2019

Context-Aware Synthesis for Video Frame Interpolation

Simon Niklaus
Portland State University, sniklaus@pdx.edu

Feng Liu
Portland State University, fliu@pdx.edu

Context-aware Synthesis for Video Frame Interpolation

Simon Niklaus
Portland State University
sniklaus@pdx.edu

Feng Liu
Portland State University
fliu@cs.pdx.edu

Abstract

Video frame interpolation algorithms typically estimate optical flow or its variations and then use it to guide the synthesis of an intermediate frame between two consecutive original frames. To handle challenges like occlusion, bidirectional flow between the two input frames is often estimated and used to warp and blend the input frames. However, how to effectively blend the two warped frames still remains a challenging problem. This paper presents a context-aware synthesis approach that warps not only the input frames but also their pixel-wise contextual information and uses them to interpolate a high-quality intermediate frame. Specifically, we first use a pre-trained neural network to extract per-pixel contextual information for input frames. We then employ a state-of-the-art optical flow algorithm to estimate bidirectional flow between them and pre-warp both input frames and their context maps. Finally, unlike common approaches that blend the pre-warped frames, our method feeds them and their context maps to a video frame synthesis neural network to produce the interpolated frame in a context-aware fashion. Our neural network is fully convolutional and is trained end to end. Our experiments show that our method can handle challenging scenarios such as occlusion and large motion and outperforms representative state-of-the-art approaches.

1. Introduction

Video frame interpolation is one of the basic video processing techniques. It is used to generate intermediate frames between any two consecutive original frames. Video frame interpolation algorithms typically estimate optical flow or its variations and use them to warp and blend original frames to produce interpolation results [1, 24, 33].

The quality of frame interpolation results depends heavily on that of optical flow. While these years have observed great progress in research on optical flow, challenges such as occlusion and large motion still remain. As reported in recent work, the optical flow accuracy decreases as the motion increases [4, 17]. Therefore, many frame interpolation methods estimate bidirectional optical flow between two input frames and use them to handle inaccuracies of motion estimation and occlusion [15, 37, 52]. Some of the recent bidirectional approaches, such as [52], also estimate weight maps to adaptively blend the optical flow-guided warped frames. However, as shown in Figure 1, blending the warped frames sometimes is limited in handling occlusion and inaccuracies of optical flow, as it requires accurate pixel-wise correspondence between the warped frames.

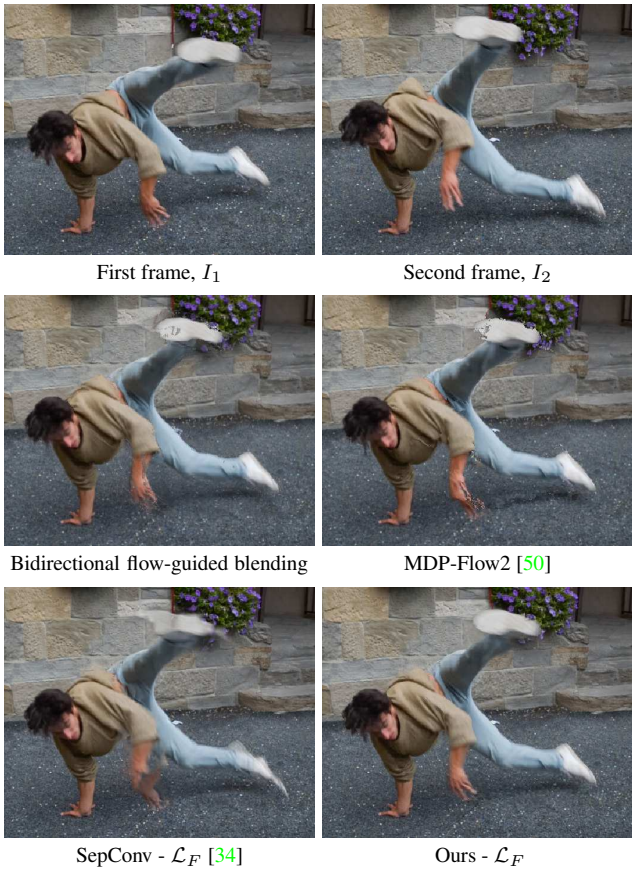


Figure 1: A challenging example. Compared to existing frame interpolation methods, our approach can better handle large motion, significant occlusion, and motion blur.

http://graphics.cs.pdx.edu/project/ctxsyn

This paper presents a context-aware frame synthesis approach to high-quality video frame interpolation. Instead of blending pre-warped input frames, our approach adopts a more flexible synthesis method that can better handle inaccuracies of optical flow and occlusion. Specifically, we develop a frame synthesis deep neural network that directly produces an intermediate frame from the pre-warped frames, without being limited to pixel-wise blending. To further improve the interpolation quality, our method employs a pre-trained image classification neural network [14] to extract per-pixel context information from input frames. These context maps are pre-warped together with the input frames guided by bidirectional optical flow. The frame synthesis neural network takes the pre-warped frames and their context maps as input and is able to handle challenging scenarios such as significant occlusion and motion to produce high-quality interpolation results.

Our experiments show that our method is able to handle difficult frame interpolation cases and produces higher quality results than state-of-art approaches [24, 31, 33, 50]. On the Middlebury interpolation benchmark, our method generates the best results among all the published ones [1]. We attribute the capability of our method to produce high-quality interpolation results to the following factors. First, our frame synthesis neural network is not limited to pixel-wise blending. It is able to make use of neighboring pixels for frame synthesis, which is important to handle occlusion and errors in the optical flow. Second, the extracted and pre-warped context maps provide information in addition to motion and enable our neural network to perform informative frame synthesis. Finally, we would like to acknowledge that our method adopts PWC-Net [43], a state-of-the-art optical flow algorithm to estimate the bidirectional flow, which provides a good initialization. Our method is able to leverage the continuing success of optical flow research to further improve our result in the future.

2. Related Work

Video frame interpolation is a classic computer vision problem. While it is a constrained problem of novel view interpolation [6, 19, 44], a variety of dedicated algorithms have been developed for video frame interpolation, which will be the focus of this section.

Classic video frame interpolation algorithms contain two steps: optical flow estimation and frame interpolation [1, 48, 51]. The quality of frame interpolation largely depends on that of optical flow, which is one of the basic problems in computer vision and has been attracting a significant amount of research effort. These years, the quality of optical flow is consistently improving [1, 4, 17, 30]. Similar to many other computer vision problems, deep learning approaches are used in various high-quality optical flow algorithms [9, 11, 43, 47, 49]. However, optical flow is an inherently difficult problem and challenges still remain in scenarios, such as significant occlusion, large motion, lack of texture, and blur. Optical flow results are often error-prone when facing these difficult cases.

Various approaches have been developed to handle the inaccuracies and missing information from optical flow results. For example, Baker *et al*. fill the holes in the optical flow results before using them for interpolation. Another category of approaches estimate bidirectional optical flow between two input frames and use them to improve the accuracy and infer missing motion information due to occlusion [15, 37, 52]. Some recent methods also estimate per-pixel weight maps to better blend the flow-guided pre-warped frames than using global blending coefficients [52]. These methods have been shown effective; however, their performance is sometimes limited by the subsequent interpolation step that blends the pre-warped frames to produce the final result. This paper builds upon these bidirectional flow approaches and extends them by developing a deep frame synthesis neural network that is not limited to pixel-wise blending and thus is more flexible to tolerate errors in optical flow. Moreover, our method extracts and warps pixel-wise contextual maps together with input frames and feeds them to the deep neural network to enable context-aware frame synthesis.

Several recent methods interpolate video frames without estimating optical flow. Meyer *et al*. developed a phase based frame interpolation approach that generates intermediate frames through per-pixel phase modification [31]. Long *et al*. directly train a deep convolutional neural network that takes two consecutive original frames as input and outputs an intermediate frame without an intermediate motion estimation step. Their results, however, are sometimes blurry [25]. Niklaus *et al*. merge motion estimation and pixel synthesis into a single step of local convolution. They employ a deep convolutional neural network to estimate a pair of convolution kernels for each output pixel and then use them to convolve with input frames to produce an intermediate frame [33, 34]. While their methods can handle occlusion and reasonable size motion, they still cannot handle large motion. Liu *et al*. developed a deep neural network to extract voxel flow that is used to sample the space-time video volume to generate the interpolation result [24]. As their method samples the 2^3 spatial-temporal neighborhood according to the voxel flow, it is still limited in accommodating inaccuracies in motion/voxel flow estimation.

3. Video Frame Interpolation

Given two consecutive video frames I_1 and I_2, our goal is to generate an intermediate frame $\hat{I}_t$ at the temporal location t in between the two input frames. Our method works in three stages as illustrated in Figure 2. We first estimate bidirectional optical flow between I_1 and I_2 and extract pixel-

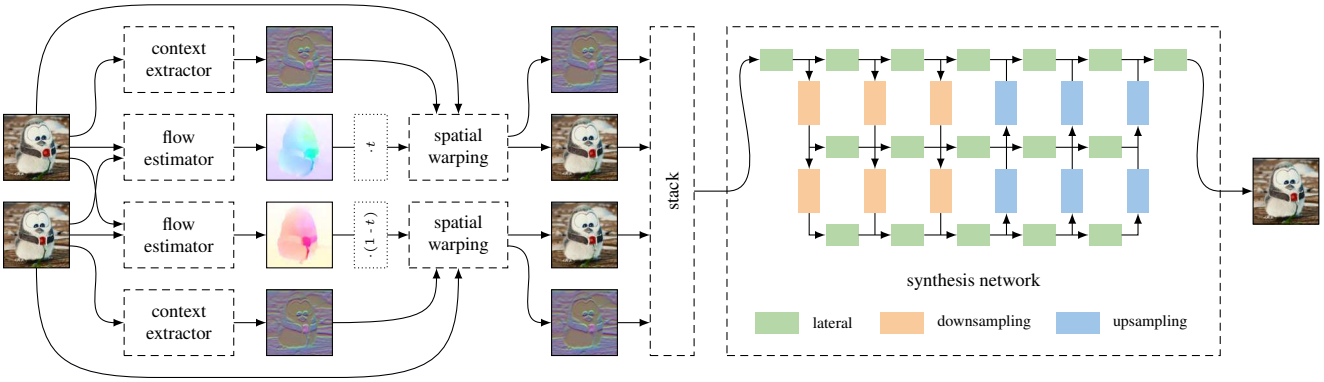


Figure 2: Algorithm overview. Given two consecutive input frames, our method first estimates bidirectional flow between them and extracts per-pixel context maps. Our method then pre-warps the input frames and their corresponding context maps. Finally, the warped frames and context maps are fed into a frame synthesis neural network to generate the interpolation result at a desired temporal position $t \in [0, 1]$. The synthesis network utilizes a modified GridNet architecture consisting of three rows and six columns. It employs 3×3 convolutions with per-row channel-sizes of 32, 64, and 96 respectively.

wise context maps. We then warp the input frames together with their context maps according to the optical flow. We finally feed them into a deep frame synthesis neural network to generate the interpolation result.

We estimate the bidirectional optical flow $F_{1\to2}$ and $F_{2\to1}$ between the two frames using the recent PWC-Net method [43]. This method utilizes a multi-scale feature pyramid in combination with warping and cost volumes. It performs well in standard benchmarks while at the same time being computationally efficient.

Guided by the bidirectional flow, we pre-warp the input frames. Specifically, we employ forward warping that uses optical flow $F_{1\to2}$ to warp input frame I_1 to the target location and obtain a pre-warped frame $\hat{I}_t^1$. During forward warping, we measure the flow quality by checking the brightness constancy and discard contributions from flow vectors that significantly violate this constraint. We warp input frame I_2 and generate the pre-warped frame $\hat{I}_t^2$ in the same way. Note that this approach can lead to holes in the warped output, mostly due to occlusion. We chose forward warping to allow the synthesis network to identify occluded regions and to avoid confusing the synthesis network with heuristic measures for filling in missing content.

3.1. Context-aware Frame Synthesis

Given two pre-warped frames $\hat{I}_t^1$ and $\hat{I}_t^2$, existing bidirectional methods combine them through weighted blending to obtain the interpolation result $\hat{I}_t$. This pixel-wise blending approach requires pixel-wise accuracy of optical flow. We develop a more flexible approach by training a synthesis neural network that takes the two pre-warped images as input and directly generates the final output, without resorting to pixel-wise blending. In this way, our method can better tolerate inaccuracies of optical flow estimation.

Generating the final interpolation result solely from the two pre-warped frames has a limitation in that rich contextual information from the original frames is lost. Note that the contextual information for each pixel in pre-warped frames could be compromised due to errors in optical flow. We therefore extract per-pixel context maps from the original input frames and warp them together with the input frames, before feeding them into the synthesis network as shown in Figure 2. We chose to extract the contextual information by utilizing the response of the `conv1` layer from ResNet-18 [14]. Each pixel in the input frame accordingly has a contextual vector that describes its 7×7 neighborhood. Note that we modify the stride of the `conv1` layer to be 1 instead of 2 such that the context map has the same size as its corresponding input frame.

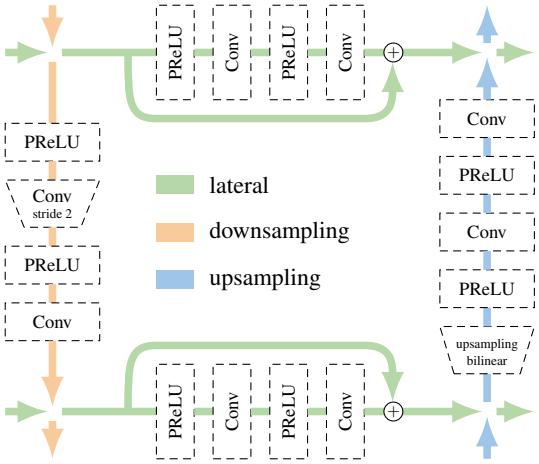


Figure 3: Building block of our frame synthesis neural network in Figure 2, adapted from the GridNet architecture.

We extend the GridNet architecture to generate the final interpolation result from the two pre-warped frames and their context maps [10]. Instead of having a single se-

quence of consecutive layers like typical neural networks, a GridNet processes features in a grid of rows and columns as shown in Figure 2. The layers in each row form a stream in which the feature resolution is kept constant. Each stream processes information at a different scale and the columns connect the streams to exchange information by using downsampling and upsampling layers. This generalizes typical encoder-decoder architectures in which features are processed along a single path [26, 36]. In comparison, a GridNet learns how information at different scales should be combined on its own, making it well-suited for pixel-wise problems where global low-resolution information guides local high-resolution predictions. We modified the horizontal and vertical connections of the GridNet architecture as shown in Figure 3. Specifically, we follow recent findings in image enhancement tasks and do not use Batch Normalization [16, 32, 23]. Furthermore, we incorporate parametric rectified linear units for improved training and use bilinear upsampling to avoid checkerboard artifacts [13, 35]. Note that our configuration of three streams, and thus three scales, leads to a relatively small receptive field of the network [27]. However, it fits our problem well since pre-warping already compensates for motion.

Loss functions. We consider various loss functions that measure the difference between the interpolated frame $\hat{I}$ and its ground truth I_{gt}. For the color-based loss function, we, in accordance with reports that ℓ_2 leads to blurry results [12, 25, 28, 39, 42], employ a ℓ_1-based loss function as follows.

$$\mathcal{L}_1 = \left\| \hat{I} - I_{gt} \right\|_1 \tag{1}$$

We also consider a feature-based loss that measures perceptual difference [5, 8, 18, 22, 40, 53]. Specifically, we follow Niklaus *et al*. [34] and utilize the response of the `relu4_4` layer from VGG-19 [41] to extract features ϕ and measure their difference as follows.

$$\mathcal{L}_F = \left\| \phi(\hat{I}) - \phi(I_{gt}) \right\|_2^2 \tag{2}$$

Another alternative that we adopt measures the difference between Laplacian pyramids [3]. This multi-scale loss separates local and global features, thus potentially providing a better loss function for synthesis tasks. By denoting the i-th layer of a Laplacian pyramid representation of an image I as $L^i(I)$, we define the loss as follows.

$$\mathcal{L}_{Lap} = \sum_{i=1}^{5} 2^{i-1} \left\| L^i(\hat{I}) - L^i(I_{gt}) \right\|_1 \tag{3}$$

This loss function takes the difference between two Laplacian pyramid representations with five layers. Additionally, we scale the contributions from the deeper levels in order to partially account for their reduced spatial support. We will discuss how these different loss functions affect the interpolation results in Section 4. Briefly, the feature loss tends to produce visually more pleasing results while the Laplacian and the ℓ_1-based loss produce better quantitative results.

3.2. Training

We train our frame synthesis neural network on examples of size 256×256 using AdaMax with $\beta_1 = 0.9$, $\beta_2 = 0.999$, a learning rate of 0.001, and a mini-batch size of 8 samples [20]. We eventually collect $50,000$ training examples and train the neural network on these for 50 epochs.

Training dataset. We collected training samples from videos by splitting each video into sets of three frames, such that the center frame in each triplet serves as ground truth. We then extracted patches with a size of 300×300 from these frame triplets, allowing us to only select patches with useful information [2]. We prioritized patches that contain sufficiently large motion as well as high-frequency details. To determine the former, we estimated the optical flow between the first and the last patch in each triplet using DIS [21]. We followed the methodology of Niklaus *et al*. to select raw videos and extract $50,000$ samples with a resolution of 300×300 pixels [34]. Specifically, we obtained high-quality videos from YouTube with the extracted samples having an estimated average optical flow of 4 pixels. The largest optical flow is is estimated to be 41 pixels and about 5% of the pixels have an estimated optical flow of at least 21 pixels.

Data augmentation. While the raw samples in the training dataset have a resolution of 300×300 pixels, we only use patches with a size of 256×256 during training. This allows us to augment the training data on the fly by choosing a different randomly cropped patch from a training sample each time it is being used. In order to eliminate potential priors, we randomly flip each patch vertically or horizontally and randomly swap the temporal order.

3.3. Implementation

We implemented our approach using PyTorch. In our implementation of PWC-Net [43], we realized the necessary cost volume layer using CUDA and utilize the grid sampler from cuDNN in order to perform the involved warping. Wherever available, we utilized optimized cuDNN layers to implement the synthesis network [7]. We implemented the pre-warping algorithm using CUDA and leverage atomic operations to efficiently deal with race conditions. Fully training the synthesis network using a Nvidia Titan X (Pascal) takes about two days. On this graphics card, it takes 0.77 seconds to interpolate a 1920×1080 frame and 0.36 seconds to interpolate a 1280×720 frame. This includes all steps, the bidirectional optical flow estimation, the context extraction, the pre-warping, and the guided synthesis.

	AVERAGE		Laboratory		Synthetic		Real World	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Ours - $\mathcal{L}_{Lap}$	**36.93**	**0.964**	**40.09**	**0.971**	**38.89**	**0.981**	**31.79**	**0.939**
Ours - $\mathcal{L}_1$	36.71	0.963	39.90	**0.971**	38.50	0.980	31.74	**0.939**
Ours - $\mathcal{L}_F$	35.95	0.959	39.21	0.968	37.17	0.975	31.46	0.933
SepConv - $\mathcal{L}_1$	35.73	0.959	39.93	**0.971**	36.78	0.974	30.47	0.932
SepConv - $\mathcal{L}_F$	35.03	0.954	39.49	0.968	35.59	0.967	30.02	0.927
MDP-Flow2	34.81	0.953	38.40	0.966	35.07	0.959	30.97	0.934
DeepFlow2	34.34	0.951	38.33	0.966	34.63	0.956	30.07	0.932
Meyer *et al.*	30.89	0.883	34.22	0.912	27.88	0.813	30.58	0.925

Table 1: Evaluation of the loss functions.

	AVERAGE		Laboratory		Synthetic		Real World	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
net. w/ context	**36.93**	**0.964**	**40.09**	**0.971**	**38.89**	**0.981**	**31.79**	**0.939**
net. w/o context	35.48	0.958	39.07	0.968	35.69	0.967	31.68	**0.939**
bidirect. blending	34.71	0.952	38.25	0.966	34.27	0.953	31.59	0.938
forward blending	34.01	0.949	37.94	0.965	33.38	0.950	30.70	0.933

Table 2: Frame synthesis network vs pixel-wise blending.

Instance normalization. We have found instance normalization to improve the interpolation quality [45]. Specifically, we jointly normalize the two input frames and reverse this normalization on the resulting output from the synthesis network. We likewise jointly normalize the extracted context information but do not need to reverse this operation afterwards. There are two arguments that support this step. First, it removes instance-specific contrast information, removing one possible type of variation in the input. Second, it ensures that the color space and the context space have a similar range, which helps to train the synthesis network.

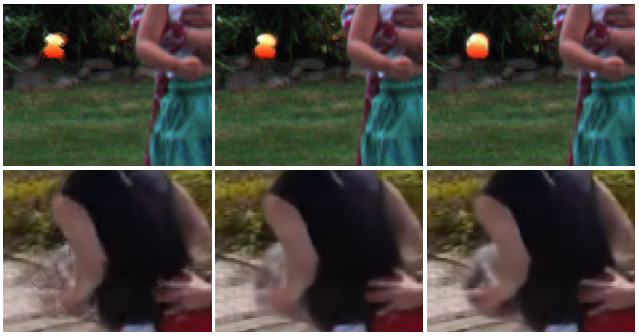
4. Experiments

To evaluate our method, we quantitatively and qualitatively compare it with several baselines as well as representative state-of-the-art video frame interpolation methods. Please also refer to the supplementary video demo to examine the visual quality of our results.

Since the interpolation category of the Middlebury optical flow benchmark is typically used for assessing frame interpolation methods [1, 31, 33, 34], we compare our approach with the methods that perform best on this interpolation benchmark and that are publicly available. Specifically, we select MDP-Flow2 [50], a classic optical flow method, as it ranks first among all the published methods on the Middlebury benchmark. We also select DeepFlow2 [47] as a representative deep learning-based optical flow algorithm. For these optical flow algorithms, we use the algorithm from Baker *et al*. to produce interpolation results [1]. We furthermore select the recent SepConv [34] method which is based on adaptive separable convolutions, as well as the the phase-based interpolation approach from Meyer *et al*. [31]. In ablation experiments, we will additionally compare several variations of our proposed approach, for example, a version that does not use contextual information.

Ours - $\mathcal{L}_1$ Ours - $\mathcal{L}_{Lap}$ Ours - $\mathcal{L}_F$

Figure 4: Examples of using different loss functions.



bidirect. blending new w/o context net. w/ context

Figure 5: Examples for the ablation experiments.

4.1. Ablation experiments

We first evaluate the different loss functions for training our frame synthesis neural network. We then discuss a few design choices and compare our method to several baseline versions in order to assess their effect.

We conduct the ablation experiments quantitatively and use the examples from the Middlebury optical flow benchmark that have publicly available ground truth interpolation results [1]. There are twelve such examples, which were either obtained in a lab environment with controlled lighting, synthetically rendered, or acquired by filming real-world scenes. We assess each category, consisting of four examples, separately and measure PSNR as well as SSIM [46].

Loss functions. We consider three different loss functions to train our frame synthesis neural network, as detailed in Section 3.1. Whereas the ℓ_1 color loss and the Laplacian loss aim to minimize the color difference, the feature loss focuses on perceptual difference. For simplicity, we refer to the model that has been trained with the color loss as "$\mathcal{L}_1$" and the model trained with the Laplacian loss as "$\mathcal{L}_{Lap}$". Following Niklaus *et al*. [34], we do not directly train the model with the feature loss. Instead, we first train it with the Laplacian loss and then refine it with the feature loss. We refer to this model as "$\mathcal{L}_F$".

As reported in Table 1, the Laplacian loss $\mathcal{L}_{Lap}$ and the

	AVERAGE		Laboratory		Synthetic		Real World	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
ResNet-18 - `conv1`	**36.93**	**0.964**	**40.09**	**0.971**	**38.89**	**0.981**	31.79	0.939
VGG-19 - `conv1_2`	36.77	**0.964**	39.97	**0.971**	38.53	0.980	**31.81**	**0.940**
VGG-19 - `conv1_1`	36.69	0.963	39.90	**0.971**	38.37	0.979	31.80	**0.940**
no context	35.48	0.958	39.07	0.968	35.69	0.967	31.68	0.939

Table 3: Effect of contextual information.

	AVERAGE		Laboratory		Synthetic		Real World	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
flow of PWC-Net	**36.93**	**0.964**	**40.09**	**0.971**	**38.89**	**0.981**	**31.79**	**0.939**
flow of SPyNet	35.78	0.959	39.89	0.971	36.87	0.971	30.58	0.935
H.264 motion vec.	35.44	0.957	39.42	0.968	36.83	0.975	30.07	0.927
no flow / warping	34.98	0.955	38.67	0.963	35.98	0.966	30.30	0.935

Table 4: Effect of different optical flow algorithms.

	AVERAGE			Mequon			Schefflera			Urban			Teddy			Backyard			Basketball			Dumptruck			Evergreen		
	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.	all	disc.	unt.
Ours - $\mathcal{L}_{Lap}$	**5.28**	**8.00**	2.19	**2.24**	**3.72**	**1.04**	**2.96**	**4.16**	1.35	4.32	**3.42**	3.18	**4.21**	**5.46**	**3.00**	**9.6**	**11.9**	3.46	**5.22**	**9.8**	2.22	7.02	**15.4**	1.58	6.66	**10.2**	1.69
SepConv - $\mathcal{L}_1$	5.61	8.74	2.33	2.52	4.83	1.11	3.56	5.04	1.90	4.17	4.15	2.86	5.41	6.81	3.88	10.2	12.8	3.37	5.47	10.4	**2.21**	6.88	15.6	1.72	**6.63**	10.3	**1.62**
SepConv - $\mathcal{L}_F$	5.81	9.04	2.40	2.60	5.00	1.19	3.87	5.50	2.07	4.38	4.29	2.73	5.78	7.16	3.94	10.1	12.7	3.39	5.98	11.4	2.42	**6.85**	15.5	1.78	6.90	10.8	1.65
MDP-Flow2	5.83	9.69	2.15	2.89	5.38	1.19	3.47	5.07	**1.26**	3.66	6.10	2.48	5.20	7.48	3.14	10.2	12.8	3.61	6.13	11.8	2.31	7.36	16.8	**1.49**	7.75	12.1	1.69
DeepFlow2	6.02	9.94	**2.06**	2.99	5.65	1.22	3.88	5.79	1.48	**3.62**	6.03	**1.34**	5.38	7.44	3.22	11.0	13.8	3.67	5.83	11.2	2.25	7.60	17.4	1.50	7.82	12.2	1.77

Table 5: Evaluation on the Middlebury benchmark. *disc.*: regions with discontinuous motion. *unt.*: textureless regions.

color loss $\mathcal{L}_1$, especially the former, outperform the feature loss $\mathcal{L}_F$ as well as the state-of-the-art methods quantitatively. As expected, the color loss functions yield better results since they directly optimize the evaluation metric. On the other hand, we find that the feature loss tends to produce visually more pleasant results, as shown in our user study in Section 4.3 and Figure 4. This finding is consistent with what was reported in recent work [34].

Frame synthesis network vs pixel-wise blending. To examine the effectiveness of our frame synthesis neural network, we compare it to a blending baseline that employs the off-the-shelf warping algorithm from the Middlebury benchmark [1] to warp the input frames and then blends them together. The same bidirectional optical flow used in our synthesis network approach is used in this baseline. As a reference, we also compare to another baseline that also employs the algorithm from the Middlebury benchmark but only uses the forward flow for warping. As shown in Table 2, our synthesis network approach shows a clear advantage over the blending approach. This can be attributed to the capability of our synthesis network in tolerating inaccuracies in optical flow estimation. Note that the demonstrated quantitative advantage also translates to improved visuals as shown in Figure 5.

Contextual information. To understand the effect of the contextual information, we trained a synthesis network that only receives two pre-warped frames as input. As shown in Table 3, the contextual information significantly helps our frame synthesis network to produce high-quality results.

Our method uses the `conv1` layer of ResNet-18 [14] to extract per-pixel contextual information. We also tested other options, such as using layers `conv1_1` and `conv1_2` of VGG-19 [41]. As reported in Table 3, while the `conv1` layer of ResNet-18 overall works better, the difference between various ways to extract contextual information is minuscule. They all significantly outperform the baseline network without contextual information.

Optical flow. To evaluate the effect of the utilized optical flow algorithm on our method, we use SPyNet [38] as an alternative to our implementation of PWC-Net [43]. In optical flow benchmarks, SPyNet performs less well than PWC-Net and as shown in Table 4, this decreased accuracy also affects the synthesis results. Furthermore, training a synthesis network to directly operate on the input frames, without any pre-warping, significantly worsens the synthesis quality. This shows that it is helpful to use motion compensation to provide a good initialization for frame interpolation. In fact, even using H.264 motion vectors is already beneficial. This is remarkable, considering that these motion vectors are only available per block.

4.2. Quantitative evaluation

We perform a quantitative comparison with state of the art frame interpolation algorithms on the interpolation section of the Middlebury benchmark for optical flow [1]. As reported in Table 5, our method establishes a new state-of-the-art and improves the previously best performing method by a notable margin. Furthermore, according to the feedback from the Middlebury benchmark organizer, our interpolation results are ranked 1st among all the over 100 algorithms listed on the benchmark website.

4.3. Visual comparison

We show a comparison of our proposed approach with state-of-the-art video frame interpolation methods in Figure 6. These examples are subject to significant motion and occlusion. In general, our approach is capable of handling these challenging scenarios, with the $\mathcal{L}_F$ loss trained synthesis network retaining more high-frequency details. In comparison, SepConv-$\mathcal{L}_F$ fails to compensate for the large motion and is limited by the size of its adaptive kernels. The optical flow based methods MDP-Flow2, DeepFlow2, and

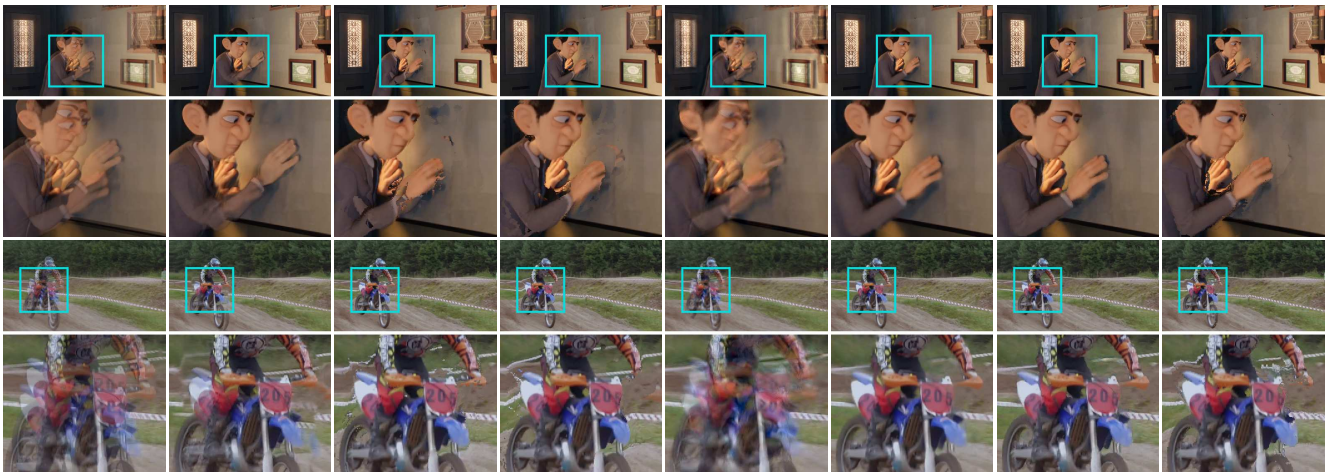


Figure 6: Visual comparison among frame interpolation methods. We used our own implementation of PWC-Net here.

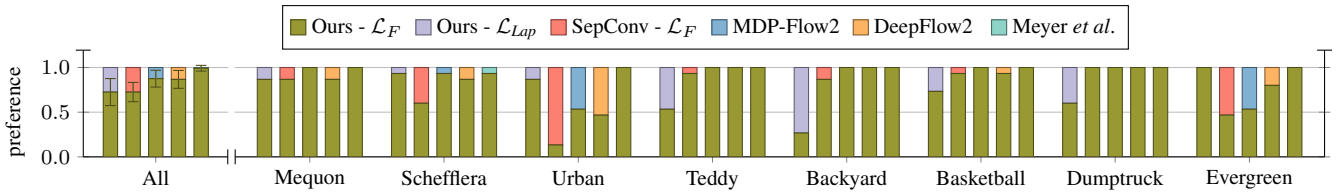


Figure 7: Results from the user study. The error bars denote the standard deviation.

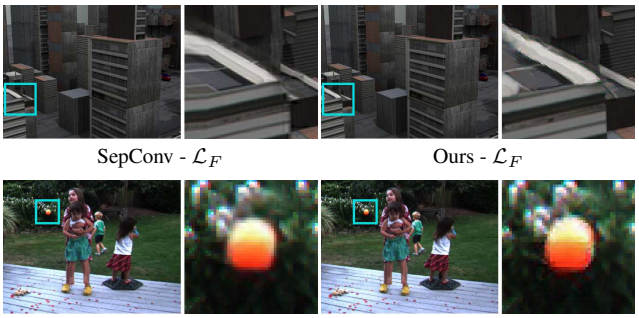


Figure 8: Examples for which our $\mathcal{L}_F$ results were not preferred in the user study.

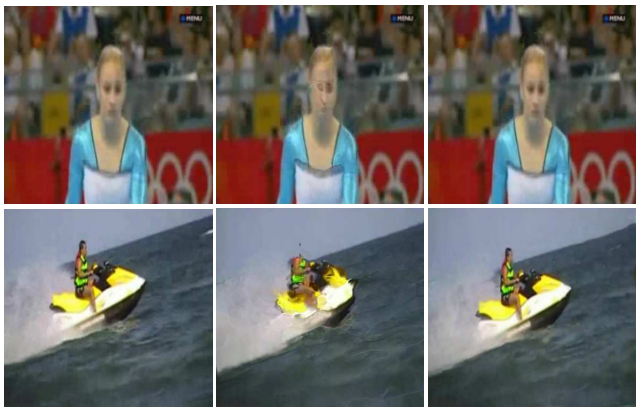


Figure 9: Comparison with the recent voxel flow method.

PWC-Net handle the large motion better than SepConv-$\mathcal{L}_F$ but introduce artifacts due to their limited synthesis capabilities. Lastly, the approach from Meyer *et al*. fails to handle large motion due to phase ambiguities. Please refer to the supplemental video to see these examples in motion.

User study. We also conducted a user study to further compare our method to other frame interpolation methods on all eight examples from the Middlebury benchmark. Specifically, we compare the results of our approach when using $\mathcal{L}_F$ loss with five state-of-the-art video frame interpolation methods. We recruited 15 students in computer science as participants and supported them with an interface that allowed them to easily switch between two images. For each pair, we asked them to select the better one and let each participant perform 40 such comparisons, thus comparing our result to each of the five competing methods for all 8 examples. The results from this study are summarized in Figure 7. Overall, the participants preferred our approach that utilizes $\mathcal{L}_F$ loss. The scenarios where the preference has not been in our favor, Urban with SepConv-$\mathcal{L}_F$ and Backward with our $\mathcal{L}_{Lap}$ loss, are shown in Figure 8. In the Urban example, SepConv-$\mathcal{L}_F$ has fewer artifacts at the boundary. In the Backyard example, $\mathcal{L}_F$ loss introduced chromatic artifacts around the orange ball that is subject to large motion. Nevertheless, this study shows that overall

Input at $t = 0$ | Ours - $\mathcal{L}_F$ at $t = 0.2$ | Ours - $\mathcal{L}_F$ at $t = 0.4$ | Ours - $\mathcal{L}_F$ at $t = 0.6$ | Ours - $\mathcal{L}_F$ at $t = 0.8$ | Input at $t = 1$

Figure 10: A sequence of frames demonstrating that our method can interpolate at an arbitrary temporal position.

Default GridNet - $\mathcal{L}_F$ | Modified GridNet - $\mathcal{L}_F$

Figure 11: Checkerboard artifacts prevention.

our method with $\mathcal{L}_F$ loss achieves better perceptual quality although $\mathcal{L}_{Lap}$ loss performs better quantitatively.

4.4. Discussion

The recent voxel flow-based video frame interpolation method estimates voxel flow to sample a 2^3 spatio-temporal neighborhood to generate the interpolation results [24]. Since it only samples a 2^3 space-time volume, it is still fairly limited in handling inaccuracies in motion estimation. As shown in Figure 9, our approach is able to produce better results due to its flexibility. Quantitatively, our method has a PSNR of 34.62 while voxel flow has a PSNR of 34.12 on their DVF dataset.

Since we perform motion compensation before synthesizing the output frame, we are able to interpolate a frame at an arbitrary temporal position $t \in [0, 1]$, as shown in Figure 10. Other efforts that use convolutional neural networks for video frame interpolation either had to retrain their synthesis network for a specific t or continue the interpolation recursively in order to achieve this [33, 34]. Furthermore, our proposed approach is not limited to video frame interpolation. In fact, it can also be utilized to synthesize between stereo frames as shown in our supplemental video.

When using $\mathcal{L}_F$ loss, checkerboard artifacts can occur if the architecture of the utilized neural network is not chosen well due to uneven overlaps [35]. To avoid such artifacts, we modified the GridNet architecture of the frame synthesis network and utilize bilinear upsampling instead of transposed convolutions. As shown in Figure 11, this decision successfully prevents checkerboard artifacts.

As discussed in Section 4.1, pre-warping input frames and context maps using optical flow is important for our method to produce high-quality frame interpolation results. While PWC-Net provides a good initialization for our method, future advances in optical flow research will benefit our frame interpolation method.

It has been shown in recent research on image synthesis that a proper adversarial loss can help to produce high quality visual results [40]. It will be interesting to explore its use to further improve the quality of frame interpolation. Furthermore, as shown in recent papers [29, 32], the composition of the training dataset can be important for low-level computer vision tasks. A comprehensive study of the effect of the training dataset in the context of video frame interpolation could potentially provide important domain-specific insights and further improve the interpolation quality.

5. Conclusion

This paper presented a context-aware synthesis approach for video frame interpolation. This method is developed upon three ideas. First, using bidirectional flow in combination with a flexible frame synthesis neural network can handle challenging cases like occlusions and accommodate inaccuracies in motion estimation. Second, contextual information enables our frame synthesis neural network to perform informative interpolation. Third, using optical flow to provide a good initialization for interpolation is helpful. As demonstrated in our experiments, these ideas enable our method to produce high-quality video frame interpolation results and outperform state-of-the-art methods.

Acknowledgments. Figures 1, 4 (top), and 6 (bottom) originate from the DAVIS challenge. Figures 2, 6 (top), and 11 are used under a Creative Commons license from the Blender Foundation. Figure 4 (bottom) originates from the KITTI benchmark. Figures 5 (top), 8, and 10 originate from the Middlebury benchmark. Figure 5 (bottom) is used under a Creative Commons license from GaijinPot. Figure 9 originates from the DVF (from UCF101) dataset. We thank Nvidia for their GPU donation.

References

[1] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, 2011. 1, 2, 5, 6

[2] A. Bansal, X. Chen, B. Russell, A. Gupta, and D. Ramanan. PixelNet: Representation of the pixels, by the pixels, and for the pixels. *arXiv/1702.06506*, 2017. 4

[3] P. Bojanowski, A. Joulin, D. Lopez-Paz, and A. Szlam. Optimizing the latent space of generative networks. *arXiv/1707.05776*, 2017. 4

[4] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision*, volume 7577, pages 611–625, 2012. 1, 2

[5] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. In *IEEE International Conference on Computer Vision*, Oct 2017. 4

[6] S. E. Chen and L. Williams. View interpolation for image synthesis. In *Conference on Computer Graphics and Interactive Techniques, SIGGRAPH*, pages 279–288, 1993. 2

[7] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer. cuDNN: Efficient primitives for deep learning. *arXiv/1410.0759*, 2014. 4

[8] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, pages 658–666, 2016. 4

[9] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. FlowNet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision*, pages 2758–2766, 2015. 2

[10] D. Fourure, R. Emonet, É. Fromont, D. Muselet, A. Trémeau, and C. Wolf. Residual conv-deconv grid network for semantic segmentation. In *British Machine Vision Conference*, 2017. 3

[11] D. Gadot and L. Wolf. PatchBatch: A batch augmented loss for optical flow. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4236–4245, 2016. 2

[12] R. Goroshin, M. Mathieu, and Y. LeCun. Learning to linearize under uncertainty. In *Advances in Neural Information Processing Systems*, pages 1234–1242, 2015. 4

[13] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *IEEE International Conference on Computer Vision*, pages 1026–1034, 2015. 4

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2, 3, 6

[15] E. Herbst, S. Seitz, and S. Baker. Occlusion reasoning for temporal interpolation using optical flow. Technical report, August 2009. 1, 2

[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, volume 37, pages 448–456, 2015. 4

[17] J. Janai, F. Guney, J. Wulff, M. J. Black, and A. Geiger. Slow flow: Exploiting high-speed cameras for accurate and diverse optical flow reference data. In *IEEE Conference on Computer Vision and Pattern Recognition*, July 2017. 1, 2

[18] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, volume 9906, pages 694–711, 2016. 4

[19] S. B. Kang, Y. Li, X. Tong, and H. Shum. Image-based rendering. *Foundations and Trends in Computer Graphics and Vision*, 2(3), 2006. 2

[20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 4

[21] T. Kroeger, R. Timofte, D. Dai, and L. V. Gool. Fast optical flow using dense inverse search. In *European Conference on Computer Vision*, pages 471–488, 2016. 4

[22] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. P. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv/1609.04802*, 2016. 4

[23] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *CVPR Workshops*, pages 1132–1140, 2017. 4

[24] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala. Video frame synthesis using deep voxel flow. In *IEEE International Conference on Computer Vision*, Oct 2017. 1, 2, 8

[25] 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*, volume 9910, pages 434–450, 2016. 2, 4

[26] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 4

[27] W. Luo, Y. Li, R. Urtasun, and R. S. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 4898–4906, 2016. 4

[28] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *International Conference on Learning Representations*, 2016. 4

[29] N. Mayer, E. Ilg, P. Fischer, C. Hazirbas, D. Cremers, A. Dosovitskiy, and T. Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *https://arxiv.org/abs/1801.06397*, 2018. 8

[30] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3061–3070, 2015. 2

[31] 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*, pages 1410–1418, 2015. 2, 5

[32] S. Nah, T. Hyun Kim, and K. Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, July 2017. 4, 8

[33] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive convolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, July 2017. 1, 2, 5, 8

[34] S. Niklaus, L. Mai, and F. Liu. Video frame interpolation via adaptive separable convolution. In *IEEE International Conference on Computer Vision*, Oct 2017. 1, 2, 4, 5, 6, 8

[35] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. http://distill.pub/2016/deconv-checkerboard. 4, 8

[36] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe. Full-resolution residual networks for semantic segmentation in street scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3309–3318, 2017. 4

[37] L. L. Rakêt, L. Roholm, A. Bruhn, and J. Weickert. Motion compensated frame interpolation with a symmetric optical flow constraint. In *Advances in Visual Computing*, volume 7431, pages 447–457, 2012. 1, 2

[38] A. Ranjan and M. J. Black. Optical flow estimation using a spatial pyramid network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2720–2729, 2017. 6

[39] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra. Video (language) modeling: a baseline for generative models of natural videos. *arXiv/1412.6604*, 2014. 4

[40] M. S. M. Sajjadi, B. Schölkopf, and M. Hirsch. EnhanceNet: Single image super-resolution through automated texture synthesis. *arXiv/1612.07919*, 2016. 4, 8

[41] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv/1409.1556*, 2014. 4, 6

[42] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *International Conference on Machine Learning*, volume 37, pages 843–852, 2015. 4

[43] D. Sun, X. Yang, M. Liu, and J. Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. *arXiv/1709.02371*, 2017. 2, 3, 4, 6

[44] R. Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010. 2

[45] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv/1607.08022*, 2016. 5

[46] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5

[47] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. In *IEEE International Conference on Computer Vision*, pages 1385–1392, 2013. 2, 5

[48] M. Werlberger, T. Pock, M. Unger, and H. Bischof. Optical flow guided TV-L 1 video interpolation and restoration. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 6819, pages 273–286, 2011. 2

[49] J. Xu, R. Ranftl, and V. Koltun. Accurate optical flow via direct cost volume processing. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5807–5815, 2017. 2

[50] L. Xu, J. Jia, and Y. Matsushita. Motion detail preserving optical flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1744–1757, 2012. 1, 2, 5

[51] Z. Yu, H. Li, Z. Wang, Z. Hu, and C. W. Chen. Multi-level video frame interpolation: Exploiting the interaction among different levels. *IEEE Trans. Circuits Syst. Video Techn.*, 23(7):1235–1248, 2013. 2

[52] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros. View synthesis by appearance flow. In *European Conference on Computer Vision*, volume 9908, pages 286–301, 2016. 1, 2

[53] J. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, volume 9909, pages 597–613, 2016. 4