

# Deblurring Low-Light Images with Light Streaks

Zhe Hu<sup>1</sup>, Sunghyun Cho<sup>2</sup>, Jue Wang<sup>3</sup>, *Senior Member, IEEE*, and  
Ming-Hsuan Yang<sup>4</sup>, *Senior Member, IEEE*

**Abstract**—Images acquired in low-light conditions with handheld cameras are often blurry, so steady poses and long exposure time are required to alleviate this problem. Although significant advances have been made in image deblurring, state-of-the-art approaches often fail on low-light images, as a sufficient number of salient features cannot be extracted for blur kernel estimation. On the other hand, light streaks are common phenomena in low-light images that have not been extensively explored in existing approaches. In this work, we propose an algorithm that utilizes light streaks to facilitate deblurring low-light images. The light streaks, which commonly exist in the low-light blurry images, contain rich information regarding camera motion and blur kernels. A method is developed in this work to detect light streaks for kernel estimation. We introduce a non-linear blur model that explicitly takes light streaks and corresponding light sources into account, and pose them as constraints for estimating the blur kernel in an optimization framework. For practical applications, the proposed algorithm is extended to handle images undergoing non-uniform blur. Experimental results show that the proposed algorithm performs favorably against the state-of-the-art methods on deblurring real-world low-light images.

**Index Terms**—Image deblurring, light streak, non-uniform blur

## 1 INTRODUCTION

TAKEING good pictures in low-light conditions is perhaps the most challenging task for non-professional photographers. Since longer exposure time is often required in such cases to generate well-lit images, the captured photos using a handheld camera are often blurry due to inevitable camera shakes. It is of great interest to develop effective image deblurring algorithms to recover sharp images from blurry low-light inputs.

Although significant advances in single-image deblurring have been recently made [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], the state-of-the-art methods are less effective for handling low-light photos as the success usually hinges on whether or not salient image features such as edges [4], [9], [11], [12] can be extracted reliably for blur kernel estimation. However in low-light images, the amount of salient image features that can be extracted is often limited, as shown by the example in Fig. 1. Furthermore, the contents of low-light images often are manipulated by various signal processors with non-linear tone mapping. Thus, these images cannot be well modeled by linear blur functions used in most deblurring approaches [13].

In this paper, we present a novel method for removing image blur caused by camera shakes, by explicitly utilizing *light streaks* that often appear in low-light images. Light streaks

are generated by blurred light sources such as light bulbs, flash lights and reflected lights, which are common in both natural (e.g., stars in the sky) and man-made scenes (e.g., street lights). Given that these light sources are small high-intensity objects in the scenes, the light streaks roughly resemble the shapes of the underlying blur kernels. Intuitively, light streaks contain rich blur information that can potentially help deblur low-light images.

However, without proper design, the presence of light streaks can adversely affect the performance of existing deblurring approaches on estimating blur kernels accurately. The reasons are twofold. First, most state-of-the-art methods extract and use salient edges for blur kernel estimation [4], [11], [14], but light streaks often contain high-contrast sharp edges around them that may mislead blur kernel estimation into a false delta blur kernel. Second, light streaks often contain saturated pixels. As shown in [15], without proper handling of saturated pixels, deconvolution of light streak pixels may result in significant ringing artifacts and severely degrade blur kernel estimation. To avoid such adverse effects of light streaks on blur kernel estimation, Harmeling et al. [16] and Whyte et al. [17] discard saturated pixels before estimating blur kernels. Regarding non-blind deconvolution, Cho et al. [15] and Whyte et al. [17] explicitly model saturated pixels in their optimization processes to suppress ringing artifacts caused by saturated pixels. However, all these methods do not exploit rich information that light streaks contain.

In this work, we propose a deblurring algorithm that exploits light streaks as additional cues for blur kernel estimation. We extend the widely-used linear blur formulation and propose a non-linear model by explicitly considering point light sources as well as streaks. We show that this model describes the formation of low-light images with streaks more accurately. Next, we present a kernel estimation energy function that takes light streaks as well as other image structures into account. Our method also automatically detects

- Z. Hu is with Hikvision Research America, Santa Clara, CA 95054. E-mail: zhe.hu@hikvision.com.
- S. Cho is with DGIST, Korea. E-mail: scho@dgist.ac.kr.
- J. Wang is with Megvii Inc., Seattle, WA 98052. E-mail: wangjue@megvii.com.
- M.-H. Yang is with the University of California at Merced, Merced, CA 95340. E-mail: mhyang@ucmerced.edu.

Manuscript received 9 July 2016; revised 1 Sept. 2017; accepted 17 Sept. 2017. Date of publication 1 Nov. 2017; date of current version 12 Sept. 2018. (Corresponding author: Ming-Hsuan Yang.)

Recommended for acceptance by J. Yu.

For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.

Digital Object Identifier no. 10.1109/TPAMI.2017.2768365

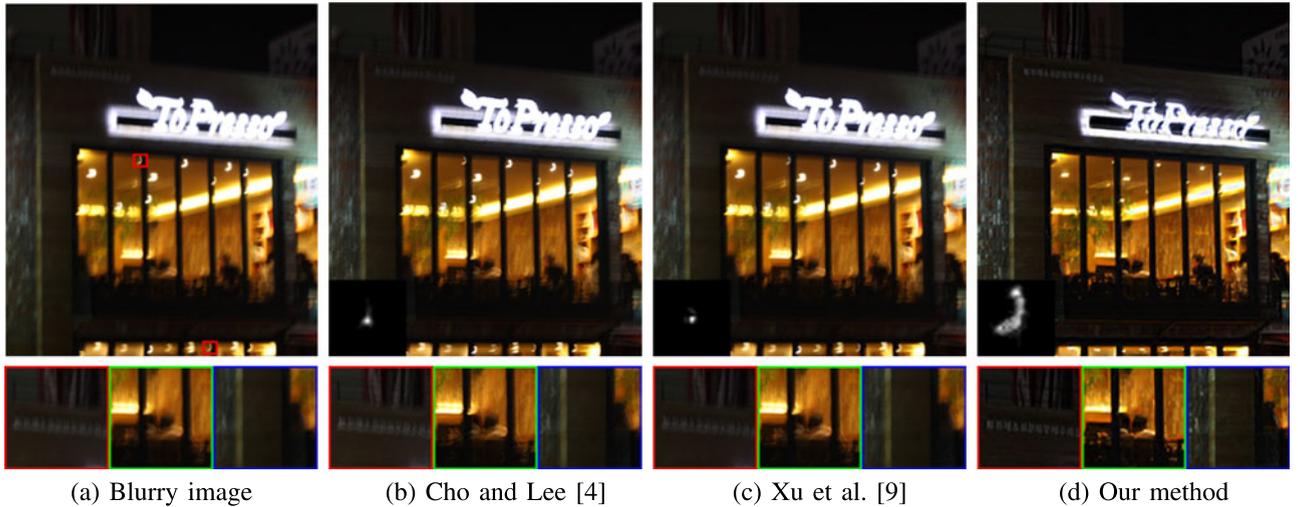


Fig. 1. Deblurring a low-light image. The amount of salient image features that can be extracted from the low-light image is limited. This leads to the failure of the state-of-the-art methods [4], [9] on the image. The proposed method (d) improves the results by detecting and making use of the light streaks (the detected light streaks are shown in the boxes of (a)).

light streaks that are useful for kernel estimation. After the blur kernel is estimated, the restored image is obtained by a regularized Richardson-Lucy deconvolution scheme with outlier handling to suppress ringing artifacts. Quantitative and qualitative experimental results show that the proposed algorithm performs favorably against the state-of-the-art methods for deblurring low-light images.

## 2 RELATED WORK AND CONTEXT

The problem of removing spatially invariant blur has been studied extensively and significant progress has been made [1], [4], [5], [14], [18], [19], [20], [21], [22]. A blurry image can be formulated by a convolution process with a spatially invariant blur kernel

$$B = K * I + N, \quad (1)$$

where  $B$ ,  $I$  and  $K$  denote a blurry image, a latent image and a blur kernel, respectively. In addition,  $N$  is used to describe image noise and  $*$  is a convolution operator. Removing blur caused by camera shakes becomes a blind deconvolution problem. As blind deconvolution is an ill-posed problem, prior knowledge is often required for effective solutions. Early work focuses on generic priors obtained from statistic properties of natural images. In [1], the heavy-tailed gradient distribution of natural images is exploited as a constraint for sharp images. A method that uses gradients of a latent image is presented in [3] in which constraints on consistency of smooth regions before and after blurring are enforced. In [23], a deblurring method is proposed to exploit sparsity constraints for both blur kernels and latent images in the wavelet domain. In contrast to methods that adopt fixed priors, recent approaches use adaptive priors to better represent image contents of specific inputs. In [21], an image restoration algorithm is developed to apply adaptive priors based on texture contents. The spectrum based algorithms [8], [24], [25] exploit the property that the power spectra of the sharp natural images can be well modeled by statistical regularities. Thus, the power spectrum of a blur kernel can be estimated from that of a blurry image and the

blur kernel is recovered using a phase retrieval scheme. Recently, an adaptive sparse prior is used in a multi-image deblurring framework, where a coupled penalty is shown to handle the local minimum problem effectively [26].

In this work, we exploit rich information contained in light streaks for blur kernel estimation. We note that there has been a limited amount of work that utilizes light streaks for image deblurring. The most related work to ours is an interactive deblurring method proposed by Hua and Low [27] in which a light streak region needs to be manually selected for blur kernel estimation. Since no other image structures are used, the blur kernel generated from a small cropped region may not be optimal for the entire image especially for non-uniform blur cases. In contrast, our method automatically detects and incorporates multiple light streaks in a principled optimization framework for more accurately estimating blur kernels.

Numerous methods model blurry images as the results of camera shakes with only translation. In practice, blurry images can be better modeled with spatially variant kernels. This problem has attracted significant attention due to its wide range of practical applications [20], [28], [29], [30], [31], [32], [33], [34], [35]. In [32], [35], geometric approaches are proposed to model an observed blurry image as the integration of all the intermediate images captured by the camera along the motion trajectory. These intermediate images are modeled as transformations (i.e., homographies) of the sharp image to be recovered. Based on this model, image blur caused by camera motion can be well formulated as an optimization problem. It is also possible to remove spatially varying blur by estimating a general camera motion function. A similar model has been used to model three degrees of camera motion (with in-plane translation and rotation) [33]. The optimization problems for removing spatially variant blur often require heavy computational load. Consequently, fast patch-based non-uniform deblurring methods have been developed [34], [36]. In [37], the back projection technique is used to initialize the camera motion from estimated kernels of several image patches for fast convergence. In this work, we present an efficient non-uniform blur extension of the proposed method with light streaks.

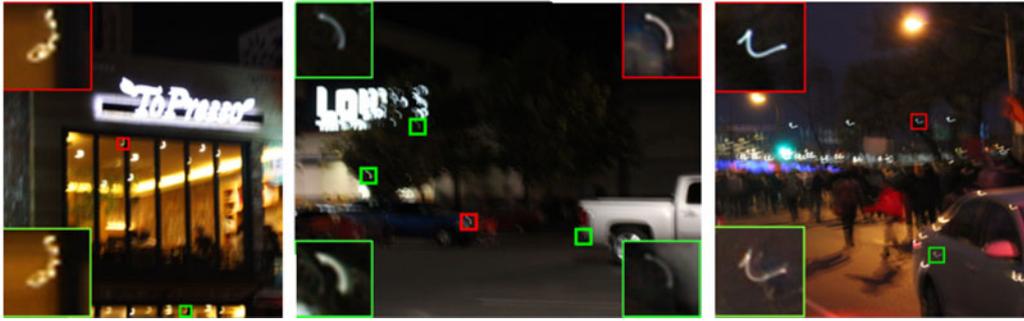


Fig. 2. Examples of light streak detection. The red box indicates the best light streak patch and the green boxes show additional light streak patches that are automatically identified by the proposed algorithm.

### 3 LIGHT STREAK DETECTION

To detect light streaks in a blurry image for kernel estimation, we first detect a set of candidate image patches. In principle, we can select a light streak that is the most similar to the underlying blur kernel based on power spectrum. In practice, we use multiple light streaks for robust kernel estimation. This is due to that one light streak may be fully or partially saturated, and may contain only limited information of the blur kernel. By using multiple light streaks, we can cumulatively extract more information. Furthermore, using multiple light streaks from different parts of the image helps alleviate issues with image noise and local distortions.

#### 3.1 Identifying Candidate Patches

We first identify a number of image patches that may contain light streaks from a blurry input. We use the following physical properties to determine *good* light streak patches: (1) pixels covered by a light streak should have relatively high intensities and those on the background should have relatively low intensities in a local neighborhood; (2) high intensity pixels in a light streak patch should have a very sparse distribution; (3) a light streak should be located near the center of a patch; and (4) there should be no other image structures in a light streak patch.

Based on these properties, we apply a set of heuristic filters to remove irrelevant image patches through the following steps. First, we take all the patches centered at all the pixels in the input blurry image, and apply two thresholds of maximum image intensity and gradient magnitude to them in order to rule out dark and flat patches based on the first property. The thresholds are set adaptively based on the global statistics, e.g., top 10 percent pixels are above the thresholds. According to the second property, we discard those patches that contain many high intensity pixels (e.g., more than 15 percent). Based on the remaining two properties, we divide each patch into the center region whose size is half of the original patch and the border region. We then compute the number of pixels with either high intensity or high gradient magnitude (above the thresholds) in the border region, and normalize it by the number computed from the center region. If the ratio is higher than a threshold (e.g., 30 percent), we discard the patch. We denote a set of light streak patches remaining after this step by  $\mathbb{P}_{\text{init}}$ . This entire process can be implemented by applying a set of simple image filters and thresholding the whole image. It can quickly remove most irrelevant patches in the image (e.g.,

more than 99 percent patches for low light images) and retain only a small amount of candidate patches (e.g., less than 1 percent patches) for further analysis.

#### 3.2 Determining the Best Light Streak Patch

From  $\mathbb{P}_{\text{init}}$ , we find the patch that best resembles the blur kernel of an input blurry image. Intuitively, the best light streak patch should contain a well-lit trajectory that has roughly the same shape as the blur kernel. This means the light source needs to be small as well as in-focus, and separated from other image structures.

We use the method proposed by [8] to obtain a good approximate power spectrum of the unknown blur kernel from a blurry image. Specifically, we first define a metric based on power spectrum to select the best light streak. The power law of natural images shows [8], [38], [39]

$$|\hat{I}(\omega)|^2 \propto \|\omega\|^{-\beta}, \quad (2)$$

where  $\hat{I}$  is the Fourier transform of an image  $I$ ,  $\omega$  is the coordinate in the frequency domain, and  $\beta$  is approximately 2. It is well known that a Laplacian filter is a good approximation to  $\|\omega\|^{-\beta}$  such that

$$|\hat{I}(\omega)|^2 |\hat{L}(\omega)| \approx C, \quad (3)$$

where  $L$  is a Laplacian filter and  $C$  is a constant. For a blurry image  $B = K * I + N$ , we have

$$|\hat{B}(\omega)|^2 |\hat{L}(\omega)| \approx |\hat{I}(\omega)|^2 |\hat{K}(\omega)|^2 |\hat{L}(\omega)| \approx C |\hat{K}(\omega)|^2. \quad (4)$$

In the spatial domain, we have  $B \otimes B * L \approx C(K \otimes K)$ , where  $\otimes$  is a correlation operator. Based on this, we define a metric

$$d(P, B) = \min_C \|B \otimes B * L - C(P \otimes P)\|^2, \quad (5)$$

where  $P$  is a candidate light streak patch. The optimal  $C$  can be computed by solving a least squares problem. Among all the candidate patches, we select the one with the smallest distance as the best light streak patch,  $P_{\text{best}} = \arg \min_P d(P, B)$ . Note that this method naturally favors unsaturated light streaks, as saturated ones would result in larger distance values. On the other hand, this method may still find a saturated light streak when it simply minimizes the above objective function. Fig. 2 shows some examples of the best light streak patch selected by the proposed method.

### 3.3 Finding Additional Light Streak Patches

Next, we use the selected best light streak patch to find additional ones from the initial candidate set  $\mathbb{P}_{\text{init}}$  based on the euclidean distance between the candidate patch  $P$  and the best patch  $P_{\text{best}}$ . The histogram of each candidate patch is normalized in order to account for the intensity difference in dark background and bright foreground from different light streak regions. We find light streak patches with the distance to the best light streak patch smaller than a certain threshold, and define a set of detected light streak patches as:  $\mathbb{P} = \{P_i\}_{i=1}^{N_P}$ . In this work, we use a threshold of  $0.13 S_P \max(P_{\text{best}})$  where  $S_P$  is the side length of a square patch. The threshold enforces the average pixel difference between the best light streak patch and additional ones to be less than  $\frac{1}{60} \max(P_{\text{best}})$ , which is empirically small enough to select similar light streak patches. Fig. 2 shows some examples of detected light streak patches.

## 4 BLUR KERNEL ESTIMATION

Once the light streaks are extracted, we estimate a blur kernel using the light streaks as additional cues. In this section, we describe how to estimate a blur kernel using the detected light streaks and other image structures for uniform image deblurring. We also describe our extension to handle spatially variant cases in Section 6.

While the conventional blur model in (1) and sparse image priors have been widely used in previous works, they are not effective to model a low-light image with light streaks because of saturated pixels and small-scale high-contrast edges around light streaks as discussed in Section 1. In order to circumvent such difficulties, we propose to separately model image structures and light streak patches. Specifically, we divide the pixels in an observed image  $B$  into three complementary sets  $\mathbb{B}^p$ ,  $\mathbb{B}^r$  and  $\mathbb{B}^s$ , which are defined as  $\mathbb{B}^p = \{x|x \in P_i \exists i\}$ ,  $\mathbb{B}^r = \{x|B(x) \text{ is not saturated} \wedge x \notin P_i \forall i\}$ , and  $\mathbb{B}^s = \{x|B(x) \text{ is saturated} \wedge x \notin P_i \forall i\}$ . These three sets represent light streaks, unsaturated, and saturated regions, respectively. We denote  $B^*$  as the complementary image corresponding to each complementary set  $\mathbb{B}^*$ ,  $\star \in \{p, r, s\}$ , and assign each  $B^*$  a binary mask  $M^*$  such that  $B^* = M^* \odot B$ . Here  $\odot$  denotes pixelwise multiplication.

As such, we introduce a more accurate, nonlinear blur model for low-light images

$$\begin{cases} B^p = \sum T_i P_i \\ B^r = M^r \odot (K * I + N) \\ B^s = M^s \odot c(K * I + N), \end{cases} \quad (6)$$

where  $c$  is a clipping function defined as  $c(v) = v$  if  $v$  is in the dynamic range of the camera sensor, and  $c(v) = 0$  or 1 otherwise (we use the dynamic range normalized to  $[0, 1]$  in this work). In (6),  $P_i$  is the light streak patch and  $T_i$  is a matrix-form transformation to insert patch  $P_i$  in the image domain, with value 0 elsewhere other than the patch location.

We use  $\hat{P}_i$  to denote the unclipped light streaks such that  $P_i = c(\hat{P}_i)$ . We introduce an auxiliary variable  $D_i$  to describe the appearance of the original point light source that produces the light streak  $P_i$ . We further assume each point light source has a disk shape, but may have different size and a different

intensity value. In this work, these variables are estimated as well. Specifically, each light streak is modeled as

$$\hat{P}_i = K * D_i + N. \quad (7)$$

Given the above model, we determine the  $K$ ,  $D_i$ , and  $I$  that can best describe the observed image and detected light streaks. This is carried out by the widely-used alternating optimization approach. Given the initial values of the three variables, we fix two of them at each time and optimize the remaining one.

### 4.1 Updating $K$

In this step, we fix  $D_i$  as well as  $I$ , and update  $K$  by optimizing the following energy function

$$\begin{aligned} f_K(K) = & \sum_{x \in \mathbb{B}^r} |(\partial_h B)(x) - (K * G_h)(x)|^2 \\ & + \sum_{x \in \mathbb{B}^r} |(\partial_v B)(x) - (K * G_v)(x)|^2 + \lambda \|K\|_1 \quad (8) \\ & + \mu \sum_{P_i \in \mathbb{P}} \sum_{x \in P_i} |(D_i * K)(x) - \hat{P}_i(x)|^2, \end{aligned}$$

where  $x$  is the pixel index. The first two terms on the right hand side are data terms based on the prediction scheme proposed by Cho and Lee [4]. In addition,  $\partial_h$  and  $\partial_v$  are partial differential operators along the horizontal and vertical axes, respectively, and  $G_h$  as well as  $G_v$  are predicted gradient maps along the two axes, respectively. Details on how to compute  $G_h$  and  $G_v$  using filters can be found in [4]. Note that the first two terms are applied only on pixels in  $M^r$  and are affected by neither saturated pixels nor light streaks. The third term is the prior on  $K$ , and the last term is derived from (7). It should be noted that, in contrast to existing edge-based approaches, such as [4] and [9] where edge extraction is inevitably affected by saturated regions and light streaks, we exclude them in the kernel update procedure.

Optimizing (8) requires to know unclipped light streaks  $\hat{P}_i$ , which are not available due to the limited dynamic range of camera sensors. Instead, we compute an approximation of  $\hat{P}_i$  before the blur kernel estimation step as described in Section 4.4. The energy in (8) is minimized using an iterative reweighted least squares method (IRLS). Here we set  $\mu$  to  $S_I/(S_P N_P)$  at the beginning, where  $S_I$  is the image size. We reduce  $\mu$  with a factor 0.75 over iterations to rely more on the data error term. We set  $\lambda$  as  $\sigma^2 S_P^2/50^2$  with  $\sigma$  denoting the noise deviation of the Gaussian prior.

### 4.2 Updating $D_i$

For each selected light streak patch, we estimate its original light source. As mentioned above, we assume that the original point light  $D_i$  has a disk shape, and its size and intensity may vary. Thus, we model  $D_i$  as a function of two parameters  $t_i$  and  $r_i$ , which denote the intensity value and the radius of the disk, respectively. Note that  $t_i$  is not restricted to the dynamic range of the image. We then define an energy function for this step as

$$f_{D_i}(t_i, r_i) = \|D_i(t_i, r_i) * K - \hat{P}_i\|^2 + \|D_i(t_i, r_i) - I_i\|^2, \quad (9)$$

where  $I_i$  is the patch in the latent image  $I$  covering the same pixels as  $P_i$ . Since we have strong prior knowledge about the light sources, e.g., they are usually small with high

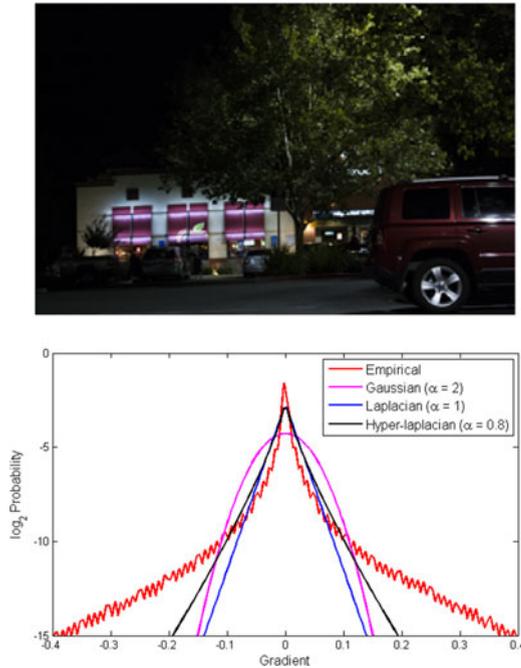


Fig. 3. Distribution fitting to empirical image gradient statistics of a typical low-light image (top) with Gaussian  $\alpha = 2$ , Laplacian  $\alpha = 1$  and hyper-Laplacian  $\alpha = 0.8$ .

intensity values, we sample a discrete set of possible  $t_i$  and  $r_i$  values, and determine the optimal one that minimizes  $f_{D_i}(t_i, r_i)$ . In practice, this exhaustive search performs well in both synthetic and real examples.

### 4.3 Updating $I$

We update the latent image  $I$  using the updated blur kernel  $K$  and light sources  $D_i$  by optimizing the following energy function

$$f_I(I) = \sum_i \mu \|D_i - I_i\|^2 + \sum_{x \in \mathbb{B}^f} |B(x) - (K * I)(x)|^2 + \gamma \sum_x (|\partial_h I(x)|^\alpha + |\partial_v I(x)|^\alpha), \quad (10)$$

where the first term corresponds to the second term in (9). The second term is the data term derived from the blur model. The third term is the sparse prior proposed in [40]. We set  $\mu$  in the same way as in (8), and  $\gamma$  is empirically set to be 0.005 in our experiments. We set the sparse prior  $\alpha = 0.8$  based on natural image statistics. As shown in Fig. 3, the hyper-Laplacian distribution with exponent  $\alpha = 0.8$  fits the major portion of small gradients well, but fails to simulate the tail. That is, the sparse prior does not regularize well in high contrast regions, e.g., regions around light sources and saturated pixels. However, the term on estimated light sources helps alleviate this issue. We use a more sophisticated algorithm to better handle saturated regions in the final deconvolution step (see Section 5). We solve (10) using the IRLS method.

### 4.4 Initialization and Implementation Details

To compute the unclipped light streak patch  $\hat{P}_i$  in (8), we need to estimate the original intensity values of the saturated pixels from the clipped light streak  $P_i$ . We apply 1D spline interpolation along horizontal and vertical axes separately to saturated pixels (e.g., using the MATLAB function `interp1`) and use

the average of the interpolated values to replace the saturated pixels. Note that, despite the simplicity of the spline approximation, it works well in our experiments because we use multiple light streaks. Light streak pixels often have locally smooth intensity values as shown in Fig. 2, and we also use other image structures. For fully saturated light streak patches, the interpolation-based approximation method may fail to recover unclipped light streak patches. However, in such cases, the fully saturated light streaks still contain sufficient shape information for initialization. Furthermore, the optimization process (8) compensates the light streak term with the data terms based on the blur model.

To update  $D_i$ , we sample a discrete set  $\{100/255, 120/255, 140/255, \dots, 1000/255\}$  for  $t_i$  and  $\{1, 2, 3\}$  for  $r_i$ . Strictly speaking, the maximum value  $1000/255$  may not be large enough for modeling the real intensity values of point light sources in real images. However, the maximum value is sufficient for the task considered in this work. Since our light streak detection naturally favors unsaturated light streaks, most of the underlying light sources  $D_i$  are not heavily exposed, i.e., the original intensity values are not large. Even when  $t_i$  is smaller than the true intensity value, it does not affect the shape of the estimated blur kernel. In such cases, the pixels in the estimated blur kernel have larger intensity values, which are then normalized after blur kernel estimation.

For each image, we estimate the blur kernel in the original resolution without the coarse-to-fine strategy. In the first iteration, we compute  $K$  by taking out the first and second terms in (8) and only considering the best detected light streak patch as

$$\arg \min_K \lambda \|K\|_1 + \mu \sum_{x \in P_1} |(D_1 * K)(x) - \hat{P}_1(x)|^2, \quad (11)$$

where we initialize  $D_i$  as the point light source with  $r_i = 1$  and  $t_i = \max \hat{P}_i$ . Given the initial  $K$  and  $D_i$ , we compute  $I$ , and iteratively update all three terms.

## 5 DECONVOLUTION WITH KERNELS

We use a non-blind deconvolution method to restore the latent contents after the blur kernel is estimated. As low-light images often contain numerous saturated pixels, they need to be handled properly to minimize ringing artifacts in the restored results.

Several approaches for handling saturated pixels in non-blind deconvolution have been proposed. Cho et al. [15] present a blur model that explicitly models outliers including saturated pixels, and use an expectation-maximization (EM) method to generate the final image. Whyte et al. [17] develop a modified Richardson-Lucy algorithm based on a blur model with a saturation function. The method by Whyte et al. includes an additional scheme to reduce ringing effects, and the approach by Cho et al. can handle other types of outliers. In this work, we describe a deconvolution algorithm that combines the advantages of these two methods.

We use the Richardson-Lucy deconvolution method as it is effective in suppressing ringing artifacts. In the Richardson-Lucy deconvolution method, the latent image  $I$  is estimated by maximizing the likelihood  $p(B|K, I)$  which is defined by a Poisson distribution. The update equation of

the deconvolution method is derived by differentiating the log-likelihood with respect to  $I$

$$I^{t+1} = I^t \odot K \otimes \frac{B}{I^t * K}, \quad (12)$$

where  $I^t$  is the latent image estimate at the  $t$ th iteration. The division operation is pixel-wise.

To better handle outliers and saturated pixels, we formulate the non-blind deconvolution process as a maximum a posteriori (MAP) problem in a way similar to the approach by Cho et al. [15]

$$p(I|B, K) \propto \sum_{M \in \mathbb{M}} p(B|M, K, I)p(M|K, I)p(I), \quad (13)$$

where  $M$  is a mask for specifying inliers and outliers, i.e.,  $M(x) = 1$  if  $B(x)$  is an inlier, and  $M(x) = 0$  if  $B(x)$  is an outlier. In addition,  $\mathbb{M}$  is a set of all possible  $M$ . We use a Poisson distribution to model inliers and a uniform distribution for outliers. The likelihood term  $p(B|M, K, I)$  is defined as  $P(B(x)|M, K, I) = \mathcal{P}(B(x)|K * I(x))$  if  $M(x) = 1$  and  $P(B(x)|M, K, I) = w$  otherwise, where  $\mathcal{P}$  is a Poisson distribution and  $w$  is a constant defined as the inverse of the width of the dynamic range. We define  $P(M|K, I)$  and  $P(I)$  in the same way as detailed by Cho et al. [15].

Given the above formulations, we derive an EM-based regularized deconvolution method. The E-step computes pixelwise weights  $W^t$  at the  $t$ th iteration as

$$W^t = \frac{\mathcal{P}(B|K * I^t)P_{\text{in}}}{\mathcal{P}(B|K * I^t)P_{\text{in}} + C(1 - P_{\text{in}})}, \quad (14)$$

where  $P_{\text{in}} \in [0, 1]$  is the probability that  $B_x$  is an inlier. The M-step updates the latent image  $I$  as

$$I^{t+1} = \frac{I^t}{1 + \varphi \rho(I^t)} \odot K \otimes \left( \frac{B \odot W^t}{I^t * K} + 1 - W^t \right), \quad (15)$$

where  $\rho(I)$  is the derivative of a sparse prior defined by

$$\rho(I) = \text{sign}(\partial_h I) \alpha |\partial_h I|^{\alpha-1} + \text{sign}(\partial_v I) \alpha |\partial_v I|^{\alpha-1}. \quad (16)$$

We set  $\alpha$  to be 0.8 and set  $\varphi$  according to the noise level as in [15], and the deconvolution process is carried out by solving (14) and (15) alternatively.

For computational efficiency, we approximate (14) using a Gaussian distribution and have

$$W^t = \frac{\mathcal{N}(B|K * I^t)P_{\text{in}}}{\mathcal{N}(B|K * I^t)P_{\text{in}} + C(1 - P_{\text{in}})}. \quad (17)$$

We typically use 40 iterations to obtain the deconvolution results. To further suppress ringing artifacts, we adopt the scheme by Whyte et al. [41] and decompose an image into unsaturated and saturated regions before performing deconvolution separately.

## 6 NON-UNIFORM DEBLURRING

In the previous sections, we describe a deblurring algorithm using a uniform blur model with light streaks. In practice, camera shakes cause spatially variant blur effects, in which the blur shapes at different locations are different. In this

section, we show how to extend the proposed method to non-uniform image deblurring.

Since the blur kernels and light streaks at different locations can vary dramatically (see Fig. 10a for an example), the power spectra of different image regions are significantly different. As a result, the power spectrum of the whole image does not necessarily approximate to that of any local blur kernel, which is the assumption in the uniform blur situation. Simply applying the proposed uniform method presented in Section 3 to non-uniform cases using the power spectrum of the whole image may lead to high false positive rate of light streak detection. Instead, we may split the image into small regions and treat each region as the uniform blur case, and apply our uniform deblurring method independently to each region. However, this approach does not work well either, as it does not consider the underlying geometric constraints of the camera motion on blur kernels of different regions.

### 6.1 Light Streak Detection in Non-Uniform Cases

To overcome the aforementioned issues for non-uniform deblurring, we adopt a local strategy to detect and use light streaks to constrain the underlying camera motion. We first split the image into a grid of tiled regions to detect light streaks. Each region is modeled by a uniform blur kernel and we apply the power spectrum method to detect the best light streak in each region. However, there may be false alarms for some regions even though they do not contain any light streaks or image structures as shown in Fig. 4a. Thus, we utilize cross validation<sup>1</sup> as light streaks at different tiles are similar in terms of shape and value distribution although they appear spatially variant. For cross validation, we use the cross-correlation based similarity metric proposed in [12] as it better captures similarity between spatially-varying PSFs than the euclidean distance as shown in [12]. We measure the similarity between two light streaks and if the similarity is higher than a certain threshold, we consider them to be similar. If a detected light streak has few (e.g., less than two) similar light streaks from other tiles, it is considered a false alarm. We set the threshold for the similarity measure to 0.75 in our work. With this validation step, we obtain light streak detection with a low false positive rate for non-uniform cases as shown in Fig. 4b.

It is worth noticing that our cross validation step requires at least three detected light streaks to perform properly. In addition, the proposed algorithm is able estimate non-uniform blur kernels more robustly and accurately when more well-distributed light strakes can be detected. With fewer than three valid light streaks after cross validation, our blur estimation process will fall back to conventional non-uniform blur estimation without light streaks.

### 6.2 Non-Uniform Model with Light Streaks

With detected light streaks, we can add kernel constraints to any geometric non-uniform blur model [32], [33], [34], [35]. In this work, we use the method by Hirsch et al. [34] in which the spatial-variant blurry image is modeled as the sum of several blurry patches by

1. In this paper, we use the term to indicate cross-checking the validity of light streaks using their neighbors, instead of the meaning used in machine learning.

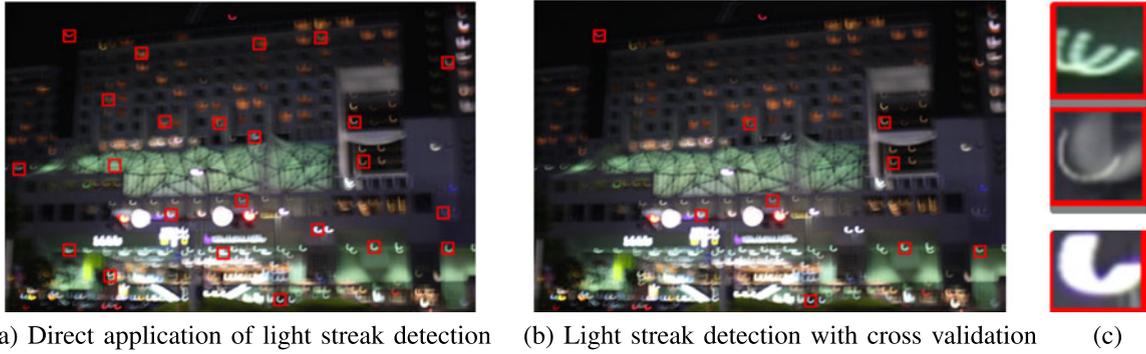


Fig. 4. Example of light streak detection for non-uniform blurred images. (a) A spatial-variant blurry image with direct application of the light streak detection method to tiled regions. (b) A spatial-variant blurry image with the light streak detection method and cross validation. (c) imperfect light streaks that are detected in (a) but are removed by cross validation in (b). False positives can be removed by cross validation of detected light streaks.

$$B = \tau \diamond I \triangleq \sum_r K^r * (w^r \odot I) + N, \quad (18)$$

where  $\tau \diamond I$  denotes the model by Hirsch et al.,  $K^r$  represents the blur kernel for the  $r$ th region, and  $w^r$  is the weight for the  $r$ th region. The kernel  $K^r$  is constrained within the geometric model as  $K^r = \sum_\theta \tau_\theta b_\theta^r$ , with  $b_\theta^r$  and  $\tau_\theta$  denoting the kernel basis in the  $r$ th region and its corresponding weight. Each kernel basis  $b_\theta$  is computed by applying the corresponding homography to a grid of cells where each one corresponds to the center of a region. The weight  $\tau_\theta$  indicates the relative exposure time of the camera at pose  $\theta$ . The above model can be written as

$$B = \sum_r \left( \sum_\theta \tau_\theta b_\theta^r \right) * (w^r \odot I) + N. \quad (19)$$

Similar to (6), we have the following model for low-light images

$$\begin{cases} B^p &= \sum T_i P_i = T_i(c(\tau \diamond D)) \\ B^r &= M^r \odot (\tau \diamond I + N) \\ B^s &= M^s \odot c(\tau \diamond I + N), \end{cases} \quad (20)$$

where  $D = \sum D_i$  is the image that includes all the original light sources for detected light streaks. Here  $\tau \diamond I$  is obtained from the model defined in (18).

### 6.3 Estimation of the Blur Parameter $\tau$

To estimate  $\tau$ , we optimize the following energy function similar to (8)

$$\begin{aligned} f_\tau(\tau) &= \sum_{x \in M^r} |(\partial_h B)(x) - (\partial_h(\tau \diamond I))(x)|^2 \\ &+ \sum_{x \in M^r} |(\partial_v B)(x) - (\partial_v(\tau \diamond I))(x)|^2 \\ &+ \mu |B^p - (\tau \diamond D)|^2 + \lambda |\tau|^2, \end{aligned} \quad (21)$$

where  $\partial_h$  and  $\partial_v$  are partial differential operators along the horizontal and vertical axes, respectively. The detected light streaks are used as constraints to regularize the weight estimation of camera motion. Since the model is linear with respect to  $\tau$ , there exists matrix-vector multiplication expression  $A_h \tau = \partial_h(\tau \diamond I)$  and  $A_v \tau = \partial_v(\tau \diamond I)$  with respect to fixed  $I$ , and  $A_D \tau = \tau \diamond D$  with respect to fixed  $D$ . We minimize the above equation using conjugate gradient descent.

We update  $D_i$  with the local blur kernel inferred using  $\tau$  at the corresponding location.

### 6.4 Estimation of the Latent Image $I$

We use a patch-wise non-blind deconvolution method with overlapping ratio of 0.1 to estimate the latent image  $I$ . For each patch, we generate the PSF at the center using the estimated blur parameter  $\tau$ , and update the patch using the same energy function in (10). Once all the patches are computed, the Bartlett-Hann window function is used to blend overlapping areas for updating the latent image  $I$ . For the final deconvolution, we apply the same patch-wise strategy using the method in Section 5 for non-blind deconvolution.

## 7 EXPERIMENTAL RESULTS

The proposed algorithm is implemented in MATLAB and experiments are carried out on a computer with 1.73 GHz Core i7 CPU and 8 GB RAM. In this work, the light streak detection and kernel estimation processes are carried out on the grayscale image of each blurry input, e.g., JPEG and PNG format. For an image of  $700 \times 1000$  pixels, the light streak detection step takes about 2/5 seconds, and the kernel estimation step takes around 5/15 minutes for uniform/non-uniform cases. The source code and dataset will be made available to the public. We present experimental results under uniform blurs from Sections 7.1, 7.2, and 7.3, and results under non-uniform blurs in Section 7.4. The patch size for light streak detection is set to be the same as the blur kernel size (which can be set for different experiments). The blur kernel size in our experiments ranges from  $25 \times 25$  to  $55 \times 55$  pixels.

### 7.1 Light Streaks

To evaluate the proposed light streak detection algorithm, we collect a set of 40 natural low-light images that contain light streaks for experiments. We then visually examine the extracted light streak patches in each image to determine if the selected best light streak patches contain light streaks. In 35 out of 40 images (87.5 percent), the proposed method successfully extracts correct light streaks. Fig. 2 shows examples of our light streak detection.

We then evaluate the proposed algorithm against the method by Hua and Low [27] which estimates the blur kernel from a manually selected light streak patch. Since this method only uses one single light streak patch, we also limit

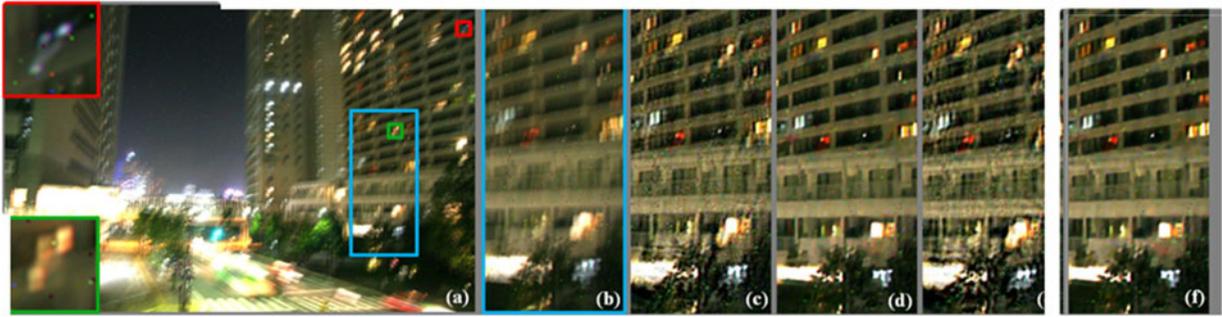


Fig. 5. A comparison with the approach by Hua and Low [27]. The image region enclosed by the red box is a manually selected light streak, and the region enclosed by the green box is determined by the proposed algorithm. (a) Input image; (b) a cropped region from (a); (c) & (e) cropped results by Hua and Low [27], using the red and green light streak patches, respectively; (d) & (f) cropped results by our method, using the red and green light streak patches, respectively.

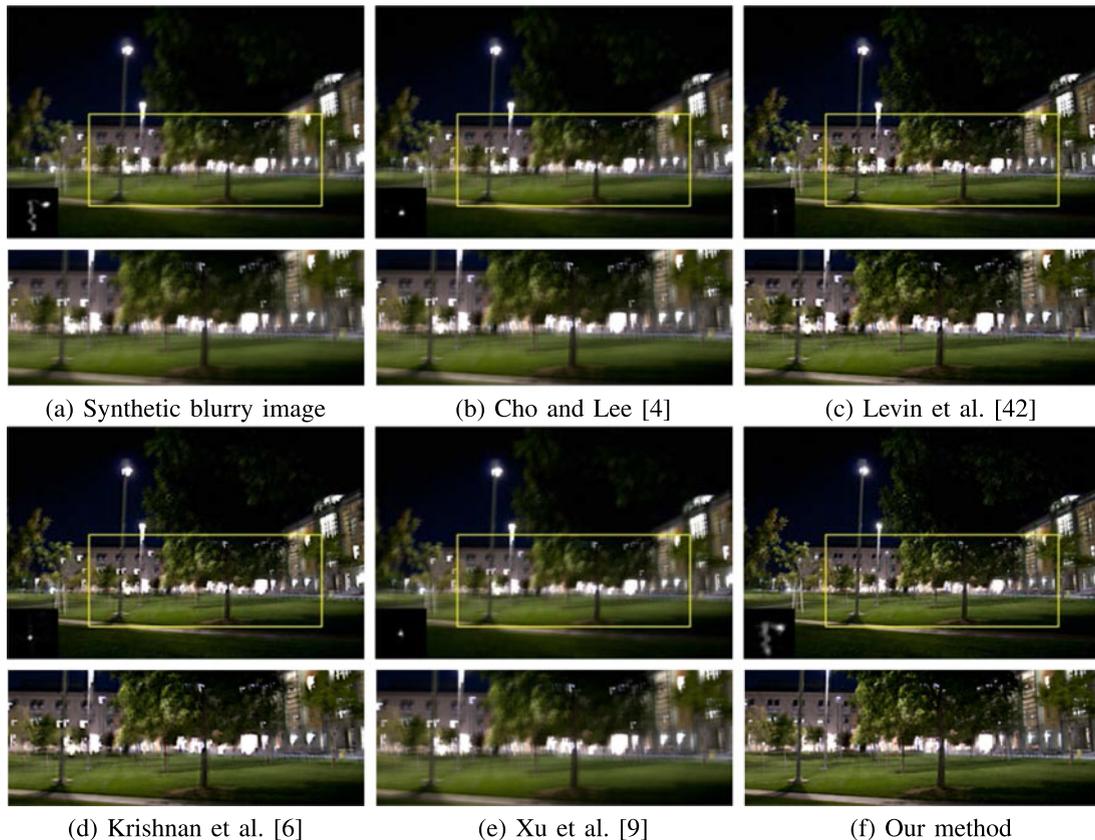


Fig. 6. Comparisons on synthetic examples of uniform blur. The images on the second and fourth rows are zoom-in views of that in the first and third rows.

the proposed algorithm to select a patch for fair comparisons. For each input image we use two different light streaks: the best patch determined by the proposed method, and a manually selected patch that is visually obvious to the user, as shown in Fig. 5.

The deblurring results show that the method by Hua and Low is sensitive to the selected input patch as it performs well with the detected non-saturated light streak by the proposed algorithm, but fails with the other manually selected image patch. As this method only relies on the light streak for extracting the blur kernel, it does not perform well when saturated pixels are included in the patch. In contrast, the proposed algorithm performs well even with the saturated patch as we also use other image structures for kernel estimation. Furthermore, with the non-saturated patch, our method

generates deblurred images of higher quality using the proposed optimization scheme. The results also demonstrate that the proposed algorithm is able to effectively select light-streak patches for deblurring. More comparisons against the method by Hua and Low on real examples are shown in Figs. 7 and 8.

## 7.2 Synthetic Images

We use a synthetic dataset of uniform blurred images for quantitative evaluation. We capture 11 low-light images in the RAW format from a variety of scenes using a Canon Rebel XSi camera with an EF-S 18-55 mm lens. For each image we apply 14 different blur kernels first, and then add Gaussian noise with 1 percent variance. We note that original noise in the RAW image is removed due to the blurring process, and thus additional noise has to be added after



Fig. 7. Comparisons with the state-of-the-art methods on a real example. The images in the second and fourth rows are zoom-in views of that in the first and third rows.

blurring to mimic the noise appearance. The whole set consists of 154 synthetic images.

Fig. 6 shows a representative example in this dataset and the uniformly deblurred images. For fair comparisons, we use the same non-blind deconvolution scheme described in Section 5 for all methods. The results show that the proposed algorithm is able to estimate more accurate kernels and deblur images well in terms of visual quality.

We compare the proposed algorithm with the state-of-art uniform deblurring methods [4], [6], [14], [42] using the metric proposed in [5]. This metric computes the relative reconstruction error, the difference between a recovered image  $I_r$  and the known ground-truth sharp image  $I_g$  over the difference between the deblurred image  $I_{k_g}$  with the ground truth kernel  $k_g$  and the ground-truth sharp image, i.e.,  $\|I_r - I_g\|^2 / \|I_{k_g} - I_g\|^2$ . Following [5], we compute the success rates of different methods based on this metric (Fig. 9), where the success rate at a given error ratio is the percentage of deblurred images with relative reconstruction error less than the given error ratio. We also compute the average kernel similarity [12] which describes the similarity between the estimated and ground-truth kernels. Table 1 shows that the proposed algorithm performs favorably against existing approaches on deblurring low-light images as a result of exploiting light streaks effectively.

### 7.3 Real Low-Light Images

We qualitatively compare the proposed algorithm with the state-of-the-art deblurring methods [4], [9] on real-world low-light images. As shown in Figs. 7 and 8, existing methods

do not perform well on the low-light images due to an insufficient number of salient edges identified for kernel estimation, and the adverse effects of light streaks discussed in Section 1 (the estimated kernels are close to delta functions). In contrast, the proposed algorithm is able to estimate more accurate kernels and generate sharper deblurred images.

### 7.4 Non-Uniform Blur

We first compare the proposed algorithm with the state-of-art uniform [4], [9] and non-uniform [41] deblurring methods on real low-light images. As shown in Fig. 10, the state-of-the-art deblurring methods do not perform well on low-light images. Although these methods are effective for deblurring generic well-lit images, the estimated kernels for low-light images are similar to delta functions as salient edges cannot be effectively extracted. In contrast, the proposed algorithm estimates non-uniform blurs well with the constraints of detected light streaks.

For comprehensive analysis, we use the same setup to [43] for our experiments. We generate 165 low-light images from 11 clear images capture at night and 15 camera motions, 3 from [43] and 12 from inertial sensors (gyroscope and accelerometer).

We compare the proposed algorithm with the state-of-art uniform [4], [6], [9], [42] and non-uniform [34], [41] deblurring methods on the dataset of 165 synthetically generated images. We use the average PSNR to evaluate the deblurring methods and present the results in Fig. 11. One example from the dataset is shown in Fig. 12. The camera motion is successfully

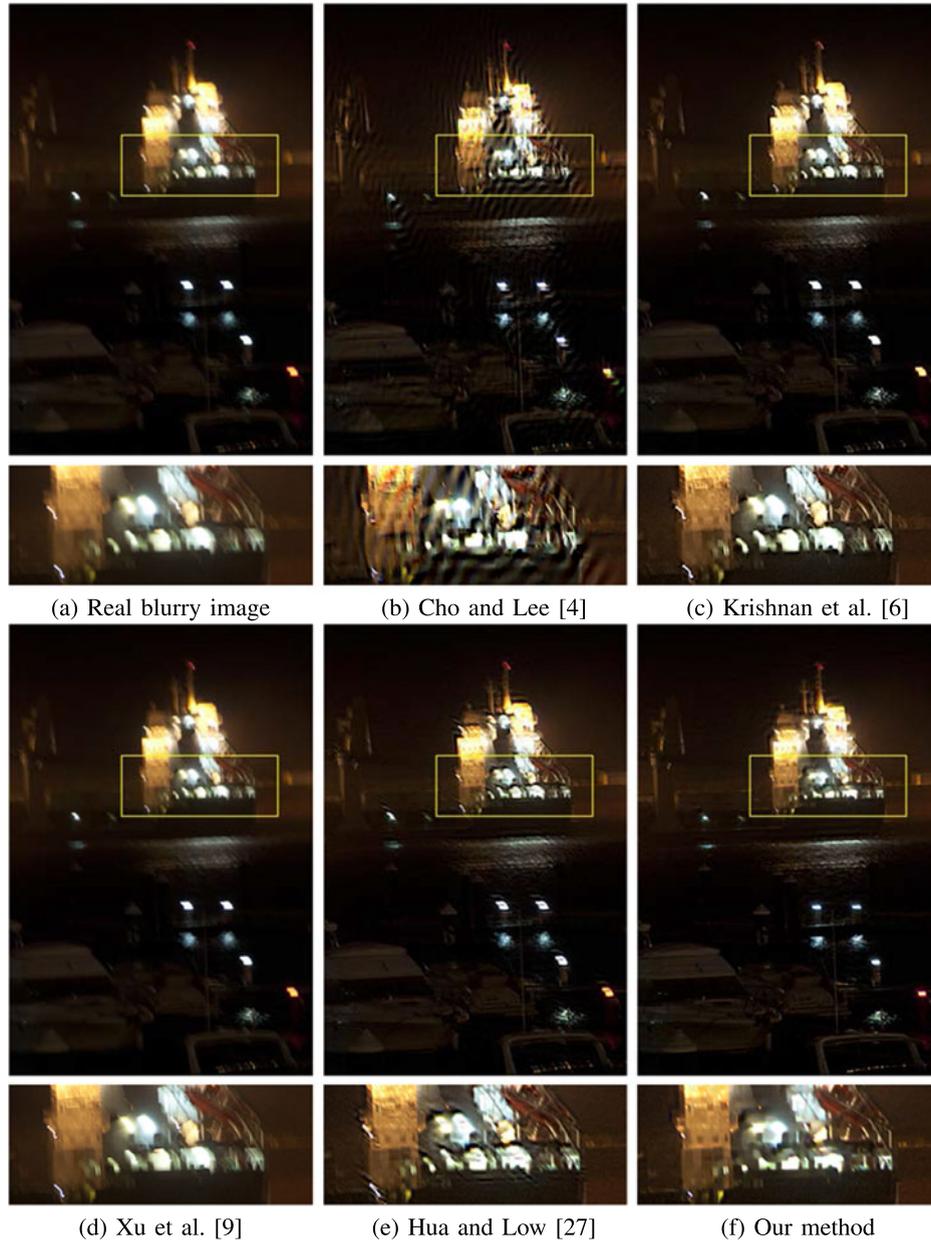


Fig. 8. Comparisons with the state-of-the-art methods on a real example. The images on the second and fourth rows are zoom-in views of that in the first and third rows.

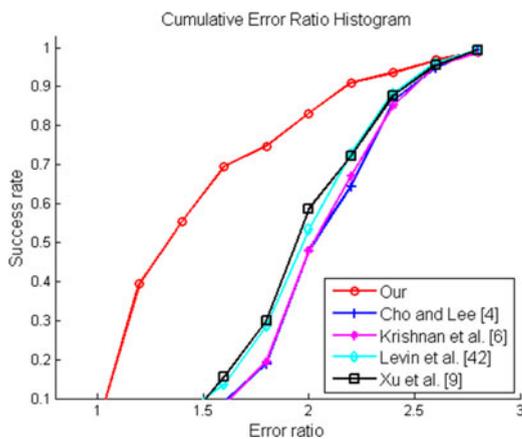


Fig. 9. Success rate of reconstruction error ratio [5] on the synthetic dataset of uniform blur. There are 154 blurry images consisting of 11 low-light images and 14 blur kernels.

estimated as the projected kernels in Fig. 12f are close to the light streaks in the blurry image (Fig. 12a). Figs. 11 and 12 also show that our non-uniform method achieves higher PSNR values and visually better results than our uniform method for non-uniformly blurred images.

### 7.5 Failure Cases

The proposed algorithm fails in some cases. One scenario is when the underlying sources of light streaks are large and cannot be modeled well with point lights, as shown in

TABLE 1  
Quantitative Comparisons Using Kernel Similarity (KS)

	Cho and Lee [4]	Krishnan et al. [6]	Levin et al. [42]	Xu et al. [9]	Ours
KS	0.5323	0.5449	0.5298	0.5312	<b>0.7069</b>

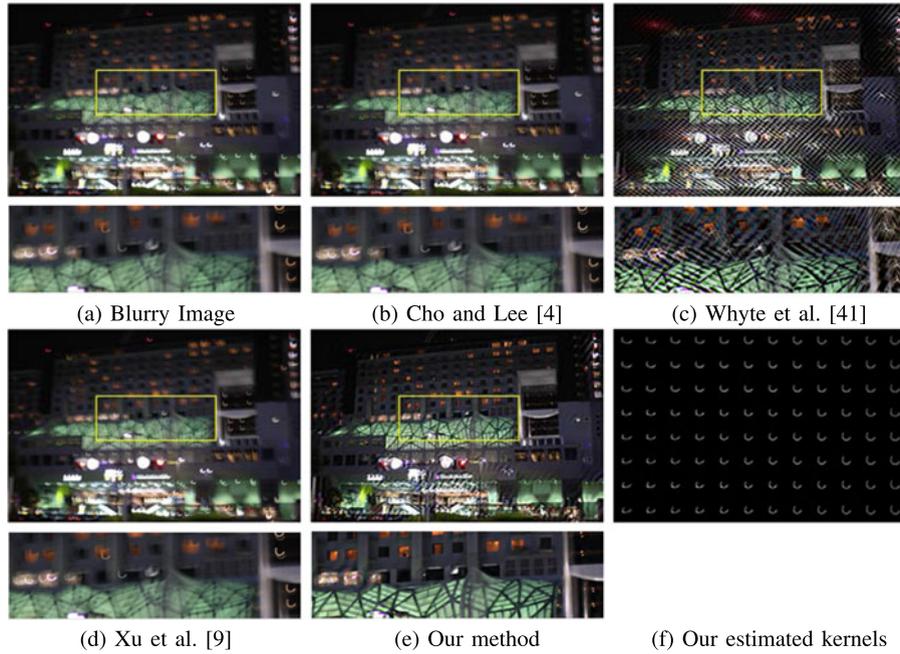


Fig. 10. Non-uniform deblurring on a real image by the evaluated methods.

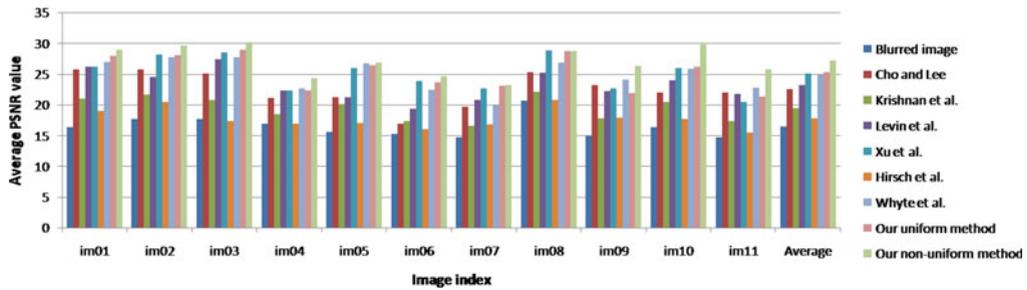


Fig. 11. Quantitative comparison on the synthetic dataset of non-uniform blurred images. There are 165 blurry images generated from 11 low-light images and 15 camera motions from [43] and data collected by inertial sensors. The  $x$ -axis denotes the image index and  $y$ -axis represents the average PSNR value. The average PSNR value over all the test images are shown on the rightmost column.

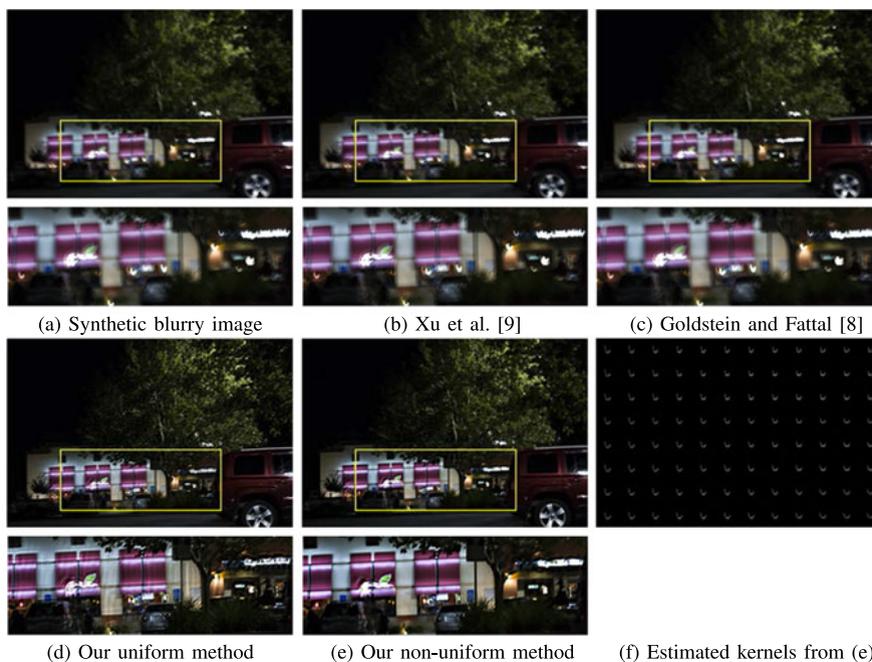


Fig. 12. An example from the synthetic dataset of non-uniform blur (the sharp image is im08 in Fig. 11). The image is generated by applying camera motion obtained from inertial sensors to a low-light image.

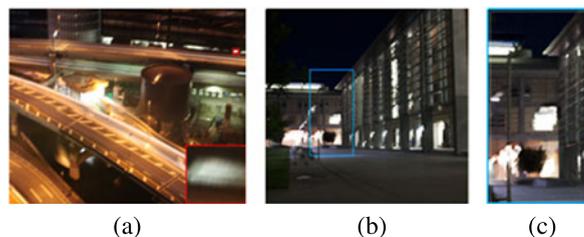


Fig. 13. Examples of failure cases. (a) a blurry image with failed light streak detection. (b)(c) a deblurred image and its cropped region using the ground truth kernel.

Fig. 13a. This type of clustered streaks may be selected by the proposed algorithm and result in an erroneous kernel estimation. This problem can be partially alleviated by manually selecting one blur patch, when available.

Another issue is related to non-blind deconvolution on a large saturated region as show in Fig. 13b. For this synthetic image, even if we supply the ground truth blur kernel, the proposed algorithm and other non-blind deconvolution methods do not generate satisfactory results due to drastic information loss, as shown in Fig. 13c.

## 8 CONCLUSIONS

In this paper we propose a deblurring algorithm that explicitly models light streaks for low-light image deblurring. The proposed method detects light streaks in blurry images and incorporates them into an optimization framework, which jointly considers light streaks and other image structures for kernel estimation. We propose a non-blind deconvolution scheme to suppress the ringing artifacts caused by light streaks. In addition, we extend the proposed algorithm to deblur low-light images undergoing non-uniform blur. Experimental results show that the proposed algorithm performs favorably against the state-of-the-art methods on deblurring low-light images.

Since the non-uniform model is more complex than the uniform one, it is usually more efficient to apply a uniform deblurring algorithm to close-to-uniform-blur scenarios. Thus, the algorithms on how to identify such cases from real blurry images can be useful for efficient image restoration in practice. Within the context of this work, one possible solution is to analyze the metric value based on power spectrum (2) of the best light streak. For an image with non-uniform blur, the power spectrum of the blurry input should have larger distance to that of any single light streak patch (if it exists) due to the varying PSFs, while in the close-to-uniform situation they are closer. Therefore, a thresholding strategy on the metric value can be used to distinguish close-to-uniform-blur scenarios from significantly non-uniform ones.

## ACKNOWLEDGMENTS

Z. Hu and M.-H. Yang are supported in part by the National Science Foundation CAREER Grant #1149783, and gifts from Adobe, NEC, Panasonic, Nvidia and STCSM Grant #16511101300. S. Cho is supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT (NRF-2017M3C4A7066316).

## REFERENCES

- [1] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," in *Proc. ACM SIGGRAPH*, 2006, pp. 787–794.
- [2] L. Yuan, J. Sun, L. Quan, and H. Shum, "Image deblurring with blurred/noisy image pairs," in *Proc. ACM SIGGRAPH*, 2007, Art. no. 1.
- [3] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," in *Proc. ACM SIGGRAPH*, 2008, Art. no. 73.
- [4] S. Cho and S. Lee, "Fast motion deblurring," in *Proc. ACM SIGGRAPH Asia*, 2009, Art. no. 145.
- [5] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1964–1971.
- [6] D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 233–240.
- [7] J.-F. Cai, H. Ji, C. Liu, and Z. Shen, "Framelet-based blind motion deblurring from a single image," *IEEE Trans. Signal Process.*, vol. 21, no. 2, pp. 562–572, Feb. 2012.
- [8] A. Goldstein and R. Fattal, "Blur-kernel estimation from spectral irregularities," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. pp 622–635.
- [9] L. Xu, S. Zheng, and J. Jia, "Unnatural L0 sparse representation for natural image deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1107–1114.
- [10] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *Int. J. Comput. Vis.*, vol. 98, no. 2, pp. 168–186, 2012.
- [11] N. Joshi, R. Szeliski, and D. J. Kriegman, "PSF estimation using sharp edge prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
- [12] Z. Hu and M.-H. Yang, "Good regions to deblur," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 59–72.
- [13] Y.-W. Tai, et al., "Nonlinear camera response functions and image deblurring: Theoretical analysis and practice," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2498–2512, Oct. 2013.
- [14] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 157–170.
- [15] S. Cho, J. Wang, and S. Lee, "Handling outliers in non-blind image deconvolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 495–502.
- [16] S. Harmeling, S. Sra, M. Hirsch, and B. Schölkopf, "Multiframe blind deconvolution, super-resolution, and saturation correction via incremental EM," in *Proc. 17th Int. Conf. Image Process.*, 2010, pp. 3313–3316.
- [17] O. Whyte, J. Sivic, and A. Zisserman, "Deblurring shaken and partially saturated images," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2011, pp. 745–752.
- [18] R. Raskar, A. Agrawal, and J. Tumblin, "Coded exposure photography: Motion deblurring using fluttered shutter," in *Proc. ACM SIGGRAPH*, 2006, pp. 795–804.
- [19] A. Levin, "Blind motion deblurring using image statistics," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2006, pp. 841–848.
- [20] Y. W. Tai, H. Du, M. S. Brown, and S. Lin, "Image/video deblurring using a hybrid camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.
- [21] T. S. Cho, N. Joshi, C. L. Zitnick, S. B. Kang, R. Szeliski, and W. T. Freeman, "A content-aware image prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 169–176.
- [22] T. S. Cho, S. Paris, B. K. P. Horn, and W. T. Freeman, "Blur kernel estimation using the radon transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 241–248.
- [23] J. Cai, H. Ji, C. Liu, and Z. Shen, "Blind motion deblurring from a single image using sparse approximation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 104–111.
- [24] Y. Yitzhaky, I. Mor, A. Lantzman, and N. Kopeika, "Direct method for restoration of motion-blurred images," *J. Opt. Soc. America A*, vol. 15, no. 6, pp. 1512–1519, 1998.
- [25] W. Hu, J. Xue, and N. Zheng, "PSF estimation via gradient domain correlation," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 386–392, Jan. 2012.
- [26] H. Zhang, D. Wipf, and Y. Zhang, "Multi-observation blind deconvolution with an adaptive sparse prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1628–1643, Aug. 2014.

- [27] B.-S. Hua and K.-L. Low, "Interactive motion deblurring using light streaks," in *Proc. 18th IEEE Int. Conf. Image Process.*, 2011, pp. 1553–1556.
- [28] J. Bardsley, S. Jefferies, J. Nagy, and R. Plemmons, "A computational method for the restoration of images with an unknown, spatially-varying blur," *Opt. Exp.*, vol. 14, no. 5, pp. 1767–1782, 2006.
- [29] S. Cho, Y. Matsushita, and S. Lee, "Removing non-uniform motion blur from images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.
- [30] Q. Shan, W. Xiong, and J. Jia, "Rotational motion deblurring of a rigid object from a single image," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.
- [31] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski, "Image deblurring using inertial measurement sensors," in *Proc. ACM SIGGRAPH*, 2010, Art. no. 30.
- [32] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 491–498.
- [33] A. Gupta, N. Joshi, L. Zitnick, M. Cohen, and B. Curless, "Single image deblurring using motion density functions," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 171–184.
- [34] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf, "Fast removal of non-uniform camera shake," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 463–470.
- [35] Y. W. Tai, P. Tan, and M. S. Brown, "Richardson-Lucy deblurring for scenes under projective motion path," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1603–1618, Aug. 2011.
- [36] S. Harmeling, M. Hirsch, and B. Schölkopf, "Space-variant single-image blind deconvolution for removing camera shake," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 829–837.
- [37] Z. Hu and M.-H. Yang, "Fast non-uniform deblurring using constrained camera pose subspace," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 136:1–136:11.
- [38] D. J. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *J. Opt. Soc. America A*, vol. 4, pp. 2379–2394, 1987.
- [39] G. J. Burton and I. R. Moorhead, "Color and spatial structure in natural scenes," *Appl. Opt.*, vol. 26, pp. 157–170, 1987.
- [40] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," in *Proc. ACM SIGGRAPH*, 2007, Art. no. 70.
- [41] O. Whyte, J. Sivic, and A. Zisserman, "Deblurring shaken and partially saturated images," *Int. J. Comput. Vis.*, vol. 110, no. 2, pp. 185–201, 2014.
- [42] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Efficient marginal likelihood optimization in blind deconvolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 2657–2664.
- [43] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling, "Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 27–40.



**Zhe Hu** received the BS degree in Mathematics from Zhejiang University and the PhD degree in computer science from University of California, Merced, in 2009 and 2015, respectively. He is a research scientist at Hikvision Research America. He was a research scientist at Light Labs Inc. from 2015 to 2017. His research interests include computer vision, computational photography and image processing.



**Sunghyun Cho** received the BS degrees in computer science, and in mathematics from POSTECH, in 2005 and the PhD degree in computer science from POSTECH, Feb. 2012. He is an assistant professor at DGIST. Before joining DGIST, he worked for Samsung Electronics from April 2014 to April 2017, and worked as a post-doctoral research scientist at Adobe Research in Seattle from March 2012 to March 2014. He spent six months in Beijing in 2006 as an intern at Microsoft Research Asia, and also spent four months in Seattle in 2010 as an intern at Adobe Research. In 2008, he was awarded Microsoft Research Asia 2008/09 Graduate Research Fellowship Award. His research interests include computational photography, image/video processing, computer vision, computer graphics, etc.



**Jue Wang** received the BE and MSc degrees from Tsinghua University, Beijing, China, and the PhD degree in electrical engineering from the University of Washington, Seattle, Washington, in 2007. He is the Director of Megvii/Face++ Research US. He was a Principal Scientist at Adobe Research from 2007 to 2017. He received the Microsoft Research Fellowship and the Yang Research Award from the University of Washington in 2006. He is a senior member of the IEEE and the ACM. His research interests include image and video processing and computational photography.



**Ming-Hsuan Yang** received the PhD degree in computer science from the University of Illinois at Urbana-Champaign, in 2000. He is a professor in Electrical Engineering and Computer Science from the University of California, Merced. He served as an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* from 2007 to 2011, and is an associate editor of the *International Journal of Computer Vision*, the *Computer Vision and Image Understanding*, the *Image and Vision Computing*, and the *Journal of Artificial Intelligence Research*. He received the NSF CAREER award in 2012, and the Google Faculty Award in 2009. He is a senior member of the IEEE and the ACM.

▷ For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).