to both Cb and Cr

As can be seen from the alf_chroma_idc value, ALF

In addition to filtering on the two color components,

Performing filtering only on my color components is also possible.

Allowed. The filter shape for the color block is the brightness block

A square shape with 5 tabs, unlike the filter applied to

It is a non-separable filter.

The performance of ALF was reported in [10], as shown in Table 2.

Currently, the common experimental conditions of JCTVC are high complexity

In conditions, ALF is the default condition and low

ALF is not included in the complexity condition.

Therefore, in Table 2, the enemy of ALF for high complexity is

The BD-bitrate value depending on the presence or absence of use has a positive value.

This indicates that the performance of ALF is excellent, and a negative value indicates excellent performance for low com-plexity.

am. As shown in Table 2, ALF is very high.

It has coding efficiency. In terms of complexity, the decoder

It was reported that the increase in decoding time was approximately 15%.

all. In [10], the time increase for the encoder is also presented.

However, it was an experiment conducted with distributed processing in a cluster environment.

Therefore, it cannot be considered a meaningful number.

IV. conclusion

In this paper, the deblocking filer included in HM 1.0 and

<Table 2> HM 1.0 Coding efficiency of ALF

	Highcomplexity (BD-bitrate)	Lowcomplexity (BD-bitrate)
Intra	2.2 % (max 4.2% for classA) 3.7	- 2.2 %
Random access	% (max 5.8% for class B) 5.2 %	- 4.8%
Low delay	(max 8.5% for classE)	- 4.3%

The technology of adaptive loop filter (ALF) was explained.	ALF in 1.0 is a new in-loop post-processing filter. a little
The deblocking filter of HM 1.0 is more complex than H.264/AVC.	Requires complex calculations, but has very high encoding efficiency
It can be seen as a version with reduced noise, which predicts	This makes it very useful in applications for storage media.
Due to the development of coding technology, complex deblocking filters	I think it is a simple encryption tool. However, it has relatively high complexity.
This can be seen as a result of the need having decreased somewhat. H.M.	, so further improvements will be needed.

references

[1] Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Test Model under Consideration, Doc.JCTVC-B205, July 2010.

[2] ITU-T and ISO/IEC JTC 1, Advanced Video Coding for Generic Audiovisual Services, ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG4-AVC), Fourth edition: Sep. 2008.

[3] Kemal Ugur, Kenneth R. Andersson, and Arild Fuldseth, Description of video coding technology proposal by Tandberg, Nokia, Ericsson, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Doc. A119, Jan. 2010.

[4] K.K. Peng and T.K. Tan, Optimum loop filter in hybrid coders, IEEE Trans. CSVT, vol.4, no.2, pp.158-167, April 1994.

[5] Takeshi Chujoh, Goki Yasuda, Naofumi Wada, Takashi Watanabe, and Tomoo Yamakage, Block-based Adaptive Loop Filter, ITU-T SG16/Q.6 Doc. VCEG-AI18, July 2008.

[6] Takeshi Chujoh, Naofumi Wada, Takashi Watanabe, Goki Yasuda, and Tomoo Yamakage, Specification and experimental results of Quadtree-based Adaptive Loop Filter, ITU-T SG16/Q.6 Doc. VCEG-AK22, April 2009.

[7] Marta Karczewicz, Giovanni Motta, Yan Ye, and Peisong Chen, Post-filter with reduced delay requirements, ITU-T SG16 Contribution, C128, Geneva, Jan. 2009.

[8] Wei-Jung Chien, and Marta Karczewicz, Adaptive Filter Based on Combination of Sum-Modified Laplacian Filter Indexing and Quadtree Partitioning, ITU-T SG16/Q.6 Doc. VCEG-AL27, July 2009.

[9] Marta Karczewicz, Peisong Chen, Rajan Joshi, Xianglin Wang, Wei-Jung Chien, and Rahul Panchal, Video coding technology proposal by Qualcomm Inc., Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Doc. A121, Jan. 2010.

[10] In Suk Chong, and Marta Karczewicz, TE 12: Evaluation of Adaptive in-loop filter, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Doc. C235, Oct. 2010.

About the author



Choi Hae-cheol

1997 Bachelor of Electronic Engineering, Kyungpook National University

Master of Electrical and Electronic Engineering, Korea Advanced Institute of Science and Technology, 1999

2004 Ph.D., Department of Electrical and Electronic Engineering, Korea Advanced Institute of Science and Technology

September 2004 February 2010 Senior Researcher, Electronics and Telecommunications Research Institute

March 2007 February 2010 Adjunct Assistant Professor, United Graduate School of Science and Technology

March 2010 Assistant Professor, Department of Information, Communication and Computer Engineering, Hanbat National University

- Areas of interest: video coding, image processing and communication convergence

Deblocking Filter 및 Adaptive Loop Filter

□ 최해철 / 한밭대학교

요약

HEVC(High Efficiency Video Coding)는 현재 표준화가 진행되고 있는 새로운 비디오 부호화 표준의 가칭이다. 이 표준화에서는 H.264/AVC를 넘어서 높은 부호화 성능을 갖기 위해서 다양한 방법들이 논의되고 있으며, 그 중에서 deblocking filter 및 adaptive loop filter 기술에 대해 본 고에서 설명하고자 한다. 기술적으로 deblocking 필터와 adaptive loop filter는 양자화 및 부호화 연산과정에서 손실되는 정보를 줄이기 위해 복원된 영상에 필터링을 수행함으로써 주관적 화질을 향상시키기 위한 기술이다.

I. 서론

ISO/IEC JCT1/SC29 WG11 Moving Picture Experts Group(MPEG)과 ITU-T Q.6/16 Video Coding Experts Group(VCEG)은 함께 새로운 비

디오 부호화 표준화를 진행 중이다. 이는 High Efficiency Video Coding(HEVC)으로 잠정적으로 명명되었으며, 2010년 1월 Call for Proposal을 받고 공식적인 표준화를 시작하였다. 이들 표준화 그룹은 HEVC 표준을 위한 새로운 비디오 부호화 알고리즘의 연구 및 기고를 적극적으로 독려하고 있다.

2010년 10월 광저우 회의에서 HEVC 표준화에서는 그 동안 경쟁을 통해 검증된 알고리즘들을 조합하여 첫 번째 테스트 모델을 설정하였다. 이를 HM(High Efficiency Video Coding Test Model)으로 가칭했으며, 현재 HM 1.0에 대한 부호화 틀을 선택한 상태이다. HM 1.0에 대한 기술적인 문서는 2010년 11월 현재 공식적으로 배포되지 않았으나 TMuC(Test Model under Consideration) 문서[1]

※ 본 고는 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2010-0023109)

가 대부분의 HM 1.0의 부호화 툴을 포함하고 있기에, 보다 자세한 기술적 정보를 얻고자 한다면 TMuC 문서를 참조하기 바란다.

HM 1.0에는 다양한 부호화 알고리즘들이 포함되어 있다. 그 중에서 복원된 영상에 대해 필터링을 수행하는 deblocking filter 및 adaptive loop filter(ALF) 기술에 대해 본 고에서 설명하고자 한다. 이 기술은 양자화 및 컴퓨터의 소수 계산 오차 등으로 발생하는 정보의 손실을 줄이기 위해서 복원된 영상에 적응적으로 필터링을 수행하는 기술이다. 여기서 ‘적응적’이란 말의 의미는 영상 전체에 대하여 동일한 필터를 적용하지 않고, 왜곡이 많이 발생하는 영역과 그렇지 않은 영역에 대해 차별적으로 필터링을 수행한다는 뜻을 갖고 있다.

Deblocking filter는 H.264/AVC[2]에서도 포함되어 있는 적응적인 루프내(in-loop) 필터링 방법이다. HM 1.0의 deblocking filtering 방법은 H.264/AVC의 deblocking filtering 방법이 갖고 있는 높은 복잡도에 대한 단점을 해결하면서도 유사한 주관적 화질 성능을 보이기 위한 방법으로써 Tandberg, Ericsson 및 Nokia가 함께 제안한 방법이

다[3].

ALF는 영상을 복호화(decoding)한 후 적용되던 후처리(post) 필터[4]를 복호화 과정 안으로 삽입시킨 루프내 필터링 방법이다. ALF의 가장 큰 특징은 각 영상에 대해 최적의 필터 계수를 매 슬라이스마다 전송한다는 것이다. 따라서 슬라이스 별로 최적의 필터 계수를 적응적으로 적용함으로써 부호화 효율을 상당히 향상시킬 수 있다. HM 1.0의 ALF 방법은 Toshiba의 ALF 방법 [5] [6]과 Qualcomm의 ALF [7]–[9]방법에 기반한다.

II. Deblocking Filter

본 장에서는 H.264/AVC와 HM 1.0의 deblocking filter에 대해 비교 분석한다.

1. H.264 deblocking filter

Deblocking Filter는 H.264/AVC에서 처음 표준으로 채택되었다. 이전까지 후처리 필터를 통해 주관



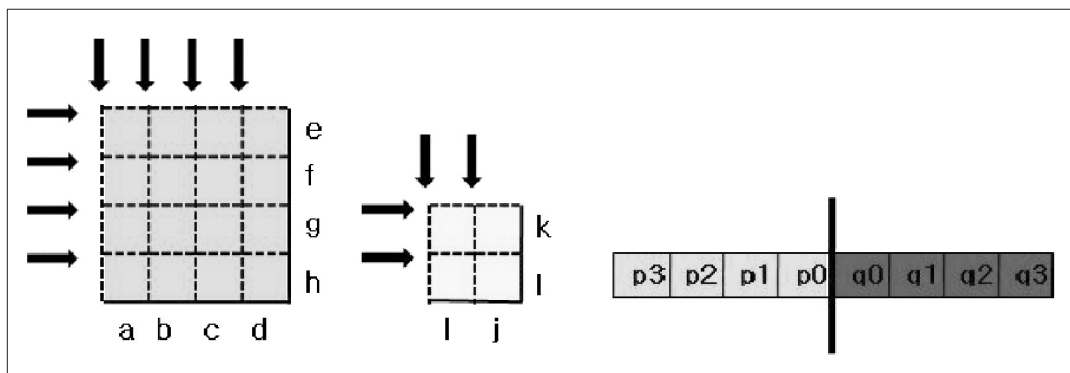
〈그림 1〉 (a) deblocking filter를 적용하지 않은 영상(a), 적용한 영상(b)

적 화질을 향상시키려는 다양한 연구가 있었으나 복호화 과정내에 포함되지 않았기 때문에 필터가 적용된 영상이 참조 영상으로 활용되지 않았다. 하지만 H.264/AVC에서는 deblocking filter가 적용된 영상을 참조 영상으로 활용함으로써 주관적 화질 뿐만이 아니라 0.5dB 내외의 객관적 화질의 향상도 이끌어냈다. Deblocking filter의 가장 큰 목적은 다음 <그림 1(a)>와 같은 블록화 현상(blocking artifact)의 효과를 제거하여 주관적 화질을 향상시키는 것이다. 이러한 블록화 현상은 부호화 과정에서 DCT(discrete cosine transform) 적용 후 고주파 영역을 일부 제거함으로써 불가피하게 발생한다. Deblocking filter는 이러한 왜곡이 발생하는 영역에 대해 저대역 필터를 적용하여 <그림 1(b)>와 같이 블록간의 경계를 부드럽게 한다. H.264/AVC의 deblocking filter는 적응적 필터이다. 즉 왜곡이 발생할 가능성이 높은 영역과 그렇지 않은 영역에 대해 차별적으로 필터를 적용하여 과도한 필터링에 의한 영상의 심한 평활화(smoothing) 현상을 막는다.

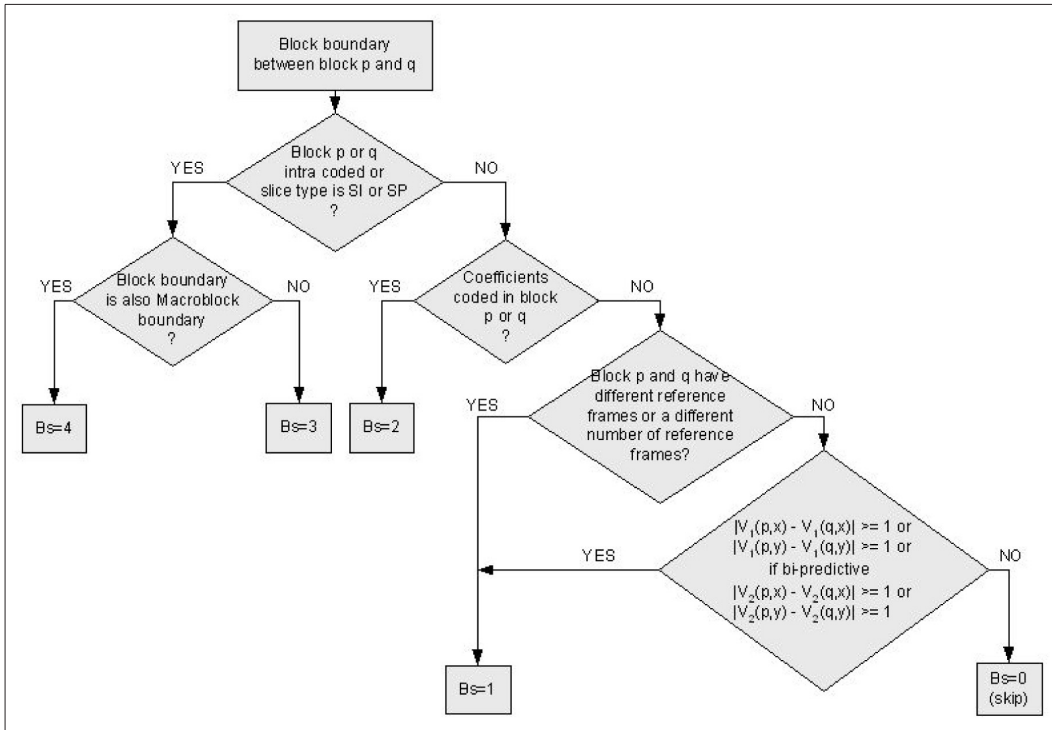
기술적으로, H.264/AVC의 deblocking filter는

4x4 단위의 블록 경계에 대해 필터의 적용 여부를 결정한다. <그림 2>와 같이 16x16 크기의 밝기(luma) 블록과 8x8 컬러(chroma) 블록의 경우 4x4 블록 단위로 <그림 2>의 각 점선에 대해 어떠한 deblocking filter를 적용시킬지는 Boundary Strength(Bs) 값에 따라 결정된다. <그림 2>의 수직 경계에 대한 화소 색인(index)은 수평 방향으로 위치한 두 개의 인접한 4x4 블록에서 수직 경계를 기준으로 좌우 각각 4 화소에 대한 색인(index)을 도시한 그림이다.

<그림 2>와 같이 8개의 화소들이 주어진 경우 Bs의 결정은 다음 <그림 3>에 따른다. 경계가 매크로 블록 경계인가, 부호화 화려는 블록이 화면내 혹은 화면간 부호화되었는가, 두 인접 블록의 움직임 정보가 상이한가, 블록 내의 화소 유사도가 높은가 등을 기준으로 <그림 3>과 같은 복잡한 과정을 거쳐 Bs를 결정하게 된다. 결국 각 경계에서의 Bs 값에 따라서 필터 계수 및 필터가 적용되어 변경되는 화소의 개수가 차별적으로 적용되어 적응적인 deblocking filter를 적용하게 된다.



<그림 2> 4x4 단위의 deblocking filter의 적용 및 수직 경계에 대한 화소 색인(index)



〈그림 3〉 Boundary Strength(Bs) 결정 방법

2. HM 1.0 deblocking filter

HM 1.0의 deblocking filter는 H.264/AVC의 deblocking filter의 복잡한 과정을 간소화시킨 방법이다. H.264/AVC의 deblocking filter는 복호기에서 전체 복호화 과정에서의 15% 내외의 연산량을 요구하는 복잡한 과정이다. 또한 H.264/AVC 이후 다양한 예측 방법의 성능 향상으로 인해 잔여 신호(residual signal)가 많이 발생하지 않는다. 따라서 4x4 단위의 과도한 필터링은 오히려 주관적 혹은 객관적 화질의 열화를 발생시킬 수 있다. 이러한 단점을 극복하고자 TENTM에서 비교적 간소화된 deblocking filter 방법을 제안했으며 이것이 HM 1.0

으로 채택되었다.

기본적으로 HM 1.0의 deblocking filter는 예측 유닛(prediction unit, PU)과 변환 유닛(transform unit, TU)의 경계에 대해 필터링을 수행한다. 다만 영상의 경계이거나 수직 혹은 수평 방향으로 그 폭이 8 픽셀보다 작은 경계에 대해서는 필터링을 수행하지 않도록 제한하여 복잡도를 간소화 시켰다. 필터링이 가해지는 경계의 구체적인 결정 방법은 PU 및 TU를 기준으로 결정한다.

첫 번째 PU를 기준으로 한 방법은 〈그림 4〉를 기준으로 수직 경계에 대해 설명한다. 수평 경계에 대한 내용은 수직 경계와 동일하다. 〈그림 4〉는 하나의 PU를 나타낸 것이다. 가장 우측의 수직 경계는

그림에 도시되지 않은(미래에 부호화될) 다음 PU에서 결정하므로 제외하고 그림의 4의 4 개의 경계에 각각에 대해 필터링을 수행할 수 있는 후보 경계로의 결정은 다음을 따른다.

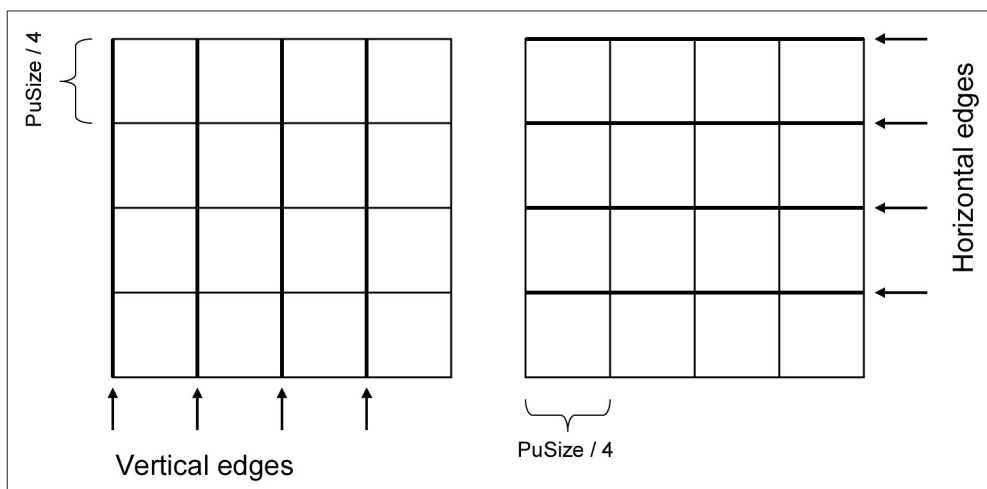
- 첫 번째 수직 경계의 경우, 좌측의 PU가 존재하고 현재 부호화 하려는 PU의 가로 크기가 8 픽셀 이상일 때 필터링을 수행할 수 있다.
- 현재 부호화 하려는 PU가 $PART_{nLx2N}$ 인 경우, 두 번째 수직 경계는 첫 번째 경계와 두 번째 경계의 수평 거리가 8 픽셀 이상일 때 필터링을 수행할 수 있다. (현재 HM 1.0에서 asymmetric prediction partition 기술이 제외되었으므로, $PART_{nLx2N}$ 의 경우는 추후에 변경될 수 있다.)
- 그렇지 않고 현재 부호화 하려는 PU가 $PART_{Nx2N}$ 혹은 $PART_{NxN}$ 인 경우, 세 번째 수직 경계는 첫 번째 경계와 세 번째 경계의 수평 거리가 8 픽셀 이상일 때 필터링을 수행

할 수 있다.

- 그렇지 않고 현재 부호화 하려는 PU가 $PART_{nRx2N}$ 인 경우, 네 번째 수직 경계는 네 번째 경계와 다음 PU의 수평 거리가 8 픽셀 이상일 때 필터링을 수행할 수 있다. (현재 HM 1.0에서 asymmetric prediction partition 기술이 제외되었으므로, $PART_{nRx2N}$ 의 경우는 추후에 변경될 수 있다.)

두 번째 TU를 기준으로 한 방법은, 하나의 TU 안에서 수평 혹은 수직 방향으로의 거리가 8 픽셀 이상이 되는 경계에만 deblocking filter를 적용시킬 후보 경계로 결정한다. 예를 들어 TU의 최소 크기가 4인 경우, 하나 걸러 매 경계마다 필터링을 수행할 수 있다.

<그림 4>는 실제 필터링 과정을 설명하기 위한 각 화소의 색인을 도시하였다. 앞서 설명한 필터링 후보 경계들 중 실제 필터링을 수행할 경계를 다음과



<그림 4> 하나의 예측 모드 블록에서의 필터링이 가해질 경계 결정

A	p_{3_0}	p_{2_0}	p_{1_0}	p_{0_0}	q_{0_0}	q_{1_0}	q_{2_0}	q_{3_0}	B
	p_{3_1}	p_{2_1}	p_{1_1}	p_{0_1}	q_{0_1}	q_{1_1}	q_{2_1}	q_{3_1}	
	p_{3_2}	p_{2_2}	p_{1_2}	p_{0_2}	q_{0_2}	q_{1_2}	q_{2_2}	q_{3_2}	
	p_{3_3}	p_{2_3}	p_{1_3}	p_{0_3}	q_{0_3}	q_{1_3}	q_{2_3}	q_{3_3}	
	p_{3_4}	p_{2_4}	p_{1_4}	p_{0_4}	q_{0_4}	q_{1_4}	q_{2_4}	q_{3_4}	
	p_{3_5}	p_{2_5}	p_{1_5}	p_{0_5}	q_{0_5}	q_{1_5}	q_{2_5}	q_{3_5}	
	p_{3_6}	p_{2_6}	p_{1_6}	p_{0_6}	q_{0_6}	q_{1_6}	q_{2_6}	q_{3_6}	
	p_{3_7}	p_{2_7}	p_{1_7}	p_{0_7}	q_{0_7}	q_{1_7}	q_{2_7}	q_{3_7}	

〈그림 5〉 HM 1.0 deblocking filter를 위한 화소 색인

같이 결정한다.

밝기 블록에서 다음 조건들 중 하나라도 참인 경우 필터링을 수행한다.

- A 혹은 B 블록의 예측 모드가 화면내(INTRA) 부호화인 경우
- A 혹은 B 블록이 0 이 아닌 변환 계수 값을 갖는 경우
- 1/4 밝기 샘플 단위에서 움직임 벡터가 $|MV_h(A) - MV_h(B)| \geq 4$ or $|MV_v(A) - MV_v(B)| \geq 4$ 인 경우
- A 블록의 참조 영상에 대한 PBIC 값 (icScale 혹은 icOffset)이 B 블록의 값과 다른 경우
- $d = |p_{2_2} - 2p_{1_2} + p_{0_2}| + |q_{2_2} - 2q_{1_2} + q_{0_2}| + |p_{2_5} - 2p_{1_5} + p_{0_5}| + |q_{2_5} - 2q_{1_5} + q_{0_5}| < \beta$ 인 경우

하지만, 밝기 블록에서 다음 조건 중 하나라도 참인 경우 필터링을 수행하지 않는다.

- A 와 B 블록이 동일한 TU와 PU에 속하는 경우
- 경계가 PU의 경계이고 A와 B의 planar_flag 값이

1 인 경우 (현재 HM 1.0에서 planar 모드는 빠져 있으므로 이 경우는 추후에 변경될 수 있다.)

밝기 블록의 deblocking filter는 weak filter와 strong filter 두 개가 존재한다. 상기의 조건에 따라 필터링을 수행하도록 결정된 블록에 대해 다음 조건들이 모두 참인 경우 strong filter가 적용되며, 그렇지 않은 경우 weak filter가 적용된다.

- $d < (\beta > 2)$
- $(|p_{3_3} - p_{0_3}| + |q_{0_3} - q_{3_3}|) < (\beta > 3)$
- $|p_{0_1} - q_{0_1}| < ((5 \cdot t_c + 1) > 1)$

상기 조건에서 t_c 와 β 는 <표 1>과 같이 QP (quantization parameter)에 의해 결정된다. 또한 블록 A 혹은 블록 B가 화면내 부호화 모드인 경우, $t_c = tc(qp+4)$ 이다.

밝기 블록의 strong filter는 다음의 필터 계수 값을 갖는다.

- $p_{0_0} = \text{Clip}_{0 \sim 255}((p_{2_2} + 2p_{1_2} + 2p_{0_2} + 2q_{0_0} + q_{1_1} + 4) > 3)$

〈표 1〉 QP, t_c , 및 β 의 관계

qp	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
t_c	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
β	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	7
qp	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
t_c	1	1	1	1	1	1	1	1	1	2	2	2	2	3	3	3	3	4
β	8	9	10	11	12	13	14	15	16	17	18	20	22	24	26	28	30	32
qp	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
t_c	4	4	5	5	6	6	7	8	9	9	10	10	11	11	12	12	13	13
β	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62	64		
qp	54	55																
t_c	14	14																
β																		

- $q_0 = \text{Clip}_{0-255}((p_{1_i} + 2p_{0_i} + 2q_{0_i} + 2q_{1_i} + q_{2_i} + 4) \gg 3)$
- $p_{1_0} = \text{Clip}_{0-255}((p_{2_i} + p_{1_i} + p_{0_i} + q_{0_i} + 2) \gg 2)$
- $q_{1_0} = \text{Clip}_{0-255}((p_{0_i} + q_{0_i} + q_{1_i} + q_{2_i} + 2) \gg 2)$
- $p_{2_0} = \text{Clip}_{0-255}((2p_{3_i} + 3p_{2_i} + p_{1_i} + p_{0_i} + q_{0_i} + 4) \gg 3)$
- $q_{2_0} = \text{Clip}_{0-255}((p_{0_i} + q_{0_i} + q_{1_i} + 3q_{2_i} + 2q_{3_i} + 4) \gg 3)$

밝기 블록의 weak filter는 다음의 필터 계수 값을 갖는다.

- $\angle = \text{Clip}(-t_c, t_c, (13 \cdot (q_{0_i} - p_{0_i}) + 4 \cdot (q_{1_i} - p_{1_i}) - 5 \cdot (q_{2_i} - p_{2_i}) + 16) \gg 5))$
- $p_{0_i} = \text{Clip}_{0-255}(p_{0_i} + \angle)$
- $q_{0_i} = \text{Clip}_{0-255}(q_{0_i} - \angle)$
- $p_{1_i} = \text{Clip}_{0-255}(p_{1_i} + \angle/2)$
- $q_{1_i} = \text{Clip}_{0-255}(q_{1_i} - \angle/2)$

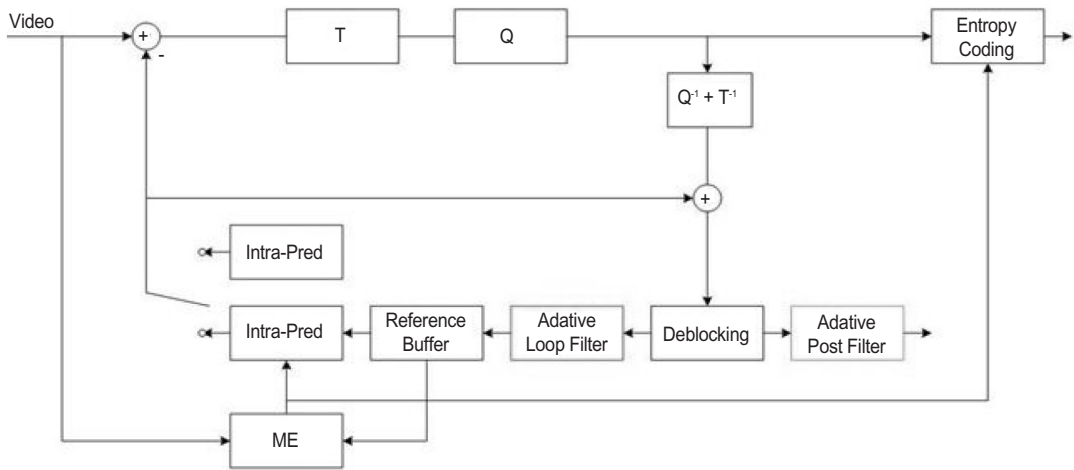
상기에서 두 개의 필터를 비교하면, strong filter는 경계를 기준으로 좌우 3 픽셀씩 필터링을 적용하여 값을 변경시키며, weak filter는 보다 적은 좌우 2 픽

셀씩 필터를 적용시킨다. 따라서 화질 왜곡이 발생할 가능성이 높은 화소는 강하게 필터링하여 블록화 현상을 줄여 주고, 그렇지 않은 블록은 과도한 평활화를 막으면서 약한 필터링을 수행한다.

컬러 블록은 경계에 있는 블록 A 혹은 블록 B가 화면내 부호화된 경우에만 필터링을 수행한다. 하지만 그 경계가 하나의 PU 내에 있거나 혹은 planar_flag가 1의 값을 갖는 경우는 필터링을 수행하지 않는다. 컬러 블록의 deblocking 필터의 구체적인 과정은 다음과 같이 경계의 좌우 하나의 픽셀만 수정한다.

- $\angle = \text{Clip}(-t_c, t_c, (((q_{0_i} - p_{0_i}) \ll 2) + p_{1_i} - q_{1_i} + 4) \gg 3))$
- $p_{0_i} = \text{Clip}_{0-255}(p_{0_i} + \angle)$
- $q_{0_i} = \text{Clip}_{0-255}(q_{0_i} - \angle)$
- $t_c = t_c(qp+4)$

HM 1.0 deblocking filter의 객관적 화질에 대한 모의 실험에서 deblocking filter의 사용 유무에 따른 PNSR 및 비트율의 차이는 1% 내외로 미미함을 보였다. 이러한 이유는 HM 1.0이 예측 부호화 및 대블



〈그림 5〉 Adaptive Loop Filter와 Adaptive Post Filter를 포함한 복호화기

록 부호화 기술 등에 의해 이미 BD-bitrate - 30%에 가까운 높은 부호화 효율 향상을 가져왔기 때문에 deblocking filter를 이용하여 끌어 올릴 수 있는 객관적 부호화 효율의 여유분이 작고, deblocking filter의 근본 목적은 주관적 화질 향상이기 때문이다. 참고로 TMuC에 기존 H.264/AVC의 deblocking filter가 구현되어 있지 않기 때문에 아직 HM 1.0의 deblocking filter와 H.264/AVC의 그것과의 직접적인 비교는 보고되지 않았다. 결론적으로 현재 HM 1.0 deblocking filter의 장점은 복잡도 감소라고 본다.

III. Adaptive Loop Filter

〈그림 5〉는 ALF와 adaptive post filter가 복호화 과정에서 언제, 어떤 입력 신호를 받아 수행되는지를 나타낸다. 두 필터는 공통적으로 deblocking filtering 과정 이후의 복원된 영상(reconstructed

picture)에 대해 적응적 필터를 적용하여 최종적인 복호된 영상(decoded picture)을 출력한다. 이들은 원영상과 복원된 영상간의 차이를 최소화시키는 최적 필터를 설계 및 적용하여 부호화 과정에서 손실되는 정보를 일부 복구할 수 있다. ALF와 adaptive post filter의 차이는 ALF의 방식은 복호화 과정 내에 포함되어 ALF가 적용된 영상이 예측 부호화에서 참조 영상으로 이용될 수 있지만 adaptive post filter는 필터링만 적용될 뿐 결과 영상이 참조 영상으로 활용되지 못하는 점이다. HM 1.0의 ALF는 Toshiba의 BALF 방법에 기반하여 Qualcomm이 HM 1.0의 부호화 유닛 구조에 맞추어 발전시킨 형태로 채택되었다. 다음 절에서 Toshiba의 방법과 Qualcomm의 방법에 대해 설명한다.

1. BALF

BALF(Block-based adaptive loop filter)는

Wiener 필터 기반의 ALF 방법이다. 필터 계수를 전송하기 위한 부가 비트량을 고려하더라도, 전체 영상열에 대해 하나의 최적 필터를 구하여 적용시키는 것보다는 매 영상마다 최적 필터를 달리 설계해서 적용시키는 것이 전체 성능을 높일 수 있다. 따라서 ALF를 포함한 부호기(encoder)에서는 매 슬라이스마다 원본 슬라이스와 복호된 슬라이스 사이의 차이를 최소화 시키는 최적 필터를 구하고, 그 필터 계수를 슬라이스 서두부(header)에 포함시켜 전송한다. 복호기(decoder)에서는 전송된 필터 계수로써 필터링을 수행한다.

최적 필터를 구하는 방식은 Wiener 필터 방식을 따른다. 즉 다음 식 1과 같이 필터가 적용된 (x,y) 위치의 화소 값을 $p_{ALF}(x,y)$ 라 하면, 이것과 원본 영상 화소 값 $p_{original}(x,y)$ 의 차이를 에러로 정의하고 이 에러의 제곱의 합이 최소가 되도록 필터 계수를 설계한다.

$$p_{ALF}(x,y) = \sum_i \sum_j h(i,j) \cdot p_{reconstructed}(x+i,y+j) + DCoffset$$

$$\min \sum_x \sum_y (p_{original}(x,y) - p_{ALF}(x,y))^2 \quad (1)$$

BALF는 영상내 모든 화소에 대해 필터를 적용시키지는 않는다. 8x8, 16x16, ..., 128x128와 같이 다양한 크기의 블록 기반으로 필터링의 적용 여부(filter on/off)를 적응적으로 결정하여 성능을 높이고 있다. 하지만 매 블록마다 필터 적용 여부에 대한 정보를 전송하기 위한 부가적인 비트를 필요하게 된다. 이를 줄이기 위해서 영상을 계층적으로 분할하고 이에 대한 필터 적용 여부를 알려 주는 quadtree-based adaptive filter(QALF)를 제안하였다.

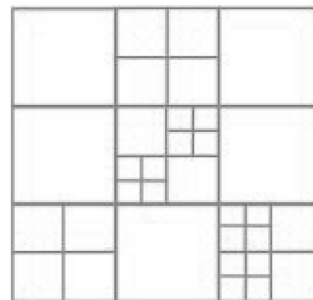
2. HM 1.0 Adaptive Loop Filter

HM 1.0의 ALF도 상기 식 1을 이용하여 최적 필

터를 Wiener filter 기반으로 설계한다. 기본적으로 QALF 방법에 기반하며 차이점은 블록 단위의 필터 적용 여부를 알려주기 위한 신택스, 필터의 모양, 그리고 필터의 개수 등이 있다.

필터 적용 여부를 알려주기 위한 신택스는 QALF의 quad-tree 기반의 블록 분할 기법을 사용한다. 이를 이용하면 <그림 6>과 같이 계층적으로 필터링을 적용 혹은 적용하지 않을 수 있다. 블록을 작게 분할할수록 필터 적용 여부를 위한 부가 정보가 증가한다. 그렇기 때문에 HM 1.0의 Coding Tree Block(CTB)에 맞추어 신택스를 설계함으로써 부가 정보를 줄이려 노력하였다. 구체적으로 설명하면, ALF에서는 블록을 분할할 수 있는 CU(coding unit)의 깊이를 제한하는 최적 깊이 값을 슬라이스마다 별도로 전송한다. 이 최적 깊이 정보의 최대 값은 CTB에서의 최대 깊이 값과 동일하며, 이보다 작은 값이 전송될 수 있다. 예를 들어 CTB에서는 최대 깊이가 3이었으나, AFL의 최대 깊이는 2로 설정할 수도 있다. 앞서 설명했듯이 작은 크기로 분할 할수록 필터의 성능은 올라가지만 필터 적용 여부에 대한 신택스 정보 또한 증가하기 때문에, 최적의 성능을 발휘하는 깊이 값은 CTB의 최대 깊이 값과 상이할 수 있다.

HM 1.0 ALF의 밝기 블록에 대한 필터 모양은 <그



<그림 7> Quadtree 기반 ALF의 ON/OFF 조정의 예

		0		
	1	2	3	
4	5	6	5	4
	3	2	1	
		0		

컬러 블록에 대한 ALF와 관련하여, alf_chroma_id라는 신택스가 부호화된다. 이는 다음 4가지 값을 갖는다.

- 0 = ALF를 적용하지 않음,
- 1 = Cb에만 ALF를 적용
- 2 = Cr에만 ALF를 적용

alf_chroma_idc 값에서 알 수 있는 바와 같이, ALF는 두 개의 컬러 성분에 대한 필터링뿐만 아니라 하나의 컬러 성분에 대해서만 필터링을 수행하는 것도 허용한다. 컬러 블록에 대한 필터 모양은 밝기 블록에 적용되는 필터와 달리 5 탭을 갖는 사각형 모양의 비분리형(non-separable) 필터이다.

IV. 결론

본 고에서는 HM 1.0에 포함된 deblocking filter와

	High complexity (BD-bitrate)	Low complexity (BD-bitrate)
Intra	2.2 % (max 4.2% for class A)	– 2.2 %
Random access	3.7 % (max 5.8% for class B)	– 4.8%
Low delay	5.2 % (max 8.5% for class E)	– 4.3%

Adaptive loop filter(ALF)의 기술에 대해 설명하였다. HM 1.0의 deblocking filter는 H.264/AVC에 비해 복잡도를 감소시킨 버전으로 볼 수 있으며, 이는 예측 부호화 기술의 발달로 인해 복잡한 deblocking filter의 필요성이 다소 감소했기 때문으로 볼 수 있다. HM

1.0의 ALF는 새로운 루프내 후처리 필터이다. 다소 복잡한 연산을 필요로 하지만 부호화 효율이 매우 높기 때문에 저장 매체를 위한 응용 분야에서 매우 유용한 부호화 툴이라고 본다. 하지만 비교적 높은 복잡도를 가지므로 이에 대한 개선이 더 필요할 것이다.

참고 문헌

- [1] Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Test Model under Consideration, Doc.JCTVC-B205, July 2010.
- [2] ITU-T and ISO/IEC JTC 1, Advanced Video Coding for Generic Audiovisual Services, ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG4-AVC), Fourth edition: Sep. 2008.
- [3] Kemal Ugur, Kenneth R. Andersson, and Arild Fuldseth, Description of video coding technology proposal by Tandberg, Nokia, Ericsson, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Doc. A119, Jan. 2010.
- [4] K.K. Peng and T.K. Tan, Optimum loop filter in hybrid coders, IEEE Trans. CSVT, vol.4, no.2, pp.158-167, April 1994.
- [5] Takeshi Chujoh, Goki Yasuda, Naofumi Wada, Takashi Watanabe, and Tomoo Yamakage, Block-based Adaptive Loop Filter, ITU-T SG16/Q.6 Doc. VCEG-AI18, July 2008.
- [6] Takeshi Chujoh, Naofumi Wada, Takashi Watanabe, Goki Yasuda, and Tomoo Yamakage, Specification and experimental results of Quadtree-based Adaptive Loop Filter, ITU-T SG16/Q.6 Doc. VCEG-AK22, April 2009.
- [7] Marta Karczewicz, Giovanni Motta, Yan Ye, and Peisong Chen, Post-filter with reduced delay requirements, ITU-T SG16 Contribution, C128, Geneva, Jan. 2009.
- [8] Wei-Jung Chien, and Marta Karczewicz, Adaptive Filter Based on Combination of Sum-Modified Laplacian Filter Indexing and Quadtree Partitioning, ITU-T SG16/Q.6 Doc. VCEG-AL27, July 2009.
- [9] Marta Karczewicz, Peisong Chen, Rajan Joshi, Xianglin Wang, Wei-Jung Chien, and Rahul Panchal, Video coding technology proposal by Qualcomm Inc., Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Doc. A121, Jan. 2010.
- [10] In Suk Chong, and Marta Karczewicz, TE 12: Evaluation of Adaptive in-loop filter, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Doc. C235, Oct. 2010.

필자 소개



최해철

- 1997년 : 경북대학교 전자공학과 학사
- 1999년 : 한국과학기술원 전기및전자공학과 석사
- 2004년 : 한국과학기술원 전기및전자공학과 박사
- 2004년 9월 ~ 2010년 2월 : 한국전자통신연구원 선임연구원
- 2007년 3월 ~ 2010년 2월 : 과학기술연합대학원대학교 겸임 조교수
- 2010년 3월 ~ 현재 : 한밭대학교 정보통신컴퓨터공학부 조교수
- 주관심분야 : 영상처리, 비디오 부호화, 방송통신 융합