

# Efficient Method for High-Quality Removal of Nonuniform Blur in the Wavelet Domain

Tao Yue, Jinli Suo, Xun Cao, *Member, IEEE*, and Qionghai Dai, *Senior Member, IEEE*

**Abstract**—This paper presents a novel nonuniform deblurring approach, which defines the blur model and calculates regularized nonuniform deconvolution in the wavelet domain to achieve high efficiency and high accuracy simultaneously. Targeting high computation efficiency, we derive a wavelet-domain hierarchical blur model, which can be calculated efficiently by exploiting the sparsity property of natural images in the wavelet domain. Correspondingly, the blur model is incorporated into a multilayer framework and at each layer spatially varying step sizes are introduced to further accelerate the convergence of the algorithm. In addition to the efficiency advantages, the proposed approach deals with intensely nonuniform blur with high accuracy due to the intrinsic tight supportness of wavelet basis. We conduct a series of experiments and comparisons to validate the efficiency and effectiveness of our algorithm.

**Index Terms**—Deblur, nonblind, nonuniform, wavelet.

## I. INTRODUCTION

IMAGE blur is a common degeneration and has been a hot topic for decades. Restoring the latent sharp image from its blurred observation is often formulated as a deconvolution with a given/estimated point spread function (PSF) (also known as blur kernel). Spatially uniform deconvolution algorithms have been studied thoroughly, either for 2D images [1]–[14] or for 3D volume data [15]–[17]. With the spatial resolution of digital images ever growing, many researchers attempt to develop efficiently deconvolution algorithms [7], [8], [17]–[21]. However, most of these algorithms focus on uniform blurred images and are based on the fast Fourier transform (FFT), while most commonly observed degradations (e.g., motion blur, defocus, optics aberrations, and atmospheric turbulence) are spatially varying and cannot be processed by the Fourier transform.

To handle the widely existing nonuniform blur in real cases, various nonuniform deblurring algorithms [22]–[30] are proposed recently, but most of them (except for [22], [26], and [30]) focus on blur kernel estimation rather than nonblind deblurring module. Besides,

Zhang and Hirakawa [31] proposed a novel blindly nonuniform restoration algorithm in the double discrete wavelet transform, which is, however, restricted to the parametric blur model (e.g., motion blur with a constant direction and speed). In spatially varying nonblind deblurring, the spatial domain blur model (nonuniform convolution) has to be calculated repeatedly to fit the observation and update the restoration result according to the fitting residuals. In general, the existing methods resort to either pixelwise blur model [22]–[26] or patch-based blur model [27]–[30].

The pixelwise model computes the local blur kernels pixel by pixel and can handle arbitrary nonuniform blur, but suffers from extremely high computational cost and demanding storage (the local blur kernels are usually stored in memory for computation efficiency). A digital image usually consists tens of megapixels and the 3D volume data are even orders of magnitude larger, so the speed of the nonblind deblurring algorithm becomes more and more important. However, the nonuniform blur model cannot be accelerated by FFT and, thus, is quite time-consuming. Differently, the patch-based model is fast and facilitates the nonuniform extension of uniform deconvolution algorithms by assuming the blur kernels vary smoothly. For example, Löfdahl [30] uses the patch-based FFT to accomplish fast nonuniform deblur of multiple images and Hirsch *et al.* [27] proposed the efficient filter flow (EFF)-based algorithm, which approximates the nonuniform deblurring using multiple local filters (implemented by FFT). In spite of their high efficiency, these patch-based approaches are of limited precision in modeling abruptly changed degradation processes, even with densely overlapping patches. Overall, the existing nonuniform blur models are either extremely slow or cannot handle intensely varying blur, and there still lack sufficient investigations for efficient and high-quality spatially varying nonblind deblurring so far.

To achieve efficient and high-quality nonuniform deblurring, we propose to derive a blur model and conduct deblurring in a transform domain satisfying two properties: 1) in which natural images are of sparse representation, and thus, the calculation is of high efficiency and 2) having excellent time/space-frequency performance (i.e., the tight support region) to ensure high computation accuracy even for abruptly varying blurred cases.

## A. Wavelet Domain Blur Model

Essentially, the patch-based methods can be regarded as a kind of 2D short time/space Fourier analysis (STFA), and the accuracy for dealing with nonuniform blur depends on

Manuscript received December 22, 2014; revised July 1, 2015 and December 31, 2015; accepted April 4, 2016. Date of publication May 10, 2016; date of current version September 5, 2017. This work was supported by the National Natural Science Foundation of China under Project 61327902, Project 61371166, and Project 61422107. This paper was recommended by Associate Editor T. D. Tran.

T. Yue, J. Suo, and Q. Dai are with the Beijing Key Laboratory of Multi-dimension and Multi-scale Computational Photography, Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: yuetao.thu@gmail.com; jlsuo@tsinghua.edu.cn; qionghaidai@tsinghua.edu.cn).

X. Cao is with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210000, China (e-mail: caoxun@nju.edu.cn).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSVT.2016.2565938

the time/space-frequency resolution (patch size). Targeting for calculating the nonuniform blur model more accurately with high computation speed, we try to formulate the blur model in a transform domain having better time/space-frequency resolution than STFA, and the wavelet analysis becomes our top choice for handling nonuniform blurring due to its excellent time/space-frequency resolution, i.e., tight support region. Furthermore, in wavelet domain, most energies concentrate on a small portion of large coefficients, while most of the zero/close-to-zero entries can be neglected without affecting the visual quality. Making use of this sparsity of the wavelet coefficients of natural images, the nonuniform blur operation, i.e., the most time-consuming module in the existing nonuniform deblurring algorithms, can be significantly accelerated. This acceleration consists with the intuition that the image patches are not equally important in deblurring, e.g., both the blurring and deblurring operations are meaningless for the textureless regions, where the corresponding high-frequency wavelet coefficients happen to be small and can be ignored. Considering efficiency and accuracy jointly, we derive a nonuniform blur model based on decimated (nonredundant) wavelet representation, which can accelerate the computation of nonuniform blur model significantly and is of high accuracy.

Note that Escande *et al.* [32], [33] proposed a fast nonblind deblurring algorithm utilizing the sparsity of nonuniform blur kernels in the wavelet domain. Since they approximate the blur kernels by attenuating their small wavelet coefficients, their approach is applicable for some specific cases with kernels varying slowly and being smooth as well. These assumptions largely limit the algorithm's applications to image degradation caused by camera shake, complex object motion, and coded aperture. Differently, our approach utilizes the sparsity in wavelet representation of natural images instead of blur kernels, and thus can handle nonsmooth kernels (as we show in experiments). Therefore, our approach largely differentiates from Escande *et al.*'s [32], [33] method in both the basic idea and technical details. However, the proposed method is not incompatible with Escande *et al.*'s [32], [33] algorithm. For the cases satisfying their assumptions, two strategies can potentially be adopted jointly to achieve higher efficiency.

### B. Wavelet-Based Efficient Nonuniform Multilayer Optimization

Deblurring is an ill-posed problem, so priors are required to regularize the results and make the problem resolvable. Considering that we use a wavelet domain nonuniform blur model, a prior in the wavelet domain rather than in the spatial domain is a natural choice for us.

To solve the deblurring problems with wavelet domain priors, Daubechies *et al.* [34] proposed thresholding Landerweber (TL) algorithm that demonstrates high performance, and proved its convergence mathematically. Similarly, many deblurring algorithms use the wavelet domain sparsity priors by alternating between image deconvolution operation in spatial domain and coefficient thresholding in wavelet domain, such as in [20] and [35]–[37]. Although regularization in wavelet domain helps deblurring, but most existing methods

still perform deconvolution in spatial domain, and thus are quite time-consuming for nonuniform deblurring. Differently, our blur model is directly computed in the wavelet domain, and the sparsity of wavelet coefficients is intensified by applying thresholding operation, so our deconvolution works much faster.

As the data scale increases, the slow convergence of a TL algorithm limits its applications and researchers begin to explore speeding up strategies. Vonesch and Unser [7], [17] proposed a wavelet domain multilayer thresholding algorithm. By extending Vonesch and Unser's method [7], [17], we propose the nonuniform step sizes to form a multilayer solver for hierarchically optimizing the objective function in the wavelet domain. This method can significantly accelerate the convergence and reduce the iteration number. In addition, it is worth noting that although the proposed algorithm focuses on the 2D images degraded by camera motion caused nonuniform blur, it can be applied to any nonuniform blurring cases on multidimensional data.

In summary, this paper addresses the high computational cost of high-quality nonuniform deblurring by two strategies: 1) formulate a wavelet domain blurring model, which can be used for efficient nonuniform restoration and 2) propose a wavelet domain multilayer framework as well as the nonuniform step size maps for solving the nonuniform deblurring problem with wavelet domain sparsity prior. Compared with most existing pixelwise nonuniform deblurring approaches, our method is significantly faster, while keeps comparable performance (and its performance is much better than the patchwise methods).

The rest of this paper is organized as follows. Section II presents the proposed nonuniform blur model in the wavelet domain. Section III describes the wavelet regularized multilayer deblurring algorithm. The experiment results are demonstrated in Section IV. Finally, the conclusion is drawn in Section V.

## II. APPROXIMATE NONUNIFORM BLURRING MODEL IN WAVELET DOMAIN

The existing wavelet-based deblurring methods usually use the sparsity of wavelet coefficients as a regularization, while this section formulates our wavelet domain nonuniform blur model, which provides the basis for efficient nonuniform convolution and deconvolution.

### A. Redundant Wavelet Transform Uniform Blur Kernels

Denoting  $\mathbf{L}$  as the latent sharp image, we can formulate its redundant wavelet transform as  $\mathbf{L} = \sum_{j,m} \hat{\mathbf{W}}_{j,m} \hat{\mathbf{C}}_{j,m}$ . Here,  $\hat{\mathbf{W}}_{j,m}$  is a Toeplitz matrix denoting the redundant wavelet reconstruction matrix of the  $(j, m)$ th subband, and  $\hat{\mathbf{C}}_{j,m}$  denotes the corresponding redundant wavelet coefficient vector, with  $j$  and  $m$  indexing the scale and subband, respectively. Mathematically,  $\mathbf{L}$  blurry version can be calculated by multiplying it with a blurring matrix  $\mathbf{H}$ . If the blur kernels are spatially uniform, the blurring matrix  $\mathbf{H}$  is also a Toeplitz matrix. Therefore, in this case, the blurring matrix

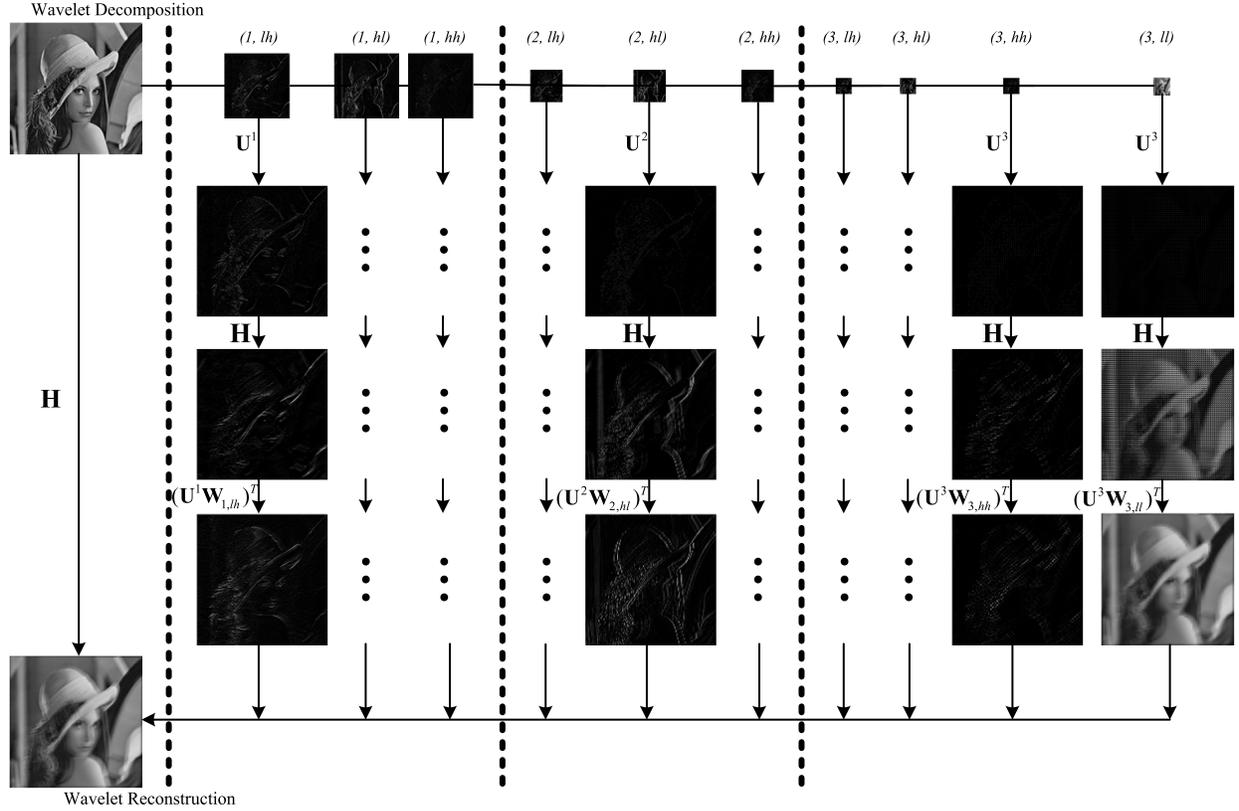


Fig. 1. Diagram of our wavelet domain nonuniform blurring model. The image is degraded by a motion blur randomly generated with six DOFs. For visualization, the intensity levels of the several subbands are modified.

can be applied to wavelet coefficients by simply exchanging the order of the matrix multiplication

$$\mathbf{HL} = \mathbf{H} \sum_{j,m} \hat{\mathbf{W}}_{j,m} \hat{\mathbf{C}}_{j,m} = \sum_{j,m} \hat{\mathbf{W}}_{j,m} \mathbf{H} \hat{\mathbf{C}}_{j,m} \quad (1)$$

and we can utilize the sparsity of wavelet coefficients to develop efficient deblurring approaches by just performing the blurring process on the nonzero coefficients.

However, directly applying this wavelet domain blur model for nonuniform deblurring suffers from two problems: 1) the redundant wavelet decomposition does not reduce (increase instead) the number of coefficients, but simply replacing the redundant wavelet  $\hat{\mathbf{W}}_{j,m}$  with its nonredundant counterpart  $\mathbf{W}_{j,m}$  encounters the nonsquare problem of reconstruction matrix and 2) Toeplitz assumption of blur matrix  $\mathbf{H}$  does not hold for nonuniform blur kernels. The above two problems make it not trivial to exchange the order of  $\mathbf{H}$  and  $\mathbf{W}$  in (1).

### B. Extension to Nonredundant Wavelet Transform by Preupsampling

The reconstruction using nonredundant wavelet coefficients can be expressed by

$$\mathbf{L} = \sum_{j,m} \mathbf{W}_{j,m}^T \mathbf{C}_{j,m} \quad (2)$$

where the nonredundant wavelet decomposition matrix  $\mathbf{W}_{j,m} \in \mathbb{R}_{N \times (N/2^j)}$  is a short matrix and its transpose  $\mathbf{W}_{j,m}^T$

(i.e., the reconstruction matrix) is a tall matrix. To address this nonsquare problem, we apply upsampling matrix  $\mathbf{U}$ , which doubles the length of a column vector by inserting zeros between elements, to the coefficients  $\mathbf{C}_{j,m}$  and the corresponding wavelet reconstruction matrix  $\mathbf{W}_{j,m}^T$ , and then, (2) becomes

$$\mathbf{L} = \sum_{j,m} (\mathbf{U}^j \mathbf{W}_{j,m})^T (\mathbf{U}^j \mathbf{C}_{j,m}) \quad (3)$$

where  $\mathbf{U}^j$  means upsampling  $j$  times and the length of coefficient vector  $\mathbf{C}_{j,m}$  is increased by  $2^j$  after premultiplying  $\mathbf{U}^j$ . Similarly, the nonredundant wavelet reconstruction matrix  $\mathbf{W}_{j,m}^T$  turns into a square Toeplitz matrix  $(\mathbf{U}^j \mathbf{W}_{j,m})^T$  after upsampling.

Suppose  $\mathbf{H}$  is also a Toeplitz matrix, the blurring operation can be performed in the wavelet domain by exchanging two Toeplitz matrices

$$\begin{aligned} \mathbf{HL} &= \mathbf{H} \sum_{j,m} (\mathbf{U}^j \mathbf{W}_{j,m})^T (\mathbf{U}^j \mathbf{C}_{j,m}) \\ &= \sum_{j,m} (\mathbf{U}^j \mathbf{W}_{j,m})^T \mathbf{H} (\mathbf{U}^j \mathbf{C}_{j,m}). \end{aligned} \quad (4)$$

For convenience, in the following, we use  $\mathbf{W}$  and  $\mathbf{C}$  to denote  $\mathbf{UW}$  and  $\mathbf{UC}$  for short, respectively.

Fig. 1 shows the diagram of preupsampling scheme for our wavelet domain blur model, including mainly three steps: 1) upsample the nonredundant wavelet coefficients in different

subbands to the original undecimated size; 2) apply blurring operation in the upsampled subbands; and 3) reconstruct the final blurry image by using the upsampled wavelet reconstruction matrix  $(\mathbf{U}^j \mathbf{W}_{j,m})^T$ . Because wavelet coefficients are sparse and preupsampling will not decrease the sparsity, the above wavelet domain blur model can save plenty of computation than on the original image.

### C. Extension to Nonuniform Blurring

The blurring matrix  $\mathbf{H}$  of nonuniform blur is not Toeplitz, and we cannot exchange it with wavelet decomposition matrix  $\mathbf{W}$  as in the above derivations. However, in the following, we will show mathematically that for common nonuniform deblurring, if  $\mathbf{H}$  can be regarded as a locally Toeplitz matrix, the difference between exchanged and unexchanged multiplication is limited. Correspondingly, the scheme in Fig. 1 becomes a close approximation.

To analyze the precision of our wavelet domain blur model for nonuniform blurs, we define the approximation error

$$\epsilon_{r,c} = ((\mathbf{H}\mathbf{W})_{r,c} - (\mathbf{W}\mathbf{H})_{r,c})^2 \quad (5)$$

where  $\epsilon$  is the error matrix, and  $r$  and  $c$  are the row and column indices.

Given an  $N \times N$  sized Toeplitz matrix  $\mathbf{W}$ , which only has  $2N - 1$  degrees of freedom (DOFs), for each element  $\mathbf{W}_{i,j}$ , we have  $\mathbf{W}_{i,j} = \mathbf{W}_{i+n,j+n}$ , with  $n = -N, \dots, -2, -1, 0, 1, 2, \dots, N$ . Hence, the element  $\mathbf{W}_{i,j}$  can be indexed by  $\mathbf{W}_{i-j}$  for convenience, and

$$\begin{aligned} \epsilon_{r,c} &= \left( \left( \sum_{s \in S} \mathbf{H}_{r,s} \mathbf{W}_{s,c} \right) - \left( \sum_{s \in S} \mathbf{W}_{r,s} \mathbf{H}_{s,c} \right) \right)^2 \\ &= \left( \left( \sum_{s \in S} \mathbf{H}_{r,s} \mathbf{W}_{s-c} \right) - \left( \sum_{s \in S} \mathbf{W}_{r-s} \mathbf{H}_{s,c} \right) \right)^2 \\ &= \left( \left( \sum_{s' \in S} \mathbf{H}_{r,c+s'} \mathbf{W}_{s'} \right) - \left( \sum_{s'' \in S} \mathbf{W}_{s''} \mathbf{H}_{r-s'',c} \right) \right)^2 \\ &= \left( \left( \sum_{s \in S} \mathbf{W}_s (\mathbf{H}_{r,c+s} - \mathbf{H}_{r-s,c}) \right) \right)^2 \end{aligned} \quad (6)$$

where  $S$  is the support region of wavelet basis. If and only if  $\mathbf{H}$  is Toeplitz,  $\mathbf{H}_{r,c+s} = \mathbf{H}_{r-c-s} = \mathbf{H}_{r-s,c}$  and the error  $\epsilon$  equals to zero.

Although for nonuniform blur  $\mathbf{H}_{r,c+s} \neq \mathbf{H}_{r-s,c}$ , the wavelet basis is tightly supported, and thus, for a single elements  $\mathbf{H}\mathbf{W}_{r,c}$ , the support region  $S$  of wavelet basis  $\mathbf{W}_s$  is in a small neighborhood. Therefore, if the blur matrix  $\mathbf{H}$  is locally Toeplitz, we have

$$\forall s \in \{s | \mathbf{W}_s \neq 0\}, \quad \mathbf{H}_{r,c+s} \simeq \mathbf{H}_{r-s,c}. \quad (7)$$

In other words, if the blur can be regarded as locally uniform in the support regions of the wavelet basis, our wavelet domain blur model can well approximate the spatial domain blur model. The final accuracy of our model depends on the size of support region of wavelet basis. Benefiting from the tightly supported basis, the wavelet analysis has excellent

time/space-frequency resolution and thus can handle the nonuniform blur much more accurately than the existing patch-based methods, whose proper patch size is significantly larger than the support region of wavelet basis. In addition, our experiments show that with the patch size decreasing, the patch-based algorithms have to deal with more patches and would lose its advantage in running speed.

Note that in this paper, the blurring matrix  $\mathbf{H}$  is computed from a unified camera motion (as the 6D motion trace recorded by an inertial sensor [23] or derived by a blind motion estimation algorithm [38]). Since the wavelet coefficient  $\mathbf{C}$  is sparse, the columns in the blurring matrix  $\mathbf{H}$  (describing the local blur kernels) corresponding to the zero elements of  $\mathbf{C}$  are not required to be computed. Conversely, all the local blur kernels of the nonzero elements are computed to make sure the accuracy.

### III. WAVELET REGULARIZED MULTILAYER DEBLURRING

Based on the wavelet domain blur model, many existing nonblind deblurring methods can be accelerated by replacing the spatial convolution module directly. To further utilize the rapidity brought by the sparsity of wavelet coefficients, we propose a multilayer wavelet domain deblurring framework corresponding to the hierarchical wavelet decomposition.

Most deblurring methods using wavelet regularization can be formulated as

$$J(\mathbf{L}^*) = \arg \min_{\mathbf{L}} \|\mathbf{H}\mathbf{L} - \mathbf{B}\|_2^2 + \lambda \|\mathbf{C}\|_1 \quad (8)$$

where  $\mathbf{B}$  is the blurry image,  $\mathbf{C}$  is the wavelet coefficients of latent image  $\mathbf{L}$ , and  $\lambda$  is the weighting factor. Mathematically, the TL algorithm alternatively performs Landweber updates

$$\mathbf{L}^{t+1} = \mathbf{L}^t + \tau \mathbf{H}^T (\mathbf{B} - \mathbf{H}\mathbf{L}^t) \quad (9)$$

and wavelet coefficient thresholding

$$\begin{aligned} \mathbf{C} &= \mathcal{T}_{\lambda\tau/2}(\mathbf{C}) = \text{sgn}(\mathbf{C})(|\mathbf{C}| - \lambda\tau/2)_+ \\ (x)_+ &= \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (10)$$

with  $\tau$  being the updating step size.

Original TL algorithm converges slowly, so Vonesch and Unser [17] proposed to accelerate by updating each subband separately for the Shannon wavelet. To generalize the method to arbitrary wavelet basis, they [7] propose to apply 1) and 2) iteratively in different subbands

$$\begin{aligned} 1) \quad \mathbf{C}_{j,m}^t &= \mathbf{C}_{j,m}^t + \tau_{j,m} \mathbf{r}_{j,m} \\ 2) \quad \mathbf{C}_{j,m}^{t+1} &= \mathcal{T}_{\lambda\tau_{j,m}/2}(\mathbf{C}_{j,m}^t) \end{aligned} \quad (11)$$

where  $\mathbf{r}_{j,m}$  is the projection of residual in subband  $(j, m)$

$$\mathbf{r}_{j,m} = \mathbf{W}_{j,m} \mathbf{H}^T (\mathbf{B} - \mathbf{H}\mathbf{L}). \quad (12)$$

Vonesch and Unser's [7], [17] methods converge much faster than the standard TL algorithm, since they use larger update steps  $\tau_{j,m}$  in most subbands. Therefore, we also resort to multilayer deblurring and flexible step sizes.

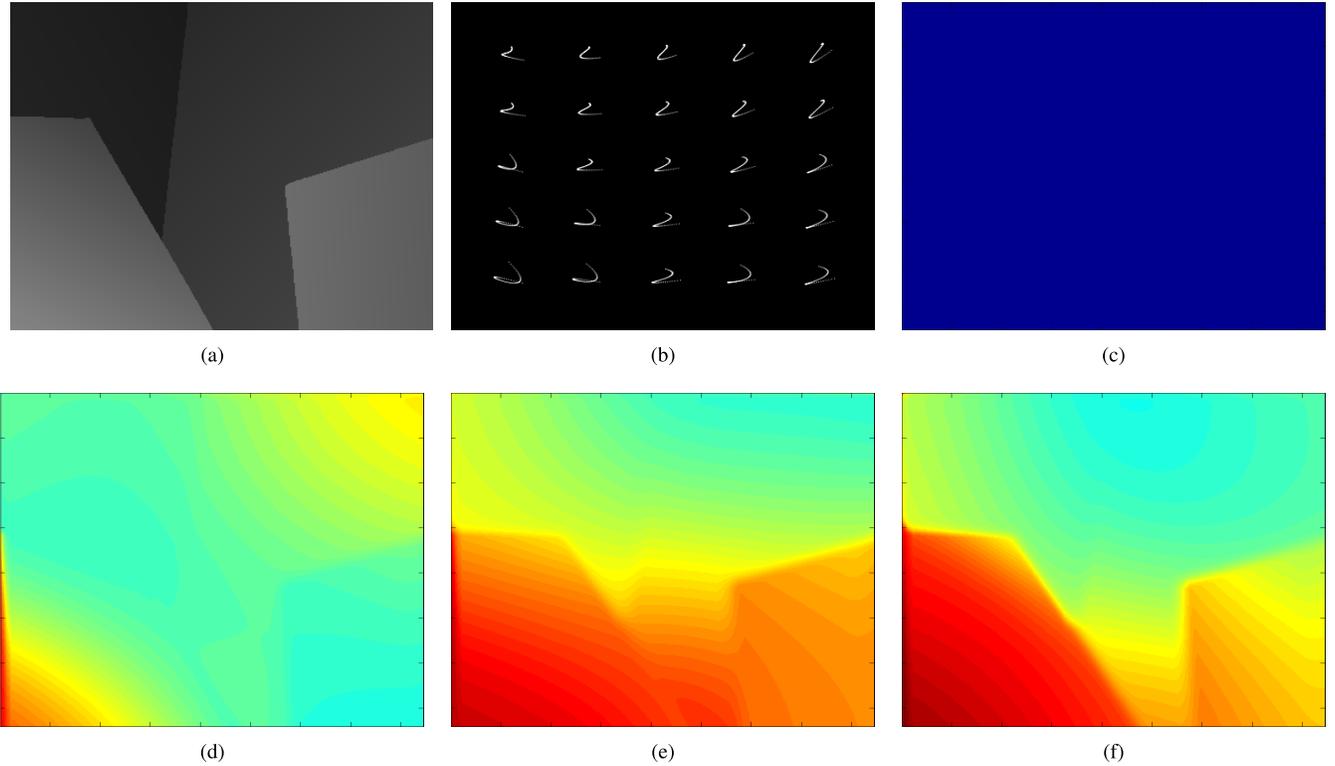


Fig. 2. Nonuniform step sizes at the third level wavelet transform. (b) Nonuniform blur generated from a 6D camera motion and depth map in (a). (c)–(f) Nonuniform step size maps in four subbands, ranging from one (blue) to six (red).

### A. Nonuniform Step Size

Theoretically, the optimal update step  $\tau_{j,m}$  for a specific kernel can be computed from its discrete Fourier coefficients. Considering the self-containment of this paper, we give the equation for computing the step size  $\tau_{j,m}$  of a given local blur kernel  $\mathbf{K}$

$$\tau_{j,m}(\mathbf{K}) = \frac{1}{\sum_{n \in \mathbb{S}_j} \rho_{j,m,n}(\mathbf{K})} \quad (13)$$

where  $\mathbb{S}_j$  is the set of all the subbands in level  $j$ , and both  $m$  and  $n$  are the subband indices and

$$\rho_{j,m,n}(\mathbf{K}) = \max_{\omega_1, \omega_2} |\text{conj}(\mathcal{F}(\mathbf{W}_{j,m}))\mathcal{F}(\mathbf{W}_{j,n})\mathcal{F}(\mathbf{K})| \quad (14)$$

in which  $\mathcal{F}(\cdot)$  is the Fourier transformation,  $\text{conj}(\cdot)$  is the conjugate operation,  $(\omega_1, \omega_2)$  indexes the element in Fourier domain, and  $\mathbf{W}_{j,m}$  and  $\mathbf{W}_{j,n}$  are the wavelet transformations from the spatial domain to the subbands  $(j, m)$  and  $(j, n)$ , respectively.

For nonuniform deblurring, both blur kernels and the optimum  $\tau_{j,m}$  values are spatially varying, so a fixed step size (minimum across the whole image lattice) largely offsets the advantage of multilayer deconvolution framework. Therefore, we propose to use a nonuniform step size map in each subband.

Dense and exact computation of  $\tau_{j,m}$  values is difficult, and is also empirically unnecessary. Therefore, we propose to compute optimal step sizes at several sampled positions and interpolate the rest pixels by bilateral propagation. In this paper, we sample the positions to compute the local blur

kernels in a gridwise manner. A uniformly distributed  $M \times M$  point grid on the image is sampled, and here,  $M$  is empirically set to be 5~8, according to the spatial variance of the blurring effect. The local blur kernels of these points are computed by using the perspective geometry from the camera parameters, the camera motion, and the scene depth, which are assumed to be known for this nonblind scenario. The related computation process for local blur kernels is a little bit complex but not the technical contribution of this paper, so we add it in the Appendix. As an example, a computed local blur kernel map of the sampled reference points is shown in Fig. 2(b). A bilateral filter extends the spatial Gaussian filter by introducing external guidance (usually intensity). In our scenario, depth and spatial coordinates are two main factors determining the nonuniformity of blur kernels. Therefore, the step size at a certain pixel  $\mathbf{x}$  can be interpolated from its neighboring pixels with known step sizes according to depth continuity and spatial proximity

$$\tau_{j,m}(\mathbf{x}) = \frac{\sum_{\mathbf{x}' \in N(\mathbf{x})} G_s(\mathbf{x}) \cdot G_d(\mathbf{x}) \cdot \tau_{j,m}(\mathbf{x}')}{\sum_{\mathbf{x}' \in N(\mathbf{x})} G_s(\mathbf{x}) \cdot G_s(\mathbf{x})} \quad (15)$$

in which

$$G_s(\mathbf{x}) = e^{-\frac{(D(\mathbf{x})-D(\mathbf{x}'))^2}{2\sigma_d^2}} \quad \text{and} \quad G_d(\mathbf{x}) = e^{-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma_s^2}} \quad (16)$$

where  $N(\mathbf{x})$  is  $\mathbf{x}$  neighborhood positions of the sampled local blur kernel set. We naturally use the step sizes of the four kernels around  $\mathbf{x}$ , i.e., the nearest kernels on its upper, under, left, and right side, to interpolate the step size in the pixel.  $G_d(\cdot)$  and  $G_s(\cdot)$  are both Gaussian filters,  $\sigma_d$  and  $\sigma_s$  are,

**Algorithm 1** Multilayer Deblurring Framework**Input:** Blurry image  $\mathbf{B}$  and blur matrix  $\mathbf{H}$ **Output:** Latent sharp image  $\mathbf{L}$ 

/\*Initialization\*/

- Sample representative local kernels and corresponding patches;
- Precompute the step size map of each subband;
- Initialize the latent sharp image  $\mathbf{L}^0$  and residual  $\mathbf{r}^0 = \mathbf{H}^T(\mathbf{B} - \mathbf{H}\mathbf{L}^0)$ ;

/\*Updating\*/

**for**  $t = 0$  to  $T$  **do**  **for**  $j = 1$  to  $j_{max}$ ,  $\forall m$  **do**

/\*Single layer update\*/

- Compute wavelet projection of residual image by  $\mathbf{r}_{j,m}^t = \mathbf{W}_{j,m}\mathbf{r}^t$ ;
- Update coefficients in each subband by  $\mathbf{C}_{j,m}^t = \mathbf{C}_{j,m}^t + \tau_{j,m}\mathbf{r}_{j,m}^t$ ;
- Threshold coefficients in each subband by  $\mathbf{C}_{j,m}^{t+1} = \mathcal{T}_{\lambda\tau_{j,m}/2}(\mathbf{C}_{j,m}^t)$ ;
- Reproject  $\mathbf{r}_{j,m}^t$ 's modification  $\Delta\mathbf{r}_{j,m}^t$  by (20);
- Correct the residue following  $\mathbf{r}^{t+1} = \mathbf{r}^t + \Delta\mathbf{r}^t$ .

**end****end**

respectively, the depth and spatial variance of the Gaussian weight, and  $D(\mathbf{x})$  is the depth at  $\mathbf{x}$ . Here, the value of  $\sigma_d$  depends on the range of the scene depth. Practically, we set  $\sigma_d$  to be 0.01DR (DR is the range of the depth map) in this paper. The spatical variance  $\sigma_s$  determines the spatially propagating scope of a single local kernel. In this paper, we set that  $3\sigma_s$ , which is the nearly zero limit for the Gaussian distribution, equals to the interval between two adjacent kernels. Fig. 2 shows the nonuniform step size maps of a 6D camera shake blur computed by our bilateral propagation method and displays apparent depth dependence of optimum step sizes.

**B. Coarse-to-Fine Deblurring**

After determining the step size map in each subband, we propose a wavelet-based coarse-to-fine deblurring strategy, with the key steps listed in Algorithm 1.

1) *Initialization*: The latent sharp image is initialized by filter flow-based method [27] with Gaussian prior [39] as

$$\mathbf{L}^0 = \sum_{i \in P} \mathbf{w}_i \frac{\text{conj}(\mathcal{F}(\mathbf{K}_i))\mathcal{F}(\mathbf{B})}{|\mathcal{F}(\mathbf{K}_i)|^2 + \lambda(|\mathcal{F}(\nabla_h)|^2 + |\mathcal{F}(\nabla_v)|^2)} \quad (17)$$

where  $P$  denotes the set of patches at the selected positions for computing the step size maps. The patch centers are coincide with the selected points in the grid form, and the patch size is set to ensure that the overlap between the adjacent patches is larger than the maximal size of the sampled  $M \times M$  local blur kernels;  $\mathcal{F}(\cdot)$  is the Fourier transform function, and  $\mathbf{K}_i$  is the local blur kernel at patch  $i$ ;  $\nabla_h$  and  $\nabla_v$  are the gradient filter in horizontal and vertical directions, respectively;  $\lambda$  is the weight for gradient regularizer;  $\mathbf{w}_i$  is the weight of the pixels in the  $i$ th patch for stitching the patches back to the

entire image

$$\mathbf{w}_i(x, y)|_{i \in P(x, y)} \propto e^{-\frac{(x-x_i)^2 + (y-y_i)^2}{2\sigma_w^2}} \quad (18)$$

where  $P(x, y)$  is the set of all the patches containing the pixel  $(x, y)$ ,  $(x_i, y_i)$  is the center of the patch  $i$ , and  $\sigma_w$  is set to be 1/6 of the patch size, so that according to the  $3\sigma$  rule of the Gaussian function, the weight of pixels in the border of each patch is about zero. The weights of each pixel for all the patches in  $P(x, y)$  have to be normalized to make sure the summation  $\sum_{i \in P(x, y)} \mathbf{w}_i(x, y) = 1$ . Then, after having  $\mathbf{L}^0$ , the residual image can be derived by  $\mathbf{r}^0 = \mathbf{H}^T(\mathbf{B} - \mathbf{H}\mathbf{L}^0)$ .

2) *Single Layer Update*: Since the subbands in the same layer do not couple with each other, we update the coefficients within a layer altogether. In particular, we calculate the values of  $\mathbf{r}$  wavelet decomposition  $\{\mathbf{r}_{j,m}, j = 1 \dots j_{max}\}$  and multiply it with the precomputed step size map in the pixelwise manner, and then, the wavelet coefficients  $\mathbf{C}$  can be updated by (11).

After updating  $\mathbf{C}$ , the residual term  $\mathbf{r} = \mathbf{H}^T(\mathbf{B} - \mathbf{H}\mathbf{L})$  needs to be corrected accordingly. Since only coefficients in a single layer change, the incremental residual can be computed by

$$\begin{aligned} \Delta\mathbf{r}_j &= \mathbf{H}^T \mathbf{H} \sum_m \mathbf{W}_{j,m}^T \Delta\mathbf{C}_{j,m} \\ &= \sum_m (\mathbf{U}\mathbf{W}_{j,m})^T \mathbf{H}^T \mathbf{H} \mathbf{U}^j \Delta\mathbf{C}_{j,m} \end{aligned} \quad (19)$$

where  $\Delta\mathbf{C}_{j,m}$  is the updating quantity of subband coefficients  $\mathbf{C}_{j,m}$  computed from (11). It is reasonable to assume that the residual image  $\mathbf{r}$  is sparse and becomes sparser with the estimation of sharp image  $\mathbf{L}$  converging, so the update in each iteration is also sparse. Therefore, to accelerate (19), we can truncate the small values before conducting the two blurring processes (i.e., multiplying  $\mathbf{H}$  and  $\mathbf{H}^T$ ), and then, (19) becomes

$$\Delta\mathbf{r}_j = \sum_m (\mathbf{U}\mathbf{W}_{j,m})^T \mathbf{H}^T \mathcal{T}_{\text{thr}_2}(\mathbf{H}\mathbf{U}^j \mathcal{T}_{\text{thr}_1}(\Delta\mathbf{C}_{j,m})) \quad (20)$$

where  $\text{thr}_1$  and  $\text{thr}_2$  are the truncation thresholds.

As for the settings of these two truncation thresholds, because  $\mathbf{r}^t = \mathbf{H}^T(\mathbf{B} - \mathbf{H}\mathbf{L}^t)$  includes the imaging noise besides the estimation error of  $\mathbf{L}^t$ , we hope to perform noise suppression to ensure a high-quality deconvolution result. Here, we refer to the Visushrink thresholding truncation strategy in Donoho and Johnstone's [40] work. Different from Visushrink denoising based on a universal threshold  $\lambda_{\text{visu}}$  that is often much higher than the optimum, we favor a more conservative estimation to preserve that the crucial details preserve the signal components well. Therefore, we empirically set  $\text{thr}_1 = 0.1\lambda_{\text{visu}}$  and  $\text{thr}_2 = 0.01\lambda_{\text{visu}}$ .

Fig. 3(a) compares the running time of the proposed thresholded and nonthresholded correction in each iteration. The difference clearly demonstrates the superior efficiency of thresholded correction to its nonthresholded counterpart. In addition, the residual maps at iterations 1, 5, and 10 are shown. It is clear that the sparsity of the residue map largely increases with the iteration and the computation time of each iteration also decreases correspondingly. In all, the computation efficiency significantly benefits from the sparsity property of the residual component. Note that the proposed

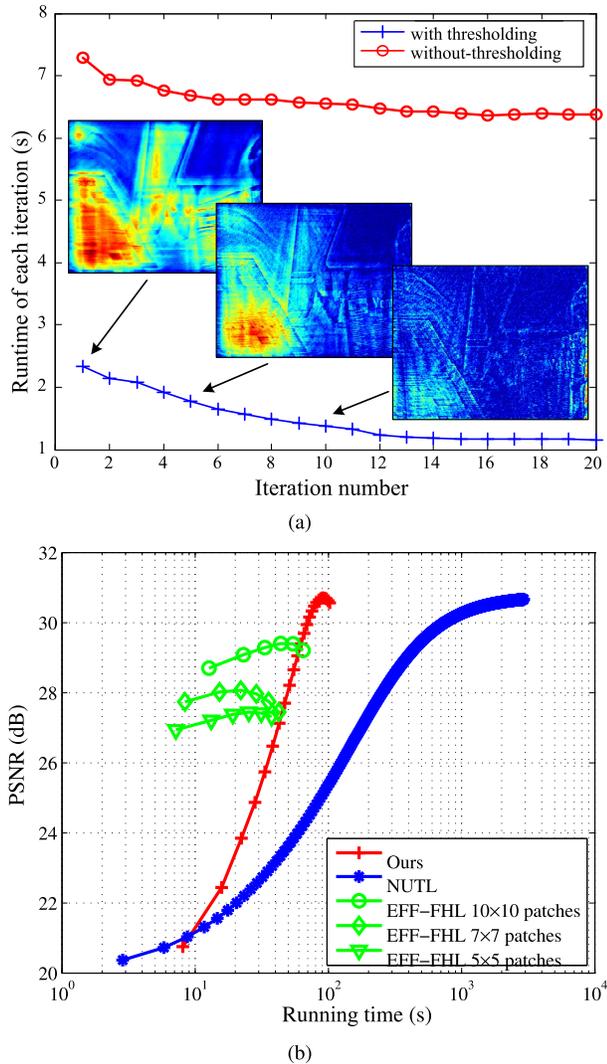


Fig. 3. Analysis on algorithm efficiency and performance. (a) Running time with and without thresholding, with the residual error maps in the first, fifth, and tenth iteration displayed. (b) Plots of the running time and the performance, with the results obtained by applying algorithms on  $512 \times 512$  pixel images with slowly varying blur kernels.

method can either be terminated at a fixed iteration number  $T$ , or stop when the update between two iterations is smaller than a constant. In this paper, we found that the algorithm always achieves the best performance on about 30 iterations, so we choose the former method and set  $T = 30$ .

#### IV. EXPERIMENTS AND ANALYSIS

In this section, we conduct a series of experiments to verify the efficiency and effectiveness of our wavelet domain nonuniform deblurring algorithm. The nonuniform deconvolution algorithm can be divided into two types: the pixelwise algorithms and the patchwise ones. Due to their inherent defect, all the pixelwise algorithms are extremely slow, since they have to compute the PSFs pixel by pixel. Conversely, the patchwise algorithms are much faster and of lower accuracy. For comparison, we choose the pixelwise nonuniform TL (NUTL) that is of high accuracy and the fast nonuniform deconvolution algorithm, which is highly efficient by

combining EFF [27] and fast uniform deconvolution with hyper-Laplacian prior [8] (EFF-FHL for short). The results of EFF-FHL methods at different patch numbers are shown, with the patch size determined by the image size and the overlap between adjacent patches. Here, we set the overlap, which is equal to the kernel size to eliminate blocking artifact. To reduce the support region size of the wavelet basis, we use the Symlet 8 wavelet in our implementation.

##### A. Discussion on Running Time and Accuracy

Benefiting the sparsity of wavelet coefficients and the compactness of wavelet hierarchy, the proposed approach raises computation efficiency significantly. Meanwhile, our approach is of high accuracy due to the small support region of wavelet basis. Here, we compare the running time of our deblurring algorithm with those of two aforementioned state-of-the-art algorithms—NUTL and EFF-FHL, and plot their convergence curves in Fig. 3(b). We test the algorithm on the Miscellaneous dataset of USC-SIPI Image Database [41]. In sum, 26 images with the same size ( $512 \times 512$ ) are used, and the performance, i.e., the peak signal-to-noise ratios (PSNRs) and the running time, is averaged over these images. The program is implemented with MATLAB and run on a workstation with Intel Core i5-3470 (3.2 GHz), 8-GB RAM.

From the comparison, we can clearly see that the running time is comparable with EFF-FHL while significantly shorter than NUTL. Although being slightly slower than EFF-FHL, our algorithm is much more accurate in the scenarios when the assumption of smoothly varying blur kernels in EFF-FHL is violated. In such cases, EFF-FHL needs quite small patches to approximate the real blur kernels and results in higher final performance at the expense of increasing running time, as plotted by the green curves. However, the performance reaches the optimum at a certain patch number ( $10 \times 10$  patches here), which is determined by the size and local nonuniformity of blur kernels, because the abruptly changing kernels cannot be approximated accurately. Differently, our approach only assumes uniformity within the support region of wavelet and thus obtains higher performance (here we used PSNR), as shown in Fig. 3(b).

It is worth noting that the efficiency of our algorithm depends on the texture richness, so the running time on extremely complex textures cannot be reduced largely. Fortunately, for the general natural images, the rich textures usually occupy a small proportion and the proposed algorithm is efficient statistically.

Besides, the efficiency of the proposed algorithm does not highly depend on the adopted wavelet types. Fig. 4 shows the convergence curves with different wavelets. Four wavelets with different taps, i.e., Haar, Symlet 4, Symlet 8, and Daubechies 9.7, are applied, and we can see that the running time and the performance of these wavelets are at the same level, except that the running time of Haar is slightly longer. That is because our implementation for wavelet decomposition and reconstruction is based on the FFT, so that the short-tap wavelets, e.g., Haar, do not have the advantages in computational efficiency. Besides, since the coefficients of the other

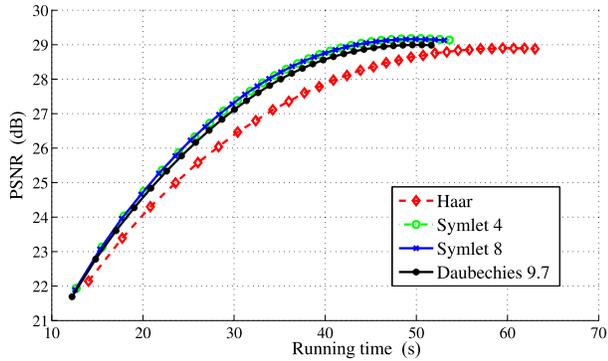
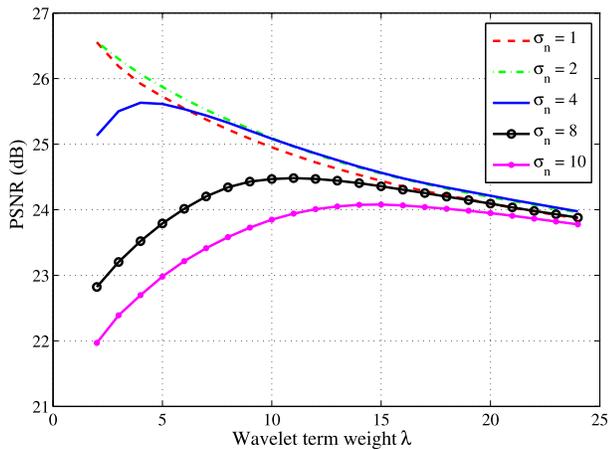
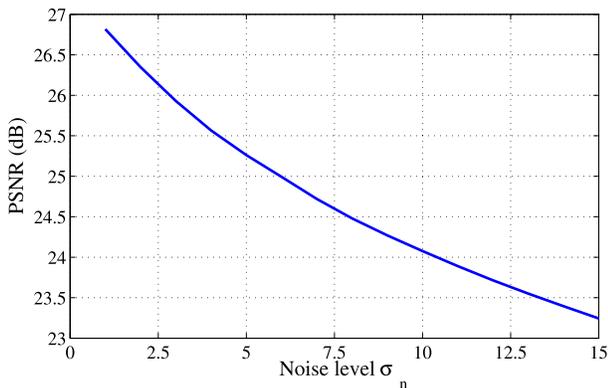


Fig. 4. Average PSNR (y-axis) and the running time (x-axis) with different wavelets.



(a)



(b)

Fig. 5. Analysis on noise tolerance. (a) PSNR (y-axis) with different weights of the wavelet regularization term ( $\lambda$ ) at different noise levels ( $\sigma_n$ ). (b) PSNR performance of the proposed algorithm by using empirically setting  $\lambda = \sqrt{2}\sigma_n$  on the images with different noise levels.

complex wavelets may be sparser, the proposed algorithm is more efficient by using more complex wavelets, e.g., Symlet 4, Symlet 8, and Daubechies 9.7.

We also test the algorithm on the synthetic data with different noise levels. As shown in Fig. 5(a), the PSNR of the results on different noise level  $\sigma_n$  images with different wavelet term weights  $\lambda$  is given. We see that as the noise level increases, the optimal weight increases as well. Empirically, the best

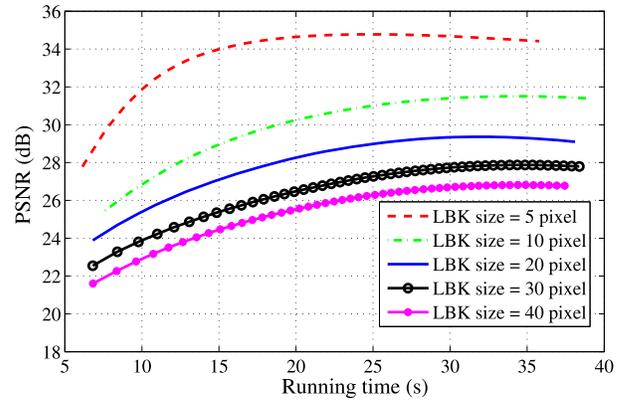


Fig. 6. PSNR (y-axis) and the running time (x-axis) with different blur sizes. The LBK size refers to the maximum local blur kernel size.

weight of the wavelet regularization term  $\lambda$  is around  $\sqrt{2}\sigma_n$ . The PSNR curve by using  $\lambda = \sqrt{2}\sigma_n$  on the images with different noise levels is shown in Fig. 5(b). The performance reduces slightly with the noise level increases. However, even for the very challenging case, i.e.,  $\sigma_n = 15$  (the intensity range is  $0 \sim 255$ ), the proposed algorithm still gives promising result.

To analyze the effect of blur kernel size, we test the algorithm on the images with different blur sizes. A series of blurry images with a similar blur pattern but different blur sizes is generated from a single motion trace multiplying with different ratios. The PSNR versus runtime curves for different blur sizes are shown in Fig. 6. We can see that for the smaller blur, the algorithm converges faster, but even for the very large blur (local blur kernel size is 40 pixels), the algorithm converges at about 30 iterations. The computation of each iteration is almost the same, so that our algorithm with constant iteration number ( $T = 50$  in this experiment) finished at about the same time. As for the deblurring performance, the PSNR of the results decreases with increasing blur size, and this trend consists with the intuition.

### B. Comparisons on Synthetic Data

1) *Slowly Varying Blur*: We compare the performances of our algorithm, EFF-FHL and NUTL on a nonuniformly blurred image caused by the 3D camera rotation, and the results are shown in Fig. 7. Apparently, both our algorithm and the NUTL algorithm achieve good performance across the whole image, while there are some artifacts in the final results of EFF-FHL.

In terms of running time, our algorithm can be about 30 times faster than the NUTL method, but a little inferior to EFF-FHL. The artifacts in the EFF-FHL are mainly caused by inaccurate approximation of the true blur kernels. Fig. 7(e)–(g) shows the results of EFF-FHL with different patch numbers, from which we can see that as the patch number increases, EFF-FHL obtains higher recovery quality while takes longer running time. When the patch number increases to  $10 \times 10$ , the computation time is not apparently advantageous over our algorithm, while the PSNR of the result is still lower. Furthermore, as mentioned earlier, further increasing patch number will not raise the performance of EFF-FHL.

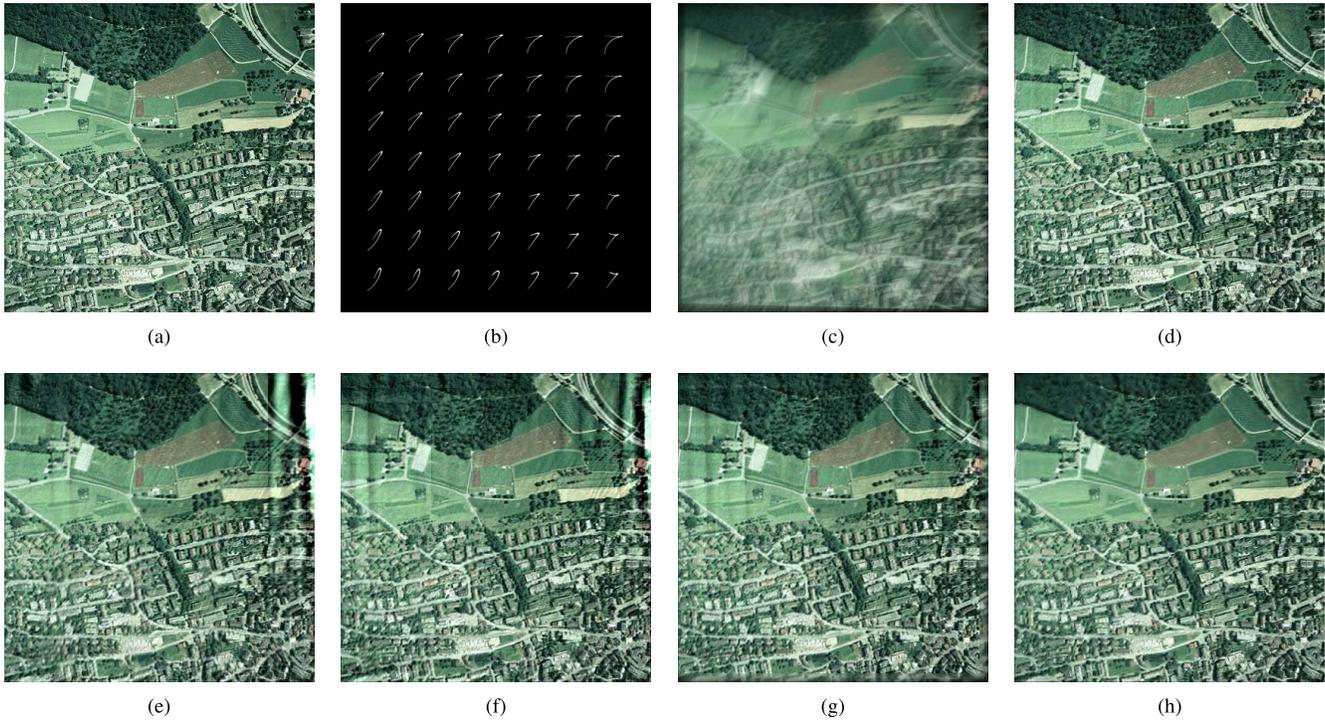


Fig. 7. Restoration results of a blurry input image with slowly varying nonuniform blur kernels caused by a synthetic 3D camera rotation. Parameter settings, performance, and running time are labeled under each subfigure. The original aerial image is provided by Zurich Hoegg data set. (a) Original sharp image. (b) Blur kernels. (c) Blurred image. (d) Ours, PSNR = 23.66 dB and  $t = 198.5$  s. (e) EFF-FHL,  $5 \times 5$  patches, PSNR = 19.07 dB, and  $t = 112.5$  s. (f) EFF-FHL,  $7 \times 7$  patches, PSNR = 20.63 dB, and  $t = 126.9$  s. (g) EFF-FHL,  $10 \times 10$  patches, PSNR = 22.37 dB, and  $t = 193.4$  s. (h) NUTL, PSNR = 22.9 dB and  $t = 5842$  s.

Overall, the EFF-FHL method is more suitable for the tasks with high-speed requirement, since it tends to give a reasonable result in very short time. In performance demanding cases, both our method and NUTL work well, while our algorithm runs as fast as the EFF-FHL method, and it can be applicable in most scenarios.

2) *Intensely Varying Blur*: We also test the performances of three algorithms on depth-dependent nonuniformly blurred images caused by a large 6D camera motion, and the results are shown in Fig. 8. Comparing the subfigures, we can see the same trend as in the last experiment: our algorithm achieves comparable result with the NUTL algorithm but at a significantly faster speed, and the EFF-FHL method does not work as well as ours in this example.

There are lots of fringes in some regions of EFF-FHL's results at each patch size setting, and these artifacts lead to significant quality deterioration, both visually and quantitatively (in PSNR). The causation of these local artifacts can be attributed to the twofold criteria for the selection of patch size: 1) the patch should be small enough to ensure within-patch uniformity and thus approximate the true kernels, especially in the regions with drastically changing blur patterns and 2) the patch should be large enough to cover the blur kernel size, or the blur model cannot describe the true blur process correctly. Therefore, a uniform decomposition will cause some fringes in some regions while perform better in the others, where the patch size is large enough to cover the local blur kernel and small enough to ensure local uniformity. This also validates the advantages of the proposed wavelet

domain acceleration in such complex cases, due to a much looser assumption on the local uniformity of blur kernels.

### C. Comparisons on Real Captured Database (Köhler *et al.* [42])

To verify the effectiveness of the proposed method on real captured images, we apply our algorithm on the benchmark data set captured by Köhler *et al.* [42]. They acquire 12 blurry versions for each of the four sharp images by mounting the camera on a delicate six DOF Stewart motion platform, which playbacks different motion trajectories. In total, this database consists 48 blurred images, with both the ground truth motion trajectories and the sharp images provided. The PSFs can be computed from the motion trajectories and some other camera parameters, e.g., focal length and scene distance. Considering that the kernel nonuniformity of some images in this database is not large, we also test the performances of some state-of-the-art uniform deblurring algorithms [2], [4] by choosing the calculated PSF at the image center as the input kernel.

Table I lists the quantitative (PSNR in dB) comparison among the proposed algorithm and the other state-of-the-art methods on Köhler *et al.*'s [42] database. It is obvious that the proposed approach achieves the best performance, while other methods, either uniform or patch-based nonuniform deblurring, suffer from the nonuniformity of the real blur kernels. Fig. 9 shows the visual comparison among different algorithms on one example from Köhler *et al.*'s [42] database

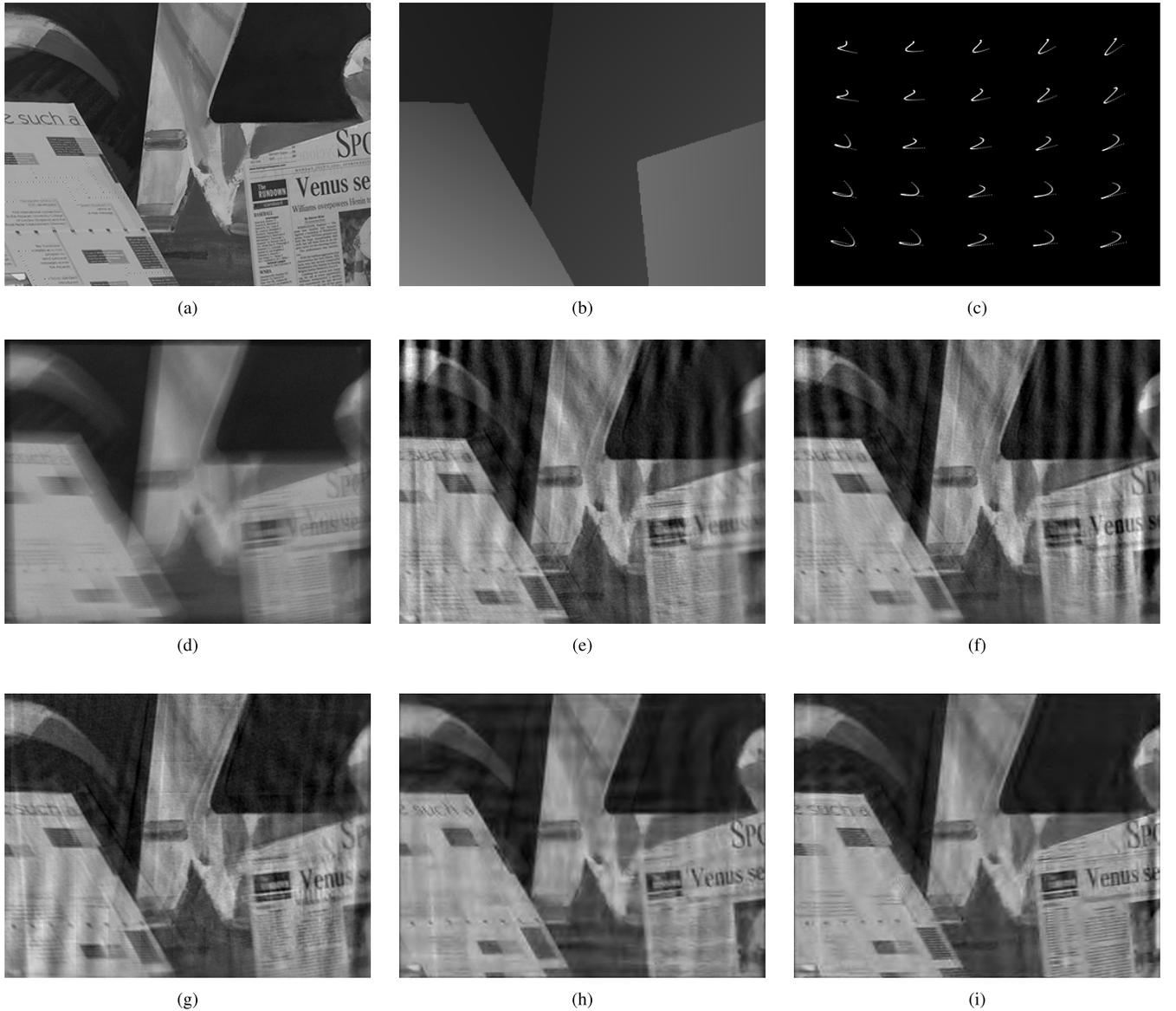


Fig. 8. Restoration results of an input synthetic blurry image degenerated by intensely varying nonuniform blur caused by a random 6D camera motion. Parameter settings, performance, and running time are labeled under each subfigure. (a) Original sharp image. (b) Depth. (c) Blur kernels. (d) Blurred image. (e) EFF-FHL,  $5 \times 6$  patches, PSNR = 19.66 dB, and  $t = 17.3$  s. (f) EFF-FHL,  $8 \times 9$  patches, PSNR = 21.07 dB, and  $t = 46.3$  s. (g) EFF-FHL,  $15 \times 18$  patches, PSNR = 23.27 dB, and  $t = 129.4$  s. (h) NUTL, PSNR = 25.75 dB, and  $t = 536.8$  s. (i) Ours, PSNR = 25.23 dB and  $t = 47.4$  s.

TABLE I  
QUALITATIVE COMPARISON (PSNR IN DECIBELS) WITH STATE OF THE ARTS ON KÖHLER *et al.*'s BENCHMARK DATABASE [42]

Methods	Image 01	Image 02	Image 03	Image 04	Average
Ours	<b>30.5583</b>	<b>28.7769</b>	<b>30.3166</b>	<b>29.1093</b>	<b>29.6903</b>
EFF-FHL( $4 \times 4$ )	28.6695	26.3968	27.8917	26.2424	27.3001
EFF-FHL( $7 \times 7$ )	28.5854	26.2374	27.6131	26.3355	27.1928
Schuler <i>et al.</i> [2]	26.0022	19.4102	23.5868	21.4316	22.6077
Zoran <i>et al.</i> [4]	26.8934	19.9867	24.3828	19.8019	22.7662

and the running time is also provided. In terms of visual quality, we can see that two learning-based uniform deblurring algorithms can effectively suppress the ringing artifacts caused by the lost high-frequency and inaccurate kernel estimation, while there exists some slight ringing in the results of our

approach and the EFF-FHL method. However, the PSNRs of these two uniform deblurring algorithms are slightly lower than those of the EFF-FHL ( $7 \times 7$ ) algorithm and ours, which apparently have some artifacts. The inconsistency from the visual quality and quantitative evaluation mainly comes

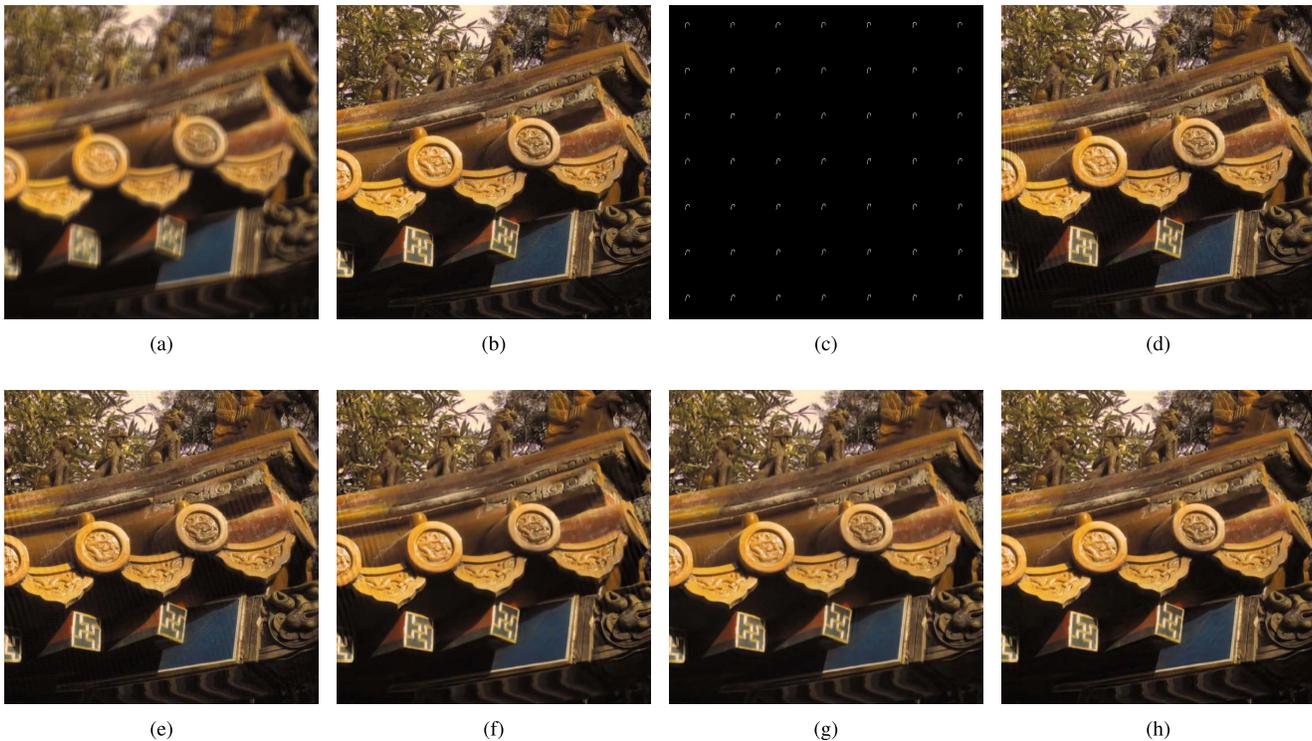


Fig. 9. Restoration results of an example from the benchmark database [42] for the deblurring algorithm evaluation. Parameter settings, performance, and running time are labeled under each subfigure. (a) Observed blurry image. (b) Ground truth sharp image. (c) Blur kernels. (d) EFF-FHL,  $4 \times 4$  patches, PSNR = 29.59 dB, and  $t = 54.2$  s. (e) EFF-FHL,  $7 \times 7$  patches, PSNR = 30.99 dB, and  $t = 138.3$  s. (f) Schuler *et al.*'s [2], PSNR = 29.92 dB, and  $t = 104.5$  s. (g) Zoran and Weiss's [4], PSNR = 30.9728 dB, and  $t = 2749.6$  s. (h) Ours, PSNR = 33.70 dB and  $t = 212.1$  s.

from the fact that learning-based methods reconstruct the latent sharp image using not only the information from input image but also from an artifact-free natural image database. Although with higher visual quality, the results from learning-based approaches may introduce some distortions or hallucination infidelity compared with the ground truth. Comparing the two nonuniform deblurring approaches, our algorithm obtains superior performance, both visually and quantitatively. Although the patch-based method (EFF-FHL) achieves promising performance using smaller patches and at expenses of higher computation cost, our restoration results are still of higher quality. The superior performance verifies the effectiveness of our algorithm.

## V. CONCLUSION

This paper proposes an efficient nonuniform nonblind deblurring approach, in which the calculation and the regularization are both performed in the wavelet domain. In term of performance, our approach achieves performance superior or comparable with the state-of-the-art nonuniform algorithms. In terms of efficiency, we obtain speed significantly faster than the pixelwise algorithms and comparable with the EFF-based method, which is, to the best of our knowledge, the fastest published algorithms.

After transforming the blur model and deblurring algorithm to wavelet domain, the preprocess and postprocess steps—wavelet decomposition and reconstruction—cost a large portion of running time. In other words, wavelet transformation has replaced the blurring process to become the bottleneck

of speed, so we will investigate these problems and raise the efficiency further in the future work.

In order to realize real-time video deblurring, the current algorithm is still not fast enough without the aid of hardware. However, the proposed algorithm can work parallelly, and thus, the GPU-based implementation is potentially to achieve real-time processing. After the algorithm is mature enough, designing the corresponding dedicated hardware chips is another acceleration solution and on the list of our future work as well.

Besides, the blind deblurring is a useful task and also can benefit from the high computation efficiency of the wavelet domain deconvolution, so blind deconvolution in the wavelet domain is as well a worth studying problem. In the future, we will try to extend the proposed algorithm to blind case.

## APPENDIX

### CALCULATION OF LOCAL BLUR KERNELS

This paper focuses on the nonblind deblurring, which assumes the camera intrinsic and motion, and the scene depth is known. We denote the camera motion during exposure time  $[t_s, t_e]$  as  $\mathbf{M}(t)$ . At each instant, the camera pose is described with at most six DoFs— $\{T_x, T_y, T_z, \theta_x, \theta_y, \theta_z\}$ , which represent the translation along and rotation around  $x$ -,  $y$ -,  $z$ -axes, respectively. Mathematically,  $\mathbf{M}(t)$  is an element in a 6D continuous function space or its subspace

$$\mathbf{M}(t) \in \mathbb{C}[t_s, t_e]^{\text{Dim}}, \quad 2 \leq \text{Dim} \leq 6 \quad (21)$$

where  $\mathbb{C}$  is the continuous function space defined over exposure time interval  $[t_s, t_e]$ . To facilitate representation and derivation, translation is represented by a vector  $\mathbf{T} = [T_x, T_y, T_z]'$ , and the rotation in axis angles is transformed to a matrix-by-matrix exponential  $\mathbf{R} = e^{[\boldsymbol{\theta}]_{\times}}$

$$[\boldsymbol{\theta}]_{\times} = \begin{bmatrix} 0 & -\theta_z & \theta_y \\ \theta_z & 0 & -\theta_x \\ -\theta_y & \theta_x & 0 \end{bmatrix}. \quad (22)$$

Because of camera motion during exposure, a single scene point projects to multiple pixels along a continuous trajectory, which depends on both the 3D coordinate of the scene point and camera motion during exposure. Fixing the focal length, the projection at time  $t$  can be expressed by

$$\mathbf{x}'(t) \sim \mathbf{K}(\mathbf{R}(t)\mathbf{X} + \mathbf{T}(t)) \quad (23)$$

where  $\mathbf{x}'(t)$  is the point spread trajectory (we borrow the concept of PSF) in image plane of scene point  $\mathbf{X}$ ,  $\mathbf{K}$  is the camera intrinsic matrix [43] that is assumed to be time invariant during exposure, and  $\mathbf{R}(t)$  and  $\mathbf{T}(t)$ , respectively, denote the rotation matrix and the translation vector at time  $t$ . By integrating  $\mathbf{x}'(t)$  in the time domain, the local blur kernel corresponding to point  $\mathbf{X}$  can be derived.

## REFERENCES

- [1] J. Zhang, D. Zhao, R. Xiong, S. Ma, and W. Gao, "Image restoration using joint statistical modeling in a space-transform domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 6, pp. 915–928, Jun. 2014.
- [2] C. J. Schuler, H. C. Burger, S. Harmeling, and B. Schölkopf, "A machine learning approach for non-blind image deconvolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2013, pp. 1067–1074.
- [3] U. Schmidt, C. Rother, S. Nowozin, J. Jancsary, and S. Roth, "Discriminative non-blind deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 604–611.
- [4] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 479–486.
- [5] W. Zhang, M. Zhao, and Z. Wang, "Adaptive wavelet-based deconvolution method for remote sensing imaging," *Appl. Opt.*, vol. 48, no. 24, pp. 4785–4793, 2009.
- [6] F. A. Flaherty, "Physically constrained Fourier transform deconvolution method," *J. Opt. Soc. Amer. A*, vol. 26, no. 5, pp. 1191–1194, 2009.
- [7] C. Vonesch and M. Unser, "A fast multilevel algorithm for wavelet-regularized image restoration," *IEEE Trans. Image Process.*, vol. 18, no. 3, pp. 509–523, Mar. 2009.
- [8] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-Laplacian priors," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 22, 2009, pp. 1033–1041.
- [9] V. Maik, D. Cho, J. Shin, and J. Paik, "Regularized restoration using image fusion for digital auto-focusing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 10, pp. 1360–1369, Oct. 2007.
- [10] B. Granier, P. Réfrégier, and J. Figue, "Restoration of images degraded by atmospheric turbulence by a least-squares method and a Markov process," *Opt. Lett.*, vol. 21, no. 6, pp. 423–425, 1996.
- [11] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE Trans. Signal Process.*, vol. 4, no. 7, pp. 932–946, Jul. 1995.
- [12] R. L. Lagendijk, J. Biemond, and D. E. Boeke, "Regularized iterative image restoration with ringing reduction," *IEEE Trans. Acoust., Speech Signal Process.*, vol. 36, no. 12, pp. 1874–1888, Dec. 1988.
- [13] L. B. Lucy, "An iterative technique for the rectification of observed distributions," *Astron. J.*, vol. 79, no. 6, p. 745, 1974.
- [14] W. H. Richardson, "Bayesian-based iterative method of image restoration," *J. Opt. Soc. Amer.*, vol. 62, no. 1, pp. 55–59, Jan. 1972.
- [15] X. Liu, L. Wang, J. Wang, and H. Meng, "A three-dimensional point spread function for phase retrieval and deconvolution," *Opt. Exp.*, vol. 20, no. 14, pp. 15392–15405, 2012.
- [16] T. Lатычевская, F. Gehri, and H.-W. Fink, "Depth-resolved holographic reconstructions by three-dimensional deconvolution," *Opt. Exp.*, vol. 18, no. 21, pp. 22527–22544, 2010.
- [17] C. Vonesch and M. Unser, "A fast thresholded Landweber algorithm for wavelet-regularized multidimensional deconvolution," *IEEE Trans. Image Process.*, vol. 17, no. 4, pp. 539–549, Apr. 2008.
- [18] M. Prato, R. Cavicchioli, L. Zanni, P. Boccacci, and M. Bertero, "Efficient deconvolution methods for astronomical imaging: Algorithms and IDL-GPU codes," *Astron. Astrophys.*, vol. 539, pp. 1–11, Mar. 2012.
- [19] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
- [20] R. D. Nowak and M. A. T. Figueiredo, "Fast wavelet-based image deconvolution using the EM algorithm," in *Proc. Conf. Rec. 35th Asilomar Conf. Signals, Syst. Comput.*, vol. 1, 2001, pp. 371–375.
- [21] D. S. C. Biggs and M. Andrews, "Acceleration of iterative image restoration algorithms," *Appl. Opt.*, vol. 36, no. 8, pp. 1766–1775, 1997.
- [22] Y.-W. Tai, P. Tan, and M. S. Brown, "Richardson–Lucy deblurring for scenes under a projective motion path," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1603–1618, Aug. 2011.
- [23] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski, "Image deblurring using inertial measurement sensors," *ACM Trans. Graph.*, vol. 29, no. 4, Jul. 2010, Art. no. 30.
- [24] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 491–498.
- [25] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless, "Single image deblurring using motion density functions," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 171–184.
- [26] E. S. Angel and A. K. Jain, "Restoration of images degraded by spatially varying pointspread functions by a conjugate gradient method," *Appl. Opt.*, vol. 17, no. 14, pp. 2186–2190, 1978.
- [27] M. Hirsch, S. Sra, B. Schölkopf, and S. Harmeling, "Efficient filter flow for space-variant multiframe blind deconvolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 607–614.
- [28] S. Harmeling, M. Hirsch, and B. Schölkopf, "Space-variant single-image blind deconvolution for removing camera shake," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 829–837.
- [29] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf, "Fast removal of non-uniform camera shake," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 463–470.
- [30] M. G. Löfdahl, "Multiframe deconvolution with space-variant pointspread functions by use of inverse filtering and fast Fourier transform," *Appl. Opt.*, vol. 46, no. 21, pp. 4686–4693, 2007.
- [31] Y. Zhang and K. Hirakawa, "Blur processing using double discrete wavelet transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1091–1098.
- [32] P. Escande, P. Weiss, and F. Malgouyres, "Spatially varying blur recovery. Diagonal approximations in the wavelet domain," in *Proc. Int. Conf. Pattern Recognit. Appl. Methods*, Barcelona, Spain, 2013, pp. 222–228.
- [33] P. Escande, P. Weiss, and F. Malgouyres, "Image restoration using sparse approximations of spatially varying blur operators in the wavelet domain," *J. Phys., Conf. Ser.*, vol. 464, no. 1, p. 012004, 2013.
- [34] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, Nov. 2004.
- [35] M. A. T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 906–916, Aug. 2003.
- [36] J.-L. Starck, D. L. Donoho, and E. J. Candès, "Astronomical image representation by the curvelet transform," *Astron. Astrophys.*, vol. 398, no. 2, pp. 785–800, 2003.
- [37] J.-L. Starck, M. K. Nguyen, and F. Murtagh, "Wavelets and curvelets for image deconvolution: A combined approach," *Signal Process.*, vol. 83, no. 10, pp. 2279–2283, 2003.
- [38] T. Yue, J. Suo, and Q. Dai, "High-dimensional camera shake removal with given depth map," *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2688–2703, Jun. 2014.
- [39] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Trans. Graph.*, vol. 26, no. 3, p. 70, 2007.
- [40] D. L. Donoho and J. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.
- [41] *The USC-SIPI Image Database*, accessed on Sep. 1, 2016. [Online]. Available: <http://sipi.usc.edu/database/database.php>

- [42] 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. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 27–40.
- [43] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.



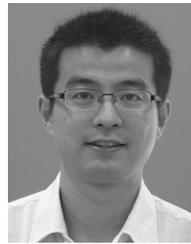
**Tao Yue** received the B.S. degree in automation from Northwestern Polytechnical University, Xi'an, China, in 2009 and the Ph.D. degree from Tsinghua University, Beijing, China, in 2015.

His research interests include computer vision, image processing, and computational photography.



**Jinli Suo** received the B.S. degree in computer science from Shandong University, Shandong, China, in 2004 and the Ph.D. degree from the Graduate University of Chinese Academy of Sciences, Beijing, China, in 2010.

She is an Assistant Professor with the Department of Automation, Tsinghua University, Beijing. Her research interests include computer vision, computational photography, and statistical learning.



**Xun Cao** (S'10–M'12) received the B.S. degree from Nanjing University, Nanjing, China, in 2006 and the Ph.D. degree from the Department of Automation, Tsinghua University, Beijing, China, in 2012.

He visited Philips Research, Aachen, Germany, in 2008 and Microsoft Research Asia, Beijing, from 2009 to 2010. He was a Visiting Scholar with University of Texas at Austin, Austin, TX, USA, from 2010 to 2011. He is currently an Associate Professor with the School of Electronic Science and Engineering, Nanjing University. His research interests include computational photography, image-based modeling and rendering, and 3D TV systems.



**Qionghai Dai** (SM'05) received the Ph.D. degree in automation from Northeastern University, Shenyang, China, in 1996.

He has been with the Department of Automation, Tsinghua University, Beijing, China, as a Faculty Member since 1997 and as a Professor since 2005. His work is motivated by challenging applications in the fields of computer vision, computer graphics, and robotics. He has authored over 120 conference and journal papers, and holds 67 patents as well. His research interests include the areas of computational photography, computational optical sensing, and compressed sensing imaging and vision.