A New Frame Interpolation Method with Pixel-level Motion Vector Field

Chuanxin Tang, Ronggang Wang*, Wenmin Wang, Wen Gao
School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, Shenzhen 518055, China
tangchuanxin1@163.com,{rgwang,wangwm,wgao}@sz.pku.edu.cn

***Abstract*—In this paper, a new frame interpolation method with pixel-level *motion vector field* (MVF) is proposed. Given that existing methods cannot handle occlusions and blocking artifacts well, there are three contributions in our method: (i) applying the pixel-level *motion vectors* (MVs) estimated by optical flow algorithm to eliminate blocking artifacts (ii) motion post-processing to keep spatial consistency (iii) robust warping method to address collisions and holes caused by occlusions. The method could remove blocking artifacts and alleviate the artifacts caused by occlusions. Experimental results show that the proposed method outperforms existing methods both in terms of objective and subjective performances, especially for sequences with complex motions.**

***Index* Terms— frame interpolation, motion estimation, motion vector field, frame rate up-conversion, motion compensated frame interpolation**

I. INTRODUCTION

Frame rate up-conversion (FRUC) increases the number of displayed images per second in a video or film sequence. A simple FRUC method is frame repetition or frame averaging, which produces blurring and motion jerkiness around moving objects. To handle those problems, frames could be interpolated using *motion-compensated frame rate up-conversion* (MC-FRUC) method. Generally, MC-FRUC is composed of two steps: *motion estimation* (ME) and *motion-compensated frame interpolation* (MCFI).

In conventional MC-FRUC algorithms, *block matching algorithm* (BMA) [1] is typically applied to estimate motion vectors and the new frame is interpolated along the motion trajectories. However, there are at least two problems in this kind of methods. The first problem is the holes and collisions caused by occlusions and motion estimation errors. The second problem is the blocking artifacts in the interpolated frame due to the block-level motion vectors.

To handle the mentioned problems, a number of algorithms have been proposed. For instance, bidirectional ME [2] and *overlapped block ME* (OBME) [3] are proposed to increase the accuracy of the estimated MVs. To reduce blocking artifacts, the *overlapped block motion compensation* (OBMC) [1][2] was proposed. However, if a block is on the boundary of an object, blocking artifacts could still occur. Another approach for alleviating blocking artifacts is to use pixel-level *MV selection* (MVS) [4]. Besides, Dikbas and Altunbasak [8] proposed the pixel-based bilateral MVF from unidirectional MVs to enhance the motion accuracy. Although the above methods alleviate somewhat blocking artifacts, the motion vector candidates for each pixel are still obtained from *block motion estimation* (BME). Since pixels in the same block may belong to different objects, the MVs obtained from BME are not robust enough.

Our scheme to frame interpolation aims to overcome the problems presented previously. Optical flow algorithm is applied to estimate bidirectional pixel-level MVs. Motion post-processing method based on image segmentation is utilized to improve the robustness of MVFs. The occluded regions are detected by motion trajectory tracking, and the detected occluded regions in the intermediate frame are generated by referencing either the previous frame or the next frame, and the non-occluded regions are generated by referencing both frames. Our experiments testified the effectiveness of the proposed approach compared with conventional block-based methods.

The rest of the paper is organized as follows. The proposed method is presented in Section II. Experimental results are discussed in Section III. Finally, this paper is concluded in Section IV.

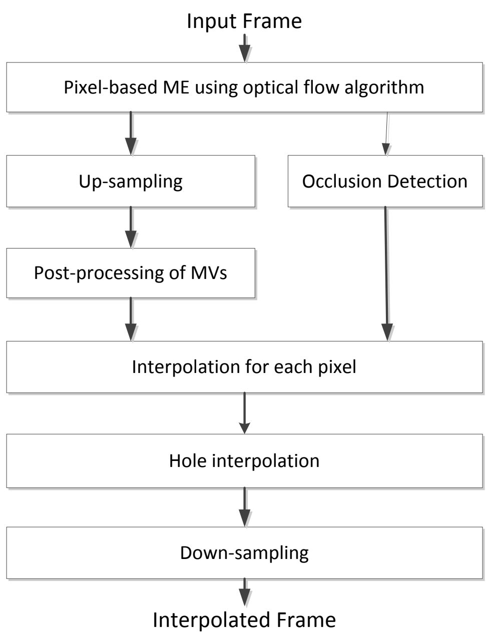

Fig. 1. The overall block diagram of the proposed system

II. PROPOSED METHOD

The proposed method uses two adjacent frames F_t and F_{t+1} to interpolate frame $F_{t+1/2}$. Fig. 1 shows the overall block diagram of the proposed method. Bidirectional pixel-level MVs are estimated using optical flow algorithm in [5]. Then we up-sample the reference frames by a factor of 2 both in

978-1-4799-6139-9/14/$31.00 ©2014 IEEE

horizontal and vertical directions and double the float MVs. Motion post-processing based on image segmentation is proposed to improve motion spatial consistency. The intermediate frame is interpolated based on the post-processed MVFs. At last, remaining holes in the intermediate frame is handled and the intermediate frame is down-sampled to original size.

A. Bidirectional Motion Estimation

We use the optical flow algorithm proposed in [5] to estimate both forward and backward motion fields V_t and V between every two adjacent frames F_t and F_{t+1}. In the forward motion fields V_t , each motion vector is associated with a pixel in the previous frame and points to the next frame; whereas in the backward motion fields V_{t+1}^b, each motion vector is associated with a pixel in the next frame and points to the previous frame.

(a)

(b)

Fig. 2. (a) Without up-sampling and down-sampling (b) With up-sampling and down-sampling

B. Up-sampling

The motion vector value obtained from optical flow is a float number. If it is round to integer, there will be jagged edges in the interpolated frame. Fig. 2(a) is an example of intermediate frame generated by integer MVs. To alleviate this problem, we up-sample the reference frames by a factor of 2 both in horizontal and vertical directions and double the float MVs before rounding. So the interpolated frame is also enlarged, we down-sample it to original size before outputting. Fig. 2(b) shows the interpolated frame with the proposed method, from which we can see the artifact of jagged edges is avoided.

C. Motion Post-processing

On the other hand, since the MV of each pixel is estimated independently by optical flow. The MVs tend to be various even within the same object. In order to maintain spatial consistency of pixel-level motion, post-processing of MVs based on image segmentation is utilized. For each MV, we select a window around the pixel and use the MVs within the window to smooth the MV of the center pixel. The window size was set to 11×11. The updated MV is calculated as follows:

$$\mathbf{MV}_C = \frac{1}{N}\sum_{j\in W}(\mathbf{MV}_j \cdot \alpha_j) \tag{1}$$

where W is the neighborhood pixel set around the current pixel and N is the number of pixel belongs to the same region with the center pixel within W and j is the index of pixel within the window and α is defined as

$$\alpha_j = \begin{cases} 1\text{, } \textit{if pixel } j \textit{ belongs to the same region} \\ \quad \textit{with center pixel} \\ 0\text{, } \textit{otherwise} \end{cases} \tag{2}$$

We use a simple image segmentation algorithm in [6] to segment the images into multiple regions.

D. Occlusion Detection and MV Refinement

Occlusion refers to the covered and uncovered areas only existing in one of the reference frames. The occluded regions in the intermediate frame should be generated from either the previous frame or the next frame. So it is important to detect occluded regions.

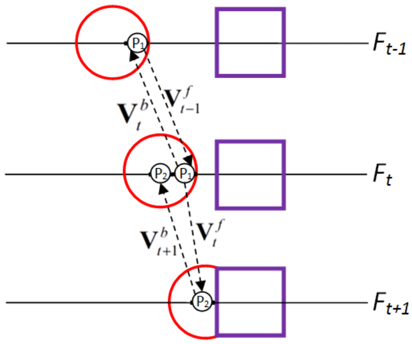

Fig. 3. Occlusion Detection based on the motion trajectory

The reliability of MV is important for occlusion detection. The proposed method utilizes SAD and *MV Distance* (MVD) to determine whether the MV is reliable. SAD is the absolute difference between corresponding pixels and MVD is determined as follows:

$$MVD_t(\mathbf{x}) = \| \mathbf{V}_{t,x}^f - \mathbf{V}_{t+1,x'}^b \|, for\ \mathbf{x} + \mathbf{V}_{t,x}^f = \mathbf{x}' \tag{3}$$

where ||.|| is the Euclidean distance. If the SAD of a pixel is smaller than a threshold of Th_SAD and the MVD described in Eq.(3) is smaller than a threshold of Th_MVD, then the MV of the pixel is reliable.

The occluded regions are detected by tracking motion trajectory as shown in Fig. 3. In Fig. 3, there are three consecutive frames and the pixel P_1 in frame t is occluded by the rectangle in frame t+1. So we can obtain correct MV (V_t^b) of P_1 between frame t-1 and t, but fail to get correct MV (V_t^f) between frame t and t+1. The correctness of the MV is determined by the mentioned SAD and MVD. Then, we mark the

V_t^f of P_1 as occlusion and set the value of V_t^f as $-V_t^b$. The reason we use opposite direction of V_t^b to update V_t^f is that we suppose pixels move continuously among successive frames.

E. Interpolation

Interpolation stage consists of two steps: first, using both forward and backward MVFs to generate the forward interpolated frame F^f and the backward interpolated frame F^b; second, merging F^f and F^b to generate the final interpolated frame $F_{t+1/2}$.

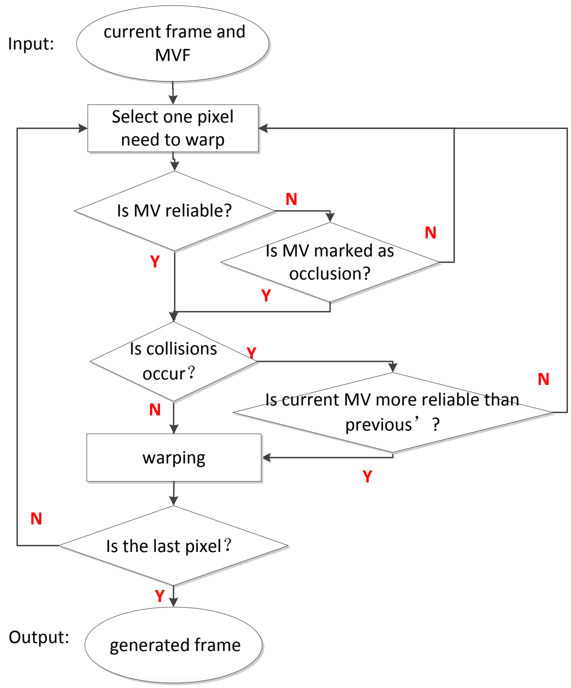

Fig. 4. Block diagram of generating forward interpolated frame (F^f)

1) Step 1: Generating F^f and F^b

Fig. 4 is the overall block diagram of generating F^f. For each pixel to be warped, if its MV is not reliable and is not marked as occlusion, then drop the pixel and deal with the next pixel. If its MV is reliable or marked as occlusion, then determine whether collisions occur. If pixel P and a previous pixel P' are warped to the same position, then it is defined as 'collisions'. If collisions occur and current MV of pixel P is less reliable than the MV of the previous pixel P', then drop the pixel and deal with the next pixel. Otherwise, we warp the pixel to the F^f. The forward warping equation is defined as follows,

$$F^f(\mathbf{x}+\frac{\mathbf{V}_t^f(\mathbf{x})}{2})=F_t(\mathbf{x}) \tag{4}$$

where **x** is the coordinate of the pixel in frame t.

The method of generating F^b is the same as F^f described in Fig. 4. The backward warping equation is defined as follows,

$$F^b(\mathbf{x}+\frac{\mathbf{V}_{t+1}^b(\mathbf{x})}{2})=F_{t+1}(\mathbf{x}) \tag{5}$$

where **x** is the coordinate of the pixel in frame t+1.

2) Step 2: Merging F^f and F^b

The interpolated frame of $F_{t+1/2}$ is obtained by the following process,

$$F_{t+\frac{1}{2}}(\mathbf{x},\mathbf{y})=\begin{cases}\dfrac{F^f(\mathbf{x},\mathbf{y})+F^b(\mathbf{x},\mathbf{y})}{2}, \text{if } F^f(\mathbf{x},\mathbf{y})\neq Hole \quad and\ F^b(\mathbf{x},\mathbf{y})\neq Hole\\ F^f(\mathbf{x},\mathbf{y}), \text{if } F^f(\mathbf{x},\mathbf{y})\neq Hole \quad and\ F^b(\mathbf{x},\mathbf{y})=Hole\\ F^b(\mathbf{x},\mathbf{y}), \text{if } F^f(\mathbf{x},\mathbf{y})=Hole \quad and\ F^b(\mathbf{x},\mathbf{y})\neq Hole\\ Hole, otherwise\end{cases} \tag{6}$$

where **x** and **y** represent the pixel location. If both the forward interpolated frame F^f and the backward interpolated frame F^b have available values, the value of to be interpolated pixel is set as the averaging of the two values. If only one value is available since there is a hole in either F^f or F^b, the value of to be interpolated pixel is set as this available value. Otherwise, the pixel is a hole. We use the same method proposed in [7] to handle holes.

III. Experimental Results and Discussion

The performance of the proposed method was evaluated by several experiments. The CIF sequences of Football, Foreman, Bus, Ice, Highway, News and Soccer were used. Even frames were skipped and the frame interpolation methods were applied to interpolate the skipped frames. All the test sequences have a frame rate of 30 frames per second and each of them consists of 101 frames.

The proposed method is compared with the recent state-of-the-art approach in [8] and other two block-based MCFI methods: dual-ME [9] and correlation [10] methods. In the proposed method, Th_MVD is set to 3, and Th_SAD is set to 20.

A. Comparison of the Objective Performance

The objective evaluation is based on PSNR. Table 1 presents the average PSNRs of 50 interpolated frames by various methods. From the table, we can see that the proposed method outperforms the other methods for all the test sequences. The proposed method is effective especially on sequences with fast motions. The algorithm in [8] and the proposed method obtain significantly improved objective results compared to the MCFI methods. The proposed method obtains better results than the algorithm in [8] and the average improvement in PSNR is 1.02dB.

TABLE I.
Average PSNRs of the 50 Interpolated Frames from Various Methods

Video Sequences	**Dual ME[9]**	**Correlation[10]**	**[8]**	**Proposed**
Football	22.38	22.63	22.58	**23.34**
Forman	33.24	33.46	34.31	**34.96**
Bus	25.53	25.24	26.86	**28.55**
Ice	29.30	31.74	32.24	**34.36**
Highway	-----	-----	33.23	**33.51**
News	-----	-----	37.77	**37.86**
Soccer	-----	-----	29.33	**30.85**
Average	-----	-----	30.90	**31.92**

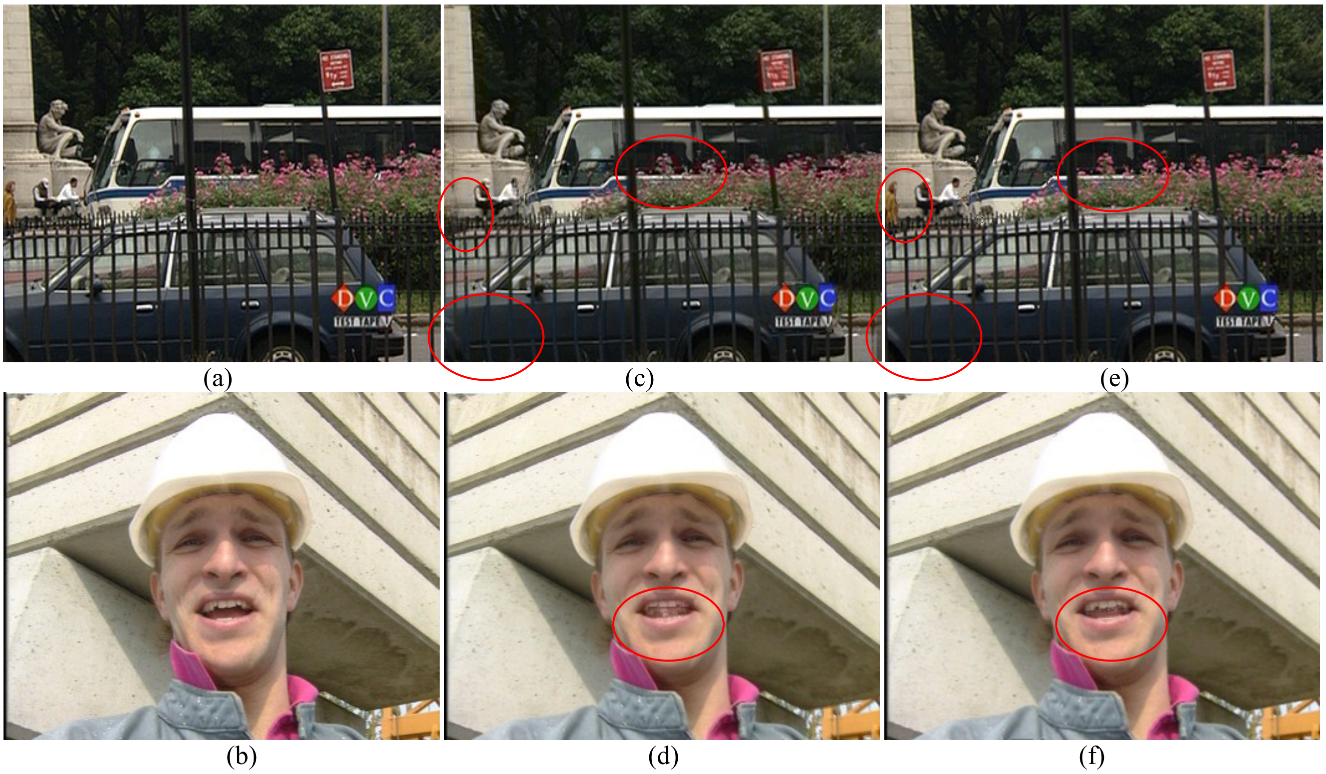

Fig. 5 Results for images *Bus* frame 12 and *Forman* frame 58. (a)-(b)Original images. (c)-(d) Output of [8] (e)-(f)Output of the proposed method.

B. *Comparison of the Subjective Quality*

The interpolated frames by MCFI methods have poor subjective quality mainly due to blocking artifacts and occlusions. The algorithm in [8] and the proposed method do not suffer from the blocking artifacts. So the proposed method is just compared to the algorithm in [8] for subjective quality.

Fig. 5 shows the original images of *Forman* and *Bus* together with the interpolated results of the considered two methods. We can see the proposed method obtains better subjective visual quality for the fast-motion regions and occluded regions within red circles.

IV. Conclusion

In this paper, a new frame interpolation method with pixel-level MVF is presented. The proposed method utilizes bidirectional pixel-level MVFs obtained by optical flow algorithm to alleviate blocking artifacts. Motion post-processing is proposed to keep spatial consistency. At the interpolation stage, a new warping method considering occlusions is proposed to gracefully obtain the interpolated frame. A good performance gain was achieved by the proposed method compared with traditional methods, especially for sequences with fast motions.

References

[1] K. Hilman, H. Park, and Y. Kim, “Using motion-compensated frame-rate conversion for the correction of 3: 2 pulldown artifacts in video sequences,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2000. 10(6): p. 869-877.

[2] B.D. Choi, et al., “Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2007. 17(4): p. 407-416.

[3] T. Ha, S. Lee, and J. Kim, “Motion compensated frame interpolation by new block-based motion estimation algorithm,” *IEEE Transactions on Consumer Electronics*, 2004. 50(2): p. 752-759.

[4] T.H. Tran and C.T. LeDinh, “Frame Rate Converter With Pixel-Based Motion Vectors Selection and Halo Reduction Using Preliminary Interpolation,” *IEEE Journal of Selected Topics in Signal Processing*, 2011. 5(2): p. 252-261.

[5] L. Xu, J. Jia, and Y. Matsushita, “Motion detail preserving optical flow estimation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*,. 2012. p. 1744-1757.

[6] R. Adams and L. Bischof, “Seeded region growing,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1994. 16(6): p. 641-647.

[7] D. Wang, et al., “Motion-compensated frame rate up-conversion Part II: New algorithms for frame interpolation,” *IEEE Transactions on Broadcasting*, 2010. 56(2): p. 142-149.

[8] S. Dikbas and Y. Altunbasak, “Novel True-Motion Estimation Algorithm and Its Application to Motion-Compensated Temporal Frame Interpolation,” *IEEE Transactions on Image Processing*, 2013. 22(8): p. 2931-2945.

[9] S.J. Kang, S. Yoo, and Y.H. Kim, “Dual motion estimation for frame rate up-conversion,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2010. 20(12): p. 1909-1914.

[10] A.M. Huang, and T. Nguyen, “Correlation-based motion vector processing with adaptive interpolation scheme for motion-compensated frame interpolation,” *IEEE Transactions on Image Processing*, 2009. 18(4): p. 740-752.