



Modeling nonstationary lens blur using eigen blur kernels for restoration

MOONSUNG GWAK AND SEUNGJOON YANG* 

School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology, Ulsan, Republic of Korea

*syang@unist.ac.kr

Abstract: Images acquired through a lens show nonstationary blur due to defocus and optical aberrations. This paper presents a method for accurately modeling nonstationary lens blur using eigen blur kernels obtained from samples of blur kernels through principal component analysis. Pixelwise variant nonstationary lens blur is expressed as a linear combination of stationary blur by eigen blur kernels. Operations that represent nonstationary blur can be implemented efficiently using the discrete Fourier transform. The proposed method provides a more accurate and efficient approach to modeling nonstationary blur compared with a widely used method called the efficient filter flow, which assumes stationarity within image regions. The proposed eigen blur kernel-based modeling is applied to total variation restoration of nonstationary lens blur. Accurate and efficient modeling of blur leads to improved restoration performance. The proposed method can be applied to model various nonstationary degradations of image acquisition processes, where degradation information is available only at some sparse pixel locations.

© 2020 Optical Society of America under the terms of the [OSA Open Access Publishing Agreement](#)

1. Introduction

Image acquisition processes can be modeled by the following system of linear equations:

$$\mathbf{b} = \mathbf{A}\mathbf{x} + \mathbf{v}, \quad (1)$$

where \mathbf{b} , \mathbf{x} , and \mathbf{v} are vector representations of the observed, original, and noise images, respectively. Matrix \mathbf{A} represents the degradation introduced by an image acquisition process. In this paper, we address blur introduced by a lens due to defocus and optical aberrations. The severity and characteristics of the blur can change based on the focusing accuracy and sensor pixel locations. Defocus and optical aberrations (e.g., spherical aberrations, coma, astigmatism, and field curvature) are responsible for the nonstationarity of the blur [1]. Modern lenses are designed to reduce optical aberrations through careful arrangements of optical elements and use of aspherical elements [2]. However, images acquired using modern lenses still suffer from lens blur—even when images are properly focused and acquired with no camera or subject motion [3–10]. Thus, unlike blur due to camera shake or subject motion, for which nonstationary blur is determined at exposure by the amount of motion or shake, [11,12], lens blur can be precalibrated once and subsequently used to restore photographed images [5,7,8,10].

A nonstationary blur operation can be written as a filtering operation using spatially variant blur kernels. However, constructing a matrix \mathbf{A} that represents the filtering with pixelwise variant blur kernels is difficult. Moreover, matrix \mathbf{A} that represents nonstationary blur is neither a Toeplitz nor a block circulant matrix. Matrix \mathbf{A} is not diagonalized by the discrete Fourier transform (DFT) [13,14]; consequently, any efficient restoration algorithms that rely on the diagonalization of matrix \mathbf{A} by the DFT [15–19] cannot be applied to restore nonstationary blur. In many practical situations, nonstationary blur information can be acquired through the available blur kernels, estimation, or measurements at sparse pixel locations [4,5,7,8]. The so-called efficient filter flow (EFF) [20,21] method is often adopted to model nonstationary blur by assuming stationarity inside the regions of an image where blur kernels are available. Because the stationary blur

operation inside each region can be implemented efficiently with the DFT, the EFF has been widely used to model nonstationary blur for efficient restoration [4,5,7,8,10,22,23]. However, the following problems have not been addressed: (1) the accuracy with which the EFF can model nonstationary blur and (2) how the modeling accuracy of the EFF affects restoration performance.

This paper presents an efficient modeling method to model nonstationary blur. Blur at each pixel location is expressed as a linear combination of eigen blur kernels, which are obtained through principal component analysis (PCA) of blur kernels available at sparse pixel locations. Blur operations using the eigen blur kernels can be implemented efficiently with the DFT. Moreover, the proposed method requires no region-wise processing, which can affect modeling accuracy and processing efficiency. PCA has been used to model point spread functions (psf's) of optical systems. In [24,25] a three dimensional (3D) depth-variant psf of a microscope is modeled using principal components of 3D blur kernels measured at various object distances. In astronomical imaging, psf at a star location is modeled as a linear combination of point source images, basis functions, or principal components of point source images [26,27]. We use the PCA to model spatially variant nonstationary lens blur caused by defocus and optical aberrations. Through controlled experiments, in which pixelwise nonstationary blur models were available, we showed that nonstationary lens blur could be modeled more accurately and more efficiently when using the eigen blur kernels than when using the EFF. We applied the proposed modeling method to total variation (TV) restoration [28–30] and showed that more accurate and efficient modeling leads to improved restoration performance. We also applied the proposed method to model nonstationary blur for real-world lenses to restore defocus and optical aberrations.

The proposed eigen blur kernel-based modeling method can be applied to model various nonstationary degradations that occur during image acquisition. When the degradation changes characteristics in a pixelwise fashion, and information concerning such degradation is estimated or measured at sparse pixel locations for practical consideration, the proposed method provides a method to accurately model and efficiently implement the degradation processes for accurate and efficient restoration.

The remainder of this paper is organized as follows. In Section 2.1, nonstationary lens blur due to defocus and optical aberrations is introduced. In Section 2.2, we present the modeling of nonstationary lens blur by the eigen blur kernels. We introduce TV restoration in Section 3.1. Section 3.2 describes how to implement nonstationary blur operations used in restoration efficiently with the proposed eigen blur kernels. In Sections 4.1 and 4.2, the modeling accuracy and TV restoration performance of the proposed method are analyzed through controlled experiments using Gaussian blur and thin lens blur models. Section 4.3 presents the modeling and restoration of real-world lenses. Finally, Section 5 concludes the paper.

2. Modeling of nonstationary lens blur using eigen blur kernels

2.1. Nonstationary lens blur

The image for a point light source not at the focal distance forms either in front of or behind a sensor. Consequently, the point source will appear as a circular image called the “circle of confusion.” Any object that does not provide an acceptable circle of confusion will appear blurry. The point spread function (psf) that represents the image shape of a point light source changes depending on the pixel locations, aperture size, and focusing accuracy. For example, the psf of a thin lens with a circular exit pupil of radius a at a sensor located at distance z is given by [1]

$$h(r, \theta, z) \propto \left| \int_0^1 \int_0^{2\pi} e^{i\Phi(\rho, \phi)} e^{-\pi i \frac{R}{z} \rho r \cos(\phi - \theta)} \rho d\rho d\phi \right|^2, \quad (2)$$

where (r, θ) are the locations at the sensor in polar coordinates, R represents the distance from the exit pupil to the focused image, and $\rho = r/a$. Function $\Phi(\rho, \phi)$ is called a phase aberration,

which is the sum of the primary optical aberration and defocus aberration. The primary optical aberration, which includes spherical aberration, coma, astigmatism, field curvature, and distortion, changes based on the pixel location (r, θ) . The defocus aberration changes with the location of sensor z relative to the focused image location R . The overall shape of the psf changes with the defocus, the position of z relative to R , and the spatial location of a pixel on the sensor, (r, θ) . Hence, the blur introduced by psfs is nonstationary with respect to the defocus and pixel locations.

Nonstationary blur can be represented by a filtering operation with spatially variant blur kernels [31]. Let $h(i, j, \xi, \zeta; d)$ be the blur kernel at the pixel location (i, j) in Cartesian coordinates, when the camera focus is behind the scene at distance d . Variables (ξ, ζ) are local spatial variables for the individual blur kernel. An image acquired through a lens can be modeled as follows:

$$b(i, j) = \sum_{\xi, \zeta} h(i, j, \xi, \zeta; d)x(i - \xi, j - \zeta) + v(i, j), \quad (3)$$

where b , x , and v are the observed, original, and noise images, respectively. The acquisition model is a nonstationary process that involves pixelwise variant blur kernels. When the blur is stationary, the acquisition model in (3) is reduced to the convolution by a single spatially invariant blur kernel with additive noise.

2.2. Modeling using eigen blur kernels

Accurate modeling of a nonstationary image acquisition process requires knowledge of the pixelwise variant blur kernels. In many practical situations, the blur of an image acquisition system is measured or estimated at some sparse pixel locations. In [4,5,7,8,32], the blur kernels are estimated on grid points using a test pattern. In this section, we present a modeling method that can construct an accurate nonstationary blur model from blur kernels measured or estimated at any focal distance and pixel locations. Figure 1 shows a scheme of the image acquisition system used in our modeling. A test image, a part of which is shown in Fig. 1(a), consists of repeated circular patterns. The test image is photographed using a lens, for which we build a blur model. A camera is positioned perpendicular to the test image and is focused at the focal distance s . The aperture is fixed at a given f-number. We take multiple photographs of the test image after placing the test image at a defocusing distance d without changing the focus. For example, to model the commercially available lens in Section 4.3, we take three photographs with the test image: at the focal distance, in front of the focal distance, and behind the focal distance. This procedure allows us to collect blur kernels at different defocus and pixel locations and construct a nonstationary lens blur model that accounts for both the defocus and primary optical aberrations.

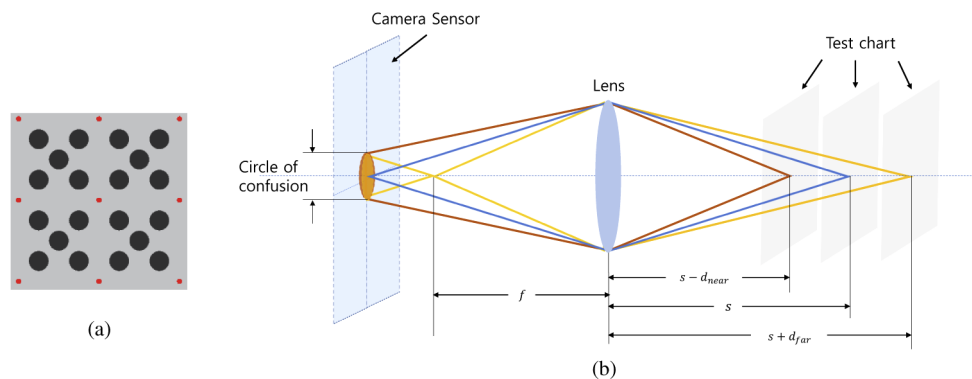


Fig. 1. Scheme of a nonstationary blur estimation: (a) test pattern; (b) estimation setup.

Patches of photographed test images containing a group of circular patterns are extracted, and blur kernels are estimated for each patch. The photographed test images are divided into patches of circular patterns, as specified by the four red dots shown in the image in Fig. 1(a). Blur is estimated in a patchwise manner, and we assume that the blur inside the patches is stationary. Under this assumption, the acquired image in a patch is the convolution of a blur kernel and the original test pattern. For the r th patch, the observed image b_r is obtained by applying a homography transform to align the four dots of the acquired image to the four dots of the original test pattern x_p . The original test pattern x_p is available from the definition of the pattern. Blur of each patch image can be estimated from the acquired and original image using a blur estimation method [4,5,7,8,10,33]. Once the blur kernels at various locations are obtained, the proposed method can be applied to model pixelwise variation of nonstationary blur. Subsequently, the modeling of nonstationary blur is independent of a patchwise blur estimation method used. In this work, the nonparametric blur kernels are found by solving the following optimization problem [4,10]:

$$\underset{\mathbf{h}_r}{\text{minimize}} \quad \|\mathbf{b}_r - \mathbf{X}_p \mathbf{h}_r\|^2 + \sum_d \mu_d \|\mathbf{C}_d \mathbf{h}_r\|^2, \quad (4)$$

where \mathbf{b}_r and \mathbf{h}_r are the vector representations of the observed image b_r and the blur kernel h_r , respectively. The elements of matrix \mathbf{X}_p are the pixel values of x_p rearranged to represent the convolution. Matrices \mathbf{C}_d are the horizontal, vertical, and diagonal difference operations, and weights μ_d are chosen to be small values. A solution is then found using the MATLAB optimization toolbox. The estimated blur kernels are postprocessed with thresholding and renormalization such that all the values are positive, and small values due to noise are removed.

A model that characterizes the shapes and spread of blur kernels of a lens can be constructed from the estimated blur kernels h_r . The principal components of the blur kernels estimated through (4) are computed using PCA. Let $e_k(\xi, \zeta)$ be the k th principal component, which we call an eigen blur kernel. We assume that eigenvalues λ_k corresponding to the eigen blur kernels $e_k(\xi, \zeta)$ are sorted in decreasing order. The blur kernel at the (i, j) th pixel location defocused by d can be approximated as a linear combination of the K eigen blur kernels:

$$h_K(i, j, \xi, \zeta; d) = \sum_{k=1}^K c_k(i, j; d) e_k(\xi, \zeta), \quad (5)$$

where $c_k(i, j; d)$ are the weights for the k th eigen blur kernel e_k .

The approximation of blur kernels by the eigen blur kernels in (5) is available only for the pixel locations (i, j) , where we estimated the blur kernel h_r . However, assuming that the blur of a lens does not change abruptly, a blur kernel at any pixel location can be represented as a weighted average of the available blur kernels in the neighborhood. Thus, the blur kernel at any (i, j) th pixel location can be written as

$$h(i, j, \xi, \zeta; d) = \sum_{s, t \in \mathcal{N}(i, j)} \phi(s, t) h_K(s, t, \xi, \zeta; d), \quad (6)$$

where $\phi(s, t)$ is the interpolation weight, and $\mathcal{N}(i, j)$ is a set of indices for the neighbors of (i, j) for which estimated blur kernels are available. Substituting (5) into (6), we obtain

$$h(i, j, \xi, \zeta; d) = \sum_{k=1}^K \tilde{c}_k(i, j; d) e_k(\xi, \zeta), \quad (7)$$

where

$$\tilde{c}_k(i, j; d) = \sum_{s, t \in \mathcal{N}(i, j)} \phi(s, t) c_k(s, t; d). \quad (8)$$

Hence, the blur kernel at any pixel location can be found by using the interpolated weights $\tilde{c}_k(i, j; d)$ and the eigen blur kernels $e_k(\xi, \zeta)$.

3. Restoration of nonstationary lens blur using eigen blur kernels

3.1. Total variation restoration

Image restoration is an inverse problem: to obtain the original image \mathbf{x} from a degraded image \mathbf{b} , which is often ill-posed [34,35]. An optimization problem with constraints that reflect our prior knowledge about images can be used to find a meaningful solution for the ill-posed problem. The solutions can be found by various methods [16,28–30].

In this paper, we consider the restoration problem given by the following optimization problem:

$$\begin{aligned} & \text{minimize}_{\mathbf{x}} && \text{TV}(\mathbf{x}) \\ & \text{subject to} && \|\mathbf{Ax} - \mathbf{b}\| \leq \epsilon, \end{aligned} \quad (9)$$

where $\text{TV}(\mathbf{x})$ is the TV. We consider three restoration methods: the primal-dual algorithm (PA) [29], NESTA [28], and KADMM [30]. The most computationally expensive algorithm operations are \mathbf{Ax} and $\mathbf{A}^H\mathbf{x}$ operations. Hence, the way these operations are implemented is vital for efficient restoration algorithms. In this study, we consider the TV restoration approach. However, the following development that shows efficient implementations of \mathbf{Ax} and $\mathbf{A}^H\mathbf{x}$ operations can be applied to any restoration methods that require these operations. For example, restoration algorithms that utilize linear operators such as $\|\mathbf{x}\|$ or $\|\mathbf{D}\mathbf{x}\|$ instead of the TV also require operations \mathbf{Ax} and $\mathbf{A}^H\mathbf{x}$. <https://www.overleaf.com/project/5f20edabff82f10001467968>

When blur is stationary, matrix \mathbf{A} represents the convolution by a stationary blur kernel and becomes Toeplitz. Moreover, given a periodic boundary condition, matrix \mathbf{A} becomes block circulant. Hence, operations \mathbf{Ax} and $\mathbf{A}^H\mathbf{x}$ can be implemented using the DFT matrix [13,14]. The operations are

$$\begin{aligned} \mathbf{Ax} &= \mathcal{F}\mathcal{A}\mathcal{F}^H\mathbf{x}, \\ \mathbf{A}^H\mathbf{x} &= \mathcal{F}\mathcal{A}^*\mathcal{F}^H\mathbf{x}, \end{aligned} \quad (10)$$

where \mathcal{A} is a diagonal matrix, \mathcal{A}^* is the complex conjugate of \mathcal{A} , and \mathcal{F} is the DFT matrix. When blur is nonstationary, matrix \mathbf{A} represents the filtering by the spatially variant blur kernels, and matrix \mathbf{A} is neither Toeplitz nor block circulant. Note that the efficient implementations of \mathbf{Ax} and $\mathbf{A}^H\mathbf{x}$ using the diagonalization by the DFT matrix are not available for nonstationary blur.

The EFF method is often used to implement the operations efficiently when blur is nonstationary [20,21]. Here, the blur in regions of an image is assumed to be stationary, and operations \mathbf{Ax} and $\mathbf{A}^H\mathbf{x}$ are approximated by

$$\begin{aligned} \mathbf{A}_{\text{eff}}\mathbf{x} &= \sum_l \mathbf{P}_l^T \mathcal{F}_l \mathcal{A}_l \mathcal{F}_l^H \mathbf{P}_l \mathbf{W}_l \mathbf{x}, \\ \mathbf{A}_{\text{eff}}^H\mathbf{x} &= \sum_l \mathbf{W}_l^T \mathbf{P}_l^T \mathcal{F}_l \mathcal{A}_l^* \mathcal{F}_l^H \mathbf{P}_l \mathbf{x}, \end{aligned} \quad (11)$$

where the sum is for the regions of image \mathbf{x} . Matrix \mathbf{W}_l is a diagonal matrix that picks up pixels in the l th region with an overlapping window, \mathbf{P}_l crops the pixels in the l th region, $\mathcal{F}_l \mathcal{A}_l \mathcal{F}_l^H$ applies the stationary blur to the cropped l th region, and \mathbf{P}_l^T inserts the results. Within the EFF, $\mathbf{A}_{\text{eff}}\mathbf{x}$ and $\mathbf{A}_{\text{eff}}^H\mathbf{x}$ operations are implemented with the DFT (applied in a regionwise fashion). Matrix \mathbf{W}_l is used as a window function to prevent ringing due to the regionwise applications of the DFT and is also responsible for smooth transitions of the blur kernels across the region boundaries. The elements in \mathbf{W}_l 's are prepared so that the sum of the elements operating on a pixel equals one. Because matrix \mathbf{W}_l serves these two purposes, designing \mathbf{W}_l to accurately approximate matrix \mathbf{A} is not straightforward.

3.2. Nonstationary blur operations using eigen blur kernels

The blur kernel at any pixel location can be written as a linear combination of K eigen blur kernels, as shown in (7), and it can be written in matrix–vector notation as follows:

$$\mathbf{A}_K = \sum_{k=1}^K \mathbf{C}_k \mathbf{E}_k, \quad (12)$$

where \mathbf{C}_k is a diagonal matrix whose elements are $\tilde{c}_k(i, j; d)$, and \mathbf{E}_k represents the convolution by the stationary eigen blur kernel e_k . Given a periodic boundary condition, matrix \mathbf{E}_k is Toeplitz and block circulant. Matrix \mathbf{E}_k is diagonalized by the DFT matrix as $\mathbf{E}_k = \mathcal{F} \mathcal{E}_k \mathcal{F}^H$, where \mathcal{E}_k is a diagonal matrix. Hence, the multiplication operations can be written as follows:

$$\begin{aligned} \mathbf{A}_K \mathbf{x} &= \sum_{k=1}^K \mathbf{C}_k \mathcal{F} \mathcal{E}_k \mathcal{F}^H \mathbf{x}, \\ \mathbf{A}_K^H \mathbf{x} &= \sum_{k=1}^K \mathcal{F} \mathcal{E}_k^* \mathcal{F}^H \mathbf{C}_k \mathbf{x}. \end{aligned} \quad (13)$$

Because \mathbf{C}_k and \mathcal{E}_k are diagonal matrices, operations $\mathbf{A}_K \mathbf{x}$ and $\mathbf{A}_K^H \mathbf{x}$ can be implemented efficiently using the DFT.

Matrix \mathbf{A} can be approximated more accurately by the eigen blur kernel-based implementation when more blur kernels are available. The interpolation of the coefficients in (8) does not require regularly sampled blur kernels. Hence, as we estimate or measure blur kernels at more pixel points, we can acquire more accurate approximations of the nonstationarity. The EFF works with blur kernels at regularly divided regions of an image. Adding more blur kernels results in smaller size regions. Overlapping the smaller regions with the window function \mathbf{W}_l , which is responsible for both overlapping and the transitions of blur kernels across the region boundaries, may hinder the approximation accuracy.

The dominant arithmetic operations of $\mathbf{A}_K \mathbf{x}$ are the computations of the DFT, which are relative to the size of the image. The computational complexity of using K eigen blur kernels for $M \times N$ size images is on the order of

$$K MN \log(MN). \quad (14)$$

For the EFF using $S_i \times S_j$ blur kernels for $M \times N$ size images, the dominant arithmetic operations of $\mathbf{A}_{\text{eff}} \mathbf{x}$ are the DFT computations of size $(M/S_i + o) \times (N/S_j + o)$, where o is the number of overlapping pixels. The computational complexity of the EFF is on the order of

$$S_i S_j (M/S_i + o)(N/S_j + o) \log(M/S_i + o)(N/S_j + o). \quad (15)$$

The computational complexity of both methods is on the order of $MN \log q$ with different q values. However, in terms of implementation, the eigen blur kernel-based method offers simpler data access than the EFF. All the operations involved in the eigen kernel-based approximation are imagewise operations; there are no regionwise accesses. In contrast, the EFF requires regionwise accesses to and from various regions of an image, which complicates the processing flow considerably.

The eigen blur kernel-based operations $\mathbf{A}_K \mathbf{x}$ and $\mathbf{A}_K^H \mathbf{x}$ in (13) can be applied to the three TV restoration algorithms that we consider (PA, NESTA, and KADMM) to perform efficient nonstationary blur restoration.

4. Experimental results and discussions

We investigated the modeling and restoration of nonstationary blur using the proposed method through two controlled experiments and an experiment using real-world lenses. For the controlled

experiments, we introduced pixelwise nonstationary blur to images using matrix \mathbf{A} , which was constructed following the analytic blur models. Subsequently, the blurred images were restored using the proposed eigen blur kernel-based modeling method using matrix \mathbf{A}_K . The controlled experiments allow us to evaluate the modeling accuracy quantitatively because the analytic expressions for the pixelwise nonstationary blur are available. Moreover, it allows us to analyze how modeling accuracy affects the restoration performance because both the pixelwise blur models and the original images are available for quantitative evaluation. For the experiments using real-world lenses, we acquired images degraded optically by lenses using a camera. Subsequently, we restored the acquired blurry images using the eigen blur kernel-based modeling, which uses matrix \mathbf{A}_K . The experiments with real-world lenses allow us to validate the application of the proposed method to calibrate nonstationary lens blur for restoration purposes.

4.1. Controlled experiment: Gaussian blur

4.1.1. Modeling

For the first controlled experiment, we consider pixelwise variant rotated Gaussian blur. A two-dimensional Gaussian distribution and its variants were used to model nonstationary lens blur [4,9,10]. The Gaussian blur kernels were prepared with the standard deviations in the principal axis set as follows:

$$\begin{aligned}\sigma_i(r; d) &= \alpha_i r^{\gamma(d)} + \beta_i, \\ \sigma_j(r; d) &= \alpha_j r^{\gamma(d)} + \beta_j,\end{aligned}\tag{16}$$

where r is the radius from the center of the image normalized by half of the diagonal sensor length, and $\gamma(d)$ determines the severity of nonstationarity depending on the defocus d . Here, α_i , α_j , β_i , and β_j are constants. The principal axes of the Gaussian kernels are rotated to align with the radial and tangential directions. The blur kernel $h(i, j, \xi, \zeta; d)$ is found for every pixel index $(i, j) \in [1, M] \times [1, N]$ and for the defocusing distance d .

We simulate a practical situation, in which blur kernels are available through measurements or estimations only at grid points, and a nonstationary blur model is constructed from the available blur kernels. The pixelwise variant Gaussian blur kernels are sampled at $S_i \times S_j$ grid points for every defocusing distance d . Each sampled blur kernel represents the blur inside a $bs \times bs$ region of an image. Figure 2 shows the blur kernels sampled at 8×8 grid points. The blur kernels simulate the lens blur at three different defocusing distances with $\gamma(d) = 0.5, 1.0, \text{ and } 2.0$. It can be seen that blur kernels change their shapes depending on the defocus and the pixel locations. The blur kernels and their contours at the two locations, indicated with the red squares, are shown for a close inspection.

Eigen blur kernels are computed from the $3 \times 8 \times 8$ sampled Gaussian blur kernels shown in Fig. 2. The PCA is applied, and the eigen blur kernels that correspond to the K largest eigenvalues are selected. The results are evaluated using the mean square error (MSE). Using K eigen blur kernels, the nonstationary blur is modeled using (12). For the interpolated weights in \mathbf{C}_k , bicubic interpolation is used throughout the experiments.

The pixelwise spatially variant blur matrix \mathbf{A} is prepared from the analytic expression for the blur kernels $h(i, j, \xi, \zeta; d)$ at a chosen distance d . The blur matrix \mathbf{A} serves as a benchmark by which the modeling accuracy of the EFF (using \mathbf{A}_{eff}) and the eigen blur kernels (using \mathbf{A}_K) are evaluated. Table 1 shows the modeling accuracy of the EFF method, $\|\mathbf{A} - \mathbf{A}_{\text{eff}}\|_F$ and that of the proposed eigen blur kernel-based method using the K eigen blur kernels $\|\mathbf{A} - \mathbf{A}_{16}\|_F$. We consider the cases with image sizes of 512×512 , 1024×1024 , and 2560×2048 and where blur kernels are available for each block of size 16×16 , 32×32 , 64×64 , 128×128 , and 256×256 . We assume that at least 8 samples exist in the shorter dimension. In all the cases, \mathbf{A}_{16} approximated \mathbf{A} more accurately than \mathbf{A}_{eff} .

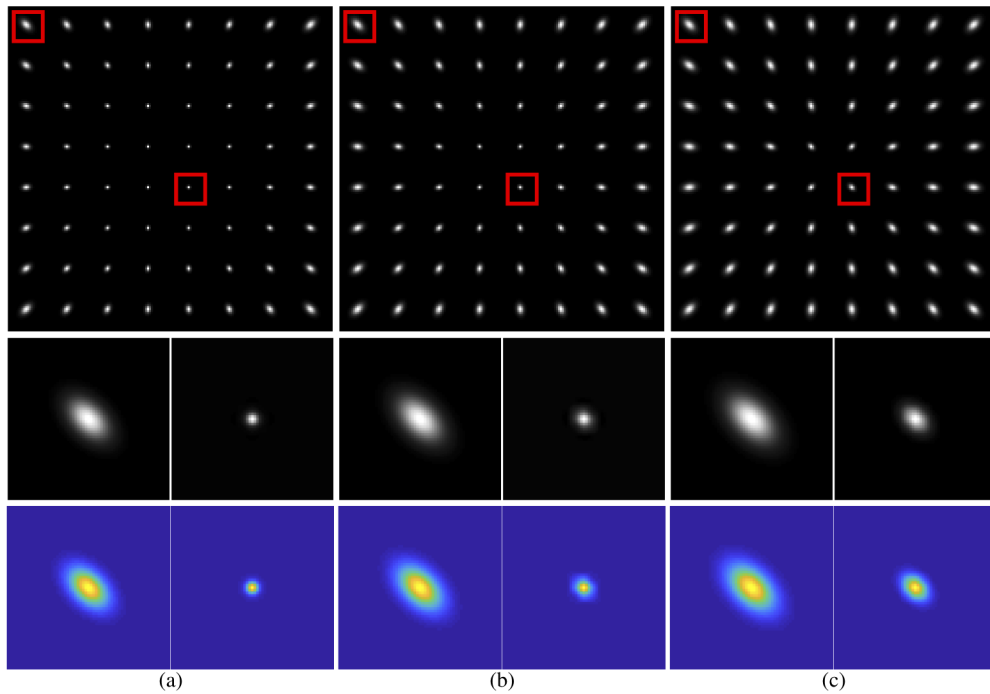


Fig. 2. Parametric Gaussian blur kernels sampled at the 8×8 grid points when the scene is located at different distances using (a) $\gamma = 0.5$; (b) $\gamma = 1.0$; (c) $\gamma = 2.0$. At the bottom are the blur kernels and their contours at the two locations indicated with the red squares.

Figure 3(a) shows MSE from approximating the sampled blur kernels using the K eigen blur kernels. As shown in Fig. 3(b), 16 eigen blur kernels accurately approximate the sampled blur kernels.

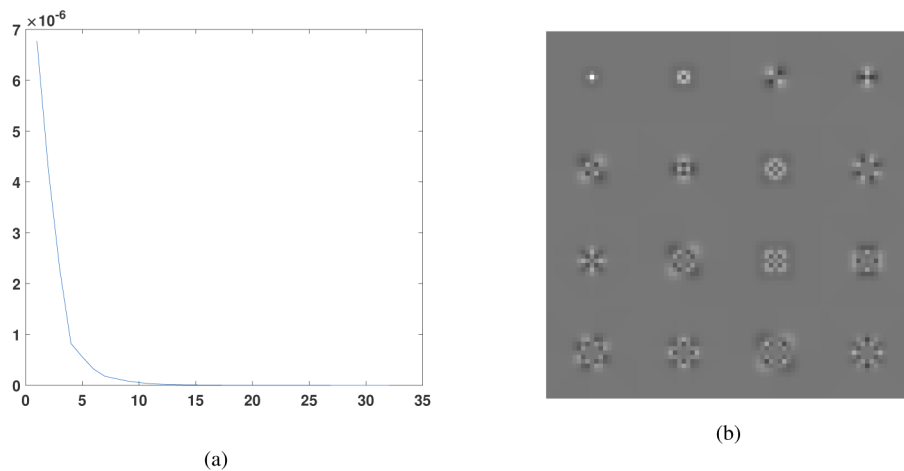


Fig. 3. PCA of the parametric Gaussian blur: (a) Average MSE of the 192 blur kernels reconstructed with K principal components; (b) Eigen blur kernels correspond to 16 largest eigenvalues.

Table 1. Modeling Accuracy: Gaussian Blur

| γ | image size | # of samples | $\ \mathbf{A} - \mathbf{A}_{\text{eff}}\ _F$ | $\ \mathbf{A} - \mathbf{A}_{16}\ _F$ |
|----------|-------------|--------------|--|--------------------------------------|
| 0.5 | 512 × 512 | 8 × 8 | 3.7521E-04 | 1.7155E-04 |
| | | 16 × 16 | 8.2818E-05 | 4.3037E-05 |
| | | 32 × 32 | 1.3104E-04 | 1.2735E-05 |
| | 1024 × 1024 | 8 × 8 | 3.4299E-04 | 1.7069E-04 |
| | | 16 × 16 | 8.0268E-05 | 4.1957E-05 |
| | | 32 × 32 | 4.1251E-05 | 1.1477E-05 |
| | 2560 × 2048 | 10 × 8 | 2.8226E-04 | 1.4061E-04 |
| | | 20 × 16 | 6.4662E-05 | 3.4573E-05 |
| | | 40 × 32 | 1.7182E-05 | 9.5204E-06 |
| 1.0 | 512 × 512 | 8 × 8 | 1.0827E-03 | 2.6955E-04 |
| | | 16 × 16 | 1.7637E-04 | 2.9555E-05 |
| | | 32 × 32 | 1.4017E-04 | 6.7830E-06 |
| | 1024 × 1024 | 8 × 8 | 9.4930E-04 | 2.6799E-04 |
| | | 16 × 16 | 1.7265E-04 | 2.7572E-05 |
| | | 32 × 32 | 1.3362E-04 | 4.6323E-06 |
| | 2560 × 2048 | 10 × 8 | 7.5741E-04 | 2.0054E-04 |
| | | 20 × 16 | 1.3293E-04 | 2.0073E-05 |
| | | 40 × 32 | 3.6864E-05 | 3.5305E-06 |
| 2.0 | 512 × 512 | 8 × 8 | 2.7101E-03 | 1.6620E-04 |
| | | 16 × 16 | 5.0467E-04 | 2.3285E-05 |
| | | 32 × 32 | 1.7473E-04 | 9.3093E-06 |
| | 1024 × 1024 | 8 × 8 | 2.2354E-03 | 1.6286E-04 |
| | | 16 × 16 | 4.9916E-04 | 1.9792E-05 |
| | | 32 × 32 | 5.0139E-04 | 5.7572E-06 |
| | 2560 × 2048 | 10 × 8 | 1.8180E-03 | 1.1772E-04 |
| | | 20 × 16 | 3.8901E-04 | 1.4833E-05 |
| | | 40 × 32 | 1.3028E-04 | 4.3582E-06 |

The Gaussian blur at any pixel location can be approximated using linear combinations of the K eigen blur kernels by (7). Figure 4 shows the interpolated weights $\tilde{c}_k(i, j; d)$ computed for the case in which $\gamma(d) = 1.0$, and the image size is 512×512 for $k = 1, 2, \dots, 16$.

Table 2 compares the average processing time for $\mathbf{A}_{\text{eff}}\mathbf{x}$ and $\mathbf{A}_{16}\mathbf{x}$ operations in terms of clock time. The average of 100 multiplications is shown. The code for the EFF implementation was provided by [12], which we run with our nonstationary lens blur kernels. The eigen blur kernel-based method executes in less time than that required by the EFF in all the cases. Moreover, it executes considerably faster when a large number of blur kernel samples are used. In contrast, the regionwise data accesses complicate the processing flow of the EFF considerably. In particular, the execution of FFT requires the number of data to be the power of two. In some cases, the use of larger number of blur kernel samples increases the number of block accesses without decreasing the size of the blocks. In those cases, the computational complexity increases with the number of blur kernel samples. The computational complexity of the eigen blur kernel-based method does not change based on the number of blur kernel samples, $S_i \times S_j$; instead, it changes based on the number of eigen blur kernels, K .

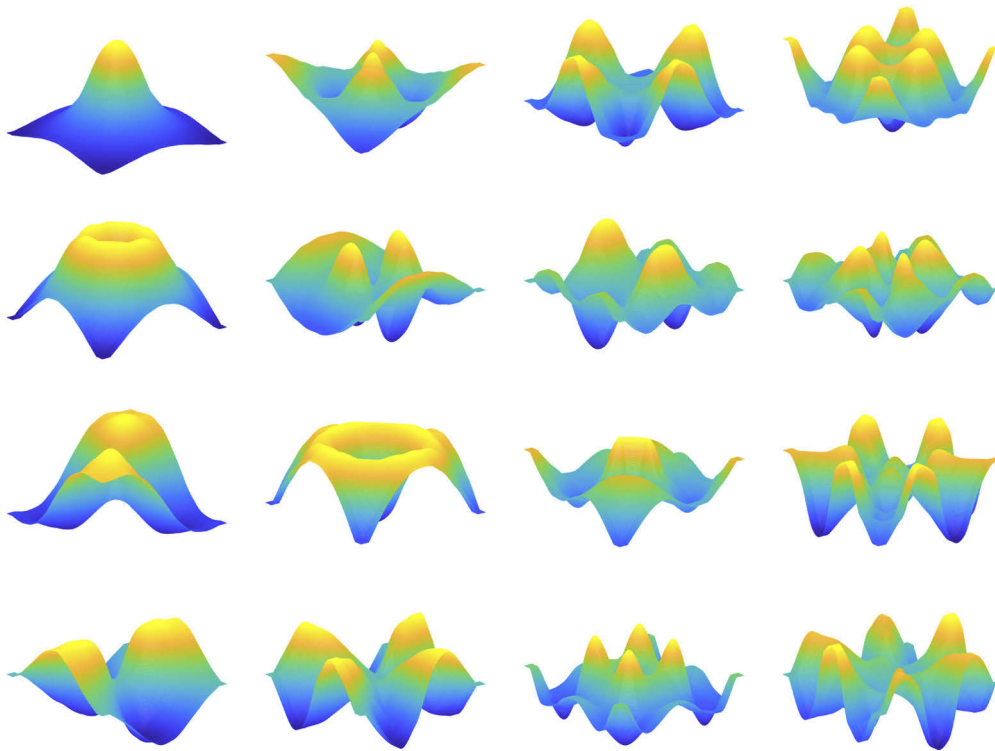


Fig. 4. Interpolated weights $\tilde{c}_k(i, j; d)$ representing nonstationary lens blur at every pixel location as linear combinations of the eigen blur kernels. Shown are $\tilde{c}_k(i, j; d)$ with $\gamma(d) = 1.0$ for $k = 1, 2, \dots, 16$.

Table 2. Computational Complexity in World Clock Time (in Seconds) for Gaussian Blur

| Image size | # of samples | $A_{\text{eff}} \mathbf{x}$ | $A_{16} \mathbf{x}$ |
|-------------|--------------|-----------------------------|---------------------|
| 512 × 512 | 8 × 8 | 0.3572 | 0.2561 |
| | 16 × 16 | 0.4630 | 0.2572 |
| | 32 × 32 | 1.6872 | 0.2459 |
| 1024 × 1024 | 8 × 8 | 1.6265 | 0.9605 |
| | 16 × 16 | 1.3708 | 0.9593 |
| | 32 × 32 | 1.6703 | 0.9586 |
| 2560 × 2048 | 10 × 8 | 8.8798 | 5.2067 |
| | 20 × 16 | 8.2095 | 5.1922 |
| | 40 × 32 | 8.5701 | 5.1269 |

4.1.2. Restoration

We tested TV restoration of nonstationary blur using nonstationary Gaussian blur. The *Lena* (512 × 512), *man* (1,024 × 1024), and *cafe* (2,560 × 2,048) images are used as test images. Test images are shown in Fig. 5(a), (b), and (c). The test images were blurred using matrix \mathbf{A} , which represented the pixelwise variant blur. Noise was added such that the blurred signal-to-noise

ratio (BSNR) given by

$$\text{BSNR} = 10 \log \frac{\text{var}(\mathbf{Ax})}{\sigma_v^2}, \quad (17)$$

where $\text{var}(\mathbf{Ax})$ is the variance of the blurred image, and σ_v^2 is the noise variance, is equal to 60 dB. Subsequently, TV was restored by PA, NESTA and KADMM. All three restoration algorithms terminated when the relative change became smaller than a threshold, which was set to 10^{-5} .

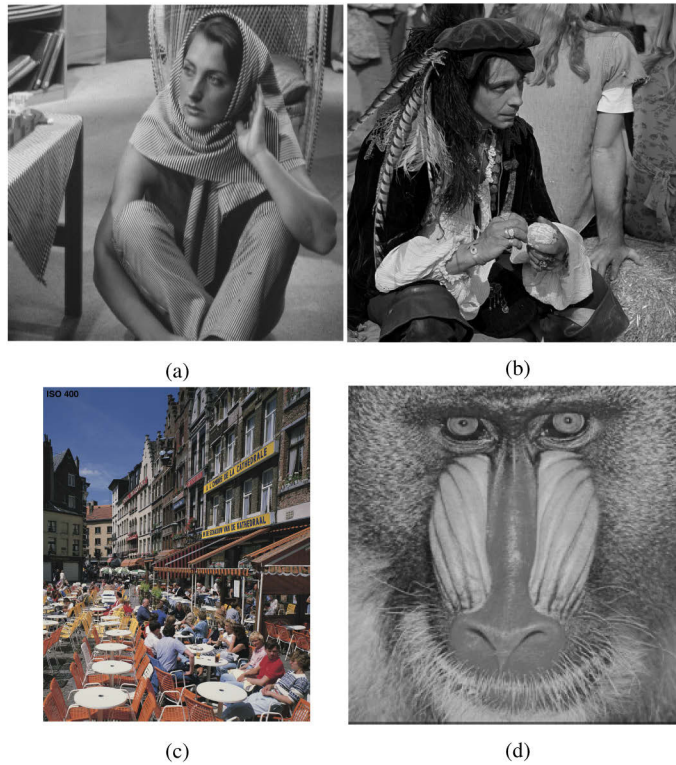


Fig. 5. Test images used in TV restoration: (a) *barbara* (512×512); (b) *man* (1024×1024); (c) *cafe* (2560×2048); (d) *baboon* (256×256).

The operations in the restoration algorithms were implemented using \mathbf{A} , \mathbf{A}_{eff} , and \mathbf{A}_{16} . The restoration using \mathbf{A} represented a case where we had exact knowledge and implementation of nonstationary blur for restoration. The restoration results using \mathbf{A} served as a benchmark for evaluating how modeling accuracy affected the restoration performance. The restoration using \mathbf{A}_{eff} and \mathbf{A}_{16} represented a case where only blur models built from sparse measurements or estimations for restoration were available. Note that we apply pixelwise variant blur using \mathbf{A} to obtain observed images and the models using \mathbf{A}_{eff} and \mathbf{A}_{16} for restoration. Hence, any modeling inaccuracy would degrade the restoration performance.

Table 3 shows the performance of the three restoration algorithms. For all cases of restoration algorithms, blur nonstationarity γ , image sizes $M \times N$, and the number of blur kernel samples $S_i \times S_j$, the MSE values are lower when the restoration algorithms are executed using \mathbf{A}_{16} than when they are executed using \mathbf{A}_{eff} . Moreover, using more blur kernel samples increases the restoration performance when the algorithms are run with \mathbf{A}_{16} ; however, that is not always true when they are run with \mathbf{A}_{eff} . For small images, adding more blur kernels increases overlapping of the EFF window to multiple patches, which may degrade the restoration performance. For example, for 512×512 images, modeling by EFF with 32×32 blur kernels provided lower

Table 3. Restoration Accuracy in MSE: Gaussian Blur

| γ | image size | # of samples | PA | | | NESTA | | | KADMM | | |
|-------------|-------------|--------------|-------------|------------------|-----------------|-------------|------------------|-----------------|-------------|------------------|-----------------|
| | | | A | A _{eff} | A ₁₆ | A | A _{eff} | A ₁₆ | A | A _{eff} | A ₁₆ |
| 0.5 | 512 × 512 | 8 × 8 | 2.2584.E-03 | 2.7465.E-03 | 2.2998.E-03 | 2.7051.E-03 | 2.7450.E-03 | 2.7433.E-03 | 2.1580.E-03 | 2.5628.E-03 | 2.3330.E-03 |
| | | 16 × 16 | 2.2584.E-03 | 2.7267.E-03 | 2.2800.E-03 | 2.7051.E-03 | 2.7318.E-03 | 2.7267.E-03 | 2.1580.E-03 | 2.2374.E-03 | 2.1683.E-03 |
| | | 32 × 32 | 2.2584.E-03 | 7.7974.E-03 | 2.2806.E-03 | 2.7051.E-03 | 8.6482.E-03 | 2.7266.E-03 | 2.1580.E-03 | 5.5754.E-03 | 2.1647.E-03 |
| | 1024 × 1024 | 8 × 8 | 4.4434.E-04 | 5.1575.E-04 | 4.5559.E-04 | 5.1958.E-04 | 5.4617.E-04 | 5.3073.E-04 | 4.4108.E-04 | 5.4366.E-04 | 5.0428.E-04 |
| | | 16 × 16 | 4.4434.E-04 | 4.7880.E-04 | 4.5796.E-04 | 5.1958.E-04 | 5.2651.E-04 | 5.2242.E-04 | 4.4108.E-04 | 4.7490.E-04 | 4.5268.E-04 |
| | | 32 × 32 | 4.4434.E-04 | 4.7179.E-04 | 4.4884.E-04 | 5.1958.E-04 | 5.2555.E-04 | 5.2131.E-04 | 4.4108.E-04 | 4.4998.E-04 | 4.4268.E-04 |
| 2.0 | 2560 × 2048 | 10 × 8 | 1.6024.E-03 | 1.7139.E-03 | 1.6544.E-03 | 1.6206.E-03 | 1.7129.E-03 | 1.6502.E-03 | 1.2288.E-03 | 1.7930.E-03 | 1.4572.E-03 |
| | | 20 × 16 | 1.6024.E-03 | 1.6394.E-03 | 1.6193.E-03 | 1.6206.E-03 | 1.6425.E-03 | 1.6289.E-03 | 1.2288.E-03 | 1.2954.E-03 | 1.2411.E-03 |
| | | 40 × 32 | 1.6024.E-03 | 1.6216.E-03 | 1.6089.E-03 | 1.6206.E-03 | 1.6311.E-03 | 1.6265.E-03 | 1.2288.E-03 | 1.2474.E-03 | 1.2301.E-03 |
| | 512 × 512 | 8 × 8 | 1.5577.E-03 | 1.9461.E-03 | 1.5886.E-03 | 1.8080.E-03 | 1.9466.E-03 | 1.8279.E-03 | 1.6239.E-03 | 1.7925.E-03 | 1.6475.E-03 |
| | | 16 × 16 | 1.5577.E-03 | 1.8427.E-03 | 1.5619.E-03 | 1.8080.E-03 | 1.8482.E-03 | 1.8142.E-03 | 1.6239.E-03 | 1.6312.E-03 | 1.5586.E-03 |
| | | 32 × 32 | 1.5577.E-03 | 6.5222.E-03 | 1.5608.E-03 | 1.8080.E-03 | 7.2428.E-03 | 1.8118.E-03 | 1.6239.E-03 | 4.7692.E-03 | 1.6291.E-03 |
| 1024 × 1024 | 8 × 8 | 2.8436.E-04 | 3.6781.E-04 | 3.1125.E-04 | 3.4580.E-04 | 3.8501.E-04 | 3.5408.E-04 | 2.8042.E-04 | 3.6198.E-04 | 2.9732.E-04 | |
| | 16 × 16 | 2.8436.E-04 | 3.2669.E-04 | 2.9779.E-04 | 3.4580.E-04 | 3.5633.E-04 | 3.4703.E-04 | 2.8042.E-04 | 3.0986.E-04 | 2.8332.E-04 | |
| | 32 × 32 | 2.8436.E-04 | 3.2386.E-04 | 2.9127.E-04 | 3.4580.E-04 | 3.5612.E-04 | 3.4612.E-04 | 2.8042.E-04 | 2.9741.E-04 | 2.8142.E-04 | |
| 2.0 | 2560 × 2048 | 10 × 8 | 8.4615.E-04 | 1.0034.E-03 | 8.9270.E-04 | 8.7054.E-04 | 1.0021.E-03 | 8.8936.E-04 | 5.8737.E-04 | 8.0925.E-04 | 6.3457.E-04 |
| | | 20 × 16 | 8.4615.E-04 | 8.9276.E-04 | 8.6368.E-04 | 8.7054.E-04 | 9.0733.E-04 | 8.7315.E-04 | 5.8737.E-04 | 6.5348.E-04 | 5.9205.E-04 |
| | | 40 × 32 | 8.4615.E-04 | 8.6888.E-04 | 8.5361.E-04 | 8.7054.E-04 | 8.8713.E-04 | 8.7169.E-04 | 5.8737.E-04 | 6.1749.E-04 | 5.8997.E-04 |
| | 512 × 512 | 8 × 8 | 1.1355.E-03 | 1.4235.E-03 | 1.1674.E-03 | 1.3072.E-03 | 1.4148.E-03 | 1.3249.E-03 | 1.0093.E-03 | 1.3049.E-03 | 1.0537.E-03 |
| | | 16 × 16 | 1.1355.E-03 | 1.3264.E-03 | 1.1407.E-03 | 1.3072.E-03 | 1.3312.E-03 | 1.3096.E-03 | 1.0093.E-03 | 1.1073.E-03 | 1.0218.E-03 |
| | | 32 × 32 | 1.1355.E-03 | 5.2996.E-03 | 1.1336.E-03 | 1.3072.E-03 | 5.9138.E-03 | 1.3063.E-03 | 1.0093.E-03 | 3.6441.E-03 | 1.0192.E-03 |
| 1024 × 1024 | 8 × 8 | 1.4487.E-04 | 2.2747.E-04 | 1.6250.E-04 | 1.7783.E-04 | 2.2550.E-04 | 1.8688.E-04 | 1.3973.E-04 | 2.2100.E-04 | 1.5493.E-04 | |
| | 16 × 16 | 1.4487.E-04 | 1.8473.E-04 | 1.5424.E-04 | 1.7783.E-04 | 1.9336.E-04 | 1.7965.E-04 | 1.3973.E-04 | 1.7249.E-04 | 1.4289.E-04 | |
| | 32 × 32 | 1.4487.E-04 | 1.8211.E-04 | 1.5043.E-04 | 1.7783.E-04 | 1.9386.E-04 | 1.7817.E-04 | 1.3973.E-04 | 1.6458.E-04 | 1.3985.E-04 | |
| 2560 × 2048 | 10 × 8 | 3.1338.E-04 | 4.8324.E-04 | 3.4544.E-04 | 3.3592.E-04 | 4.7910.E-04 | 3.5247.E-04 | 2.0775.E-04 | 4.0729.E-04 | 2.2897.E-04 | |
| | 20 × 16 | 3.1338.E-04 | 3.7577.E-04 | 3.3298.E-04 | 3.3592.E-04 | 3.8357.E-04 | 3.3847.E-04 | 2.0775.E-04 | 2.7116.E-04 | 2.1632.E-04 | |
| | 40 × 32 | 3.1338.E-04 | 3.4205.E-04 | 3.2459.E-04 | 3.3592.E-04 | 3.5636.E-04 | 3.3629.E-04 | 2.0775.E-04 | 2.2503.E-04 | 2.1163.E-04 | |

performance than modeling by EFF with 16×16 blur kernels. These results indicate that the more accurate modeling by the eigen blur kernel-based method improves the restoration performance.

Figure 6 shows the restoration results using the *cafe* image with $\gamma = 2.0$ using 8×8 blur kernel samples. The figure shows portions of the image from the lower right side. The improvement in MSEs in this case corresponds to 1.45 dB, 1.33 dB, and 2.50 dB in the peak signal-to-noise ratio (PSNR) for PA, NESTA, and KADMM, respectively. Errors between the original images and the restored images using KADMM are also shown. According to the results, the eigen blur kernel based method restored the edges and details in the original images more closely.

Figure 7 shows the evolution of cost functions vs. processing time for the tested restoration algorithms. Note that PA and NESTA run considerably faster when using \mathbf{A}_{16} than when using \mathbf{A}_{eff} . PA requires one \mathbf{Ax} operation per iteration, and NESTA requires one \mathbf{Ax} and one $\mathbf{A}^H \mathbf{x}$ operation per iteration. The use of the more efficient implementations of $\mathbf{A}_{16} \mathbf{x}$ and $\mathbf{A}_{16}^H \mathbf{x}$ operations results in faster restoration. KADMM reduces the solution space to the n th order Krylov subspace; therefore, it requires only n multiplications by \mathbf{A}_{16} or \mathbf{A}_{eff} , where n is typically a small number. The processing times for the two implementations of KADMM are similar. In all cases, the restoration accuracy is higher when the restoration algorithms are run with \mathbf{A}_{16} than when they are run with \mathbf{A}_{eff} .

4.2. Controlled experiment: thin lens

4.2.1. Modeling

For the second controlled experiment, we consider a blur by a thin lens. We prepared pixelwise variant blur kernels for 256×256 -pixel images. A blur kernel at every pixel location is obtained through the numerical integration given in (2). The parameters from the numerical example in [1] are used, and a matrix \mathbf{A} representing the pixelwise variant blur of a thin lens was prepared.

We simulated a practical situation, in which the blur kernels are available only at grid points. The blur kernels of a thin lens were sampled at 8×8 grid points. Each sampled blur kernel represents the blur inside a 32×32 -pixel region of an image. Figure 8 shows the sampled blur kernels for three different defocusing distances: $z = 8.375, 8.750, \text{ and } 9.125$. As shown, the blur kernels change their shapes depending on the defocus distance and the pixel locations. Moreover, the blur kernel shapes are more complicated than the simple Gaussian model we considered in Section 4.1. The blur kernels and their contours at the two locations, indicated with the red squares, are shown for a closer inspection.

Eigen blur kernels were computed from the $3 \times 8 \times 8$ sampled blur kernels, and PCA was applied to find the eigen blur kernels.

Figure 9(a) shows the MSE from approximating the sampled blur kernels using the K eigen blur kernels. According to the results, 16 eigen blur kernels accurately approximate the sampled blur kernels. The 16 eigen blur kernels are shown in Fig. 9(b).

Table 4 shows the modeling accuracy of the EFF method, $\|\mathbf{A} - \mathbf{A}_{\text{eff}}\|_F$ and of the proposed eigen blur kernels using the K eigen blur kernels $\|\mathbf{A} - \mathbf{A}_{16}\|_F$. In all cases, \mathbf{A}_{16} approximated \mathbf{A} more accurately than \mathbf{A}_{eff} .

Table 4. Modeling Accuracy: Thin Lens

| z | image size | # of samples | $\ \mathbf{A} - \mathbf{A}_{\text{eff}}\ _F$ | $\ \mathbf{A} - \mathbf{A}_{16}\ _F$ |
|-------|------------------|--------------|--|--------------------------------------|
| 8.375 | 256×256 | 8×8 | 5.6636.E-04 | 6.4375.E-05 |
| 8.750 | 256×256 | 8×8 | 4.2576.E-03 | 1.4856.E-04 |
| 9.125 | 256×256 | 8×8 | 1.1355.E-03 | 2.7705.E-05 |

Table 5 compares the average processing time for $\mathbf{A}_{\text{eff}} \mathbf{x}$ and $\mathbf{A}_{16} \mathbf{x}$ operations in terms of clock time. The average time of 100 multiplications is shown. The blur kernels of a thin lens have

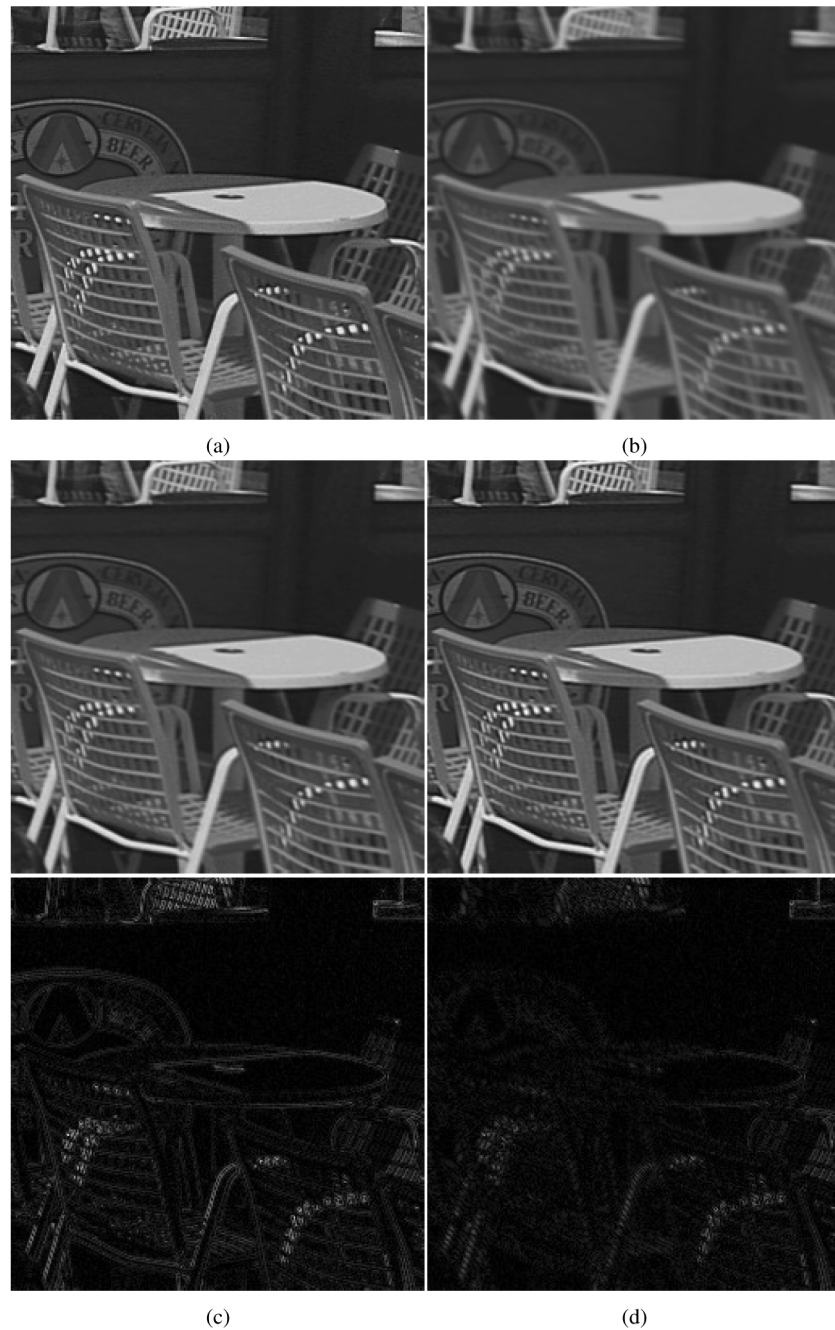


Fig. 6. Examples of TV restoration, with *cafe* image blurred by non-stationary Gaussian blur with $\gamma = 2.0$ using 8×8 blur kernel samples. (a) original image; (b) observed image, followed by restored images and restoration errors when restored by KADMM using (c) EFF with \mathbf{A}_{eff} , and (d) eigen blur kernels with \mathbf{A}_{16} . The figure shows portions of the image from the lower right side.

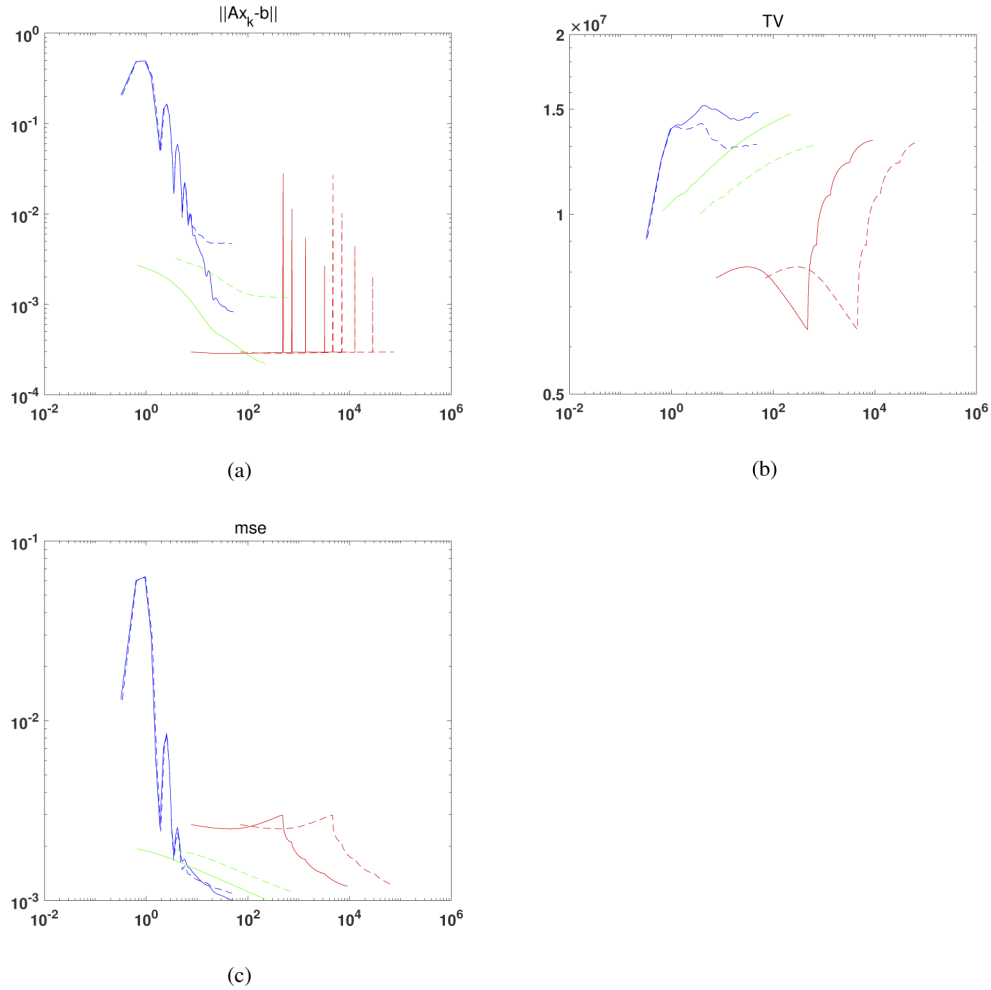


Fig. 7. Processing times when the restoration algorithms are run with A_{eff} and with A_{16} . The evolution of the cost functions vs. time is shown in log-log scale: (a) $\|A\mathbf{x}_k - \mathbf{b}\|$; (b) $\text{TV}(\mathbf{x}_k)$; and (c) MSE between \mathbf{x} and \mathbf{x}_k , where \mathbf{x}_k is the image at the k th iteration. Red: NESTA; Green: PA; Blue: KADMM; Dotted: when run with A_{eff} ; Solid: when run with A_{16} .

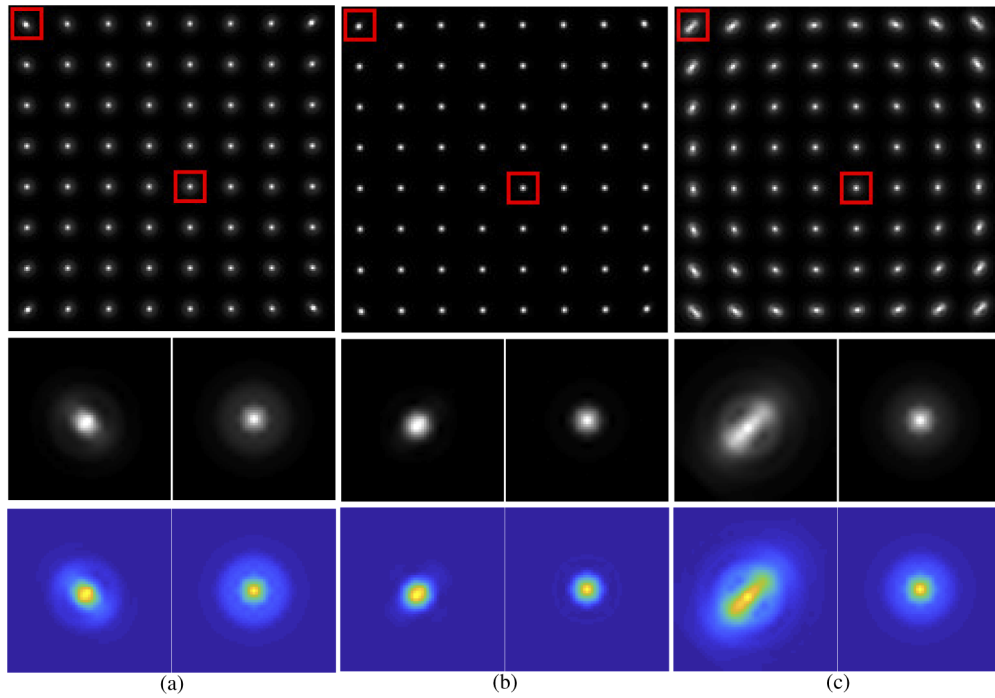


Fig. 8. Blur kernels of a thin lens sampled at the 8×8 grid points when the scene is located at different distances: (a) $z = 8.375$; (b) $z = 8.750$; (c) $z = 9.125$. At the bottom, the blur kernels and their contours are shown at the two locations indicated with the red squares.

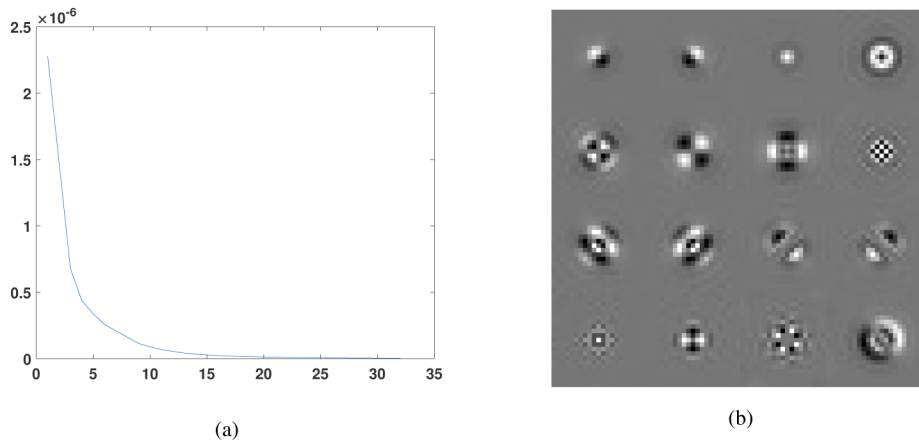


Fig. 9. PCA of blur kernels of a thin lens: (a) average MSE of the 192 blur kernels reconstructed with K principal components; (b) eigen blur kernels corresponding to 16 largest eigenvalues.

larger support for $z = 8.375$ and 9.125 . The EFF implementation requires more padding to accommodate the larger blur kernels. The processing time of the $\mathbf{A}_{\text{eff}}\mathbf{x}$ operation is affected by the larger size of the blur kernels. The $\mathbf{A}_{16}\mathbf{x}$ operation, which is not affected by the size of blur kernels, is considerably faster than $\mathbf{A}_{\text{eff}}\mathbf{x}$.

Table 5. Computational Complexity in World Clock Time in Seconds for Thin Lens Blur

| z | image size | # of samples | $\mathbf{A}_{\text{eff}}\mathbf{x}$ | $\mathbf{A}_{16}\mathbf{x}$ |
|-------|------------------|--------------|-------------------------------------|-----------------------------|
| 8.375 | 256×256 | 8×8 | 0.0878 | 0.0393 |
| 8.750 | 256×256 | 8×8 | 0.0887 | 0.0391 |
| 9.125 | 256×256 | 8×8 | 0.0887 | 0.0397 |

4.2.2. Restoration

We applied TV restoration to restore the blur of a thin lens using the *baboon* image (256×256) shown in Fig. 5(d). The test image is blurred using matrix \mathbf{A} prepared using (2) for each pixel, and noise is added for BSNR at 60 dB. Subsequently, TV was restored by NESTA and KADMM. The restoration algorithms terminated when the relative change was smaller than a threshold, which was set to 10^{-5} .

Table 6 shows the performances of the two restoration methods. For all the tested restoration algorithms and defocusing distances, the MSE values are lower when the restoration algorithms are executed with \mathbf{A}_{16} than \mathbf{A}_{eff} .

Table 6. Restoration Accuracy in MSE: Thin Lens

| z | image size | # of samples | NESTA | | | KADMM | | |
|-------|------------------|--------------|--------------|---------------------------|--------------------|--------------|---------------------------|--------------------|
| | | | \mathbf{A} | \mathbf{A}_{eff} | \mathbf{A}_{16} | \mathbf{A} | \mathbf{A}_{eff} | \mathbf{A}_{16} |
| 8.375 | 256×256 | 8×8 | 1.0084.E-03 | 1.6313.E-03 | 1.1153.E-03 | 7.3973.E-04 | 1.3088.E-03 | 6.3987.E-04 |
| 8.750 | 256×256 | 8×8 | 9.4191.E-04 | 1.9038.E-03 | 9.6763.E-04 | 6.5280.E-04 | 1.7446.E-03 | 6.2062.E-04 |
| 9.125 | 256×256 | 8×8 | 2.0141.E-03 | 3.1115.E-03 | 2.0340.E-03 | 1.7733.E-03 | 2.7542.E-03 | 1.8491.E-03 |

Figure 10 shows restoration results using the *baboon* image with $z = 9.135$ using 8×8 blur kernel samples. The improvement in MSEs in this case corresponds to 1.85 dB and 1.73 dB in the PSNR for NESTA, and KADMM, respectively. Errors between the original images and the restored images using KADMM are also shown. It can be seen that the eigen blur kernel based method restored the edges and details in the original images more closely.

4.3. Commercial lenses

4.3.1. Modeling

We also characterized the blur of a commercially available lens—a Nikon AF-G 50 mm f/1.8 lens. Blur kernels at f-number 1.8 are modeled for a 4016×6016 -pixel sensor. The test image is photographed at three different distances from the camera. When a camera with a lens with focal length f and aperture A (in f-number) is focused at s , any objects between $s - d_{\text{near}}$ and $s + d_{\text{far}}$ provides an acceptable circle of confusion c , where

$$d_{\text{near}} = \frac{sf^2}{f^2 + Ac(s-f)} \quad (18)$$

$$d_{\text{far}} = \frac{sf^2}{f^2 - Ac(s-f)}$$

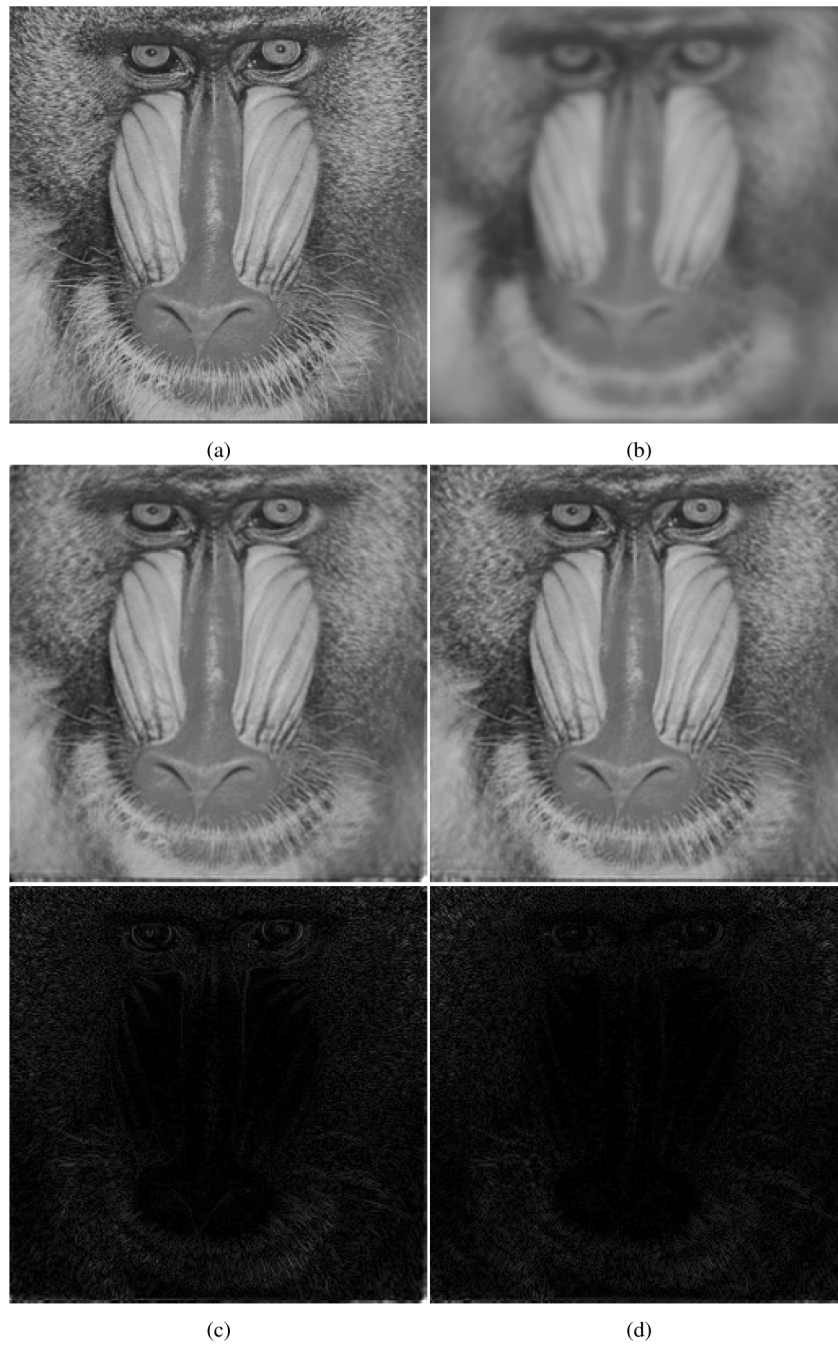


Fig. 10. Examples of TV restoration, with *baboon* image blurred by non-stationary thin lens blur with $z = 9.135$ using 8×8 blur kernel samples. (a) original image; (b) observed image, followed by restored images and restoration errors when restored by KADMM using (c) ERF with \mathbf{A}_{eff} , and (d) eigen blur kernels with \mathbf{A}_{16} .

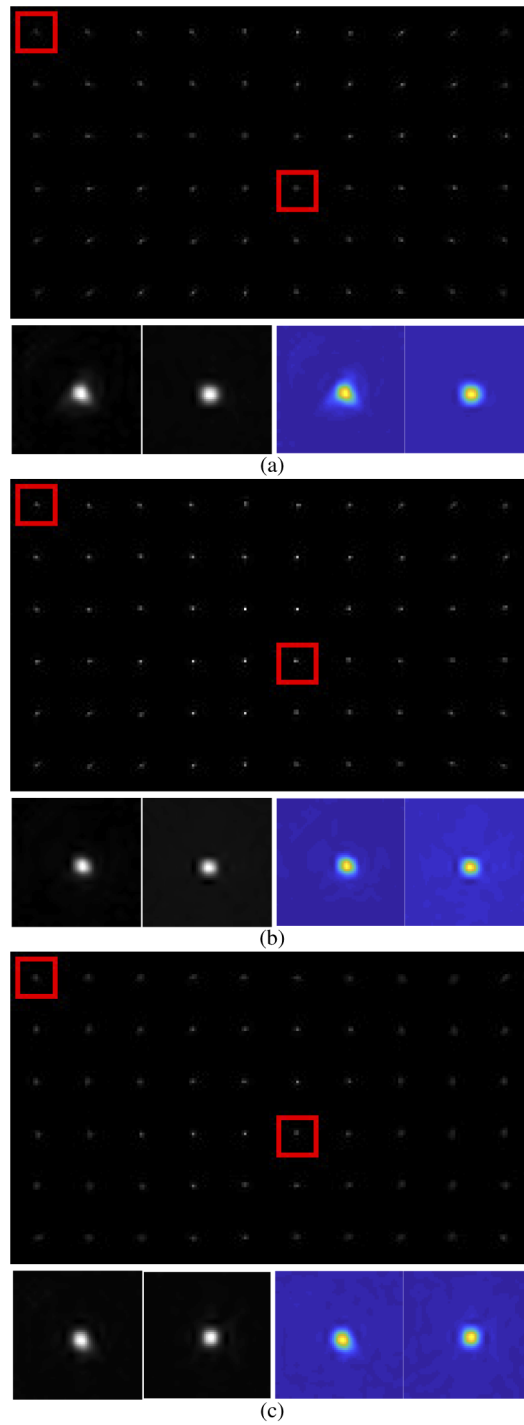


Fig. 11. Estimated blur kernels for a Nikon AF 50 mm f/1.8 lens at f/1.8 when offset from the focusing distance s by (a) $-2d_{\text{near}}$; (b) 0; (c) $+2d_{\text{far}}$. At the bottom are the blur kernels and their contours from the two locations indicated with the red squares.

We took three photographs of the test image at three distances, placing the test image at $s - 2d_{\text{near}}$, s , and $s + 2d_{\text{far}}$. This setup allows us to model nonstationary lens blur not only when it is properly focused and but also when it is slightly defocused.

We estimated the blur kernels at 6×10 locations and 3 defocus distances using (4) (the estimated blur kernels are shown in Fig. 11). The blur kernels at different defocusing distances and different pixel locations show a variety of shapes and spreads and include some shapes quite different from those of the Gaussian model. The AF 50 mm $f/1.8$ lens consists of seven lens elements in six groups designed to provide excellent optical properties. However, the lens still suffers from nonstationary lens blur, even when properly focused, revealing blur kernels with complicated shapes.

Eigen blur kernels of the AF-G 50 mm $f/1.8$ lens are found through PCA applied to the $3 \times 6 \times 10$ blur kernels shown in Fig. 11.

Figure 12(a) shows the MSE between the estimated blur kernels and their approximations by the K eigen blur kernels. The average MSE of the blur kernels for a given K is shown. The figure shows that the blur kernels can be approximated accurately using only a small number of eigen blur kernels. Figure 12(b) shows the 16 eigen blur kernels found from the estimated blur kernels.

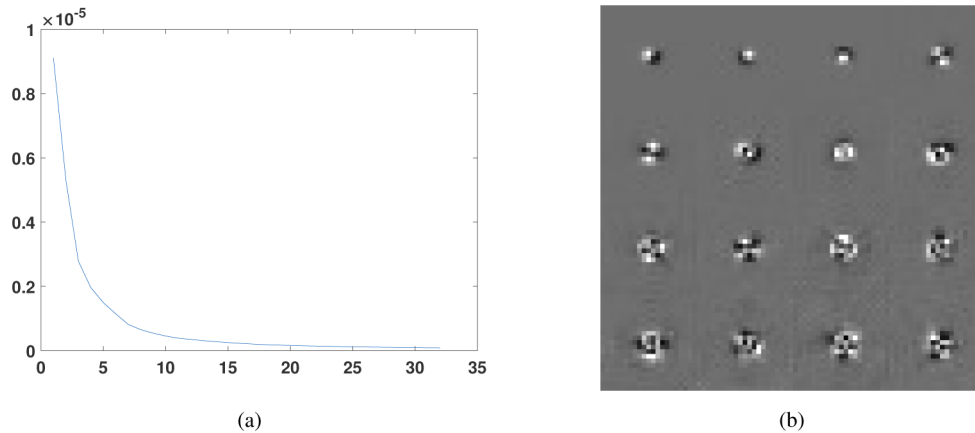


Fig. 12. PCA of blur kernels of Nikon AF-G 50 mm $f/1.8$ lens at $f/1.8$: (a) average MSE of the 180 blur kernels reconstructed with K principal components; (b) eigen blur kernels corresponding to the 16 largest eigenvalues.

4.3.2. Restoration

For commercial lens blur, matrix \mathbf{A} that represents the pixelwise variant lens blur is not available. However, any image photographed using a lens is an observed image b that is degraded optically by pixelwise variant lens blur and observation noise. We use KADMM for the restoration with \mathbf{A}_{eff} and \mathbf{A}_{16} . Because no original image free of lens blur and noise is available, it is difficult to evaluate the restoration performance quantitatively. Therefore, for quantitative evaluation, we use the same photographed test images used to estimate the blur kernels for restoration. We extracted the patches inside of the four red dots of the restored images and compared them with the definition of the circular patterns. We measured the MSE after compensating for the dynamic ranges of the patches.

Table 7 shows the average MSE of the patches. For the photographed test images at all three distances, the average MSE of the patches restored using \mathbf{A}_{16} is higher than that using \mathbf{A}_{eff} .

Figure 13 shows examples of the restored images with the *DCU3* test chart photographed at the three distances. It can be seen blur due to defocus and optical aberrations can be restored using precalibrated nonstationary blur models.

Table 7. Restoration Accuracy in MSE: Nikon AF-G 50 mm f/1.8 lens

| defocus | image size | # of samples | KADMM | |
|---------------------|--------------------|---------------|------------------|--------------------|
| | | | A_{eff} | A_{16} |
| $-2d_{\text{near}}$ | 4016×6016 | 6×10 | 4.8874.E-02 | 5.3521.E-03 |
| 0 | 4016×6016 | 6×10 | 4.8014.E-02 | 8.4809.E-03 |
| $+2d_{\text{far}}$ | 4016×6016 | 6×10 | 4.4842.E-02 | 6.3501.E-03 |

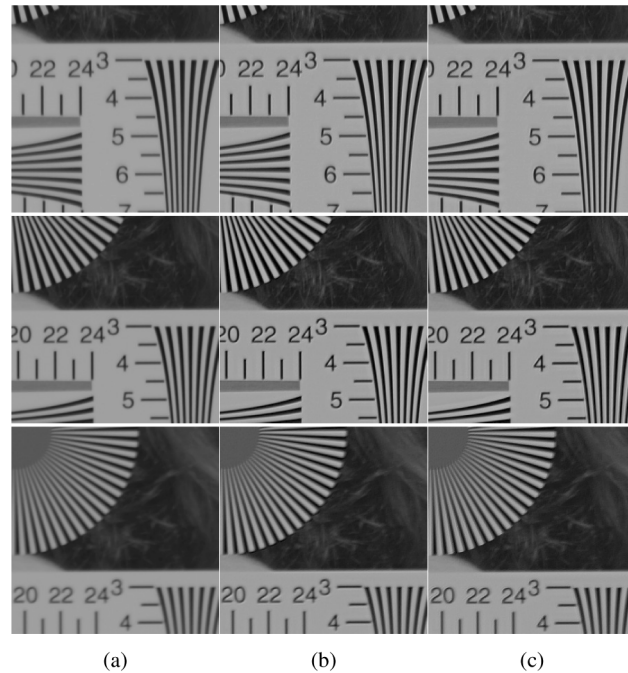


Fig. 13. Restoration of lens blur of Nikon AF 50 mm f/1.8 lens at f/1.8 when defocused by (a) photographed image, and images restored by KADMM using (b) A_{eff} ; and (c) A_{16} . Top : $d = -2d_{\text{near}}$; Middle : $d = 0$; Bottom: $d = +2d_{\text{far}}$. The figure shows portions of the image from the upper left corner.

5. Conclusion

In this study, nonstationary lens blur was modeled by eigen blur kernels, which are the principal components of sampled blur kernels. The proposed eigen blur kernel-based method provides a more accurate and efficient method for modeling nonstationary lens blur compared with a widely used modeling method that assumes regionwise stationary blur. The improved restoration performance achieved by the proposed modeling method was validated with TV restoration. The proposed method can be applied to model various nonstationary degradations of image acquisition processes in practical situations, where degradation information is available only at some sparse pixel locations. We are currently working on modeling nonstationary blur due to camera shake using eigen blur kernels.

Funding

Ulsan National Institute of Science and Technology (1.200033.01); National Research Foundation of Korea (2016R1D1A1B01016041).

Disclosures

The authors declare no conflicts of interest.

References

1. V. N. Mahajan and V. N. Mahajan, *Aberration theory made simple* (SPIE optical engineering press, 1991).
2. Zemax, *Zemax, Zbase, Optical Design Database* (Zemax Development Corp., 2003).
3. J. Brauers, C. Seiler, and T. Aach, "Direct PSF estimation using a random noise target," in *IS&T/SPIE Electronic Imaging*, (International Society for Optics and Photonics, 2010), pp. 75370B–75370B.
4. E. Kee, S. Paris, S. Chen, and J. Wang, "Modeling and removing spatially-varying optical blur," in *Computational Photography (ICCP), 2011 IEEE International Conference on*, (IEEE, 2011), pp. 1–8.
5. C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf, "Non-stationary correction of optical aberrations," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, (IEEE, 2011), pp. 659–666.
6. M. B. Hullin, J. Hanika, and W. Heidrich, "Polynomial optics: A construction kit for efficient ray-tracing of lens systems," in *Computer Graphics Forum*, vol. 31 (Wiley Online Library, 2012), pp. 1375–1383.
7. F. Heide, M. Rouf, M. B. Hullin, B. Labitzke, W. Heidrich, and A. Kolb, "High-quality computational imaging through simple lenses," *ACM Trans. Graph.* **32**(5), 1–14 (2013).
8. H. Tang and K. N. Kutulakos, "What does an aberrated photo tell us about the lens and the scene?" in *Computational Photography (ICCP), 2013 IEEE International Conference on*, (IEEE, 2013), pp. 1–10.
9. J. Simpkins and R. L. Stevenson, "Parameterized modeling of spatially varying optical blur," *J. Electron. Imaging* **23**(1), 013005 (2014).
10. J. Jang, J. D. Yun, and S. Yang, "Modeling non-stationary asymmetric lens blur by normal sinh-arcsinh model," *IEEE Trans. on Image Process.* **25**(5), 2184–2195 (2016).
11. R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," in *ACM Transactions on Graphics (TOG)*, vol. 25 (ACM, 2006), pp. 787–794.
12. O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *Int. J. Comput. Vis.* **98**(2), 168–186 (2012).
13. P. J. Davis, *Circulant matrices* (American Mathematical Soc., 2012).
14. P. C. Hansen, J. G. Nagy, and D. P. O'leary, *Deblurring images: matrices, spectra, and filtering* (SIAM, 2006).
15. B. R. Hunt, "The application of constrained least squares estimation to image restoration by digital computer," *IEEE Trans. Comput.* **C-22**(9), 805–812 (1973).
16. L. B. Lucy, "An iterative technique for the rectification of observed distributions," *The Astronomical J.* **79**, 745 (1974).
17. M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. on Image Process.* **19**(9), 2345–2356 (2010).
18. M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo, "An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Tran. on Image Process.* **20**(3), 681–695 (2011).
19. S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, "An augmented lagrangian method for total variation video restoration," *IEEE Trans. on Image Process.* **20**(11), 3097–3111 (2011).
20. J. G. Nagy and D. P. O'Leary, "Restoring images degraded by spatially variant blur," *SIAM J. Sci. Comput.* **19**(4), 1063–1082 (1998).
21. M. Hirsch, S. Sra, B. Schölkopf, and S. Harmeling, "Efficient filter flow for space-variant multiframe blind deconvolution," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, (IEEE, 2010), pp. 607–614.
22. H.-S. Lee, E. Heo, W. Lee, D.-C. Ahn, J. Cheon, K. Kim, K. Lee, J. Lee, Y. Choi, and S. Chang, "A real time implementation of a bayer domain image deblurring core for optical blur compensation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (IEEE, 2020), pp. 1673–1677.
23. X. Wu, H. Yang, B. Liu, and X. Liu, "Non-uniform deblurring for simple lenses imaging system," in *2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*, (IEEE, 2020), pp. 274–278.
24. M. Arigovindan, J. Shaevitz, J. McGowan, J. W. Sedat, and D. A. Agard, "A parallel product-convolution approach for representing depth varying point spread functions in 3d widefield microscopy based on principal component analysis," *Opt. Express* **18**(7), 6461–6476 (2010).
25. N. Patwary and C. Preza, "Computationally tractable approach to pca-based depth-variant psf representation for 3d microscopy image restoration," in *Computational Optical Sensing and Imaging*, (Optical Society of America, 2014), pp. CW1C–5.
26. M. Jarvis and B. Jain, "Principal component analysis of psf variation in weak lensing surveys," arXiv preprint astro-ph/0412234 (2004).
27. E. Bertin, "Automated morphometry with sextractor and psfex," in *Astronomical Data Analysis Software and Systems XX*, vol. 442 (2011), p. 435.
28. J. Bobin, S. Becker, and E. Candes, "A fast and accurate first-order method for sparse recovery," *SIAM J. Imaging Sci.* **4**(1), 1–39 (2011).

29. L. Condat, "A generic proximal algorithm for convex optimization—application to total variation minimization," *IEEE Signal Proc. Letters* **21**(8), 985–989 (2014).
30. J. D. Yun and S. Yang, "Admm in krylov subspace and its application to total variation restoration of spatially variant blur," *SIAM J. Imaging Sci.* **10**(2), 484–507 (2017).
31. R. C. Gonzalez and R. E. Woods, "Digital image processing second edition," (Publishing House of Electronics Industry, Beijing, 455, 2002).
32. A. Mosleh, P. Green, E. Onzon, I. Begin, and J. Pierre Langlois, "Camera intrinsic blur kernel estimation: A reliable framework," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, (2015), pp. 4961–4968.
33. N. Joshi, R. Szeliski, and D. J. Kriegman, "PSF estimation using sharp edge prediction," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, (IEEE, 2008), pp. 1–8.
34. H. C. Andrews and B. R. Hunt, "Digital image restoration," (Prentice-Hall Signal Processing Series, Prentice-Hall, Englewood Cliffs, 1977), 1.
35. M. R. Banham and A. K. Katsaggelos, "Digital image restoration," *IEEE Signal Process. Mag.* **14**(2), 24–41 (1997).