Forward and Backward Warping for Optical Flow-Based Frame Interpolation

Joi Shimizu[†], Heming Sun[‡*], Jiro Katto[†]
[†]Dept. of Computer Science and Communications Engineering, Waseda University, Tokyo, Japan
[‡]Waseda Research Institute for Science and Engineering, Waseda University, Tokyo, Japan
[*]JST, PRESTO, Kawaguchi, Saitama, Japan
joey.shimizu@toki.waseda.jp, hemingsun@aoni.waseda.jp, katto@waseda.jp

***Abstract*—Frame interpolation methods generate intermediate frames by taking consecutive frames as inputs. This enables the generation of high frame rate videos from low frame rate videos. Recently, many deep learning-based frame interpolation methods have been proposed. One way of frame interpolation is by using the bi-directional optical flow. In many cases, these methods use backward warping to warp the input images to the desired frame. However, forward warping can also be used to warp the input frames. In this paper, we propose a frame interpolation method that utilizes both forward warping and backward warping. Experimental results show that utilizing both warping methods can enhance the performance compared to only using backward warping.**

Keywords—frame interpolation, deep learning, optical flow

I. Introduction

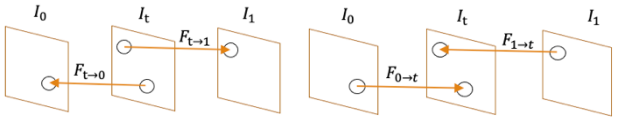
Frame interpolation allows us to generate intermediate frames of consecutive frames. With this technology, high frame rate videos can be generated from lower frame rate videos. For example, slow motion videos can be obtained from ordinary videos without using high speed cameras. Also, frame interpolation is applied in some video compression models for inter prediction. Video compression models with deep learning [1, 2] can replace the block-based flow estimation, that is a process in video compression standards (such as AVC [3], HEVC [4], and VVC [5]), with frame interpolation.

One approach for frame interpolation is by using bi-directional optical flow and using them to warp the input frames to the desired frame. Super SloMo [6] is one of those methods. In Super SloMo, the bi-directional flow is estimated, and the flows are used for backward warping. The challenge in backward warping, however, is that optical flow used for backward warping is not stable, often resulting in inaccurate predictions. We believe that forward warped images can also provide meaningful information for interpolation, as they are able to use more accurate flows for warping. Therefore, we utilize both forward and backward warping in Super SloMo. Experimental results show the effectiveness of using both warping methods.

II. Related Work

A. Frame Interpolation

Many deep learning-based frame interpolation have been proposed. There are several types of approaches. One approach is the flow-based approach [6, 7, 8]. Liu *et al*. [8] proposed a network that learns to synthesize video frames by flowing pixel values from existing ones, which they call deep voxel flow. [6, 7] calculate the bi-directional flow and use backward warping. Backward warping is a popular warping method for flow-based approaches.



(a) Backward Warping (b) forward Warping

Fig. 1. Backward warping and forward warping

However, recently, forward warping-based approach is proposed [9]. In Fig. 1, we explain the differences between the two warping methods. In backward warping, each pixel of the warped frame is mapped from the reference frame, thus creates less occlusion. On the other hand, forward warping maps each pixel of reference frame to the warped frame. Different pixels in the reference frame may be mapped to the same pixel on the warped frame. Also, no pixels may be mapped to a pixel on the warped frame, creating occlusions. Examples of occlusions caused by forward warping can be seen in Fig. 3 (a). For these reasons, backward warping is a popular method for frame interpolation. However, [9] proposed a new way to handle the cases where multiple source pixels are mapped to the same target location.

While methods like [6, 7] require computationally expensive calculations to get the optical flows from the interpolated image to the input images, Huang *et al*. [10] proposed estimating $F_{t\to0}$ and $F_{t\to1}$ directly. Another approach is the kernel-based approach [11]. Combination of flow-based and kernel-based approaches using a network inspired by deformable convolution are also proposed [12]. Meyer *et al*. [13] regards video frames as linear combinations of wavelets and propose a phase-based approach.

B. Optical Flow Estimation

Convolutional Neural Networks (CNNs) are used in many computer vision tasks, including optical flow estimation. Optical flow plays an important role for flow-based interpolation methods. FlowNet [14], an encoder-decoder-based model, was the first work that implemented optical flow estimation with end-to-end training. Later research further enhanced the precision by developing better end-to-end architectures, such as coarse-to-fine flow prediction model using a pyramid architecture [15, 16]. Long *et al*. [17] use a CNN to predict optical flow by synthesizing interpolated frames, and then inverting the

CNN. Another work, proposed a network which extracts per-pixel features of the input frames, create 4D correlation volumes from those features, and iteratively update the flow field [18].

C. Video Compression

Recent trend in video compression research is using deep learning, partially or end-to-end. These methods aim to outperform the widely used video compression standards, such as AVC [3], HEVC [4], and VVC [5]. Frame interpolation models are often used in deep learning-based video compression models. [1] uses a Variational Autoencoder (VAE)-based model for intra prediction and replace the block-based optical flow estimation with frame interpolation. [2] combines the current video compression standards with state-of-the-arts frame interpolation methods, proving that similar performances compared with the compression standards can be achieved with deep learning-based interpolation models.

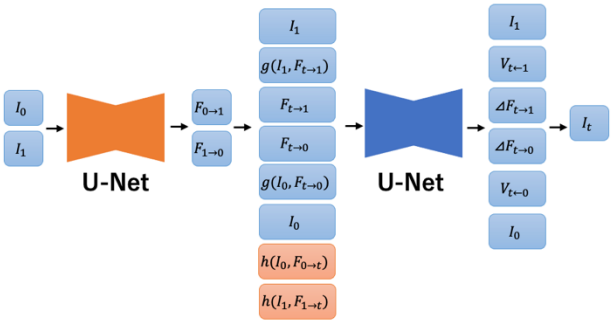


Fig. 2. Proposed approach

III. Proposed Approach

A. Utilization of Forward Warped Images

We incorporate forward warping into Super SloMo. Fig. 2 shows our proposed method. First, the two input frames, I_0 and I_1, are fed into the first U-Net and the bi-directional optical flow, $F_{0\to1}$ and $F_{1\to0}$ is computed. Next, in order to perform backward warping, $F_{t\to0}$ and $F_{t\to1}$ are calculated with the equation below:

$$F_{t\to0} = -(1-t)tF_{0\to1} + t^2F_{1\to0} \quad (1)$$

$$F_{t\to1} = (1-t)^2F_{0\to1} - t(1-t)F_{1\to0} \quad (2)$$

By using I_0 ,I_1, $F_{t\to0}$ and $F_{t\to1}$, backward warped images $g(I_0, F_{t\to0})$ and $g(I_1, F_{t\to1})$ are calculated. The original Super SloMo inputs these features to the second U-Net, but in our model, we also use the forward warped images $h(I_0, F_{0\to t})$ and $h(I_1, F_{1\to t})$ as inputs. For forward warping, we use Softmax Splatting, which was proposed in [9]. Optical flows $F_{0\to t}$ and $F_{1\to t}$, which are needed for forward warping, are calculated with the equation below:

$$F_{0\to t} = t * F_{0\to1} \quad (3)$$

$$F_{1\to t} = (1-t) * F_{1\to0} \quad (4)$$

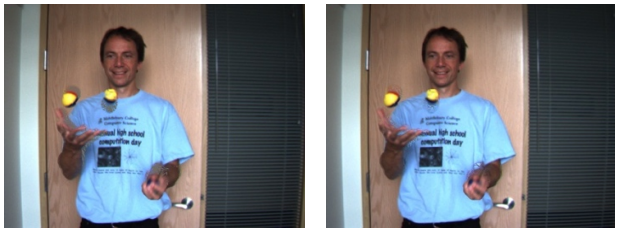
The outputs of the second U-Net are optical flow residuals $\Delta F_{t\to0}$ and $\Delta F_{t\to1}$, and the Visibility Map $V_{t\leftarrow0}$ and $V_{t\leftarrow1}$. The Visibility Map satisfy the following constraint:

$$V_{t\leftarrow0} = 1 - V_{t\leftarrow1} \quad (5)$$

Finally, with the warped frames and the visibility map, the final interpolated frame is calculated.

B. Utilization of RAFT Optical Flows for Forward Warping

Our method explained in section A uses the optical flow from the first U-Net for forward warping. However, the flows $F_{0\to1}$ and $F_{1\to0}$ from the first U-Net are optimized for backward warping, meaning they are not as precise as other optical flow estimation models (since optical flow models are normally optimized for forward warping, rather than backward warping). Therefore, we utilize an off-the-shelf optical flow estimator, RAFT [18], for flow estimations needed for forward warping. This results in a much more precise forward warped image, but bigger occlusions can be spotted, which leads to worse results. To overcome this issue, we fill the occlusions with the already calculated backward warped images. The occlusions on $h(I_0, F_{0\to t})$ are filled with pixel values from $g(I_1, F_{t\to1})$. Similarly, the occlusions on $h(I_1, F_{1\to t})$ are filled with pixel values from $g(I_0, F_{t\to0})$.



(a) Forward warped image with occlusions, especially around balls

(b) Filled with backward warped image pixels

Fig. 3. Forward warped image with RAFT

IV. Experiments

A. Dataset

For training, a combination of different 240 fps videos and datasets were used. First, just like the original Super SloMo reported in [6], the Adobe 240-fps dataset from [19] was used. Also, the GOPRO dataset [20] was used. In addition, we collect 190 sequences from YouTube. Finally, we collect our original 240 fps videos using iPhone. Figs. 4 and 5 show a snapshot of randomly selected video frames of the YouTube and iPhone videos. In the dataset, there are a great variety of scenes such as sports, animals, moving vehicles, etc. Table I shows the number of video clips, the number of frames, and the resolution of each video set.

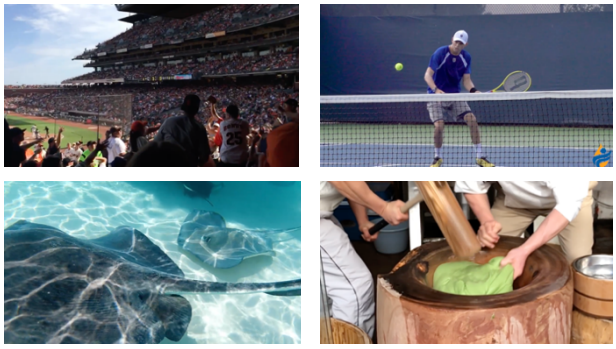


Fig. 4. Snapshot of YouTube videos

Fig. 5. Snapshot of iPhone videos

TABLE I. DATASET INFORMATION

	Adobe240	GOPRO	YouTube	iPhone
# video clips	133	33	190	83
# video frames	124,841	34,874	93,161	117,320
resolution	720p	720p	Various resolutions	1080p

B. Training Settings

Training of the models is conducted by comparing the output $\hat{I}_t$ and the actual intermediate frame I_t. The loss function is a linear combination of four terms:

$$l = \lambda_r l_r + \lambda_p l_p + \lambda_w l_w + \lambda_s l_s \quad (6)$$

Reconstruction loss l_r determines how well the reconstruction of the interpolated frame is. It is calculated with the following equation:

$$l_r = \frac{1}{N}\sum_{i=1}^{N} \| \hat{I}_t - I_t \|_1 \quad (7)$$

Perceptual loss l_p is also used to reduce blur and make interpolated frames sharper. It is calculated by using VGG16 model [21] ϕ:

$$l_p = \frac{1}{N}\sum_{i=1}^{N} \| \phi(\hat{I}_t) - \phi(I_t) \|_2 \quad (8)$$

Warping loss l_w is calculated for optimization of optical flow predictions. The warping loss includes errors of warped frames using $F_{0\to1}$, $F_{1\to0}$, $F_{t\to0}$, and $F_{t\to1}$:

$$l_w = \| I_0 - g(I_1, F_{0\to1}) \|_1 + \| I_1 - g(I_0, F_{1\to0}) \|_1 + \frac{1}{N}\sum_{i=1}^{N} \| I_t - g(I_0, F_{t\to0}) \|_1 + \frac{1}{N}\sum_{i=1}^{N} \| I_t - g(I_1, F_{t\to1}) \|_1 \quad (9)$$

Smoothness loss l_s is also added to encourage neighboring pixels to have similar values. ∇ represents total variation regularization which was also used for training of DVF [8]. Smoothness loss is calculated with the following equation:

$$l_s = \| \nabla F_{0\to1} \|_1 + \| \nabla F_{1\to0} \|_1 \quad (10)$$

The weights are kept the same as [6].

$$\lambda_r = 0.8 \quad (11)$$

$$\lambda_p = 0.005 \quad (12)$$

$$\lambda_w = 0.4 \quad (13)$$

$$\lambda_s = 1 \quad (14)$$

The models are trained for 250 epochs. The learning rate is set to 0.0001 and decreases by a factor of 10 every 100 epochs.

All the videos are divided into groups of 12 consecutive frames. During training, 9 consecutive frames are randomly chosen out of the 12. The first frame and the nineth frame are used as inputs, and the target frame for interpolation is randomly chosen.

C. Evaluation Results

Two datasets, Middlebury and DAVIS, are used for evaluation. The evaluation metric is Peak Signal-to-Noise Ratio (PSNR). For the DAVIS dataset, the 10th frame and the 12th frame were used as inputs to interpolate the 11th frame. Results are shown on Table II. "Ours w/o RAFT" indicates our model which uses the optical flow from the first U-Net for forward warping. "Ours w/ RAFT" indicates our model which uses the optical flow from RAFT for forward warping. The red numbers indicate the best performance and the blue numbers indicate the second best performance.

TABLE II. EVALUATION RESULTS WITH MIDDLEBURY AND DAVIS DATASETS

	Middlebury	DAVIS
Overlapping	27.97	-
Phase-Based [13]	31.12	-
MIND [17]	31.35	-
DVF [8]	34.34	-
Super SloMo	34.24	27.00
Ours w/o RAFT	**34.43**	**27.04**
Ours w/ RAFT	**34.51**	**27.13**

The results for Overlapping, Phase-Based, MIND and DVF are directly taken from [12]. Also, the results of Super SloMo for the Middlebury dataset are almost identical to the results reported in [12]. We can see that by using the optical flow calculated by U-Net for forward warp, the interpolation accuracy enhances for both datasets. This model (Ours w/o RAFT) performed 0.19dB better in Middlebury and 0.04dB better in DAVIS dataset. For the Middlebury dataset, the original Super SloMo performs worse than DVF, but with our new model, it exceeds DVF's performance.

The accuracy becomes even better when better optical flow is used for forward warping. When compared with the original Super SloMo, our final model performed 0.27dB better in Middlebury and 0.13dB better in DAVIS dataset.

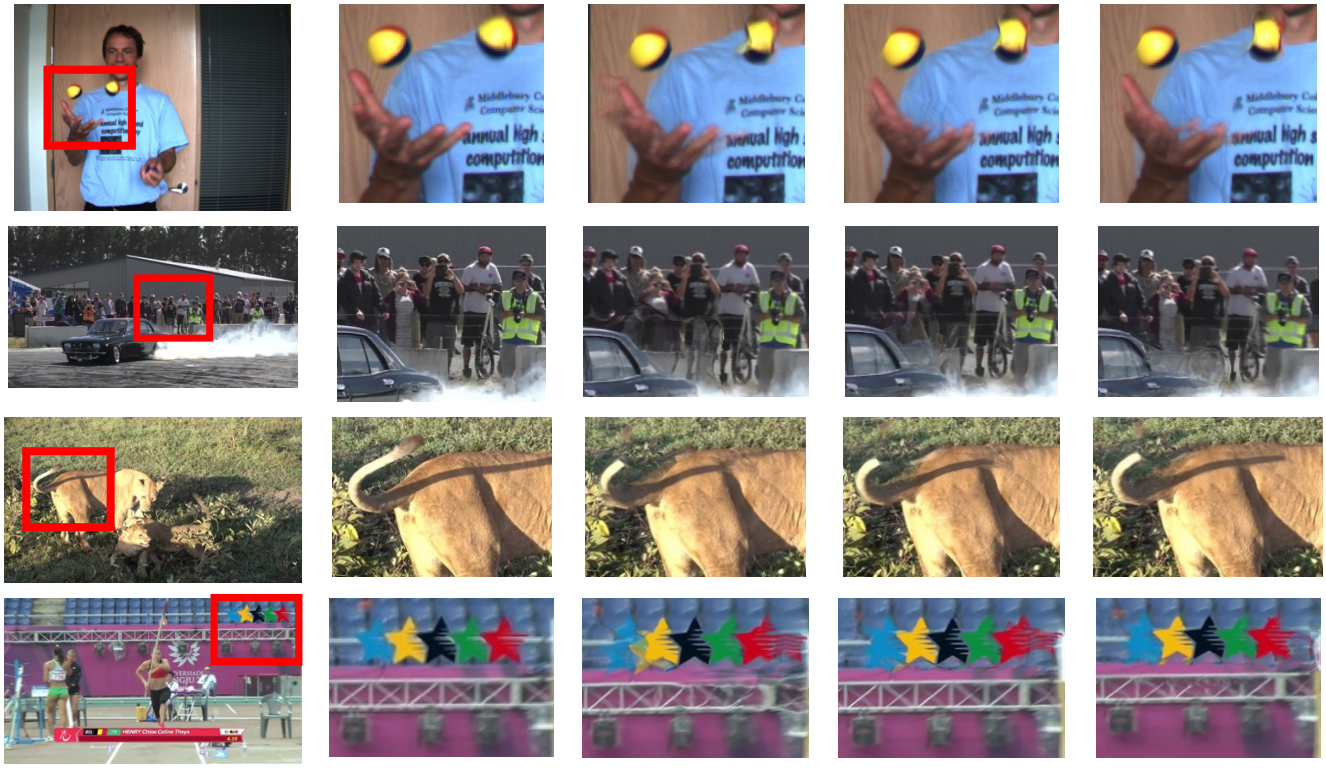


Fig. 6. Visual results for Super SloMo and our proposed approach. The sequences are Beanbags from the Middlebury dataset and burnout, lions, pole-vault from the DAVIS dataset.

Visual results of Super SloMo and our proposed approach are shown on Fig. 6. We can see from Fig. 6 that our proposed approach, especially the one using RAFT, outputs images that are visually closer to the ground truth image. Although, we still see distorted regions in images. One reason for this is the large motions. Large motions are still hard to predict in our model, resulting in blurry or distorted images. Another reason is the non-linear motions. For equations (1), (2), (3) and (4), linear calculations are used to obtain the desired optical flow. Although, in real life, movements are not linear (ex. the movements of the fingers in the Beanbags sequence).

V. Conclusions

We have proposed a frame interpolation method that utilizes both forward warping and backward warping. We have learned that by adding forward warping to a backward warping-based model, Super SloMo, our method can enhance the performance. Also, we found that by using a better optical flow method for forward warping, even greater performance can be achieved. As future work, we would like to conduct similar experiments with other models that only uses one warping method. Also, we would like to use non-linear calculations to better understand the movements in the sequences.

VI. Acknowledgement

This work was supported in part by NICT, Grant Number 03801, Japan and JST, PRESTO Grant Number JPMJPR19M5, Japan. Also, we would like to thank Yuya Ishii, Masaaki Kitamoto, Tatsuhiko Furusawa, Alaric-Yohei Kawai and Ryoichi Sakamoto for helping us create the iPhone 240fps dataset.

References

[1] Z. Cheng, H. Sun, M. Takeuchi, J. Katto, "Learning Image and Video Compression through Spatial-Temporal Energy Compaction", IEEE CVPR, pp. 10063-10072, 2019.

[2] J. Shimizu, Z. Cheng, H. Sun, M. Takeuchi, J. Katto, "HEVC Video Coding with Deep Learning Based Frame Interpolation", IEEE GCCE, pp. 433-434, 2020.

[3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, A. Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no.7, pp. 560-576, 2003.

[4] G. J. Sullivan, J. Ohm, W. Han and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649-1668, 2012.

[5] G. J. Sullivan, J. R. Ohm, "Versatile Video Coding Towards the Next Generation of Video Compression", Picture Coding Symposium, 2018.

[6] H. Jiang, D. Sun, V. Jampani, M. Yang, E. Miller, J. Kautz, "Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation", IEEE CVPR, pp. 9000-9008, 2018.

[7] W. Bao, W. Lai, C. Ma, X. Zhang, Z. Gao, M. Yang, "Depth Aware Video Frame Interpolation", IEEE CVPR, pp. 3698-3707, 2019.

[8] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, A. Agarwala, "Video Frame Synthesis using Deep Voxel Flow", IEEE ICCV, pp. 4473-4481, 2017.

[9] S. Niklaus, F. Liu, "Softmax Splatting for Video Frame Interpolation", IEEE CVPR, pp. 5436-5445, 2020.

[10] Z. Huang T. Zhang, W. Heng, B. Shi, S. Zhou, "RIFE: Real-Time Intermediate Flow Estimation for Video Frame Interpolation", arXiv:2011.06294, 2020.

[11] S. Niklaus, L. Mai, F. Liu, "Video Frame Interpolation via Adaptive Separable Convolution", IEEE ICCV, pp. 261-270, 2017.

[12] H. Lee, T. Kim T. Chung, D. Pak, Y. Ban, S. Lee, "AdaCoF: Adaptive Collaboration of Flows for Video Frame Interpolation", IEEE CVPR, pp. 5315-5324, 2020.

[13] S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. S. Hornung, “Phase-Based Frame Interpolation for Video”, IEEE CVPR, pp. 1410-1418, 2015.

[14] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. Smagt, D. Cremers, T. Brox, “FlowNet: Learning Optical Flow with Convolutional Networks”, IEEE ICCV, pp. 2758-2766, 2015.

[15] A. Ranjan, M.J. Black, “Optical Flow Estimation Using a Spatial Pyramid Network”, IEEE CVPR, pp. 2720-2729, 2017.

[16] D. Sun, X. Yang, M. Liu, J. Kautz, “PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume”, IEEE CVPR, pp.8934-8943, 2018.

[17] Gucan Long, Laurent Kneip, Jose M Alvarez, Hongdong Li, Xiaohu Zhang, and Qifeng Yu. “Learning Image Matching by Simply Watching Video”, Springer ECCV, 2016.

[18] Z. Teed, J. Deng, “RAFT: Recurrent All-Pairs Field Transforms for Optical Flow”, Springer ECCV, 2020.

[19] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, O. Wang. “Deep Video Deblurring for Hand-held Cameras”, IEEE CVPR, pp. 237-246, 2017.

[20] S. Nah, T. H. Kim, K. M. Lee, “Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring”, IEEE CVPR, pp.257-265, 2017.

[21] K. Simonyan, A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”, *CoRR*, abs/1409.1556, 2014.