Motion-blurred Video Interpolation and Extrapolation

Dawit Mureja Argaw, Junsik Kim, Francois Rameau, In So Kweon

Robotics and Computer Vision Laboratory, KAIST

AAAI-21
Feb. 2-9

This work was supported by NAVER Labs

Introduction

Abrupt motion of camera or objects in a scene result in a blurry video

- Contents in the video are degraded by blur
- Temporal gap between consecutive frames is relatively large

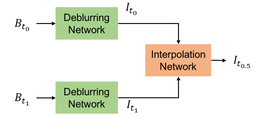
Recovering high quality video from a blurry video

- Temporal upsampling
  - Video Frame Interpolation
- Visual enhancement
  - Deblurring

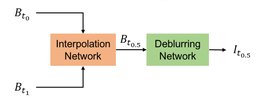
- Interpolated frames are blurry as well
- Temporal gap between frames is still large

Cascaded approaches

- Deblurring + Interpolation



- Interpolation + Deblurring



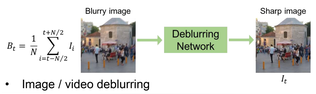
- Not optimal
  - Error propagation
  - Motion smoothness can not be ensured

Goal

Our work proposes a novel and optimal framework for interpolating and extrapolating motion-blurred videos.

Related Works

Deblurring



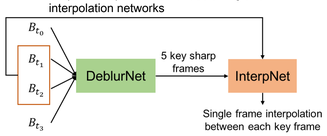
- Image / video deblurring

Video sequence restoration from blur

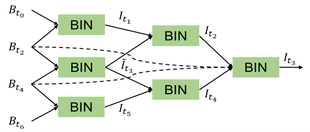
- Temporal ambiguity - can not be extended to multiple frames

Blurry video interpolation

- Jin-Slomo [3]
  - Sequential optimization of deblurring and interpolation networks



- BIN [4]
  - Joint optimization using an inter-pyramid recurrent network

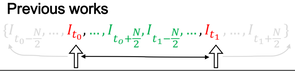


Problem Formulation

Given 2 blurry frames $\{B_{t_0}, B_{t_1}\}$

where $B_{t_0} = \frac{1}{N}\sum_{t_0-N/2}^{t_0+N/2} I_i \ , \ldots , \ B_{t_1} = \frac{1}{N}\sum_{t_1-N/2}^{t_1+N/2} I_i$ for $t_1 \leq t_0 + N$

- **Previous works**



1. Deblur the reference frames
2. Recursively interpolate frames between the reference frames

- **Ours**

- Interpolate and extrapolate all latent frames in a single pass

Contribution

Previous works

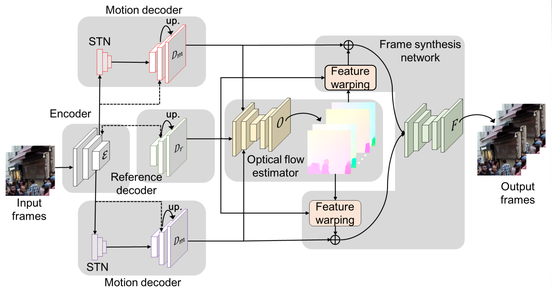
- ✓ Recursive approach for multi-frame interpolation
- ✓ Limited to small blurs
- ✓ Can not be extended for extrapolation task due to temporal ambiguity

Our work

- ✓ Multi-frame interpolation from 2 blurry inputs in a single forward pass
- ✓ Robust to large blurs
- ✓ Joint interpolation and extrapolation tasks by addressing temporal ambiguity

Methodology

Proposed Architecture



Feature encoding and decoding

Feature encoding

- Top-down feature extraction from each blurry input
- Feed-forward CNN network with 6 convolutional blocks

$$\{U_{t_0}^l\}_{l=1}^K = \mathcal{E}(B_{t_0})$$

$$\{U_{t_1}^l\}_{l=1}^K = \mathcal{E}(B_{t_1})$$

Feature decoding

- Reference (middle) features are decoded directly from encoded features
- Bottom-up feature upsampling using deconvolution layers

Feature decoding

- Non-middle features are decoded by learning the global and local motion from encoded features
- Global motion decoding
  - Spatial transformer network (STN) [5]
  - Affine transformation parameter
- Local motion decoding
  - To capture locally-varying motion
  - CNN motion decoder

$$V_{s_0}^l = D_m^l\big(\text{STN}_{s_0}^l\{U_{t_0}^l\} \,||\, \mathbf{up}.\{V_{s_0}^{l+1}\} || \, U_{t_0}^l\big)$$

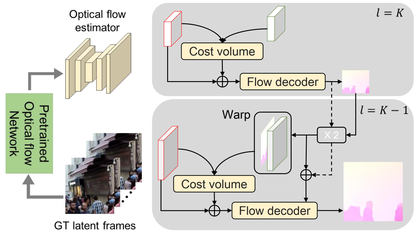
where $s_0 = \left\{t_0 - \frac{N}{2}, \ldots, t_0 - 1, t_0 + 1, \ldots, t_0 + \frac{N}{2}\right\}$

$$V_{s_1}^l = D_m^l\big(\text{STN}_{s_1}^l\{U_{t_1}^l\} \,||\, \mathbf{up}.\{V_{s_1}^{l+1}\} || \, U_{t_1}^l\big)$$

where $s_1 = \left\{t_1 - \frac{N}{2}, \ldots, t_1 - 1, t_1 + 1, \ldots, t_1 + \frac{N}{2}\right\}$

Optical flow estimation

- How do STNs and local motion decoders learn the correct motion to decode features? via optical flow supervision

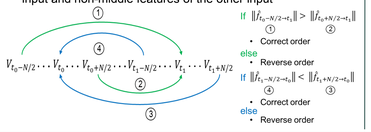


Temporal ordering and Addressing temporal ambiguity

- Optical flow between the reference (middle) feature and non-middle features within each blurry input

$$V_{t_0-N/2} \cdots V_{t_0} \cdots V_{t_0+N/2} \cdots V_{t_1-N/2} \cdots V_{t_1} \cdots V_{t_1+N/2}$$

- Constraining these flows enforces our model to learn motion in a symmetric manner
- Optical flow between the reference (middle) feature of one input and non-middle features of the other input



Frame synthesis

- Decoded features and estimated flows are then used to interpolate and extrapolate frames
- Reference (Middle) frames
  - Directly regressed from decoded reference features in a bottom-up fashion

$$\langle \hat{I}_{t_0}^l, \hat{I}_{t_1}^l \rangle = F\big(\{V_{t_0}^l, V_{t_1}^l\} \,||\, \{\hat{I}_{t_0}^{l+1}, \hat{I}_{t_1}^{l+1}\}\big)$$

- Non-middle frames
  - By back-warping the decoded reference features with the corresponding estimated flows

$$W = \text{warp}\big(V_{t_0}^l, \hat{f}_{s\to t_0}^l\big) \,||\, \text{warp}\big(V_{t_1}^l, \hat{f}_{s\to t_1}^l\big)$$

$$\hat{I}_s^l = F\{W || \boxed{V_s^l} || \, \hat{I}_s^{l+1}\}$$

where $s \in \{t_0 - N/2, \ldots, t_1 + N/2\} \backslash (t_0, t_1)$

Experiments and Results

Loss function

•Multi-scale photometric loss

$$L_{frame} = \sum_{m=1}^{M}\sum_{l=1}^{K} w^l \cdot |I_m^l - \hat{I}_m^l|_1$$

•Multi-scale endpoint error

$$L_{flow} = \sum_{m=1}^{2M-4}\sum_{l=1}^{K} \hat{w}^l \cdot |f_m^l - \hat{f}_m^l|_2$$

•Total loss

$$L_{total} = \alpha_1 L_{frame} + \alpha_2 L_{flow}$$

Dataset

- GOPRO [6]
  - 33 videos at 240 fps
  - Blurred image generated by averaging 7 consecutive frames
- Sony RX V [3]
  - 60 videos at 250 fps
  - Blurred image generated by averaging 9 consecutive frames

Metrics

- Peak signal-to-noise ratio (PSNR)
- Structural similarity index measure (SSIM)

Quantitative analysis

Table 1: Comparison with standard interpolation methods

Method	GoPro PSNR	GoPro SSIM	Sony RX V PSNR	Sony RX V SSIM
SepConv (Niklaus *et al.*)	26.977	0.769	26.181	0.716
SloMo (Jiang *et al.*)	27.240	0.785	26.360	0.728
DAIN (Bao *et al.*)	27.220	0.783	26.410	0.731
Ours	**32.202**	**0.914**	**31.019**	**0.894**

Table 2: Comparison with cascaded approaches

Method	GoPro PSNR	GoPro SSIM	Sony RX V PSNR	Sony RX V SSIM
DVD ⊕ DAIN	25.650	0.722	27.885	0.791
DAIN ⊕ DVD	28.885	0.843	28.157	0.797
DeepDeblur ⊕ DAIN	28.154	0.831	27.192	0.782
DAIN ⊕ DeepDeblur	28.176	0.829	27.195	0.778
SRN ⊕ DAIN	29.966	0.870	29.245	0.828
DAIN ⊕ SRN	30.045	0.867	29.074	0.822
Ours	**32.202**	**0.914**	**31.019**	**0.894**

Table 3: Comparison with previous works

Method	GoPro PSNR	GoPro SSIM	Sony RX V PSNR	Sony RX V SSIM
Jin-Seq (2018)	26.848	0.785	25.785	0.735
Jin-Seq + *flow fix*	29.761	0.877	27.348	0.779
Jin-SloMo (2019)	30.321	0.878	29.267	0.816
Ours	**32.202**	**0.914**	**31.019**	**0.894**

Table 4: Middle frame deblurring

Method	GoPro PSNR	GoPro SSIM	Sony RX V PSNR	Sony RX V SSIM
DVD (Su *et al.*)	26.547	0.742	28.937	0.805
DeepDeblur (Nah *et al.*)	29.671	0.867	27.882	0.788
SRN (Tao *et al.*)	**33.382**	**0.931**	30.827	0.851
Jin-Seq (2018)	31.442	0.906	29.752	0.812
Jin-SloMo (2019)	31.318	0.900	30.325	0.829
Ours	32.994	0.927	**31.650**	**0.904**

- Our approach outperforms related works by a significant margin on motion-blurred video interpolation and gives a competitive performance on video deblurring task

Qualitative analysis of interpolated frames in comparison with related works

Qualitative analysis of extrapolated frames

Ablation studies

Optical flow estimation

- Directly regressing frames without estimating motion
  - Subpar network performance
  - Temporal coherence can't be ensured

Feature decoding

- Local motion decoder can successfully capture both local and global motions
- Explicitly modelling global motion with STN boosted network performance

Table: Ablation studies

STN	$\mathcal{D}_m$	Flow	GoPro PSNR	GoPro SSIM	Sony RX V PSNR	Sony RX V SSIM
✓	✓	✗	29.509	0.836	28.316	0.805
✓	✓	*flow fix*	30.219	0.870	29.163	0.812
✓	✗	✓	28.789	0.855	27.467	0.798
✗	✓	✓	31.317	0.893	30.125	0.857
✓	✓	✓	**32.202**	**0.914**	**31.019**	**0.894**

References

[1]. Wenbo Bao, Wei-Sheng Lai, Chao Ma, Xiaoyun Zhang, Zhiyong Gao, and Ming-Hsuan Yang. Depth-aware video frame interpolation. CVPR2019 (DAIN)
[2]. Tao, X.; Gao, H.; Shen, X.; Wang, J.; and Jia, J. Scale recurrent Network for Deep Image Deblurring, CVPR'18 (SRN)
[3]. Jin, M., Hu, Z., Favaro, P.: Learning to extract flawless slow motion from blurry videos, CVPR 2019 (Jin-Slomo)
[4]. Shen,W.; Bao,W.; Zhai,G.; Chen,L.; Min,X.; and Gao,Z. Blurry Video Frame Interpolation, CVPR 2020 (BIN)
[5]. Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al. Spatial transformer networks, NIPS 2015 (STN)
[6]. Nah, S., Kim, T.H., Lee, K.M.: Deep multi-scale convolutional neural network for dynamic scene deblurring, CVPR 2017 (DeepDeblur)
[7]. Su, S.; Delbracio, M.; Wang, J.; Sapiro, G.; Heidrich, W.; and Wang, O. Deep Video Deblurring for Hand-held Cameras, CVPR 2017 (DVD)
[8]. Jin, M., Meishvili, G., Favaro, P.: Learning to extract a video sequence from a single motion-blurred image, CVPR 2018 (Jin-Seq)
[9]. Jiang, H.; Sun, D.; Jampani, V.; Yang, M.; Learned-Miller, E. G.; and Kautz, J. Super SloMo: High Quality Estimation of Multiple Intermediate Frames for Video Interpolation, CVPR 2018 (SloMo)
[10]. Niklaus, S.; Mai, L.; and Liu, F. Video frame interpolation via adaptive separable convolution, ICCV 2017 (SepConv)
[11]. Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; and Brox, T. 2017. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks, CVPR 2017 (FlowNet 2)
[12]. Sun, D.; Yang, X.; Liu, M.-Y.; and Kautz, J. 2018. PWCNet: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume, CVPR 2018 (PWC-Net)