

A Scalable Pattern Omission Method of Deblocking Filter for H.264/AVC Baseline Profile

Jae Gon Kim, Jin Woo Hwang and Jun-Dong Cho
Sung Kyun Kwan University

300 Cheoncheon-dong, Jangan-gu, Suwon, Gyeonggi-do 440-746, Korea

Tel: +82-31-290-7200, Fax: +82-31-290-7683

E-mail: gon9031@skku.edu

Abstract: In this paper, we present the so-called scalable pattern omission method for a deblocking filter. This exploits specific characteristics of the human visual system. The baseline of our method is to reduce processing time of the deblocking filter. In the development of the H.264/AVC decoder, the processing Deblocking filter consumes the most computation resource. We proposed a scalable pattern omission method during filtering to decrease macroblocks that are filtered by the deblocking filter. Simulation result shows that our method can achieved performance enhancement ranging about 20% to 70% in the processing time.

1. Introduction

H.264/AVC or MPEG4 part 10 is developed by the Video Coding Experts Group (VCEG) of ITU-T and the Moving Picture Experts Group (MPEG) of ISO/IEC [1]. Comparing with the existing video compression standard like H.263 or MPEG-2, H.264/AVC provides better coding efficiency than previous coding standards does. With 4x4 block size that is smaller than block size of other previous standard, integer discrete cosine transform, and adaptive in-loop deblocking filter, high coding efficiency could be achieved [2]. The main features of H.264/AVC are 1) introduction of advanced motion estimation, 2) improvement of entropy coding such as CAVLC (Content Adaptive Variable Length Coding) and CABAC (Content Adaptive Binary Arithmetic Coding), 3) advancement of Deblocking filter [3]. The deblocking filter reduces blocking artifacts that are produced by block-based video coding. In the previous version, the deblocking filter was not standardized. In H.264, the deblocking filter is positioned in a prediction loop to increase coding efficiency and improve video quality. However, the deblocking filter increases computational complexity in encoding and decoding.

In this study, we propose a scalable pattern omission method for the deblocking filter. This exploits specific characteristics of the human visual system. In general, humans concentrate on the center of screen when they watch a video and show an optical illusion. Research [4] shows that humans usually disregard outlines of an image frame in a video sequences. We designed a filtering method using this characteristic. Our method saves deblocking time by macroblocks omission in the picture outline.

The structure of this paper is as follows. Section 2 describes a pattern omission method for the deblocking filter and introduces various patterns. Section 3 elaborates on a simulation method to prove our idea and shows the result of this research. Finally, the conclusions and future work are presented in Section 4.

2. Pattern omission method

2.1 Method concept

A person's eye has one focus. When people stare at a sequence of image, their relative focus is sensitive to the center of the image rather than the periphery. Based on this fact, we attempted to skip deblocking filtering in outline of a frame or between frames.

Figure 1 shows a basic raster scan order, which is recommended to read macroblocks in a frame by JM software. In this case, we try to skip outline of a frame. Figure 2 shows new raster scan order. Level means a number of boundaries. In this example, square level 1 represents that we will skip one of boundary and square level 2 will skip two of boundaries. Originally, deblocking filter will filter 99 macroblocks when image sequence is QCIF format. However, in our proposal, square level 1 will filter 63 macroblocks, and square level 2 will filter only 35 macroblocks. The processing ratio of pattern omission method is 63% (level 1) and 35% (level 2), respectively, compared with JM software method. Also, we realized that our method decreases processing time dramatically.

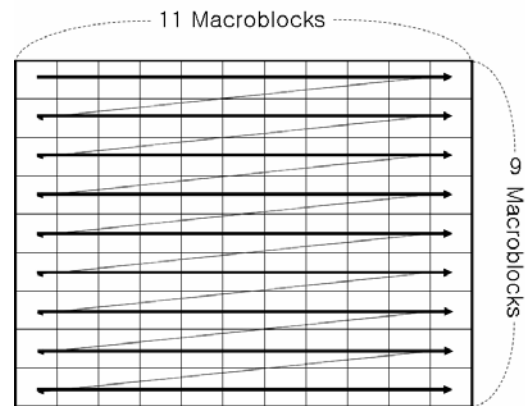


Figure 1. A basic raster scan order. QCIF format (11 x 9 Macroblocks, Macroblock = 16 x 16 pixels)

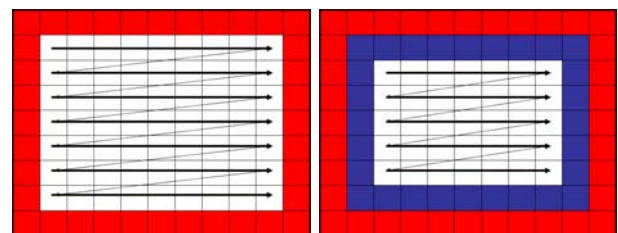


Figure 2. A raster scan order of pattern omission method. QCIF format (left: square level 1, right: square level 2)

2.2 Various omission patterns

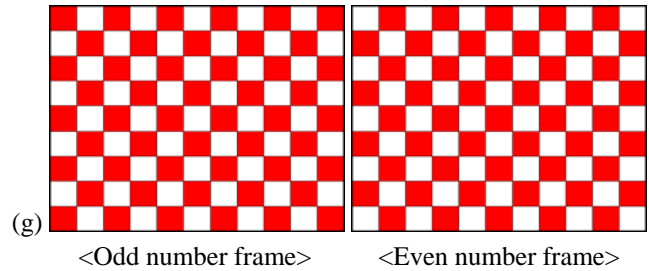
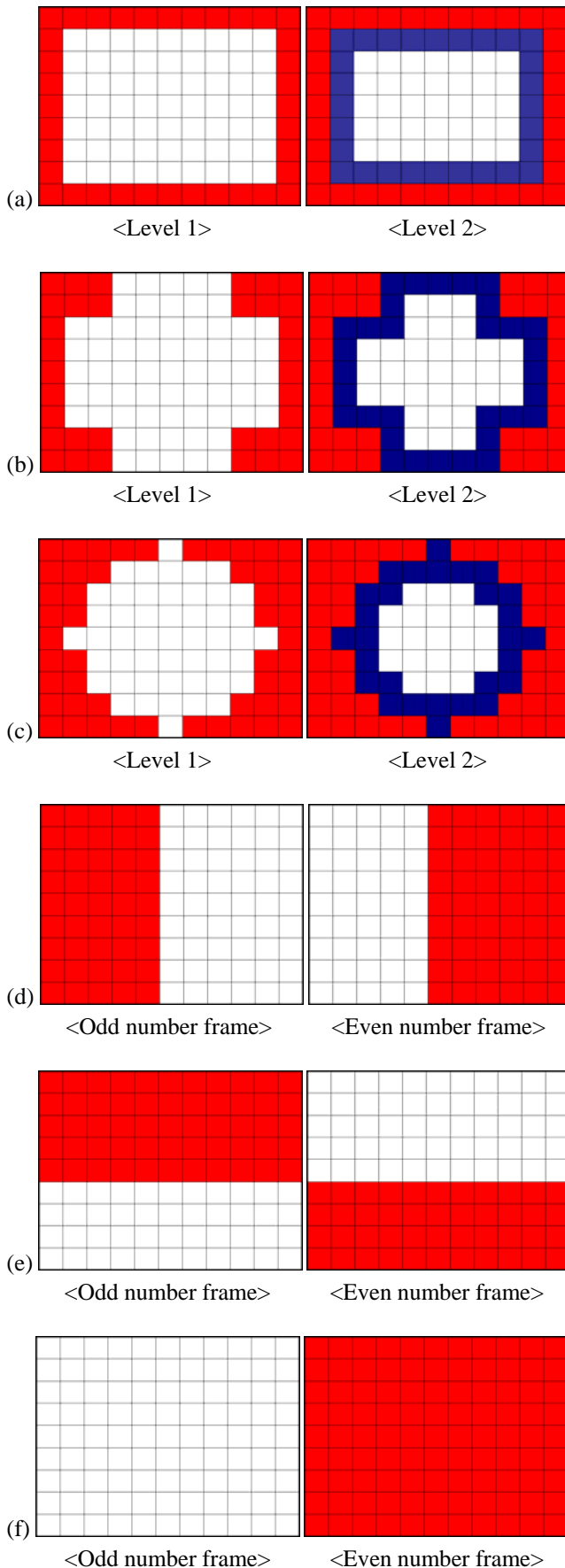


Figure 3. Various test patterns. QCIF format. Only white blocks will be filtered. (a) left: square level 1, right: square level 2 (b) left: Cross level 1, right: Cross level, (c) left: Circle level 1, right: Circle level 2, (d ~ g) Omission patterns during frame order, (d) half and half (left – right), (e) half and half (top – bottom), (f) one and one, (g) check

Figure 3 shows various omission test patterns. Test pattern type from (a) to (c) are designed by using specific characteristics of the human visual system: humans concentrate on the center of screen when they watch a video. Test pattern type from (d) to (g) are originated by using specific characteristics of the human visual system: an optical illusion.

3. Simulation method and result

3.1 Simulation method

The proposed skipping method is simulated on an Intel Pentium4 Processor 2.40GHz with 1024MB of memory. The simulation is run under Windows XP Operating System and compiled with Microsoft Visual C++ 8.0. In all of our experiments, six sequences (300 frames), i.e., foreman, coastguard, container, mobile, mother and daughter, news, and silent are used with QCIF format (176×144) and encoding to the baseline profile.

3.2 Simulation result

Figure 4 shows the result of simulation such that we could recognize some difference among images. In the magnified image of each result, the difference was larger. Among the sequences of image which is filtered with two different methods, we identified that the quality of most sequences were almost of the same.

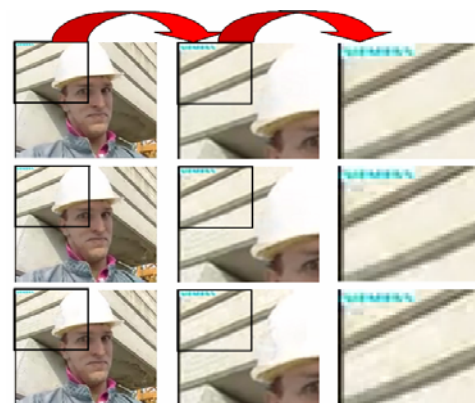


Figure 4. Result images of simulation. (top: JM software <reference>, middle: square level 1, bottom: square level 2)

Figure 5 shows the center area pixel value variation between images applying the different methods. Each image contains red, green, and blue components; three kinds of pixel values are exist. In Figure 5, we observed less difference among image pixel values. Comparing between JM software and square level 1, we discovered that 91% pixel values were of the same on the average. Comparing between JM software and square level 2, we discovered that 87% pixel values were of the same.

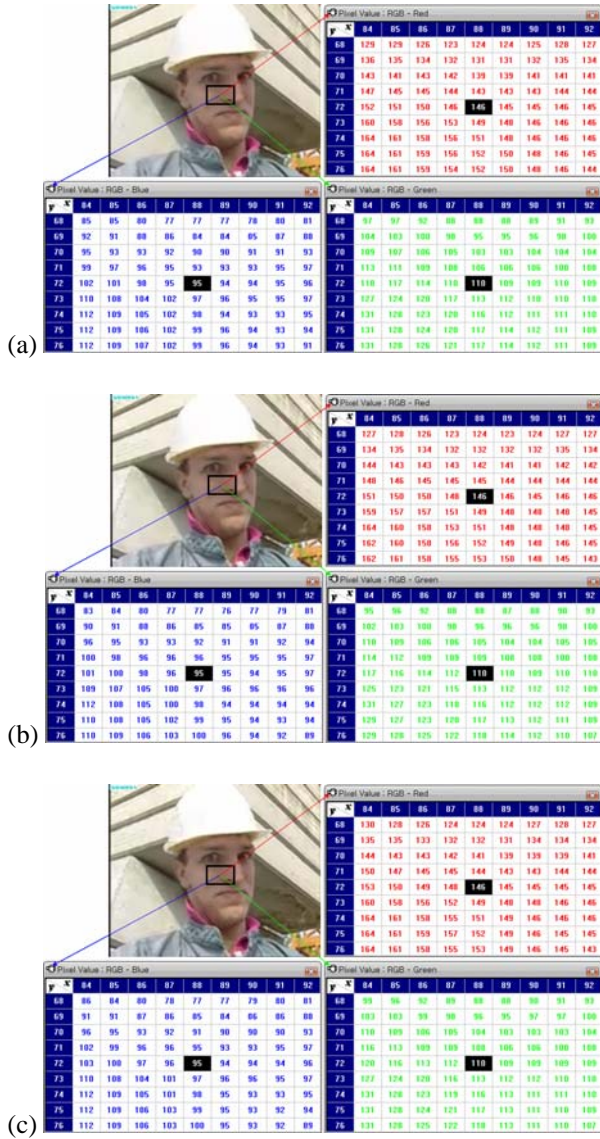


Figure 5. The variation of center area pixel values between images applying different methods. (a) JM software (b) Square level 1 (c) Square level 2

Figure 6 shows the boundary area pixel value's variation among images which was applied to different methods. In Figure 6, we observed a lot of difference among image pixel values. Comparing between JM software and square level 1, we discovered that 32% pixel values of the same were same. Comparing between JM software and square level 2, we discovered that 26% pixel values were of the same. However, when we simulated with new sequence image that shows less movement of action,

we observed that boundary area of an image frame were 80% of the same in case of square level 1 and 44% of the same in case of square level 2.

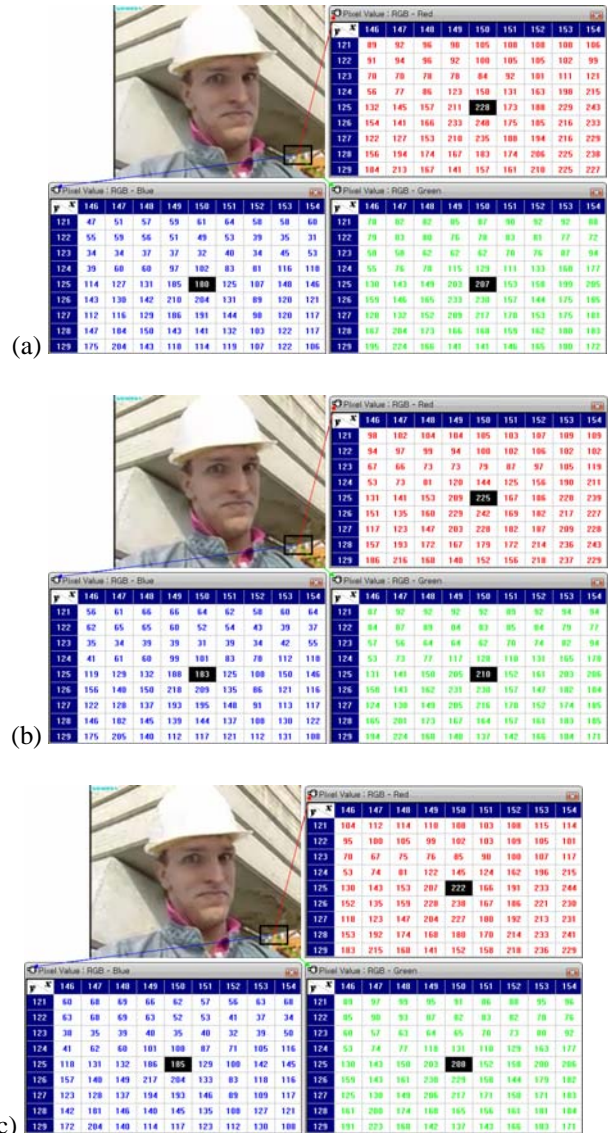


Figure 6. The variation of boundary area pixel values between images applying different methods. (a) JM software (b) Square level 1 (c) Square level 2

Table 1. Comparisons between square level 1 compare to JM software.

Sequence Name	Speed up ratio	PSNR of JM reference	PSNR of Square Level 1	PSNR difference
Foreman	28.3 %	37.12 db	35.44 db	-1.68 db
Coastguard	31.7 %	34.77 db	33.49 db	-1.28 db
Container	34.9 %	36.17 db	35.92 db	-0.25 db
Mobile	37.8 %	34.22 db	33.25 db	-0.97 db
Mother and daughter	38.1 %	37.66 db	36.64 db	-1.02 db
News	23.5 %	37.13 db	36.95 db	-0.18 db
Silent	23.8 %	36.19 db	35.58 db	-0.61 db

Table 1 shows the result PSNR and speed up ratio of square level 1. In table 1, simulation result shows that square level 1 can achieved performance enhancement ranging about 20% to 40% in the processing time.

Table 2. Comparisons between square level 2 compare to JM software.

Sequence Name	Speed up ratio	PSNR of JM reference	PSNR of Square Level 1	PSNR difference
Foreman	59.2 %	37.12 db	34.82 db	-2.30 db
Coastguard	63.6 %	34.77 db	33.09 db	-1.68 db
Container	56.6 %	36.17 db	35.50 db	-0.67 db
Mobile	61.5 %	34.22 db	33.01 db	-1.21 db
Mother and daughter	65.3%	37.66 db	35.96 db	-1.70 db
News	50.5 %	37.13 db	36.12 db	-1.01 db
Silent	40.7 %	36.19 db	35.17 db	-1.02 db

Table 2 shows the result PSNR and speed up ratio of square level 2. In Table 1 and Table 2, we discovered some facts as follow: sequence images, which shown a lot of movement (i.e. foreman, coastguard, and mobile), showed a lot of PSNR loss, but gained much speed up rate. However, sequence images, which shown less movement of action (i.e. container, news, and silent), showed less PSNR loss, but gained a little speed up rate. This result can be used for decision that a number of boundaries will be skipped.

Table 3. Comparisons among various omission patterns compare to JM software (Used sequence frame: Foreman).

Pattern Name	Speed up ratio (level 1)	PSNR difference (level 1)	Speed up ratio (level 2)	PSNR difference (level 2)
Square	28.3 %	-1.68 db	59.2 %	-2.30 db
Cross	22.2 %	-1.95 db	52.3 %	-2.56 db
Circle	12.7 %	-1.82 db	26.5 %	-2.22 db
Half and half (left – right)	58.1 %	-2.64 db	None	None
Half and half (top – bottom)	76.4%	-2.62 db	None	None
One and one	50.9 %	-1.78 db	None	None
Check	50.8 %	-1.77 db	None	None

Table 3 shows the result PSNR and speed up ratio of arious omission patterns. From Table 3, we can determine that complex pattern, i.e. Cross and Circle, shows lower performance than simple pattern. There are no difference between one & one and check. In half and half cases, we expected that both will show similar result, but performance was quite different. Omission order of half and half (top-bottom) pattern follows raster scan order.

Figure 7 shows simulation result graph of various omission patterns in table 3.

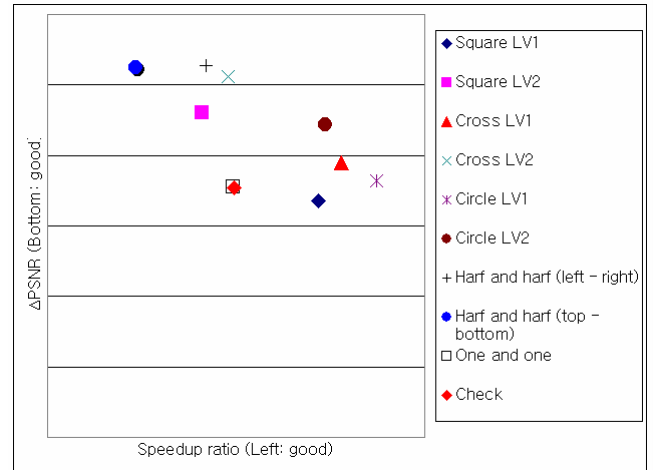


Figure 7. Simulation result graph of various omission patterns

In figure 7, we can find out that check pattern and one and one pattern are most useful patterns. Square level 1 pattern shows most good quality image. Half and half (top-bottom) pattern shows fastest processing time.

4. Conclusion

A scalable pattern omission method of deblocking filter is a new concept by using the specific characteristics of the human visual system. In this study we used QCIF format which is the smallest standard image format. However, actually, this method may apply large size of image format more efficiently. Through the simulation, we determined that check pattern can be most used in practical applications. Even though we used the smallest standard image format, we were able to show the feasibility of our method.

In the near future, we will extend this method to be implemented in architect hardware and elaborate a scalability of permission range for large size of image format.

Acknowledgments

This work was partially supported by "System IC 2010" project of Korea Ministry of Knowledge Economy, KOSEF, and IDEC.

References

- [1] Joint Video Team of ITU-T and ISO/IEC Recommendation H.264, "Advanced Video Coding for Generic Audiovisual Services," ITU-T, March. 2005.
- [2] Thomas, W. et al, "Overview of the H.264/AVC Video Coding Standard," IEEE Trans. on Circuit and Systems for Video Technology, Vol. 13, No. 7, July. 2003, pp. 560-576.
- [3] Thomas, W. et al, "Rate-constrained coder control and comparison of video coding standards," IEEE Trans. on Circuit and Systems for Video Technology, Vol. 13, No. 7, July. 2003, pp. 688-703.
- [4] Gwang-Hoon, P. et al, "Water Ring Scan Method for FGS Video Coding Schemes," IEICE Trans. on Communications, Vol. E88-B, No. 2, February. 2005, pp. 835-840.