Edge-Based Blur Kernel Estimation Using Sparse Representation and Self-Similarity

Jing Yu · Zhenchun Chang · Chuangbai Xiao

Abstract Blind image deconvolution is the problem of recovering the latent image from the only observed blurry image when the blur kernel is unknown. In this paper, we propose an edge-based blur kernel estimation method for blind motion deconvolution. In our previous work, we incorporate both sparse representation and self-similarity of image patches as priors into our blind deconvolution model to regularize the recovery of the latent image. Since almost any natural image has properties of sparsity and multi-scale self-similarity, we construct a sparsity regularizer and a cross-scale non-local regularizer based on our patch priors. It has been observed that our regularizers often favor sharp images over blurry ones only for image patches of the salient edges and thus we define an edge mask to locate salient edges that we want to apply our regularizers. Experimental results on both simulated and real blurry images demonstrate that our method outperforms existing state-of-the-art blind deblurring methods even for handling of very large blurs, thanks to the use of the edge mask.

Keywords Blind deconvolution · deblurring · sparse representation · self-similarity · cross-scale

CR Subject Classification 10010147 · 10010371 · 10010382 · 10010383

1 Introduction

Motion blur caused by camera shake has been one of the most common artifacts in digital imaging. Blind image deconvolution is an inverse process that attempts to recover the latent (unblurred) image from the observed blurry image when the blur

J. Yu · C. Xiao
Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
E-mail: jing.yu@bjut.edu.cn; cbxiao@bjut.edu.cn

Z. Chang
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

kernel is unknown. In general, for most of the work, the degradation is assumed that the observed image is the output of a linear shift invariant (LSI) system to which noise is added.

If the blur is shift-invariant, it can be modeled as the 2-D convolution of the latent image with the blur kernel:

$$\boldsymbol{y} = \boldsymbol{h} * \boldsymbol{x} + \boldsymbol{v}, \tag{1}$$

where $*$ stands for the 2-D convolution operator, $\boldsymbol{y}$ is the observed blurry image, $\boldsymbol{h}$ is the blur kernel (or point spread function), $\boldsymbol{x}$ is the latent image and $\boldsymbol{v}$ is the additive noise. Then, removing the blur from the observed blurry image becomes a deconvolution operation. When the blur kernel is unknown, the blind deconvolution is a more severely ill-posed inverse problem. The key to the solution of the ill-posed inverse problem is proper incorporation of various image priors about the latent image into the blind deconvolution process. Non-blind image deconvolution seeks an estimate of the latent image assuming the blur is known. In contrast, blind image deconvolution tackles the more difficult, but realistic, problem where the degradation is unknown.

Despite over three decades of research in the field, blind deconvolution still remains a challege for real-world photos with unknown kernels. Recently, blind deconvolution has received renewed attention since Fergus et al.'s work [1] and impressive progress has been made in removing motion blur only given a single blurry image. Some methods explicitly or implicitly exploit edges for kernel estimation [2,3,4,5]. This idea was introduced by Jia [2], who used an alpha matte to estimate the transparency of blurred object boundaries and performed the kernel estimation using transparency. Joshi et al. [3] predict sharp edges using edge profiles and estimate the blur kernel from the predicted edges. However, their goal is to remove small blurs, for it is not trivial to directly restore sharp edges from a severely blurred image. In [4,5], strong edges are predicted from the latent image estimate using a shock filter and gradient thresholding, and then used for kernel estimation. Unfortunately, the shock filter could over-sharpen image edges, and is sensitive to noise, leading to an unstable estimate.

Another family of methods exploit various sparse priors for either the latent image $\boldsymbol{x}$ or the motion blur kernel $\boldsymbol{h}$, and formulate the blind deconvolution as a joint optimization problem with some regularizations on both $\boldsymbol{x}$ and $\boldsymbol{h}$ [1,6,7,8,9,10]:

$$(\hat{\boldsymbol{x}}, \hat{\boldsymbol{h}}) = \arg\min_{\boldsymbol{x},\boldsymbol{h}} \left\{ \sum_* \omega_* \|\partial_* \boldsymbol{y} - \boldsymbol{h} * \partial_* \boldsymbol{x}\|_2^2 + \lambda_x \rho(\boldsymbol{x}) + \lambda_h \rho(\boldsymbol{h}) \right\}, \tag{2}$$

where $\partial_* \in \{\partial_0, \partial_x, \partial_y, \partial_{xx}, \partial_{xy}, \partial_{yy}, \cdots\}$ denotes the partial derivative operator in different directions and orders, ω_* is a weight for each partial derivative, $\rho(\boldsymbol{x})$ is a regularizer on the latent sharp image $\boldsymbol{x}$, $\rho(\boldsymbol{h})$ is a regularizer on the blur kernel $\boldsymbol{h}$, and λ_x and λ_h are regularization weights. The first term in the energy minimization formulation of blind deconvolution uses image derivatives for reducing ringing artifacts. Many techniques based on sparsity priors of image gradients have been proposed to deal with motion blur. Most previous methods assume that gradient magnitudes of natural images follow a heavy-tailed distribution. Fergus et al. [1] represent the heavy-tailed distribution over gradient magnitudes with a zero-mean mixture of Gaussian based on natural image statistics. Levin et al. [11] propose

a hyper-Laplacian prior to fit the heavy-tailed distribution of natural image gradients. Shan et al. [8] construct a natural gradient prior for the latent image by concatenating two piece-wise continuous convex functions. However, sparse gradient priors always prefer the trivial solution, that is, the delta kernel and exactly the blurry image as the latent image estimate because the blur reduces the overall gradient magnitude. To tackle this problem, there are mainly two streams of research works for blind deconvolution. They use the maximum marginal probability estimation of h alone (marginalizing over x) to recover the true kernel [6,7,1] or optimize directly the joint posterior probability of both x and h by performing some empirical strategies or heuristics to avoid the trivial solution during the minimization [8,9,10]. Levin et al. [6,7] suggest that a MAP (maximum a posterior) estimation of h alone is well conditioned and recovers an accurate kernel, while a simultaneous MAP estimation for solving blind deconvolution by jointly optimizing x and h would fail because it favors the trivial solution. Perrone and Favaro [9,10] confirm the analysis of Levin et al. [6,7] and conversely also declare that total variation-based blind deconvolution methods can work well by performing specific implementation. In their work, the total variation regularization weight is initialized with a large value to help avoiding the trivial solution and iteratively reduced to allow for the recovery of more details. Blind deblurring is in general achieved through an alternating optimization scheme. In [9,10], the projected alternating minimization (PAM) algorithm of total variation blind deconvolution can successfully achieve the desired solution.

More present-day works often involve priors over larger neighborhoods or image patches, such as image super resolution [12], image denoising [13], no-blind image deblurring [14] and more. Gradient priors often consider two or three neighboring pixels, which are not sufficient for modeling larger image structures. Patch priors that consider larger neighborhoods (*e.g.*, 5×5 or 7×7 image patches) model more complex structures and dependencies in larger neighborhoods. Image patches are usually overlapped with each other to suppress block effect. Sun et al. [15] use a patch prior learned from an external collection of sharp natural images to restore sharp edges. Michaeli and Irani [16] construct a cross-scale patch recurrence prior for the estimation of the blur kernel. Lai et al. [17] obtain two color centers for every image patch and build a normalized color-line prior for blur kernel estimation. More recently, Pan et al. [18] introduce the dark channel prior based on statistics of image patches to kernel estimation, while Yan et al. [19] propose a patch-based bright channel prior for kernel estimation.

Recent work suggests that image patches can always be well represented sparsely with respect to an appropriate dictionary and the sparsity of image patches over the dictionary can be used as an image prior to regularize the ill-posed inverse problem. Zhang et al. [20] use sparse representation of image patches as a prior for blur kernel estimation and learn an over-complete dictionary from a collection of natural images or the observed blurry image itself using the K-SVD algorithm. Li et al. [21] combine the dictionary pair and the sparse gradient prior with assumption that the blurry image and the sharp image have the same sparse coefficients under the blurry dictionary and the sharp dictionary respectively, to restore the sharp image via sparse reconstruction using the blurry image sparse coefficients on the sharp dictionary. The key issue of sparse representation is to identify a specific dictionary that best represents latent image patches in a sparse manner. Most methods use a database collecting enormous images as training samples to

learn a universal dictionary. To make each patch of the latent image sparsely represented over such a universal dictionary, the database need involve massive training images, and thus this may lead to an inefficient learning and a potentially unstable dictionary. Meanwhile, the database needs to provide patches similar to the patches from the latent image, which cannot hold all the time. Alternatively, the dictionary is trained from the observed blurry image itself. However, the sparsity of the latent sharp image over the learned dictionary cannot be constantly guaranteed.

In this paper, we focus on an edge-based regularization approach for blind motion deblurring using patch priors. In our previous work, sparse representation and self-similarity are combined to work for image super resolution (SR) [12]. Super resolution approaches typically assume that the blur kernel is known (either the point spread function of the camera, or some default low-pass filter, *e.g.* a Gaussian), while blind deblurring refers to the task of estimating the unknown blur kernel. Michaeli and Irani [16] have showed image super resolution approaches cannot be applied directly to blind deblurring. In [22], we have proposed a blur kernel estimation method for blind motion deblurring using sparse representation and self-similarity of image patches as priors to guide the recovery of the latent image. In the previously proposed method, we construct a sparsity regularizer and a cross-scale non-local regularizer based on our priors. This method works quite well for a wide range of blurs but fails to deal with some extremely difficult cases. The edge-based method proposed in this paper is based on the observation that our regularizers often prefer sharp images to blurry ones only for image patches of salient edges. This fundamental observation enable us to build our regularizers on salient edge patches. Finally, we take an approximate iterative approach to solve the optimization problem by alternately updating the blur kernel and the latent image in a coarse-to-fine framework.

The remainder of this paper is organized as follows. Section 2 describes the background on sparse representation and multi-scale self-similarity. Section 3 makes detailed description on the proposed method, including our patch regularizers, our blind deconvolution model and the solution to our model. Section 4 presents experimental results on both simulated and real blurry images. Section 5 draws the conclusion.

2 SPARSE REPRESENTATION AND MULTI-SCALE SELF-SIMILARITY

2.1 Sparse Representation

Image patches can always be represented well as a sparse linear combination of atoms (*i.e.* columns) in an appropriate dictionary. Suppose that the image patch can be represented as $\mathbf{Q}_j \boldsymbol{X}$, here $\mathbf{Q}_j \in \mathbb{R}^{n \times N}$ is a matrix extracting the jth patch from $\boldsymbol{X} \in \mathbb{R}^N$ ordered lexicographically by stacking either the rows or the columns of $\boldsymbol{x}$ into a vector, and the image patch $\mathbf{Q}_j \boldsymbol{X} \in \mathbb{R}^n$ can be represented sparsely over $\mathbf{D} \in \mathbb{R}^{n \times t}$, that is:

$$\mathbf{Q}_j \boldsymbol{X} = \mathbf{D}\boldsymbol{\alpha}_j, \|\boldsymbol{\alpha}_j\|_0 \ll n, \tag{3}$$

where $\mathbf{D} = [\boldsymbol{d}_1, \cdots, \boldsymbol{d}_t] \in \mathbb{R}^{n \times t}$ refers to the dictionary, each column $\boldsymbol{d}_j \in \mathbb{R}^n$ for $j = 1, \cdots, t$ represents the atom of the dictionary $\mathbf{D}$, $\boldsymbol{\alpha}_j = [\alpha_1, \cdots, \alpha_t]^{\mathrm{T}} \in \mathbb{R}^t$ is the sparse representation coefficient of $\mathbf{Q}_j \boldsymbol{X}$ and $\|\boldsymbol{\alpha}_j\|_0$ counts the nonzero entries in $\boldsymbol{\alpha}_j$.

Given a set of training samples $\boldsymbol{s}_i \in \mathbb{R}^n, i = 1, \cdots, m$, here m is the number of training samples, dictionary learning attempts to find a dictionary $\mathbf{D}$ that forms sparse representations $\boldsymbol{\alpha}_i, i = 1, \cdots, m$ for the training samples by jointly optimizing $\mathbf{D}$ and $\boldsymbol{\alpha}_i, i = 1, \cdots, m$ as follows:

$$\min_{\mathbf{D}, \boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_m} \sum_{i=1}^{m} \|\boldsymbol{s}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 \quad \text{s.t. } \forall i \ \|\boldsymbol{\alpha}_i\|_0 \leqslant T, \tag{4}$$

where $T \ll n$ controls the sparsity of $\boldsymbol{\alpha}_i$ for $i = 1, \cdots, m$. The K-SVD method [23] is an effective dictionary learning method which solves Eq.(4) by alternately optimizing $\mathbf{D}$ and $\boldsymbol{\alpha}_i, i = 1, \cdots, m$.

We firstly use the K-SVD method [23] to obtain the dictionary $\mathbf{D}$. Then, we have to derive the sparse coefficient $\boldsymbol{\alpha}_j$ for the patch $\mathbf{Q}_j \boldsymbol{X}$. Eq.(3) can be formulated as the following ℓ_0-norm minimization problem:

$$\min_{\boldsymbol{\alpha}_j} \|\mathbf{Q}_j \boldsymbol{X} - \mathbf{D}\boldsymbol{\alpha}_j\|_2^2 \quad \text{s.t. } \|\boldsymbol{\alpha}_j\|_0 \leqslant T, \tag{5}$$

where T is the sparsity constraint parameter. In our method, we obtain an approximation solution $\hat{\boldsymbol{\alpha}}_j$ for Eq.(5) by using the orthogonal matching pursuit (OMP) method [24].
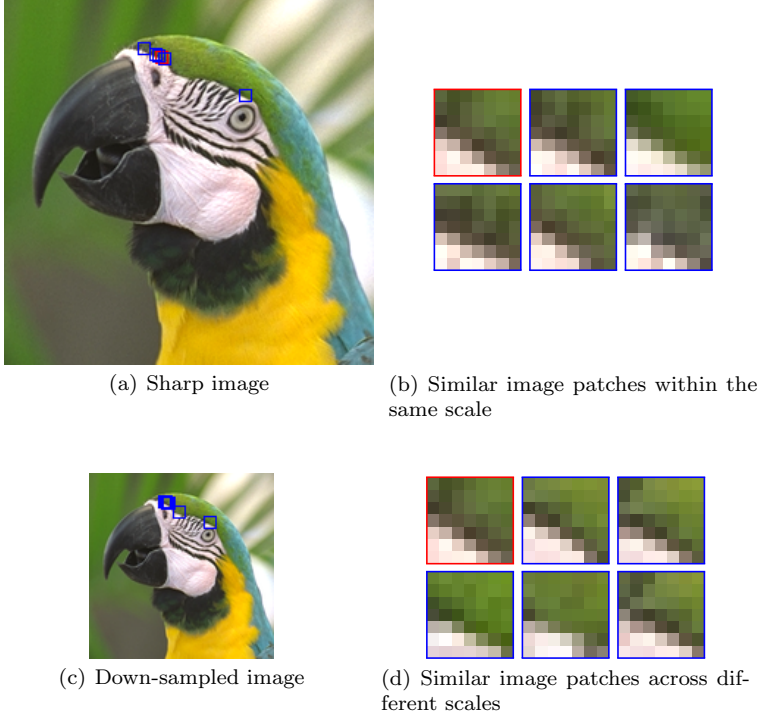
As a matter of fact, the precision of the K-SVD method can be controlled either by constraining the representation error or by constraining the number of nonzero entries in $\boldsymbol{\alpha}_i$. We use the latter formulated in Eq.(4), because it is required in the OMP method [24]. In other words, the objective could be met by constraining the number of nonzero entries in the sparse representation coefficients $\boldsymbol{\alpha}_i$. Once the sparse coefficient $\hat{\boldsymbol{\alpha}}_j$ is derived by solving Eq.(5), the reconstructed image patch $\mathbf{Q}_j \hat{\boldsymbol{X}}$ can be represented sparsely over $\mathbf{D}$ through $\mathbf{Q}_j \hat{\boldsymbol{X}} = \mathbf{D}\hat{\boldsymbol{\alpha}}_j$.

2.2 Multi-Scale Self-Similarity and Non-local Regularization

Most natural images have properties of multi-scale self-similarity: structures from image fragments tend to repeat themselves at the same or different scales in natural images. In particular when small image patches are used, patch repetitions are found abundantly in multiple image scales of almost any natural image, even when we do not visually perceive any obvious repetitive structure. This is due to the fact that very small patches often contain only an edge, a corner, *etc.* [25]. Glasner et al. [25] have showed that almost any image patch in a natural image has multiple similar patches in down-scaled versions of itself.

Fig.1 schematically illustrates patch repetitions of self-similar structures both within the same scale and across different scales of a single image. For a patch of size 7×7 (marked with a red box) in Fig.1(a), we search for its 5 similar patches (marked with blue boxes) in this image. Fig.1(b) shows close-ups of these similar patches within the same scale. In this example, the image is down-sampled by a factor of $a = 2$, as shown in Fig.1(c). For the patch marked with a red box in

Fig.1(a) at the original scale, we also search for its 5 similar patches of the same size in Fig.1(c), marked with blue boxes. Fig.1(d) shows close-ups of these similar patches searched from the down-sampled image, *i.e.* cross-scale similar patches. The patches shown in Fig.1 are displayed with clear repetitive structure in this image.

(a) Sharp image

(b) Similar image patches within the same scale

(c) Down-sampled image

(d) Similar image patches across different scales

Fig. 1 Patch repetitions occur abundantly both within the same scale and across different scales of a single image.

The non-local means was firstly introduced for image denoising based on this self-similarity property of natural images in the seminal work of Buades [26], and since then, the non-local means is extended succesfully to other inverse problems such as image super resolution and non-blind image deblurring [27,28]. The non-local means is based on the observation that similar image patches within the same scale are likely to be appeared in a single image, and these same-scale similar patches can provide additional information. In our blind deconvolution model, we use similar image patches across different scales to construct a cross-scale non-local regularization prior by exploiting the correspondence between these cross-scale similar patches of the same image. Suppose that $\boldsymbol{X} \in \mathbb{R}^N$ and $\boldsymbol{X}^a \in \mathbb{R}^{N/a^2}$ represent the sharp image and its down-scaled version respectively, where N is the size of the sharp image, and a is the down-scaling factor. For each patch $\mathbf{Q}_j \boldsymbol{X}$ in the sharp image $\boldsymbol{X}$, we can search for its similar patches $\mathbf{R}_i \boldsymbol{X}^a$ in $\boldsymbol{X}^a$ that the similarity is measured by the distance between $\mathbf{Q}_j \boldsymbol{X}$ and $\mathbf{R}_i \boldsymbol{X}^a$, here $\mathbf{Q}_j \in \mathbb{R}^{n \times N}$

and $\mathbf{R}_i \in \mathbb{R}^{n \times N/a^2}$ are matrices extracting the jth and the ith patch from $\boldsymbol{X}$ and $\boldsymbol{X}^a$ respectively, and n is the size of the image patch. The linear combination of the L most similar patches of $\mathbf{Q}_j\boldsymbol{X}$ (put into the set $\mathcal{S}_j$) is used to predict $\mathbf{Q}_j\boldsymbol{X}$, that is, the prediction can be represented as the following weighted sum:

$$\mathbf{Q}_j\boldsymbol{X} \approx \sum_{i \in \mathcal{S}_j} w_i^j \mathbf{R}_i \boldsymbol{X}^a, \tag{6}$$

where

$$w_i^j = \frac{\exp(-\|\mathbf{Q}_j\boldsymbol{X} - \mathbf{R}_i\boldsymbol{X}^a\|_2^2/h)}{\sum_{l \in \mathcal{S}_j} \exp(-\|\mathbf{Q}_j\boldsymbol{X} - \mathbf{R}_l\boldsymbol{X}^a\|_2^2/h)} \tag{7}$$

is the weight and h is the control parameter of the weight. It is noted from self-similarity that any patch can, in some way, be approximated by other similar patches of the same image. Obviously the difference between $\mathbf{Q}_j\boldsymbol{X}$ and its prediction should be small and the prediction error can be used as the regularization in our blind deconvolution model (*i.e.* the cross-scale non-local regularizer).

3 Blind Deconvolution

3.1 Use of Cross-Scale Self-Similarity

We incorporate both sparse representation and self-similarity of image patches as priors into our blind deconvolution model to regularize the recovery of the latent image with these priors as regularizers. Since patches repeat across scales in natural images, our patch-based regularizers can depend on abundant patch repetitions across different scales of the same image. Typically we partition the latent image into small overlapping patches. For every patch of the latent image, we search for similar patches of the same size in a down-scaled version of itself. We construct a sparsity regularizer by sparsely representing the latent sharp image over the dictionary that these cross-scale similar patches are used as training samples to learn, denoted by $\mathrm{Reg}_c(\boldsymbol{x})$:

$$\mathrm{Reg}_c(\boldsymbol{x}) = \sum_j \|\mathbf{Q}_j\boldsymbol{X} - \mathbf{D}\boldsymbol{\alpha}_j\|_2^2, \tag{8}$$

and a cross-scale non-local regularizer according to the correspondence between the latent image patch and its similar patches searched from the down-scaled latent image to enforce the recovery of sharp edges, denoted by $\mathrm{Reg}_s(\boldsymbol{x})$:

$$\mathrm{Reg}_s(\boldsymbol{x}) = \sum_j \|\mathbf{Q}_j\boldsymbol{X} - \sum_{i \in \mathcal{S}_j} w_i^j \mathbf{R}_i \boldsymbol{X}^a\|_2^2, \tag{9}$$
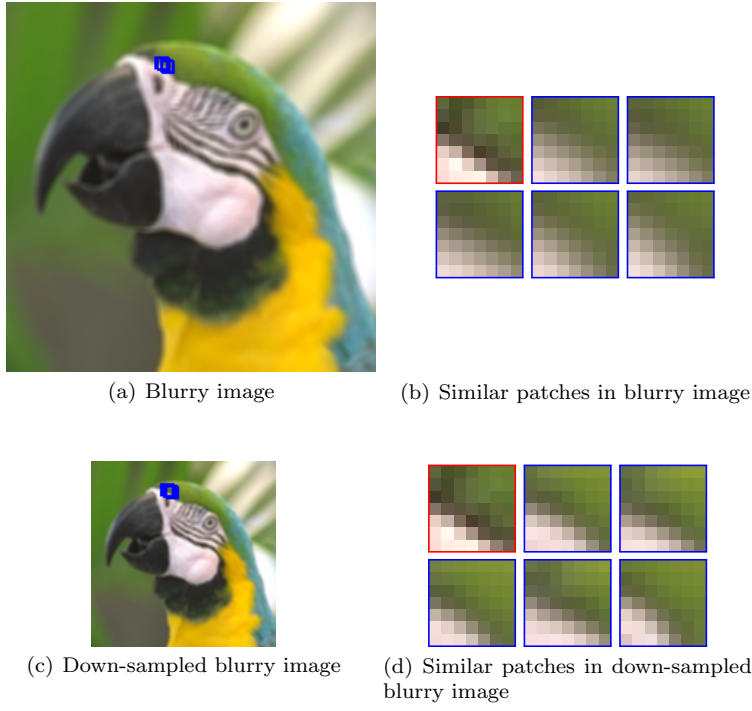
where $\mathbf{D}$ is the learned dictionary for sparse representation, $\boldsymbol{X}$ is the vector-form notion of $\boldsymbol{x}$, $\boldsymbol{X}^a$ is the down-scaled version of $\boldsymbol{X}$ by a factor a, $\mathbf{Q}_j\boldsymbol{X}$ and $\mathbf{R}_i\boldsymbol{X}^a$ represent the jth and the ith patch extracted from the latent image $\boldsymbol{X}$ and its down-scaled version $\boldsymbol{X}^a$ respectively, and $\mathcal{S}_j$ denotes the set of the p most similar patches of $\mathbf{Q}_j\boldsymbol{X}$ searched from $\boldsymbol{X}^a$. We only use similar image patches at down-sampled scales of the latent image to construct the non-local regularizer, without involving those within the same scale into our non-local regularizer.

The choice of training samples is very important for dictionary learning problem. Ideally the dictionary $\mathbf{D}$ should be trained from the patches sampled from the unknown latent sharp image. In our previous single-image super-resolution work [12], the dictionary is trained from the low-resolution image itself. Unforturnately, it is not a good choice for blind deblurring to learn a dictionary using the observed blurry image itself as training samples. This is because the dictionary trained from the blurry image cannot guarantee the sparsity of sharp image patches. In the previously proposed method [22], we used an adaptive over-complete dictionary trained from the down-scaled blurry image, more similar to the latent sharp image than the blurry image itself. In this paper, we present an improvement to collect training samples from the down-scaled latent image estimate, as will be detailed later.
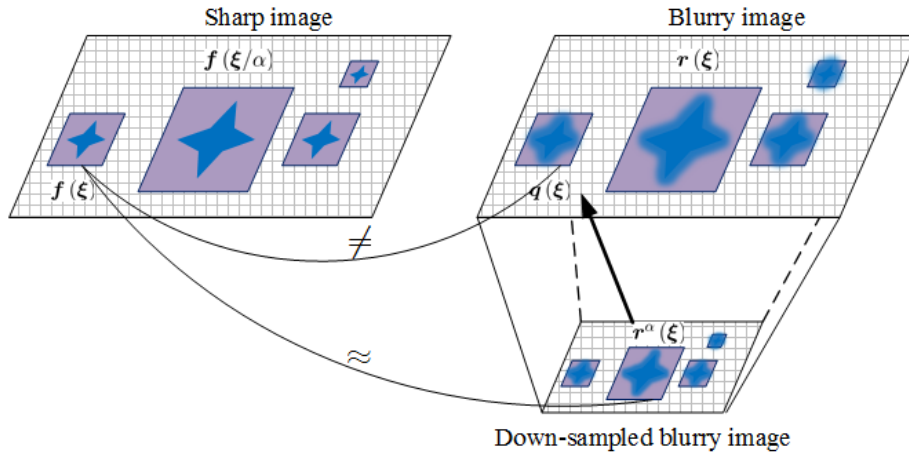
We now provide illustration to account for the use of cross-scale self-similarity. Although patches repeat within and across scales of the sharp image, as illustrated in Fig.1, the similarity diminishes significantly between the sharp image and its blurred counterpart. For the patch marked with a red box from the sharp image shown in Fig.1(a), we still search for its 5 most similar patches from the blurry image (Fig.2(a)) and its down-scaled version (Fig.2(c)) by using block matching, respectively. Fig.2 shows that the patches from the down-scaled blurry image (Fig.2(d)) that are more similar to the patch from the sharp image than the patches from the blurry image itself (Fig.2(b)). This is because the blur effect tends to weaken at coarser scales of the image despite the strong blur at the original scale. It is easy to verify that down-scaling an image by a factor of a produces a-times sharper patches of the same size that are more similar to patches from the latent sharp image. Please refer to [16] for the proof.

Fig.3 illustrates the reason why similar patches across different scales are available for providing a prior for restoration. Suppose that $f(\boldsymbol{\xi})$ and $f(\boldsymbol{\xi}/a)$ are cross-scale similar patches and $f(\boldsymbol{\xi}/a)$ is an a-times larger patch in the sharp image, here $\boldsymbol{\xi}$ denotes the spatial coordinate. Accordingly, their blurry counterparts $q(\boldsymbol{\xi})$ and $r(\boldsymbol{\xi})$ are similar across image scales, and the size of $r(\boldsymbol{\xi})$ is a times as large as that of $q(\boldsymbol{\xi})$ in the blurry image. In Fig.3, the blurry image is a times the size of its down-sampled version. Down-scaling the blurry patch $r(\boldsymbol{\xi})$ by a factor of a generates an a-times smaller patch $r^a(\boldsymbol{\xi})$. Obviously, $q(\boldsymbol{\xi})$ and $r^a(\boldsymbol{\xi})$ are of the same size and the patch $r^a(\boldsymbol{\xi})$ from the down-sampled image is exactly an a-times sharper version of the patch $q(\boldsymbol{\xi})$ in the blurry image. In such a case, $r^a(\boldsymbol{\xi})$ can offer much exact prior information for the recovery of $q(\boldsymbol{\xi})$. Fig.3 schematically demonstrates that the patches at coarser image scales can serve as a good prior, although it is an ideal case.

In summary, we incorporate effectively prior knowledge provided by cross-scale similar patches into our regularizers. As stated above, the down-scaled latent image estimate can provide sharper patches of the same size that are more similar to patches from the latent sharp image. In the sparsity regularizer, the dictionary is trained from sharper patches sampled from the down-scaled latent image estimate to make latent image patches well represented sparsely. In the cross-scale non-local regularizer, meanwhile, all latent image patches are optimized to be as close to their sharper similar patches searched from the down-scaled latent image estimate to enforce the sharp recovery of the latent image as possible.

(a) Blurry image

(b) Similar patches in blurry image

(c) Down-sampled blurry image

(d) Similar patches in down-sampled blurry image

Fig. 2 Down-scaled blurry patches are more similar to the sharp patch than blurry patches at the original scale.

Fig. 3 Similar patches across different scales are available for providing a prior for restoration.

3.2 Analysis on Regularizers

In regularization approaches, blind deconvolution is generally formulated as an energy minimization problem with appropriate regularizers, which tends to be minimal at the desired latent image. The regularizers are used to impose additional constraints on the optimization problem. They significantly benefit the solution of the blind deconvolution problem based on the condition that the regularization functions with respect to the sharp image $\boldsymbol{x}$ should be significantly smaller than those with respect to its blurry counterpart $\boldsymbol{y}$. We will make the sparsity and the self-similarity comparison between the sharp image and the blurry image based on our patch regularizers respectively, and discuss whether the condition is satisfied or which patches satisfy this condition.

3.2.1 Sparsity Regularizer

First of all, we compare the sparsity regularization functions $\mathrm{Reg}_c(\boldsymbol{x})$ and $\mathrm{Reg}_c(\boldsymbol{y})$ with respect to the sharp image $\boldsymbol{x}$ and the blurry one $\boldsymbol{y}$, respectively. For comparison, we generate the blurred image by the convolution of the sharp image shown in Fig.1(a) with the averaging blur kernel. The dictionary is trained from patches sampled from the down-sampled blurry image. We calculate the values of the sparsity regularization functions with respect to the sharp image and several blurred images with blur kernels of varying sizes of 2×2, 3×3 and 5×5, respectively, which are averaged over all pixels, as shown in Table 1, where N is the size of the image, n is the size of image patch. The smaller the value, the smaller the sparse representation error. This means that the image is better represented over the learned dictionary. From Table 1, we can see that the sharp image has larger sparse representation error than any blurred image over the learned dictionary, and the larger blur corresponds to the sparser representation of the blurred image in terms of the entire image.

Table 1 Comparison of sparsity regularizer between sharp image and blurry images with blur kernels of different sizes

	Sharp	2×2 blur	3×3 blur	5×5 blur
$\sqrt{\mathrm{Reg}_c(\cdot)/(N \cdot n)}$	5.40	3.70	2.70	1.72

Note: the intensity range is $[0, 1]$.

Then we compare the sparsity regularization functions with respect to the sharp image and the blurred counterpart on a patch-by-patch basis. Let $\mathcal{R}_c$ represent the set of pixels at which the sharp patch has smaller sparse representation error than the blurred one over the learned dictionary. That is,

$$\mathcal{R}_c = \{j | \ \|\mathbf{Q}_j \boldsymbol{X} - \mathbf{D}\boldsymbol{\alpha}_j\|_2^2 \leqslant \|\mathbf{Q}_j \boldsymbol{Y} - \mathbf{D}\boldsymbol{\alpha}_j\|_2^2\}, \tag{10}$$

where $\boldsymbol{X}$ and $\boldsymbol{Y}$ denote the vector notations of the sharp image $\boldsymbol{x}$ and the blurred image $\boldsymbol{y}$ respectively. Fig.4(a) shows the blurred image with the averaging blur kernel of size 2×2. In Fig.4(b), the set $\mathcal{R}_c$ are indicated with white pixels where the sharp patch achieves smaller sparse representation error than the blurred patch

over the learned dictionary. From Fig.4(b), we can see that the sparsity regularizer of the sharp image is smaller than that of the blurred image only for some certain patches. Intuitively, these regions comprised of white pixels coincide with edges and sharp changes in this image. It is believed that most image structures are often reflected around edges and areas of high variation. The optimal dictionary should produce sparsest representation of edge patches in the latent sharp image.

3.2.2 Non-local Regularizer

For the same reason, we compare the non-local regularization functions $\text{Reg}_s(\boldsymbol{x})$ and $\text{Reg}_s(\boldsymbol{y})$ with respect to the sharp image $\boldsymbol{x}$ and the blurry one $\boldsymbol{y}$, respectively. Similarly, we calculate the values of the non-local regularization functions with respect to the sharp image and the blurred images with blur kernels of varying sizes of 2×2, 3×3 and 5×5 respectively, averaged over all pixels, as shown in Table 2. The smaller the value, the smaller the prediction error. It means that there is stronger cross-scale self-similarity throughout the image. From Table 2, we can see that the sharp image reveals the weakest cross-scale self-similarity, and the blurred image with larger blur kernel displays stronger cross-scale self-similarity in terms of the entire image.

Table 2 Comparison of cross-scale non-local regularizer between sharp image and blurry images with blur kernels of different sizes

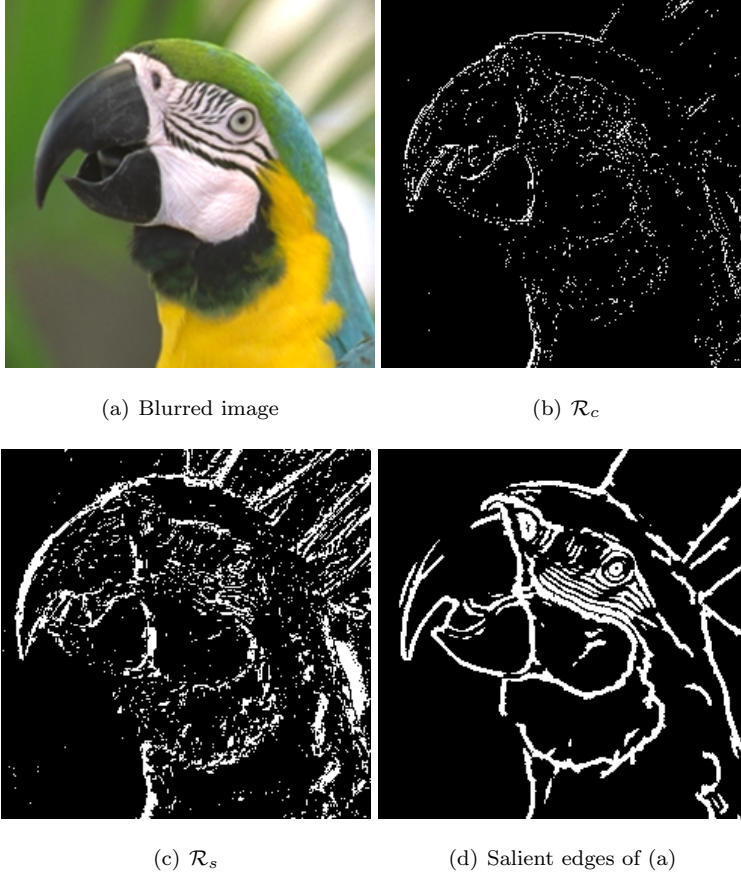|	Sharp	2×2 blur	3×3 blur	5×5 blur
$\sqrt{\text{Reg}_s(\cdot)/(N \cdot n)}$	0.0448	0.0385	0.0339	0.0271

Note: the intensity range is $[0, 1]$.

We still compare the non-local regularization functions with respect to the sharp image and the blurred counterpart on a patch-by-patch basis. Let $\mathcal{R}_s$ represent the set of pixels at which the sharp patch has smaller prediction error than the blurred one. That is,

$$\mathcal{R}_s = \{j|\ \|\mathbf{Q}_j \boldsymbol{X} - \sum_{i \in \mathcal{S}_j} w_i^j \mathbf{R}_i \boldsymbol{X}^a\|_2^2 \leqslant \|\mathbf{Q}_j \boldsymbol{Y} - \sum_{i \in \mathcal{S}_j} w_i^j \mathbf{R}_i \boldsymbol{Y}^a\|_2^2\}, \qquad (11)$$

where $\boldsymbol{Y}$ and $\boldsymbol{Y}^a$ denote the vector notation of the blurred image $\boldsymbol{y}$ and its downsampled version by a factor of a. From Fig.4(c), the set $\mathcal{R}_s$ indicated with white pixels is also roughly consistent with edges of the image. Our further observation shows that image edges do not always help kernel estimation when the scale of the edge is smaller than that of the blur kernel, while salient edges can effectively avoid the trivial solution and get an accurate blur kernel. We use Sun et al's strategy [15] (see the following edge mask $\boldsymbol{M}$ for more details) to detect and select salient edges of the blurred image, as is shown in Fig.4(d).

It can be observed from the comparison of Figs. 4(c) and (d) that the cross-scale non-local regularizer of the sharp image is smaller than that of the blurred image roughly around salient edges. The blur alters to different extent edges of repetitive structures across different scales and thus deteriorates cross-scale self-similarity properties of edge structures in the blurry image.

(a) Blurred image   (b) $\mathcal{R}_c$

(c) $\mathcal{R}_s$ (d) Salient edges of (a)

Fig. 4 Sharp image has stronger sparsity and cross-scale self-similarity than blurred image roughly around salient edges.

3.3 Modeling and Optimization

Although natural images generally have properties of sparsity and self-similarity, in the previous part, we have made detailed discussions on our two regularizers $\mathrm{Reg}_c(\boldsymbol{x})$ and $\mathrm{Reg}_s(\boldsymbol{x})$, and come to the conclusion that $\mathrm{Reg}_c(\boldsymbol{x}) < \mathrm{Reg}_c(\boldsymbol{y})$ and $\mathrm{Reg}_s(\boldsymbol{x}) < \mathrm{Reg}_s(\boldsymbol{y})$ are often satisfied only for image patches of salient edges. In other words, they only favor the sharp solution over the blurred one around salient image edges. In order to generate more exact solutions, our regularization constraints are only imposed on image patches of salient edges.

In this paper, we define the edge mask $\boldsymbol{M}$ according to the corresponding salient edge pixels, which is a binary mask indicating pixel locations that we want to apply our priors. We employ a heuristic process to detect and select salient edges of the latent image estimate during the optimization in a coarse-to-fine framework for kernel estimation and thus we do not present a joint energy minimization formulation of both the latent image $\boldsymbol{x}$ and the blur kernel $\boldsymbol{h}$. In each level of

the image pyramid, we take an approximate approach to solve the optimization problem by directly alternating between optimizing the kernel $\boldsymbol{h}$ and the latent image $\boldsymbol{x}$.

1. Updating M

This step chooses pixel locations to apply our patch priors. Since our regularizers prefer the sharp image to the blurry one only around salient edges, in order to benefit the blur kernel estimation, we first detect and select useful salient edges. We adopt Sun et al.'s strategy [15] to filter the latent image estimate $\hat{\boldsymbol{x}}_k$ with a filter bank consisting of derivatives of Gaussians in eight directions and obtain the edge mask $\boldsymbol{M}$ by keeping the top 2% of pixel locations from the largest filter responses of the filter bank. In our model, regions outside the mask are weakly regularized by our patch priors, resulting in noise amplification in flat or smooth regions, and therefore the Gaussian low-pass filter are utilized before salient edge selection.

2. Updating h

In this step, we fix $\hat{\boldsymbol{x}}_k$ and update $\hat{\boldsymbol{h}}_{k+1}$. The minimization problem is defined with a Gaussian regularizer as:

$$\hat{\boldsymbol{h}}_{k+1} = \arg\min_{\boldsymbol{h}} \left\{ \|\nabla \boldsymbol{y} - \boldsymbol{h} * (\nabla \hat{\boldsymbol{x}}_k \odot \boldsymbol{M})\|_2^2 + \lambda_h \|\boldsymbol{h}\|_2^2 \right\}, \qquad (12)$$

where $\nabla = \{\partial_x, \partial_y\}$ denotes the spatial derivative operator in two directions, $\odot$ stands for the pixel-wise multiplication, and λ_h is the regularization weight to control the tradeoff between the fidelity to the observation model (as accounted for by the former term) and the smoothness of the estimated blur kernel (as reflected by the latter term). We multiply $\nabla \hat{\boldsymbol{x}}_k$ by the mask $\boldsymbol{M}$ (*i.e.* $\nabla \hat{\boldsymbol{x}}_k \odot \mathbf{M}$) to enforce that regions outside the mask do not participate in estimating $\boldsymbol{h}$. We only allow salient edges in the mask $\boldsymbol{M}$ to participate in the constraint of the observation model by setting the gradient $\nabla \hat{\boldsymbol{x}}_k$ outside $\boldsymbol{M}$ to zero.

On the other hand, we take a common way to eliminate the influence of smooth or flat regions of the image on kernel estimation [4,5,15,17]. The pixels whose gradient magnitudes are less than a certain threshold in the intermediate latent image estimate are set to zero. Let τ denote a threshold of the gradient magnitude and N_h denote the size of the blur kernel. The threshold for truncating gradients is determined as follows. We construct the histograms of gradient magnitudes and directions for each $\partial_* \hat{\boldsymbol{x}}_k$. Angles are quantized by $45°$, and gradients of opposite directions are counted together. Then, we find a threshold that keeps at least $r\sqrt{N_h}$ pixels from the largest magnitude for each quantized angle. We use 2 for r by default. To allow for inferring subtle structures during kernel refinement, we gradually decrease the value of the threshold τ in iterations by dividing by 1.1 at each iteration, to include more and more edges. Eq.(12) excludes part of the gradients, depending jointly on the magnitude and the edge mask $\boldsymbol{M}$. In order to suppress the noise in flat or smooth regions, however, we do nothing on $\nabla \boldsymbol{y}$. This selection process reduces ambiguity in the following kernel estimation.

Eq.(12) is a quadratic funciton of unknown $\boldsymbol{h}$, which has a closed-form solution for $\hat{\boldsymbol{h}}_{k+1}$. We solve Eq.(12) in the Fourier domain by performing FFTs on all variables and setting the derivative with respect to $\boldsymbol{h}$ to zero:

$$\hat{\boldsymbol{h}}_{k+1} = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(\partial_x \hat{\boldsymbol{x}}_k \odot \boldsymbol{M})}\mathcal{F}(\partial_x \boldsymbol{y}) + \overline{\mathcal{F}(\partial_y \hat{\boldsymbol{x}}_k \odot \boldsymbol{M})}\mathcal{F}(\partial_y \boldsymbol{y})}{\mathcal{F}(\partial_x \hat{\boldsymbol{x}}_k \odot \boldsymbol{M})^2 + \mathcal{F}(\partial_y \hat{\boldsymbol{x}}_k \odot \boldsymbol{M})^2 + \lambda_h} \right), \qquad (13)$$

where $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denote the fast Fourier transform and inverse Fourier transform respectively, and $\overline{\mathcal{F}(\cdot)}$ is the complex conjugate operator.

3. Updating x

In this step, we fix $\hat{h}_{k+1}$, and given $\hat{x}_k$ update $\hat{x}_{k+1}$. With our patch priors as regularizers, we establish our regularizers on salient edge patches of the image, and get the following regularized minimization:

$$
\begin{aligned}
\hat{x}_{k+1} = \arg\min_{x} \Big\{ & \|\nabla y - \hat{h}_{k+1} * \nabla x\|_2^2 + \lambda_c \frac{N}{|M|} \sum_{j \in M} \|\mathbf{Q}_j X - \mathbf{D}\alpha_j\|_2^2 \\
& + \lambda_s \frac{N}{|M|} \sum_{j \in M} \|\mathbf{Q}_j X - \sum_{i \in \mathcal{S}_j} w_i^j \mathbf{R}_i X^a\|_2^2 + \lambda_g \|\nabla x\|_2^2 \Big\} \\
& \text{s.t. } \forall j \ \|\alpha_j\|_0 \leqslant T
\end{aligned}
\tag{14}
$$

where $|M|$ is the number of non-zero elements in the mask M, and N is the size of the latent image, $\mathbf{D}$ is the dictionary trained from the down-scaled latent image estimate, X is the vector notation of the latent image x, X^a is the down-sampled version of X by a factor of a, and λ_c, λ_s, and λ_g are regularization weights controlling the effect of the regularizers. In Eq.(14), the first term is the fidelity to the observation model, the second term is the sparsity regularizer, the third term is the cross-scale non-local regularizer, and the last term is the smoothness constraint of the estimated latent image.

Rearranging y in vector form, denoted by $Y \in \mathbb{R}^N$, and rewriting the convolution of the blur kernel and the latent image in matrix-vector form, Eq.(14) can be rewritten as

$$
\begin{aligned}
\hat{X}_{k+1} = \arg\min_{X} \Big\{ & \|\mathbf{G}_x Y - \mathbf{H}_{k+1}\mathbf{G}_x X\|_2^2 + \|\mathbf{G}_y Y - \mathbf{H}_{k+1}\mathbf{G}_y X\|_2^2 \\
& + \lambda_c \frac{N}{|M|} \sum_{j \in M} \|\mathbf{Q}_j X - \mathbf{D}\alpha_j\|_2^2 + \lambda_s \frac{N}{|M|} \sum_{j \in M} \|\mathbf{Q}_j X - \sum_{i \in \mathcal{S}_j} w_i^j \mathbf{R}_i X^a\|_2^2 \\
& + \lambda_g (\|\mathbf{G}_x X\|_2^2 + \|\mathbf{G}_y X\|_2^2) \Big\} \\
& \text{s.t. } \forall j \ \|\alpha_j\|_0 \leqslant T
\end{aligned}
\tag{15}
$$

where $\mathbf{G}_x$ and $\mathbf{G}_y \in \mathbb{R}^{N \times N}$ are the matrix forms of the partial derivative operators ∂_x and ∂_y in two directions respectively, and $\mathbf{H}_{k+1} \in \mathbb{R}^{N \times N}$ is the blur matrix. Setting the derivative of Eq.(15) with respect to X to zero and letting $\mathbf{G} = \mathbf{G}_x^{\mathrm{T}}\mathbf{G}_x + \mathbf{G}_y^{\mathrm{T}}\mathbf{G}_y$, we derive

$$
\begin{aligned}
& \big[(\mathbf{H}_{k+1}^{\mathrm{T}}\mathbf{H}_{k+1} + \lambda_g)\mathbf{G} + (\lambda_c + \lambda_s)\tfrac{N}{|M|} \sum_{j \in M} \mathbf{Q}_j^{\mathrm{T}}\mathbf{Q}_j \big]\hat{X}_{k+1} = \\
& \mathbf{H}_{k+1}^{\mathrm{T}}\mathbf{G}Y + \lambda_c \tfrac{N}{|M|} \sum_{j \in M} \mathbf{Q}_j^{\mathrm{T}}\mathbf{D}\alpha_j + \lambda_s \tfrac{N}{|M|} \sum_{j \in M} \mathbf{Q}_j^{\mathrm{T}} \sum_{i \in \mathcal{S}_j} w_i^j \mathbf{R}_i \hat{X}_{k+1}^a
\end{aligned}
\tag{16}
$$

Since both sparse representation coefficients α_j and the down-sampled image $\hat{X}_{k+1}^a$ on the right-hand side of Eq.(16) depend on unknown $\hat{X}_{k+1}$, Eq.(16) cannot be solved in closed form. Instead we approximately solve Eq.(16) with the following procedure:

(1) The K-SVD method [23] is used to attain the dictionary $\mathbf{D}$ by approximately solving Eq.(4). For each patch $\mathbf{Q}_j \hat{X}_k$ in $\hat{X}_k$ that the mask M selects, the

OMP method [24] is used here to derive the sparse representation coefficient $\boldsymbol{\alpha}_j$ over the dictionary $\mathbf{D}$ by approximately solving the following constrained minimization problem:

$$\hat{\boldsymbol{\alpha}}_j = \arg\min_{\boldsymbol{\alpha}_j} \|\mathbf{Q}_j \hat{\boldsymbol{X}}_k - \mathbf{D}\boldsymbol{\alpha}_j\|_2^2 \quad \text{s.t.} \ \|\boldsymbol{\alpha}_j\|_0 \leqslant T. \tag{17}$$

Since the sparse coefficient $\boldsymbol{\alpha}_j$ on the right-hand side of Eq.(16) depends on unknown $\hat{\boldsymbol{X}}_{k+1}$, we approximate $\hat{\boldsymbol{X}}_{k+1}$ using $\hat{\boldsymbol{X}}_k$ to solve the sparse coefficient $\hat{\boldsymbol{\alpha}}_j$ over the dictionary $\mathbf{D}$.

(2) For the same reason, since $\hat{\boldsymbol{X}}_{k+1}$ and its down-scaled $\hat{\boldsymbol{X}}_{k+1}^a$ are both unknown, we approximate $\hat{\boldsymbol{X}}_{k+1}$ and $\hat{\boldsymbol{X}}_{k+1}^a$ using $\hat{\boldsymbol{X}}_k$ and $\hat{\boldsymbol{X}}_k^a$ respectively. For each patch $\mathbf{Q}_j \hat{\boldsymbol{X}}_k$ in $\hat{\boldsymbol{X}}_k$ that the mask $\boldsymbol{M}$ selects, we search for its similar patches $\mathbf{R}_i \hat{\boldsymbol{X}}_k^a, i \in \hat{\mathcal{S}}_j$ in the down-scaled image $\hat{\boldsymbol{X}}_k^a$ of $\hat{\boldsymbol{X}}_k$, and use the linear combination of these similar patches $\sum_{i \in \hat{\mathcal{S}}_j} \hat{w}_i^j \mathbf{R}_i \hat{\boldsymbol{X}}_k^a$ to predict it. Here $\hat{\mathcal{S}}_j$ and $\hat{w}_i^j$ are updated according to $\hat{\boldsymbol{X}}_k$ and $\hat{\boldsymbol{X}}_k^a$.

(3) Eq.(16) can be reformulated by substituting the sparse coefficient $\hat{\boldsymbol{\alpha}}_j$, the set of similar patches $\hat{\mathcal{S}}_j$ and the weights $\hat{w}_i^j$ derived from the above approximations into the right-hand side of Eq.(16), such that:

$$\begin{aligned}
&\left[(\mathbf{H}_{k+1}^{\mathrm{T}}\mathbf{H}_{k+1} + \lambda_g)\mathbf{G} + (\lambda_c + \lambda_s)\tfrac{N}{|\boldsymbol{M}|} \sum_{j \in \boldsymbol{M}} \mathbf{Q}_j^{\mathrm{T}}\mathbf{Q}_j\right]\hat{\boldsymbol{X}}_{k+1} = \\
&\mathbf{H}_{k+1}^{\mathrm{T}}\mathbf{G}\boldsymbol{Y} + \lambda_c \tfrac{N}{|\boldsymbol{M}|} \sum_{j \in \boldsymbol{M}} \mathbf{Q}_j^{\mathrm{T}}\mathbf{D}\hat{\boldsymbol{\alpha}}_j + \lambda_s \tfrac{N}{|\boldsymbol{M}|} \sum_{j \in \boldsymbol{M}} \mathbf{Q}_j^{\mathrm{T}} \sum_{i \in \hat{\mathcal{S}}_j} \hat{w}_i^j \mathbf{R}_i \hat{\boldsymbol{X}}_k^a.
\end{aligned} \tag{18}$$

Since it is a linear equation with respect to $\hat{\boldsymbol{X}}_{k+1}$, Eq.(18) can be solved by direct matrix inversion or the conjugate gradient method. In our method, $\hat{\boldsymbol{X}}_{k+1}$ are updated by solving it using the bi-conjugate gradient (BICG) method.

4. Repeat steps 1-3 until convergence or for a fixed number of iterations.

3.4 Implementation

To speed up the convergence and handle of large blurs, following most existing methods, we estimate the blur kernel in a coarse-to-fine framework. We apply our alternating iterative minimization procedure described in Section 3.3 to each of the levels of the image pyramid constructed from the blurred image $\boldsymbol{y}$. The blur kernel refinement starts from the coarsest level and works down to the finest level with the original image resolution. At the coarsest level, the latent image estimate is initialized with the observed blurry image. The intermediate latent image estimated at each coarser level is interpolated and then propagated to the next finer level as an initial estimate of the latent image to refine the blur kernel estimate in higher resolutions.

Different from [22], in which the dictionary is trained from patches randomly sampled from the down-scaled blurry image, in this paper, the dictionary is trained from edge patches sampled directly from the intermediate latent image estimated at the coarser scale, and iteratively updated once for each image scale during the solution. We do not pay attention to the sparsity of the entire image over the learned dictionary, but only the sparsity of edge patches in the image, for

our sparsity regularizer prefers the sharp image to the blurred one only for edge patches.

Blind deconvolution in general involves two stages. The motion blur kernel $\boldsymbol{h}$ is firstly estimated by alternately updating the motion blur kernel $\boldsymbol{h}$ and the latent image $\boldsymbol{x}$. The intermediate latent images estimated during the iterations have no direct influence on the final deblurring result, and only affect this result indirectly by contributing to the refinement of the blur kernel estimate $\hat{\boldsymbol{h}}$. Then, the final deblurring result $\hat{\boldsymbol{x}}$ is recovered from the given blurry image $\boldsymbol{y}$ with the estimated blur kernel $\hat{\boldsymbol{h}}$ for the finest level by performing a variaty of non-blind deconvolution methods, such as fast TV-ℓ_1 deconvolution [5], sparse deconvolution [6] and EPLL [29] *etc.*.

We estimate the blur kernel $\boldsymbol{h}$ by the implementation of the pseudo-code outlined in Algorithm 1. We construct an image pyramid with L levels from the given blurry image $\boldsymbol{y}$. The number of pyramid levels is chosen such that, at the coarsest level, the size of the blur is smaller than that of the patch used in the blur kernel estimation stage. Let us use the notation $\hat{\boldsymbol{x}}_k^l$ for the intermediate latent image estimate, where the superscript l indicates the lth level in the image pyramid, while the subscript k indicates the kth iteration at each scale level. The iterative procedure starts from the coarsest level $l = 1$ of the image pyramid initialized with $\hat{\boldsymbol{x}}_0^1 = \boldsymbol{y}$. At each scale level $l \in \{1, \cdots, L\}$, we take the iterative procedure that alternately optimizes the motion blur kernel $\boldsymbol{h}$ and the latent image $\boldsymbol{x}$ as detailed in Section 3.3, which is implemented repeatedly until the convergence or for a fixed number of iterations. Then the outcome of updating the latent image at the lth level is upsampled by interpolation and then used as an initial estimate of the latent image for the next finer level $l + 1$ to progressively refine the motion blur kernel estimate $\hat{\boldsymbol{h}}$, which is repeated to achieve the final refinement of the blur kernel estimate $\hat{\boldsymbol{h}}$ for the finest level.

Algorithm 1: Edge-Based Blur Kernel Estimation Using Sparse Representation and Self-Similarity

---

Input: Blurry image $\boldsymbol{y}$
Output: Blur kernel estimate $\hat{\boldsymbol{h}}$
Set down-scaling factor a, regularization weights λ_g, λ_c, λ_s, λ_h, size of patch n, size of dictionary t, sparsity constraint parameter T, number of similar patches p, convergence tolerance ϵ and maximum allowed number of iterations maxIters;
Build an image pyramid with L levels;
Initialize $\hat{\boldsymbol{x}}_0^1 = \boldsymbol{y}$;
Train dictionary $\mathbf{D}$ using $\hat{\boldsymbol{x}}_0^1$;
Outer loop: for $l = 1$ *to* $l = L$ **do**   `// for each level of image pyramid`
    Initialize $k = 0$, gradient threshold τ;
    Inner loop: repeat   `// for each iteration`
        Predict the edge mask $\boldsymbol{M}$;
        Compute blur kernel $\hat{\boldsymbol{h}}_{k+1}^l$ using Eq.(13);
        Given $\hat{\boldsymbol{x}}_k^l$, update latent image $\hat{\boldsymbol{x}}_{k+1}^l$ by solving Eq. (18) using BICG;
        $\tau = \tau/1.1$; $k = k + 1$
    until $k > $ maxIters *or* $\|\hat{\boldsymbol{x}}_k^l - \hat{\boldsymbol{x}}_{k-1}^l\|_2^2 \leqslant \epsilon$;
    Update dictionary $\mathbf{D}$ using $\hat{\boldsymbol{x}}_k^l$;
    Upscale image $\hat{\boldsymbol{x}}_k^l$ to initialize $\hat{\boldsymbol{x}}_0^{l+1}$ for the next finer level;
$\hat{\boldsymbol{h}} = \hat{\boldsymbol{h}}_k^L$; $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}_k^L$.

In the blur kernel estimation process, we use the gray-scale versions of the blurry image y and the intermediate latent image estimate $\hat{x}$. Once the blur kernel estimate $\hat{h}$ has been obtained with the original image scale, we perform the final non-blind deconvolution with $\hat{h}$ on each color channel of y to obtain the deblurring result.

Finally, our method need perform deconvolution in the Fourier domain. To avoid ringing artifacts at the image boundaries, we process the image near the boundaries using the simple *edgetaper* command in Matlab.

4 EXPERIMENTS

Several experiments are conducted to demonstrate the performance of our method. We first test our method on the widely used datasets introduced in [6] and [15], and make qualitative and quantitative comparisons with the state-of-the-art blind deblurring methods. Then we show visual comparisons on real blurry photographs with unknown blurs. The relevant parameters of our method are set as follows: the dictionary $\mathbf{D}$ is of size $t = 100$, and the sparsity constraint parameter $T = 4$, designed to handle image patches of size $n = 5 \times 5$, the number of iterations is fixed as 14 for the inner loop, and the regularization weights are empirically set to $\lambda_c = 0.04/n$, $\lambda_s = 0.04/n$, $\lambda_g = 0.003$ and $\lambda_h = 0.0003N$. As the down-scaling factor increases, image patches become sharper, but there exist less similar patches at the down-sampled scale. Following the setting of [16], the image pyramid is constructed with scale-gaps of $a = 4/3$ using down-scaling with a sinc function. Additional speed up is obtained by using the fast approximate nearest neighbor (NN) search of [30] in the blur kernel estimation stage, working with a single NN for every patch.

An additional important parameter is the size of the blur kernel. Small blurs are hard to solve if it is initialized with a very large kernel. Conversely, large blurs will be truncated if too small a kernel is used [1]. Following the setting of [15], we do not assume that the size of the kernel is known and initialize that the size of the kernel is 51×51 in most cases except for some extremely difficult cases. Experiment results on both simulated and real blurry images show the size of the blur kernel is generally not larger than 51×51 for most blurry images. Even though the input blurry image has a small blur kernel, our method is still able to obtain a good deblurring result, less sensitive to the initial setting of the kernel size.

4.1 Quantitative Evaluation with Reference to Ground Truth

We test our method on two publicly available datasets. One dataset, which is provided by Levin et al. [6], contains 32 images of size 255×255 blurred by real camera shake. The blurred images with spatially invariant blur and 8 different ground-truth kernels were captured simultaneously by locking the Z-axis rotation handle but loosening the X and Y handles of the tripod. The kernels range in size from 13×13 to 27×27. The other dataset provided by Sun et al. [15] comprises 640 natural images of diverse scenes, which were obtained by synthetically blurring 80 high-resolution images with the 8 blur kernels from [6] and adding 1% white Gaussian noise to the blurred images. We present qualitative and quantitative

comparisons with the state-of-the-art blind deblurring methods [1,4,5,7,9,16,15, 31,32,33].

We measure the quality of the blur kernel estimate $\hat{\boldsymbol{h}}$ using the error ratio measure ER [16]:

$$\text{ER} = \frac{\|\boldsymbol{x} - \hat{\boldsymbol{x}}_{\hat{\boldsymbol{h}}}\|_2^2}{\|\boldsymbol{x} - \hat{\boldsymbol{x}}_{\boldsymbol{h}}\|_2^2}, \tag{19}$$

where $\hat{\boldsymbol{x}}_{\hat{\boldsymbol{h}}}$ corresponds to the deblurring result with the recovered kernel $\hat{\boldsymbol{h}}$, and $\hat{\boldsymbol{x}}_{\boldsymbol{h}}$ corresponds to the deblurring result with the ground-truth kernel $\boldsymbol{h}$. The smaller ER corresponds to the better quality. In principle, if ER = 1, the recovered kernel yields a deblurring result as good as the ground-truth kernel.

On the dataset provided by Levin et al. [6], we compare our error ratios with those of Fergus et al. [1], Cho and Lee [4], Xu and Jia [5], Perrone and Favaro [9], Levin et al. [7], Perrone et al. [33] and our previous method [22]. Fig.5 shows the cumulative distribution of the error ratio of our method compared with the other methods over the dataset of [6]. Levin et al. [7] use sparse deconvolution [6] to generate the final results, and observe that deconvolution results are usually visually plausible when their error ratios are below 3. Therefore, we standardize the final non-blind deconvolution by using sparse deconvolution [6] to obtain the results, for fair comparison. Table 3 lists the success rate and the average error ratio over 32 images for each method. The success rate is the percentage of images which achieve good deblurring results, that is, the percentage of images that have an error ratio below a certain threshold. On this dataset, the success rate is the percentage of the results under the error ratio of 3. Table 3 shows our method takes the lead with a success rate of 100%, a higher success rate than our previous method without considering salient edges [22]. Levin et al. [7], Perrone and Favaro [9] and Perrone et al. [33] initialize the size of the blur kernel with ground truth, while the size of the blur kernel is unknown for real scenes. Even so, our method still achieves a much higher success rate than the other methods over the dataset of [6].

Table 3 Quantitative comparison of different methods over the dataset of [6]

	Success rate%	Mean error ratio
Ours	100	1.4433
Yu et al. [22]	96.88	1.4653
Perrone et al. [33]	93.75	1.2024
Xu & Jia [5]	93.75	2.1365
Perrone & Favaro [9]	87.50	2.0263
Levin et al. [7]	87.50	2.0583
Fergus et al. [1]	75.00	13.5268
Cho & Lee [4]	68.75	2.6688

On this dataset provided by Sun et al. [15], we compare our error ratios with those of Cho and Lee [4], Xu and Jia [5], Levin et al. [7], Sun et al. [15], Michaeli and Irani [16], Cho et al. [31], Krishnan et al. [32] and our previous method [22].
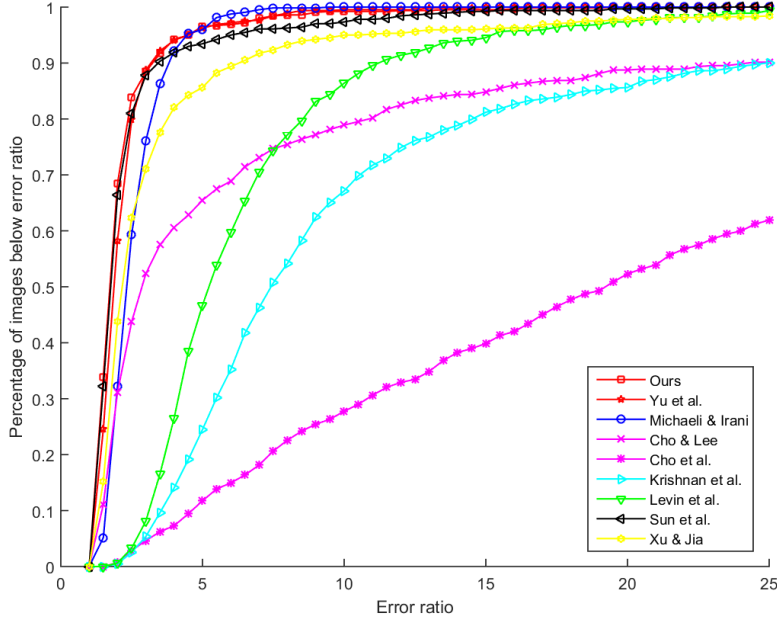
Fig. 5 Cumulative distributions of error ratios with different methods on the dataset of [6]

Fig.6 shows the cumulative distribution of error ratios over the entire dataset for each method. We apply the blur kernel estimated by each method to perform deblurring with the non-blind deblurring method of [29] to recover latent images. It is empirically observed by Michaeli and Irani [16] that the deblurring results are still visually acceptable for error ratios ER $\leqslant$ 5, when using the non-blind deconvolution of [29]. Table 4 lists the success rate (*i.e.*, an error ratio below 5) and the average error ratio over 640 images with different methods. Table 4 shows our method achieves the highest success rate and the lowest average error ratio followed by Michaeli and Irani [16] and Sun et al. [15]. Moreover, these two methods by Michaeli and Irani [16] and Sun et al. [15] take 9213 and 4899 seconds on average to process an image of size 1024×800 from this dataset respectively, and our method take 1823 seconds, much faster than their methods.

Figs.7 and 8 show qualitative comparisons of cropped results on two blurred images from the synthetic dataset of [15] by different methods. Compared with the other methods, our method usually obtains more accurate blur kernels, suffers from fewer ringing artifacts and restores more and sharper image details.

4.2 Qualitative Comparison on Real Images

We also experiment with real blurry images which are blurred with unknown kernels. In this part, we process blurry images with very large blurs to demonstrate the robustness of our method. We recover the latent image from the observed
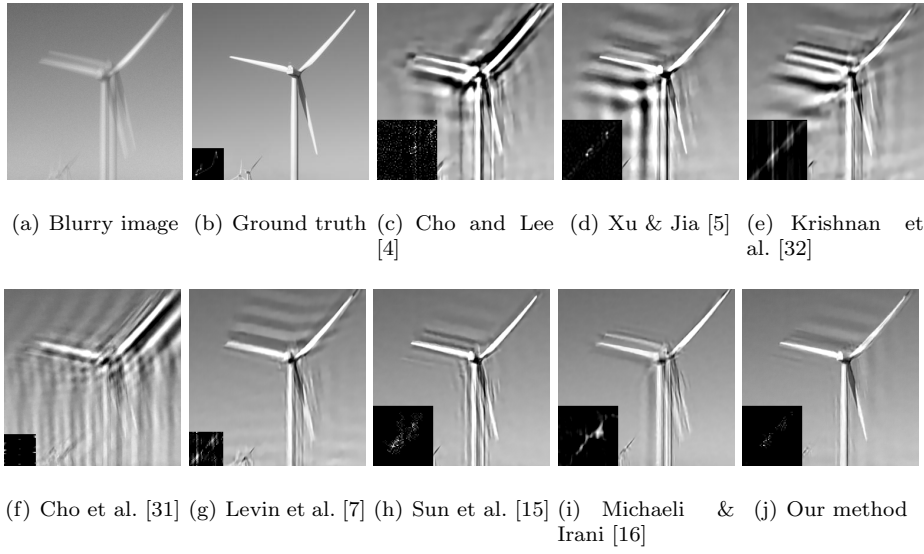
Fig. 6 Cumulative distributions of error ratios with different methods on the dataset of [15]
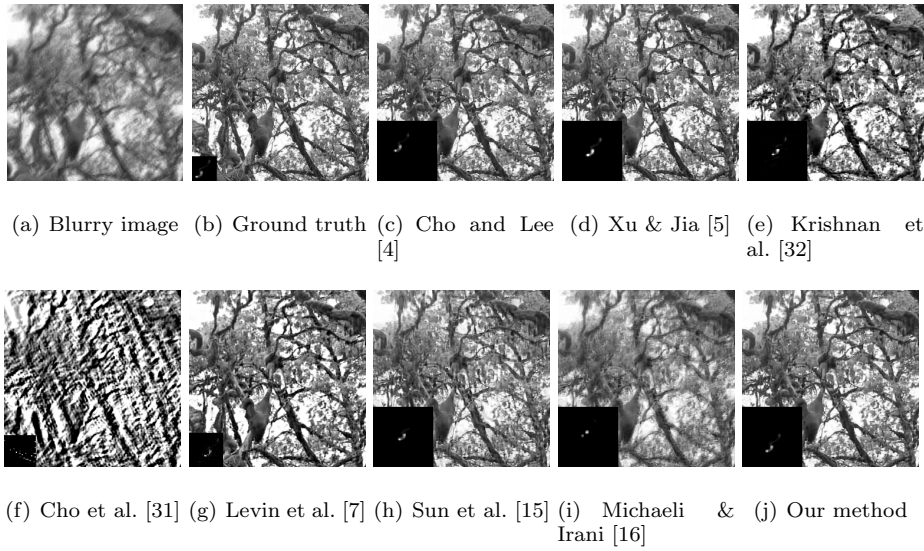
Table 4 Quantitative comparison of different methods over the dataset of [15]

	Success rate%	Mean error ratio
Ours	96.56	2.1134
Yu et al. [22]	96.25	2.2047
Michaeli & Irani [16]	95.94	2.5662
Sun et al. [15]	93.44	2.3764
Xu & Jia [5]	85.63	3.6293
Levin et al. [7]	46.72	6.5577
Cho & Lee [4]	65.47	8.6901
Krishnan et al. [32]	24.49	11.5212
Cho et al. [31]	11.74	24.7020

blurry image by performing the non-blind deconvolution method of [29] in the deblurring stage once the blur kernel has been estimated. Several methods are terminated early during the iteration due to lack of memory caused by too large the blur kernel. Fig.9 shows a visual comparison example with the state-of-the-art blind deconvolution methods [5,32,15,9,33,18,19] on one blurred image from Kohler et al.'s dataset [34], at the bottom of which are close-ups of different parts of these images. The results illustrate a noticeable contrast improvement that our method recovers sharper edges and more fine details with negligible artifacts,

(a) Blurry image (b) Ground truth (c) Cho and Lee [4] (d) Xu & Jia [5] (e) Krishnan et al. [32]

(f) Cho et al. [31] (g) Levin et al. [7] (h) Sun et al. [15] (i) Michaeli & Irani [16] (j) Our method

Fig. 7 Qualitative comparison of different methods on a cropped image from the synthetic dataset of [15]

(a) Blurry image (b) Ground truth (c) Cho and Lee [4] (d) Xu & Jia [5] (e) Krishnan et al. [32]

(f) Cho et al. [31] (g) Levin et al. [7] (h) Sun et al. [15] (i) Michaeli & Irani [16] (j) Our method

Fig. 8 Qualitative comparison of different methods on another cropped image from the synthetic dataset of [15]

and achieves better visual quality, as it estimates more accurate blur kernels. We observe from Fig. 9 that the deblurred images by Perrone et al. [9,33] suffer from ringing artifacts, and some fine details such as the fence and the lantern are not properly recovered by Pan et al. [18] and Yan et al. [19]. Fig.10 gives another visual comparison example with the state-of-the-art blind deconvolution methods [5,32,15,16,9,33,22]. The size of the blur kernel can be automatically estimated in the pre-processing stage. In the above examples, the sizes of the blur kernels are empirically initialized to 151×151 and 91×91 respectively. Experimental results on real blurry photographs with unknown large blurs validate that our method is quite robust to deal with large blurs.

When the blur is close to or even wider than the edge, the structure of the sharp edge will significantly change after blur. For such a highly blurred image, insignificant edges do not always provide useful information and instead mistake the kernel estimation. Nevertheless, large-scale structures are confused slightly by the blur due to their salient edges and provide informative edges for blur kernel estimation. Accordingly, it is more reasonable to obtain an accurate estimate of the blur kernel relying on salient edges. For small blurs, most of the edges are wider than the blur kernel and all helpful for kernel estimation besides salient edges. In this case, the edge-based method proposed in this paper only has a slight improvement over our previous method without considering salient edges [22]. But for large blurs, since insignificant edges could disturb kernel estimation and only salient edges around large-scale structures help kernel estimation, the edge-based method can achieve much better deblurring results and successfully handle severely blurred images.

5 Conclusion

In this paper, we have presented an edge-based blur kernel estimation method for blind motion deblurring unifying sparse representation and self-similarity of edge patches as image priors to guide the recovery of the latent image. We construct the sparsity regularizer and the cross-scale non-local regularizer based on our patch priors, exploiting thoroughly prior knowledge from similar patches across different scales of the latent image, and incorporate these two regularizers into our blind deconvolution model. We find that our regularizers prefer the sharp image to the blurred one only around salient edges, and accordingly impose our regularizers on salient edge patches of the image for blur kernel estimation. We have extensively validated the performance of our method, and it is able to deblur images with excessively large blur kernels.
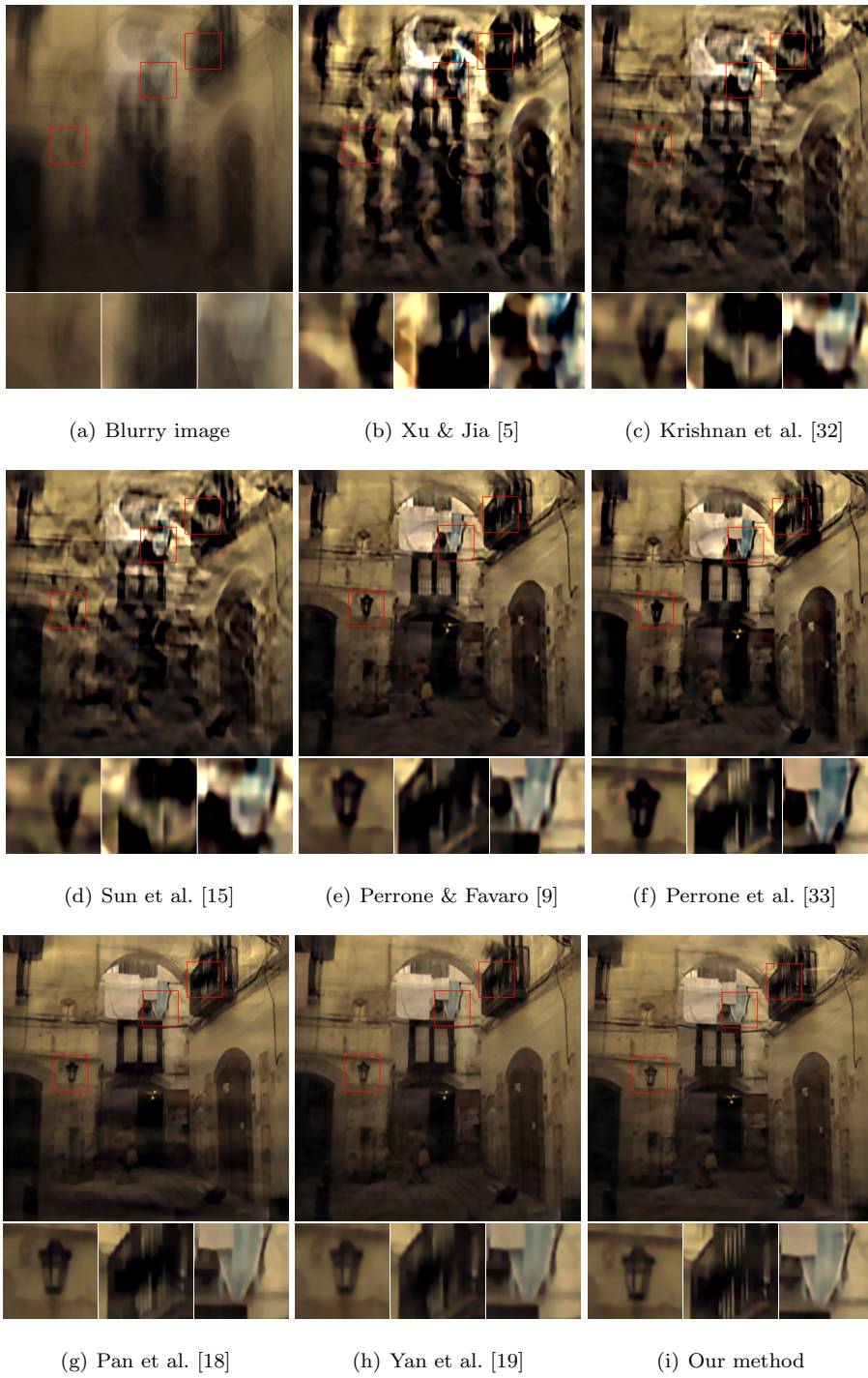
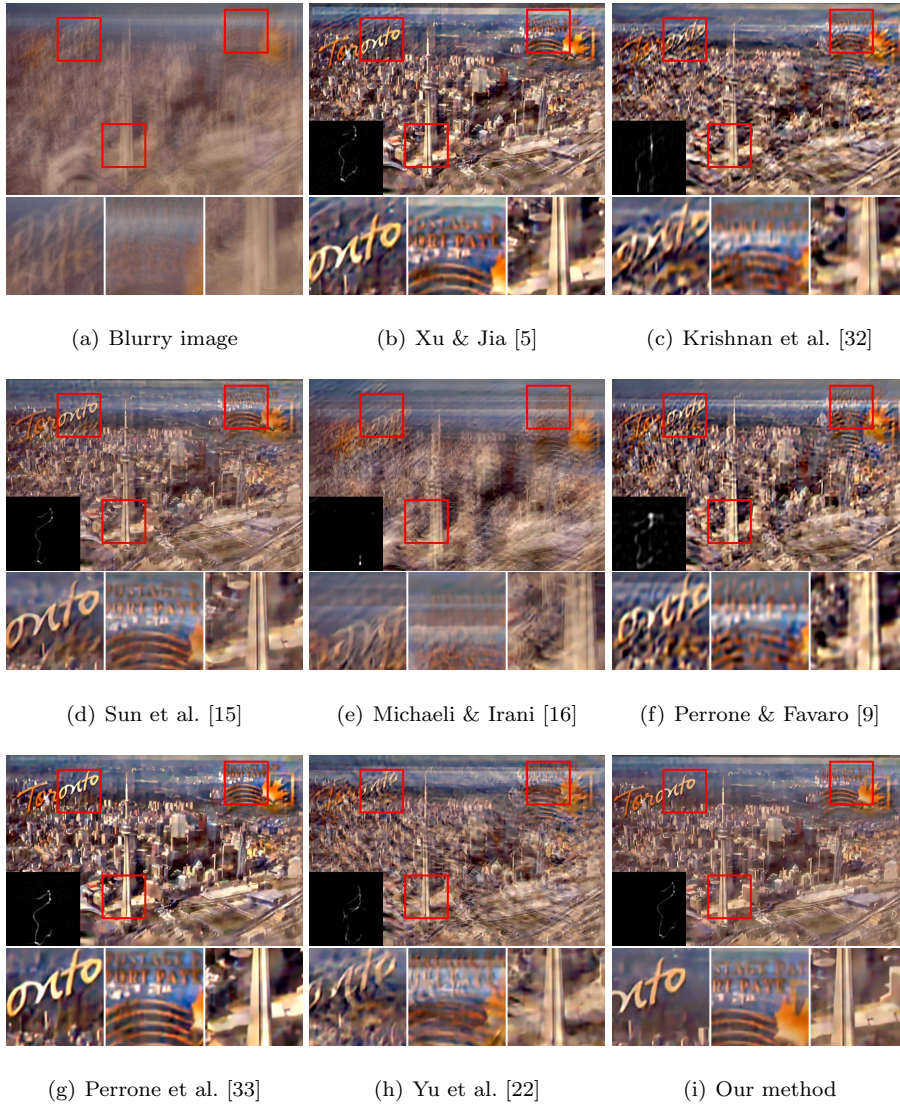### Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflicts of interest.

References

1. R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, W. T. Freeman, Removing camera shake from a single photograph, ACM Transactions on Graphics 25 (3) (2006) 787–794.

(a) Blurry image   (b) Xu & Jia [5]   (c) Krishnan et al. [32]

(d) Sun et al. [15]   (e) Perrone & Favaro [9] (f) Perrone et al. [33]

(g) Pan et al. [18]   (h) Yan et al. [19]   (i) Our method

Fig. 9 Visual comparison between our method and some state-of-the-art methods on real blurry image with unknown large blur

(a) Blurry image   (b) Xu & Jia [5]   (c) Krishnan et al. [32]

(d) Sun et al. [15]   (e) Michaeli & Irani [16]   (f) Perrone & Favaro [9]

(g) Perrone et al. [33]   (h) Yu et al. [22]   (i) Our method

Fig. 10 Visual comparison between our method and some state-of-the-art methods on another real blurry image with unknown large blur

2. J. Jia, Single image motion deblurring using transparency, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Minneapolis, Minnesota, USA, 2007, pp. 1–8.

3. N. Joshi, R. Szeliski, D. Kriegman, Psf estimation using sharp edge prediction, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Anchorage, AK, 2008, pp. 1–8.

4. S. Cho, S. Lee, Fast motion deblurring, ACM Transactions on Graphics 28 (5) (2009) 89–97.

5. L. Xu, J. Jia, Two-phase kernel estimation for robust motion deblurring, in: European conference on Computer vision: Part I, European conference on Computer vision: Part I, Springer Berlin Heidelberg, Heraklion, Crete, Greece, 2010, pp. 157–170.

6. A. Levin, Y. Weiss, F. Durand, W. T. Freeman, Understanding and evaluating blind deconvolution algorithms, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Miami, FL, 2009, pp. 1964–1971.

7. A. Levin, Y. Weiss, F. Durand, W. T. Freeman, Efficient marginal likelihood optimization in blind deconvolution, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Providence, RI, 2011, pp. 2657–2664.

8. Q. Shan, J. Jia, A. Agarwala, High-quality motion deblurring from a single image, ACM Transactions on Graphics 27 (3) (2008) 15–19.

9. D. Perrone, P. Favaro, Total variation blind deconvolution: The devil is in the details, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Columbus, OH, 2014, pp. 2909–2916.

10. D. Perrone, P. Favaro, A clearer picture of total variation blind deconvolution, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (6) (2016) 1041–1055.

11. A. Levin, R. Fergus, F. E. D. Durand, W. T. Freeman, Image and depth from a conventional camera with a coded aperture, ACM Transactions on Graphics (TOG) 26 (3).

12. Z. Pan, J. Yu, H. Huang, S. Hu, A. Zhang, H. Ma, W. Sun, Super-resolution based on compressive sensing and structural self-similarity for remote sensing images, IEEE Transactions on Geoscience and Remote Sensing 51 (9) (2013) 4864–4876.

13. M. Wang, J. Yu, W. Sun, Group-based hyperspectral image denoising using low rank representation, in: 2015 IEEE International Conference on Image Processing (ICIP), 2015 IEEE International Conference on Image Processing (ICIP), 2015, pp. 1623–1627.

14. C. Jia, B. L. Evans, Patch-based image deconvolution via joint modeling of sparse priors, in: IEEE International Conference on Image Processing (ICIP), IEEE International Conference on Image Processing (ICIP), IEEE, Brussels, Belgium, 2011, pp. 681–684.

15. L. Sun, S. Cho, J. Wang, J. Hays, Edge-based blur kernel estimation using patch priors, in: IEEE International Conference on Computational Photography (ICCP), IEEE International Conference on Computational Photography (ICCP), IEEE, Cambridge, MA, 2013, pp. 1–8.

16. T. Michaeli, M. Irani, Blind deblurring using internal patch recurrence, in: European Conference on Computer Vision (ECCV), European Conference on Computer Vision (ECCV), Springer International Publishing, Zurich, Switzerland, 2014, pp. 783–798.

17. W. S. Lai, J. J. Ding, Y. Y. Lin, Y. Y. Chuang, Blur kernel estimation using normalized color-line priors, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Boston, MA, United states, 2015, pp. 64–72.

18. J. Pan, D. Sun, H. Pfister, M. H. Yang, Blind image deblurring using dark channel prior, 2016, pp. 1628–1636.

19. Y. Yan, W. Ren, Y. Guo, R. Wang, X. Cao, Image deblurring via extreme channels prior, 2017, pp. 6978–6986.

20. H. Zhang, J. Yang, Y. Zhang, T. S. Huang, Sparse representation based blind image deblurring, in: IEEE International Conference on Multimedia and Expo (ICME), IEEE International Conference on Multimedia and Expo (ICME), IEEE, Barcelona, Spain, 2011, pp. 1–6.

21. H. Li, Y. Zhang, H. Zhang, Y. Zhu, J. Sun, Blind image deblurring based on sparse prior of dictionary pair, in: International Conference on Pattern Recognition (ICPR), International Conference on Pattern Recognition (ICPR), IEEE, Tsukuba, 2012, pp. 3054–3057.

22. J. Yu, Z. Chang, C. Xiao, W. Sun, Blind image deblurring based on sparse representation and structural self-similarity, in: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, New Orleans, LA, USA, 2017, pp. 1328–1332.

23. M. Aharon, M. Elad, A. Bruckstein, Svd: An algorithm for designing overcomplete dictionaries for sparse representation, IEEE Transactions on Signal Processing 54 (11) (2006) 4311–4322.

24. J. A. Tropp, A. C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, IEEE Transactions on Information Theory 53 (12) (2007) 4655–4666.
25. D. Glasner, S. Bagon, M. Irani, Super-resolution from a single image, in: International Conference on Computer Vision, ICCV 2009, International Conference on Computer Vision, ICCV 2009, IEEE, Kyoto, Japan, 2009, pp. 349–356.
26. A. Buades, B. Coll, J.-M. Morel, A non-local algorithm for image denoising, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, San Diego, CA, United states, 2005, pp. 60–65.
27. M. Protter, M. Elad, H. Takeda, P. Milanfar, Generalizing the nonlocal-means to super-resolution reconstruction, IEEE Transactions on Image Processing 18 (1) (2009) 36–51.
28. W. Dong, L. Zhang, G. Shi, X. Wu, Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization, IEEE Transactions on Image Processing 20 (7) (2011) 1838–1857.
29. D. Zoran, Y. Weiss, From learning models of natural image patches to whole image restoration, in: IEEE International Conference on Computer Vision (ICCV), IEEE International Conference on Computer Vision (ICCV), IEEE, Barcelona, 2011, pp. 479–486.
30. I. Olonetsky, S. Avidan, Treecann - k-d tree coherence approximate nearest neighbor algorithm, in: European Conference on Computer Vision, European Conference on Computer Vision, Springer Berlin Heidelber, Florence, Italy, 2012, pp. 602–615.
31. T. S. Cho, S. Paris, B. K. P. Horn, W. T. Freeman, Blur kernel estimation using the radon transform, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 42 of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, 2011, pp. 241–248.
32. D. Krishnan, T. Tay, R. Fergus, Blind deconvolution using a normalized sparsity measure, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Providence, RI, 2011, pp. 233–240.
33. D. Perrone, R. Diethelm, P. Favaro, Blind deconvolution via lower-bounded logarithmic image priors, in: International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR), International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR), Springer International Publishing, Hong Kong, China, 2015.
34. R. Kohler, M. Hirsch, B. Mohler, B. Sch Lkopf, S. Harmeling, Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database, in: European Conference on Computer Vision (ECCV), European Conference on Computer Vision (ECCV), Springer Verlag, Germany, Florence, Italy, 2012, pp. 27–40.