



# Frame Interpolation Algorithm Using Improved 3-D Recursive Search

HongGang Xie<sup>1,2(✉)</sup>, Lei Wang<sup>1</sup>, JinSheng Xiao<sup>3</sup>, and Qian Jia<sup>4</sup>

<sup>1</sup> School of Electrical and Electronic Engineering,  
Hubei University of Technology, Wuhan 430068, China  
xiehg@hbut.edu.cn

<sup>2</sup> Collaborative Innovation Center of Industrial Bigdata,  
Hubei University of Technology, Wuhan, China

<sup>3</sup> School of Electronic Information, Wuhan University, Wuhan 430072, China

<sup>4</sup> School of Physics and Information Engineering, Jiangnan University,  
Wuhan, China

**Abstract.** A low-complexity and high efficiency method for Motion-Compensated Frame Interpolation is developed in this paper. The 3-D recursive search technique is used together with bilateral motion estimation scheme to predict the block motion vector field without yielding the hole and overlapping problems. A stepwise multi-stage block-motion estimation scheme is designed to deal with the complex motion object in a block. To reduce the block artifact and keep the computational efficiency, a simplified median filter is developed to smooth the estimated motion vector field. Experimental results show that the proposed algorithm provides a better image quality than several broadly used methods both objectively and subjectively. The high computational efficiency makes this proposed algorithm a useful tool for real-time decoder of high-quality video sequences.

**Keywords:** Frame rate up conversion

Motion Compensated Interpolation (MCI) · Bilateral Motion Estimation (BME)  
3-D Recursive Search (3-D RS) · Multi-stage block segmentation

## 1 Introduction

Video data is usually encoded to low bitrate when it is transmit through bandwidth-limited channels. To restore the original frame rate and improve the temporal quality, Frame rate up-conversion (FRUC) is necessary at the decoder side. People usually use frame interpolation technique to reconstruct the video. How to accurately reconstruct the skipped frames without introducing significant computational complexity is a key challenge in real-time video broadcast applications.

As most of the video including moving object, algorithms considering motion-compensated frame interpolation (MCFI) have been developed to reduce the motion jerkiness and blurring of moving objects in the interpolated frames caused by some simple approaches of frame reconstruction. The interpolation performance can be improved significantly in this way. The key point in MCFI algorithms to accurately

obtain the motion vector field of the moving objects basing on which interpolating frames including true motion information could be reconstructed faithfully. Considering the lower computational complexity, block-matching algorithms (BMA) are usually used for motion estimation (ME) in most MCFI algorithms [1, 2]. Several approaches for accurate motion estimation have been proposed recently [3–5], among these, the 3-D recursive ME proposed by Hann et al. [6] have been applied to several MCFI scheme due to its fast convergence and the good performance on smoothness of velocity field.

When BMA is used for MCFI, hole and overlapping problems often occur which degrade the qualities of the interpolate frames significantly. Several methods have been proposed to handle the hole and overlapped regions [7–10], for example the median filter [7], and an improved sub-pixel blocking matching algorithm [9]. However these methods are complicated. Bilateral ME (BME), which has been used by several MCFI schemes to estimate the motion vectors of an interpolating frame directly [11, 12], is a scheme preventing the hole and overlapping problems with high efficiency.

General BMAs are based on the assumption that the motion vector in a block is uniform. Block artifact will occur in the interpolated frame when the objects in a block have multiple motions. Block artifact can be reduced by using overlapped block MC (OBMC) technique [13]. However, the quality of the interpolated frame may be degraded due to over smoothing effect when OBMC is used to all blocks uniformly. Kim and Sunwoo [11] dealt well with the block artifact by employing adaptive OBMC and applying the variable-size block MC scheme. Though their algorithm is rather complex, they provide a proper way to reduce block artifact.

In this paper, we propose a low-complexity MCFI method with good performance. The 3DRS and BME are integrated to work for the motion estimation of the interpolated frame, which predict a smooth and accurate motion vector field with low complexity and prevent the occurrences of hole and overlapping regions. The block artifact is reduced by applying a simplified median filter without introducing much computing burden. Moreover, the proposed algorithm applies a motion segmentation scheme to divide a frame into several object regions and using a three-stage block MC (TSBMC) scheme to further reduce the blocking artifacts.

## 2 Proposed Algorithm

The proposed method comprises several steps, as shown in Fig. 1. First, the 3DRS is used together with BME to predict the motion vector field of the interpolated frame from the information in the former and the following frames. The initial block is set to be  $16 \times 16$ . Second, the up-to-three-stage motion segmentation will be performed to ensure that each motion vector in a complicate motion could be accurately estimated. Third, a simplified median filter is performed to further smooth the motion vectors of all the three-stage blocks. Finally, overlapped block motion compensation (OBMC) is employed to generate the interpolated frame.

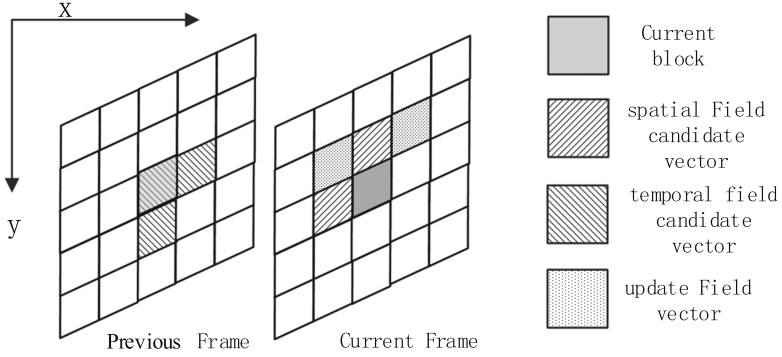


Fig. 1. 3-D RS temporal and spatial estimation candidate vector.

## 2.1 3-D Recursion Search and Bilateral Motion Estimation

We employ 3DRS [6] method to predict the motion vectors of the interpolated frame. The search of the block motion vector is in the order of raster scan. We get the first motion vector estimator  $\vec{V}_a$  of each block in the interpolated frame by scanning the blocks forward from top left to bottom right, and then calculate the second estimator  $\vec{V}_b$  by scanning the blocks backward from bottom right to top left. For a block  $B(\vec{X})$  with  $N \times N$  pixels in the interpolated frame, where  $\vec{X} = (X, Y)$  is the position in the block grids, the  $\vec{V}(\vec{X})$  is obtained by searching the candidate vector set  $CV_a$ :

$$CV_a = \left\{ \begin{array}{l} \vec{V}(\vec{X} - u_x, t), \vec{V}(\vec{X} - u_y, t), \\ \vec{V}(\vec{X} + u_x, t - T), \vec{V}(\vec{X}, t - T), \vec{V}(\vec{X} + u_y, t - T), \\ \vec{V}(\vec{X} - u_x - u_y, t) + U_{\vec{v}}, \vec{V}(\vec{X} + u_x - u_y, t) + U_{\vec{v}} \end{array} \right\} \quad (1)$$

where  $u_x$  and  $u_y$  are horizontal and vertical unit grid in block grids,  $t$  is the time,  $T$  is the field period,  $\vec{V}(\cdot, t)$  is spatial correlated candidate vector which has been estimated,  $\vec{V}(\cdot, t - T)$  is temporal correlated candidate vector which has been obtained from the previously interpolated frame,  $U_{\vec{v}}$  is the update vector which follows [6] as:

$$U_{\vec{v}} = \left\{ \begin{array}{l} \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 0 \\ -2 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} -3 \\ 0 \end{pmatrix} \end{array} \right\} \quad (2)$$

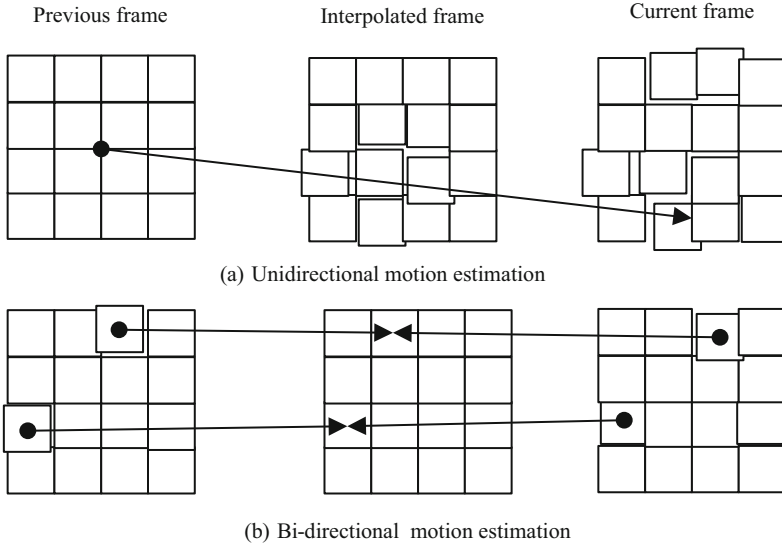
The candidate vectors are shown in Fig. 1. The resulting  $\vec{V}(\vec{X})$  should equal to the candidate vector  $\vec{V}$  in  $CV_a$  with the smallest match error  $e(\vec{V}, \vec{X}, t)$ .

To avoid the occurrence of hole or overlapping problems in the interpolated frame, we apply BME instead of unidirectional estimation (Fig. 2). Information in previous and the following frames are used to calculate the match error. Let  $x$  denote a pixel in

the interpolated frame  $f_t, f_{t-1}$  and  $f_{t+1}$  denote consecutive frames in a video sequence. The match error function  $e(\vec{V}, \bar{X}, t)$  is set to be:

$$e(\vec{V}, \bar{X}, t) = \sum_{x \in B(\bar{X})} |f_{t-1}(x - \vec{V}) - f_{t+1}(x + \vec{V})| \quad (3)$$

Hann et al. [6] added penalties related to the length of the difference vector to the error function to distinguish the priority of different types of candidate vectors.



**Fig. 2.** Unidirectional motion estimation and bilateral motion estimation.

Here we simplify the added penalties  $\alpha$  to three constants 0, 1, 2 for spatial candidate vector, temporal candidate vector, and update vector, respectively. Which assure the priority of the candidate vector being in the order of spatial estimation, temporal estimation and update vector estimation. The estimator  $\vec{V}_a$  is obtained by the following formula:

$$\vec{V} = \arg \min_{\vec{V} \in CV_a} \{e(\vec{V}, \bar{X}, t) + \alpha\} \quad (4)$$

We then search backward to get the second estimator  $\vec{V}_b$  for each block  $B(\bar{X})$ . The candidate set of motion vector now is  $CV_b$  (as shown in Fig. 1):

$$CV_b = \left\{ \begin{array}{l} \vec{V}(\bar{X} + u_x, t), \vec{V}(\bar{X} + u_y, t), \\ \vec{V}(\bar{X} - u_x, t - T), \vec{V}(\bar{X}, t - T), \vec{V}(\bar{X} - u_y, t - T), \\ \vec{V}(\bar{X} - u_x + u_y, t) + U_{\vec{V}}, \vec{V}(\bar{X} + u_x + u_y, t) + U_{\vec{V}} \end{array} \right\} \quad (5)$$

$\vec{V}_b(\bar{X})$  is obtained from  $CV_b$  by the same way as obtaining  $\vec{V}_a(\bar{X})$ . The final estimated displacement vector  $\vec{V}(\bar{X})$  for block  $B(\bar{X})$  is set to be the estimator with the smaller match error. i.e.

$$\vec{V}(\bar{X}) = \begin{cases} \vec{V}_a(\bar{X}), & \text{if } e(\vec{V}_a, \bar{X}, t) < e(\vec{V}_b, \bar{X}, t) \\ \vec{V}_b(\bar{X}), & \text{if } e(\vec{V}_a, \bar{X}, t) > e(\vec{V}_b, \bar{X}, t) \end{cases} \quad (6)$$

$\vec{V}(\bar{X})$  is assigned to all the pixels in block  $B(\bar{X})$ .

## 2.2 Multi-stage Block Motion Estimation

After the 3DRS and BME, we get the estimated motion vector and the match error for each block  $B(\bar{X})$  in the interpolated frame. The initial block size is set to be  $16 \times 16$  pixels in this paper. For a block with multiple moving object, the estimated vector is not the actual vector for all the pixels in this block which will result in a quite big match error. Thus we can find these blocks out and search the proper motion vectors for different pixels in this block in a way described as follows.

### Multi-stage Block Segmentation

1. *Perform the simplified median filter.* If the match error of a block is larger than a predefined threshold, the block is labeled to be processed further.
2. *Split the labeled block with size of  $16 \times 16$  pixels into four  $8 \times 8$  sub-blocks; Estimate the motion vector of each sub-block by using the 3DRS and BME method.* Perform the simplified median filter; Assign the new estimated motion vector to pixels in the sub-block. If the match error of a sub-block is larger than  $\tau/4$ , the sub-block is labeled.
3. *Split the labeled  $8 \times 8$  sub-block into four  $4 \times 4$  sub-blocks; Estimate the motion vector of each sub-block by using Hexagon search method.* Assign the new estimated motion vector to pixels in the corresponding  $4 \times 4$  sub-blocks. Perform the simplified median filter. If the match error of a  $4 \times 4$  sub-block is larger than, the motion vector of this sub-block is set to be the median of its neighbor blocks.

The simplified median filter method will be described in the following section.

### Multi-stage Block Motion Vector Correction.

If the motion field estimated in some positions (usually at boundaries of some blocks) are discontinuous, motion compensation may introduce visible block structures in the interpolated picture. The size we adopted here will give rise to very visible artifacts. A post-filter on the vector is often used to overcome this problem [1].

It has to be pointed out that the classical  $3 \times 3$  block median filter is rather complex for an on-time FRUC algorithm. Therefore we simplify the median filter to lower the computational complexity of proposed MCI algorithm.

For a block  $B(\bar{X})$  of size  $N \times N$  ( $N = 16, 8, \text{ or } 4$ ), the median filter is performed on a window of  $3 \times 3$  blocks of the same size centered at  $B(\bar{X})$ . We label each of the nine blocks with a certain number between 1 and 9, and denote them as  $B_k, k = 1, \dots, 9$ . We set penalties  $P_x(k)$  and  $P_y(k)$  to each of the  $x$  and  $y$  components of the estimated vector of block  $B_k$ . we sort the  $x$  and  $y$  of the estimated vector separately in descending order, and denote the respective ordered matrix of subscript as  $I_x$  and  $I_y$ . Let  $AP = (4, 3, 2, 1, 0, 1, 2, 3, 4)$  and  $BP = (20, 15, 10, 5, 0, 5, 10, 15, 20)$  be two constant matrixes. We also denote the estimated vector of the center block as  $\vec{V} = (v_x, v_y)$ .  $P_x(k)$  and  $P_y(k)$  are set as following:

$$\begin{aligned} \text{if } v_x > v_y, & \begin{cases} P_x(k) = BP(I_x(k)), \\ P_y(k) = AP(I_y(k)) \end{cases} & k = 1, \dots, 9 \\ \text{else,} & \begin{cases} P_x(k) = AP(I_x(k)), \\ P_y(k) = BP(I_y(k)) \end{cases} & k = 1, \dots, 9 \end{aligned} \tag{7}$$

After that, we find out the block  $B_{k_0}$  with the minimum sum of  $P_x(k_0)$  and  $P_y(k_0)$ . The median vector  $\vec{V}_m = (v_{mx}, v_{my})$  of this  $3 \times 3$  window is set to be estimated vector of  $B_{k_0}$ . The estimated vector  $\vec{V} = (v_x, v_y)$  of the central block  $B(\bar{X})$  is replaced according to the following rule:

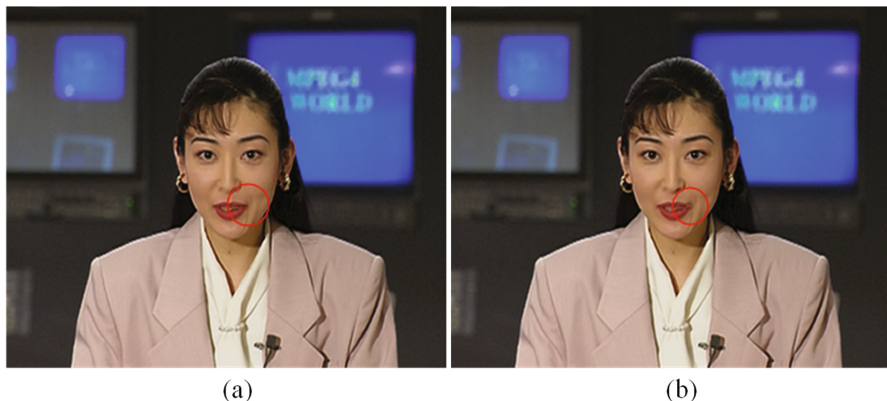
$$\vec{V} = \begin{cases} \vec{V}, & \text{when } |v_x - v_{mx}| < T, \text{ and } |v_y - v_{my}| < T \\ \vec{V}_m, & \text{otherwise} \end{cases} \tag{8}$$

where  $T = 8, 4$  and  $2$  for the blocks of size  $16 \times 16, 8 \times 8,$  and  $4 \times 4$  pixels, respectively. This simplified median filter method is effective in finding out the actual motion vector and lower the complexity of the post-filter significantly.

After the motion field of the interpolated frame is obtained, we reconstruct the interpolated frame by using the information in the previous and the following frames according to the following formula:

$$f(x, t) = \frac{1}{2} (f_{t-1}(x - \vec{V}) + f_{t+1}(x + \vec{V})) \tag{9}$$

We perform this simplified median filter method and a classical median-filter method [1] to interpolate the even frames in akiyo video sequence for comparison. The interpolated 142th frames by these two methods are shown in Fig. 3. It shows that the proposed filter method is effective in reducing the block artifacts.



**Fig. 3.** The 142th interpolated frame in akiyo Sequence. (a) Interpolated Frame Obtained by MV Median Filter Method in [1]. (b) Interpolated Frame Obtained by improved MV Median Filter.

### 3 The Experiment Result and Analysis

Eight video sequences (YUV4:2:0) are used to demonstrate the performance of the proposed algorithm. Seven of them are in CIF standard format, which are Football, Bowling, Susan, Carphone, News, Silent, and Forman sequences; the Sunflower sequence is in HD standard format. These eight video sequences involve almost all kinds of motions except for rotating and zooming, therefore, the evaluation of the proposed algorithm is convincing.

In evaluating, the frame rate of each sequence is halved first by skipping the even frames. And then we interpolate the skipped frames to restore the original frame rate by applying the proposed MCFI algorithm.

#### 3.1 Objective Evaluation

The quality of interpolated frame is measured by computing the PSNR between the interpolated frame and the corresponding original frame. We implemented two other methods and compare the PSNR with our proposed method. Method 1 is full search BME algorithm with traditional median filter for post-processing of estimated motion vector. The block size is set to be  $16 \times 16$  pixels for BME step, and the search radius is 8 blocks. Method 2 is a MCI algorithm based on predictive motion vector field adaptive search technique described in [14]. We also cite the PSNR results of Method 3 [15], where only four video sequences in CIF standard format are involved. The PSNR results are shown in Table 1. The average PSNR values of the eight test sequences are 32.47, 33.14, and 33.22 for method 1, method 2 and the proposed method. The proposed method achieves higher PSNR performance in average than the other methods. The proposed method performs better than method 1 in 6 test sequences except for Carphone and Forman sequences, and better than method 2 in 7 test sequences except for the Sunflower sequence. In the Football sequence and Susan sequence, the PSNR of proposed method is increased more than 2 dB comparing to method 1.

**Table 1.** Average PSNR (dB) of different test sequences adopting

Sequence	Number of Interpolated Frames	Method 1	Method 2 [14]	Method 3 [15]	Proposed Method
Bowing	150	42.55	42.61	—	42.71
News	150	34.62	35.33	35.60	35.46
Silent	150	34.90	35.67	35.44	35.79
Forman	150	33.72	33.21	32.37	33.37
Susan	14	28.60	30.73	—	30.98
Football	130	20.37	22.60	21.32	22.70
Carphone	191	29.99	29.65	—	29.71
Sunflower	250	35.02	35.31	—	35.06

Table 2 compares the average processing time of three methods. For the seven test sequences in CFI standard format, the total average processing time are 178.76 ms, 44.95 ms, and 30.85 ms for method 1, method 2 and the proposed, respectively. The speed of proposed method is obviously faster than the other two methods. While for the Sunflower sequence in HD standard format, the advantage of the proposed method is more prominent. These indicate the computational complexity of the proposed method is greatly lower than the other two methods.

**Table 2.** Average times (ms) to interpolate frame for algorithms above

Sequence	Method 1	Method 2	Proposed Method
Carphone	24.95	7.57	4.01
An	25.64	8.92	8.92
Bowing	30.94	6.77	4.29
News	31.57	5.56	4.18
Football	31.60	9.20	4.84
Silent	34.06	6.93	4.61
Sunflower	590.07	174.46	56.80

### 3.2 Subjective Evaluation

As most of the video sequences are used for viewing, subject image quality is as important as the object quality. Figure 4 shows the 570 interpolated frame in Kristen And Sara 720P video sequence. It can be seen that the subject quality of the proposed method is better than method 1 in the parts of hand and necklace, and better than method 2 in the detail of hand.





**Fig. 4.** Subjective Quality in the interpolated Frame of Kristen And Sara Sequence. (a) No. 1 Method. (b) No. 2 Method. (c) Method in this paper. (d) Enlarge Fig (a) partially. (e) Enlarge Fig (b) partially. (f) Enlarge Fig (c) partially.

## 4 Conclusion

This paper proposes a multi-stage block MCI FRUC algorithm. 3DRS and BME is adopted to estimate the motion vector of the interpolated frame. A simplified median filter method is designed to post process the motion field. The penalty in error function of classical 3DRS is improved. We compared the performance of the proposed algorithm with those of other two methods. Method 1 is the conventional full search motion estimation plus median filter, method 2 is an adaptive BME algorithm. Test results demonstrate that the proposed algorithm provides better image quality than the other two methods both objectively and subjectively. Specifically it is shown that the computational complexity of the proposed algorithm is rather low. For all the seven CFI test sequences, the proposed algorithm runs 5.7 times faster than method 1 in average, and 1.5 times faster than method 2; while for the HD test sequence, the proposed algorithm runs 10 times faster than method 1 and 3 times faster than method 2. The proposed algorithm is suitable for the application of real-time FRUC of HD videos.

**Acknowledgment.** This work was supported in part by the National Natural Science Foundation of China (Grant No. 61573002), and Hubei Provincial Natural Science Foundation of China (Grant No. 2016CFB499).

## References

1. Zhai, J., et al.: A low complexity motion compensated frame interpolation method. In: IEEE International Symposium on Circuits and Systems, pp. 4927–4930. IEEE (2005)
2. Wu, C.M., Huang, J.Y.: A new block matching algorithm for motion estimation. In: Applied Mechanics and Materials, vol. 855, pp. 178–183. Trans Tech Publications (2017)
3. Konstantoudakis, K., et al.: High accuracy block-matching sub-pixel motion estimation through detection of error surface minima. *Multimedia Tools Appl.* 1–20 (2017)
4. Al-kadi, G., et al.: Meandering based parallel 3DRS algorithm for the multicore era. In: IEEE International Conference on Digest of Technical Papers, pp. 21–22 (2010)
5. Takami, K., et al.: Recursive Bayesian estimation of NFOV target using diffraction and reflection signals. In: IEEE International Conference on Information Fusion (FUSION), pp. 1923–1930 (2016)
6. De Haan, G., et al.: True-motion estimation with 3-D recursive search block matching. *IEEE Trans. Circuits Syst. Video Technol.* **3**(5), 368–379 (1993)
7. Kuo, T.Y., Kuo, C.-C.J.: Motion-compensated interpolation for low-bit-rate video quality enhancement. In: Proceedings of SPIE Visual Communication Image Process, vol. 3460, pp. 277–288 (1998)
8. Yang, Y.-T., Tung, Y.-S., Wu, J.-L.: Quality enhancement of frame rate up-converted video by adaptive frame skip and reliable motion extraction. *IEEE Trans. Circuits Syst. Video Technol.* **17**(12), 1700–1713 (2007)
9. Xiao, J., et al.: Detail enhancement of image super-resolution based on detail synthesis. *Signal Process. Image Commun.* **50**, 21–33 (2017)
10. Jeon, B.-W., Lee, G.-I., Lee, S.-H., Park, R.-H.: Coarse-to-fine frame interpolation for frame rate up-conversion using pyramid structure. *IEEE Trans. Consum. Electron.* **49**(3), 499–508 (2003)
11. Kim, U.S., Sunwoo, M.H.: New frame rate up-conversion algorithms with low computational complexity. *IEEE Trans. Circuits Syst. Video Technol.* **24**(3), 384–393 (2014)
12. Kim, J.-H., et al.: Frame rate up-conversion method based on texture adaptive bilateral motion estimation. *IEEE Trans. Consum. Electron.* **60**(3), 445–452 (2014)
13. Orchard, M.T., Sullivan, G.J.: Overlapped block motion compensation: an estimation-theoretic approach. *IEEE Trans. Image Process.* **3**(5), 693–699 (1994)
14. Li, L., Hou, Z.-X.: Research on adaptive algorithm for frame rate up conversion. *Appl. Res. Comput.* **4**(26), 1575–1577 (2009)
15. Choi, K.-S., Hwang, M.-C.: Motion-compensated frame interpolation using a parabolic motion model and adaptive motion vector selection. *ETRI J.* **33**(2), 295–298 (2011)