UNSUPERVISED CONVOLUTIONAL NEURAL NETWORKS FOR ACCURATE VIDEO FRAME INTERPOLATION WITH INTEGRATION OF MOTION COMPONENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Optical flow and video frame interpolation are considered as a chicken-egg problem such that one problem affects the other and vice versa. This paper presents a deep neural network that integrates the flow network into the frame interpolation problem, with end-to-end learning. The proposed approach exploits the relationship between the two problems for quality enhancement of interpolation frames. Unlike recent convolutional neural networks, the proposed approach learns motions from natural video frames without graphical ground truth flows for training. This makes the network learn from extensive data and improve the performance. The motion information from the flow network guides interpolator networks to be trained to synthesize the interpolated frame accurately from motion scenarios. In addition, diverse datasets to cover various challenging cases that previous interpolations usually fail in is used for comparison. In all experimental datasets, the proposed network achieves better performance than state-of-art CNN based interpolations. With Middebury benchmark, compared with the top-ranked algorithm, the proposed network reduces an average interpolation error by about 9.3%. The proposed interpolation is ranked the 1st in Standard Deviation (SD) interpolation error, the 2nd in Average Interpolation Error among over 150 algorithms listed in the Middlebury interpolation benchmark.

1 INTRODUCTION

Video frame interpolation is a classic video processing problem and is important for applications like frame rate up conversion and slow motion playback. Traditional algorithms for frame interpolation generally consist of two steps: motion estimation (ME) and motion-compensated frame interpolation (MCFI). ME estimates the motion trajectories between consecutive frames and MCFI generates interpolated frames by using the motion trajectories. The image quality of an interpolated frame depends on the accuracy of the motion trajectories. In the ME step, most algorithms generate dense motion fields between two consecutive frames using block matching (Bartels & Haan (2010), Wang et al. (2010b), and Nguyen & Lee (2017a)) or optical flow algorithms (Zach et al. (2007), Horn & Schunck (1980)). It is difficult to obtain accurate motion vectors for real-world sequences due to several challenges such as occlusion, reveal, fast and complex motions. In the MCFI step (Nguyen & Lee (2017b) and Wang et al. (2010a)), the interpolation based on linear time-scale assumption often generates overlapped pixels and hole (missing pixels) in the interpolated frame even with accurate motion vectors.

Recently, a deep neural network is applied to frame interpolation with end-to-end learning approach. Starting from the work by Long et al. (2016) which employs an encoder-decoder (or auto-encoder) network, a number of recently-proposed deep networks successfully improves the quality of frame interpolation. The auto-encoder architecture or U-net architecture used in Niklaus et al. (2017b) and Liu et al. (2017) extracts features that are given to the sub-nets for the synthesis of the intermediate frame. SepConv network in Niklaus et al. (2017b) successfully handles blurry artifacts thanks to the independent estimation of four 1D kernels which are then convolved with the input frames to produce the interpolated frame. The SepConv network does not take into consideration of the smoothness among neighboring kernels because the kernels for each pixel are learned independently from those of neighboring pixels. A deep neural network is also used to directly estimate the

phase decomposition of the intermediate frame in Meyer et al. (2018) based on the application of the phase based frame interpolation which is originally proposed by Meyer et al. (2015) to generate intermediate frames by modifying per-pixel phase.

In this paper, different from CNN based previous methods, the proposed method integrates a motion estimation component into the end-to-end learning framework. The proposed network integrates an optical flow estimation network as a driver for a motion-guided interpolator to complement original end-to-end network. The proposed network contains two interpolator sub-networks and a flow sub-network that learns motions in order to drive the interpolator sub-networks to generate accurate intermediate pixels. The interpolator networks and flow network help each other to learn efficiently. The motion branch roles as a feedback or a smoothness constraint for the first end-to-end learning interpolator network. Meanwhile inputs for the motion branch are supplied by the first interpolator network. Therefore, the coefficients of the first interpolator are learned with two loss functions by both branches, the end-to-end learning one and the motion branch. In other words, it learns not only for pixel matching but also for the motion smoothness constraint. The proposed method integrates the flow network that blends the smoothness constraint into the frame interpolation problem, with end-to-end learning. In summary, the main contributions of the paper are

- An end-to-end unsupervised learning with the smoothness constraint for video frame interpolation.
- Highly accurate frame interpolation network that integrates motion information estimated by a flow estimation network as a driver for the motion-guided frame interpolator.
- A standard test dataset for various challenging cases in frame interpolation.

Experimental results show that the proposed network effectively deals with very challenging cases and produces higher visual quality results than state-of-art networks do. For Middlebury frame interpolation benchmark, the proposed network generates the best results among all the published ones, especially for real scenes.

The rest of the paper is organized as follows. Section 2 reviews previous related works. The proposed network is described in Section 3 and experimental results are presented in Section 4. Section 5 concludes this paper.

2 FRAME INTERPOLATION WITH CONVOLUTIONAL NEURAL NETWORKS

A new approach using deep neural network makes promising progress in frame interpolation (Long et al. (2016), Niklaus et al. (2017a), Niklaus et al. (2017b), and Liu et al. (2017)). A convolutional neural network (CNN) based on encoder-decoder architecture or U-net architecture estimates spatially-adaptive convolution kernels for every output pixel and convolve the kernels with the input frames for generating of an intermediate frame. The convolution kernels jointly represent the two de-coupled steps, motion estimation and image synthesis involved in traditional frame interpolation. The U-net architecture becomes a baseline for several CNNs. Long et al. (2016), Niklaus et al. (2017b), and Liu et al. (2017) propose to use the U-net architecture for frame interpolation. Another approach for frame interpolation attempts to use flow networks by taking advantage of improvement of optical flow. Super-Slomo network in Jiang et al. (2018) is proposed to use two U-net networks: the first network to estimate the initial optical flow between two original frames and the second one to refine and convert the motions between two original frames into the motions between the intermediate frames to two original frames. This idea is similar to the mapping method from classical FRUC proposed by Nguyen & Lee (2017b). The difference here is that the conversion is implemented by a deep neural network with U-net architecture, instead of the classical mapping. Therefore, it improves performance at motion boundaries as claimed by Jiang et al. (2018). Xue et al. (2017) propose to use two networks for video frame interpolation: the first network for motion estimation and the second one to generate an interpolated frame. CtxSyn in Niklaus & Liu (2018) is another method that employs a state of art optical flow estimation in Sun et al. (2018) in order to generate highly accurate frame interpolation. Optical flow is estimated from original frames and is later used for spatial warping operations in order to provide inputs for a grid architecture network that synthesizes the interpolated frame from the stacked warped images. This network shows

the importance of both the accurate estimation of the optical flow and efficient synthesis algorithm. SepConv inNiklaus et al. (2017b) uses four sub-nets in addition to a typical encoder-decoder or U-net architecture in order to train horizontal and vertical filters separately. Therefore it reduces blur artifacts that still exists in the interpolated frames generated by the previous typical U-net based methods (Long et al. (2016) and Liu et al. (2017)).

3 CONVOLUTIONAL NEURAL NETWORKS WITH MOTION INTERGRATION FOR FRAME INTERPOLATION

Figure 1 shows the proposed CNN for frame interpolation which composes of two branch networks: the first branch network is an end-to-end learning, corresponding to the first term in the loss function and the second branch network is a motion guided interpolation, corresponding to the second term in the loss function. Both branch networks are connected via Motion derivation layers. Consequently, both branches help each other for efficient training. In addition, in the view point of the first branch, the second branch network roles as a smoothness term that integrates motion components into the end-to-end learning. Meanwhile Interpolator network 1 in the first branch generates the initial flows to be used as the input to Flow net in the second branch. This leads the Flow net starts to learn motions from a quite good initialization values, not from scratch. Thus, more accurate flows are estimated by the Flow net in the second branch. Two consecutive original frames are used as the inputs to Interpolator network 1. From the kernels estimated by Interpolator network 1, the initial motion vectors are derived by Motion derivation layers. The output of Motion derivation layers are used as the input for Flow Network which refines the initial optical flows. Flow network adopts a U-net or auto-encoder architecture with skip layers for efficient learning. With the refined optical flows as the output of the Flow Network, warping operations map the original frames into the target locations in the temporal intermediate frames with blending motion constraints and scenarios into the network. These intermediate frames are used as the inputs for Interpolator network 2 to generate a new intermediate frame in the second branch.

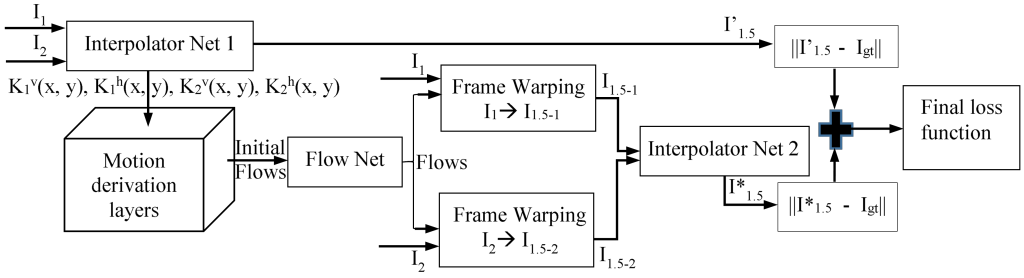


Figure 1: Diagram of the proposed method

Loss function: The proposed network is trained with combination of two loss functions. The first one measures the difference between the ground truth I_{gt} and the interpolated frame $I'_{1.5}$ generated by Interpolator network 1. On the other hand, the second loss function computes the difference between the ground truth I_{gt} and the interpolated frame $I^*_{1.5}$ estimated by Interpolator network 2. For both loss functions, l_1 norm based difference is used as shown in Equation (1)

$$lossfuntion = ||I'_{1.5} - I_{gtr}|| + ||I^*_{1.5} - I_{gtr}|| \tag{1}$$

3.1 NETWORK ARTCHITECTURE

Both Interpolator network and Flow network share the same architecture that is an encoder-decoder network with skip connections, called Core Network. The only difference between two networks is in the final layer. In Interpolator networks like the final layer of SepConv [23], four sub-nets are used to generate four kernel coefficients that implicate the motion information and reduce blurry artifacts caused by direct synthesis with the same architecture. In Flow network, the final layer only uses one sub-net to estimate the flow value.

3.1.1 CORE NETWORK

The core network is a fully-convolutional encoder-decoder architecture which is the main component for both of Interpolator Networks and Flow Network. It consists of five hierarchies of convolution layers, four hierarchies of deconvolution layers and four skip connections. A basic block contains three convolution layers followed respectively by three rectified linear units (ReLU) applied for both an encoder and a decoder. The convolution kernel size is 3x3. For the encoder of the network, a pair of basic blocks and an average-pooling layer with the pooling size and pooling stride equal to 2 is used to down-sample the input features with a down-scale factor of 2. For the decoder, each processing unit consists of the basic block, bilinear up-sampling and a convolution layer. To maintain spatial information, skip connections are added between the corresponding convolution and deconvolution layers at the same spatial resolution. The corresponding deconvolution layers and convolution layers are concatenated together before being fed forward.

3.1.2 INTERPOLATOR NETWORKS

Similar to SepConv network, four sub-nets are followed by the core network to learn two pairs of 1D kernels: horizontal and vertical kernels. For fair comparison with SepConv Network, the size of the 1D kernel is chosen to 51 pixels for each kernel which is the same as that in SepConv Network. The input frames are convolved with kernels to generate the interpolated frame as the following equation.

$$I(x,y) = K_1^v(x,y) * K_1^h(x,y) * P_1(x,y) + K_2^v(x,y) * K_2^h(x,y) * P_2(x,y) \tag{2}$$

where $P_1(x,y)$ and $P_2(x,y)$ are the patches centered at (x, y) in I_1 and I_2. The pixel-dependent kernels $K_1^h(x,y), K_1^v(x,y), K_2^h(x,y)$, and $K_2^h(x,y)$ encode both motion and sampling information from two original patches.

3.1.3 MOTION DERIVATION LAYERS

The coefficients of the above 1D kernels implicate motion information and they are exploited to derive the flow information. The motions are encoded as the offsets of the non-zero kernel values to the kernel center. The motion vector is the weighted sum of the offsets. Therefore, the values of the coefficients and the offsets are used in order to compute the motions. There are four 1D kernels, two corresponding to the displacement of frame 1 to the interpolated frame, and the others corresponding to the displacement of frame 2 to the interpolated frame. Therefore, the optical flows for both the forward and backward directions with point of view from the intermediate frame are computed directly. Unlike the method in Jiang et al. (2018), that estimates the intermediate bi-directional optical flows indirectly by approximating the original uni-directional flows; that produces artifacts and errors around motion boundaries. The formulations of the motion derivation layers are represented by following equations:

$$u_{1.5->1} = \frac{\sum_{i=1}^{51} weight_i^{h_1} * offset_i^{h_1}}{\sum_{i=1}^{51} weight_i^{h_1}} \tag{3}$$

$$v_{1.5->1} = \frac{\sum_{i=1}^{51} weight_i^{v_1} * offset_i^{v_1}}{\sum_{i=1}^{51} weight_i^{v_1}} \tag{4}$$

$$u_{1.5->2} = \frac{\sum_{i=1}^{51} weight_i^{h_2} * offset_i^{h_2}}{\sum_{i=1}^{51} weight_i^{h_2}} \tag{5}$$

$$v_{1.5->2} = \frac{\sum_{i=1}^{51} weight_i^{v_2} * offset_i^{v_2}}{\sum_{i=1}^{51} weight_i^{v_2}} \tag{6}$$

where $u_{1.5->1}$ and $v_{1.5->1}$ are horizontal and vertical components of the flow from the intermediate frame to frame 1, $u_{1.5->2}$ and $v_{1.5->2}$ are horizontal and vertical components of the flow from the intermediate frame to frame 2. $offset_i^{h_1}, offset_i^{v_1}, offset_i^{h_2}, offset_i^{v_2}$ are the displacements of the coefficients $weight_i^{h_1}, weight_i^{v_1}, weight_i^{h_2}, weight_i^{v_2}$ to the center position in the corresponding 1D kernels.

3.1.4 FLOW NETWORK

The input of the flow network is the initial optical flow obtained by the above motion derivation layers. However, those values are estimated separately for each pixel, based on kernel values of each pixel. There is no smoothness constraint between them. Therefore, the initial optical flows are quite noisy and sometimes inaccurate. The flow network works as a refinement process that blends the smoothness constraint into the raw initial optical flow of pixels through a convolution–deconvolution neural network. Similarly to the interpolator networks, the flow network is also an encoder-decoder network that contains two main components: the core network and a final layer that composes up bilinear up-sampling and convolution operations without rectified linear units.

3.1.5 FRAME WARPING

Guided by the estimated optical flow, the proposed method warps the input frames into the intermediate timescale by using both forward and backward warping functions, which can be implemented via bilinear interpolation and are differentiable. Specifically, the proposed method employs forward warping that uses the refined optical flow to warp the input frame I_1 to the target locations in the intermediate frame and obtain a warped frame $I_{1.5-1}$. The proposed method warps the input frame I_2 and generates warped frame $I_{1.5-2}$ in the same way by using backward warping. Two warped frames are the closest frames to the true interpolated frame. Therefore, they are very suitable for the inputs of Interpolator network 2 that works as a frame refinement to generate the intermediate frame. This step narrows down distances between two consecutive input frames and the intermediate one. In addition, it is easier for the network to learn kernels when two inputs are closer. However, this approach may lead to holes or missing pixels in the warped outputs, mostly due to occlusion and reveal (Lu et al. (2018)). In addition, the performance of warping operations depends on the accuracy of the estimated flow, and therefore, errors in optical flow cause a loss of contextual information from original frames. Therefore, in some cases the quality of warped pixels become worse than the original ones.

3.2 TRAINING

Following Niklaus et al. (2017b), the proposed neural network parameters are initialized by a convolution aware initialization (Aghajanyan (2017) and trained by using AdaMax (Kingma & Ba (2014)) with $\beta_1 = 0.9, \beta_2 = 0.999$, a learning rate of 0.001 and a mini-batch size of 16 samples. Training dataset: The proposed network uses the training dataset provided by Xue et al. (2017). A high resolution dataset is chosen because the resolution of training dataset affects the quality of learned parameters. In addition, the size of images in the training dataset is a multiple of 32, that is suitable with proposed network architecture that contains down-sampling layers. For frame rate up conversion, the video sequences that are used for training should be not an up-rated video sequences. In addition, downloaded videos should be not compressed videos due to some compression artifacts. In addition, the histogram of motion in the dataset is diverse and the dataset contains both outdoor and indoor scenes. The video size for training are 448x256, are cropped from the full high resolution videos 1920x1080. This size is chosen because the cropped image has almost the same ratio between the width and height of the original. Furthermore, the height and width of the cropped image are multiples of 32, making them convenient down-sampling and up- sampling operations in the network. In addition, the patches of size 448x256 instead of training with entire full HD frames make it possible to avoid patches that contain no useful information and leads to diverse mini-batches as proposed by Bansal et al. (2017) to improves training efficiency. For data augmentation during training process, the trainer randomly swaps the temporal order between input frames, frame1 becomes frame2 and vice versa. This makes dataset larger and eliminates potential priors.

4 EXPERIMENTAL RESULTS

4.1 FRAME INTERPOLATION EVALUATION

The proposed network for frame interpolation is compared with a representative selection of state-of-the-art methods in both quantitative and qualitative manners with various datasets including a well-known benchmark, as well as a new challenging dataset proposed in this paper to include difficult

Table 1: Evaluation on Middebury benchmark

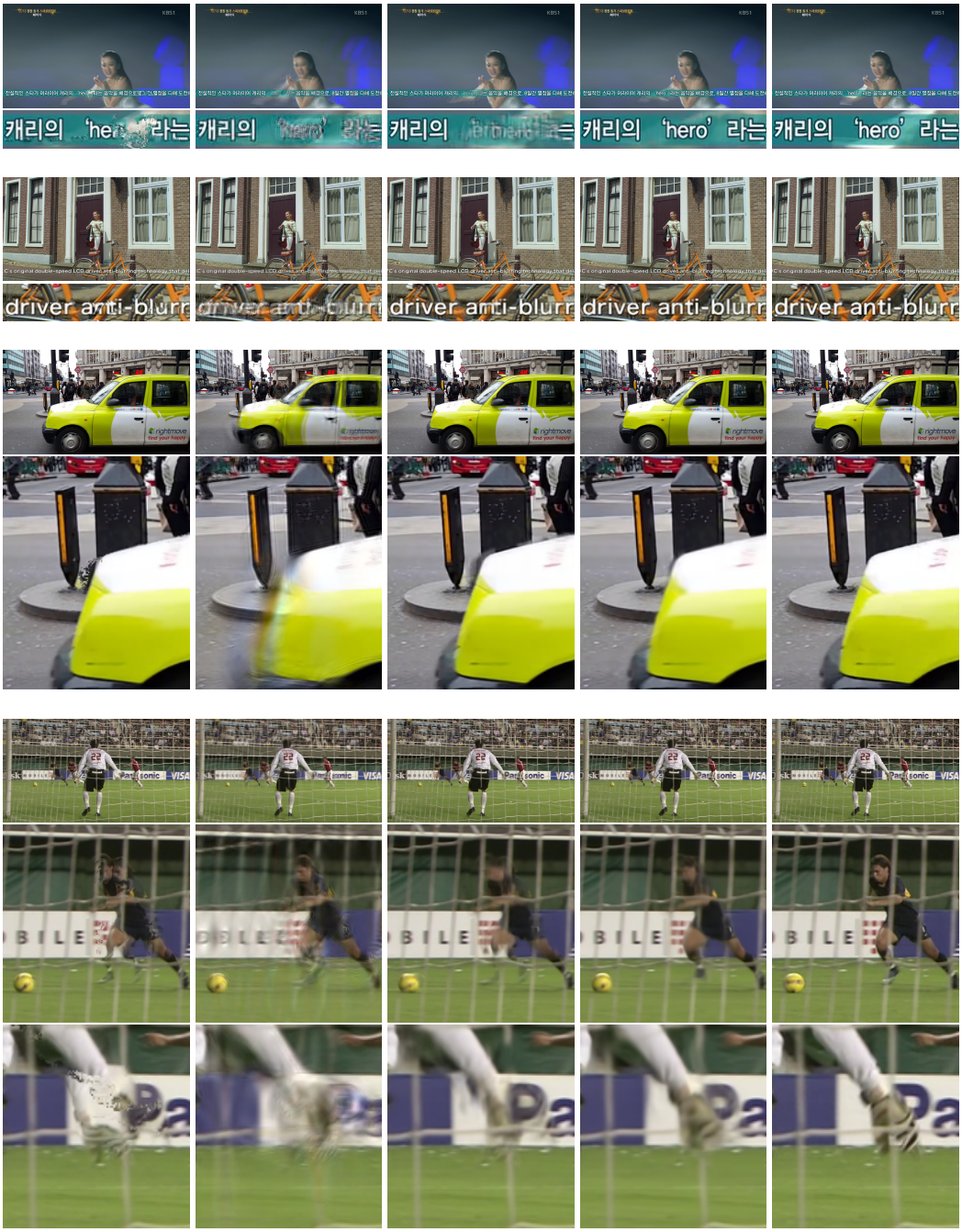
	Average	Mequon	Schef.	Urban	Teddy	Backy.	Basket.	Dump.	Ever.
Proposed	**4.79**	2.66	3.37	3.23	4.87	**7.51**	**4.59**	**5.87**	**6.18**
CtxSyn	5.28	**2.24**	**2.96**	4.32	**4.21**	9.59	5.22	7.02	6.66
MDP-Flow2	5.83	2.89	3.47	3.66	5.20	10.2	6.13	7.36	7.75
SuperSlomo	5.31	2.51	3.66	**2.91**	5.05	9.56	5.37	6.69	6.73
SepConv	5.61	2.52	3.56	4.17	5.41	10.2	5.47	6.88	6.63
DeepFlow	5.97	2.98	3.88	3.62	5.39	11.0	5.91	7.14	7.80

cases for frame interpolation. MDP-Flow2 in Xu et al. (2012) is chosen as a representative of optical flow which performs the second best interpolation with the Middlebury benchmark. To synthesize interpolated frames from the computed optical, the same algorithm as in Baker et al. (2011) is employed. The performance of MDP-Flow2 is followed closely by a neural network based frame interpolation called SepConv in Niklaus et al. (2017b), as well as a representative phased-based method in Meyer et al. (2015) are also compared. Two datasets are used for performance evaluation. The first one is the well-known Middlebury benchmark. The second one is a new dataset proposed in this paper to cover the difficult cases for frame interpolation. These cases include movement of text objects, occlusion, reveal, and movement of small fast objects. Movement of text objects as a subtitle and logos often is difficult for interpolation because the movement often takes place in a background and it is in a different direction from the background. Object occlusion and reveal are difficult in a classical computer vision problem such as optical flow and they are also difficult in frame interpolation. A small object is difficult to estimate its motion and so is fast and complex movement in a video. This new dataset is used to measure the performance of frame interpolation algorithms that focus on enhancement of visual quality. For explanation, this new data set is called *Hard Cases for Display (HCD)* which consists of six high definition video sequences with hard and challenging cases for frame interpolations such as scenes with sub-title, occlusion and reveals, halo artifacts and fast complex motions. Three sub-title sequences, denoted by Sub1, Sub2, and Sub3 are digital broadcasting videos with sub-titles displayed on background regions. Two occlusion and reveal sequences are denoted by Occlusion1 and Occlusion2, respectively. Occlusion1 sequence contains scenes at an inter-section of a crowed street with several layers of objects in which one layer occludes the others because of moving objects. Occlusion2 sequence is captured when a person suddenly enters the scene and later he is occluded by a wall. The last sequence captures a soccer match where the movement of players is fast and complex and the ball is a small object.

Table 1 shows the average interpolation error (AIE) as defined in Baker et al. (2011) where the interpolation error is the root-mean-square (RMS) difference between the ground-truth image and the estimated interpolated image. The proposed network outperforms state-of-art methods (Niklaus & Liu (2018), Xu et al. (2012), Jiang et al. (2018), Niklaus et al. (2017b), and Weinzaepfel et al. (2013)) and improves the best previous method by a significant margin. Especially with *Backyard, Basketball* and *Evergreen* datasets which show real-world scenes, captured with a real camera and containing real sources of noise, the proposed network is consistently the best by notable margins. The proposed interpolation is ranked the 1st in Standard Deviation (SD) interpolation error, the 2nd in Average Interpolation Error among over 150 algorithms listed in the benchmark website.

Table 2 show quantitative comparisons among the proposed methods with representative state of art methods with HCD dataset. In both peak signal-to-noise ratio (PSNR) and structural similarity (SSIM), the proposed method outperforms the representative state of art methods with notable margins. Figure 2 shows the interpolated frames for subjective quality comparison. In the top rows of Figure 2, the text objects in the sub-title include artifacts still exist with the previous methods that are based on optical flow estimation, the phased based method and deep neural network, SepConv [23]. Meanwhile, the proposed method successfully removes these artifacts. In occlusion dataset, as shown in the middle rows in Figure 3, the area surrounding a moving car occludes a background area when the car is moving. There are blurry artifacts in interpolated frames generated by Phase-based method (Meyer et al. (2015) and SepConv network (Niklaus et al. (2017b)) and Salt and Pepper artifact in MDP-Flow2 (Xu et al. (2012) due to pixels in an occlusion area caused by wrong optical flow. Meanwhile, both blurry and Salt and Pepper artifacts are alleviated by the proposed method.

In fast and complex motion dataset as shown in the bottom rows of Figure 3, the movement of the leg of the soccer player and that of the hand of the goalkeeper is fast and complex. The proposed method improves significantly visual quality in comparison with the previous methods.



(a) MDPFlow2 (b) Phase-based (c) SepConv (d) Proposed (e) Groundtruth

Figure 2: Visual Comparison of Interpolated Frames on HCD Dataset

4.2 Performance Analysis

Loss function: The two loss functions are used for training in the proposed frame interpolation neural networks. In order to evaluate the effect of the interpolator network 1 on the second branch,

Table 2: Objective comparisons on HCD dataset

Sequences	MDPFlow2		Phase-based		SepConv		Proposed	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Subtitle 1	33.98	0.987	29.97	0.976	34.26	0.988	**35.00**	**0.989**
Subtitle 2	32.84	**0.992**	26.14	0.969	33.12	**0.992**	**34.23**	**0.992**
Subtitle 3	36.71	0.990	34.64	0.985	**37.18**	**0.991**	37.11	**0.991**
Occlusion 1	30.80	0.963	24.13	0.831	32.60	0.974	**33.24**	**0.975**
Occlusion 2	41.16	0.990	39.47	0.985	42.67	**0.992**	**42.73**	0.991
Halo	29.38	0.960	24.87	0.866	29.76	0.964	**31.25**	**0.971**
Average	34.15	0.9803	29.87	0.9353	34.93	0.9835	**35.59**	**0.9848**

the first term in the loss function is removed. The network is now a straight-forward network that composes of three consequential ones, Interpolator network 1, Flow net and Interpolator network 2. Table 3 shows performance comparisons between the two loss functions and the only second term of the loss function.As reported in the table, the use of two loss functions outperforms that of only the second loss function. Because the first term in loss function helps to estimates kernels correctly, thus derived motion vectors are estimated better than the case without the first term. In other words, the first term roles as a correction for motion derivation layers of the second branch. In addition, with two loss functions, Interpolator network 1 is learned deeper, by both branches of the networks with two combinative targets, pixel matching and smoothness constrains on motion fields. Thus, it is updated by both loss functions and trained with both end-to-end framework and motion based learning.

Flow Net: The second branch of the proposed method includes Flow net that refines an initial flow that is encoded in kernels' coefficients of Interpolator network 1. The performance of Flow net is compared with state of art optical flow (Xu et al. (2012)), the warping operations and Interpolator network 2 are also used to generate the intermediate frame for fair comparison between both optical flow methods. Table 4 reports the performance of both approaches.

Frame Synthesis Network: Warping operations and the second interpolator network compose a new video frame synthesis method from estimated motions. In order to evaluate the contribution of the frame synthesis network, with the same optical flow generated by Xu et al. (2012), the interpolated frame generated by the synthesis network outperforms the results obtained by the benchmark algorithm (Baker et al. (2011)) as reported in Table 5. This shows that the motion compensation or frame interpolation algorithm is able to be improved by neural networks.

5 CONCLUSION

This paper proposes a convolutional neural network with motion integration for video frame interpolation or frame rate up conversion. The proposed method is an end-to-end learning approach. It adopts a motion estimation network as a driver for optical flow-guided Interpolator Networks in order to support end-to-end learning. The motion network branch provides another loss function to make the first Interpolator network learn deeper and more efficient. This alleviates errors caused by learned pixels' wrong kernel coefficients with only original one-time loss function. The interpolated frames by the proposed method are high-quality frames and outperform interpolation results obtained by state-of-the-art methods both quantitatively and qualitatively. The proposed interpolation

Table 3: Effect of loss functions

Sequences	Two loss functions	Only the second term
Subtitle 1	**35.00**	34.22
Subtitle 2	**34.23**	32.51
Subtitle 3	**37.11**	35.87
Occlusion 1	**33.24**	32.57
Occlusion 2	**42.73**	41.17
Halo	**31.25**	30.85
Average	**35.59**	34.53

Table 4: Effect of Flownet

Sequences	MDPFlow2	Flownet
Subtitle 1	**34.30**	34.22
Subtitle 2	31.80	**32.51**
Subtitle 3	**36.65**	35.87
Occlusion 1	31.85	**32.57**
Occlusion 2	**41.48**	41.17
Halo	29.81	**30.85**
Average	34.32	**34.53**

Table 5: Effect of Frame Synthesis Network

Sequences	The benchmark algorithm	Synthesis network
Subtitle 1	33.98	**34.30**
Subtitle 2	**32.84**	31.80
Subtitle 3	**36.71**	36.65
Occlusion 1	30.80	**31.85**
Occlusion 2	41.16	**41.48**
Halo	29.38	**29.81**
Average	34.15	**34.32**

is ranked the 1st in Standard Deviation (SD) interpolation error, the 2nd in Average Interpolation Error among over 150 algorithms listed in the Middlebury interpolation benchmark.

ACKNOWLEDGMENTS

REFERENCES

Armen Aghajanyan. Convolution aware initialization. *CoRR*, abs/1702.06295, 2017. URL `http://arxiv.org/abs/1702.06295`.

S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *Int. J. Computer Vision*, 92:1–31, 2011.

Aayush Bansal, Xinlei Chen, Bryan C. Russell, Abhinav Gupta, and Deva Ramanan. Pixelnet: Representation of the pixels, by the pixels, and for the pixels. *CoRR*, abs/1702.06506, 2017. URL `http://arxiv.org/abs/1702.06506`.

C. Bartels and De Haan. Smoothness constraints in recursive search motion estimation for picture rate conversion. *IEEE Trans. Circuits Syst. Video Technol.*, 20:1310–1319, 2010.

Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. Technical report, 1980.

Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

Ziwei Liu, Raymond Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *IEEE International Conference on Computer Vision*, October 2017.

G. Long, L. Kneip, J. M. Alvarez, H. Li, X. Zhang, and Q. Yu. Learning image matching by simply watching video. In *European Conference on Computer Vision*, 2016.

Yao Lu, Jack Valmadre, Heng Wang, Juho Kannala, Mehrtash Harandi, and Philip H. S. Torr. Devon: Deformable volume network for learning optical flow. *CoRR*, abs/1802.07351, 2018. URL http://arxiv.org/abs/1802.07351.

S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. Sorkine-Hornung. Phase-based frame interpolation for video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

S. Meyer, A. Djelouah, B. McWilliams, A. S. Hornung, M. Gross, and C. Schroers. Phasenet for video frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Van Thang Nguyen and H. J. Lee. A semi-global motion estimation of a repetition pattern region for frame interpolation. In *IEEE International Conference on Image Processing*, 2017a.

Van Thang Nguyen and H. J. Lee. An efficient non-selective adaptive motion compensated frame rate up conversion. In *IEEE Symposium on Circuits and Systems*, 2017b.

Simon Niklaus and Feng Liu. Context-aware synthesis for video frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017a.

Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *IEEE International Conference on Computer Vision*, 2017b.

Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

D. Wang, A. Vincent, P. Blanchfield, and R. Klepko. Motion-compensated frame rate up-conversion—part ii: New algorithms for frame interpolation. *IEEE Trans. Broadcasting.*, 56: 142–149, 2010a.

D. Wang, L. Zhang, and A. Vincent. Motion-compensated frame rate up-conversion—part i: Fast multi-frame motion estimation. *IEEE Trans. Broadcasting.*, 56:133–141, 2010b.

P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: large displacement optical flow with deep matching. In *IEEE International Conference on Computer Vision*, December 2013.

L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *IEEE Trans. Pattern Analys. Machine Intel.*, 34:1744–1757, 2012.

Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T. Freeman. Video enhancement with task-oriented flow. *CoRR*, abs/1711.09078, 2017. URL http://arxiv.org/abs/1711.09078.

C. Zach, Pock T., and Bischof H. A duality based approach for realtime tv-l1 optical flow. In *Pattern Recognition*. Springer, 2007.