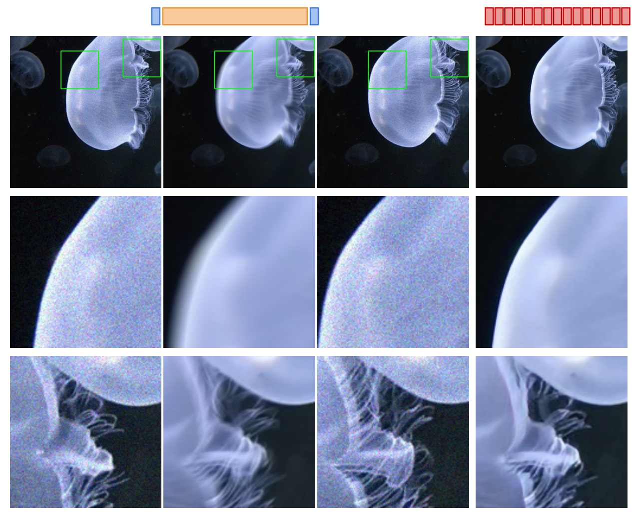
Photosequencing of Motion Blur using Short and Long Exposures

Vijay Rengarajan[1*], Shuo Zhao[1], Ruiwen Zhen[2], John Glotzbach[2], Hamid Sheikh[2], Aswin C. Sankaranarayanan[1]
[1]Carnegie Mellon University, [2]Samsung Research America
[*]vangarai@andrew.cmu.edu

Abstract

Photosequencing aims to transform a motion blurred image to a sequence of sharp images. This problem is challenging due to the inherent ambiguities in temporal ordering as well as the recovery of lost spatial textures due to blur. Adopting a computational photography approach, we propose to capture two short exposure images, along with the original blurred long exposure image to aid in the aforementioned challenges. Post-capture, we recover the sharp photosequence using a novel blur decomposition strategy that recursively splits the long exposure image into smaller exposure intervals. We validate the approach by capturing a variety of scenes with interesting motions using machine vision cameras programmed to capture short and long exposure sequences. Our experimental results show that the proposed method resolves both fast and fine motions better than prior works.



(a) Short exp. (b) Long exp. (c) Short exp. (d) Recovered
The whole sequence is shown in the supplementary material.

Figure 1. We capture three photographs in quick succession, a short-long-short 1ms-15ms-1ms exposure triplet (a-c), to capture both textures and fast motions. The short exposures (a,c) capture sharpness while being noisy. The long exposure (b) captures motion in the form of blur. (d) Our photosequencing method leverages the complementary information from these exposures to recover the scene with smooth motion and sharp textures.

1. Introduction

Capturing photosequences of fast events is a challenge in the photography milieu. High speed capture is costly to implement since it requires specialized electronics capable of handling high data bandwidth as well as highly sensitive image sensors that can make low noise measurements at short exposure periods. As a consequence, it is still a major obstacle in commodity cameras, especially without sacrificing the full spatial resolution of the sensor. This paper provides a pathway for the implementation of such a capability with no additional hardware.

One approach to obtain photosequences is from motion blur. A single motion-blurred photograph embeds motion events inherently, but recovering them can be hard due to ill-posed temporal ordering, as well as the erasure of spatial textures. While prior single-blur sequencing approaches by Jin et al. [10] and Purohit et al. [17] handle both these issues using data-driven priors encoded using deep neural networks, they are incapable of correctly sequencing multiple local motion events happening across the sensor space. The recent multi-blur sequencing technique of Jin et al. [9] handles the overall sequencing problem by processing multiple blurred photographs captured in succession. As much as texture recovery from blur is conditioned using deep neural networks, fine textures can still be lost due to fast motion since all of the captured photographs are blurred. An orthogonal approach to motion blur is to first capture a short-exposure photosequence at higher frame-rates followed by frame-wise denoising as done by Suo et al. [25]. Even though this looks promising, texture loss could still be an issue due to denoising. Further, static regions, which would have benefited from long exposures, suffer due to postprocessing which would otherwise be clean.

This paper proposes to capture a short-long-short exposure triplet to handle both motion ordering and texture recovery in photosequencing. The two additional short-exposure photographs are designed to have an exposure that is much smaller than the original long exposure photograph. Hence, while these additional images might be noisy, they are practically free of motion-blur. Further, the short exposure photographs resolve temporal ordering by acting as peripheral keypoints of the long exposure.

Once we capture the short-long-short exposure images, we computationally recover the photosequence equivalent to a high-speed capture of the underlying scene. The blurred image is first decomposed into two *half-blurred* images with reduced blur corresponding to the two halves of the exposure period and a single sharp image corresponding to the mid-point. We learn this decomposition by training a neural network using images synthesized from high framerate video datasets. A recursive decomposition leads to the sharp photosequence without any enforced temporal order. The recovered images gain from the complementary goodness of the short and long exposures for static and moving scene regions.

Figs. 1(a,b,c) show the short-long-short exposure images captured in succession of a scene with a moving jellyfish. The short exposures are noisy while the long exposure embeds motion blur. The intricate tentacles are captured in the noisy image but are blurred in the long exposure. In our recovered 15x photosequence corresponding to the long exposure duration, both texture and motion are recovered correctly in our reconstruction. The middle image of the photosequence is shown in Fig. 1(d). Most modern sensors, found in machine vision and cellphone cameras, can implement this exposure capture functionality. While we have shown our results on a machine vision camera, applying it to other cameras is a matter of having access to the correct API (which cellphone manufacturers invariably do).

The main contributions of this work are as follows:

1. We propose the novel idea of capturing short-long-short exposure images for the problem of photosequencing of motion blur.
2. We propose a recursive blur decomposition strategy in which increase in temporal super-resolution can be achieved through multiple levels of decomposition till the removal of blur.
3. We present a new short-long-short exposure dataset for a variety of dynamic events captured using a machine vision camera. We provide qualitative evaluation and comparisons to prior art on this data.

2. Related Works

Traditional deblurring approaches [2,3,11,16,24,27,29] aim to get a single sharp image along with local motion kernels and camera trajectories. Reconstructing the whole photosequence for combined object and camera motions has been dealt only recently [10, 17]. Single-blur techniques produce photosequence from a single image by imposing ordering constraints through explicit temporal image costs as done by Jin et al. [10] or through an implicit cost using a video encoder-decoder framework as done by Purohit et al. [17]. These methods suffer from ambiguities in temporal ordering of multiple objects. The recent approach by Jin et al. [9] does photosequencing using multiple motion-blurred images to impart more information for reconstruction and ordering. All these photosequencing approaches use neural network priors to reconstruct sharp textures from blur.

One can also increase framerate through temporal interpolation of video frames. Gradient and phase interpolation approaches are employed by Mahajan et al. [13] and Meyer et al. [14], respectively, while Shahar et al. [21] fuse recurring video patches to perform spatio-temporal super-resolution. Recently, the work by Jiang et al. [8] employed neural networks to linearly interpolate optical flows. However, frame interpolation methods are inherently limited from the assumption of video frames being non-blurred.

A hybrid high resolution, lower frame rate and a low resolution, higher frame rate camera setup is used in the works of Ben-Ezra and Nayar [1], and Tai et al. [26] to remove motion blur from the high resolution capture, but a further photosequencing step is not performed. The coded exposure work of Raskar et al. [18] suggests changing exposure design to improve the conditioning of the deblurring problem. The idea of capturing a noisy-blurry pair by Yuan et al. [28] regularizes deblurring better. While this work employs an additional short exposure image to produce a single sharp image, our goal is to recover the whole photosequence.

3. Problem

Let I_t be the clean ground-truth image of a scene at $t \in [0,1]$. Let $Y(a,b)$ be an observed image for the time interval $[a,b]$ defined as $Y(a,b) = 1/(b-a) \cdot \int_a^b I_t dt + n$ where n represents noise. The goal of motion-blur photosequencing is capture a blurred image $Y_{0\to1} = Y(0,1)$ and to estimate the images $\{\widehat{I}_{t_j}\}_{j=1}^{N}$ such that $Y_{0\to1} \approx \sum_{j=1}^{N} \widehat{I}_{t_j}/N$ where t_js are equi-spaced timepoints and N represents the *sequence-rate*. In essence, the recovered sequence represents the series of images of the scene if it had been captured with a hypothetical higher frame-rate camera operating at the same spatial resolution and without the noise level associated with that frame rate.

3.1. Approach

We propose to capture two short exposure images $Y_{0^-} = Y(-\Delta t, 0)$ and $Y_{1^+} = Y(1, 1+\Delta t)$, one before and one after the long exposure image $Y_{0\to1}$. Though the short exposure images will be noisy owing to the very short exposure interval Δt, they provide the much-needed information of scene texture. They also act as temporal endpoints for the

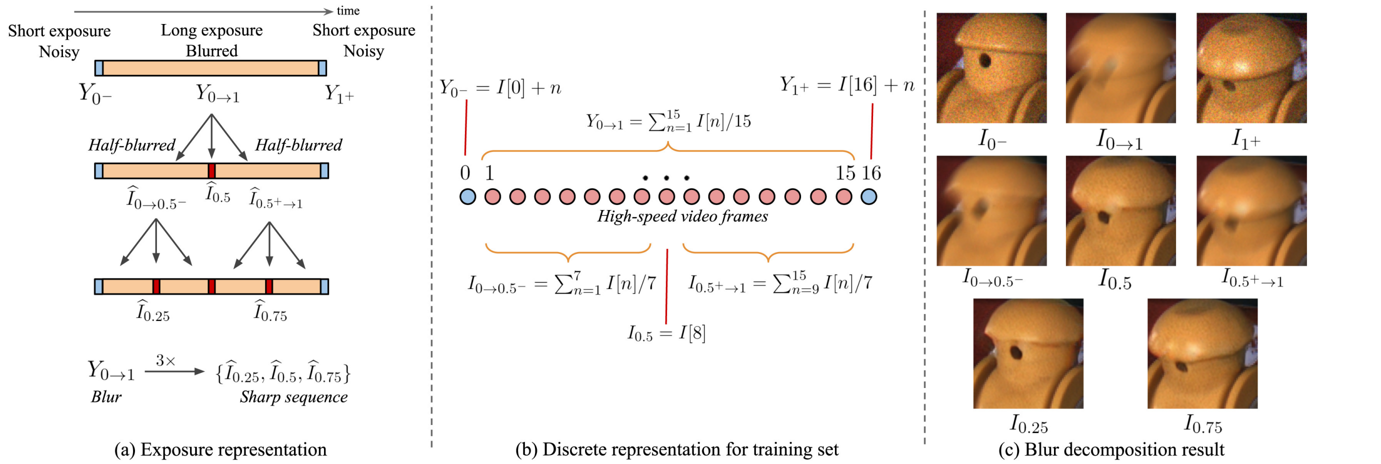


Figure 2. *Recursive blur decomposition till 3× recovery.* Our method takes short-long-short exposure triplet as inputs to produce a sharp sequence of images. Our core decomposition step is to split the blurred image into two half-blurred images and a midpoint sharp image. On recursive decomposition, we arrive at the desired sequence of sharp images. Two levels of decomposition, i.e. 3x recovery, is illustrated in (a). The discrete representation for preparing training set for an example of blurring 15 images is illustrated in (b). In (c), we show an example of our blur decomposition.

blurred image to disambiguate motion directions. Our problem statement is to estimate the image sequence $\{\widehat{I}_{t_j}\}$ happened during long exposure, given the three input images, $\{Y_{0^-}, Y_{0\to1}, Y_{1^+}\}$ as inputs.

3.2. Recursive Blur Decomposition

We adopt a multi-step sequencing strategy wherein we progressively increase the number of reconstructed sharp images. We first decompose the long exposure into two half-blurred images $\widehat{I}_{0\to0.5^-}$ and $\widehat{I}_{0.5^+\to1}$ and the sharp image $\widehat{I}_{0.5}$. Estimating just $\widehat{I}_{0.5}$ would correspond to deblurring. In addition, we also estimate $\widehat{I}_{0\to0.5^-}$ and $\widehat{I}_{0.5^+\to1}$ that contain lesser motion blur corresponding to each half of the original exposure interval. Our core blur decomposition step, hence, is the following:

$$\{Y_{0^-}, Y_{0\to1}, Y_{1^+}\} \to \{\widehat{I}_{0\to0.5^-}, \widehat{I}_{0.5}, \widehat{I}_{0.5^+\to1}\}. \quad (1)$$

Our idea then is to perform blur decomposition on both the half-blurred images to further split the blur interval and get a sharp image at their respective middle timepoints as shown in Fig. 2(a). The second-level will result in the sharp sequence $\widehat{I}_{0.25}, \widehat{I}_{0.5}, \widehat{I}_{0.75}$ corresponding to 3x frame-rate. We could perform blur decomposition recursively to achieve a desired sequence-rate; a k-level decomposition will provide $2^k - 1$ sharp images. In practice, one could stop the decomposition at a desired level when the blur in half-blurred images is negligible.

4. Implementation

We learn the blur decomposition mapping in Eq. (1) by training a neural network.

4.1. Network Architecture

We adopt an encoder-decoder architecture similar to U-net [7, 19] as shown in Fig. 3. We use dense residual blocks [30] which have rich local connections, instead of serialized convolutional layers. We use carry-on convolutional layers through the skip connections from encoder to decoder. The inputs are the short-long-short exposure observations $\{Y_{0^-}, Y_{0\to1}, Y_{1^+}\}$ and the outputs are the estimates of half-blurred images and the deblurred image, $\{\widehat{I}_{0\to0.5^-}, \widehat{I}_{0.5}, \widehat{I}_{0.5^+\to1}\}$. The input images pass through different initial convolutional layers; similarly the output layers are different for the half-blurred and deblurred images. The input layer of the short exposures share the same weights; the output layer of the half-blurred images share the same weights. The image intensities are in the range [0,1]. All the convolutional layers are followed by Leaky ReLU nonlinearity except for the last layer which is followed by rescaled-tanh to enforce outputs to [0,1]. All convolutional layers use 3x3 filters except for the first layers of both noisy and blurred images which use a filter size of 7x7. More details are provided in the supplementary material.

4.2. Training Data

Our goal is to have the neural network learn to decompose different amounts of blur in the long exposure image with the help of the noisy short exposure image. Training such a network requires a large number of blurred, half-blurred, noisy and clean images according to Eq. (1). Hence, to create training images, we follow the same process as existing photosequencing works [9, 10] that use high-speed video datasets – which we augment with

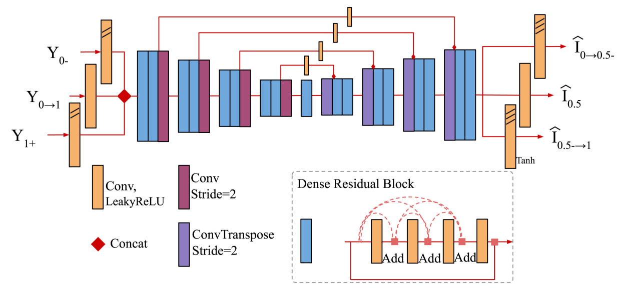


Figure 3. *Blur decomposition network.* We use a Unet-like architecture with dense residual blocks to provide rich local connections for encoder feature extraction and carry-on convolutions for encoder-to-decoder feature transfers.

a physically-accurate implementation of photon and read noise which will be discussed shortly. We employ multiple video datasets, Adobe240fps [23], GoProTrain240fps [15], and Sony240fps [9], to avoid camera bias. Our single training sample is defined by random 128x128 crops created from full-sized video frames. We follow the procedure in Fig. 2(b) to create a training sample from a series of high-speed video frames. For instance, as shown for $N = 15$ images, we have $Y_{0\to1} = (N_1\widehat{I}_{0\to0.5^-} + \widehat{I}_{0.5} + N_2\widehat{I}_{0.5^+\to1})/(N_1 + N_2 + 1)$, where $N_1 = N_2 = 7$, $Y_{0\to1} = \sum_{n=1}^{15} I[n]/15$, $\widehat{I}_{0\to0.5^-} = \sum_{n=1}^{7}\widehat{I}[n]/7$, $\widehat{I}_{0.5^+\to1} = \sum_{n=9}^{15}\widehat{I}[n]/7$, and $\widehat{I}_{0.5} = \widehat{I}[8]$. We vary N based on the variance of pixel-wise intensities of the chosen video frames along the temporal dimension to aggregate different amounts of blur. The higher the variance, the larger is the motion. We ignore static examples in the training set. Typically the value of N ranged from 11 to 39 frames. We demonstrate our blur decomposition idea further in a discrete time representation in Fig. 2(b). We also augment data by randomly employing the following operations: (i) horizontal spatial flipping, (ii) $90°$ or $-90°$ spatial rotations, and (iii) temporal flipping by swapping Y_{0^-} and Y_{1^+}, and $\widehat{I}_{0\to0.5^-}$ and $\widehat{I}_{0.5^+\to1}$, during training.

To emulate proper short exposure images Y_{0^-} and Y_{1^+}, we add scene-dependent noise according the calibrated noise parameters on-the-fly during training based on the noise model described by Hasinoff et al. [6]. For the gain level used in our experiments, we calibrate the noise parameters of the camera based on an affine model for the noise variance given by $var(n) := i\alpha + \beta$, where i is the observed mean intensity, and α and β are the calibrated noise parameters. We use the machine vision Blackfly BFS-U3-16S2C camera to capture our short and long exposure images.

Since our technique uses recursive decomposition, the inputs to the network beyond the first level would have denoised short exposure images. However, we observed no significant difference in our test results between employing three separate trained networks with two, one, and zero noisy images for the short-exposure input images appropriately at different decomposition levels, and a single trained network with two noisy images. Hence, we report our results for the single trained model approach, that takes in noisy short exposure images as inputs, and provide comparisons to the three model approach in the supplementary material.

4.3. Optimization

We train the neural network by employing the following costs during optimization: (a) supervised cost and sum cost for $\widehat{I}_{0\to0.5^-}$, $\widehat{I}_{0.5}$, and $\widehat{I}_{0.5^+\to1}$, and (b) gradient, perceptual, and adversarial costs on the sharp image $\widehat{I}_{0.5}$.

The supervised cost is defined as the mean square error between the estimated and ground-truth outputs corresponding to $\widehat{I}_{0\to0.5^-}$, $\widehat{I}_{0.5}$, and $\widehat{I}_{0.5^+\to1}$. The sum cost is defined by the mean square error according to the blur decomposition process. The gradient cost is based on the isotropic total-variation norm [20] on the sharp image that encourages sharp edges with homogeneous regions. The perceptual cost is defined as the mean squared error between the VGG [12, 22] features at the conv54 layer of the estimated and ground truth sharp images. We also train a two-class discriminator alongside our network following a generative adversarial training procedure [5], which contributes the generator adversarial cost p_{adv} to encourage the sharp image to lie in the natural image distribution.

The cost function is given as follows:

$$\begin{aligned} E =& \|\widehat{I}_{0\to0.5^-} - I_{0\to0.5^-}\|_2^2 + \|\widehat{I}_{0.5} - I_{0.5}\|_2^2 \\ &+ \|\widehat{I}_{0.5^+\to1} - I_{0.5^+\to1}\|_2^2 \\ &+ \lambda_{sum}\|Y_{0\to1} - (N_1\widehat{I}_{0\to0.5^-} + \widehat{I}_{0.5} + N_2\widehat{I}_{0.5^+\to1})/N\|_2^2 \\ &+ \lambda_{perc}\|\mathrm{VGG}(\widehat{I}_{0.5}) - \mathrm{VGG}(I_{0.5})\|_2^2 \\ &+ \lambda_{grad}TV_2(\widehat{I}_{0.5}) + \lambda_{adv}p_{adv}(\widehat{I}_{0.5}) \end{aligned} \tag{2}$$

where λ_{sum}, λ_{perc}, λ_{adv}, and λ_{grad} are weights of the individual costs, $TV_2(\widehat{I}_{0.5}) = \sum_{i,j}\sqrt{\nabla_x^2\widehat{I}_{0.5}(i,j) + \nabla_y^2\widehat{I}_{0.5}(i,j)}$ is the total variation norm. We use $\lambda_{perc} = 3\times10^{-4}$, $\lambda_{adv} = 10^{-4}$, $\lambda_{grad} = 10^{-4}$, and $\lambda_{sum} = 10^{-2}$ in our experiments for the image intensity range $[0, 1]$. We train for 10^5 iterations using Adam as our optimizer with initial learning rate as 10^{-4} scaling it by 0.1 every 2.5×10^4 iterations.

5. Experiments

We first demonstrate our blur decomposition through a visualization of blur kernels. We then provide quantitative comparisons with existing methods followed by an analysis on blur amount. Finally, we show qualitative results on real data captured by our cameras.

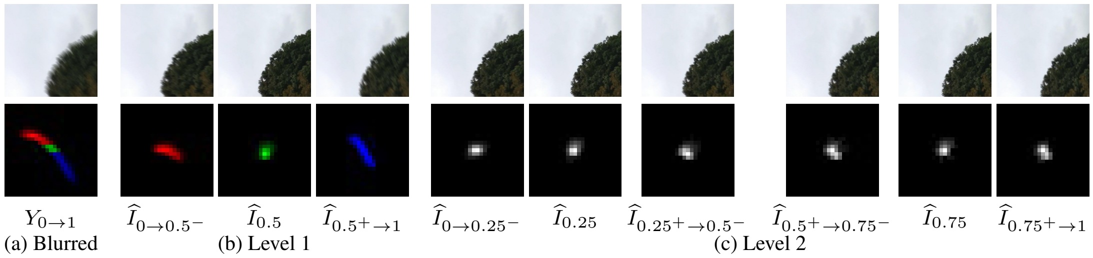


Figure 4. *Successive Reduction of Blur.* At each recursive decomposition level, the blur decreases as depicted in the image patches and blur kernels. The input blurred image has the longest kernel, while it is split into two in the first level and further into four in the second level. The deblurred images at both levels have close-to-delta kernels indicating sharpness of deblurred images.

Table 1. PSNRs (dB) on GoProTest [15] for blur of 11 frames.

Method	$I_{0.25}$	$I_{0.5}$	$I_{0.75}$
Single-blur sequencing [10]	27.24	28.81	27.83
Multi-blur sequencing [9]	29.38	29.52	29.41
Ours (long only)	27.13	27.81	27.32
Ours (short-long-short)	**30.38**	**30.77**	**30.22**

Table 2. PSNR (dB) of $\widehat{I}_{0.5}$ based on blur frame length.

Blur frame length	9	11	15	19
Multi-blur sequencing [9]	30.31	29.52	29.34	29.13
Ours (short-long-short)	**30.87**	**30.77**	**30.62**	**30.21**

5.1. Successive Reduction of Blur

We demonstrate our blur decomposition through blur kernels estimated using the state-of-the-art blind deblurring method of Pan et al. [16]. A blur kernel describes the motion experienced by a point light source located at the image center, and thus acts as an indicator of residual blur/motion present in the image. Fig. 4(a) shows a patch from a motion blurred image $Y_{0\to 1}$ and its long blur kernel. The network takes this image and the short exposure images as inputs (not shown). At first level of decomposition, the half-blurred images have blur kernels of reduced length indicating shorter blur. One can neatly trace down the long kernel trajectory of $Y_{0\to 1}$ by conjoining the split trajectories of level-1. (The kernel estimation is blind, and therefore, it is always centered.) Also, the blur kernel of the middle image $\widehat{I}_{0.5}$ has a close-to-delta kernel indicating a negligible blur. Similarly, in the second level, we get four half-blurred images with further blur reduction and the corresponding two deblurred images. Thus, our recursive decomposition provides an elegant way to remove blur and achieve our goal.

5.2. Quantitative Analysis

We analyze the performance of 3x sequence-rate level, i.e. the PSNR of the middle deblurred image $\widehat{I}_{0.5}$ and that of the second-level decomposition images, $\widehat{I}_{0.25}$ and $\widehat{I}_{0.75}$ against the single-blur sequencing of Jin et al. [10] and multi-blur sequencing of Jin et al. [9]. In addition, we also consider an ablation case of our method by considering only the long exposure blurred image as input without any short exposure images, indicated by *long-only*. Since each of these methods need different types of inputs, we prepare the testing set as follows. We created a set of eleven examples from the eleven test videos of the GoPro 240fps data [15]. Each example is a sequence of alternating short and long exposures. The short exposure is created by taking a single video frame, while the long exposure is synthesized as an average of 11 frames. Our method takes a single consecutive short-long-short triplet as input with added noise to the short exposures. The single-blur sequencing method takes only a single long exposure image, while multi-blur sequencing takes four long exposure images as inputs.

The results of our analysis are provided in Table 1. First, our network trained with short-long-short exposure inputs performs better than training with only the long exposure image indicating the benefit of capturing additional short exposure images in photosequencing. The multi-blur sequencing performs better than the single-blur sequencing owing to more available information as expected. In turn, we are able to perform better than the multi-blur sequencing. Our method recovers textures missed in heavy blur from the short exposure images. Fig. 5(top) depicts this behavior where the prior works are able to reconstruct the leg on the ground which has lesser blur, while the other leg is not recovered even by the multi-blur technique. Our output shows better textures with minimal residual blur and is practically noise-free.

5.3. Blur Amount Analysis

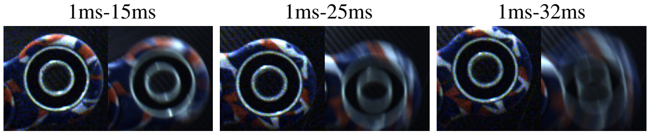
The amount of blur observed in the long exposure image is a synergy of exposure time and motion. We repeated our experiments on the test data for different frame lengths shown in Table 2. The multi-blur sequencing method performed better for shorter frame lengths almost on par with our method and worse for the longer ones.

Fig. 6(a) shows our real captures of a spinner for multiple exposure configurations. At the lowest long exposure,

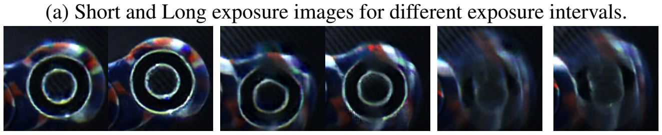
Short/Noisy | Long/Blurry | Single-blur [10] | Multi-blur [9] | Ours (L-only) | Ours (S-L-S) | Ground-truth

Figure 5. *Quasi-real blur and noise on high-speed video GoProTest [15] data.* Comparisons with single-blur sequencing and multi-blur sequencing prior works.

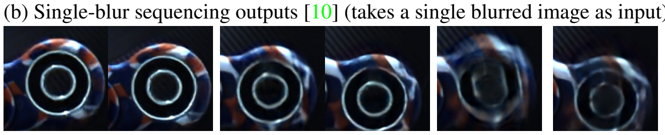
1ms-15ms | 1ms-25ms | 1ms-32ms



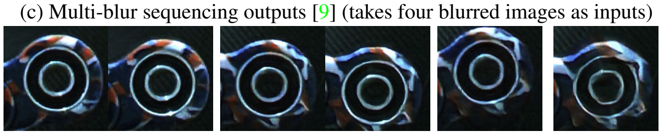
(a) Short and Long exposure images for different exposure intervals.



(b) Single-blur sequencing outputs [10] (takes a single blurred image as input)

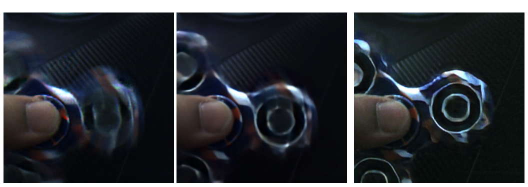


(c) Multi-blur sequencing outputs [9] (takes four blurred images as inputs)



(d) Proposed short-long-short sequencing outputs

Figure 6. *Real Spinner data.* Comparisons with single and multi-blur photosequencing prior works for different blur amounts.



Single-blur [10] | Multi-blur [9] | Proposed sequencing

Figure 7. *Temporal ordering in photosequencing.* The whole sequences are available in the supplementary material.

the blur is minimal, while at the longest setting, the spinner texture is almost lost. Three frames from sequencing results of different methods are shown in (b,c) and (d). Single-blur technique (b) shows residual blur artifacts even at medium blur shown in the third column-set. While the multi-blur technique (c) could recover some texture at the highest setting, our method outperforms it. Further, single-blur technique could not disambiguate temporal ordering leading to wrong local spinner motion as shown in the supplementary material, while our short-long-short sequencing correctly recovers the ordering.

5.4. Results on Real Data

We capture sequences of short and long exposures of a wide variety of scenes using our Blackfly camera to show the efficacy of our photosequencing, some of which are shown in Fig. 8. Our captures comprise both indoor and outdoor scenes with different types of motions including linear blur in *Race*, rotations in *Foosball*, human nonlinear motions in *Gym* and *Skate*, fine textured motion of *Jellyfish*, and closeup finger movement in *Keyboard*. Note that we cannot use any of the existing real blurred videos since our technique requires short exposure images as well.

Comparison with multi-blur sequencing. Fig. 9 shows the *Foosball* scene with noisy short exposures, and blurry player and ball in long exposures, wherein the partially occluded ball moves from behind the bar into view. (Please zoom-in to see noise and blur clearly in first two rows.) The third and fourth rows show the cropped results of Jin et al. [9] and our method. The method of Jin et al. [9] recovers the motion direction correctly but the blur is not completely removed. Our method correctly recovers the sharpness of the player and the ball as well as the motion ordering.

Comparison with frame interpolation. Fig. 10 shows input images and crops of the Skate scene in (a,b). in which heavy noise and blur can be observed. The multi-blur sequencing fails to recover intricate textures of the skateboard and fist as seen in (c). We follow two pipelines to compare with frame interpolation [8]. First, we deblur two long exposures using the state-of-the-art deblurring method of Tao et al. [27] and interpolate frames. Second, we denoise two short exposures using BM3D [4] followed by frame interpolation. In the first case result shown in (d), the heavy resid-

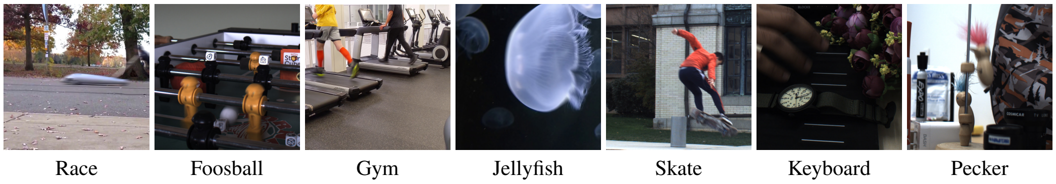
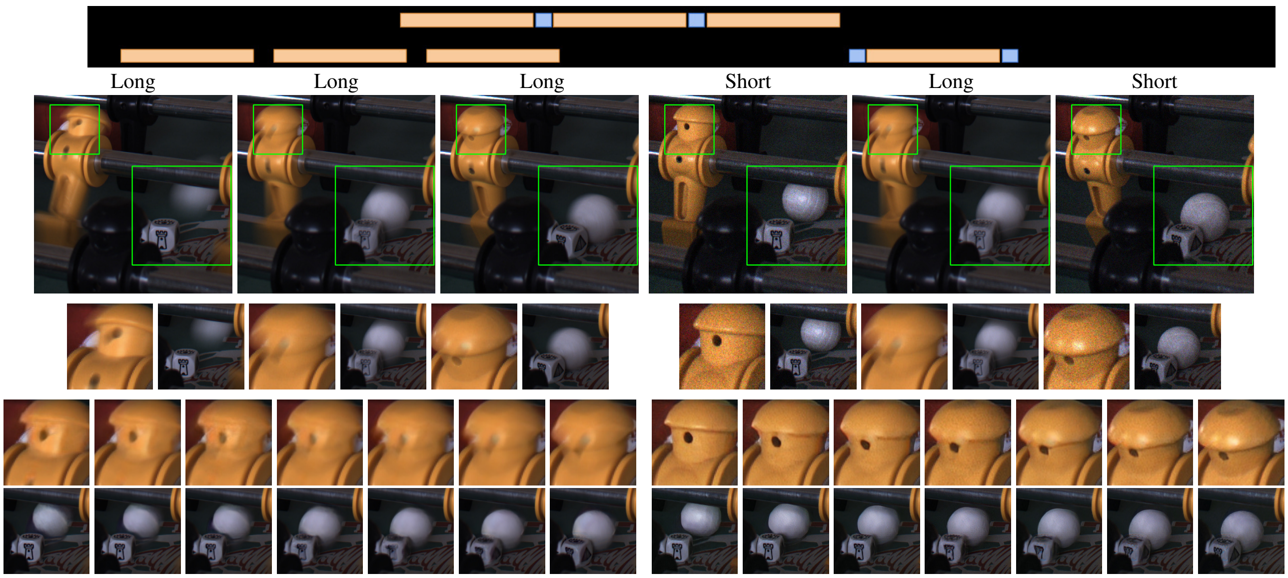


Figure 8. *Some scenes from our real data captures.* The whole sequence is available as supplementary.



(a) Multi-blur sequencing [9]

(b) Proposed short-long-short sequencing

Figure 9. *Comparison with multi-blur sequencing technique on real Foosball data.* Recovering the sequence from multiple blurred images [9] lose out on sharp textures albeit correct motion sequencing. Our short-long-short exposure inputs recover both the motion and texture successfully.

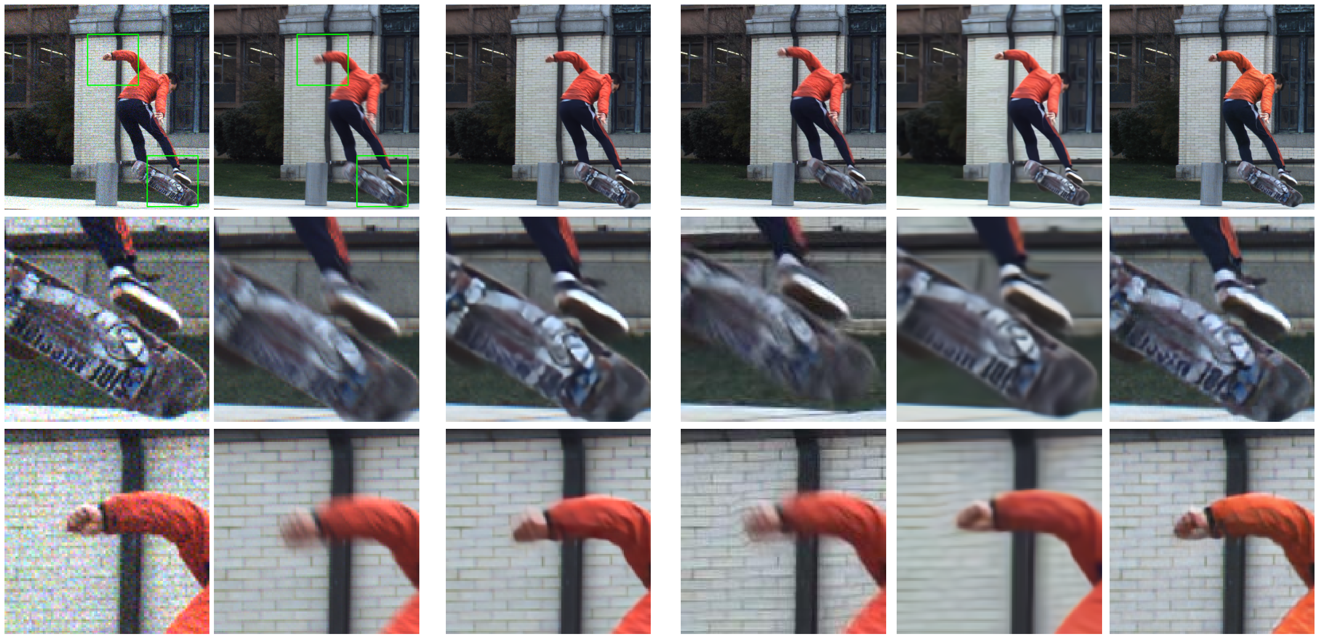
ual blur due to poor deblurring of heavy motion is passed on to interpolation which can only recover blurred frames. In the second case (e), we do note that the texture in the fist is devoid of blur since it uses only the short exposure frames, however the denoised image has a washed-out appearance. Further, the static wall undergoes noise-denoise process for no reason only to have its texture erased. Our method correctly exploits textures of static regions (wall) from the blurred image and textures of moving objects (hand, fist) from the short exposure image as shown in (f).

The proposed technique recovers a sharp frame sequence for a short-long-short triplet. By combining the outputs from a longer sequence of alternating short and long exposures, we can produce high frame-rate videos of long time durations. We showcase some of such examples in the supplementary material. We do note that such sequences have temporal tiling artifacts since each triplet leads to sequences in isolation. We show XT and YT slices for the result of *Skate* data over multiple triplets in Fig. 11.

Failure Example. A challenging scenario is that of heavy motion of thin structures. Fig. 12(a) portrays the *Jumping-Toy* scene wherein the thin legs of the jumping metal toy move very fast. The motion is so ragged that the line of blur marked in yellow is disconnected from the two short exposure points (white to red) as shown in the image crops of Fig. 12(b). Our method shows artifacts in the photosequence with partial leg reconstructions as shown in (c). In comparison, the multi-blur sequencing method performs worse as shown in (d), and suffers residual blur artifacts in other regions as well where we perform better.

6. Conclusion

We proposed a new technique to record fast motion events by capturing short-long-short exposure images. We utilize their complementary information of texture and motion to estimate the sharp high frame-rate photosequence associated with the long exposure. Our technique provides an approach that can be easily implemented on mobile devices, thereby providing the capability of high-speed capture with little change to existing hardware. We believe this would be a fascinating new application of mobile computational photography.

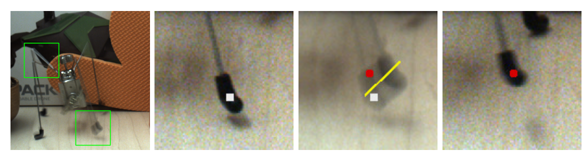


(a) Short (b) Long (c) Multi-blur sequencing [9] (d) Deblur+Interp [8, 27] (e) Denoise+Interp [4, 8] (f) Ours

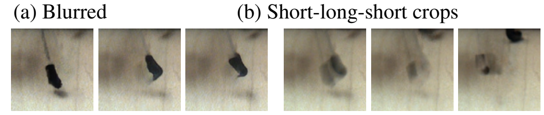
Figure 10. *Results on Skate data.* Our method correctly transfers textures of static regions (wall linings) from the blurred image and textures of moving objects (hand-fist) from the short exposure image.

(a) Input capture (b) Recovered photosequence

Figure 11. *Time slices on Skate.* We recovered the photosequence of consecutive short and long exposures using our method. XT (top) and YT (bottom) slices show discontinuites in the horizontal time axis for the input capture indicating lower frame rate. The slices of our photosequencing result on the right show smooth interpolation of motion during the long exposure intervals.



(a) Blurred (b) Short-long-short crops



(c) Proposed sequencing (d) Multi-blur sequencing [9]

Figure 12. *Results on JumpingToy data.* Very fast jumping of the thin-legged object misaligns the line of blur with the short exposures as in (b). This causes imperfect reconstructions by our method, and also in multi-blur sequencing as shown in (c,d).

Acknowledgement The work of VR, SZ, and ACS was supported by the ARO Grant W911NF-16-01-0441, a gift from Samsung, and the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DOI/IBC) contract number D17PC00345. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon.
Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied of IARPA, DOI/IBC, or the U.S. Government.

References

[1] Moshe Ben-Ezra and Shree K Nayar. Motion deblurring using hybrid imaging. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2003. 2

[2] Ayan Chakrabarti. A neural approach to blind motion deblurring. In *European Conference on Computer Vision*, 2016. 2

[3] Tony F Chan and Chiu-Kwong Wong. Total variation blind deconvolution. *IEEE Transactions on Image Processing*, 1998. 2

[4] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 2007. 6, 8

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014. 4

[6] Samuel W Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics*, 2016. 4

[7] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3

[8] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 6, 8

[9] Meiguang Jin, Zhe Hu, and Paolo Favaro. Learning to extract flawless slow motion from blurry videos. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[10] Meiguang Jin, Givi Meishvili, and Paolo Favaro. Learning to extract a video sequence from a single motion-blurred image. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 2, 3, 5, 6

[11] Dilip Krishnan, Terence Tay, and Rob Fergus. Blind deconvolution using a normalized sparsity measure. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 2

[12] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 4

[13] Dhruv Mahajan, Fu-Chung Huang, Wojciech Matusik, Ravi Ramamoorthi, and Peter Belhumeur. Moving gradients: A path-based method for plausible image interpolation. *ACM Transactions on Graphics*, 2009. 2

[14] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. Phase-based frame interpolation for video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[15] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 4, 5, 6

[16] Jinshan Pan, Deqing Sun, Hanspeter Pfister, and Ming-Hsuan Yang. Blind image deblurring using dark channel prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 5

[17] Kuldeep Purohit, Anshul Shah, and AN Rajagopalan. Bringing alive blurred moments. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2

[18] Ramesh Raskar, Amit Agrawal, and Jack Tumblin. Coded exposure photography: Motion deblurring using fluttered shutter. *ACM Transactions on Graphics*, 2006. 2

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Ccomputer-Assisted Intervention*, 2015. 3

[20] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 1992. 4

[21] Oded Shahar, Alon Faktor, and Michal Irani. Space-time super-resolution from a single video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 2

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 4

[23] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 4

[24] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[25] Jinli Suo, Yue Deng, Liheng Bian, and Qionghai Dai. Joint non-Gaussian denoising and superresolving of raw high frame rate videos. *IEEE Transactions on Image Processing*, 2014. 1

[26] Yu-Wing Tai, Hao Du, Michael S Brown, and Stephen Lin. Image/video deblurring using a hybrid camera. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 2

[27] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2, 6, 8

[28] Lu Yuan, Jian Sun, Long Quan, and Heung-Yeung Shum. Image deblurring with blurred/noisy image pairs. *ACM Transactions on Graphics*, 2007. 2

[29] Jiawei Zhang, Jinshan Pan, Jimmy Ren, Yibing Song, Linchao Bao, Rynson WH Lau, and Ming-Hsuan Yang. Dynamic scene deblurring using spatially variant recurrent neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[30] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3