Deep Video Deblurring for Hand-held Cameras

Shuochen Su
Univerity of British Columbia

Mauricio Delbracio
Universidad de la República

Jue Wang
Adobe Research

Guillermo Sapiro
Duke University

Wolfgang Heidrich
KAUST

Oliver Wang
Adobe Research

Abstract

Motion blur from camera shake is a major problem in videos captured by hand-held devices. Unlike single-image deblurring, video-based approaches can take advantage of the abundant information that exists across neighboring frames. As a result the best performing methods rely on the alignment of nearby frames. However, aligning images is a computationally expensive and fragile procedure, and methods that aggregate information must therefore be able to identify which regions have been accurately aligned and which have not, a task that requires high level scene understanding. In this work, we introduce a deep learning solution to video deblurring, where a CNN is trained end-to-end to learn how to accumulate information across frames. To train this network, we collected a dataset of real videos recorded with a high frame rate camera, which we use to generate synthetic motion blur for supervision. We show that the features learned from this dataset extend to deblurring motion blur that arises due to camera shake in a wide range of videos, and compare the quality of results to a number of other baselines [1].

1. Introduction

Hand-held video capture devices are now commonplace. As a result, video stabilization has become an essential step in video capture pipelines, often performed automatically at capture time (e.g., iPhone, Google Pixel), or as a service on sharing platforms (e.g., Youtube, Facebook). While stabilization techniques have improved dramatically, the remaining motion blur is a major problem with all stabilization techniques. This is because the blur becomes obvious when there is no motion to accompany it, yielding highly visible "jumping" artifacts. In the end, the remaining camera shake motion blur limits the amount of stabilization that can be applied before these artifacts become too apparent.

[1]Datasets, pretrained models and source code are available at https://www.cs.ubc.ca/labs/imager/tr/2017/DeepVideoDeblurring

Figure 1: Blur in videos can be significantly attenuated by learning how to aggregate information from nearby frames. Top: crops of consecutive frames from a blurry video; Bottom: outputs from the proposed data-driven approach, in this case using simple homography alignment.

The most successful video deblurring approaches leverage information from neighboring frames to sharpen blurry ones, taking advantage of the fact that most hand-shake motion blur is both short and temporally uncorrelated. By borrowing "sharp" pixels from nearby frames, it is possible to reconstruct a high quality output. Previous work has shown significant improvement over traditional deconvolution-based deblurring approaches, via patch-based synthesis that relies on either lucky imaging [4] or weighted Fourier aggregation [6].

One of the main challenges associated with aggregating information across multiple video frames is that the differently blurred frames must be aligned. This can either be done, for example, by nearest neighbor patch lookup [4], or optical flow [6]. However, warping-based alignment is not robust around disocclusions and areas with low texture, and often yields warping artifacts. In addition to the alignment computation cost, methods that rely on warping have

to therefore disregard information from mis-aligned content or warping artifacts, which can be hard by looking at local image patches alone.

To this end, we present the first end-to-end data-driven approach to video deblurring, the results of which can be seen in Fig. 1. We address specifically blur that arises due to hand-held camera shake, i.e., is temporally uncorrelated, however we show that our deblurring extends to other types of blur as well, including motion blur from object motion. We experiment with a number of differently learned configurations based on various alignment types: no-alignment, frame-wise homography alignment, and optical flow alignment. On average optical flow performs the best, although in many cases projective transform (i.e. homography) performs comparably with significantly less computation required. Notably, our approach also enables the generation of high quality results *without* computing any alignment or image warping, which makes it highly efficient and robust to scene types. Essential to this success is the use of an autoencoder-type network with skip connections that increases the receptive field and is yet easy to train.

Our main contribution is an end-to-end solution to train a deep neural network to learn how to deblur images, given a short stack of neighboring video frames. We describe the architecture we found to give the best results, and the method we used to create a real-world dataset from high frame rate capture. We compare qualitatively to videos previously used for video deblurring, and quantitatively with our ground truth data set. We also present a test set of videos showing that our method generalizes to a wide range of scenarios. Both datasets are made available to the public to encourage follow up work.

2. Related Work

There exist two main approaches to deblurring: deconvolution-based methods that solve inverse problems, and those that rely on multi-image aggregation and fusion.

Deblur using deconvolution. Modern single-image deblurring approaches jointly estimate a blurring kernel (either single or spatially varying) and the underlying sharp image via deconvolution [23]. In recent years many successful methods have been introduced [3, 8, 22, 32, 39, 42, 51, 52], see [47] for a recent survey. Multiple-image deconvolution methods use additional information to alleviate the severe ill-posedness of single-image deblurring. These approaches collect, for example, image bursts [14], blurry-noisy pairs [53], flash no-flash image pairs [36], gyroscope information [34], high frame rate sequences [44], or stereo pairs [38] for deblurring. These methods generally assume static scenes and require the input images to be aligned. For video, temporal information [25], optical flow [17] and scene models [33, 49] have been used for improving both kernel and latent frame estimation.

All of the above approaches strongly rely on the accuracy of the assumed image degradation model (blur, motion, noise) and its estimation, thus may perform poorly when the simplified degradation models are insufficient to describe real data, or due to suboptimal model estimation. As a result, these approaches tend to be more fragile than aggregation-based methods [6], and often introduce undesirable artifacts such as ringing and amplified noise.

Multi-image aggregation. Multi-image aggregation methods directly combine multiple images in either spatial or frequency domain without solving any inverse problem. Lucky-imaging is a classic example, in which multiple low quality images are aligned and best pixels from different ones are selected and merged into the final result [15, 24]. For denoising, this has been extended to video using optical flow [26] or piecewise homographies [28] for alignment.

For video deblurring, aggregation approaches rely on the observation that in general not all video frames are equally blurred. Sharp pixels thus can be transferred from nearby frames to deblur the target frame, using for example homography alignment [30]. Cho et al. further extend this approach using patch-based alignment [4] for improved robustness against moving objects. The method however cannot handle large depth variations due to the underlying homography motion model, and the patch matching process is computationally expensive. Klose et al. [20] show that 3D reconstruction can be used to project pixels into a single reference coordinate system for pixel fusion. Full 3D reconstruction however can be fragile for highly dynamic videos.

Recently, Delbracio and Sapiro [5] show that aggregating multiple aligned images in the Fourier domain can lead to effective and computationally highly efficient deblurring. This technique was extended to video [6], where nearby frames are warped via optical flow for alignment. This method is limited by optical flow computation and evaluation, which is not reliable near occlusions and outliers.

All above approaches have explicit formulations on how to fuse multiple images. In this work, we instead adopt a data-driven approach to *learn* how multiple images should be aggregated to generate an output that is as sharp as possible.

Data-driven approaches. Recently, CNNs have been applied to achieve leading results on a wide variety of reconstruction problems. These methods tend to work best when large training datasets can be easily constructed, for example by adding synthetic noise for denoising [50], removing content for inpainting [35], removing color information for colorization [13], or downscaling for superresolution [7, 27]. Super resolution networks have been applied to video sequences before [12, 16, 40], but these approaches

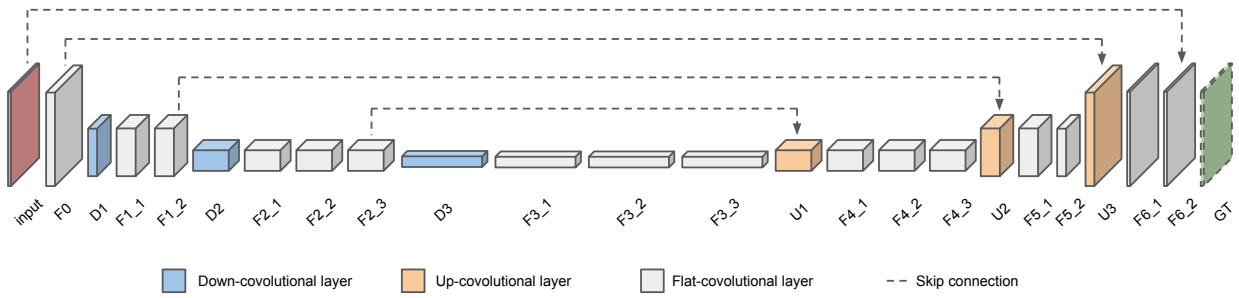

Figure 2: Architecture of the proposed *DeBlurNet* model, that takes the stacked nearby frames as input, and processes them jointly through a number of convolutional layers until generating the deblurred central frame. The depth of each block represents the number of activation maps in response to learned kernels. See Tab. 1 for detailed configurations.

address a different problem, with its own set of challenges. In this work we focus on deblurring, where blurry frames can vary greatly in appearance from their neighbors, making information aggregation more challenging.

CNNs have also been used for single- [2, 43] and multi- [48] image deblurring, using synthetic training data. One problem with synthetic blur is that real blur has significantly different characteristics, as it depends on both the scene depth and object motion. In our experiments, we show that by leveraging multiple video frames, training on real blur, and directly estimating the sharp images, our method can produce substantially better results.

3. Our Method

Overview. Image alignment is inherently challenging as determining whether the aligned pixels in different images correspond to the same scene content can be difficult with only low-level features. High-level features, on the other hand, provide sufficient additional information to help separate incorrectly aligned image regions from correctly aligned ones. To make use of both low-level and high-level features, we therefore train an end-to-end system for video deblurring, where the input is a stack of neighboring frames and the output is the deblurred *central* frame in the stack. Furthermore, our network is trained using real video frames with realistically synthesized motion blur. In the following, we first present our neural network architecture, then describe a number of experiments for evaluating its effectiveness and comparing with existing methods. The key advantage of our approach is the allowance of lessening the requirements for accurate alignment, a fragile component of prior work.

3.1. Network Architecture

We use an encoder-decoder style network, which have been shown to produce good results for a number of generative tasks [35, 41]. In particular, we choose a variation of the fully convolutional model proposed in [41] for sketch cleanup. We add symmetric skip connections [29] between corresponding layers in encoder and decoder halves of the

layer	kernel size	stride	output size	skip connection
input	-	-	$15\times H\times W$	to F6_2*
F0	5×5	1×1	$64\times H\times W$	to U3
D1	3×3	2×2	$64\times H/2\times W/2$	-
F1_1	3×3	1×1	$128\times H/2\times W/2$	-
F1_2	3×3	1×1	$128\times H/2\times W/2$	to U2
D2	3×3	2×2	$256\times H/4\times W/4$	-
F2_1	3×3	1×1	$256\times H/4\times W/4$	-
F2_2	3×3	1×1	$256\times H/4\times W/4$	-
F2_3	3×3	1×1	$256\times H/4\times W/4$	to U1
D3	3×3	2×2	$512\times H/8\times W/8$	-
F3_1	3×3	1×1	$512\times H/8\times W/8$	-
F3_2	3×3	1×1	$512\times H/8\times W/8$	-
F3_3	3×3	1×1	$512\times H/8\times W/8$	-
U1	4×4	$1/2\times1/2$	$256\times H/4\times W/4$	from F2_3
F4_1	3×3	1×1	$256\times H/4\times W/4$	-
F4_2	3×3	1×1	$256\times H/4\times W/4$	-
F4_3	3×3	1×1	$256\times H/4\times W/4$	-
U2	4×4	$1/2\times1/2$	$128\times H/2\times W/2$	from F1_2
F5_1	3×3	1×1	$128\times H/2\times W/2$	-
F5_2	3×3	1×1	$64\times H/2\times W/2$	-
U3	4×4	$1/2\times1/2$	$64\times H\times W$	from F0
F6_1	3×3	1×1	$15\times H\times W$	-
F6_2	3×3	1×1	$3\times H\times W$	from input*

Table 1: Specifications of the DBN model. Each convolutional layer is followed by batch normalization and ReLU, except those that are skip connected to deeper layers, where only batch normalization has been applied, before the sum is rectified through a ReLU layer [11]. For example, the input to F4_1 is the rectified summation of U1 and F2_3. Note that for the skip connection from input layer to F6_2, only the central frame of the stack is selected. At the end of the network a Sigmoid layer is applied to normalize the intensities. We use the Torch implementation of `SpatialConvolution` and `SpatialFullConvolution` for down- and up-convolutional layers.

Figure 3: A selection of blurry/sharp pairs (split left/right respectively) from our ground truth dataset. Images are best viewed on-screen and zoomed in.

network, where features from the encoder side are added element-wise to each corresponding layer. This significantly accelerates the convergence and helps generate much sharper video frames. We perform an early fusion of neighboring frames that is similar to the FlowNetSimple model in [9], by concatenating all images in the input layer. The training loss is MSE to the ground truth sharp image, which will be discussed in more detail in Sec. 4. We refer to this network as *DeBlurNet*, or DBN, and show a diagram of it in Fig. 2. It consists of three types of convolutional layers: down-convolutional layers, that compress the spatial resolution of the features while increasing the spatial support of subsequent layers; flat-convolutional layers, that perform non-linear mapping and preserve the size of the image; and finally up-convolutional layers, that increase the spatial resolution. Please refer to Tab. 1 for detailed configurations.

Alignment. One of the main advantages of our method is the ability to work well without accurate frame-to-frame alignment. To this end, we create three versions of our dataset with varying degrees of alignment, and use these to train DBN. At one end, we use no alignment at all, relying on the network to abstract spatial information through a series of down-convolution layers. This makes the method significantly faster, as alignment usually dominates running time in multi-frame aggregation methods. We refer to this network as DBN+NOALIGN. We also use optical flow [37] to align stacks (DBN+FLOW), which is slow to compute and prone to errors (often introducing additional warping artifacts), but allows pixels to be aggregated more easily by removing the spatial variance of corresponding features. Finally, we use a single global homography to align frames, which provides a compromise in approaches, in terms of computational complexity and alignment quality (DBN+HOMOG). The homography is estimated using SURF features and a variant of RANSAC [46] to reject outliers.

Implementation details. During training we use a batch size of 64, and patches of 15×128×128, where 15 is the total number of RGB channels stacked from the crops of 5 consecutive video frames. We observed that a patch size of 128 was sufficient to provide enough overlapping content in the stack even if the frames are not aligned. We use ADAM [19] for optimization, and fix the learning rate to be 0.005 in the first 24,000 iterations, then halves for every subsequent 8,000 iterations until it reaches the lower bound of 10^{-6}. For all the results reported in the paper we train the network for 80,000 iterations, which takes about 45 hours on an NVidia Titan X GPU. Default values of β_1, β_2 and ϵ are used, which are 0.9, 0.999, and 10^{-8} respectively, and we set weight decay to 0.

As our network is fully convolutional, the input resolution is restricted only by GPU memory. At test time, we pass a 960×540 frame into the network, and tile this if the video frame is of larger resolution. Since our approach deblurs images in a single forward pass, it is computationally very efficient. Using an NVidia Titan X GPU, we can process a 720p frame within 1s without alignment. Previous approaches took on average 15s [6] and 30s [4] per frame on CPUs. The recent neural deblurring method [2] takes more than 1 hour to fully process each frame, and the approach of Kim et al. [17] takes several minutes per frame.

4. Training Dataset

Generating realistic training data is a major challenge for tasks where ground truth data cannot be easily collected/labeled. For training our neural network, we require two video sequences of exactly the same content: one blurred by camera shake motion blur, and its corresponding sharp version. Capturing such data is extremely hard. One could imagine using a beam-splitter and multiple cameras to build a special capturing system, but this setup would be challenging to construct robustly, and would present a host of other calibration issues.

One solution would be to use rendering techniques to create synthetic videos for training. However if not done properly, this often leads to a domain gap, where models trained on synthetic data do not generalize well to real world data. For example, we could apply synthetic motion blur on sharp video frames to simulate camera shake blur. However, in real world scenarios the blur not only depends on camera motion, but also is related to scene depth and object motion, thus is very difficult to be rendered properly.

In this work, we propose to collect real-world sharp videos at very high frame rate, and synthetically create

Method	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Average
Input	24.14 / .859	30.52 / .958	28.38 / .914	27.31 / .900	22.60 / .852	29.31 / .951	27.74 / .939	23.86 / .906	30.59 / .976	26.98 / .926	27.14 / .918
PSDEBLUR	24.42 / .908	28.77 / .952	25.15 / .928	27.77 / .928	22.02 / .890	25.74 / .932	26.11 / .948	19.75 / .822	26.48 / .963	24.62 / .938	25.08 / .921
WFA [6]	25.89 / .910	32.33 / .974	28.97 / .931	28.36 / .925	23.99 / .910	31.09 / .975	28.58 / .955	24.78 / .926	31.30 / .981	28.20 / .960	28.35 / .944
DBN+SINGLE	25.75 / .901	31.15 / .966	29.30 / .946	28.38 / .922	23.63 / .885	30.70 / .962	29.23 / .959	25.62 / .936	31.92 / .983	28.06 / .949	28.37 / .941
DBN+NOALIGN	27.83 / .940	**33.11** / .980	**31.29 / .973**	29.73 / .948	25.12 / .930	**32.52** / .978	**30.80 / .975**	**27.28** / .962	**33.32 / .989**	29.51 / .969	**30.05** / .964
DBN+HOMOG.	27.93 / .945	32.39 / .975	30.97 / .969	29.82/ .948	24.79 / .925	31.84 / .972	30.46 / .972	26.64 / .955	33.15 / **.989**	29.30 / .969	29.73 / .962
DBN+FLOW	**28.31 / .956**	33.14 / **.982**	30.92 / **.973**	**29.99 / .954**	**25.58 / .944**	32.39 / **.981**	30.56 / **.975**	27.15 / **.963**	32.95 / **.989**	**29.53 / .975**	**30.05 / .969**

Table 2: PSNR/MSSIM [21] measurements for each approach, averaged over all frames, for 10 test datasets (#1→#10).

blurred ones by accumulating a number of short exposures to approximate a longer exposure [45]. In order to simulate realistic motion blur at 30fps, we capture videos at 240fps, and subsample every eighth frame to create the 30fps ground truth sharp video. We then average together a temporally centered window of 7 frames (3 on either side of the ground truth frame) to generate synthetic motion blur at the target frame rate.

Since there exists a time period between adjacent exposures (the "duty cycle"), simply averaging consecutive frames will yield ghosting artifacts. To avoid this, [18] proposed to only use frames whose relative motions in-between are smaller than 1 pixel. To use all frames for rendering, we compute optical flow between adjacent high fps frames, and generate an additional 10 evenly spaced inter-frame images, which we then average together. Examples of the dataset are shown in Fig. 3. We have also released this dataset publicly for future research.

In total, we collect 71 videos, each with 3-5s average running time. These are used to generate 6708 synthetic blurry frames with corresponding ground truth. We subsequently augment the data by flipping, rotating (0°, 90°, 180°, 270°), and scaling ($1/4$, $1/3$, $1/2$) the images, and from this we draw on average 10 random 128×128 crops. In total, this gives us 2,146,560 pairs of patches. We split our dataset into 61 training videos and 10 testing videos. For each video, its frames are used for either training or testing, but not both, meaning that the scenes used for testing have not been seen in the training data.

The training videos are capture at 240fps with an iPhone 6s, GoPro Hero 4 Black, and Canon 7D. The reason to use multiple devices is to avoid bias towards a specific capturing device that may generate videos with some unique characteristics. We test on videos captured by other devices, including Nexus 5x and Moto X mobile phones and a Sony a6300 consumer camera.

Limitations. We made an significant effort to capture a wide range of situations, including long pans, selfie videos, scenes with moving content (people, water, trees), recorded with a number of different capture devices. While it is quite diverse, it also has some limitations. As our blurry frames are averaged from multiple input frames, the noise characteristics will be different in the ground truth image. To reduce this effect, we recorded input videos in high light situations, where there was minimal visible noise even in the original 240fps video, meaning that our dataset only contains scenes with sufficient light. An additional source of error is that using optical flow for synthesizing motion blur adds possible artifacts which would not exist in real-world data. We found that however, as the input video is recorded at 240fps, the motion between frames is small, and we did not observe visual artifacts from this step.

As we will show in Sec. 5, despite these limitations, our trained model still generalizes well to new capture devices and scene types, notably on low-light videos. We believe future improvements to the training data set will further improve the performance of our method.

5. Experiments and Results

We conduct a series of experiments to evaluate the effectiveness of the learned model, and also the importance of individual components.

Effects of using multiple frames. We analyze the contribution of using a temporal window by keeping the same network architecture as DBN, but replicating the central reference frame 5 times instead of inputing a stack of neighboring frames, and retrain the network with the same hyperparameters. We call this approach DBN+SINGLE. Qualitative comparisons are shown in Fig. 4 and 6, and quantitative results are shown in Table 2 and Fig. 5. We can see that using neighboring frames greatly improves the quality of the results. We chose a 5 frame window as it provides a good compromise between result quality and training time [16]. *Single-image* methods are also provided as reference: PSDEBLUR for blind uniform deblurring with off-the-shelf shake reduction software in Photoshop, and [52] for non-uniform comparisons.

Effects of alignment. In this set of experiments, we analyze the impact of input image alignment in the output restoration quality, namely we compare the results of DBN+NOALIGN, DBN+HOMOG., and DBN+FLOW. See

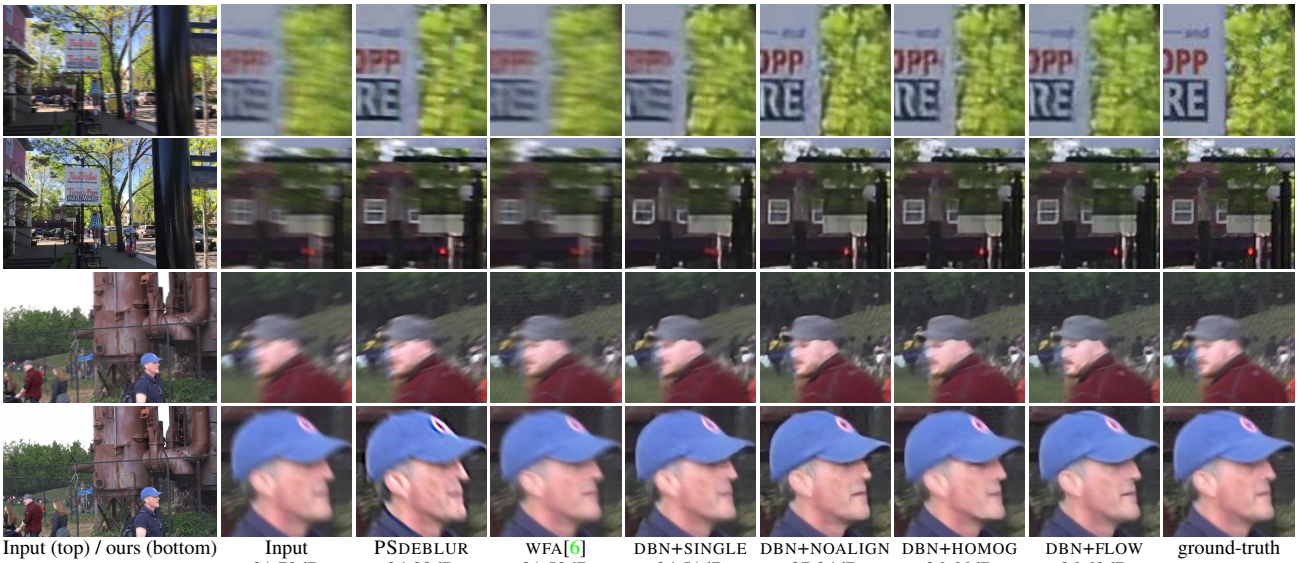

Figure 4: Quantitative results from our test set, with PSNRs relative to the ground truth. Here we compare DBN with a *single-image* approach, PSDEBLUR, and a start-of-the-art multi-frame video deblurring method, WFA [6]. DBN, achieves comparable results to [6] without alignment, and improved results with alignment.

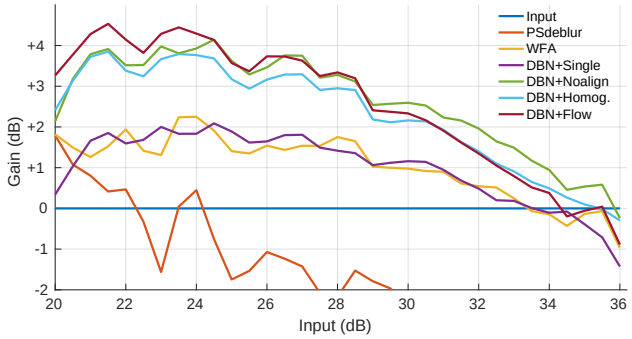

Figure 5: Quantitative comparison of different approaches. In this plot, the PSNR gain of applying different methods and configurations is plotted versus the sharpness of the input frame. We observe that all multi-frame methods provide a quality improvement for blurry input frames, with diminishing improvements as the input frames get sharper. DBN+NOALIGN and DBN+FLOW perform the best, but qualitatively, DBN+FLOW and DBN+HOMOG. are often comparable, and superior to no alignment. We provide a *single-image* uniform blur kernel deblurring method as reference (PSDEBLUR).

Tab. 2 and Fig. 5 for quantitative comparisons, and the qualitative comparison in Fig. 6. Our main conclusions are that *DeBlurNet* with optical flow and homography are often qualitatively equivalent, and DBN+FLOW often has higher PSNR. On the other hand, DBN+NOALIGN performs even better than DBN+FLOW and DBN+HOMOG in terms of PSNR, especially when the input frames are not too blurry, e.g. >29dB. However, we observe that DBN+FLOW fails gracefully when inputs frame are much blurrier, which leads to a drop in PSNR and MSSIM (see Tab. 2 and Fig. 5). In this case, DBN+FLOW and DBN+HOMOG. perform better. One possible explanation for this is that when the input quality is good, optical flow errors will dominate the final performance of the deblurring procedure. Indeed, sequences with high input PSNR have small relative motion (consequence of how the dataset is created) so there is not too much displacement from one frame to the next, and DBN+NOALIGN is able to directly handle the input frames without any alignment.

Comparisons to existing approaches. We compare our method to existing approaches in Fig. 6. Specifically, we show a quantitative comparison to WFA [6], and qualitative comparisons to Cho et al. [4], Kim et al. [18], and WFA [6]. We also compare to *single image* deblurring methods, Chakrabarti [2], Xu et al. [52], and the Shake Reduction feature in Photoshop CC 2015 (PSDEBLUR). We note that PSDEBLUR can cause ringing artifacts when used in an automatic setting on sharp images, resulting in a sharp degradation in quality (Fig. 5). The results of [4] and [18] are the ones provided by the authors, WFA [6] was applied a single iteration with the same temporal window, and for [52, 2] we use the implementations provided by the authors. Due to the large number of frames, we are only able to

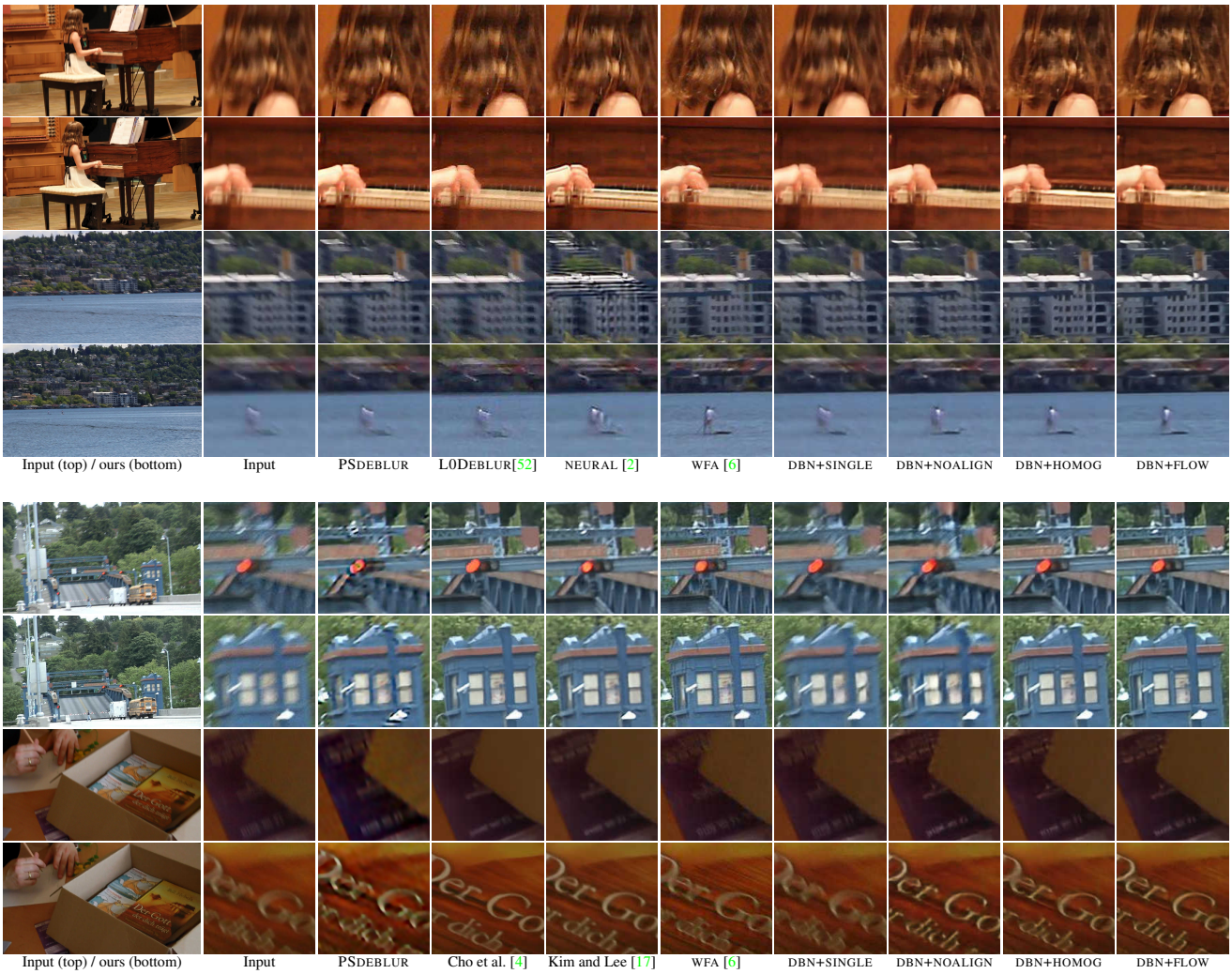

Figure 6: Qualitative comparisons to existing approaches. We compare DBN under various alignment configurations, with prior approaches, e.g. Cho et al. [4], Kim and Lee [17], Chakrabarti [2], Xu et al. [52], WFA [6], and Photoshop CC Shake Reduction. In general DBN achieves decent quality without alignment, and is comparable or better when simpler frame-wise homography is applied. Note that [4] adapts homography-based motion model, while [6] and [17] are estimating the optical flow for alignment.

compare quantitatively to approaches which operate sufficiently fast, which excludes many non-uniform deconvolution based methods. The complete sequences are given in the supplementary material. It is important to note that the test images have not been seen during the training procedure, and many of them have been shot by other cameras. Our conclusion is that DBN often produces superior quality deblurred frames, even when the input frames are aligned with a global homography, which requires substantially less computation than prior methods.

Generalization to other types of videos. As discussed in Sec. 4, our training set has some limitations. Despite these, Fig. 7 shows that our method can generalize well to other types of scenes not seen during training. This includes videos captured in indoor, low-light scenarios and motion blur originating from an object moving, rather than the temporally uncorrelated blur from camera shake. While our dataset has instances of motion blur in it, it is dominated by camera-shake blur. Nonetheless, the network is able to produce a moderate amount of object motion deblurring as well, which is not handled by other lucky imaging approaches.

Other experiments. We tested with different fusion strategies, for example late fusion, i.e. aggregating features from deeper layers after high-level image content has been extracted from each frame, with both shared and non-

Figure 7: Our proposed method can generalize to types of data not seen in the training set. The first example shows a low-light, noisy video, and the second shows an example with motion blur, rather than camera shake. The biker is in motion, and is blurred in all frames in the stack, but the network can still perform some moderate deblurring.

shared weights. Experimental results show that this produced slightly worse PSNR and training and validation loss, but it occasionally helped in challenging cases where DBN+NOALIGN fails. However this improvement is not consistent, so we left it out of our proposed approach.

Multi-scale phase-based methods have proven to be able to generate sharp images using purely Eulerian representations [31], so we experimented with multiscale-supervised, Laplacian reconstructions, but found similarly inconclusive results. While the added supervision helps in some cases, it likely restricts the network from learning useful feature maps that help in other frames.

We also tried directly predicting the sharp Fourier coefficients, as in [5], however this approach did not work as well as directly predicting output pixels. One possible reason is that the image quality is more prone to reconstruction errors of Fourier coefficients, and we have not found a robust way to normalize the scale of Fourier coefficients during training, compared with the straightforward way of applying Sigmoid layers when inputs are in the spatial domain.

Visualization of learned filters. Here we visualize some filters learned from DBN+FLOW, specifically at F0, to gain some insights of how it deblurs an input stack. It can be observed that DBN not only learns to locate the corresponding color channels to generate the correct tone (Fig. 8, left), but is also able to extract edges of different orientations (Fig. 8, middle), and to locate the warping artifacts (Fig. 8, right).

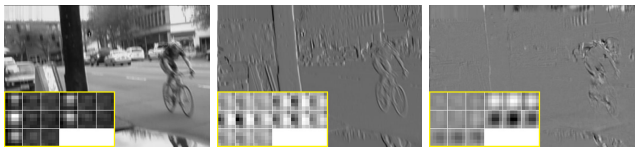

Figure 8: Here we selectively visualize 3 out of 64 filters (highlighted) and their response at F0 from DBN+FLOW.

Limitations. One limitation of this work is that we address only a subset of the types of blur present in video, in particular we focus on motion blur that arises due to camera-shake from hand-held camera motion. In practice, our dataset contains all types of blur that can be reduced by a shorter exposure time, including object motion, but this type of motion occurs much less frequently. Explicitly investigating other sources of blur, for example focus and object motion, which would require different input and training data, is an interesting area for future work.

Although no temporal coherence is explicitly imposed and no post-processing is done, the processed sequences are in general temporally smooth. We refer the reader to the video provided in the supplementary material. However, when images are severely blurred, our proposed model, especially DBN+NOALIGN, can introduce temporal artifacts that becomes more visible after stabilization. In the future, we plan to investigate better strategies to handle unaligned cases, for example through the multi-scale reconstruction [10, 1].

We would like also to augment our training set with a wider range of videos, as this should increase general applicability of the proposed approach.

6. Conclusion

We have presented a learning-based approach to multi-image video deblurring. Despite the above limitations, our method generates results that are often as good as or superior to the state-of-the-art approaches, with no parameter tuning and without the explicit need for challenging image alignment. It is also highly efficient due to the relaxation of the quality of alignment required – using a simplified alignment method, our approach can generate high quality results within a second, which is substantially faster than existing approaches many of which take minutes per frame.

In addition, we conducted a number of experiments showing the quality of results varying the input requirements. We believe that similar strategies could be applied to other aggregation based applications.

Acknowledgement

This work was supported in part by Adobe Research and the Baseline Funding of KAUST.

References

[1] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *European Conference on Computer Vision*, pages 354–370. Springer, 2016. 8

[2] A. Chakrabarti. A neural approach to blind motion deblurring. *arXiv preprint arXiv:1603.04771*, 2016. 3, 4, 6, 7

[3] S. Cho and S. Lee. Fast motion deblurring. *ACM Trans. Graph.*, 28(5):145:1–145:8, 2009. 2

[4] S. Cho, J. Wang, and S. Lee. Video deblurring for hand-held cameras using patch-based synthesis. *ACM Trans. Graph.*, 31(4):64, 2012. 1, 2, 4, 6, 7

[5] M. Delbracio and G. Sapiro. Burst deblurring: Removing camera shake through fourier burst accumulation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015. 2, 8

[6] M. Delbracio and G. Sapiro. Hand-held video deblurring via efficient fourier aggregation. *IEEE Trans. Comp. Imag.*, 1(4):270–283, 2015. 1, 2, 4, 5, 6, 7

[7] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, 2014. 2

[8] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graph.*, 25(3):787–794, 2006. 2

[9] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. *arXiv preprint arXiv:1504.06852*, 2015. 4

[10] G. Ghiasi and C. C. Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, 2016. 8

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 3

[12] Y. Huang, W. Wang, and L. Wang. Bidirectional recurrent convolutional networks for multi-frame super-resolution. In *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2015. 2

[13] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Trans. Graph.*, 35(4):110, 2016. 2

[14] A. Ito, A. C. Sankaranarayanan, A. Veeraraghavan, and R. G. Baraniuk. Blurburst: Removing blur due to camera shake using multiple images. *ACM Trans. Graph.*, Submitted. 2

[15] N. Joshi and M. F. Cohen. Seeing Mt. Rainier: Lucky imaging for multi-image denoising, sharpening, and haze removal. In *Proc. IEEE Int. Conf. Comput. Photogr. (ICCP)*, 2010. 2

[16] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos. Video super-resolution with convolutional neural networks. *IEEE Trans. Comp. Imag.*, 2(2):109–122, 2016. 2, 5

[17] T. H. Kim and K. M. Lee. Generalized video deblurring for dynamic scenes. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015. 2, 4, 7

[18] T. H. Kim, S. Nah, and K. M. Lee. Dynamic scene deblurring using a locally adaptive linear blur model. *arXiv preprint arXiv:1603.04265*, 2016. 5, 6

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

[20] F. Klose, O. Wang, J.-C. Bazin, M. Magnor, and A. Sorkine-Hornung. Sampling based scene-space video processing. *ACM Trans. Graph.*, 34(4):67, 2015. 2

[21] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, 2012. 5

[22] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2011. 2

[23] D. Kundur and D. Hatzinakos. Blind image deconvolution. *IEEE Signal Process. Mag.*, 13(3):43–64, 1996. 2

[24] N. Law, C. Mackay, and J. Baldwin. Lucky imaging: high angular resolution imaging in the visible from the ground. *Astron. Astrophys.*, 446(2):739–745, 2006. 2

[25] Y. Li, S. B. Kang, N. Joshi, S. M. Seitz, and D. P. Huttenlocher. Generating sharp panoramas from motion-blurred videos. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2010. 2

[26] C. Liu and W. T. Freeman. A high-quality video denoising algorithm based on reliable motion estimation. In *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, 2010. 2

[27] D. Liu, Z. Wang, B. Wen, J. Yang, W. Han, and T. S. Huang. Robust single image super-resolution via deep networks with sparse prior. *IEEE Trans. Image Proc.*, 25(7):3194–3207, 2016. 2

[28] Z. Liu, L. Yuan, X. Tang, M. Uyttendaele, and J. Sun. Fast burst images denoising. *ACM Trans. Graph.*, 33(6):232, 2014. 2

[29] X.-J. Mao, C. Shen, and Y.-B. Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. *arXiv preprint arXiv:1603.09056*, 2016. 3

[30] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum. Full-frame video stabilization with motion inpainting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(7):1150–1163, July 2006. 2

[31] S. Meyer, O. Wang, H. Zimmer, M. Grosse, and A. Sorkine-Hornung. Phase-based frame interpolation for video. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015. 8

[32] T. Michaeli and M. Irani. Blind deblurring using internal patch recurrence. In *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, 2014. 2

[33] C. Paramanand and A. N. Rajagopalan. Non-uniform motion deblurring for bilayer scenes. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2013. 2

[34] S. H. Park and M. Levoy. Gyro-based multi-image deconvolution for removing handshake blur. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2014. 2

[35] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. *arXiv preprint arXiv:1604.07379*, 2016. 2, 3

[36] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph.*, 23(3):664–672, 2004. 2

[37] J. Sánchez Pérez, E. Meinhardt-Llopis, and G. Facciolo. TV-L1 Optical Flow Estimation. *J. Image Proc. On Line (IPOL)*, 3:137–150, 2013. 4

[38] A. Sellent, C. Rother, and S. Roth. Stereo video deblurring. In *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, 2016. 2

[39] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Trans. Graph.*, 27(3), 2008. 2

[40] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016. 2

[41] E. Simo-Serra, S. Iizuka, K. Sasaki, and H. Ishikawa. Learning to simplify: fully convolutional networks for rough sketch cleanup. *ACM Trans. Graph.*, 35(4):121, 2016. 3

[42] S. Su and W. Heidrich. Rolling shutter motion deblurring. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1529–1537. IEEE, 2015. 2

[43] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015. 3

[44] Y.-W. Tai, H. Du, M. S. Brown, and S. Lin. Image/video deblurring using a hybrid camera. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2008. 2

[45] J. Telleen, A. Sullivan, J. Yee, O. Wang, P. Gunawardane, I. Collins, and J. Davis. Synthetic shutter speed imaging. *Comput. Graph. Forum*, 26(3):591–598, 2007. 5

[46] P. H. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *Comput. Vis. Image Und.*, 78(1):138–156, 2000. 4

[47] R. Wang and D. Tao. Recent progress in image deblurring. *arXiv preprint arXiv:1409.6838*, 2014. 2

[48] P. Wieschollek, B. Schölkopf, H. Lensch, and M. Hirsch. End-to-end learning for image burst deblurring. In *Computer Vision - ACCV 2016 - 13th Asian Conference on Computer Vision*, 2016. 3

[49] J. Wulff and M. J. Black. Modeling blurred video with layers. In *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, 2014. 2

[50] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012. 2

[51] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *Proc. IEEE Eur. Conf. Comput. Vis. (ECCV)*, 2010. 2

[52] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2013. 2, 5, 6, 7

[53] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Image deblurring with blurred/noisy image pairs. *ACM Trans. Graph.*, 26(3):1, 2007. 2