INTERACTIVE MOTION DEBLURRING USING LIGHT STREAKS

Binh-Son Hua and *Kok-Lim Low*

Department of Computer Science
National University of Singapore

ABSTRACT

We propose a single-image, shift-invariant motion deblurring approach where the blur kernel is directly estimated from light streaks in the blurred image. Combining with the sparsity constraint, the blur kernel can be solved quickly and accurately from a user input region containing a light streak. This kernel can then be applied to state-of-the-art single-image motion deblurring methods to restore the sharp image. As our approach does not require verification of the blur kernel against the blurred image, the deblurring can be performed quickly enough for interactive use. For example, our method can be used for interactively revealing scene details in different image regions when the motion blur is not shift-invariant.

***Index Terms*—** image restoration, deconvolution, motion deblurring, light streak.

1. INTERACTIVE MOTION DEBLURRING

Motion deblurring has been a challenging task in computer vision and image processing. During exposure, camera sensor integrates light continuously over time. When camera shake occurs, the sensor integrates several unaligned images of the scene, producing a blurred image. This motion blur is often assumed to be shift-invariant and the blurred image is modeled as a convolution between the sharp image and the blur kernel with the addition of noise:

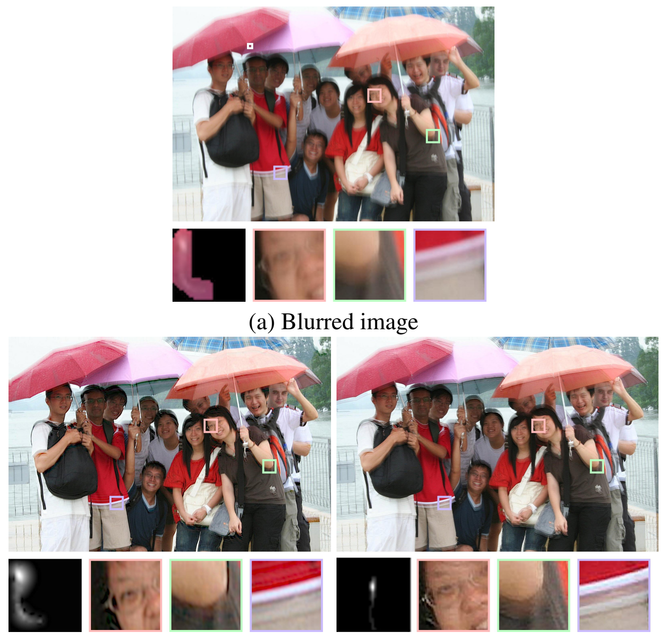
$$B = I \otimes K + N, \tag{1}$$

where B is the blurred image, I the latent sharp image, K the blur kernel, N the sensor noise, and $\otimes$ the convolution operator.

Single-image motion deblurring is a classical blind deconvolution problem in which only the blurred image B is provided as input and both the blur kernel K and the sharp image I are the unknowns. The blur kernel and latent image estimation can be formulated into a maximum-a-posteriori (MAP) framework and regularized by a variety of priors as blind deconvolution is highly ill-posed.

The MAP framework, which is turned into a convex optimization, performs an alternating optimization between the blur kernel and the latent sharp image until convergence. There are two drawbacks in this formulation. Firstly, although convolution can be computed quickly by FFT, alternating optimization is still a slow process. Secondly, as the problem is highly ill-posed, there is no guarantee that the blur kernel and the latent sharp image are correctly estimated given only a single blurred image.

In this paper, we propose an interactive motion deblurring approach that makes use of point light streaks in the blurred image to directly estimate the blur kernel. These light streaks are approximate motion paths of the camera shake and are commonly found in blurred images. Typically, they originate from distant point lights and specular highlights. For input, the user interactively selects an image region of the blurred image in which there exists a noticeable point light streak. The blur kernel is then estimated from the input region using a small-scale L1-norm optimization without the need of using alternating optimization. A high-quality latent sharp image can be estimated afterwards using a state-of-the-art non-blind motion deblurring method. With our approach, firstly, the ill-posedness of the problem is mitigated because the light streaks are good approximations of the blur kernel. Secondly, slow traditional blur kernel estimation is avoided so that deblurring can work fast enough for interactive applications.



(a) Blurred image

(b) Our result (33 sec.) (c) Shan et al. [6] (10 min.)

Fig. 1. Motion deblurring using the light streak from a specular highlight. (a) The user selected light streak is shown in the left-most subimage. (b) The extracted blur kernel is shown in the left-most subimage. (c) Result obtained by an existing state-of-the-art method.

2. RELATED WORK

Motion deblurring is a well-studied problem in the past. To reduce ill-posedness, many approaches have been proposed to utilize various sources of prior information. In multiple-image motion deblurring, an auxiliary image can be used together with the blurred image as input, e.g. blurred/noisy image pair [1], dual blurred images [2], and blurred/flash image pair [3]. An image captured from a dif-

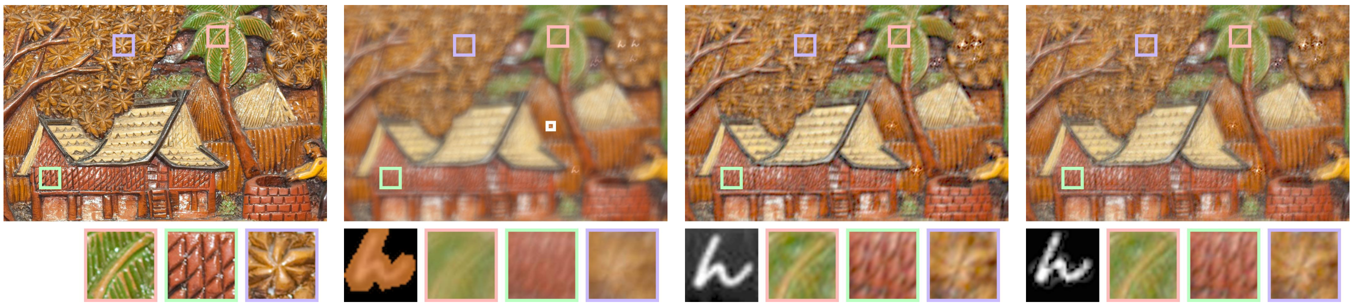


Fig. 2. Deblurring of a synthetically-blurred image containing light streaks from specular highlights. (a) The ground truth image, (b) blurred image, (c) image deblurred using ground-truth kernel, and (d) image deblurred using blur kernel extracted from a light streak.

ferent view can also be used to support blur kernel estimation [4]. While multiple-image motion deblurring can produce high-quality deblurred images, it is sometimes not applicable in practice due to the lack of auxiliary images.

In single-image motion deblurring, Fergus et al. [5] used a variational Bayes approach that approximates the kernel directly without the requirement for an approximated latent sharp image. In the alternating optimization framework, Shan et al. [6] proposed a more robust noise model based on image derivatives, and used variable splitting to optimize the blur kernel and the latent image. Recently, Cho et al. [7] proposed a fast motion deblurring method that works in the gradient domain in order to save FFT computations. These methods use a sparse prior to constrain the blur kernels and the latent sharp images.

While previous approaches try to automate the entire motion deblurring process for general scenes, our approach utilizes additional information that can be easily provided by the user. We observe that motion-blurred images of scenes with distant point lights or specular highlights often contain noticeable light streaks that may approximate the motion paths of the camera shake. Such information allows us to directly obtain the blur kernel, without the need to perform expensive alternating optimization. In comparison to multiple-image deblurring, we are similar in utilizing auxiliary information, but the information comes directly from the blurred image itself. When compared to single-image motion deblurring, our method can reduce ill-posedness, due to the availability of extra information. Moreover, our method can run much more quickly due to the avoidance of blur kernel verification against the blur model.

According to [8], blur kernels estimated by Fergus et al. [5] are more accurate than those obtained in the alternating optimization framework. Therefore, in this paper, we compare the kernel produced by our method to the kernel produced by [5]. We also compare our method to the state-of-the-art deconvolution by Shan et al. [6].

Moreover, while previous methods work well for shift-invariant blur, it has been shown in [8] that the shift-invariant blur model is often violated in practice. This has motivated recent research in spatially-varying motion deblurring, which still remains a difficult problem due to the very high ill-posedness. Thanks to the relatively short computation time, our method can be used to interactively deblur different regions in a spatially-varying blurred image.

3. OUR METHOD

Our method can be summarized in two main steps. First, in the blurred image, the user selects an image region that contains a light streak. A small-scale L1-norm optimization is performed to produce the blur kernel. In the second step, a non-blind motion deblurring is performed using the blur kernel derived in the first step to produce the final sharp image.

3.1. Light Streaks and Blur Kernels

Given a static, point-sized, bright object in the scene, the motion path of the camera shake can be easily observed in the blurred image in the form of a light streak. These bright objects must have intensities much higher than other regions in their local neighborhoods so that they can remain observable as bright light streaks even after being smeared by motion blur.

To get a light streak that approximates the blur kernel well, it is necessary that the bright object is tiny or stays far away from the camera. For example, distant street lights can be regarded as point lights and their light streaks in the blurred image are very similar to the blur kernel. Very often, specular highlights can also be utilized for the same purpose, as they are usually very small and bright.

3.2. Blur Kernel Extraction

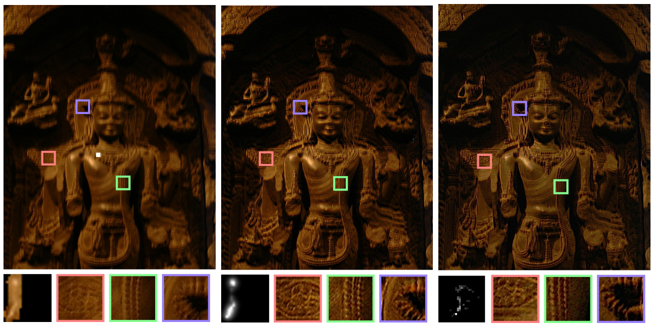
Although the light streak is clearly visible in the user input region, it has been blended with the scene background, and therefore is generally not accurate enough to produce a good deblurred result. The main objective of this step is to sparsify the light streak to produce a more accurate blur kernel, which is then used for the latent image estimation.

Let P be the image patch selected by the user that contains a light streak. In order to automatically produce the blur kernel from patch P, a small-scale optimization to constrain the kernel sparsity using L1-norm regularization is performed using the following cost function:

$$E(K) = \|L \otimes K - G\|_2^2 + \lambda \|K\|_1, \tag{2}$$

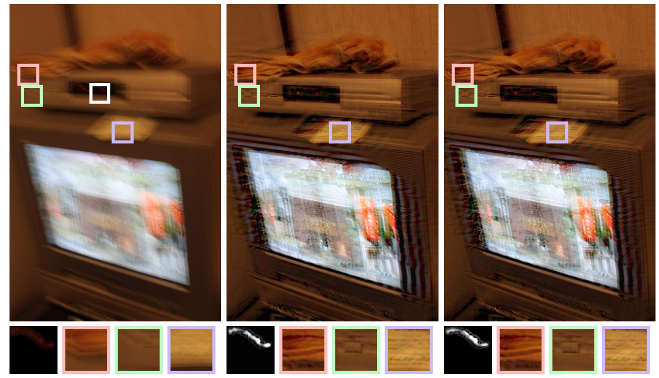
where K is the blur kernel to be estimated, L a 3×3 Laplacian filter, $G = L \otimes P$ the pre-computed derivative of patch P, and λ a scalar to control the sparseness of the kernel. The data term (the first term) constrains the derivative of the blur kernel to be similar to the derivative of the given patch P, and the regularization term sparsifies the blur kernel, which is known to be sparse. For easier implementation, we use the Laplacian filter to compute the derivative of the kernel. Our experiments have shown that replacing the Laplacian filter with other derivative filters does not affect the result much.

Here we note that although shift-invariant motion blur model is being employed, kernel estimation against the blurred image as



(a) Blurred image (b) Our result (c) Fergus et al. [5]

Fig. 3. Comparison with other deblurring methods.



(a) (b) (c)

Fig. 4. Motion deblurring using the light streak of a LED. (a) Blurred image, (b) image deblurred using extracted kernel, and (c) image deblurred using kernel improved by alternating optimization.

in the previous methods has been avoided. The correctness of the blur kernel now relies on the accuracy of the user selected region. Of course, one can easily use our extracted blur kernel to initialize the alternating optimization loop in traditional single-image motion deblurring to accelerate its convergence. We present results from this approach in Section 5.

3.3. Motion Deblurring

Following [7] for fast motion deblurring, the latent sharp image is estimated by minimizing the following cost function:

$$E(I) = \sum_i \omega_i \|\partial_i I \otimes K - \partial_i B\|_2^2 + \lambda \|\partial_i I\|_2^2, \tag{3}$$

where $i = 0..5$, $\partial_i \in \{\partial_0, \partial_x, \partial_y, \partial_{xx}, \partial_{xy}, \partial_{yy}\}$ are the first and second derivative filters, ω_i the weight for each derivative, $\|\partial_i I\|_2^2$ the Tikhonov regularization to smooth the output image, and λ a scalar value to control the degree of smoothness. The solution of the minimization in Equation (3) can be computed quickly in closed-form using FFT:

$$I = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(K)} \circ \mathcal{F}(B) \circ \Delta}{\overline{\mathcal{F}(K)} \circ \mathcal{F}(K) \circ \Delta + \lambda}\right), \tag{4}$$

where $\Delta = \sum_i \omega_i \overline{\mathcal{F}(\partial_i)} \circ \mathcal{F}(\partial_i)$, function $\mathcal{F}(\cdot)$ denotes FFT, $\circ$ denotes element-wise multiplication in the frequency domain, and $\overline{z}$ is the complex conjugate of a complex number z.

4. IMPLEMENTATION AND EXPERIMENTAL RESULTS

In our implementation, user can select an input region Ω that has an arbitrary shape. Such flexibility helps to avoid selection of other background pixels as much as possible. The square input patch P has a size derived from the bounding box of Ω, and only pixels in Ω are copied into P. The derivative of patch P is computed by convolving P with a 3×3 Laplacian filter. To avoid boundary artifacts, we set the derivative values at the boundary of Ω in P to zero.

The blur kernel optimization in Equation (2) is solved using the L1-LS package [9]. The latent image estimation is implemented using the closed-form solution in Equation (4). We noticed that boundary artifacts may occur due to the behavior of the FFT function in MATLAB, which pads the image with zeros until the image size is a power of two. We replicate the boundary values instead of zero-padding to avoid such artifacts. For kernel extraction, we set λ in the range of 0.0005 to 0.1, depending on the average intensity value of the light streak selected by the user. As for latent image estimation, we fixed λ to 0.1.

Our blur kernel extraction and latent image estimation run entirely on the CPU in MATLAB. Table 1 presents the running times of our examples on an Intel Core 2 Duo 2 GHz CPU with 4 GB of RAM. For similar image and kernel sizes, previous motion deblurring methods based on alternating optimization have running times ranging from tens of minutes to hours, e.g., the result by Shan et al. [6] in Figure 1.

In one of the experiments, we performed deblurring of a synthetically-blurred image using our method. To generate a synthetically-blurred image that contains light streaks, we manually marked a few specular highlights in the sharp image and scale up their intensities by a few times to restore part of their actual intensities in the real scene. The image was then convolved with a synthetic kernel. We then manually selected an image region that has one of the light streaks and used it for kernel extraction and latent image estimation in our method. Figure 2 shows our result compared to one that uses the ground truth kernel for deblurring.

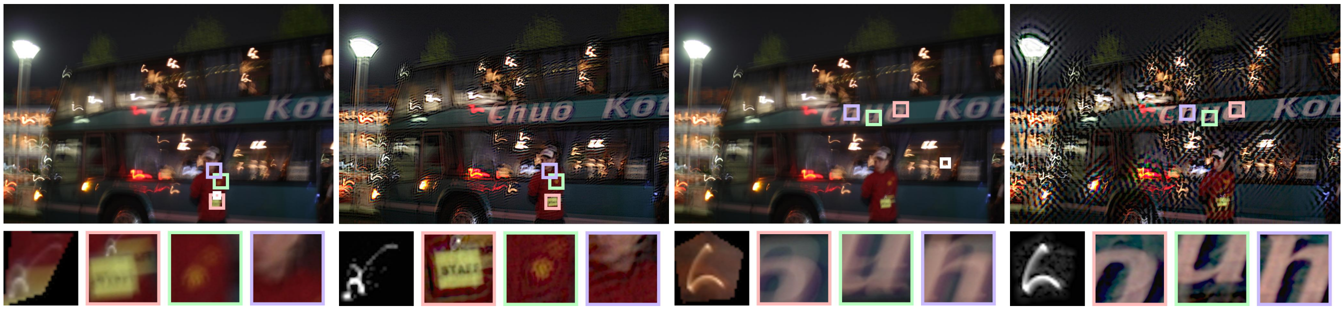
We then tested our method using photos taken in both night and day time. Light streaks in our photos are mostly from specular highlights or small light sources. Our results show that the latent sharp images can be estimated accurately. Figure 1, Figure 3, and Figure 4 show our deblurring results using light streaks in both day and night time scenes. For more deblurring examples, we refer the reader to our website[1].

Our method can produce results with quality comparable to that of the best existing deblurring methods, such as [6] and [5]. Figure 1 and Figure 3 show the deblurring results from our method and from the methods in [6] and [5]. Thanks to the specular highlights in the image, we were able to recover the sharp image without going through the slow kernel estimation process as in [6] and [5].

5. BLUR KERNEL VERIFICATION

The blur kernel extracted from a light streak in the blurred image may sometimes be a crude approximation to the ground truth blur kernel. For example, when the light streak is not thin enough, or

[1] http://www.comp.nus.edu.sg/~huabinhs/deblur



(a) Blurred image (b) Details on the man (c) Blurred image (d) Details on the bus

Fig. 5. Interactive deblurring to reveal details in different regions of an image blurred by spatially-varying motion blur.

	Size		Time (sec.)		
Figure	Image	Kernel	A	B	Total
1. chek-jawa	1600 × 1067	31 × 31	11.58	21.98	33.56
2. frame2	1024 × 683	33 × 33	1.75	10.89	12.64
3. lyndsey2	1024 × 1280	21 × 21	5.90	22.03	27.93
4. tv	683 × 1024	67 × 67	12.20	10.55	22.75
5. bus	1024 × 683	27 × 27	1.98	10.83	12.81
		35 × 35	2.76	11.26	14.02

Table 1. Running times for our examples. A: kernel extraction time. B: latent image estimation time.

when the blur kernel is extracted from a severely overexposed light streak, the kernel values may be inaccurate.

The blur kernel accuracy can be improved by using the extracted kernel and the deblurred image produced by our method to initialize the alternating optimization. Given the extracted kernel K and the deblurred image I solved from the blurred image B, the following cost function can be used to optimize K:

$$E(K) = \|I \otimes K - B\|_2^2 + \lambda \|K\|_1, \tag{5}$$

where λ is a scalar to control the sparseness of the kernel. Only new kernel values corresponding to non-zeros in the extracted kernel are used to update the blur kernel. After the new kernel K is computed, the deblurred image can be estimated using the image deblurring method in Section 3.3. The blur kernel optimization and the latent image estimation are alternately iterated until convergence. Figure 4 shows an example where the deblurred image with kernel optimization is sharper than the deblurred image with kernel directly extracted from the blurred image.

6. SPATIALLY-VARYING MOTION DEBLURRING

Although we assume the shift-invariant blur model, our method can be applied to interactively reveal local scene details in images blurred by spatially-varying motion blur. We assume that in a local neighborhood, the motion blur is shift-invariant. In order to reveal details of the scene in a local region, a light streak in the region is interactively selected for blur kernel estimation and latent image estimation. Compared to traditional approaches, our method is more convenient as no explicit image crop is required.

Figure 5 shows an example image that contains complex spatially-varying blur. Two different kernels from two regions are selected for deblurring, thus revealing details in the respective regions.

7. DISCUSSION AND CONCLUSIONS

Motion deblurring using light streaks has a few caveats. Firstly, the selected light streak may not come from a point light, causing the blur kernel to be inaccurate. Similarly, the light streak may be severely overexposed. Fortunately, our method is reasonably fast enough to allow a few rounds of trial-and-error before a good deblurred image is obtained. Secondly, kernel verification is not performed in our method in order to trade for user interactivity. This may leave out chances to further optimize for higher kernel accuracy and deblurred image quality.

For future work, we hope to develop an algorithm to automatically detect good point light streaks. We are also interested in porting our implementation to C/C++ to gain better speed.

In summary, we have proposed a motion deblurring method that can use light streaks to approximate the blur kernels. As light steaks occur commonly in both day and night time scenes, we believe our method is useful for fast motion deblurring in practice.

8. REFERENCES

[1] Lu Yuan, Jian Sun, Long Quan, and Heung-Yeung Shum, “Image deblurring with blurred/noisy image pairs,” in *SIGGRAPH*, 2007.

[2] Jia Chen, Lu Yuan, Chi-Keung Tang, and Long Quan, “Robust dual motion deblurring,” in *CVPR*, 2008.

[3] Shaojie Zhuo, Dong Guo, and Terence Sim, “Robust flash deblurring,” in *CVPR*, 2010.

[4] Lu Yuan, Jian Sun, Long Quan, and Heung-Yeung Shum, “Blurred/non-blurred image alignment using sparseness prior,” in *ICCV*, 2007.

[5] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman, “Removing camera shake from a single photograph,” in *SIGGRAPH*, 2006.

[6] Qi Shan, Jiaya Jia, and Aseem Agarwala, “High-quality motion deblurring from a single image,” in *SIGGRAPH*, 2008.

[7] Sunghyun Cho and Seungyong Lee, “Fast motion deblurring,” in *SIGGRAPH Asia*, 2009.

[8] A. Levin, Y. Weiss, F. Durand, and W.T. Freeman, “Understanding and evaluating blind deconvolution algorithms,” in *CVPR*, 2009.

[9] Seung-Jean Kim, K. Koh, M. Lustig, and S. Boyd, “An efficient method for compressed sensing,” in *ICIP*, 2007.