Learning Spatially Varying Pixel Exposures for Motion Deblurring

Cindy M. Nguyen, Member, IEEE, Julien N. P. Martel, Member, IEEE, and Gordon Wetzstein, Senior Member, IEEE

Abstract—Computationally removing the motion blur introduced by camera shake or object motion in a captured image remains a challenging task in computational photography. The post-processing algorithm must deblur a longer exposure that contains relatively little noise or denoise a short exposure that intentionally removes the opportunity for blur at the cost of increased noise. We present a novel approach of leveraging spatially varying pixel exposures for motion deblurring using next-generation focal-plane sensor–processors along with an end-to-end design of these exposures and a machine learning–based motion-deblurring framework. We demonstrate in simulation and a physical prototype that learned spatially varying pixel exposures (L-SVPE) can successfully deblur scenes while recovering high frequency detail. Our work illustrates the promising role that focal-plane sensor–processors can play in the future of computational imaging.

Index Terms—Motion deblurring, programmable sensors, in-pixel intelligence, end–to-end optimization, computational imaging

1 INTRODUCTION

C ameras have become ubiquitous, their presence felt in every smartphone and countless other devices sold today. Whether their images are designed for social media or processed by a self-driving car, modern cameras are still susceptible to the age-old problem of motion blur. This artifact is the result of either object or scene motion during the image exposure or camera shake via handheld photography, thus rendering the picture of a touching memory or an image used for computer vision tasks useless.

Unfortunately, removing motion blur remains an arduous task. Due to the heterogeneity of local and global motion blur, blind deblurring is difficult to address with pure deconvolution. Recent deep learning methods have popularized multi-scale [1], [2], [3], [4] or multi-temporal [5], [6] deep learning approaches to address this issue. The coarse-to-fine nature of these networks allows for the gradual refining of motion deblurring kernels that are applied spatially invariantly, even in cases of non-uniform motion blur.

These methods have demonstrated the power of machine learning for deblurring, but notably do not take advantage of offloading computation to the hardware. On the other hand, computational imaging methods have excelled in combining software and hardware solutions to simplify many ill-posed computer vision tasks. These include engineering point spread functions (PSFs) [7], [8], [9] and custom coded exposures [10], [11], [12], [13], [14] for motion deblurring. However, these methods are often limited to heuristic designs, and can be restricted by fabrication limits, sensor capabilities, or human intuition.

Fortunately, we are on the brink of a new era of sensor design. The rise of programmable sensors [15], [16], [17], in which sensing and processing can be executed together on the same silicon chip, has brought forth a new age of computing that operates at the time of image capture. Programmable sensors, or focal-plane sensor-processors, open up avenues for analog, digital, or mixed signal processing directly in-pixel. These sensors provide opportunities to preserve optical information of a scene that is otherwise irreversibly destroyed during the sensor integration or capture process. Moreover, the in-pixel intelligence running on these sensors can be learned using end-to-end (E2E) design strategies that jointly optimize the in-pixel programs with downstream computer vision algorithms.

Given these capabilities, we address the above concerns with our jointly Learned Spatially Varying Pixel Exposures (L-SVPE) and machine learning–based reconstruction network to perform high quality motion deblurring (Fig. 1). The benefits are two-fold. Within short exposures, the low signal-to-noise ratios are challenging to handle, but these short exposures still retain high frequency information that we desire in a reconstruction. Within longer exposures,
images will have reduced noise levels thanks to signal accumulation. However, as the sensor integration time increases, these longer exposures are more prone to motion blur. We seek to use the complementary information from both in a single snapshot in our E2E framework.

As programmable sensors grow in production and popularity, we believe in the need to develop appropriate computational imaging algorithms to take advantage of these new capture capabilities. Future mobile phones may no longer be limited to the fixed global exposures they have today for capturing scenes with dynamic motion or low lighting. High dynamic range (HDR) imaging, most popularly done with some form of exposure bracketing [18], will struggle less with image alignment since programmable sensors can capture information at multiple exposures in a single snapshot [19], [20]. We demonstrate that motion deblurring is just one of many tasks that highlight the utility of these powerful sensors.

Specifically, we make the following contributions:

- We introduce a fully differentiable model with a learned in-pixel encoder and deep convolutional decoder for motion deblurring. The encoder can be implemented on a programmable sensor offering “in-pixel intelligence.”
- We demonstrate in simulation that the E2E optimization of the coded exposures along with the decoding network yields superior reconstructions of motion blurred images over those of non-optimized exposures.
- We implement a prototype camera using the SCAMP-5 programmable sensor–processor and demonstrate that our results translate to real-world captures, where the coded exposures are realized electronically.

Source code and trained models can be found on our project website.

2 Related Work

Motion deblurring. Early examples of blind motion deblurring include learning motion blur kernels using CNNs [21], [22], [23]. More recent examples increase the receptive field by using multi-resolution inputs [1], [2], [3], [4] to learn on a combined global and local scale. Attention [3] and atrous convolutions [24], [25] can help apply spatially varying weights for local blurs. Inspired by transformers, Tu et al. [26] use multi-layer perceptrons (MLPs) and attention with multi-resolution features for deblurring, reducing the required number of learnable parameters. Though these methods have produced plausible reconstructions, deblurring networks are fundamentally limited due to the ill-posed nature of the problem, which can only be overcome by changing the image formation model through methods like E2E optimization.

End-to-end optimization. Jointly designing optics and reconstruction networks has emerged as an incredibly useful paradigm in computational imaging [27]. This E2E optimization has been applied to a number of problems such as extended depth of field [28], depth estimation [29], [30], [31], HDR [32], [33], super resolution localization microscopy [34], lensless imaging [35], [36], and Fourier ptychographic microscopy [37]. However, once the optics are fabricated, the challenge of calibrating them arises. To alleviate the need for calibration, in-pixel sensing strategies can also be designed in an E2E fashion for general capture [38], HDR [19], [20], and compressive imaging [39]. We propose following this route, learning a spatially varying exposure pattern jointly with a network for the task of motion deblurring.

Computational imaging for deblurring. Deblurring can be made less of an ill-posed problem with a variety of hardware modifications. Raskar et al. [10] use a random binary coded global exposure pattern, implemented via a liquid-crystal shutter. The coded exposure provides a PSF which preserves high spatial frequencies, allowing the blur to be decoded using traditional deconvolution algorithms. Agrawal and Xu [13] further improve this design with heuristics on PSF invertibility and estimation. Other works use a custom rolling shutter [11] or deblur using information from a camera’s default rolling shutter [40]. Jeon et al. [41] use communication theory to design fluttering patterns, while more recently, Jiang et al. [14] use an interlaced short, medium, and long exposure pattern with a specialized network for reconstruction. We compare this exposure pattern with ours, while using a simpler network for reconstruction. Elmalem et al. [8] design an optic that encodes blur in its color PSF, and Yosef et al. [9] build on this work by additionally using a focus mechanism to recover video frames from a capture with motion blur. Rengarajan et al. [42] proposed using short-long-short exposures with recursive blur decomposition to deblur. Notably, all of the above approaches have been designed using heuristics and theory, making the search space limited to human intuition.

Programmable sensors. The emergence of programmable sensors, offering unprecedented flexibility of in-pixel processing, has brought forth numerous interesting ideas for computational photography. These sensors, also known as focal-plane sensor–processors [16], conduct low-level image processing during the capture. They reduce the need for excessive computational post-processing and enable new capture processes of an image to preserve information that would be lost otherwise, such as dynamic range. Programmable sensors, such as the SCAMP-5 [15] which we use as our prototype, offer programmable pixels and have been successfully used in HDR imaging [19], video compressive imaging [19], [43], depth from defocus [44], feature classification [45], and ego-motion estimation [46]. More recently, coded exposures have also been used for compressive light-field and hyperspectral imaging [39]. Here, we propose to use one variant of these new vision chips, the SCAMP-5, to program pixel-wise coded exposures. To our knowledge, this is the first application of these sensor–processors to motion deblurring.

3 Formulation

We simulate the capture of a scene with our learned programmable sensor using spatially varying pixel exposures (Sec. 3.1), which we also refer to as coded exposures. We then form a multi-channel image with C channels from the single snapshot, where C represents the number of unique
exposure lengths in the learned coded exposure. We do so by interpolating (Sec. 3.2) exposure pixels that were not explicitly captured. This multi-channel image of dimensions $H \times W \times C$ is fed as the input of the decoding network (Sec. 3.3) to produce a reconstructed image. The pipeline is illustrated in Fig. 2.

### 3.1 Learned spatially varying pixel exposures

We model the exposure at pixel location $p$ as the integration of the incident irradiance $V_p$ over an exposure time $t$, which can be written as

$$E_p(t) = \int_0^t V_p(\tau) d\tau,$$

where $t = T_c$ for a typical camera with a fixed global exposure and $E_p$ represents the exposure at pixel $p$. We introduce a learned spatially varying exposure time $T_p$ for each pixel $p$ to indicate an “on” (contrary to “off”) shutter. Thus Eq. 1 can be rewritten as

$$E_p(t) = \int_0^{T_p} V_p(\tau) d\tau.$$  

The exposure $E_p$ relates to the captured image via the camera response function as $I_p(t) = R(E_p(t))$, which captures the noise and quantization effects of the camera.

### 3.2 Exposure length–specific interpolation

A single-shot measurement on our focal-plane sensor–processor produces a single-channel, grayscale image. We expand the measurement as a preconditioning step before decoding. Since our learned exposure times will vary spatially, we interpolate the missing exposure pixels to generate a full resolution estimate of the utilized exposure lengths.

In our framework, we discretize the continuous varying exposure time to a set of discrete values $S = \{t_1, \ldots, t_c\}$, in which $t_c$ is the maximum exposure time. We then interpolate all pixels of the single-channel sensor image with the same exposure to form a separate channel of the image stack that serves as the input to our motion deblurring network. We argue this interpolation step speeds up training since the learned kernels in the decoding network can be applied spatially invariantly. We apply the interpolation function $U : \mathbb{R}^{H \times W} \rightarrow \mathbb{R}^{H \times W \times C}$ where $C$ is the unique number of exposure lengths used from $S$.

We use two different types of interpolation: Bilinear and Scatter-weighted interpolation. Bilinear interpolation (denoted as B) is applied in cases where the exposures have a $2 \times 2$ (Quad) or $3 \times 3$ (Nonad) tiled arrangement (see Sec. 5.3), using information from regular sampling grids. Scatter-weighted interpolation (denoted as S) is applied in cases where pixel exposures of the same length have varying distances from each other across the sensor. Scatter interpolation is applicable when the coded exposures are random or learned, but this method can also be used in the tiled case. Scatter interpolation requires finding the $k$ closest neighbors using a fast $k$-d tree [47] and adding the values of neighboring pixels together, each weighted by their inverse distance from the point of interest to some power $r$. The interpolated value at point $p$ can be written as

$$U(I)_p^c = \frac{\sum_{i=1}^{k} w(p, p_i)I_{p_i}}{\sum_{i=1}^{k} w(p, p_i)},$$

where $c$ is the channel value, $k$ represents the number of nearest neighbors sharing the same exposure length. The weighting function is $w(p, p_i) = \frac{1}{d(p, p_i)^r}$ for $p_i$ such that $T_{p_i} = c$. Each channel has its own interpolation, so we can view this operation as a set of functions $U = [U^1, \ldots, U^C]$. In other words, we interpolate only the missing pixels, with values weighted by inverse Euclidean distance from the

---

**Fig. 2.** Illustration of our L-SVPE optimization framework, including the learned coded shutter, interpolation step, and decoding network. The error between the reconstructed image and ground truth is backpropagated to the coded exposure lengths and decoding network. Here, $T$ represents the maximum exposure length. This pipeline illustrates an example use of four different exposure lengths, providing four full-resolution interpolations.
3.5 Learning discrete exposure values

Learning the relative exposure values, which range from $[1, 8]$, is essentially a quantization step, which is non-differentiable. To this end, we assume a strategy to propagate the gradients through this step. Specifically, we apply a straight-through estimator (STE) [54, 55], which will apply the quantization function $q$ in the forward pass and use the gradients of a differentiable proxy function $\hat{q}$ in the backward pass.

We learn the exposure end points $\phi \in \mathbb{R}^{H \times W}$, $\phi_{h,w} \in [1, \ldots, 8]$. We define a repeat function $M : \mathbb{R}^{H \times W} \rightarrow \mathbb{R}^{8 \times H \times W}$, $M(I)_{t,h,w} = I_{h,w}, t \in [1, \ldots, 8]$ and a fixed matrix

$$P \in \mathbb{R}^{8 \times H \times W} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,W} \\ \vdots & \ddots & \vdots \\ p_{H,1} & \cdots & p_{H,W} \\ p_{h,w} & \cdots & p_{h,w} \end{bmatrix},$$

(6)

We then build our shutter in the forward pass as $q(\phi) = \mathbb{1}[(M(\phi) - P) > 0]$. This guarantees that the pattern will be consecutive along the temporal axis. The gradients with respect to $\phi$ can be decomposed as follows:

$$\frac{\partial \mathcal{L}}{\partial \phi} \approx \frac{\partial \mathcal{L}}{\partial W_\psi} \cdot \frac{\partial W_\psi}{\partial U} \cdot \frac{\partial U}{\partial \hat{q}} \cdot \frac{\partial \hat{q}}{\partial \phi},$$

(8)

where $\mathcal{L}$ is the loss, $W_\psi$ is the decoding network, $U$ is the interpolation step, $q$ is the forward quantization step, $\hat{q}$ is the differentiable proxy function, and $\phi$ is the learned exposure end points.

During the backward pass, we use the gradients of a differentiable proxy function $\hat{q}(\frac{\partial}{\partial q} = \Pi_{[\cdot, 1]}([\frac{\partial}{\partial q}, \mathbb{1}]), \Pi_{[-1, \cdot]}([\frac{\partial}{\partial q}, \mathbb{1}]),$ in which we simply clip the gradients. For the interpolation step, we sum the gradients in the channel dimension so that they will be consistent in dimensions with $\phi$. The gradients from each interpolated point are inversely weighed in the summation to reflect changes to the points from which they derive their interpolated value from. We apply the gradients to our learned exposure end points:

$$\phi^{(k)} \leftarrow \phi^{(k-1)} - \alpha \left(\frac{\partial \mathcal{L}}{\partial \phi}^T \mathcal{L}(W_\psi(U(I)), Y)\right).$$

(9)

Each pixel is limited to $7$ analog memories and $13$ single-bit digital memories. We use $6$ of the digital memories to create $2^6 = 64$ possible time-slots, where $64$ out of $64$ “on” encoded bits would be equivalent to our Long exposure (32 for Medium and 8 for Short). We segment the $64$ time-slots into $8$ to get $8$ learnable options for exposure times. Here, $8$ out of $8$ is Long, $4$ out of $8$ is Medium, and $1$ out of $8$ would be a Short exposure length.

4 Implementation

Dataset. To train our network and evaluate its performance, we use the Need for Speed (NfS) dataset [56], which consists of $100$ videos obtained from the internet. Each video is captured at $240$ frames per second (fps) with a $1280 \times 720$ resolution that we center crop to $512 \times 512$. We allocate $80$ videos for our training set and $20$ videos for our test set. For each video, we select $8$ random $8$-frame-long segments

Fig. 3. Qualitative comparison between the reconstructions of two learned exposure models using L-SVPE trained with an $L_2$ loss and our Combo loss. As LPIPS decreases, the quality of the reconstruction increases. $L_2$ lends itself to more blurry reconstructions, which may miss high frequency details.

point of interest. We use $C = 8$ only for baselines with interpolation where all exposure lengths may be present (e.g. Uniform Random, L-SVPE). For Quad, $C = 4$.

3.3 Capture decoding

Once we obtain an interpolated multi-channel image, we decode this image using the well-known U-Net architecture [48]. We opt for the widely used U-Net architecture that has been useful for a variety of imaging problems [49, 50] and has less parameters than the state-of-the-art deep deblurring methods [1]. Our CNN $W_\psi : \mathbb{R}^{H \times W \times C} \rightarrow \mathbb{R}^{H \times W}$ contains skip connections with a depth of $6$ without batch normalization, and each downsampling block contains a single convolution and ReLU with learned parameters $\psi$. The upsampling blocks use ConvTranspose to upsample the features at each stage. Our decoding step can be written as

$$\hat{Y}_p = W_\psi(U(I)_p)$$

(4)

where $\hat{Y}_p$ is the reconstructed grayscale value at pixel $p$.

3.4 Loss

We use a linear combination of an MSE loss ($L_2$) and a perceptual loss using VGG features ($L_{VGG}$) [51], which we will call our Combo loss. Our loss can be written as

$$L_{combo} = L_2 + \lambda L_{VGG},$$

(5)

where $\lambda$ is the regularizer for the VGG-based loss. We use $\lambda = 100$. We choose the first frame to reduce the need to deblur and prioritize using the information from the shortest exposure, as denoising is typically easier than deblurring.

The VGG-based loss is exceedingly helpful in reaching the perception-distortion trade-off [52, 53], which describes networks as trading off PSNR and SSIM performance for perceptual quality. Since there are many plausible reconstructions for motion blurred images, we find that the perceptual loss helps find a more suitable reconstruction over an unregularized MSE loss, which optimizes for per-pixel accuracy (Fig. 3).
5 EXPERIMENTS

We perform a series of experiments to highlight the value of our learned coded exposure. Specifically, we evaluate our method on different coded exposures and ablate several design choices made in our E2E model.

5.1 Baselines

To demonstrate the utility of our learned exposure, we compare its performance against the following fixed exposures, where B represents Bilinear interpolation and S represents Scatter interpolation. Our baselines (Fig. 4) include:

- **Burst Average**: All 8 frames are averaged together, simulating a Long exposure, and compared to the ground truth. This baseline does not use a decoder.
- **Short**: We simulate a 240 fps capture which is equivalent to a single frame of the video segment input.
- **Medium**: We simulate a 120 fps capture by averaging the first four frames of the input.
- **Long**: We simulate a 30 fps capture by averaging all eight frames of the input.
- **Uniform Random (S)**: We initialize a fixed 512 × 512 array of pixel exposures uniformly selected from length 1 through 8.
- **Poisson Random (S)**: We use a multi-class Poisson disk sampling algorithm [59] to generate a Poisson-distributed fixed pixel exposure from length 1 through 8.
- **Nonad (nine-tuple) (B, S)**: We use the full range of exposures in a 3 × 3 arrangement. We randomly arrange an array of pixel exposures, each pixel a unique length from 1 through 8, with an additional exposure length 1 to make a total of 9 pixels in the 3 × 3. We then tile this fixed arrangement to fill the 512 × 512 resolution.

- **Quad (B, S)**: Following Jiang et al. [14], we use a fixed tile coded arrangement of LMMS (long-medium-medium), where the short pixel is 240 fps, medium pixels are 120 fps, and long pixel is 30 fps. We use this baseline to also initialize L-SVPE which explains its resemblance to Quad (Fig. 4).
- **Full**: We concatenated a full-resolution stack of the Short, Medium, and Long exposure all captured with the same start time and same viewpoint (3-channel stacked image). This baseline serves as a practical theoretical upper bound for how well our method can do without requiring an interpolation step and with full resolution information at different exposure times. This stack of captures would be physically impossible to capture but can be easily simulated. We trained an 8-channel version of this upper bound, but found it challenging to train to convergence without substantial hyperparameter tuning.

For fair comparison, we use the same decoder architecture as our method on all baselines except the Burst Average, which does not use a decoding network. Each baseline, however, has its own set of trained weights. L-SVPE, by default, uses Scatter interpolation. See the supplemental material for more details on our choice of baselines.
5.2 Determining interpolation parameters
To determine the optimal parameters for Scatter interpolation, we compute the average accuracy and time to perform interpolation using $k$ neighbors on a test set of video segments. We first simulate the exposure capture of each video segment using the chosen spatially varying exposures to get a single-channel capture. We then use Scatter interpolation to interpolate pixels of exposure lengths that were not captured from a single-channel capture. We then use Scatter interpolation to interpolate pixels of exposure lengths that were not explicitly captured to acquire a $H \times W \times C$ multi-channel image. We then compute the accuracy of each interpolation of each individual channel and average the metrics for each channel together for each image. We then compute the average of the metrics over the entire test dataset.

We present these results in Table 1. Specifically, we test the interpolation on Uniform and Poisson Random exposures. We observe that as $k$ increases, so does the time needed for computation per image. We observe that $k = 3$ or 4 provides comparably good interpolations, with $k = 3$ overall being the fastest. Thus, for all our Scatter-based methods, we use $k = 3$ and $r = 1$ which allows for relatively fast computation and accuracy.

5.3 Comparison against baselines
Figure 5 presents qualitative comparisons between these baselines. The Short exposure performs better than Medium and Long exposures, due to the network more easily learning to denoise than deblur. The Full exposure outperforms all single exposure baselines, demonstrating the utility of combining information from different exposure lengths.

We see that baselines with spatially variant pixels outperform the Medium and Long baselines, while also notably performing better than the Short exposure significantly in LPIPS. This result demonstrates the utility of varying exposures for perceptual quality. More structured shutters, like Nonad and Quad, do better than random shutters likely because of the uniformity of data given from the sensor. Our approach with learned exposures can outperform all baselines, reaching the closest to the theoretical upper bound, Full, in performance. We believe that L-SVPE may learn a better priority weighting for different exposures and mixes the small periodic structure of Quad with semi-regular deviations that reoccur over a larger period to mitigate artifacts.

5.4 Ablation studies
Table 2 presents ablation studies on each component of the network. We compare our decoder choice, U-Net, against
DnCNN [60], another popular memory-efficient image reconstruction network. The DnCNN is trained to predict the residual noise of the first channel of the multi-channel input into the decoder, if applicable. These studies demonstrate the utility of each component of our method. Note that the Scatter interpolation may not provide the best PSNR performance over no interpolation, but improves SSIM and LPIPS, which is consistent with our baselines study results. We found that DnCNN with a perceptual loss can be often detrimental to performance, which may suggest that DnCNN is not universally optimal for providing perceptually plausible deblurring from spatially varying exposures.

5.5 Image reconstruction quality

In Figure 6, we present a few qualitative examples of reconstructions with the top-performing baselines. To produce reconstructed RGB images, we individually reconstruct each channel with our baselines, which are trained on grayscale images. L-SVPE generalizes well across different color channels, while methods such as Short and Quad Bilinear can introduce discolored artifacts.

We believe these artifacts may also arise from the reconstruction network trying to denoise by blurring, leading to color “bleeding.” We highlight the robust performance of our model against optical blur observed in Row 2 and global blur observed in Row 3. Row 1 and Row 4 demonstrate how L-SVPE is capable of recovering high frequency detail in the wrinkles of the coat and the fencing in front of the white car, respectively.

6 Prototype

We implement a physical prototype of our model using a focal-plane sensor–processor. Specifically, we use the SCAMP-5 [15], a 256 × 256 pixel array in which each pixel contains a programmable processing element (PE). Each PE contains a small number of analog and single-bit digital memories that can be set using dedicated instructions. We program the learned coded exposure pixel-wise into the analog memories using a micro-controller.

Figure 7 shows reconstructions from trained global exposure models (Long, Medium, and Short) and the two top-performing spatially varying exposures (Quad with Bilinear interpolation and L-SVPE) on two captured scenes, Swing and Jump Rope. Although Long and Medium reconstructions preserve the static background details, there are few improvements in motion deblurring for both scenes. The Short model tends to blur high frequency details such as the grass in Swing and the shoe detail in Jump Rope in an attempt to denoise the image. Quad Bilinear introduces blurring in areas with fine detail, losing details in the overhead wires in
Fig. 7. Reconstructions of images captured using a physical focal-plane sensor–processor prototype (SCAMP-5). We compare reconstructions of two scenes (Swings and Jump Rope) from global exposures (Long, Medium, and Short) models and the best performing spatially varying exposures (Quad with Bilinear interpolation and L-SVPE). Rows 1 and 5: Captured images. The Short capture is noisier than its longer exposure counterparts. Rows 2 and 6: Reconstruction from the networks. Rows 3 and 4: Zoom ins of the reconstructed Swings scene. L-SVPE can successfully recover the lower shoe in addition to maintaining the high frequency detail of the static tree. Rows 7 and 8: Zoom ins of the reconstructed Jump Rope. L-SVPE preserves the sharp edges of the ground tiles and reconstructs the details of the shoe.
Discussions. Only L-SVPE can successfully recover the fine details of each scene in addition to deblurring. More details on how the captures were processed and videos of these scenes can be found in the supplemental material.

7 Discussion

We present a novel method for motion deblurring using learned coded pixel exposures. We demonstrate that our joint hardware-software approach is better than deep learning for comparable reconstruction networks. L-SVPE is able to address dynamic motion blurs and varying levels of noise across many different scenes. We demonstrate that its performance translates to a physical prototype, in which we show that L-SVPE can deblur while preserving high frequency details in real-world scenes.

Limitations. Our programmable sensor operates in grayscale, without any color filter on top of the sensor. Thus, we design our method around capturing grayscale scenes and process RGB channels individually. Additionally, many motion blur datasets do not contain many samples that would allow robust training against over- and under-exposed scenes. To our knowledge, no dataset with sufficient motion blur captured at a high frame rate exists. Therefore, due to data limitations, we do not test against these lighting scenarios.

We could alternatively use an arbitrarily short exposure to mitigate motion deblurring and focus solely on denoising. However, shorter exposures suffer from quantization artifacts on the sensor and require extensive denoising algorithms [61], [62], [63], [64]. Thus, our method is a more robust solution to different lighting scenarios.

Future Work. In future work, we would like to address the aforementioned limitations. Such work would include incorporating HDR scenes so that we can train for over- and under-exposed scenes. We also do not explicitly focus on optical blur or defocus blurs, and thus an improved E2E model could include modeling the optics for improved robustness against different types of blurs. It would be additionally useful to expand E2E methods like ours to programmable sensors that can capture different color channels as well. Different colors captured at different exposures such as that of Jiang et al. [14] can provide helpful cues in reconstruction.

Conclusion. These efforts add to the growing foundation for emerging programmable sensors in computational imaging. As these sensors become more widespread in their use, we may begin to reframe our thinking of how to address ill-posed computer vision tasks, from object classification to HDR imaging. This work serves as a step in that direction.

Acknowledgments

This project was in part supported by NSF Award 1839974, an NSF Graduate Research Fellowship (DGE-1656518), a PECASE by the ARL, and SK Hynix. We thank Piotr Dudek for providing the SCAMP-5 sensor and related discussions. The authors would also like to thank Mark Nishimura, Alexander Bergman, and Nittish Padmanaban for thoughtful discussions.

References

[1] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, “Multi-stage progressive image restoration,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14821–14831.
[2] S. Nah, T. Hyun Kim, and K. Mu Lee, “Deep multi-scale convolutional neural network for dynamic scene deblurring,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3883–3891.
[3] M. Sun, K. Purohit, and A. Rajagopalan, “Spatially-attentive patch-hierarchical network for adaptive motion deblurring,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 3606–3615.
[4] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, “Scale-recurrent network for deep image deblurring,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8174–8182.
[5] D. Park, D. U. Kang, J. Kim, and S. Y. Chun, “Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training,” in European Conference on Computer Vision. Springer, 2020, pp. 327–343.
[6] M. Jin, G. Meishvili, and P. Favaro, “Learning to extract a video sequence from a single motion-blurred image,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6334–6342.
[7] A. Levin, P. Sand, T. S. Cho, F. Durand, and W. T. Freeman, “Motion-invariant photography,” ACM Transactions on Graphics (TOG), vol. 27, no. 3, pp. 1–9, 2008.
[8] S. Elmalem, R. Giryes, and E. Marom, “Motion deblurring using spatiotemporal phase aperture coding,” Optica, vol. 7, no. 10, pp. 1332–1340, 2020.
[9] E. Yosef, S. Elmalem, and R. Giryes, “Video reconstruction from a single motion blurred image using learned dynamic phase coding,” arXiv preprint arXiv:2112.14768, 2021.
[10] R. Raskar, A. Agrawal, and J. Tumblin, “Coded exposure photography: motion deblurring using fluttered shutter,” in ACM SIGGRAPH 2006 Papers, 2006, pp. 795–804.
[11] L. Gu, Y. Hitomi, T. Mitsunaga, and S. Nayar, “Coded rolling shutter photography: Flexible space-time sampling,” in 2010 IEEE International Conference on Computational Photography (ICCP). IEEE, 2010, pp. 1–8.
[12] Y.-W. Tai, N. Kong, S. Lin, and S. Y. Shin, “Coded exposure imaging for projective motion deblurring,” in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2010, pp. 2408–2415.
[13] A. Agrawal and Y. Xu, “Coded exposure deblurring: Optimized codes for psf estimation and invertibility,” in 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2009, pp. 2066–2073.
[14] Y. Jiang, I. Choi, J. Jiang, and J. Gu, “Hdr video reconstruction with tri-exposure quad-bayer sensors,” arXiv preprint arXiv:2103.10982, 2021.
[15] S. J. Carey, A. Lopich, D. R. Barr, B. Wang, and P. Dudek, “A 100,000 fps vision sensor with embedded 535gops/w 256× 256 simd processor array,” in 2013 Symposium on VLSI Circuits. IEEE, 2013, pp. C182–C183.
[16] A. Zarandy, Focal-plane sensor-processor chips. Springer Science & Business Media, 2013.
[17] M. Z. Wong, B. Guillard, R. Murai, S. Saeedi, and P. H. Kelly, “Analognet: Convolutional neural network inference on analog focal plane sensor processors,” arXiv preprint arXiv:2006.01765, 2020.
[18] P. E. Debevec and J. Malik, “Recovering high dynamic range radiance maps from photographs,” in ACM SIGGRAPH 2008 classes, 2008, pp. 1–10.
[19] J. N. Martel, L. K. Mueller, S. J. Carey, P. Dudek, and G. Wetzstein, “Neural sensors: Learning pixel exposures for hrd imaging and video compressive sensing with programmable sensors,” IEEE transactions on pattern analysis and machine intelligence, vol. 42, no. 7, pp. 1642–1653, 2020.
[20] H. M. So, J. N. Martel, P. Dudek, and G. Wetzstein, “Manussacam: Learning snapshot high-dynamic-range imaging with perceptually-based in-pixel irradiance encoding,” arXiv preprint arXiv:2112.05221, 2021.
[21] 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, 2015, pp. 769–777.

22. A. Chakrabarti, “A neural approach to blind motion deblurring,” in European conference on computer vision. Springer, 2016, pp. 221–235.

23. C. J. Schuler, M. Hirshc, S. Harmeling, and B. Schölkopf, “Learning to deblur,” IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 7, pp. 1439–1451, 2015.

24. K. Purohit and A. Rajagopal, “Region-adaptive dense network for efficient motion deblurring,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 20, pp. 11882–11889.

25. H. Miao, W. Zhang, and J. Bai, “Aggregated dilated convolutions for efficient motion deblurring,” in 2018 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2018, pp. 1–6.

26. Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. Bovik, and Y. Li, “Maxim: Multi-axis mlp for image processing,” arXiv preprint arXiv:2201.02973, 2022.

27. G. Wetzstein, A. Ozcan, S. Gigan, S. Fan, D. Englund, M. Soljacic, C. Denz, D. A. B. Miller, and D. Psaltis, “Inference in artificial intelligence with deep optics and photonics,” Nature, vol. 588, pp. 528–534, 2021.

28. V. Sitzmann, S. Diamond, Y. Peng, X. Dun, S. Boyd, W. Heidrich, F. Heide, and G. Wetzstein, “End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging,” ACM Transactions on Graphics (TOG), vol. 37, no. 4, pp. 1–13, 2018.

29. J. Chang and G. Wetzstein, “Deep optics for monocular depth estimation and 3d object detection,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 10193–10202.

30. H. Ikoma, C. M. Nguyen, C. A. Metzler, Y. Peng, and G. Wetzstein, “Depth from defocus with learned optics for imaging and occlusion-aware depth estimation,” in 2021 IEEE International Conference on Computational Photography (ICCP). IEEE, 2021, pp. 1–12.

31. W. Wu, V. Boominathan, H. Chen, A. Sankaranarayanan, and A. Veeraraghavan, “Phasecam3d—learning phase masks for passive single view depth estimation,” in 2019 IEEE International Conference on Computational Photography (ICCP). IEEE, 2019, pp. 1–12.

32. C. A. Metzler, H. Ikoma, Y. Peng, and G. Wetzstein, “Deep optics for single-shot high-dynamic-range imaging,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1375–1385.

33. Q. Sun, E. Tseng, Q. Fu, W. Heidrich, and F. Heide, “Learning rank-1 diffractive optics for single-shot high dynamic range imaging,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1386–1396.

34. E. Nehme, D. Friedman, R. Gordon, B. Ferdman, L. E. Weiss, O. Alalouf, T. Naor, R. Orange, T. Michaeli, and Y. Shechtman, “Photo-realistic single image super-resolution using a generative adversarial network,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4681–4690.

35. Y. Bengio, N. Léonard, and A. Courville, “Estimating or propagating gradients through stochastic neurons for conditional computation,” arXiv preprint arXiv:1308.3432, 2013.

36. G. Hinton, N. Srivastava, and K. Swersky, “Neural networks for machine learning,” Cited on, vol. 14, no. 8, p. 2, 2018.

37. H. Kiani Gallowghi, A. Fage, C. Huang, D. Ramanan, and S. Lucey, “Need for a model: A benchmark for human frame rate object tracking,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1125–1134.

38. I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” arXiv preprint arXiv:1711.05101, 2017.

39. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” IEEE transactions on image processing, vol. 13, no. 4, pp. 600–612, 2004.

40. L.-Y. Wei, “Multi-class blue noise sampling,” ACM Transactions on Graphics (TOG), vol. 29, no. 4, pp. 1–8, 2010.

41. K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising,” IEEE transactions on image processing, vol. 26, no. 7, pp. 3142–3155, 2017.

42. V. Rengarajan, S. Zhao, R. Zhen, J. Glotzbach, H. Sheikh, and A. C. Sankaranarayanan, “Photocoding of motion blur using short and long exposures,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 510–511.

43. M. Iliadis, L. Spinoulas, and A. K. Katsaggelos, “Deepbinarymask: Learning a binary mask for video compressive sensing,” arXiv preprint arXiv:1607.03343, 2016.

44. J. N. Martel, L. K. Müller, S. J. Carey, and P. Dudek, “High-speed depth from focus on a programmable vision chip using a focus tunable lens,” in 2017 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2017, pp. 1–4.

45. J. Chen, S. J. Carey, and P. Dudek, “Feature extraction using a portable vision system,” in IEEE/RSJ Int. Conf. Intel. Robots Syst., Workshop Vis.-based Agile Auton. Navigation UAVs, 2017.

46. L. Bose, J. Chen, S. J. Carey, P. Dudek, and W. Mayol-Cuevas, “Visual odometry for pixel processor arrays,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4604–4612.

47. T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “An efficient k-means clustering algorithm: Analysis and implementation,” IEEE transactions on pattern analysis and machine intelligence, vol. 24, no. 7, pp. 881–892, 2002.

48. O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.

49. S.-J. Cho, S.-W. Ji, J.-P. Hong, S.-W. Jung, and S.-J. Ko, “Rethinking coarse-to-fine approach in single image deblurring,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 4641–4650.

50. B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, “Enhanced deep residual networks for single image super-resolution,” in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2017, pp. 136–144.

51. R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 586–595.

52. Y. Blau and T. Michaeli, “The perception-distortion tradeoff,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6228–6237.

53. C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang et al., “Photo-realistic single image super-resolution using a generative adversarial network,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4681–4690.

54. B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, “Enhanced deep residual networks for single image super-resolution,” in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2017, pp. 136–144.

55. R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 586–595.
Cindy M. Nguyen is currently a Ph.D. candidate in Electrical Engineering at Stanford University, Stanford, CA, USA. She received her B.S. in Bioengineering at Stanford University, Stanford, CA, USA in 2019. Her research interests include problems in computational photography, such as denoising, deblurring, and depth estimation. She is a recipient of the NSF Graduate Research Fellowship.

Julien N. P. Martel obtained his Ph.D. in 2019 at ETH Zurich in the Institute of Neuroinformatics. He was a postdoctoral research scholar at Stanford University from 2019 to 2022 in the Computational Imaging Lab. His broad interests are in vision and graphics but specifically in computational imaging and rendering systems and algorithms. Many of his projects have revolved around exploiting sensors that embed little processors in their pixels, allowing them to change the way light is captured.

Gordon Wetzstein (Senior Member, IEEE) received the graduation (with Hons.) degree from the Bauhaus-Universität Weimar, Weimar, Germany and the Ph.D. degree in computer science from the University of British Columbia, BC, Canada, in 2011. He is currently an Associate Professor of Electrical Engineering and, by courtesy, of Computer Science, with Stanford University, Stanford, CA, USA. He is the Leader of Stanford Computational Imaging Lab and a Faculty Co-Director of the Stanford Center for Image Systems Engineering. At the intersection of computer graphics and vision, computational optics, and applied vision science, his research has a wide range of applications in next-generation imaging, display, wearable computing, and microscopy systems. He is the recipient of an NSF CAREER Award, an Alfred P. Sloan Fellowship, an ACM SIGGRAPH Significant New Researcher Award, a Presidential Early Career Award for Scientists and Engineers (PECASE), an SPIE Early Career Achievement Award, a Terman Fellowship, an Okawa Research Grant, the Electronic Imaging Scientist of the Year 2017 Award, an Alain Fournier Ph.D. Dissertation Award, Laval Virtual Award, and the Best Paper and Demo Awards at ICCP 2011, 2014, and 2016 and at ICIP 2016.