Deep Learning Approach to Video Frame Rate Up-Conversion Using Bilateral Motion Estimation

Junheum Park*, Chul Lee†, and Chang-Su Kim*
*School of Electrical Engineering, Korea University, Seoul, Korea
†Department of Multimedia Engineering, Dongguk University, Seoul, Korea
E-mail: jhpark@mcl.korea.ac.kr, chullee@dongguk.edu, changsukim@korea.ac.kr

***Abstract*—We propose a deep learning-based frame rate up-conversion algorithm using bilateral motion estimation. We first estimate bilateral motion fields by employing a convolutional neural network. Also, we approximate intermediate bi-directional motion fields, assuming linear motions between successive frames. Finally, we develop the synthesis network to produce an intermediate frame by merging the warped frames, which are obtained using the two kinds of motion fields. Experimental results demonstrate that the proposed algorithm generates high-quality intermediate frames on challenging sequences with large motions and occlusion, and outperforms state-of-the-art conventional algorithms.**

I. Introduction

The frame rate is an important factor affecting the quality of a video, since a low frame rate causes motion aliasing, yields abrupt motion artifacts, and degrades the video quality. To improve the quality of such low frame rate videos, frame rate up-conversion (FRUC) is used to increase temporal resolutions by generating intermediate frames between two frames. FRUC is widely used in practical applications, including visual quality enhancement, video compression, and slow-motion video generation. Due to its practical importance, a lot of researches have been made to develop effective FRUC techniques that provide smooth temporal transitions [1]–[6].

Conventional FRUC algorithms exploit the motion information in videos to interpolate intermediate frames. Specifically, a typical FRUC algorithm estimates pixel-wise correspondences between adjacent frames and then interpolates an intermediate frame based on the correspondences [1]–[3]. Since the accuracy of the correspondence matching has great impacts on the quality of the interpolated frame, most FRUC algorithms have focused on the development of accurate correspondence matching [4]–[6]. In addition, attempts have been made to compensate for inaccurate motion information. For example, in [1], [7], holes in warped frames, which are caused by imperfect correspondence matching, are filled in to yield interpolated frames. However, these conventional algorithms may fail to provide high-quality intermediate frames for videos with large motions, since their interpolation performance heavily depends on the correspondence matching techniques.

Recently, inspired by the success of deep learning in various computer vision and image processing tasks, many CNN-based FRUC techniques have been developed [2], [3], [8]–[10]. Long *et al.* [8] developed a convolutional neural network (CNN) that takes two consecutive frames as input and synthesizes an intermediate frame without requiring explicit motion estimation. Their network attempts to perform motion estimation and frame interpolation simultaneously, but sometimes yields visual artifacts and blurry results. To address this issue, Niklaus *et al.* [3], [9] designed a CNN that outputs pixel-wise convolution kernels instead of an interpolated frame. Then, an intermediate frame is synthesized by convolving the kernels with two input frames. Similarly, Liu *et al.* [2] developed a CNN that estimates a spatiotemporal optical flow field, called voxel flow, between two frames. However, the Niklaus *et al.*'s and Liu *et al.*'s networks [2], [3], [9] may fail to provide high-quality results if input frames have large motions, since they do not exploit contextual information in the frames. Thus, in [10], estimated motion information and contextual information of input frames are fed into a CNN to interpolate an intermediate frame. However, the performance of their algorithm is greatly affected by the accuracy of the motion estimation scheme.

In this work, to address the aforementioned limitations of the conventional CNN-based techniques, we propose a novel FRUC algorithm. First, we develop two complementary motion estimation schemes: *bilateral motion estimation* and *intermediate flow approximation*. Then, we design a synthesis network that fuses the intermediate frames, obtained by employing the two complementary motion information, to generate a final interpolated frame. Experimental results show that the proposed algorithm provides better FRUC results than the existing state-of-the-art algorithms in [9], [11], [12].

The reminder of this paper is organized as follows: Section II describes the proposed FRUC algorithm, and Section III discusses experimental results. Finally, Section IV concludes this paper.

II. Proposed Algorithm

Fig. 1 is an overview of the proposed algorithm that takes two successive frames I_0 and I_1 as input and synthesizes the middle frame $I_{0.5}$ as output. First, we estimate bilateral motions between the input frames. Second, we estimate optical flows between I_0 and I_1 and then approximate intermediate bi-directional flows. Third, pixel-wise context maps C_0 and C_1 are extracted from I_0 and I_1, respectively. Then, the input frames and the corresponding context maps are warped using the bilateral motions and the approximate flows. Finally, the

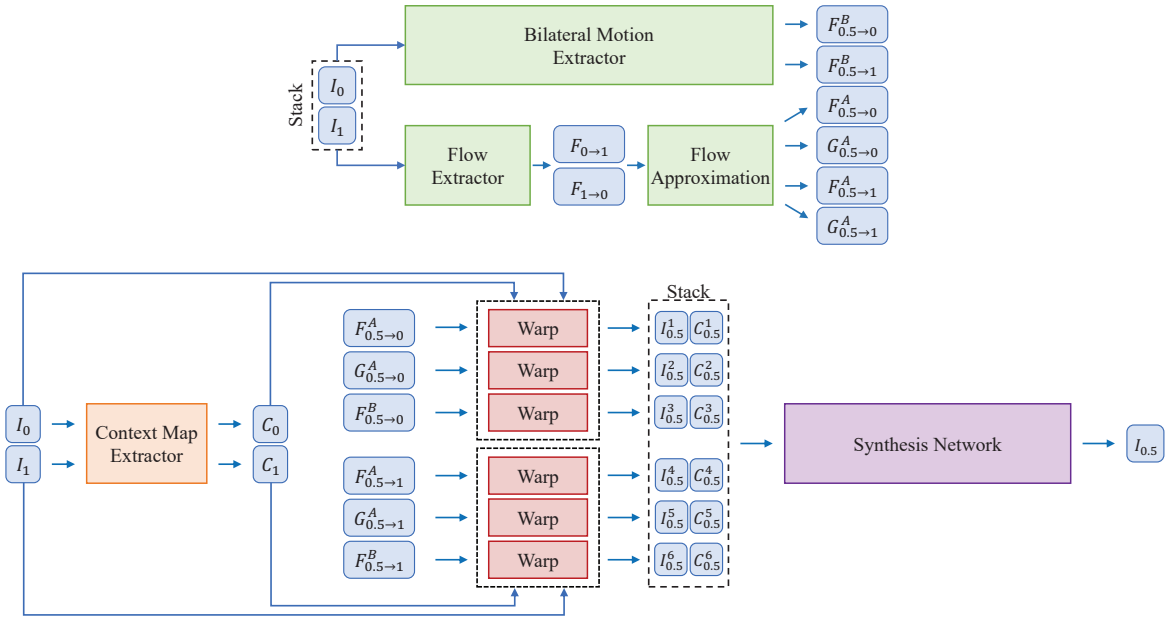


Fig. 1. An overview of the proposed FRUC algorithm. Two motion fields $F^B_{0.5\to0}$ and $F^B_{0.5\to1}$ are estimated by the bilateral motion estimation, and four motion fields $F^A_{0.5\to0}$, $G^A_{0.5\to0}$, $F^A_{0.5\to1}$, and $G^A_{0.5\to1}$ are estimated by the flow approximation. Pixel-wise context maps C_0 and C_1 are extracted from the input frames. Each motion field is used to generate a warped frame and the corresponding context map. A stack of the six pairs of a warped frame and its context map are fed into the synthesis network to yield the intermediate frame $I_{0.5}$.

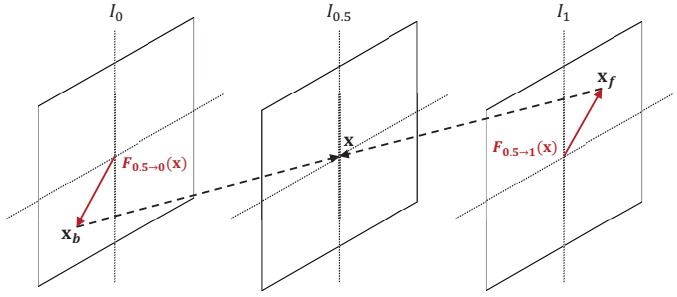


Fig. 2. Bilateral motion estimation: $\mathbf{x}_b$ and $\mathbf{x}_f$ are pixel locations in the previous frame I_0 and the following frame I_1, respectively, which correspond to $\mathbf{x}$ in the intermediate frame $I_{0.5}$.

synthesis network takes the warped frames and their context maps to generate the final intermediate frame $I_{0.5}$.

A. Bilateral Motion Estimation

Given two consecutive frames I_0 and I_1, the objective is to predict the intermediate frame $I_{0.5}$ using motion information. However, because of the lack of the intermediate frame, it is impossible to directly estimate the motion information between the intermediate frame and one of the input frames. To address this issue, we assume linear motion between successive frames. More specifically, let $\mathbf{x}$ denote a pixel location in the intermediate frame $I_{0.5}$, and $F_{0.5\to0}(\mathbf{x})$ and $F_{0.5\to1}(\mathbf{x})$ be the backward and forward motion vectors at $\mathbf{x}$, respectively. Then, based on the linear assumption, we have $F_{0.5\to0}(\mathbf{x}) = -1 \times F_{0.5\to1}(\mathbf{x})$. Fig. 2 illustrates this bilateral motion estimation.

We develop a CNN to estimate bilateral motion fields $F_{0.5\to0}$ and $F_{0.5\to1}$ using the previous and following frames I_0 and I_1. To this end, we adopt PWC-Net [13], a CNN-based optical flow algorithm, as a basis network and modify it accordingly for the bilateral motion estimation. Fig. 3 summarizes key components of the modified PWC-Net. The original PWC-Net regards I_0 and I_1 as reference and target frames, respectively. On the other hand, the bilateral motion estimation regards the intermediate frame $I_{0.5}$ as reference, and I_0 or I_1 as target. Thus, whereas the original PWC-Net warps the feature c^l_1 of I_1 toward the feature c^l_0 of I_0, we warp both features c^l_0 and c^l_1 toward the intermediate frame, yielding $c^l_{0\to0.5}$ and $c^l_{1\to0.5}$, respectively. Moreover, we compute matching costs between $c^l_{0\to0.5}$ and $c^l_{1\to0.5}$ in the cost volume layer. Also, we use $c^l_{0\to0.5}$ and $c^l_{1\to0.5}$ as input to the optical flow estimator.

B. Intermediate Flow Approximation

Although the bilateral motion estimation effectively finds motion fields $F_{0.5\to0}$ and $F_{0.5\to1}$ from the intermediate frame to the previous and following frames, it may fail to estimate accurate motions especially at regions with large motions or occlusion. For example, Figs. 4 (b) and (c) are interpolated regions, reconstructed using bilateral motion fields. Many visual artifacts and blurry effects are observed. To improve the quality of an interpolated frame, in addition to the bilateral motion estimation, we propose an approximation technique to estimate the intermediate bi-directional flows $F_{0.5\to1}$ and $F_{0.5\to0}$ using optical flow fields $F_{0\to1}$ and $F_{1\to0}$ between the two input frames. We use PWC-Net to estimate $F_{0\to1}$ and $F_{1\to0}$, which is shown in Fig. 1 as the flow extractor.

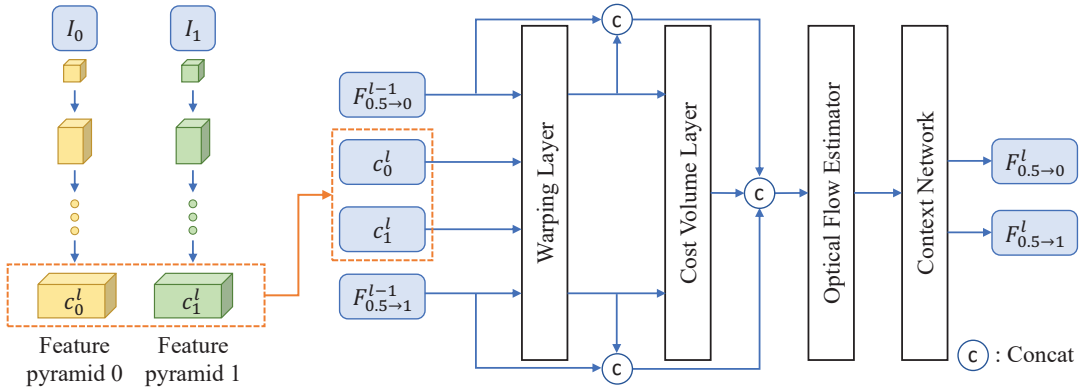


Fig. 3. The architecture of the bilateral motion extractor in Fig. 1, which is based on PWC-Net [13]. The feature maps of the previous and following frames I_0 and I_1 at the lth level, denoted by c_0^l and c_1^l, and the up-sampled motion fields $F_{0.5\to0}^{l-1}$ and $F_{0.0\to1}^{l-1}$ estimated at the $(l-1)$th level are fed into the CNN to generate the motion fields $F_{0.5\to0}^{l}$ and $F_{0.5\to1}^{l}$ at the lth level.

Fig. 4. Comparison of interpolation results obtained by the complementary motion estimation schemes: (a) a ground-truth intermediate frame and (b)∼(e) enlarged parts for the blue and yellow rectangles in (a). The subimages in (b) and (c) are interpolated using the bilateral motion estimation, and (d) and (e) using the intermediate flow approximation. The green rectangles in (b) and (e) contain severe artifacts caused by motion inaccuracies.

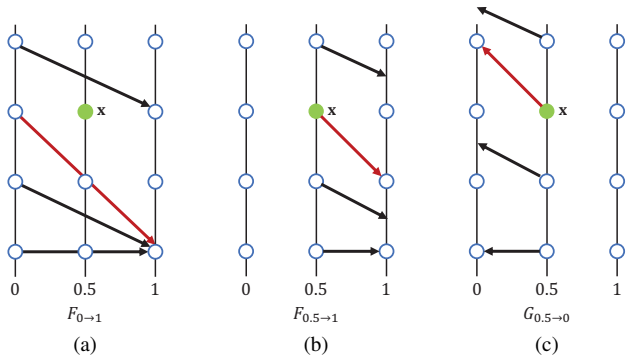


Fig. 5. Illustration of the intermediate flow approximation: $F_{0.5\to1}$ is approximated by halving $F_{0\to1}$, and $G_{0.5\to0}$ is obtained by reversing $F_{0.5\to1}$.

Fig. 5 illustrates the intermediate flow approximation, in which each column represents a frame at a time instance and a dot corresponds to a pixel in the frame. Given an optical flow field $F_{0\to1}$ in Fig. 5(a), we first approximate $F_{0.5\to1}$ in Fig. 5(b). For pixel $\mathbf{x}$ at $t = 0.5$, depicted by a green dot, an intermediate flow $F_{0.5\to1}(\mathbf{x})$ is approximated by halving the flow at the same location at $t = 0$, depicted by the red line, assuming that the optical flow is locally smooth. Since the time interval from $I_{0.5}$ to I_1 is half of that between I_0 and I_1, we halve the flows from I_0 to I_1. Similarly, we approximate $F_{0.5\to0}(\mathbf{x})$ from $F_{1\to0}(\mathbf{x})$. More specifically, we compute the approximate flows as

$$F_{0.5\to1}(\mathbf{x}) = \frac{1}{2} \times F_{0\to1}(\mathbf{x}), \tag{1}$$

$$F_{0.5\to0}(\mathbf{x}) = \frac{1}{2} \times F_{1\to0}(\mathbf{x}). \tag{2}$$

We assume that the motion trajectory between two consecutive frames is linear. Then, the approximate flows in (1) and (2) should be symmetric with respect to $\mathbf{x}$ at $t = 0.5$. Thus, we obtain additional intermediate flows by reversing the directions of the flows in (1) and (2),

$$G_{0.5\to0}(\mathbf{x}) = (-1) \times F_{0.5\to1}(\mathbf{x}), \tag{3}$$

$$G_{0.5\to1}(\mathbf{x}) = (-1) \times F_{0.5\to0}(\mathbf{x}). \tag{4}$$

Fig. 5(c) illustrates an approximate flow $G_{0.5\to0}$ in (3), which is obtained by reversing $F_{0.5\to1}$ in Fig. 5(b).

C. Synthesis Network

If we use only input frames directly to interpolate intermediate frames based on the motion information, rich contextual information in the input frames may be lost during the interpolation [10], [12], [14], degrading the FRUC performance. Therefore, we further exploit contextual information in the input frames, called context maps. Specifically, as done in [10], we extract the output of the conv1 layer of ResNet-18 [15] as a context map. This process is done by the context extractor in Fig. 1.

To synthesize the intermediate frame, we employ the backward warping operation [13], [16]–[18], which warps a target frame I_{target} into a reference frame I_{ref} using a motion vector field by

$$I^{w}_{\text{ref}}(\mathbf{x}) = I_{\text{target}}(\mathbf{x} + F_{\text{ref}\to\text{target}}(\mathbf{x})) \tag{5}$$

where $F_{\text{ref}\to\text{target}}$ is the motion vector field from the reference to the target. Since the estimated motion field contains errors in practice, the warped frame I^{w}_{ref} is not identical to the reference frame, but can be regarded as a candidate for its approximation. By warping two input frames and the corresponding context maps, we obtain six pairs of a warped frame and its context map: two pairs are reconstructed using the bilateral motion estimation, and four pairs using the intermediate flow approximation. Since these six warped pairs have different characteristics, they are used as complementary candidates of the intermediate frame to improve the interpolation performance. Fig. 1 shows these six pairs.

We develop the synthesis neural network that takes the aforementioned six pairs of candidates as input and outputs an interpolated frame $I_{0.5}$, as similarly done in [10]. Specifically, we employ the residual dense network (RDN) [19] to exploit hierarchical features of the candidates. In this work, the upscaling layer in the original RDN is removed, because the synthesis network should preserve the spatial resolution of the input. The number D of residual dense blocks (RDBs), the number C of convolution layers per RDB, and the growth rate G are set to 20, 6, and 32, respectively. Note that these parameters are smaller than the empirically found optimal values in [19]. This is because the warped frames fed into the synthesis network are already motion-compensated and thus the network requires a smaller receptive field.

D. Training

We have two networks in the proposed algorithm: the bilateral motion extractor and the synthesis network. To train the bilateral motion network, we use the Adam optimizer [20] with an input patch size of 384×320 and a mini-batch size of 6 samples. We start with a learning rate $\eta = 0.0001$ and shrink it via $\eta \leftarrow 0.1\eta$ at 100,000 and 150,000 iterations. We compute the warping loss l_w as the L_1 loss between a ground-truth frame and a warped frame using the bilateral motion, given by

$$l_w = \|I_{0.5} - I^{w}_{0\to0.5}\|_1 + \|I_{0.5} - I^{w}_{1\to0.5}\|_1. \tag{6}$$

We also compute the smoothness loss [2] to constrain neighboring pixels to have similar flow vectors, given by

$$l_s = \|\nabla F_{0.5\to0}\|_1 + \|\nabla F_{0.5\to1}\|_1. \tag{7}$$

Then, the bilateral loss is defined as the weighted sum

$$l_b = \lambda_w l_w + \lambda_s l_s \tag{8}$$

where $\lambda_w = 0.4$ and $\lambda_s = 1$. We perform data augmentation to increase the size of training data as in [11]; we use random cropping and random flipping horizontally and vertically.

To train the synthesis network, we use the Adam optimizer with a patch size of 256×256 and a mini-batch size of 8 samples. We fix the learning rate η to 0.0005. We define the reconstruction loss as the L_1 norm between a ground-truth frame $I_{0.5}$ and a synthesized frame $\hat{I}_{0.5}$ as

$$l_r = \|\hat{I}_{0.5} - I_{0.5}\|_1. \tag{9}$$

We also compute the perceptual loss [21] based on the difference of features as

$$l_p = \|\phi(\hat{I}_{0.5}) - \phi(I_{0.5})\|_2 \tag{10}$$

where ϕ is a function to extract the conv4_3 feature of VGG16 [22]. Then, the synthesis loss is defined as the weighted sum

$$l_s = \lambda_r l_r + \lambda_p l_p \tag{11}$$

where $\lambda_r = 0.8$ and $\lambda_p = 0.005$. For data augmentation, we use random cropping. Note that, when we train the synthesis network, the pre-trained bilateral motion network is used after fixing its parameters.

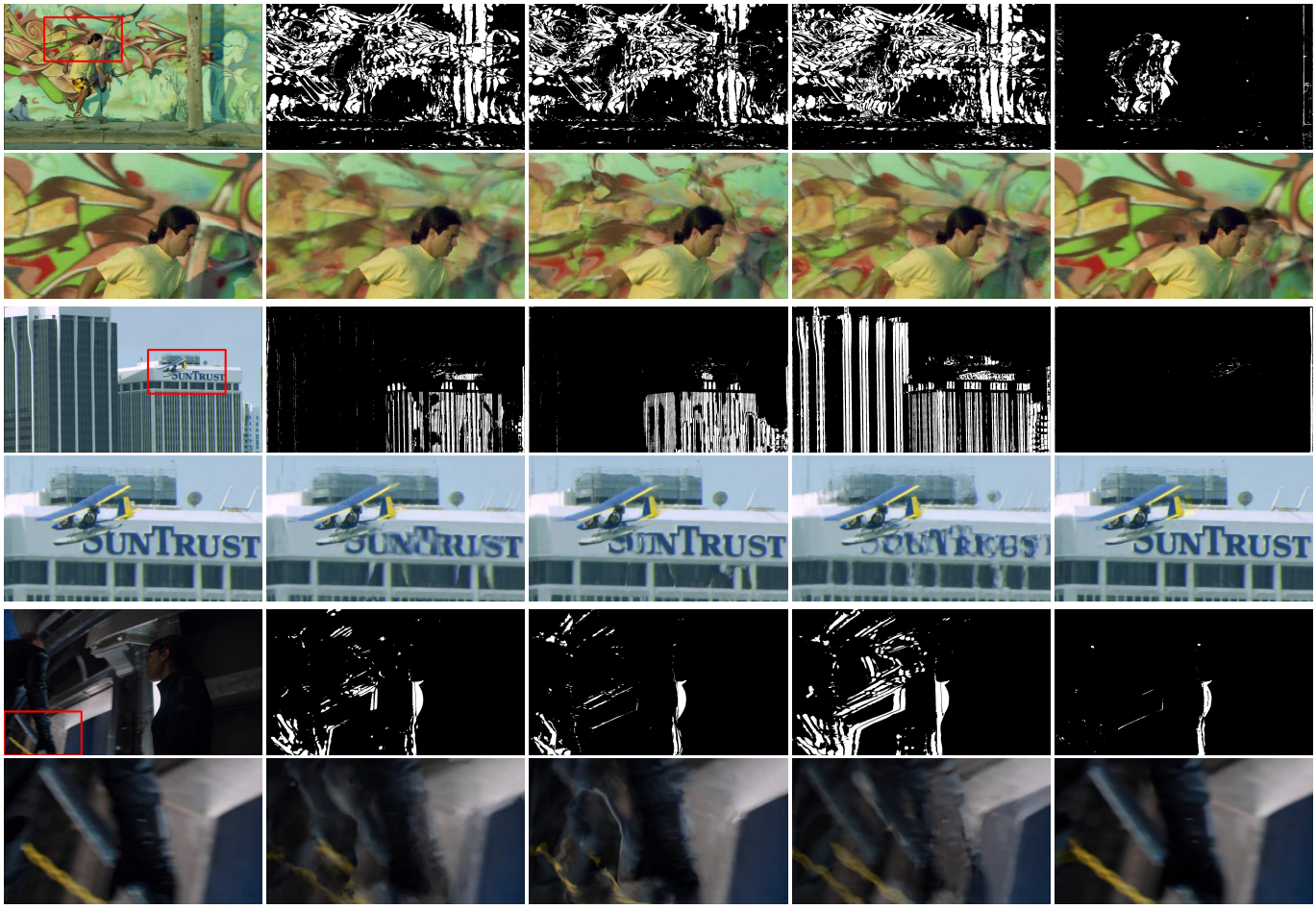
III. Experimental Results

We evaluate the performance of the proposed FRUC algorithm on nine test sequences, which have a spatial resolution of 1920×1080 and a temporal resolution of 24 frames per second (fps). All test sequences are challenging and have large motions. We compare the performance of the proposed algorithm with those of SepConv [9], SuperSlomo [11], and CyclicGen [12]. Odd frames in the test sequences are skipped and then interpolated.

A. Datasets

We train the bilateral motion network using the Adobe240-fps dataset [23]. We compose 13 frames as one sequence sample. Then, among the 13 frames in a sequence sample, three frames with identical frame intervals are randomly chosen, such as (2nd, 4th, 6th) or (3rd, 7th, 11th). The frame intervals are also randomly chosen from 1 to 4. Since only three consecutive frames compose one sample in conventional FRUC datasets, temporal information, which can be obtained from a video sequence (more than 3 frames), cannot be effectively exploited. In contrast, using sequence samples helps the proposed network to learn temporal information among frames in a sequence. We finally obtain 6,000 sequence samples that have a spatial resolution of 640×360 pixels.

To train the synthesis network, we collect triplets, each of which consists of three consecutive frames from 720p high-quality YouTube videos. As in [9], [10], we select samples with large motions and sufficient texture. Specifically, the average value of estimated flow vectors between previous and following frames should be larger than 4 pixels. We finally obtain 135,000 samples.



(a) Ground truth (b) SepConv [9] (c) SuperSlomo [11] (d) CyclicGen [12] (e) Proposed

Fig. 6. Difference maps for interpolated frames (top, third, and fifth rows) or enlarged interpolated regions for the red rectangles (second, fourth, and sixth rows) in the ground-truth in (a) are shown. They are obtained by (b) SepConv [9], (c) SuperSlomo [11], (d) CyclicGen [12], and (e) the proposed algorithm.

B. Comparison with State-of-the-Art Algorithms

We assess the proposed algorithm in comparison with the conventional algorithms. Fig. 6 compares interpolation results. A difference map between ground-truth and interpolated frames is shown to visualize the quality of the interpolation frame. White pixels indicate large differences between corresponding pixels, whereas black pixels mean negligible differences. The conventional algorithms fail to preserve complex texture. For example, the details of the wall are blurred in Figs. 6(b)∼(d). Also, SepConv [9] and CyclicGen [12] in Figs. 6(b) and (d), respectively, cannot faithfully reconstruct the text "SUNTRUST" on the building facade. In Fig. 6(c), SuperSlomo [11] provides better results. However, the proposed algorithm outperforms all these conventional algorithms and preserves the texture information more faithfully. This is because the proposed algorithm synthesizes an intermediate frame by merging complementary candidate frames with context maps, obtained by the bilateral motion estimation and the intermediate flow approximation.

Next, we compare interpolation results objectively. Table I lists the average PSNR results over all interpolated frames in each test sequence. The proposed algorithm is implemented in two ways: using only the intermediate flow approximation (FA) and using both the intermediate flow approximation and the bilateral motion estimation (FA+BM). Even FA outperforms all the conventional algorithms. By adding the bilateral motion estimation, the proposed algorithm further improves the interpolation performance by 0.28 dB. Especially, FA+BM significantly improves the interpolation performance on the sequences #8 and #9, which have large camera motions and complex texture, by 3.01 dB and 4.05 dB, respectively. This confirms that the proposed algorithm can synthesize high-quality intermediate frames effectively using complementary motion information.

IV. Conclusions

We proposed a new FRUC algorithm to handle large motions. We employed two complementary motion estimation schemes: the bilateral motion estimation and the intermediate flow approximation. The bilateral motion estimation is more reliable, while the flow approximation can predict the motion information more accurately in case of large motions. Then, we proposed the synthesis network to fuse the two complementary motion information and generate an intermediate

TABLE I
PSNR COMPARISON OF SEPCONV [9], SUPERSLOMO [11], CYCLICGEN [12], AND THE PROPOSED ALGORITHM. FOR EACH SEQUENCE, THE BEST AND THE SECOND BEST RESULTS ARE BOLDFACED AND UNDERLINED, RESPECTIVELY.

	#1	#2	#3	#4	#5	#6	#7	#8	#9	Avg.
SepConv [9]	22.39	17.66	**34.69**	27.37	28.09	30.33	34.43	32.12	**25.52**	28.78
SuperSlomo [11]	24.14	18.92	32.90	27.91	28.94	30.55	36.06	31.43	24.07	28.85
CyclicGen [12]	17.73	17.36	32.76	24.41	23.74	27.24	28.48	29.31	24.37	25.96
Proposed (FA)	25.57	24.46	28.14	**28.71**	**33.28**	**30.66**	**37.70**	30.17	21.30	29.30
Proposed (FA+BM)	**26.74**	**24.65**	28.98	27.53	32.38	30.48	36.92	**33.18**	25.35	**29.58**

frame. Experiments demonstrated that the proposed algorithm outperforms the state-of-the-art algorithms on challenging sequences with large motions.

ACKNOWLEDGMENT

This work was supported partly by the Cross-Ministry Giga KOREA Project Grant funded by the Korean Government (MSIT) (development of 4D reconstruction and dynamic deformable action model based hyper-realistic service technology) under Grant GK18P0200, partly by the National Research Foundation of Korea Grant funded by the Korean Government (MSIP) under Grant NRF-2018R1A2B3003896 and Grant NRF-2019R1A2C4069806.

REFERENCES

[1] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *Int. J. Comput. Vis.*, vol. 92, no. 1, pp. 1–31, Mar. 2011.

[2] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 4463–4471.

[3] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive convolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 670–679.

[4] E. Herbst, S. Seitz, and S. Baker, "Occlusion reasoning for temporal interpolation using optical flow," University of Washington, Seattle, Tech. Rep. UW-CSE-09-08-01, Aug. 2009.

[5] L. L. Rakêt, L. Roholm, A. Bruhn, and J. Weickert, "Motion compensated frame interpolation with a symmetric optical flow constraint," in *Int. Symp. Vis. Comput.*, Jul. 2012, pp. 447–457.

[6] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View synthesis by appearance flow," in *Proc. European Conf. Comput. Vis.*, Oct. 2016, pp. 268–301.

[7] D. Mahajan, F.-C. Huang, W. Matusik, R. Ramamoorthi, and P. Belhumeur, "Moving gradients: A path-based method for plausible image interpolation," *ACM Trans. Graphics*, vol. 28, no. 3, pp. 42:1–42:11, Aug. 2009.

[8] G. Long, L. Kneip, J. M. Alvarez, H. Li, X. Zhang, and Q. Yu, "Learning image matching by simply watching video," in *Proc. European Conf. Comput. Vis.*, Oct. 2016, pp. 434–450.

[9] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive separable convolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 261–270.

[10] S. Niklaus and F. Liu, "Context-aware synthesis for video frame interpolation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1701–1710.

[11] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz, "Super SloMo: High quality estimation of multiple intermediate frames for video interpolation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9000–9008.

[12] Y.-L. Liu, Y.-T. Liao, Y.-Y. Lin, and Y.-Y. Chuang, "Deep video frame interpolation using cyclic frame generation," in *AAAI Conf. Artificial Intell.*, Jan. 2019.

[13] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8934–8943.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Medical Image Comput. Comput.-Assisted Intervention*, Oct. 2015, pp. 234–241.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[16] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2758–2766.

[17] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4161–4170.

[18] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[19] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2472–2481.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learning Representations*, May 2015.

[21] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. European Conf. Comput. Vis.* Springer, Oct. 2016, pp. 694–711.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learning Representations*, May 2015.

[23] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, "Deep video deblurring for hand-held cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1279–1288.