Online Video Deblurring via Dynamic Temporal Blending Network

Tae Hyun Kim[1], Kyoung Mu Lee[2], Bernhard Schölkopf[1] and Michael Hirsch[1]

[1]Department of Empirical Inference, Max Planck Institute for Intelligent Systems
[2]Department of ECE, ASRI, Seoul National University
{tkim,bernhard.schoelkopf,michael.hirsch}@tuebingen.mpg.de, kyoungmu@snu.ac.kr

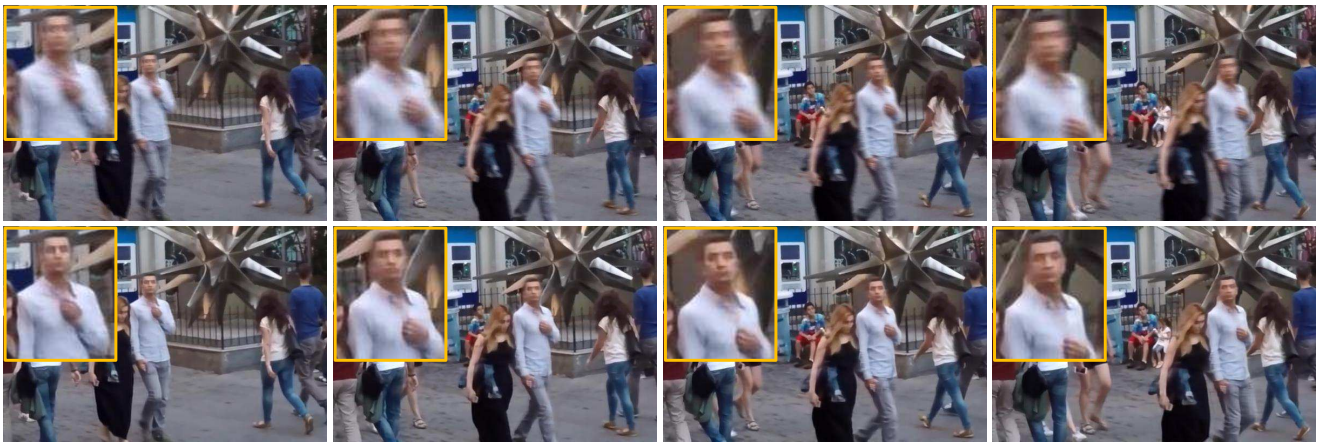

Figure 1: Our online deblurring results (bottom) on a number of challenging real-world video frames (top) suffering from strong object motion. Our proposed approach is able to process the input video (VGA) in real-time, i.e. ∼24 fps on a standard graphics card (NVIDIA GTX 1080).

Abstract

State-of-the-art video deblurring methods are capable of removing non-uniform blur caused by unwanted camera shake and/or object motion in dynamic scenes. However, most existing methods are based on batch processing and thus need access to all recorded frames, rendering them computationally demanding and time-consuming and thus limiting their practical use. In contrast, we propose an online (sequential) video deblurring method based on a spatio-temporal recurrent network that allows for real-time performance. In particular, we introduce a novel architecture which extends the receptive field while keeping the overall size of the network small to enable fast execution. In doing so, our network is able to remove even large blur caused by strong camera shake and/or fast moving objects. Furthermore, we propose a novel network layer that enforces temporal consistency between consecutive frames by dynamic temporal blending which compares and adaptively (at test time) shares features obtained at different time steps. We show the superiority of the proposed method in an extensive experimental evaluation.

1. Introduction

Moving objects in dynamic scenes as well as camera shake can cause undesirable motion blur in video recordings, often implying a severe degradation of video quality. This is especially true for low-light situations where the exposure time of each frame is increased, and for videos recorded with action (hand-held) cameras that have enjoyed widespread popularity in recent years. Therefore, not only to improve video quality [6, 18] but also to facilitate other vision tasks such as tracking [16], SLAM [21], and dense 3D reconstruction [22], video deblurring techniques and their applications have seen an ever increasing interest recently. However, removing motion blur and restoring sharp

frames in a blind manner (i.e., without knowing the blur of each frame) is a highly ill-posed problem and an active research topic in the field of computational photography.

In this paper, we propose a novel discriminative video deblurring method. Our method leverages recent insights within the field of deep learning and proposes a novel neural network architecture that enables run-times which are orders of magnitude faster than previous methods without significantly sacrificing restoration quality. Furthermore, our approach is the first online (sequential) video deblurring technique that is able to remove general motion blur stemming from both egomotion and object motion in real-time (for VGA video resolution).

Our novel network architecture employs deep convolutional residual networks [12] with a layout that is recurrent both in time and space. For temporal sequence modeling, we propose a network layer that implements a novel mechanism that we dub *dynamic temporal blending*, which compares the feature representation at consecutive time steps and allows for dynamic (i.e. input-dependent) pixel-specific information propagation. Recurrence in the spatial domain is implemented through a novel network layout that is able to extend the spatial receptive field over time without increasing the size of the network. In doing so, we can handle large blurs better than typical networks for video frames, without run-time overhead.

Due to the lack of publicly available training data for video deblurring, we also have collected a large number of blurry and sharp videos by adapting the work of Kim et al. [19] and the recent work of Nah et al. [26]. Specifically, we recorded sharp frames using a high-speed camera and generated realistic blurry frames by averaging over several consecutive sharp frames. Using this new dataset, we successfully trained our novel video deblurring network in an end-to-end manner.

Using the proposed network and new dataset, we perform deblurring in a sequential manner, in contrast to many previous methods that require access to all frames, while at the same time being much faster than existing state-of-the-art video deblurring methods. In the experimental section, we demonstrate the performance of our proposed model on a number of challenging real-world videos capturing dynamic scenes such as the one shown in Fig. 1, and illustrate the superiority of our method in a comprehensive comparison with the state of the art, both qualitatively and quantitatively. In particular, we make the following contributions:

- we present, to the best of our knowledge, the first discriminative learning approach to real-time video deblurring which is capable of removing spatially varying motion blurs in a sequential manner
- we introduce a novel spatio-temporal recurrent architecture with small computational footprint and increased receptive field along with a dynamic temporal blending mechanism that enables adaptive information propagation during test time
- we generate a large-scale high-speed video dataset that enables discriminative learning
- we show promising results on a wide range of challenging real-world video sequences

2. Related Work

Multi-frame Deblurring. Early attempts to handle motion blur caused by camera shake considered multiple blurry images [27, 4], and adapted techniques for removing uniform blur in single blurry images [10, 30]. Other works include Cai et al [2], and Zhang et al [38] which obtained sharp frames by exploiting the sparsity of the blur kernels and gradient distribution of the latent frames. More recently, Delbracio and Sapiro [8] proposed Fourier Burst Accumulation (FBA) for burst deblurring, an efficient method to combine multiple blurry images without explicit kernel estimation by averaging complex pixel coefficients of multiple observations in the Fourier domain. Wieschollek et al. [35] extended the work with a recent neural network approach for single image blind deconvolution [3], and achieved promising results by training the network in an end-to-end manner.

Most of the afore-mentioned methods assume stationarity, i.e., shift invariant blur, and cannot handle the more challenging case of spatially varying blur. To deal with spatially varying blur, often caused by rotational camera motion (roll) around the optical axis [34, 11, 14], additional non-trivial alignment of multiple images is required. Several methods have been proposed to simultaneously solve the alignment and restoration problem [5, 37, 39]. In particular, Li et al. [24] proposed a method to jointly perform camera motion (global motion) estimation and multi-frame deblurring, in contrast to previous methods that estimate a single latent image from multiple frames.

Video Deblurring. Despite some of these methods being able to handle non-uniform blur caused by camera shake, none of them is able to remove spatially-varying blur stemming from object motion in a video recording of a dynamic scene. More generally, blur in a typical video might originate from various sources including moving objects, camera shake, and depth variation, and thus it is required to estimate pixel-wise different blur kernels which is a highly intricate problem.

Some early approaches make use of sharp "lucky" frames which sometimes exist in long videos. Matsushita et al. [25] detected sharp frames using image statistics, performed global image registration and transferred pixel intensities from neighboring sharp frames to blurry ones in order to remove blur. Cho et al. [6] improved deblurring quality significantly by employing additional local search and a blur model for aligning differently blurred image re-

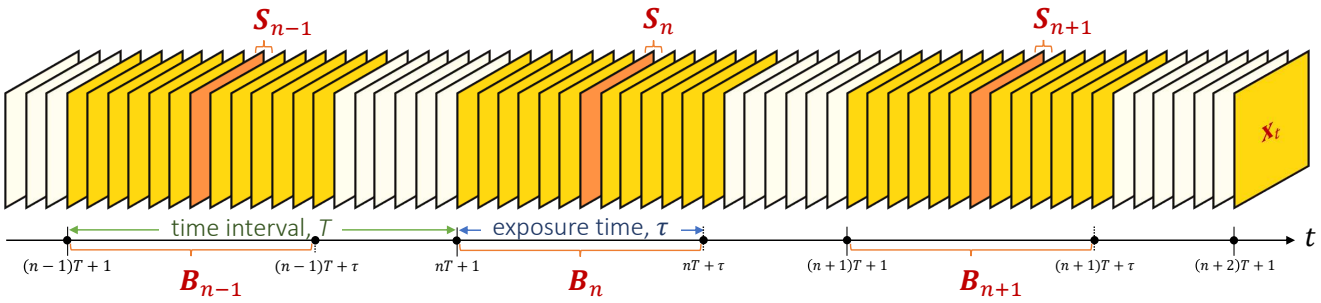

Figure 2: Generation of our blur dataset $\{\mathbf{S}_n, \mathbf{B}_n\}$ by averaging neighboring frames from a high-speed video $\{\mathbf{X}_{nT}\}$.

gions. However, these exemplar-based methods still have some limitations in treating distinct blurs by fast moving objects due to the difficulty of accurately finding corresponding points between severely blurred objects and the sharp reference one.

Other deblurring attempts segment differently blurred regions. Both Levin [23] and Bar et al. [1] automatically segmented a motion blurred object in the foreground from a (constant) background, and assumed a uniform motion blur model in the foreground region. Wulff and Black [36] considered differently blurred bi-layered scenes and estimated segment-wise accurate blur kernels by constraining those through a temporally consistent affine motion model. While they achieved impressive results especially at the motion boundaries, extending and generalizing their model to handle multi-layered scenes in real situations are difficult as we do not know the number and depth ordering of the layers in advance.

In contrast, there are some recent works that estimate pixel-wise varying kernels directly without segmentation. Kim and Lee [17] proposed a method to parametrize pixel-wise varying kernels with motion flows in a single image, and they naturally extended it to deal with blurs in videos [18]. Delbracio and Sapiro [9] also employed bi-directional optical flows for pixel-wise registration of consecutive frames, however, managed to keep processing time low by using their fast FBA [8] method for local blur removal. Recently, Sellent et al. [29] tackled independent object motions with local homographies, and their adaptive boundary handling rendered promising results with stereo video datasets. Although these methods are applicable to remove general motion blurs, they are rather time-consuming due to optical flow estimation and/or pixel-wise varying kernel estimation. Probably the closest approach related to our method is the concurrent work of Su et al. [31], which trains a CNN to remove blur stemming from both ego and object motions. In a comprehensive comparison, we show the merits of our novel network architecture both in terms of computation time as well as restoration quality.

3. Training Datasets

A key factor for the recent success of deep learning in computer vision is the availability of large amounts of training data. However, the situation is more tricky for the task of blind deblurring. Previous learning-based single-image blind deconvolution [3, 28, 32] and burst deblurring [35] approaches have considered only ego motion and assumed a uniform blur model. However, adapting these techniques to the case of spatially and temporally varying motion blurs caused by both ego motion and object motion is not straightforward. Therefore, we pursue a different strategy and employ recently proposed techniques [19, 31, 26] that generate pairs of sharp and blurry videos using high-speed cameras.

Given a high-speed video, we "simulate" long shutter times by averaging several consecutive short-exposure images, thereby synthesizing a video with fewer longer-exposed frames. The rendered (averaged) frames are likely to feature motion blur which might arise from camera shake and/or object motion. At the same time, we use the center short-exposure image as a reference sharp frame. We thus have,

$$\begin{cases} \mathbf{B}_n = \frac{1}{\tau} \sum_{j=0}^{\tau} \mathbf{X}_{nT+j} \\ \mathbf{S}_n = \mathbf{X}_{nT+\lceil \frac{\tau}{2} \rceil} \end{cases}, \tag{1}$$

where n denotes the time step, and $\{\mathbf{X}_{nT}\}$, $\mathbf{B}_n$, and $\mathbf{S}_n$ are the short-exposure frames (high-speed video), synthesized blurry frame, and reference sharp frame respectively. A parameter τ corresponds to the effective shutter speed which determines the number of frames to be averaged. A time interval T, which satisfies $T \geq \tau$ controls the frame rate of the synthesized video. For example, the frame rate of the generated video is $\frac{f}{T}$ for a high-speed video captured at a frame rate f. Note that with these datasets, we can handle motion blurs only, but not other blurs (e.g., defocus blur). We can control the strength of the blurs by adjusting τ (a larger τ generates more blurry videos), and can also change the duty cycle of the generated video by controlling the time interval T. The whole process is visualized in Fig. 2.

For our experiments, we collected high-speed sharp frames using a GoProHERO4 BLACK camera which supports recording HD (1280x720) video at a speed of $f = 240$

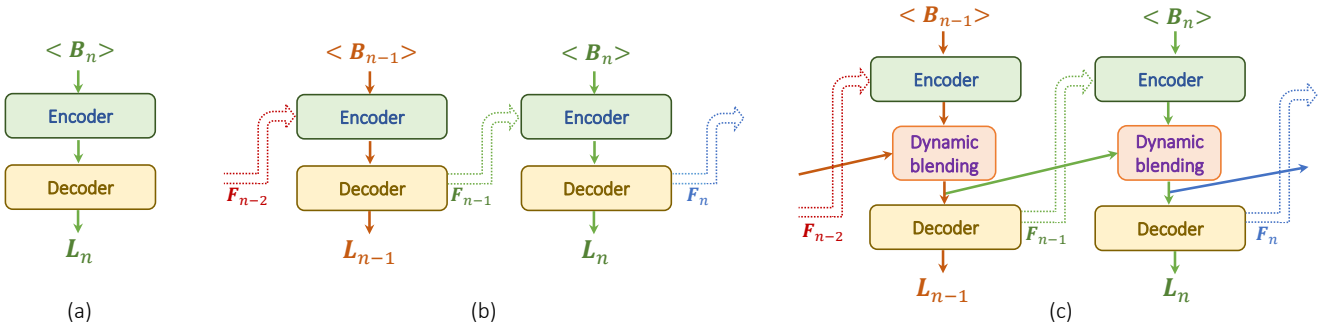

Figure 3: (a) Baseline model (CNN). (b) Spatio-temporal recurrent network (STRCNN). Feature maps at time step $(n-1)$ are added to the input of the network at time step n. (c) Spatio-temporal recurrent network with a proposed dynamic temporal blending layer (STRCNN+DTB). Intermediate feature maps are blended adaptively to render a clearer feature map by using weight map generated at runtime.

frames per second, and then downsampled frames to the resolution of 960x540 size to reduce noise and jpeg artifacts. To generate more realistic blurry frames, we carefully captured videos to have small motions (ideally less than 1 pixel) among high-speed sharp frames as suggested in [19]. Moreover, we randomly selected parameters as $\tau \in \{7, 9, 11, 13, 15\}$ and $\tau \leq T < 2\tau$ to generate various datasets with different frame rates, blur sizes, and duty cycles.

4. Method Overview

In this paper, using our large dataset of blurry and sharp video pairs, we propose a video deblurring network estimating the latent sharp frames from blurry ones. As suggested in the work of Su et al. [31], a straightforward and naive technique to deal with a video rather than a single image is employing a neural network repeatedly as shown in Fig. 3 (a). Here, input to the network are consecutive blurry frames $\langle \mathbf{B}_n \rangle_m = \{\mathbf{B}_{n-m}, \ldots, \mathbf{B}_{n+m}\}$ where $\mathbf{B}_n$ is the mid-frame and m some small positive integer[1]. The network predicts a single sharp frame $\mathbf{L}_n$ for time step n. In contrast, we present networks specialized for treating videos by exploiting temporal information and improve the deblurring performance drastically without increasing the number of parameters and the overall size of the networks.

In the present section, we introduce network architectures which we have found to improve the performance significantly. First, in Fig. 3 (b), we propose a spatio-temporal recurrent network which effectively extends the receptive field without increasing the number of parameters of the network, facilitating the removal of large blurs caused by severe motion. Next, in Fig. 3 (c), we additionally introduce a network architecture that implements our dynamic temporal blending mechanism which enforces temporal coherence between consecutive frames and further improves our spatio-temporal recurrent model. In the following, we describe our proposed network architectures in more detail.

4.1. Spatio-temporal recurrent network

A large receptive field is essential for a neural network being capable of handling large blurs. For example, it requires about 50 convolutional layers to handle blur kernels of a size of 101x101 pixels with conventional deep residual networks using 3x3 small filters [12, 13]. Although employing a deeper network and/or larger filters are a straightforward and an easy way to ensure large receptive field, the overall run-time does increase with the number of additional layers and increasing filter size. Therefore, we propose an effective network which retains large receptive field without increasing its depth and filter size, i.e. number of layers and therewith its number of parameters.

The architecture of the proposed spatio-temporal network in Fig. 3 (b) is based on conventional recurrent network [33], but has a point of distinction and profound difference. To be specific, we put $\mathbf{F}_{n-1}$ which is the feature map of multiple blurry input frames $\langle \mathbf{B}_{n-1} \rangle$ coupled with the previous feature map $\mathbf{F}_{n-2}$, as an additional input to our network together with blurry input frames $\langle \mathbf{B}_n \rangle$ at time step n. By doing so, at time step n, the feature of a single blurry frame $\mathbf{B}_n$ passes through the same network $(m+1)$ times, and ideally, we could increase the receptive field by the same factor without having to change the number of layers and parameters of our network. Notice that, in practice, the increase of receptive field is limited by the network capacity.

In other words, in a high dimensional feature space, each blurry input frame is recurrently processed multiple times by our recurrent network over time, thereby effectively experiencing a deeper spatial feature extraction with an increased receptive field. Moreover, further (temporal) information obtained from previous time steps is also transferred to enhance the current frame, thus we call such a network *spatio-temporal recurrent* or simply *STRCNN*.

[1] For simplicity we dropped index m from $\langle \mathbf{B}_n \rangle_m$.

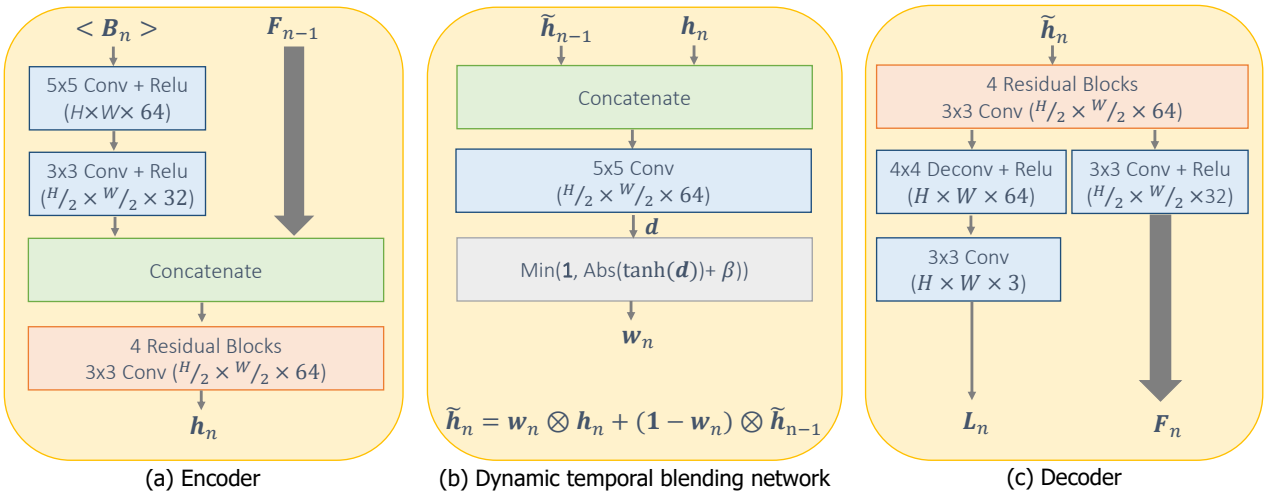

Figure 4: Detailed configurations of the proposed model. Our network is composed of encoder, dynamic temporal blending network, and decoder.

4.2. Dynamic temporal blending network

When handling video rather than a single frame, it is important to enforce temporal consistency. Although we recurrently transfer previous feature maps over time and implicitly share information between consecutive frames, we developed a novel mechanism for temporal information propagation that significantly improves the deblurring performance.

Motivated by the recent deep learning approaches of [7, 15] which dynamically adapt network parameters to input data at test time, we also generate weight parameters for temporal feature blending that encourages temporal consistency, as depicted in Fig. 3 (c). Specifically, based on our spatio-temporal recurrent network, we additionally propose a *dynamic temporal blending* network, which generates weight parameter $\mathbf{w}_n$ at time step n used for linear blending between the feature maps of consecutive time steps, i.e.

$$\tilde{\mathbf{h}}_n = \mathbf{w}_n \otimes \mathbf{h}_n + (\mathbf{1} - \mathbf{w}_n) \otimes \tilde{\mathbf{h}}_{n-1}, \tag{2}$$

where $\mathbf{h}_n$ denotes the feature map at current time step n, $\tilde{\mathbf{h}}_n$ denotes its filtered version, and $\tilde{\mathbf{h}}_{n-1}$ denotes the previously filtered feature map at time step $(n-1)$. Weight parameters $\mathbf{w}_n$ have a size equal to the size of $\mathbf{h}_n$, and have values between zero and one. As a linear operator $\otimes$ denotes element-wise multiplication, our filter parameter $\mathbf{w}_n$ can be viewed as a locally varying weight map. Notably, $\mathbf{h}_n$ is a feature activated in the middle of the entire network and thus it is different from $\mathbf{F}_n$ which denotes the final activation.

It is natural that the previously filtered (clean) feature map $\tilde{\mathbf{h}}_{n-1}$ is favored when $\mathbf{h}_n$ is a degraded but corresponding version of $\tilde{\mathbf{h}}_{n-1}$. Therefore we introduce a new cell which generates filter parameter $\mathbf{w}_n$ by comparing similarity between two feature maps, given by

$$\mathbf{w}_n = \min(\mathbf{1}, |\tanh(\mathbf{A}\tilde{\mathbf{h}}_{n-1} + \mathbf{B}\mathbf{h}_n)| + \beta)) \tag{3}$$

where $\tanh(.)$ denotes a hyperbolic tangent function, $\mathbf{A}$ and $\mathbf{B}$ correspond to linear (convolutional) filters. A trainable parameter $0 \leq \beta \leq 1$ denotes a bias value, and it controls the mixing rate, i.e. it satisfies $\mathbf{w}_n = \beta \cdot \mathbf{1}$ when the hyperbolic tangent function returns zero, and $\tilde{\mathbf{h}}_{n-1}$ is favored.

Notably, to this end, we could embed a temporal filtering unit into our deblurring network with only one single additional convolutional layer. By doing so, we could learn the way to generate weights (coefficients) of the temporal filter in an end-to-end manner, and do the filtering process in the high dimensional feature space rather than the image domain. Moreover, as the proposed module is shallow and light, ours still performs quickly with little computational overhead. We refer to this network as *STRCNN+DTB*.

5. Implementation and Training

In this section, we describe our proposed network architecture in full detail. An illustration is shown in Fig. 4, where we show a configuration at time step n only since our model shares all trainable parameters across time. Our network comprises three modules, i.e. encoder, dynamic temporal blending network, and decoder. Furthermore, we also discuss our objective function and training procedure.

5.1. Network architecture

5.1.1 Encoder

Fig. 4 (a) depicts the *encoder* of our proposed network. Input is $(2m+1)$ consecutive blurry frames $\langle \mathbf{B}_n \rangle$ where $\mathbf{B}_n$ is

the mid-frame, along with feature activations $\mathbf{F}_{n-1}$ from the previous stage. All input images are in color and range in intensity from 0 to 1. The feature map $\mathbf{F}_{n-1}$ is half the size of a single input image and has 32 channels. All blurry input images are filtered first, before being concatenated with the feature map and being fed into a deep residual network. Our encoder has a stack of 4 residual blocks (8 convolutional layers) similar to [12]. The output of our encoder is feature map $\mathbf{h}_n$.

5.1.2 Dynamic temporal blending

Our *dynamic temporal blending* network is illustrated in Figure 4 (b). It takes two concatenated feature maps $\tilde{\mathbf{h}}_{n-1}$ and $\mathbf{h}_n$ as input and estimates weight maps $\mathbf{w}_n$ through a convolutional layer with filters of size 5x5 pixels and a subsequent squashing function ($\tanh(.)$ and $Abs(.)$). Finally, the generated weight map $\mathbf{w}_n$ is used for blending between $\tilde{\mathbf{h}}_{n-1}$ and $\mathbf{h}_n$ according to Eq. 2.

5.1.3 Decoder

Input to our *decoder*, depicted in Fig. 4 (c), is the blended feature map $\tilde{\mathbf{h}}_n$ of the previous stage which is fed into a stack of 4 residual blocks (8 convolutional layers) with 64 convolutional filters of size 3x3 pixels. Outputs are a latent sharp frame $\mathbf{L}_n$ that corresponds to the blurry input frame $\mathbf{B}_n$, and a feature map $\mathbf{F}_n$. Notably, our output feature map $\mathbf{F}_n$ is handed over as input to the network at the next time step.

5.2. Objective function

As our final objective function, we use the mean squared error (MSE) between the latent frames and their corresponding sharp ground-truth frames, and regularize the network parameters to prevent overfitting, i.e.

$$\mathbf{E} = \frac{1}{N_{mse}} \sum_{n} ||\mathbf{S}_n - \mathbf{L}_n||^2 + \lambda \|\mathbf{W}\|^2, \qquad (4)$$

where N_{mse} denotes the number of pixels in a latent frame, and $\mathbf{W}$ denotes the trainable network parameters. User parameter λ trades off the data fidelity and regularization terms. In all our experiments we set λ to 10^{-5}.

5.3. Training parameters

For training, we randomly select 13 consecutive blurry frames from artificially blurred videos (i.e., $\mathbf{B}_1, \ldots, \mathbf{B}_{13}$) , and crop a patch per frame. Each patch is 128x128 pixels in size, and a randomly chosen pixel location is used for cropping all 13 patches. Moreover, we use a batch size of 8. For optimization, we employ Adam [20] with an initial learning rate of 0.0001, and the learning rate exponentially decays every 10k steps with a base of 0.96. We trained the network for 300k iterations, and the training takes about 3 days on a NVIDIA GTX 1080 graphic card.

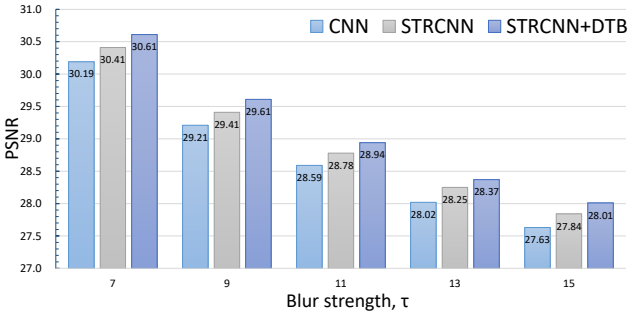

Figure 5: Performance comparisons among models proposed in Sec. 4 in terms of PSNR for varying blur strength.

6. Experiments

6.1. Model comparison

We study the three different network architectures that we discussed in Sec. 4, and evaluate deblurring quality in terms of peak signal-to-noise ratio (PSNR). For the fair comparison, we use the same number of network parameters, except for one additional convolutional layer that is required in the dynamic temporal blending network. We use our dataset (described in Sec. 3) for training, and use the dataset of [31] for evaluation at test time.

First, we compare the PSNR values of the three different models for varying blur strength by changing the effective shutter speed τ in Eq. (1). We take three consecutive blurry frames as input to the networks. As shown in Fig. 5, our STRCNN+DTB model shows consistently better results for all blur sizes. On average, the PSNR value of our STRCNN is 0.21dB higher than the baseline (CNN) model, and STRCNN+DTB achieves a gain of 0.38dB against the baseline.

Next, in Table 1, we evaluate and compare the performance of the models with a varying number of input blurry frames. Our STRCNN+DTB model outperforms other networks for all input settings. We choose STRCNN+DTB using three input frames ($m = 1$) as our final model.

6.2. Quantitative results

For objective evaluations, we compare with the state-of-the-art video deblurring methods [18, 31] whose source codes are available at the time of submission. In particular, as Shuochen et al. [31] provide their fully trained network parameters with three different input alignment methods. Specifically, they align input images with optical flow (FLOW), or homography (HOMOG.), and they also take raw inputs without alignment (NOALIGN). For fair comparisons, we train our STRCNN+DTB model with their dataset and evaluate performance with our own dataset.

We provide a quantitative comparison for 25 test videos

Number of blurry inputs	3	5	7
CNN	29.38	29.43	29.27
STRCNN	29.67	29.71	29.58
STRCNN+DTB	29.83	29.85	29.76

Table 1: PSNR values for a varying number of input frames. STRCNN+DTB model, which encompasses a dynamic blending network, shows consistently better results.

Method	PSNR [dB]	Run-time [Sec]
Kim and Lee. [18]	27.42	∼60k (cpu)
Cho et al. [6]	-	∼6k (cpu)
Delbracio et al. [9]	-	∼1.5k (cpu)
Su et al. [31] (FLOW)	28.81	∼570 (cpu+gpu)
Su et al. [31] (HOMOG.)	28.09	∼160 (cpu+gpu)
Su et al. [31] (NOALIGN)	28.47	∼25 (gpu)
Ours	29.11	∼12.6 (gpu)

Table 2: Quantitative comparison with state-of-the-art video deblurring methods in terms of PSNR. A total of 25 (test) videos is used for evaluation. Moreover, execution times for processing 100 HD frames are given.

generated with our high-speed camera described in Sec.3. Our model outperforms the state-of-the-art methods in terms of PSNR as shown in Table. 2.

6.3. Qualitative results

To verify the generalization capabilities of our trained network, we provide qualitative results for a number of challenging videos. Fig. 7 shows a comparison with [18, 31] on challenging video clips. All these frames have spatially varying blurs caused by distinct object motion and/or rotational camera shake. In particular, blurry frames shown in the third and fourth rows are downloaded from YouTube, and thus contain high-level noise and severe encoding artifacts. Nevertheless, our method successfully restores the sharp frames especially at the motion boundaries in real-time. In the last row, the offline (batch) deblurring approach by Kim and Lee [18] shows the best result however at the cost of long computation times. On the other hand, our approach yields competitive results though orders of magnitudes faster.

6.4. Run time evaluations

At test time, our online approach can process VGA (640x480) video frames at ∼24 frames per second with a recent NVIDIA GTX 1080 graphics card, and HD (1280x720) frames at ∼8 frames per second. In contrast, other conventional (offline) video deblurring methods take much longer. In Table. 2, we compare run-times for processing 100 HD (1280x720) video frames. Notably, our proposed method runs at a much faster rate than other conventional methods.

Figure 6: Top to bottom: Consecutive blurry frames, deblurred results by STRCNN and STRCNN+DTB. Notably, the arrow indicates erroneous region by STRCNN model.

6.5. Effects of dynamic temporal blending

In Fig. 6, we show a qualitative comparison of the results obtained with STRCNN and STRCNN+DTB. Although STRCNN could also remove motion blur by camera shake in the blurry frames well, it causes some artifacts on the car window. In contrast, STRCNN+DTB successfully restores sharp frames with fewer artifacts by enforcing temporal consistency using the proposed dynamic temporal blending network.

7. Conclusion

In this work, we proposed a novel network architecture for discriminative video deblurring. To this end, we have acquired a large dataset of blurry/sharp video pairs for training and introduced a novel spatio-temporal recurrent network which enables near real-time performance by adding the feature activations of the last layer as an additional input to the network at the following time step. In doing so, we could retain large receptive field which is crucial to handle large blurs, without introducing a computational overhead. Furthermore, we proposed a dynamic blending network that enforces temporal consistency, which provides a considerable performance gain. We demonstrate the efficiency and superiority of our proposed method by intensive experiments on challenging real-world videos.

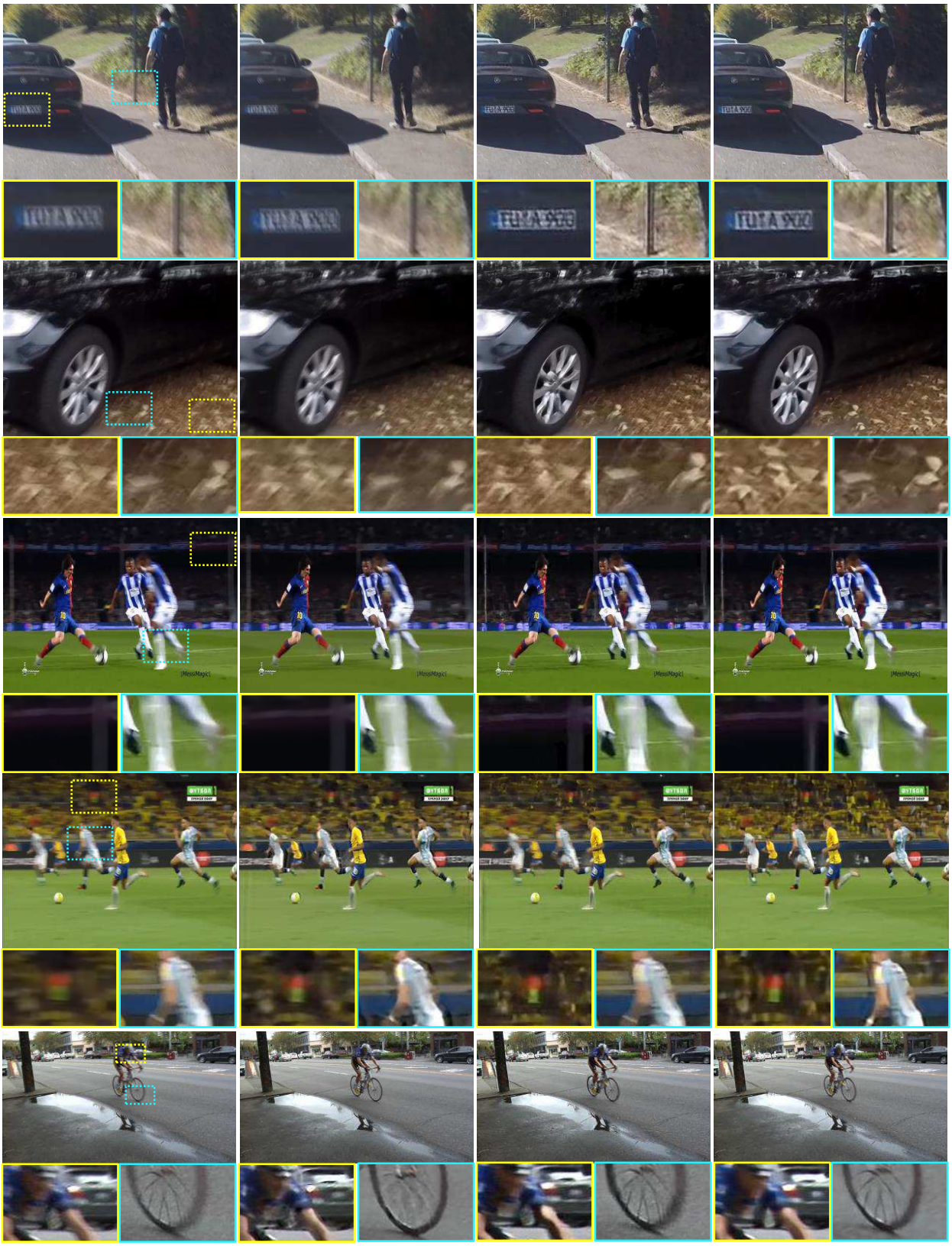

Figure 7: Left to right: Real blurry frames, Kim and Lee [18], Su et al. [31] (FLOW), and our deblurring results.

References

[1] L. Bar, B. Berkels, M. Rumpf, and G. Sapiro. A variational framework for simultaneous motion estimation and restoration of motion-blurred video. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007. 3

[2] J.-F. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring using multiple images. *Journal of computational physics*, 228(14):5057–5071, 2009. 2

[3] A. Chakrabarti. A neural approach to blind motion deblurring. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 2, 3

[4] J. Chen, L. Yuan, C.-K. Tang, and L. Quan. Robust dual motion deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 2

[5] S. Cho, H. Cho, Y.-W. Tai, and S. Lee. Registration based non-uniform motion deblurring. *Computer Graphics Forum*, 31(7):2183–2192, 2012. 2

[6] S. Cho, J. Wang, and S. Lee. Vdeo deblurring for hand-held cameras using patch-based synthesis. *ACM Transactions on Graphics (SIGGRAPH)*, 2012. 1, 2, 7

[7] B. De Brabandere, X. Jia, T. Tuytelaars, and L. Van Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 5

[8] M. Delbracio and G. Sapiro. Burst deblurring: Removing camera shake through fourier burst accumulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 3

[9] M. Delbracio and G. Sapiro. Hand-held video deblurring via efficient fourier aggregation. *IEEE Transactions on Computational Imaging*, 2015. 3, 7

[10] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics (SIGGRAPH)*, 2006. 2

[11] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010. 2

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 4, 6

[13] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 4

[14] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf. Fast removal of non-uniform camera shake. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. 2

[15] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 5

[16] H. Jin, P. Favaro, and R. Cipolla. Visual tracking in the presence of motion blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005. 1

[17] T. H. Kim and K. M. Lee. Segmentation-free dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 3

[18] T. H. Kim and K. M. Lee. Generalized video deblurring for dynamic scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 3, 6, 7, 8

[19] 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. 2, 3, 4

[20] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[21] H. S. Lee, J. Kwon, and K. M. Lee. Simultaneous localization, mapping and deblurring. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. 1

[22] H. S. Lee and K. M. Lee. Dense 3d reconstruction from severely blurred images using a single moving camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 1

[23] A. Levin. Blind motion deblurring using image statistics. In *Advances in Neural Information Processing Systems (NIPS)*, 2006. 3

[24] Y. Li, S. B. Kang, N. Joshi, S. M. Seitz, and D. P. Huttenlocher. Generating sharp panoramas from motion-blurred videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 2

[25] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum. Full-frame video stabilization with motion inpainting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(7):1150–1163, 2006. 2

[26] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3

[27] A. Rav-Acha and S. Peleg. Two motion-blurred images are better than one. *Pattern recognition letters*, 26(3):311–317, 2005. 2

[28] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schlkopf. Learning to deblur. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 38(7):1439–1451, 2016. 3

[29] A. Sellent, C. Rother, and S. Roth. Stereo video deblurring. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 3

[30] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Transactions on Graphics (SIGGRAPH)*, 2008. 2

[31] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang. Deep video deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3, 4, 6, 7, 8

[32] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[33] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan. Saliency detection with recurrent fully convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 4

[34] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *International Journal of Computer Vision*, 98(2):168–186, 2012. 2

[35] P. Wieschollek, B. Schölkopf, H. P. Lensch, and M. Hirsch. Burst deblurring: Removing camera shake through fourier burst accumulation. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2016. 2, 3

[36] J. Wulff and M. J. Black. Modeling blurred video with layers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. 3

[37] H. Zhang and L. Carin. Multi-shot imaging: joint alignment, deblurring and resolution-enhancement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[38] H. Zhang, D. Wipf, and Y. Zhang. Multi-image blind deblurring using a coupled adaptive sparse prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 2

[39] H. Zhang and J. Yang. Intra-frame deblurring by leveraging inter-frame camera motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2