New Frame Rate Up-Conversion Algorithms with Low Computational Complexity

Un Seob Kim *Student Member, IEEE* and Myung Hoon Sunwoo, *Fellow, IEEE*

***Abstract*—This paper proposes a new frame rate up-conversion (FRUC) algorithm to reduce the computational complexity and to improve the peak signal-to-noise ratio (PSNR) performance. The proposed FRUC algorithm includes prediction-based motion vector smoothing (PMVS), partial average-based motion compensation (PAMC), and intrapredicted hole interpolation (IPHI). PMVS can efficiently remove outliers using motion vectors of neighboring blocks and PAMC performs motion compensation with the region-based partial average to reduce blocking artifacts of the interpolated frames. For hole interpolation, IPHI uses intraprediction of H.264/AVC to eliminate blurring and also uses the fixed weights implemented using only shift operations, which result in low computational complexity. Compared to the existing algorithms, which use bilateral motion estimation, the proposed algorithm improves the average PSNR of the interpolated frames by 3.44 dB and lowers PSNR performance only by 0.13 dB than the existing algorithm that employs unilateral ME; however, it can significantly reduce the computational complexity of FRUC about 89.3% based on the absolute difference.**

***Index Terms*—Frame interpolation, frame rate up-conversion, H.264/AVC, High Efficiency Video Coding (HEVC), motion compensation, motion estimation.**

I. Introduction

RECENT GROWTH and popularization of multimedia devices have led to the rapid development of video formats; thus, efficient format conversion algorithms have gained more attention. In addition, algorithms for improving the visual quality of video are important because video processing should be performed with limited data by bandwidth capacity and power consumption. Frame rate up-conversion (FRUC) interpolates new frames between original frames to increase the number of frames; thus, FRUC can increase motion continuity. Therefore, it can be used for format conversion, low bit-rate video coding in any video compression standards, and slow motion playback. In particular, FRUC has been in the spotlight with regard to digital TV (DTV) display devices that dramatically use liquid crystal display (LCD). FRUC can reduce the motion blur of LCD by increasing frame rates from 60 frame/s to 120 frame/s or 240 frame/s.

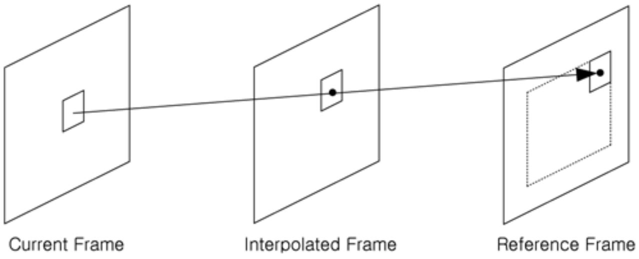

Fig. 1. Unilateral motion estimation.

FRUC algorithms can be divided into two types. One method interpolates intermediate frames by simple frame repetition or averaging. However, this method provides improper results in a picture that contains a lot of motion. The other method, called motion-compensated FRUC (MC-FRUC), considers object movement when it generates intermediate frames and consists of two steps: motion estimation (ME) and motion-compensated interpolation (MCI). ME generates motion vectors (MVs), which represent object motion using vectors, whereas MCI uses MVs to generate intermediate frames.

The block-matching algorithm (BMA) is widely used for ME as it is simple to implement. BMA divides an image into blocks and detects the movement of those blocks. Two kinds of ME are primarily used for BMA: unilateral ME and bilateral ME. Many FRUC algorithms using unilateral ME have been proposed [1]–[8]. As shown in Fig. 1, MVs obtained using unilateral ME pass in one direction through an interpolated frame, which results in overlaps and holes.

To handle overlaps, simple FRUC algorithms [2], [3] merely involve averaging and overwriting the overlapped pixels. Moreover, holes are covered by the pixel values from a reference or a current frame in [2], [3]. However, these algorithms result in blocking artifacts and blurring. Hence, motion field segmentation [1], [4]–[7], successive extrapolation using the discrete Hartley transform [8], and image inpainting [9]–[11] are proposed to handle holes and overlaps without increasing blocking artifacts and blurring. Since these algorithms [1], [4]–[11] iterate global ME and use many filters such as morphological, high-pass, and low-pass filters, they require large computational complexity.

As shown in Fig. 2, bilateral ME is another solution that can be used to avoid the problems caused by overlaps and holes. Bilateral ME obtains MVs passing through a block in the intermediate frame using the temporal symmetry between

Manuscript received November 26, 2012; revised March 29, 2013, July 1, 2013; accepted July 31, 2013. Date of publication August 15, 2013; date of current version March 4, 2014. This work was supported by the Mid-career Researcher Program through an NRF Grant funded by the MEST (2013031132). This paper was recommended by Associate Editor Raouf Hamzaoui.

U. S. Kim is with Telechips Inc., Seoul, Korea (e-mail: uskim@telechips.com).

M. H. Sunwoo is with the School of Electrical and Computer Engineering, Ajou University, Suwon 442-749, Korea (e-mail: sunwoo@ajou.ac.kr).

Digital Object Identifier 10.1109/TCSVT.2013.2278142

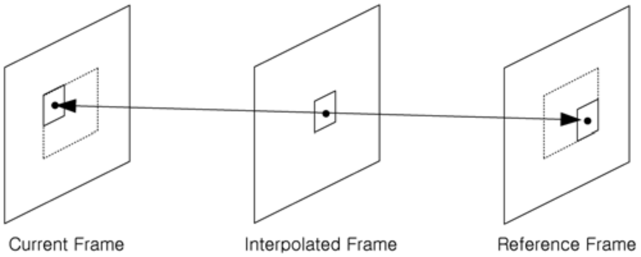

Fig. 2. Bilateral motion estimation.

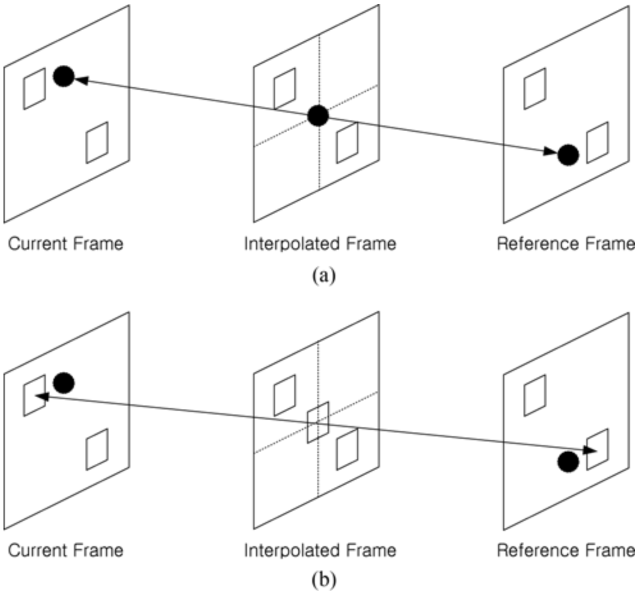

Fig. 3. Problem of bilateral motion estimation. (a) Correct MV. (b) Incorrect MV.

blocks of the reference and current frames. As a result, it does not generate overlaps and holes. There are various algorithms that use bilateral ME [12]–[14]. However, these algorithms show a low peak signal-to-noise ratio (PSNR) performance result since ME based on an unknown block in the intermediate frame has the limited prediction accuracy.

The small squares in Fig. 3 represent stationary objects or backgrounds of the same color, while the black circles indicate moving objects. In this case, bilateral ME uses the temporal symmetry between the current frame and the reference frame. Therefore, the small squares can be detected as moving objects rather than black circles as shown in Fig. 3(b). This phenomenon limits the prediction accuracy and degrades the picture quality.

To obtain more accurate MVs, which indicate true motion, ME and MV smoothing algorithms have been recently proposed [15]–[17]. These algorithms generate multiple MV fields to consider both temporal and spatial correlations; thus, they may achieve good performance. However, to obtain multiple MV fields, these algorithms will have to iterate an ME process that may lead to higher computational complexity.

As described above, the conventional FRUC approaches [1]–[8], [12]–[16] are either too complicated to implement or cannot provide reasonable performance with low computational complexity. Consequently, an algorithm featuring relatively low computational complexity and reasonably good PSNR performance is demanded. Hence, this paper proposes an efficient FRUC algorithm that has low computational complexity. The proposed algorithm is composed of prediction-based motion vector smoothing (PMVS), partial average-based motion compensation (PAMC), and intrapredicted hole interpolation (IPHI).

Conventional MV smoothing [13] employs simple average calculation of neighboring MVs to obtain the threshold value, which is then used to detect outliers. However, neighboring MVs can also be outliers; thus, the average of these MVs may lead to an inaccurate threshold value. However, PMVS neglect neighboring outlier MVs to obtain accurate refined MVs during MV smoothing.

Various overlapped block motion compensation (OBMC) algorithms are used in the FRUC algorithms [13], [18]. However, these OBMC algorithms use a large number of weighting factors and require additional computation to reduce blocking artifacts of the interpolated frames. In contrast, PAMC that considers pixel locations can efficiently reduce blocking artifacts of the interpolated frames with low computational complexity.

The hole interpolation algorithm, which considers correlation with neighboring pixels, was recently proposed [19]. The algorithm employs the trilateral filter using the Markov random model and Euclidean distance to calculate correlation. Because of the exponential computations in the Markov random model and L2 norm computations in the Euclidean distance, these algorithms require large computational complexity and, thus, their hardware implementation is difficult. Therefore, a simple method considering correlation with neighboring pixels is required. The proposed algorithm, named IPHI, uses the intraprediction of H.264/AVC to obtain a smooth image at the borders of objects and regularly patterned objects with reasonably low computational complexity.

The remainder of this paper is organized as follows. Section II describes the proposed FRUC scheme consisting of three proposed algorithms: PMVS, PAMC, and IPHI. Section III presents the experimental results and evaluates the performance of the proposed FRUC. Section IV concludes this paper.

II. Proposed FRUC Algorithms

This section proposes PMVS, PAMC, and IPHI algorithms for low-complexity and reasonably good performance FRUC.

A. Overall Scheme of the Proposed FRUC Algorithms

As mentioned before, bilateral ME based on an unknown block in the intermediate frame has limited ability to accurately locate object movements. Therefore, the proposed FRUC algorithm employs unilateral ME to obtain accurate MVs and uses a combination of forward and backward ME to reduce the number of holes.

Fig. 4 shows the forward and backward ME. The forward ME finds a block that has the minimum sum of absolute difference (SAD) from the reference frame, while the backward ME finds a block that has the minimum SAD from the

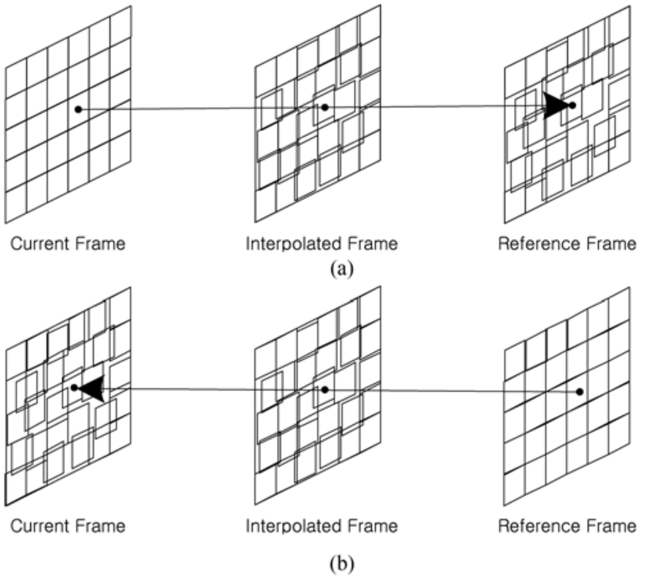

Fig. 4. Forward motion estimation and backward motion estimation. (a) Forward motion estimation. (b) Backward motion estimation.

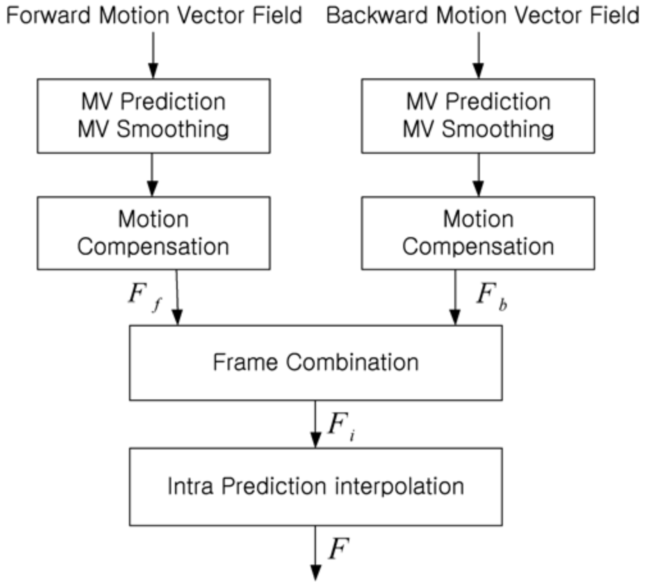

Fig. 5. Flowchart of the proposed FRUC algorithm.

current frame. Forward MV fields and backward MV fields can be obtained by these two ME methods, while MV smoothing is processed to get more reliable MVs.

Fig. 5 shows the flowchart of the proposed FRUC algorithm. The MVs determined by SAD can occasionally be outliers. Unlike ME in video codecs, the MVs used in FRUC should contain information that shows the real movement of objects rather than the minimum residual energy. For this reason, MV smoothing is required to correct these outliers and create a high-quality picture. To refine the MVs, neighboring MVs are used because target object movements are similar with the movements of neighboring objects. Therefore, PMVS, described next in detail, uses the predicted MV to consider MVs of neighboring blocks and efficiently omit outliers.

Next, the MC is performed using the MVs obtained in the previous step. However, it generates blocking artifacts and thus, the proposed algorithm employs PAMC that considers pixel locations. PAMC generates the forward interpolated frame F_f and the backward interpolated frame F_b and linearly combines them to create frame F_i.

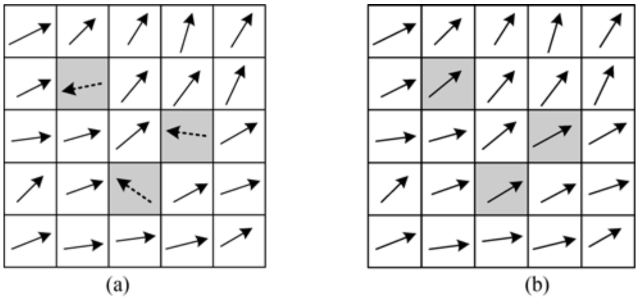

Fig. 6. Comparison of MV fields. (a) Before MV smoothing. (b) After MV smoothing.

The unilateral ME generates holes that should be interpolated to complete FRUC. The proposed algorithm uses IPHI, which considers the correlation among neighboring pixels to produce smoother pictures. To realize hole interpolation, IPHI employs the intraprediction of H.264/AVC. The final interpolated frame F is generated by IPHI.

B. Proposed Prediction-based Motion Vector Smoothing (PMVS)

The MVs determined by the minimum SAD cannot exactly indicate object movement because a SAD is the value that represents residual energy rather than object movement. Therefore, MV outliers indicating incorrect directions appear and they should be corrected by the specific so-called MV smoothing process [13]. Fig. 6(a) shows MVs and outliers of the MV fields before MV smoothing. In Fig. 6(a), the shaded MVs are outliers. Fig. 6(b) also shows the results of MV smoothing, which applied the property that object movement is similar to the movements of neighboring objects. Therefore, the MVs of neighboring blocks are needed to refine the current MV.

The proposed PMVS algorithm consists of three steps. First, the predicted MV of the current block should be obtained by considering MVs of neighboring blocks. The MVs of neighboring blocks can also be outliers and should be neglected from computation. Thus, PMVS uses the MV smoothing results that were previously obtained, which is the main difference between the proposed PMVS and the conventional algorithm [13]. Fig. 7 shows the calculation of the predicted MV, *Vpred,* which is given by

$$Vpred = \frac{1}{3}\left(V1' + V2' + V4'\right). \tag{1}$$

V0 is the MV to be refined, while *V1–V8* are the neighboring MVs. To prevent iterative calculation, only three previously determined MV smoothing results—*V1′*, *V2′* and *V4′*—are used to calculate the predicted MV.

The final refined MV is determined by averaging the neighboring MVs. Thus, outliers are recognized from MVs of nine blocks including the current block *V0*. Thus, the predicted MV obtained in the first step is used. Second, the

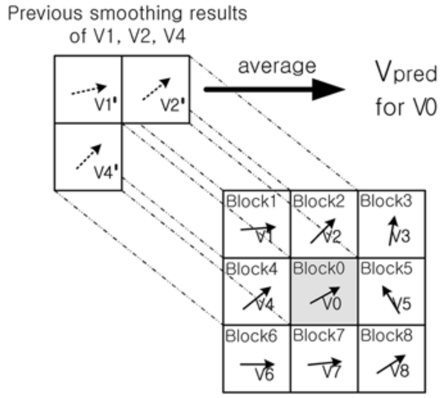

Fig. 7. Predicted MVs.

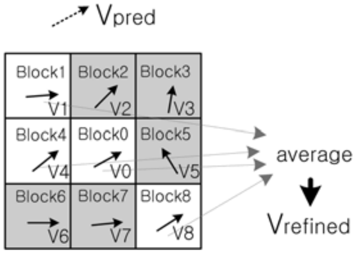

Fig. 8. MV smoothing.

absolute differences between the neighboring MVs and the predicted MV are calculated. The four MVs with the smallest absolute differences among the nine MVs are selected for the final average calculation. Third, the refined MV *Vrefined* is determined by averaging the selected four MVs. Fig. 8 shows an example in which *V1*, *V4*, *V0*, and *V8* are selected as candidates to be averaged.

PMVS is compared with two other algorithms, named the mean filter and the nearest filter. The mean filter uses the mean value of nine neighboring MVs as the refined MV while the nearest filter uses the MV that is the nearest one to the mean value as the refined MV. Table I presents the comparison results, which shows that PMVS has the best PSNR performance. Hence, we select PMVS for MV smoothing.

The 3-D recursive search block matching (3DRS) algorithm [20] is the fast search ME algorithm that finds true motions of objects. As 3DRS uses the neighboring MVs for MV predictions, it can find true motion. Since the proposed FRUC algorithm includes MV smoothing, it can be flexible enough to be used with any ME algorithm. If 3DRS is used for the ME algorithm with the proposed FRUC, MV smoothing can be skipped, which results in low computational complexity.

C. Proposed Partial Average-based Motion Compensation (PAMC)

The MC process is performed using the MV smoothing results. The forward interpolated frame, F_f, between the current frame F_n and reference frame F_{n-1} is given by

$$F(x, y) = \frac{1}{2}\left(F_n\left(x - \frac{v_x}{2}, y - \frac{v_y}{2}\right) + F_{n-1}\left(x + \frac{v_x}{2}, y + \frac{v_y}{2}\right)\right) \qquad (2)$$

TABLE I
PSNR COMPARISONS OF MV SMOOTHING ALGORITHMS

Sequence (dB)	*Foreman*	*Flower*	*Mobile*	*Stefan*	average
Mean filter	27.72	23.39	23.97	22.78	24.47
Nearest filter	28.86	26.73	26.25	23.14	26.25
PMVS	32.71	27.93	26.18	26.32	28.29

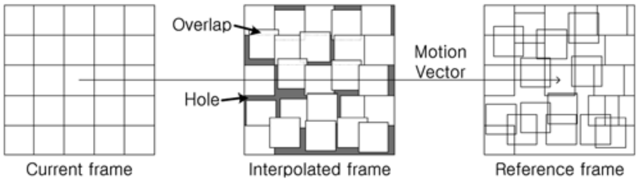

Fig. 9. Overlapped regions and holes.

where (x, y) and (v_x, v_y) represent pixel locations and MV, respectively. MV represented as (v_x, v_y) is associated with F_n and F_{n-1}. The backward interpolated frame is also constructed in the same way.

In the case of unilateral ME, overlaps and holes are generated according to MV trajectory. In Fig. 9, the black regions in the interpolated frame represent holes. Besides, the overlapped areas between the small squares in the interpolated frame are called overlaps.

To reduce the blocking artifacts, various OBMC algorithms are used in FRUC algorithms [13], [18], [21]. However, these OBMC algorithms use a large number of weighting factors, and, thus, they require high computational complexity. In contrast, PAMC performs MC with the region-based partial average to reduce blocking artifacts of the interpolated frames with low computational complexity.

Fig. 10 shows one example. If the location of the current block A is determined by MV and the overlapped regions occur, the SADs of the overlapped blocks are compared. Then, if the SAD of A is smaller than the SADs of the neighboring blocks, the averaged pixels of the current block and the neighboring blocks are taken as the PAMC result for region1 in Fig. 10. In addition, the pixels of A are taken as the PAMC result for the central region except region1 and region2. The region1 in Fig. 10 represents the locations where the averaged pixels are taken. The pixels of the neighboring blocks, on the other hand, are taken as the PAMC result when overlapped regions occur and the SAD of A is larger than those of the neighboring blocks.

The MC algorithm in [13] uses a SAD value of each block as a weight for the overlapped regions. For convenience, the MC algorithm in [13] is called SAD weighted motion compensation (SWMC) in this paper. Table II shows the PSNR comparisons of PAMC, SWMC [13], and OBMC [21]. In OBMC, each block is enlarged from 8×8 to 12×12 for MC to reduce blocking artifacts and holes. Therefore, OBMC shows slightly better PSNR performance compared to PAMC and SWMC for the videos containing many moving objects such as *Football* and *Stefan*.

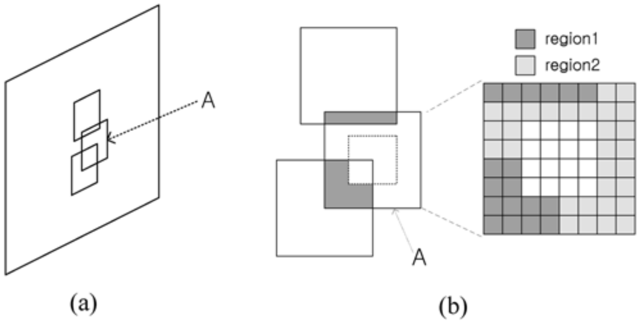

Fig. 10. Motion compensation of overlapped regions. (a) Occurrence of overlapped regions. (b) Proposed motion compensation.

TABLE II
PSNR COMPARISONS OF MC ALGORITHMS

Sequence (dB)	SWMC [13]	OBMC [21]	PAMC
Foreman	32.71	32.55	32.71
Stefan	26.31	26.67	26.32
Mobile	26.17	26.29	26.18
Flower	27.93	27.84	27.93
Football	22.18	22.23	22.17
average	27.06	27.12	27.06

TABLE III
REQUIRED OPERATIONS OF MC ALGORITHMS FOR AN 8 × 8 OVERLAPPED REGION

Required operations	SWMC [13]	OBMC [21]	PAMC
Addition	128	128 + α	48
Subtraction	-	-	1
Multiplication	128	128 + α	-
Division	64	64 + α	-
Shift	-	-	48

Table III shows the comparisons of the required operations among the MC algorithms. To clearly show the required operations, two 8×8 regions are assumed to be completely overlapped. As shown in Table III, PAMC does not require multiplications and divisions because it does not use the complicated weighting factors for the overlapped regions. However, SWMC and OBMC require 128 multiplications and 128 divisions along with three times more additions compared to PAMC. Moreover, using the enlarged blocks in OBMC results in additional computations that are denoted by α in Table III. Therefore, as the overlapped regions are increased by the enlarged blocks, α values are also increased. For the test sequences used in Table III, OBMC with the enlarged blocks generates about ten times more overlapped regions than OBMC without the enlarged blocks. In summary, PAMC shows slight loss of PSNR compared to OBMC but it has much lower computational complexity compared to SWMC and OBMC as mentioned in Table II and III.

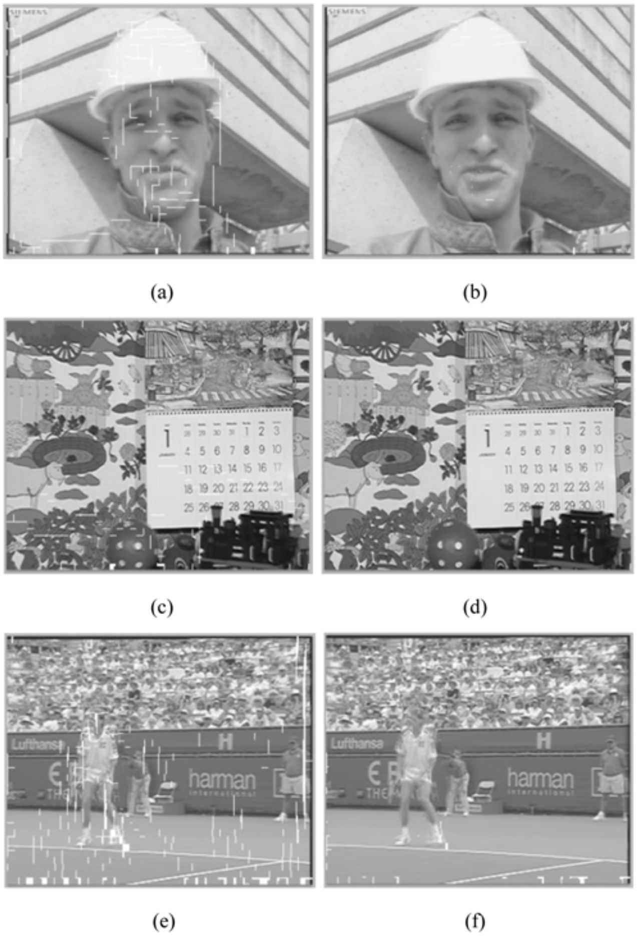

Fig. 11. Comparisons of holes before and after frame combination. (a), (c), and (e) Forward interpolated frame only. (b), (d), and (f) Combined frame.

As described previously, the proposed FRUC algorithm uses the combined results of the forward and backward ME. The proposed FRUC algorithms make one interpolated frame F_i defined as

$$F_i(x,y) = \begin{cases} \dfrac{F_f(x,y)+F_b(x,y)}{2}, \text{ if } F_f(x,y) \neq \mathit{Hole} \text{ and } F_b(x,y) \neq \mathit{Hole} \\ F_f(x,y), \text{ if } F_f(x,y) \neq \mathit{Hole} \text{ and } F_b(x,y) = \mathit{Hole} \\ F_b(x,y), \text{ if } F_f(x,y) = \mathit{Hole} \text{ and } F_b(x,y) \neq \mathit{Hole} \\ \mathit{Hole}, \text{ otherwise} \end{cases} \tag{3}$$

where (x, y) represents the pixel location. If one pixel has two available values respectively from the forward interpolated frame F_f and the backward interpolated frame F_b, the average of the two values is taken as the final pixel. But if only one value is available since there is a hole in either F_f or F_b, the frame combination uses this available value as the final pixel. Otherwise, if there is no available value from the two frames, the pixel remains as a hole and is interpolated in the hole interpolation process.

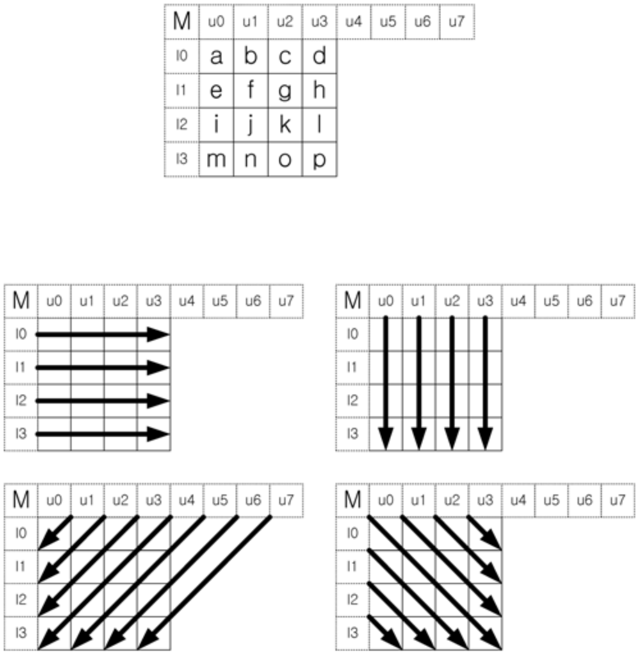

Fig. 12. Pixel locations and direction modes used for intraprediction.

During the joining of two interpolated frames, the hole areas are reduced and the picture quality is improved. Fig. 11 compares the holes before and after frame combination. The irregular white spots and lines represent holes. The pictures come from the 49th frame of *Foreman*, *Mobile* and *Stefan* sequences, which are 352 × 288 pixels in size and have frame rate of 30 frame/s. In general, more holes appear in low-resolution video sequences than those in high-resolution and, thus, the standard CIF (352 × 288) format is used for comparisons. Fig. 11(a), (c), and (e) are holes of the forward interpolated frame only, while Fig. 11(b), (d), and (f) are holes of the frame combination. These figures clearly show that frame combination considerably reduces the number of holes.

D. Proposed Intrapredicted Hole Interpolation (IPHI)

The conventional FRUC algorithms using unilateral ME [1]–[8] commonly use a median filter or a mean filter for hole interpolation. These filters show good performance for flat objects; however, they blur the borders of objects and regularly patterned objects. Therefore, the correlation with neighboring pixels should be considered to eliminate this blurring. The proposed IPHI algorithm employs intraprediction of H.264/AVC to calculate this correlation.

The intraprediction of H.264/AVC analyzes nine directions of a 4 × 4 block; however, IPHI analyzes only four directions, including the horizontal, vertical, down-left and down-right, to reduce computation. We found four directions to be enough to eliminate blurring. In Fig. 12, *u0–u7*, *l0–l3*, and *M* are the neighboring pixels used for prediction, while *a–p* are the predicted pixels. In addition, Table IV lists the prediction equations used for intraprediction.

TABLE IV

PREDICTION EQUATIONS FOR INTRAPREDICTION

DIRECTION MODE	PREDICTION EQUATIONS

VERTICAL	A=E=I=M=U0
	B=F=J=N=U1
	C=G=K=O=U2
	D=H=L=P=U3

HORIZONTAL	A=B=C=D=L0
	E=F=G=H=L1
	I=J=K=L=L2
	M=N=O=P=L3

DOWN-LEFT	A=(U0+U1+U1+U2+2)>>2
	B=E=(U1+U2+U2+U3+2)>>2
	C=F=I=(U2+U3+U3+U4+2)>>2
	D=G=J=M=(U3+U4+U4+U5+2)>>2
	H=K=N=(U4+U5+U5+U6+2)>>2
	L=O=(U5+U6+U6+U7+2)>>2
	P=(U6+U7+U7+U7+2)>>2

DOWN-RIGHT	D=(U1+U2+U2+U3+2)>>2
	C=H=(U0+U1+U1+U2+2)>>2
	B=G=L=(M+U0+U0+U1+2)>>2
	A=F=K=P=(U0+M+M+L0+2)>>2
	E=J=O=(M+L0+L0+L1+2)>>2
	I=N=(L0+L1+L1+L2+2)>>2
	M=(L1+L2+L2+L3+2)>>2

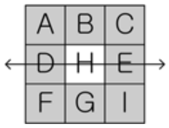

Fig. 13. Hole interpolation using the weighted average.

IPHI determines the best direction mode among the four modes as follows. First, the predicted pixels are obtained using the equations in Table IV. The SAD is then calculated between the pixels that are not holes and the predicted pixels according to the four direction modes. Next, the direction mode with the minimum SAD is determined as the block orientation. After the block orientation is determined, the holes are interpolated with the weighted average based on the orientation. Fig. 13 is an example of the horizontal mode and holes are filled using the following equation:

$$H = \frac{1}{W}(A + B + C + 4D + 4E + F + G + I) \qquad (4)$$

where W is the sum of the weights. The average calculation with the weight of the best direction 4 showed the best performance in various experiments.

IPHI using intraprediction has a considerable effect at the borders of objects and regularly patterned objects. Fig. 14 compares the hole interpolation results using the median filter and IPHI. The pictures are obtained from the *Highway* sequence, which is 352 × 288 pixels in size and has 30 frame/s. The points highlighted with circles are the regions interpolated at the borders of objects and the left picture processed by the median filter show the improper result. However, the right picture processed using IPHI shows a smoother result than the left picture.

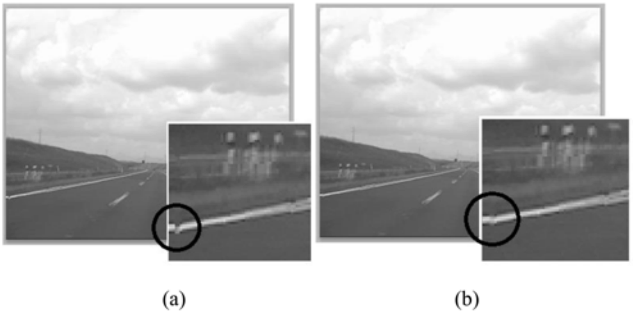

Fig. 14. Comparison of the median filter and IPHI. (a) Median filter. (b) Proposed IPHI.

TABLE V

PSNR COMPARISONS WITH EXISTING ALGORITHM [21]

Sequence (dB)	BDHI (4 × 4) [21]	BDHI (8 × 8) [21]	IPHI
Foreman	32.68	32.70	32.71
Stefan	26.33	26.33	26.32
Mobile	26.16	26.16	26.18
Flower	27.94	27.94	27.93
Football	22.17	22.18	22.17
average	27.06	27.07	27.06

The proposed IPHI and the block-wise directional hole interpolation (BDHI) algorithm [21] have similarity in terms of analyzing four directions with different weighting factors. BDHI first calculates the averages of the absolute differences (ADs) among adjacent interpolated pixels in each of the four directions within the block. Then, the four average ADs are used for weights. Table V shows the PSNR comparisons among IPHI, BDHI using 4×4 blocks and BDHI using 8×8 blocks. Table V shows that the PSNR performances of IPHI and BDHI are quite similar.

Table VI shows the required operations of BDHI (4×4) and IPHI. The number of holes is assumed four to calculate the required operations because multiple holes in general are generated instead of a single hole. Compared to IPHI, BDHI requires less AD operations to interpolate four holes in a 4×4 block. However, BDHI requires 28 multiplications, 28 divisions and fractional operations because it uses the reciprocals of average AD values as weights. Therefore, hardware implementation of IPHI is much easier than that of BDHI having multiplications, divisions, and fractional operations.

In summary, the proposed FRUC consists of all of the three ideas described above. First, using PMVS, neighboring MVs can be considered and outliers can be efficiently neglected. Second, PAMC is used to reduce blocking artifacts of the interpolated frames. Last, IPHI using intraprediction of H.264/AVC is applied to consider the correlations with neighboring pixels.

III. EXPERIMENTAL RESULTS

The image quality of the interpolated frames is evaluated using PSNR. All sequences used for experiments are in the standard CIF (352×288), 720p (1280×720), and 1080p (1920×1080) format and 30 frame/s. The odd frames are removed and the new odd frames are constructed from the even frames using the existing algorithms [2], [17]–[19] and the proposed FRUC schemes. Two FRUC schemes that consist of the proposed algorithms are presented to show how each algorithm improves the performance. Scheme1 is composed of PAMC, frame combination, and hole interpolation using the 5×5 median filter while Scheme2 consists of PMVS, PAMC, frame combination, and IPHI. For all of the experiments, a full-search ME is used with the 8×8 macro block and ± 16 search range.

Table VII presents the average PSNR of 50 interpolated frames. The existing method [2] uses the forward ME and the

TABLE VI

REQUIRED OPERATIONS TO INTERPOLATE FOUR HOLES

Required operations	BDHI (4 × 4) [21]	IPHI
Multiplication	28	-
Division	28	4
Absolute difference	16	48
Addition	32	72
Subtraction	-	3
Shift	-	8

TABLE VII

PSNR COMPARISONS OF SEQUENCES

Sequence (dB)		Existing method [2]	Proposed	
			Scheme1	Scheme2
Foreman	CIF	29.49	31.54	32.71
Akiyo		41.05	41.94	45.11
Coastguard		26.22	28.38	30.75
Flower		25.83	26.81	27.93
Football		20.78	21.79	22.17
Highway		28.65	31.46	32.23
Mobile		21.88	22.66	26.18
News		33.24	34.84	37.25
Hall		32.16	34.59	35.46
Stefan		22.67	24.03	26.32
Silent		32.88	34.99	35.81
Mother&Daughter		37.87	39.23	41.45
Container		34.35	37.52	43.13
Parkrun	720p	23.38	25.51	26.09
Shields	720p	32.17	34.53	34.78
Stockholm	720p	29.12	30.90	33.04
Sunflower	1080p	30.82	31.75	32.60
Bluesky	1080p	23.93	25.01	26.41
average		29.25	30.97	32.75

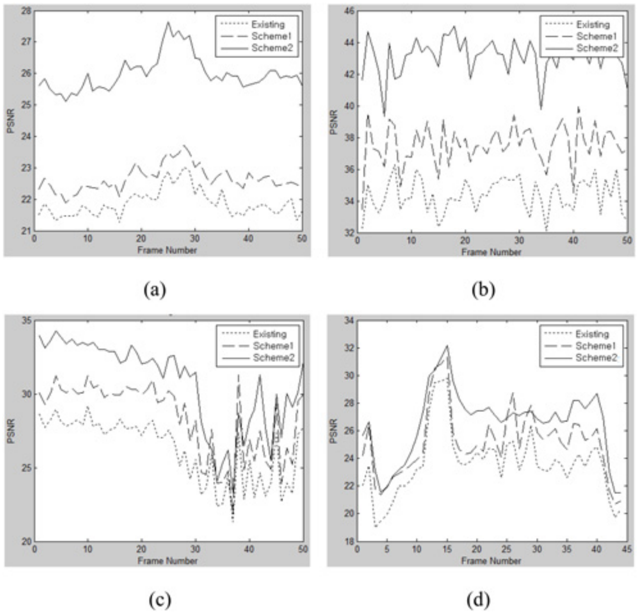

Fig. 15. Comparison of existing and the proposed FRUC algorithms. (a) *Mobile*. (b) *Container*. (c) *Coastguard*. (d) *Stefan*.

hole regions caused by the forward ME are covered by the pixel values from a reference frame or a current frame.

Scheme1 provides 1.72 dB higher average PSNR performance than the existing method [2], while Scheme2 provides 1.77 dB higher average PSNR performance than Scheme1. In particular, Scheme2 shows a notable PSNR improvement for the *Container* sequence. Using PAMC and frame combination, the average PSNR performance improves about 3.17 dB for the *Container* sequence. In addition, PMVS and IPHI improve the average by 5.61 dB.

Fig. 15 shows the PSNRs for the interpolated frames in four sequences: *Mobile*, *Container*, *Coastguard*, and *Stefan*. The graphs show the proposed FRUC algorithms are effective for entire frames.

Table VIII shows PSNR performances between the dual ME [17] and Scheme2 and Table IX compares PSNR performances among VS-BMC [18], the trilateral filter [19], and Scheme2. The existing algorithms [18], [19] used the CIF format, and, thus, the proposed FRUC also use the CIF format to compare with the existing algorithms [18], [19] in Table IX. In Table VIII, the proposed Scheme2 provides an average of 2.69 dB better PSNR performance than the algorithm [17] using the bilateral ME, median filter, and MV smoothing with side match distortion (SMD). On the other hand, Table IX shows that Scheme2 provides an average of 0.13 dB lower PSNR performance than the algorithm [19] using the unilateral ME, high-pass filter, and trilateral filter. VS-BMC [18] in Table IX uses the bilateral ME and SMD to reduce blocking artifacts of the interpolated frames. As shown in Table IX, Scheme2 provides an average PSNR performance about 4.18 dB better than that of VS-BMC.

TABLE VIII

PERFORMANCE COMPARISONS WITH EXISTING ALGORITHM [17]

Sequence (dB)		Dual ME [17]	Proposed Scheme2
News	CIF	34.95	37.25
Stefan		24.32	26.32
Foreman		29.19	32.71
Mother&Daughter		38.73	41.45
Mobile		23.11	26.18
Highway		31.25	32.23
Crew	720p	28.34	32.78
Soccer	1080p	29.62	32.11
Average		29.94	32.63

TABLE IX

PERFORMANCE COMPARISONS WITH EXISTING ALGORITHMS [18], [19]

Sequence (dB)		VS - BMC [18]	Trilateral filter [19]	Proposed Scheme2
Football	CIF	19.69	22.74	22.17
Tennis		25.38	29.58	27.95
Flower		22.56	26.96	27.93
Mobile		19.57	25.09	26.18
Paris		28.05	33.53	31.63
Container		38.65	41.87	43.13
Average		25.65	29.96	29.83

TABLE X

COMPUTATIONAL COMPLEXITIES BASED ON ABSOLUTE DIFFERENCE

Algorithms	Absolute difference (AD)
Dual ME	$2\times(1.5N)^2\times(2\times W+1)^2$ $-\{(1.5N)^4-24(1.5N)^2+144\}=296,208$
Trilateral filter	$2\times\{N^2\times(2\times W+1)^2\}+\alpha=1,305,728+\alpha$
VS-BMC	$N^2\times(2\times W+1)^2+(N^2+4\times N)\times(2\times W+1)^2=174,240$
Scheme2	$2\times\{N^2\times(2\times W+1)^2\}+(N^2\times 4+9\times 2)=139,666$

The computational complexities for the algorithms are calculated in terms of AD required for processing a macro block. Table X shows the AD comparison results, where N and W represent the size of macro block and the size of search window, respectively.

The proposed Scheme2 uses the 8×8 macro block and forward and backward ME with the ± 16 search window. The second term of Scheme2, $(N^2\times 4+9\times 2)$, is also added to AD computations because $N^2\times 4$ is required to determine the best direction in IPHI and PMVS requires 9×2 AD computations.

The dual ME [17], on the other hand, uses bilateral ME as well as uses unilateral ME for MV smoothing with the overlapped block size $1.5N$. The dual ME also uses the 8×8 macro block and a ± 16 search window. The second term in dual ME, $(1.5N)^4\times 24(1.5N)^2\times 144$, denotes the computational complexity for the overlapped area between two blocks with the size of $1.5N$, which should be subtracted from the first term $2\times(1.5N)^2\times(2\times W+1)^2$. The bilateral ME cannot obtain accurate MV; thus, the dual ME that uses bilateral ME also has inaccurate MV estimation. Therefore, the dual ME shows 2.18 dB lower PSNR result and it also has 52.8% more computation than Scheme2 according to

$$\begin{array}{c}\text{Computationalcomplexity}\\ \text{comparison}\end{array} = \frac{comp1 - comp2}{comp1} \times 100 \qquad (5)$$

where comp1 and comp2 represent the computational complexities of the dual ME and Scheme2, respectively.

The trilateral filter [19] uses the luminance SAD, high-pass filtered image SAD, the 8×8 macro block, forward ME and backward ME with ± 50 search range. The number of iterative calculations for the trilateral filtering is denoted by α. The trilateral filter provides an average of 0.16 better PSNR performance than Scheme2 but it has about 89.3% more computations than Scheme2. Moreover, it requires additional computations for the iterative trilateral filtering.

VS-BMC [18] also uses the 8×8 macro block and a ± 16 search window. Moreover, the second term, $(N^2 \pm 4 \times N) \times (2 \times W \pm 1)^2$, is added because it has to carry out iterative calculations and SMD. VS-BMC obtains MVs for interpolation by using the bilateral ME like the dual ME and, thus, VS-BMC shows 4.15 dB lower PSNR performance than Scheme2. Consequently, the number of AD operations in Table X shows that the proposed Scheme2 requires much less computation than the other algorithms [17]–[19].

IV. Conclusion

This paper proposes new PMVS, PAMC, and IPHI algorithms to build a highly efficient FRUC scheme. The proposed FRUC scheme has relatively low computational complexity and reasonably good PSNR performance. The proposed FRUC improves the average PSNR by 4.15 dB over VS-BMC and has about 89.3% less computation than the existing trilateral filter with slight loss of PSNR. The proposed FRUC also shows 2.18 dB better PSNR result and has 52.9% less computation than the dual ME. Moreover, the proposed FRUC can greatly reduce blocking artifacts and pixel permeation at the borders of objects that have rapid motion and regular patterns.

In this paper, only fixed size rectangular blocks are used for ME and MC. Using variable size and shape blocks, the quality of the interpolated frame can be improved further. However, in this case, the computational complexity can sharply increase and this issue can be further investigated in the development of a low complexity MCI method using variable blocks.

References

[1] C.-L. Huang and T.-T. Chai, "Motion-compensated interpolation for scan rate up-conversion," *Opt. Eng.*, vol. 35, no. 1, pp. 166–176, Jan. 1996.

[2] S.-C. Han and J. W. Woods, "Frame-rate up-conversion using transmitted motion and segmentation fields for very low bit-rate video coding," in *Proc. Int. Conf. Image Process.*, 1997, pp. 747–750.

[3] T.-Y. Kuo and C.-C. J. Kuo, "Motion-compensated interpolation for low-bit-rate video quality enhancement," in *Proc. SPIE Conf. Appl. Digital Image Process.*, 1998, pp. 277–288.

[4] S. Liu, C.-C. J. Kuo, and J. W. Kim, "Hybrid global-local motion compensated frame interpolation for low bit rate video coding," *J. Visual Commun. Image Represent.*, vol. 14, pp. 61–79, 2003.

[5] T. Liu, K.-T. Lo, J. Feng, and X. Zhang, "Frame interpolation scheme using inertia motion prediction," *Signal Process.: Image Commun.*, vol. 18, pp. 221–229, 2003.

[6] J. Benois-Pineau and H. Nicolas, "A new method for region-based depth ordering in a video sequence: Application to frame interpolation," *J. Visual Commun. Image Represent.*, vol. 13, pp. 363–385, 2002.

[7] P. Blanchfield, D. Wang, A. Vincent, F. Speranza, and R. Renaud, "Advanced frame rate conversion and performance evaluation," *SMPTE Motion Imag. J.*, pp. 153–159, Apr. 2006.

[8] A. Kaup and T. Aach, "Efficient prediction of uncovered background in interframe coding using spatial extraprediction," in *Proc. ICASSP*, 1994, pp. 501–504.

[9] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. Comput. Graph. SIGGRAPG*, 2000, pp. 417–424.

[10] S. D. Rane, G. Sapiro, and M. Bertalmio, "Structure and texture filling-in of missing image blocks in wireless transmission and compression applications," *IEEE Trans. Image Process.*, vol. 12, no. 3, pp. 296–303, Mar. 2003.

[11] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

[12] T. Ha, S. Lee, and J. Kim, "Motion compensated frame interpolation by new block-based motion estimation algorithm," *IEEE Trans. Consum. Electron.*, vol. 50, no. 2, pp. 752–759, May 2004.

[13] S.-J. Kang, K.-R. Cho, and Y. H. Kim, "Motion compensated frame rate up-conversion using extended bilateral motion estimation," *IEEE Trans. Consum. Electron.*, vol. 53, no. 4, pp. 1759–1767, Nov. 2007.

[14] S.-J. Kang, D.-G. Yoo, S.-K. Lee, and Y. H. Kim, "Multiframe-based bilateral motion estimation with emphasis on stationary caption processing for frame rate up-conversion," *IEEE Trans. Consum. Electron.*, vol. 54, no. 4, pp. 1830–1838, Nov. 2008.

[15] H. Liu, R. Xiong, D. Zhao, S. Ma, and W. Gao, "Multiple hypotheses bayesian frame rate up-conversion by adaptive fusion of motion-compensated interpolations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 8, pp. 1188–1198, Aug. 2012.

[16] T.-H. Tsai and H.-Y. Lin, "High Visual quality particle based frame rate up conversion with acceleration assisted motion trajectory calibration," *J. Display Technol.*, vol. 8, no. 6, pp. 341–351, Jun. 2012.

[17] S.-J. Kang, S.-J. 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.

[18] 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 Trans. Circuits Syst. Video Technol.*, vol. 17, no. 4, pp. 407–416, Apr. 2007.

[19] 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.

[20] G. D. Haan, P. W. A. C. Biezen, H. Huijgen, and O. A. Ojo, "True motion estimation with 3-D recursive search block matching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 5, pp. 368–379, Oct. 1993.

[21] 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.

Un Seob Kim (S'09) received the B.S. and M.S. degrees in electronics engineering from Ajou University, Suwon, Korea, in 2006 and 2011, respectively.

He is with Telechips Research and Development Center, Seoul, Korea. His research interests include video processing and multimedia processor design.

Myung Hoon Sunwoo (SM'01–F'11) received the B.S. degree in electronic engineering from Sogang University, Seoul, South Korea, in 1980, the M.S. degree in electrical and electronics from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 1982, and the Ph.D. degree in electrical and computer engineering from University of Texas at Austin, Austin, TX, USA, in 1990.

He was with the Electronics and Telecommunications Research Institute, Daejeon, Korea from 1982 to 1985, and with Digital Signal Processor Operations, Motorola, Austin, TX, USA, from 1990 to 1992. Since 1992, he has been a Professor with the School of Electrical and Computer Engineering, Ajou University, Suwon, Korea. In 2000, he was a Visiting Professor at University of California, Davis, CA, USA. He had been a Director of the National Research Laboratory sponsored by the Ministry of Science and Technology, Korea, Director of the New Growth Engine Semiconductor Center, and an Executive Director of the Institute of Electronics Engineers of Korea (IEEK). He has authored over 390 papers and also holds 60 patents. His research interests include low power algorithms and architectures, SOC design for multimedia and communications, and application-specific design.

Dr. Sunwoo is a recipient of more than 30 awards including the IEEE Circuits and Systems Society 2013 Chapter of the Year Award, the Best Paper Award from the IEEE Workshop on Signal Processing Systems in 2005, the International SoC Conference in 2003, 2005, 2008, and 2009, and the IEEE Seoul Section in 2004, as well as awards from the Ministry of Commerce, Industry and Energy, Samsung Electronics, and the Institute of Electronics Engineers of Korea. He served as General Chair of International Symposium on Circuits and Systems in 2012, Seoul, Korea, a Technical Program Chair of the IEEE Workshop on SiPS in 2003, and the General Co-Chair of ISOCC and General Chair of the IEEK SOC Conference in 2008. He has been a technical committee member for numerous conferences and societies including IEEE ISCAS, SiPS, BioCAS, DATE, APCCAS, A-SSCC, ISOCC, ASQED, Cool Chips, VLSI-DAT, etc. Dr. Suwoon was an Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS from 2002 to 2003, Guest Editor for *Journal of VLSI Signal Processing* in 2005, and the Guest Editor for *Journal of Signal Processing Systems* in 2012. He has been elected as a member of the Board of Governors for IEEE CASS since 2011 and was a Distinguished Lecturer of the IEEE CASS from 2009 to 2010. He is also the President of the IEEK Semiconductor Society and had been a Chair of the IEEK SOC Design Technical Committee. He was an honorary ambassador of the Korean Tourism Organization. He is also a Chair of the IEEE CASS, Seoul Chapter.