Motion Compensated Frame Rate Up-Conversion Using Adaptive Extended Bilateral Motion Estimation

Daejun Park
Department of Electronics and Computer Engineering
Hanyang University
Seoul, South Korea
e-mail: daejoon12@naver.com

Jechang Jeong
Department of Electronics and Computer Engineering
Hanyang University
Seoul, South Korea
e-mail: jjeong@hanyang.ac.kr

***Abstract*—In this paper, a novel frame rate up conversion (FRUC) algorithm using adaptive extended bilateral motion estimation (AEBME) is proposed. Conventionally, extended bilateral motion estimation (EBME) conducts dual motion estimation (ME) processes on the same region, therefore involves high complexity. However, in this proposed scheme, a novel block type matching procedure is suggested to accelerate the ME procedure. We calculate the edge information using sobel mask, and the calculated the edge information is used in block type matching procedure. Based on the block type matching, decision will be made whether to use EBME. Motion vector smoothing (MVS) is adopted to detect outliers and correct outliers in the motion vector field (MVF). Finally, overlapped block motion compensation (OBMC) and motion compensated frame interpolation (MCFI) are adopted to interpolate the intermediate frame in which OBMC is employed adaptively based on the frame motion activity. Experimental results show that the proposed algorithm has outstanding performance and fast computation comparing with EBME.**

Keywords- Frame rate up conversion; extended bilateral motion estimation; overlapped block motion compensation; block type matching; frame motion activity.

I. INTRODUCTION

Frame rate up conversion (FRUC) is used in various display devices with the purpose of increasing frame rates. Liquid crystal display (LCD) has annoying motion blur effect especially in sequences with dynamic motion [1]. This is due to its hold-type display characteristics which tend to sustain the light intensity for a longer moment than cathode ray tube (CRT). Viewers have difficulty in tracking a fast moving object in LCD because image from previous frame may still remain on the display. This results in annoying effect, which is called ghost effect. FRUC is the ideal technique used to counter this problem. This noticeable motion blur is resolved by doubling the frame rates. FRUC algorithm is also useful in limited bandwidth situation. In narrow bandwidth channel, encoder has to decrease transmission data rating. So encoder transfers either odd or even frames of sequence. At the decoder side, the removed frames are to be restored using FRUC technique.

Various FRUC algorithms have been developed [2]. Approaches that do not consider the motion of objects are the simplest in FRUC algorithms, e.g., frame repetition, frame averaging. These algorithms are easy to be implemented in software and hardware. However, they have problems such as motion jerkiness. To reduce these artifacts, motion compensation techniques can be applied. Such methods are called motion compensation interpolation (MCI). Motion-compensated frame rate up conversion (MC-FRUC) is a popular method in FRUC [3]. It consists of motion estimation (ME) and MCI. ME produces motion vectors (MVs) by using the block matching algorithm (BMA) for its low complexity and ease of implementation. However, BMA suffers from various artifacts, e.g., blocking artifact, halo effect.

MC-FRUC can be classified further into two approaches, true motion based type and non-true motion based type. Extended bilateral motion estimation (EBME) carries out full search on both original and intermediate grid and therefore it is classified as a non-true motion based approach [4]. EBME scheme is a slow algorithm and regarded as unsuitable for real-time applications. Nonetheless, it has the advantage of executing BME precisely twice, on two different overlapping grids. Moreover, its outlier detection and correction function is effective in smoothing out false motion vector.

The proposed adaptive extended bilateral motion estimation (AEBME) is an modified version of conventional EBME algorithm. The novel block type matching procedure is proposed to accelerate the ME procedure. We calculate the edge information using sobel mask, and the calculated edge information is used in block type matching procedure. Based on the block type matching, decoder will decide whether to use EBME. Motion vector smoothing (MVS) is adopted to detect outliers and correct outliers in the motion vector field (MVF). Overlapped block motion compensation (OBMC) technique is adopted during interpolation process to reduce the blocking artifacts that may occur due to irregularity of motion vector [5]. OBMC is employed adaptively based on frame motion activity. Finally, motion compensated frame interpolation (MCFI) is adopted to restore the missing frames.

The rest of the paper is organized as follows: Section II presents an overview of EBME algorithm. Section III describes our proposed algorithm in details. The experimental results and test conditions are provided in Section IV. Finally, we conclude the paper in Section V.

 ISBN: 978-1-61208-444-2

II. Extended Bilateral Motion Estimation

A. Bilateral Motion Estimation

The bilateral motion estimation (BME), as illustrated in Figure 1, is executed under the assumption that object motion is temporally symmetric from the intermediate frame's point-of-view.

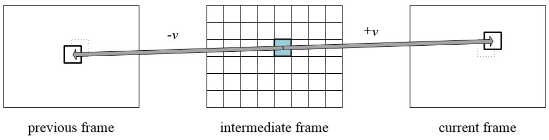

Figure 1. Illustration of the bilateral motion estimation.

The traditional BMA suffers from holes and occluded regions during compensation. However, bilateral motion estimation and compensation has no holes and occluded regions after reconstruction. In BME, the block in interpolated frame is regarded as the search center. The search is performed by comparing a block at a shifted position in the previous frame and another block at the opposite position in the current frame. We can compute the sum of bilateral absolute differences (SBAD) by (1) and find the MV which minimizes SBAD by (2).

$$SBAD(v_x, v_y) = \sum_{v_x \in SR} \sum_{v_y \in SR} \left| f_{n-1}(x - v_x, y - v_y) - f_n(x + v_x, y + v_y) \right| \quad (1)$$

$$v = \arg \min_{(v_x, v_y) \in SR} \{SBAD(v_x, v_y)\} \quad (2)$$

where *(v_x, v_y)* is the MV candidate, f_{n-1} and f_n are the previous and current frames, repectively. v is the selected MV, *BLK* is the block size, and *SR* is the search range.

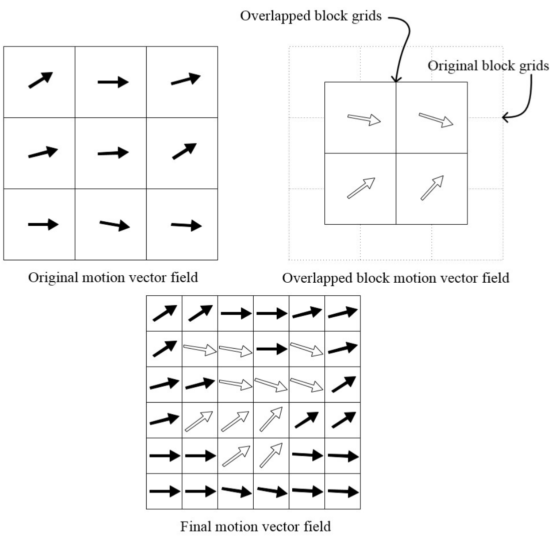

Figure 2. Illustration of the extended bilateral motion estimation.

B. Extended Bilateral Motion Estimation

The computational complexity of the BME is much lower than the quarter of the computational complexity of the BMA, because the search range of the BME is one quarter of that of BMA. However, when the motion trajectory of an object is not symmetrical from the intermediate frame's viewpoint, the true MV cannot be estimated. The EBME performs the BME for the overlapped blocks to search for a more accurate MV. Figure 2 illustrates how the EBME modifies the motion vector field, which is more precise than the original MVF. By comparing SBAD of the original block grids and the overlapped block grids, the final MVF will be decided.

C. Recursive Motion Vector Smoothing

An outlier causes the block artifact and degrades the image quality. MVS is adopted to detect outliers and eliminate outliers from the MVF.

$$v_m = \frac{1}{9} \sum_{i=1}^{9} v_i \quad (3)$$

$$D_i = abs(v_m - v_i) \quad (4)$$

$$D_n = \frac{1}{8} \sum_{i=2}^{9} D_i \quad (5)$$

In (3), v_m is an average MV of v_1 and all neighboring block's MVs surrounding v_1. In (4), D_i is the absolute difference between v_m and v_i, and D_n in (5) is the mean of the absolute difference between v_m and each of eight neighboring MVs. If $D_1 > D_n$, v_1 is an outlier.

After detecting all outliers, the smoothing process will correct them. The smoothing process is shown in Figure 3. First, it selects v_1 whose eight surrounding MVs are considered reliable. Median filtering will be employed to correct v_1. Next, it selects v_1, whose neighboring MVs contain one outlier. Median filtering will be conducted for reliable neighboring MVs. The smoothing process will be progressed by increasing the number of neighboring outliers one by one. If it cannot increase the number of neighboring outliers, the smoothing process will start over again.

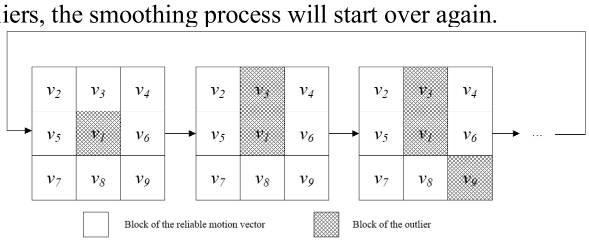

Figure 3. Recursive motion vector smoothing.

D. Overlapped Block Motion Compensation

OBMC process can drastically reduce blocking artifacts and provide a good visual quality in almost all sequences under an assumption that we have the accurate MVF. To enhance a visual quality, we employ bilinear window, illustrated in Figure 4, which is formulated as a linear estimator of pixel intensities given the limited block motion information. The coefficients of the filter are determined by (6).

$$w(u,v) = w_u \cdot w_v,\ w_u = \begin{cases} \frac{1}{N}(u+\frac{1}{2}) & for\ \ u = 0,...,N-1 \\ w_{(2N-1)-u} & for\ \ u = N,...,2N-1 \end{cases} \quad (6)$$

where N is the block size.

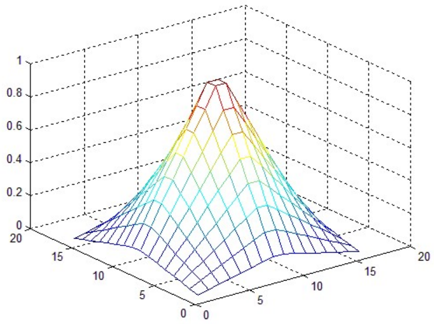

Figure 4. Bilinear window.

E. Motion Compensated Frame Interpolation

In order to construct the intermediate frame, MCFI is employed by using the final MVs. The intermediate frame is interpolated by (7). We select a block to which we want to apply MCFI, and enlarge block's size to the window size for OBMC process. Then, OBMC and MCFI are conducted.

$$f_{n-1/2} = \frac{1}{2}\{f_{n-1}(x - v_{x,final}, y - v_{y,final}) + f_n(x + v_{x,final}, y + v_{y,final})\} \quad (7)$$

where $(v_{x,final}, v_{y,final})$ is the final MV, and $f_{n-1/2}$ is the intermediate frame.

III. THE PROPOSED ALGORITHM

The flow chart of the proposed AEBME is shown in Figure 5. The proposed verification process is comprised of two components: block type matching and frame motion activity check.

In the previous work, we applied scene change detection algorithm and omitted motion vector smoothing algorithm [6]. But scene change detection algorithm shows better results only in specific case. So we omitted scene change detection algorithm in the proposed algorithm to make better AEBME algorithm.

In the previous work, we used CIF test sequences on experiment. But CIF size is not suited for the current imaging technology. So in the proposed algorithm, we used various test sequences used in HEVC test model.

A. Edge Detection

Sobel mask is used to calculate edge information [7]. The operator uses two 3x3 kernels to calculate approximations of the derivatives. The mask is slid over an area of the image. The edge magnitude *M(x,y)* is calculated by (8).

$$M(x,y) = |g_x| + |g_y| \quad (8)$$

B. Block Type Matching

Non-true MV indicates different objects in the previous and the current frame. We need to check if the objects from MV are same. The edge information, which was calculated using sobel mask, is used to check block type of each object. First, we calculate the mean of edge values of each block by (9). If the edge value of pixel in each block is larger than the mean, the pixel is classified into the edge pixel. Otherwise, the pixel is classified into the flat pixel. If the percentage of edge pixels in the block is larger than T_1, the block is classified into the edge block. Otherwise, the block is classified into the flat block.

$$edge_{mean} = \frac{1}{W \times H} \sum_{x \in W} \sum_{y \in H} M(x,y) \quad (9)$$

After BME, if the block types of the reference blocks of the intermediate block are different, EBME is performed. This block type matching process can improve an accuracy of indicating same objects in the previous and current frame.

C. Frame Motion Activity

Static images have the zero MVs for most blocks. On the other hand, dynamic images have large MVs except background. If OBMC is applied in a static region, visual quality degradation is inevitable. So OBMC should be applied depending on the characteristic of frames after checking frame motion activity. First, we calculate the average MV of all MVs in a frame using (10) and (11). If the average of MVs is larger than T_2, the frame is classified into a dynamic frame. In the opposite case, the frame is classified into a static frame. And then, OBMC is applied only in dynamic frames.

$$v_i = |v_{x,i}| + |v_{y,i}| \quad (10)$$

$$v_{avg} = \frac{1}{(W/BLK) \times (H/BLK)} \sum_{i \in frame} v_i \quad (11)$$

where v_i is the magnitude of *i-th* block's MV, *(W/BLK)x(H/BLK)* is the number of blocks in a frame, v_{avg} is an average MV of whole MVs in a frame.

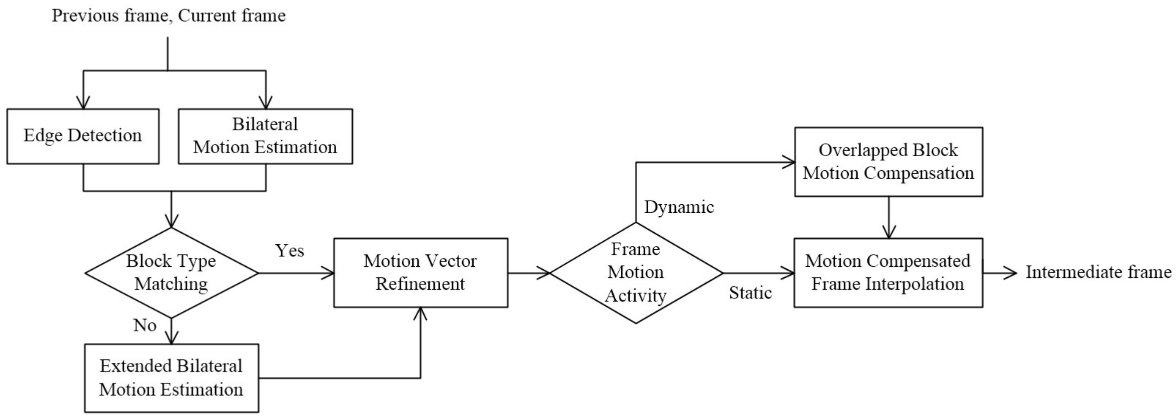

Figure 5. Flowchart of the proposed AEBME algorithm.

IV. EXPERIMENTAL RESULTS

The experiments are conducted using 25 odd frames of test sequences as input and 24 even frames are interpolated as a result. And original even frames are used as reference to calculate the peak signal to noise ratio (PSNR). The performance of the proposed AEBME algorithm has been evaluated through the objective evaluations. PSNR values of the intermediate frames are compared with EBME algorithm. In addition we have calculated the average number of the conducted EBME to show the result of computational complexity reduction.

For experiments, we set the original block size to 32x32 pixels and the search range to -16 +16. After BME, EBME and MVS, we set the block size to 16x16, OBMC filter size N to 32. The threshold T_1 in block type matching process is set to 0.6 and T_2 in checking frame motion activity process is set to 0.5. We used 11 test sequences which are the test sequences for HEVC.

PSNR is used as the metric for objective performance evaluation. The average PSNR values and computation times for the results are presented in Table I. The PSNR difference and computation time gain are also presented. The proposed AEBME algorithm has higher PSNR and comsumes less time than the anchor algorithm. These results are caused by skipping EBME process and OBMC process in static sequences.

TABLE I. PSNRS AND COMPUTATION TIMES OF TEST SEQUENCES.

Class	Sequence	EBME		EBME+MVS+OBMC		AEBME	
		PSNR(dB)	Time(s)	PSNR(dB)	Time(s)	PSNR(dB)	Time(s)
A	Traffic	44.03	120.11	44.13	122.43	45.21	67.14
	PeopleOnStreet	39.58	120	39.59	122.54	40.71	66.59
	Average	**41.81**	**120.06**	**41.86**	**122.49**	**42.96**	**66.87**
	Gain	**0**	**1**	**+0.05**	**0.98**	**+1.15**	**1.8**
B	Kimono1	41.25	55.57	41.32	57.08	41.69	32.52
	ParkScene	41.77	68.23	41.86	69.41	42.29	38.88
	Cactus	38.12	70.15	38.15	71.02	38.46	40.28
	BQTerrace	37.05	72.63	37.08	73.83	37.03	38.43
	BasketballDrive	36.8	59.47	36.87	61.98	37.25	33.49
	Average	**39**	**65.21**	**39.06**	**66.66**	**39.34**	**36.72**
	Gain	**0**	**1**	**+0.06**	**0.98**	**+0.34**	**1.78**
C	RaceHorses	34.12	11.25	34.31	11.65	34.56	6.52
	BQMall	41.06	12.86	41.18	13.05	41.62	7.47
	PartyScene	42.82	14.14	42.79	14.33	43.04	8.02
	BasketballDrill	40.2	13.2	40.17	13.48	40.9	7.52
	Average	**39.55**	**12.86**	**39.13**	**13.13**	**40.03**	**7.38**
	Gain	**0**	**1**	**-0.42**	**0.98**	**+0.48**	**1.74**

Table II shows the number of block type mismatch blocks where EBME algorithm is conducted.

TABLE II. THE AVERAGE NUMBER OF EBME PROCESS PERFORMED.

Class		EBME	EBME+MVS+OBMC	AEBME
A	Average	3871	3871	68.48
	Gain	1	1	56.53
B	Average	1888	1888	74.58
	Gain	1	1	25.32
C	Average	350	350	10.27
	Gain	1	1	34.08

V. CONCLUSIONS

This paper proposed AEBME algorithm, FRUC scheme that considers block type and frame motion activity. The novel block type matching algorithm is proposed to reduce additional BME process. We calculate the edge information using sobel mask, and the calculated edge information is used to decide whether to use EBME. MVS is adopted to detect and eliminate outliers in a MVF. OBMC is selectively applied by utilizing frame motion activity check. Finally, the missing frame are restored by adopting MCFI.

Experimental results show that the proposed algorithm has outstanding performance and fast computation comparing with the anchor algorithm.

ACKNOWLEDGMENT

This research was supported by the MSIP(Ministry of Science, ICT&Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2015-H8501-15-1005) supervised by the IITP(Institute for Information&communications Technology Promotion).

REFERENCES

[1] S. H. Chan, T. X. Wu, and T. Q. Nguyen, "Comparison of wo frame conversion schemes for reducing LCD motion blurs," IEEE Signal Processing Letters, vol. 17, pp. 782-786, 2010.

[2] D. Wang, A. Vincent, P. Blanchfield, and R. Klepko, "Motion-compensated frame rate up-conversion Part II: new algorithms for frame interpolation," IEEE Transactions on Broadcasting, vol. 56, pp. 142-149, 2007.

[3] B. D. Choi, J. W. Han, C. S. Kim, and S. J. Ko, “Motion compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, pp. 407-416, 2007.

[4] S. J. Kang, K. R. Cho, and Y. H. Kim, “Motion compensated frame rate up-conversion using extended bilateral motion estimation,” IEEE Transactions on Consumer Electronics, vol. 53, pp. 1759-1767, 2007.

[5] M. T. Orchard and C. J. Sullivan, “Overlapped block motion compensation: an estimation-theoretic approach,” IEEE Transactions on Image Processing, vol. 9, pp. 1509-1521, 2007.

[6] D. Park and J. Jeong, “Motion Compensated Frame Rate Up-Conversion Using Modified Adaptive Extended Bilateral Motion Estimation,” Journal of Automation and Control Engineering, vol. 2, no. 4, pp. 371-375, Dec. 2014.

[7] R. C. Gonzalez and R. E. Woods, “Digital image processing, 3rd edition,” Prentice Hall, New Jersey, 2010.