

Hierarchical Motion-Compensated Frame Rate Up-Conversion Using Gaussian Pyramid and Guidance Motion Vector

Bumjun Park¹, and Jechang Jeong^{2,*}

¹Department of Electronics and Computer Engineering, Hanyang University, Seoul, Korea

²Department of Electronic Engineering, Hanyang University, Seoul, Korea

Abstract. This paper proposes fast and accurate frame rate up-conversion algorithm for both software and hardware. The proposed algorithm first construct Gaussian pyramid structure using 3×3 Gaussian mask. After constructing Gaussian pyramid, forward and backward unilateral motion estimation is implemented at the top level of the pyramid. Then, motion vector field that is obtained at the top level is projected onto the middle level and the bottom level of the pyramid with unilateral motion vector refinement. Next, on the bottom level, we implement motion vector smoothing and proposed motion vector shifting with bilateral motion vector refinement. By shifting unilateral motion vector to bilateral motion vector, we could avoid generating holes and overlapped regions. Finally, we select final MV between forward and backward MVs according to their reliability. After all the processes, we interpolate new frames by using overlapped block motion compensation. Experimental results show that proposed algorithm is up to 80 times faster than conventional algorithm with 0.1dB peak signal-to-noise gain in average.

1 Introduction

Frame rate up-conversion (FRUC) is the technique that interpolates virtual frame which exists in the middle of two original frames' time interval. As a result of FRUC process, we can get high frame rate video with low frame rate video.

There have been proposed many FRUC algorithms. In the early stage, FRUC was implemented by copying original frames. However, simply copying frames couldn't make video that has enough quality to be used. As the research progresses, motion compensated frame rate up-conversion (MC-FRUC) was proposed. MC-FRUC is composed of two parts, motion estimation (ME) and motion compensated interpolation (MCI) [1]. ME is the process that finds motion vector (MV). There are many ME algorithms but block matching algorithm (BMA) is mainly used because it is easy to implement both in software and hardware [2]. BMA finds the MV by finding the block that is most similar to the current block in the search range. Using BMA, MC-FRUC finds MV with two methods, unilateral ME and bilateral ME. Unilateral ME finds the MV from the view point of one original frame to other original frame. On the other hand, bilateral ME finds the MV from the view point of interpolated frame to two original frames. In general, unilateral ME finds more accurate MV than bilateral ME because bilateral ME can find incorrect MV when there exist periodical or uniform background. However, unilateral ME generates holes and overlapped regions in the interpolated frame which doesn't appears to bilateral

ME. Therefore, unilateral ME need additional process to handle holes and overlapped regions. The MCI is the process that interpolates new frames by using MV information. When interpolating frames, blocking artifact occurs because of BMA. To solve this problem, overlapped block motion compensation (OBMC) was proposed [3]. OBMC reduces blocking artifact by using bilinear window.

Up to this point, many MC-FRUC algorithms were proposed [4] - [6]. Choi et al. proposed bi-directional MC-FRUC (BI_FRUC) using subsampled image which is a simple kind of pyramid structure [7]. However, their subsampling method was too simple to find correct MV. More than that, they used only forward unilateral MV as an initial MV. As a result, their algorithm estimated less accurate MV. Furthermore, when shifting unilateral MV to bilateral MV, they only used MV that is located in the same position of the current block in the reference frame. Consequently, miss match between unilateral MV and bilateral MV caused loss of quality despite of their refinement process. To overcome their weakness, Yoo et al. proposed direction-select ME (DSME) which use both forward and backward unilateral MV as an initial MV to estimate accurate MV [8]. However, they also used simple subsampling method for reducing computational complexity which is not proper to find accurate MV. In addition, they couldn't propose the solution to minimize error that occurs when shifting unilateral MV to bilateral MV.

In this paper, we propose hierarchical MC-FRUC using Gaussian pyramid and guidance motion vector to

* Corresponding author: jjeong@hanyang.ac.kr

solve existing defects [9]. To reduce computational complexity, conventional algorithms used simple subsampling which causes loss of accuracy of ME. To solve this problem, we adopted 3-level Gaussian pyramid to make subsampled image that contains a local average of lower level image at every pixel. In addition, we consider nine MV candidate and propose guidance MV to get rid of miss match that occurs when shifting unilateral MV to bilateral MV.

This paper is organized as follows. Section 2 describes details about proposed algorithm. Section 3 shows experimental results and compares proposed algorithm with conventional algorithms. Finally, section 4 concludes the paper.

2 Proposed algorithm

Fig. 1 shows overall flow chart of proposed algorithm. At first, proposed algorithm make 3-level Gaussian pyramid using 3×3 Gaussian mask. Fig. 2 shows the structure of 3-level Gaussian pyramid. Next, forward and backward unilateral ME is implemented at the top level. Then, MV field that is obtained at the top level is projected onto the middle level and the bottom level of the pyramid with unilateral MV refinement. Next, on the bottom level, we implement weighted MV smoothing and proposed MV shifting with bilateral MV refinement. MV shifting is the process that shifts unilateral MV to bilateral MV. Finally, in the MV selecting process, we select MV between forward and backward MVs according to their reliability. After all the process, we interpolate new frames with OBMC.

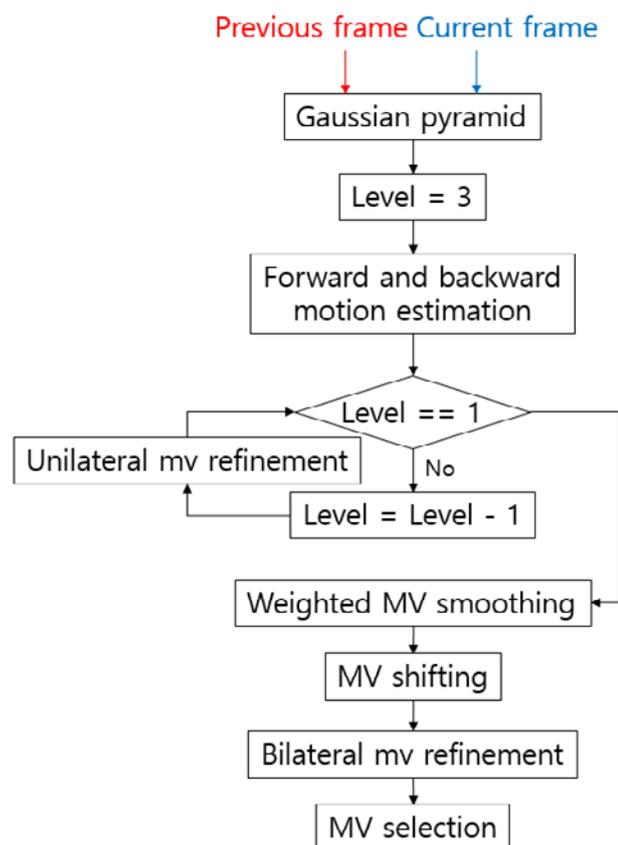


Fig. 1. Overall flow chart of proposed algorithm.

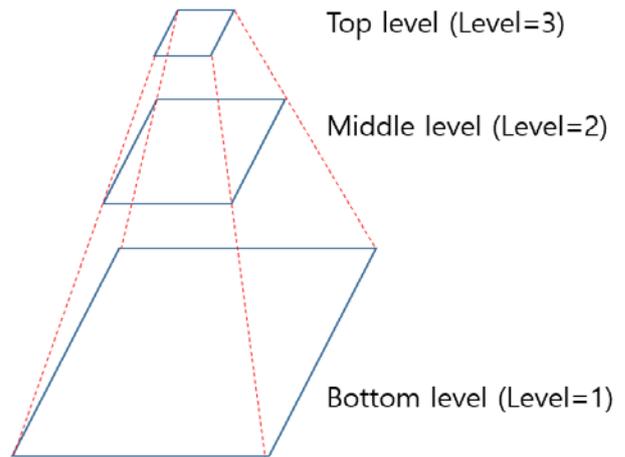


Fig. 2. Structure of 3-level Gaussian pyramid.

2.1. Gaussian pyramid

Many algorithms have been proposed to reduce computational complexity. Among them, pyramid structure is most popular because it is easy to implement both for software and hardware [9]. However, conventional algorithms used simple subsampling method which reduces accuracy of ME [7] – [8]. To overcome this weakness, our proposed algorithm uses 3-level Gaussian pyramid structure using 3×3 Gaussian mask. By using Gaussian mask, our subsampled images contain local average of lower level image which increases accuracy of ME. 3×3 Gaussian mask is defined as follows.

$$G = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix} \times \frac{1}{16} \quad (1)$$

where G denotes Gaussian mask.

2.2 Forward and backward motion estimation

After Gaussian pyramid structuring, forward and backward ME is implemented at the top level of the pyramid. Forward ME finds the MV from the view point of previous frame to current frame and backward ME finds MV from the view point of current frame to previous frame. When finding MV, we calculate sum of absolute difference (SAD). MV is achieved by finding position that has minimum SAD in the search range. SAD and MV for forward and backward directions are defined as follows.

$$SAD_f(dx, dy) = \sum_{x, y \in B} |f_{n-1}(x, y) - f_n(x - dx, y - dy)|$$

$$SAD_b(dx, dy) = \sum_{x, y \in B} |f_{n-1}(x - dx, y - dy) - f_n(x, y)| \quad (2)$$

$$v_f = \arg \min_{dx, dy \in R} \{SAD_f(dx, dy)\}$$

$$v_b = \arg \min_{dx, dy \in R} \{SAD_b(dx, dy)\}$$

where SAD_f and SAD_b denotes forward and backward SAD respectively. (dx, dy) denotes candidate MV position, B denotes current block, and f_{n-1} and f_n denote previous and current frame respectively. v_f and v_b denote forward and backward MV respectively and R denotes search range.

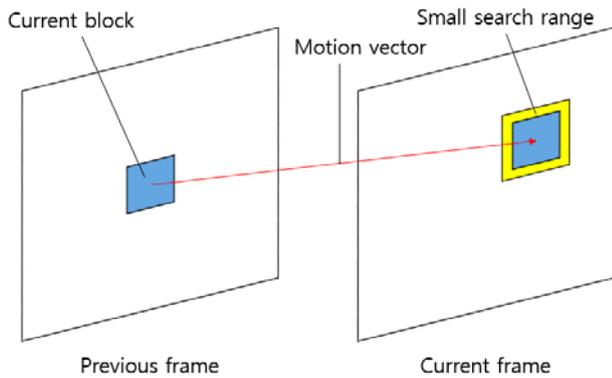


Fig. 3. Example of unilateral motion vector refinement.

2.3 Motion vector projection

After obtaining MV field at the top level, we project MV field onto the middle level and the bottom level of the pyramid. When projecting MV field, MVs should be doubled every time the level increases. However, simply doubled MV field does not fit at the next level. To solve this problem, unilateral MV refinement is implemented every time the level increases. Fig. 3 shows an example of unilateral MV refinement. Using (2), we refine MV by finding minimum SAD in the small search range.

2.4 Weighted motion vector smoothing

After projecting MV field onto the bottom level, we implement weighted MV smoothing. Weighted MV smoothing contains two parts, detecting outlier MV and MV smoothing. In the detecting outlier MV process, we detect outlier MV by calculating (3).

$$\begin{aligned} v_{avg} &= \frac{1}{9} \sum_{i=0}^8 v_i \\ D_i &= \text{abs}(v_{avg} - v_i) \\ D_n &= \frac{1}{8} \sum_{i=1}^8 D_i \end{aligned} \quad (3)$$

where v_{avg} denotes average of nine MVs, D_i denotes difference between v_{avg} and v_i , D_n denotes average MV difference of neighboring MVs. If D_0 is bigger than D_n , we set current block MV v_0 as outlier MV. Fig. 4 shows notation of motion vectors.

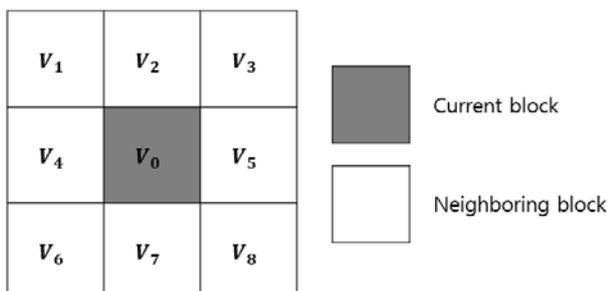


Fig. 4. Notation of motion vectors.

After detecting outlier MVs, we smoothen outlier MVs with weighted sum of neighboring inlier MVs. Weights of neighboring inlier MVs are assigned in inversely proportional to Euclidian distance. The weight window and smoothing method is defined as follows.

$$\begin{aligned} w_i &= \begin{cases} \frac{1}{\sqrt{2}}, & i = 1,3,6,8 \\ 1, & i = 2,4,5,7 \end{cases} \\ v_{0s} &= \frac{\sum_{i=\text{inliers}} v_i \times w_i}{\sum_{i=\text{inliers}} w_i} \end{aligned} \quad (4)$$

where w_i denotes weight window of neighboring MVs and v_{0s} denotes smoothen v_0 .

2.5 Motion vector shifting

In this step, we shift unilateral MV to bilateral MV. In the conventional algorithms [7] – [8], they considered only one MV that is located on the same position of current block in the reference frame. However, considering only one MV candidate caused MV shifting error despite of their additional process to refine it. In this paper, we consider additional eight MV candidates that is neighboring with current block. By considering additional MV candidates, we could use symmetrical properties to reduce miss match between unilateral MV and bilateral MV. Proposed MV shifting consists of two process, finding guidance MV and bilateral MV refinement. When finding guidance MV, first thing to do is comparing the area of nine MV candidates that overlaps with current block. Fig. 5 shows an example of finding guidance MV. In proposed MV shifting method, we choose most overlapped MV as a guidance MV. However, we can encounter three kinds of cases as we process proposed method. First, if we have no MV that overlaps with current block, we choose MV that is located on the same position of the current block as a guidance MV. Second, if we have only one MV that overlaps most, we choose that MV as a guidance MV. Finally, if we have more than two MV that overlaps most, we calculate sum of bilateral absolute difference (SBAD) of each MV. After calculating SBAD, we choose the MV that has minimum SBAD as a guidance MV. In this case, SBAD and guidance MV is defined as follows.

$$\begin{aligned} SBAD(v_{hx}, v_{hy}) &= \sum_{x,y \in B} |f_{n-1}(x - v_{hx}, y - v_{hy}) - f_n(x + v_{hx}, y + v_{hy})| \\ v_g &= \arg \min_{v_i \in M} \{SBAD(v_{hx}, v_{hy})\} \end{aligned} \quad (5)$$

Where v_{hx} and v_{hy} denote half of x directional coefficient and y directional coefficient of v_i . M denotes the set of MV that overlaps most.

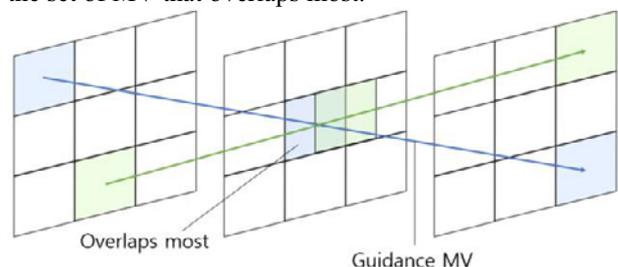


Fig. 5. Example of finding guidance motion vector.

In the proposed algorithm, we minimized motion vector shifting error by finding proper guidance MV. However, to reduce remaining errors, we implement bilateral MV refinement. This process is similar to unilateral MV refinement. The difference is that we use SBAD when implementing bilateral MV refinement instead of SAD. Using (5), we refine guidance MV by finding the position that has minimum SBAD in the small search range.

2.6 Motion vector selection

In the proposed algorithm, all processes are applied both to forward and backward MVs respectively. In this step, after applying all processes, we choose one MV that is more reliable. To select more reliable MV, we calculate SBAD of forward and backward MV and use it as a reliability. Reliability and final MV are defined as follows.

$$\begin{aligned} r_f &= SBAD(v_{fx}, v_{fy}) \\ r_b &= SBAD(v_{bx}, v_{by}) \\ v_{final} &= \min(r_f, r_b) \end{aligned} \quad (6)$$

where r_f and r_b denote reliability of forward and backward v_g . v_{fx} and v_{fy} denote x directional coefficient and y directional coefficient of forward v_g . v_{bx} and v_{by} denote x directional coefficient and y directional coefficient of backward v_g . v_{final} denotes final MV.

3 Experimental results

In this paper, we interpolate even frames by using two odd frames of original sequence. When interpolating frames, we interpolated frames by using OBMC [3]. For the test sequences, we used HEVC test sequences. When experimenting, we set block size as 8pixel and search range as 33pixel at the top level. Small search range was set to 5pixel. When evaluating quality of algorithm, we calculated peak signal to ratio (PSNR) with original even frames and interpolated even frames.

Table 1 and Table 2 shows PSNR and time results comparing with conventional algorithms [7], [8] and proposed algorithm. As we used 3-level Gaussian pyramid, proposed algorithm is about 80 times faster than [7] with 0.1dB PSNR gain in average. Also, proposed algorithm is about 2 times faster than [8] with 0.17dB PSNR gain in average.

Fig. 6 shows result image of conventional algorithms and proposed algorithm. In this image, DSME and BI_FRUC couldn't interpolate lines of playground clearly because they found incorrect MV. Furthermore, both DSME and proposed algorithm interpolated foot of player smoothly but BI_FRUC interpolated with incorrect MV. More than that, the method BI_FRUC used to interpolate frames couldn't reduce blocking artifact when the movement is fast. This kind of weakness also can be detected on the basketball. DSME and proposed interpolated moving basketball smoothly. However, basketball of BI_FRUC was distorted by blocking artefact. Overall, proposed algorithm shows better performance than conventional algorithms in both objective and subjective comparison.

Table 1. PSNR(dB) results

Class	Sequence	DSME	BI_FRUC	Proposed
Class A	<i>PeopleOnStreet_2560x1600</i>	27.91	26.86	27.61
	<i>Traffic_2560x1600</i>	37.20	37.72	37.28
Class B	<i>BasketballDrive_1920x1080</i>	30.64	30.30	31.02
	<i>BQTerrace_1920x1080</i>	32.68	31.77	32.5
	<i>Cactus_1920x1080</i>	31.8	32.01	31.93
	<i>Kimono1_1920x1080</i>	36.2	35.61	35.97
	<i>ParkScene_1920x1080</i>	34.59	35.10	35.22
Class C	<i>BasketballDrill_832x480</i>	29.67	30.80	30.88
	<i>BQMall_832x480</i>	32.93	33.01	32.93
	<i>PartyScene_832x480</i>	30.68	30.02	30.49
	<i>RaceHorses_832x480</i>	27.05	26.98	27.05
Class E	<i>FourPeople_1280x720</i>	40.49	40.58	40.27
	<i>Johnny_1280x720</i>	41.52	41.39	41.75
	<i>KristenAndSara_1280x720</i>	41.32	41.50	41.31
Average		33.90	33.83	34.01

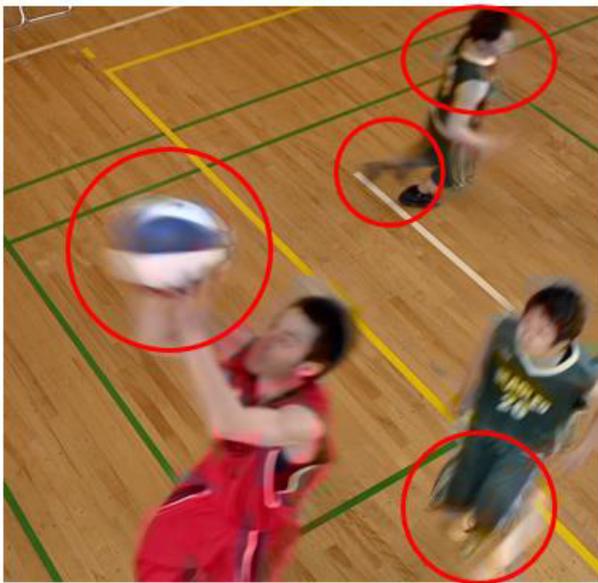
Table 2. Time(sec) results

Class	Sequence	DSME	BI_FRUC	Proposed
Class A	<i>PeopleOnStreet_2560x1600</i>	522	67	30
	<i>Traffic_2560x1600</i>	340	49	26
Class B	<i>BasketballDrive_1920x1080</i>	1322	149	98
	<i>BQTerrace_1920x1080</i>	10427	180	73
	<i>Cactus_1920x1080</i>	1029	137	57
	<i>Kimono1_1920x1080</i>	11040	72	36
	<i>ParkScene_1920x1080</i>	462	69	26
Class C	<i>BasketballDrill_832x480</i>	158	21	12
	<i>BQMall_832x480</i>	179	26	13
	<i>PartyScene_832x480</i>	160	25	10
	<i>RaceHorses_832x480</i>	138	17	10
Class E	<i>FourPeople_1280x720</i>	12842	48	24
	<i>Johnny_1280x720</i>	569	75	27
	<i>KristenAndSara_1280x720</i>	478	66	26
Average		2833	71.61	33.43

4 Conclusion

In this paper, we proposed fast and accurate FRUC algorithm by using 3-level Gaussian pyramid and guidance MV. Proposed algorithm can find more accurate MV when implementing ME process because we used Gaussian pyramid structure instead of simple subsampling. Moreover, we proposed new MV shifting method using guidance MV. By adopting guidance MV, we could reduce errors that occurs when shifting unilateral MV to bilateral MV. As a result, proposed algorithm showed 0.1dB PSNR gain with up to 80times faster than conventional algorithm in average.

“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)”



(a)



(b)



(c)

Fig. 6. Result image: (a) DSME, (b) BI_FRUC and (c) proposed

References

- [1] C. Cafforio, F. Rocca, and S. Turbaro, "Motion compensated image interpolation," *IEEE Trans. Communications*, vol. 38, no. 2, pp. 215 - 222, Feb. 1990.
- [2] F. Dufaux and F. Moscheni, "Motion estimation techniques for digital TV : a review and a new contribution," *Proc. IEEE*, vol. 83, pp. 858 - 876, June 1995.
- [3] M.T. Orchard and C.J. Sullivan, "Overlapped block motion compensation : an estimation-theoretic approach," *IEEE Trans. Image Processing*, vol. 3, no. 9, pp. 693 - 699, Sept. 1994.
- [4] 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.
- [5] 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.
- [6] S.-J. Kang, K.-R. Cho, and Y.H. Kim, "Motion Compensated Frame Rate Up-Conversion Using Extended Bilateral Motion Estimation," *IEEE Trans. Communications*, vol. 53, no. 4, pp. 1759 - 1767, Nov. 2007.
- [7] B.-T Choi, S.-H Lee, and S.-J Ko, "New Frame Rate Up-Conversion Using Bi-Directional Motion Estimation," *IEEE Trans. Consumer Electronics*, vol. 46, no. 3, pp. 603 - 609, Aug. 2000.
- [8] D.-G. Yoo, S.-J. Kang, and Y.H. Kim, "Direction-Select Motion Estimation for Motion-Compensated Frame Rate Up-Conversion," *Journal of Display Technology*, vol. 9, no. 10, pp. 840 - 850, Oct. 2013.
- [9] Adelson, Edward H., et al. "Pyramid methods in image processing." *RCA engineer* 29.6 (1984): 33-41.