Multidirectional motion estimation algorithm for frame rate up-conversion

Songhyun Yu[1], *Jechang* Jeong[1]

[1] Department of Electroincs and Computer Engineering, Hanyang University, Seoul, Korea

Abstract. In this paper, we propose a new motion estimation algorithm for motion-compensated frame rate up-conversion. To reduce computational complexity, it constructs a hierarchical structure using Gaussian image pyramid and unidirectionally estimates motion vectors (MVs) between two original frames at the top level. Since the accuracy of the motion vector may deteriorate at low resolution, each block has five candidates for motion vector so that disadvantages of hierarchical motion estimation are reduced. After motion estimation (ME), MVs of bottom level are corrected based on the sum of absolute difference (SAD) values of the blocks. All unidirectional MVs are assigned to bidirectional MVs, and new frames are interpolated using these MVs. As a result, proposed algorithm shows up to 1.3 dB increase in PSNR compared with conventional algorithm with low computational complexity, and it shows clearer images than conventional algorithms.

1 Introduction

Frame rate up-conversion (FRUC) is a technique which interpolates frame by inserting new frames between originally existed frames. In the early days of technology, frame interpolation was performed using frame repetition or simple averaging of two frames. However, this causes image quality deterioration. In recent years, motion-compensated FRUC (MC-FRUC) algorithm has been developed [1-3]. This algorithm generally has three steps: motion estimation (ME), motion vector smoothing (MVS), and motion compensated interpolation (MCI).

ME is a process of finding motion of each object using original frames, and motion vectors (MVs) are obtained in this step. Depending on the direction in which the motion is predicted, ME is divided into two major categories: unidirectional motion estimation (UME), bidirectional motion estimation (BME). Motions are predicted unidirectionally in the viewpoint of the original frame, and overlapped areas and hols occur in the interpolated frame. BME bidirectionally predicts motions in the viewpoint of the interpolated frame, so there are not holes and overlapped areas. However, it can find false MVs since there in no information about the current reference block.

Basically, ME has a high computational complextiy, so for real-time processing of high-resolution video, it is essential to reduce the complexity. Several studies proposed low-complexity FRUC algorithms; [13] used Gaussian pyramid structure and Laplacian pyramid structure in ME step, and [15] used subsampling of the original image for ME. These algorithms, however, have reduced the accuracy of MVs due to the resolution changes of the original images. In this paper, we propose a multidirectional motion estimation (MME) algorithm to reduce complexity and maintain the accuracy of the motion vector.

For the MVS process, [4] used mean and variance of neighboring MVs and weighted sum of these motion vectors replace the current motion vector, and in [6], current motion information is used to correct false motion vectors. However, when the current block is at the boundary of an object or when the current motion vector is not correct, these algorithms can cause an error. In this paper, an adaptive global MVS algorithm is proposed to overcome these limitations.

The remainder of this paper is organized in the following order; secsion 2 reveal the proposed algorithm in detail, section 3 shows experimental results compared with other algorithms, and section 4 concludes this paper.

2 Proposed algorithm

Fig. 1 shows overall flowchar of the proposed algorithm. Two original frames form a pyramid structure using [7]. The unidirectional motion estimation is performed at the top level, in which five candidate positions are designated. At the lower levels, the candidates are modified by small search range motion estimation around each position. Once the MVs of the all blocks are determined, these MVs are modified according to the sum of absolute difference (SAD) value. Final MVs are converted to the bidirectional MVs using [5]. As a result, a new frame is interpolated in the MCI step. The details of the proposed algorithm are described in the following subsections.

2.1. Motion estimation

Prior to the motion estimation process, each frame forms an image pyramid as shown in Fig. 2. As shown in Fig. 3,

[*] Corresponding author: jjeong@hanyang.ac.kr

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

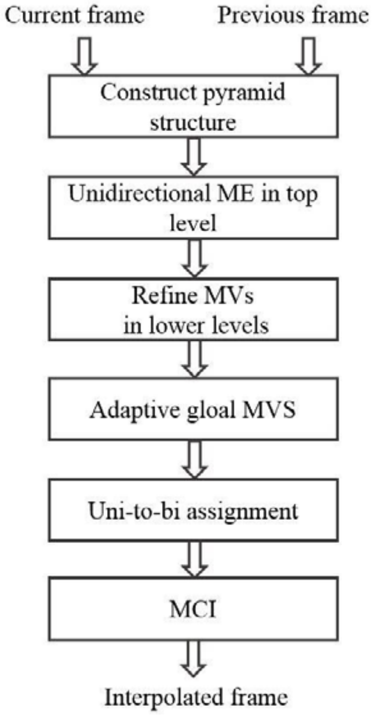


Fig. 1. Flowchart of the proposed algorithm.

the proposed MME algorithm divides the search range into four regions according to directions and finds the candidate positions with the smallest SAD for each region. SAD is given by:

$$SAD_u(B, v_m) = \sum_{p\in B} \left| f_p(p) - f_c(p + v_m) \right| \qquad (1)$$
$$v_o = \arg \min_{v_m \in SR_o} \{SAD(B, v_m)\}, \qquad o = 1,2,3,4$$

where SAD_u denotes the SAD value in unidirection, B is the current block, p is a pixel position in the block B, and v_o means candidate MVs of the current block in each search range "SR_o". Four resions of the search range are represented as n.

After top level processing , each candidates points are updated using ME with small search range. The size of this small search range is determined based on the SAD value of each candidates, and it is given by :

$$SR_{small}(S) = \alpha(S - 255)^2 + \beta \qquad (2)$$

where S is a SAD value, and SR_{small} is the size of the search range. β means maximum size of the search range, and α is weighting factor. We set $\beta = 50$, $\alpha = 0.0007689$.

At the bottom level, current block decides final motion vector which has the smallest SAD among the candidates

2.2. Adaptive global motion vector smoothing

When the ME process of one frame is ompleted, the MVS is performed to reduce an erorr generated in the ME.

Fig. 2. Pyramid structure.

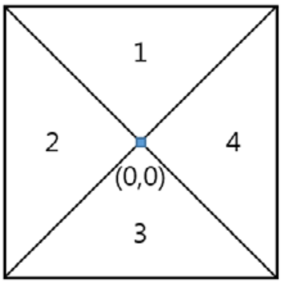


Fig. 3. Four regions of the search area.

Proposed algorithm utilizes SAD values of final MVs to determine which MV is incorrect one. This process is shown as follows:

$$\begin{aligned} R(B) &= 1 \qquad & if\ SAD(B) > Th \\ R(B) &= 0 \qquad & otherwise \end{aligned} \qquad (3)$$

where $R(B)$ indicates reliability of the current block B, and Th means threshold value.

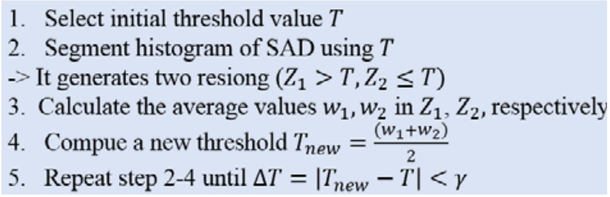
To determine threshold value, we used recursive algorithm in [8], and it is shown in Fig. 4.

3 Experimental results

Objective and subjective evaluations are performed in this section. For the objective ecaluation, peark signal to noise ratio (PSNR) is used, and complexity of each algorithms are considered. High efficiency video coding (HEVC) standard test sequences were chosen for the test so that high resolution videos are includes in the experiment. Block size is set to 16×16, and search range of the top level is set to $\pm$12 pixels.

Table 1 shows PSNR results compared with conventional algorithms [15, 4, 10]. In all sequences, the proposed algorithm shows the highest PSNR value; up to 1.17 dB compared to [15], up to 1.34 dB compared to [4], and up to 1.05 dB compared to [10]. Average values also shows that proposed algorithm has highest PSNR value.

DOI: 10.1051/matecconf/201712505002



```
1. Select initial threshold value $T$
2. Segment histogram of SAD using $T$
-> It generates two resiong $(Z_1 > T, Z_2 \leq T)$
3. Calculate the average values $w_1, w_2$ in $Z_1$, $Z_2$, respectively
4. Compue a new threshold $T_{new} = \frac{(w_1+w_2)}{2}$
5. Repeat step 2-4 until $\Delta T = |T_{new} - T| < \gamma$
```

Fig. 4. Algorithm to determine threshold.

Table 2 shows computational complexity of the algorithms. N means block size (16), R means search range (25), and R_s denotes small search range (2). The small search range of the proposed algorithm is adaptively determined, but approximated to an average value of 2. Ratio represents the relative amount of computation when the computation complexity of the proposed algorithm is 1. It can be seen that the proposed algorithm has less computational complexity than the other algorithms. [15] does not make much difference; however, when MV is modified at lower levels, there are many overlapping parts, so the actual computational complexity of the proposed algorithm is smaller than 1.

Fig. 5 shows results images in *BasketballDrive* sequence. In the square part, the result image of the proposed algorithm is clearer than the other algorithms.

Table 1. PSNR comparison

	[15]	[4]	[10]	Proposed
PeopleOnStreet (2460×1600)	26.86	26.69	27.91	28.03
BasketballDrive (1920×1080)	30.30	30.39	30.46	30.84
ParkScene (1920×1080)	35.10	34.95	34.30	35.35
Johnny (1280×720)	41.39	41.40	41.52	41.62
BQMall (832× 480)	33.01	32.53	32.94	33.01
Average	33.33	33.19	33.43	33.77

Table 2. Computational complexity

Algorithm	Complexity	Ratio
[15]	$(\frac{N}{2})^2(R+1)^2$	1.01
[4]	$2(1.5N)^2(2R+1)^2$ $-((1.5N)^2-1.5N)^2$	63.03
[10]	$2N^2(2R+1)^2$ $+2N^2(2R_s+1)^2$	31.48
Proposed	$\left(\frac{N}{4}\right)^2\left(\frac{R}{2}+1\right)^2$ $+5(\frac{N}{2})^2(2R_s+1)^2$ $+5(N)^2(2R_s+1)^2$	1

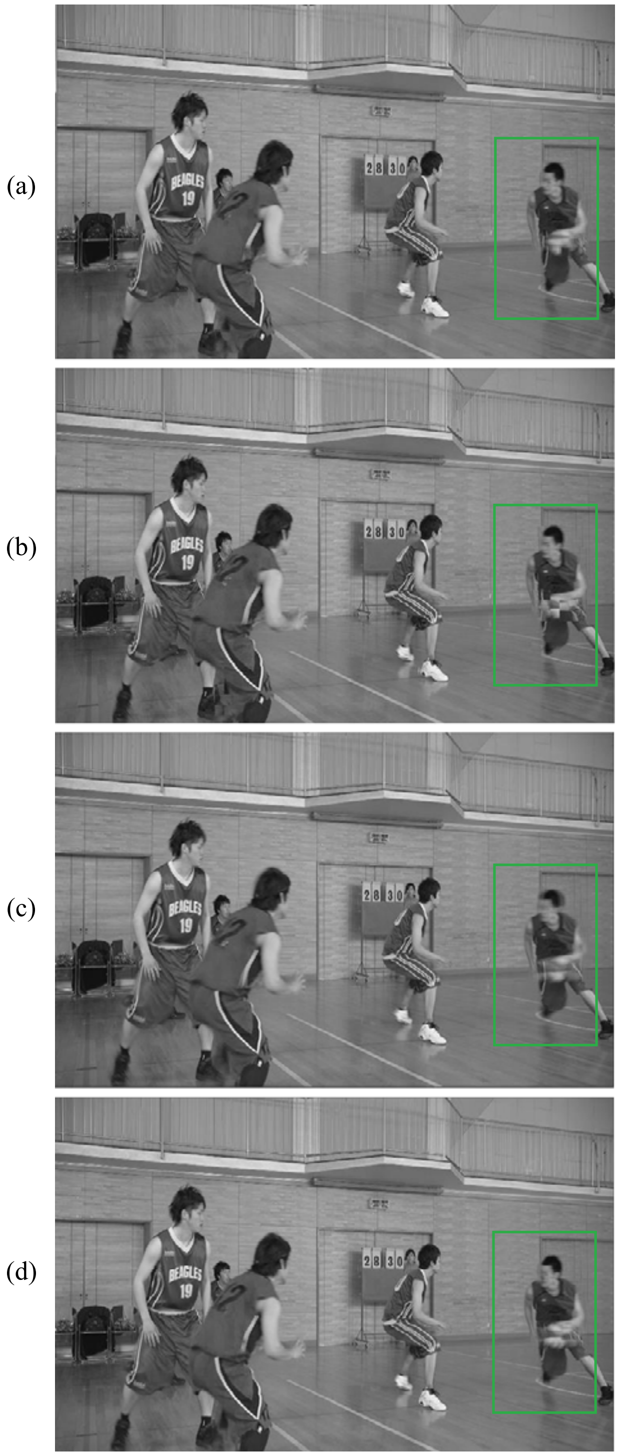


Fig. 5. Result images (a) [15], (b) [4], (c) [10], (d) proposed.

4 Conclusion

In this paper, ME algorithm and MVS algorithm for the MC-FRUC were proposed. In the ME process, multidirectional motion estimation was proposed to find accurate MVs with low computational complexity, and adaptive global motion vector smoothing algorithm was proposed in the MVS step, so the error of the MVs is corrected by adaptively determining the threshold value in frame unit. As a result, the proposed algorithm shows high PSNR in objective evaluation and clear image in subjective evaluation while keeping the computational complexity low.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT and future Planning(NRF-2015R1A2A2A01006004)

References

1. C. Cafforio, F. Rocca, and S. Tubaro, "Motion compensated image interpolation," *IEEE Trans. Communication*, vol.**38**, no.2, pp.215 -222, Feb.(1990).
2. S. H. Lee, Y. C. Shin, S. Yang, H. H. Moon, and R.H. Park,"Adaptive motion-compensated interpolation rate-up-conversion*," IEEE Trans. Consumer Electronics*, vol.**48**,no.3, pp.444-450, Aug. (2002).
3. S. H. Lee, O. Kwon, and R. H. Park, "Weighted-adaptive motion compensated frame rate up-conversion," *IEEE Trans. Consumer Electronics*, vol.**49**, no.3, pp.485-492, Aug. (2003).
4. D. G. Yoo, S. J. Kang, and Y. H. Kim, “Direction-select motion estimation for motion-compensated frame rate up-conversion,” *J. Display Technol.*, vol. **9**, no. 10, pp. 840–850, Oct. (2013).
5. Y. Guo, L. Chen, Z. Gao, and X. Zhang, “Frame rate up-conversion using linear quadratic motion estimation and trilateral filtering motion smoothing,” J. Display Technol., vol. **12**, no. 1, pp. 89-98, Jan. (2016).
6. D. Wang, L. Zhang, and A. Vincent, “Motion-compensated frame rate up-conversion—Part I: Fast multi frame motion estimation,” *IEEE Trans. Broadcast*, vol. **56**, no. 2, pp. 133–141, Jun. (2010).
7. P. J. Burt, and E. H. Adelson, “The laplacian pyramid as a compact image code,” *IEEE Trans. Commun.*, vol. **31**, no. 4, pp. 532-540, Apr. (1983).
8. R. C. Gonzalez, and R. E. Woods, *Digital Image Processing*, Pearson Prentice Hall. (2010).
9. C. Wang, L. Zhang, Y. He, and Y. P. Tan, “Frame rate up-conversion using trilateral filtering,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. **20**, no. 6, pp. 886-893, Jun. (2010).
10. S. J. Kang, S. Yoo, and Y. H. Kim, “Dual motion estimation for frame rate up-conversion,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. **20**, no. 12, pp. 1909–1914, Dec. (2010).
11. D. Wang, A. Vincent, P. Blanchfield, and R. Klepko, “Motion-compensated frame rate up-conversion—Part II: New algorithms for frame interpolation,” *IEEE Trans. Broadcast.*, vol. **56**, no. 2, pp. 142–149, Jun. (2010).
12. S. J. Kang, K. R. Cho, and Y. H. Kim, “Motion compensated frame rate up-conversion using extended bilateral motion estimation,” *IEEE Trans. Consumer Electron.*, vol. **53**, no. 4, pp. 1759–1767, Nov. (2007).
13. B. W. Jeon, G. I. Lee, S. H. Lee, and R. H. Park, "Coarse-to-fine frame interpolation for frame rate up-conversion using pyramid structure," *IEEE Trans. Consum. Electron.*, vol. **49**, no. 3, pp. 499-508, (2003).
14. M. T. Orchard and G. J. Sullivan, "Overlapped block motion compensation: An estimation-theoretic approach,” *IEEE Trans. Image Process*., vol. **3**, no. 5, pp. 693-699, (1994).
15. B. Choi, S. Lee, and S. Ko, “New frame rate up-conversion using bi-directional motion estimation,” *IEEE Trans. Consumer Electronics*, vol. **46**, no. 3, pp. 603-609, (2000).