Two-Phase Kernel Estimation for Robust Motion Deblurring

Li Xu and Jiaya Jia

Department of Computer Science and Engineering
The Chinese University of Hong Kong
{xuli,leojia}@cse.cuhk.edu.hk

Abstract. We discuss a few new motion deblurring problems that are significant to kernel estimation and non-blind deconvolution. We found that strong edges do not always profit kernel estimation, but instead under certain circumstance degrade it. This finding leads to a new metric to measure the usefulness of image edges in motion deblurring and a gradient selection process to mitigate their possible adverse effect. We also propose an efficient and high-quality kernel estimation method based on using the spatial prior and the iterative support detection (ISD) kernel refinement, which avoids hard threshold of the kernel elements to enforce sparsity. We employ the TV-ℓ_1 deconvolution model, solved with a new variable substitution scheme to robustly suppress noise.

1 Introduction

Motion deblurring was hotly discussed in the computer vision and graphics community due to its involvement of many challenges in problem formulation, regularization, and optimization. Notable progress has been made lately [1–6]. The blur process caused by camera shake is generally modeled as a latent image convolved with a blur point-spread-function (a.k.a. kernel).

The success of recent single-image methods partly stems from the use of various sparse priors, for either the latent images or motion blur kernels [1, 3, 6]. It was found that without these constraints, iterative kernel estimation is easily stuck in local minima and possibly results in a dense kernel and many visual artifacts in the restored image. However, minimizing a non-convex energy function with the kernel-sparsity prior is usually costly.

Another group of methods seek high efficiency and resort to explicitly detecting salient image structures. They use the Gaussian kernel priors [4, 5, 7] instead of the sparse ones. These approaches greatly shorten the computation time; but the Gaussian priors sometimes issue in noisy or dense kernel estimates, which need to be post-processed by threshold-like operations.

Despite the efficiency and accuracy issues, another critical motion deblurring problem that was not known yet is on how image structure influences kernel estimation. Our intriguing finding is that salient edges do not always help kernel refinement, but instead in some commonly encountered circumstances greatly

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part I, LNCS 6311, pp. 157–170, 2010.

increase the estimation ambiguity. We will analyze this problem and propose an automatic gradient selection algorithm to exclude the detrimental structures.

Our method also makes several other contributions. 1) First, we propose a novel two-phase kernel estimation algorithm to separate computationally expensive non-convex optimization from quick kernel initialization, giving rise to an efficient and robust kernel estimation process. 2) We introduce a new spatial prior to preserve sharp edges in quick latent image restoration. 3) In the kernel refinement stage, we employ the Iterative Support Detection (ISD) algorithm, which is a powerful numerical scheme through iterative support detection, to adaptively enforce the sparsity constraint and properly preserve large-value elements. Soft-threshold-like effect is achieved in this step. 4) Finally, to restore the latent image, we employ a TV-ℓ_1 objective function that is robust to noise and develop an efficient solver based on half-quadratic splitting.

We applied our method to challenging examples, where many images are blurred with very large PSFs (spanning up to 100 pixels in width or height) due to camera shake. Our "robust deblurring" project website is put online[1], which includes the motion deblurring executable and image data.

1.1 Related Work

Shift-invariant motion blur can be modeled as image convolution with a PSF. We briefly review the blind and non-blind deconvolution methods.

Blind Deconvolution. Early work on blind image deconvolution focuses on estimating small-size blur kernels. For example, You and Kaveh [8] proposed a variational framework to estimate small Gaussian kernels. Chan and Wong [9] applied the Total Variation regularizers to both kernels and images. Another group of methods [10–12] did not compute the blur kernels, but studied the reversion of a diffusion process.

Lately, impressive progress has been made in estimating a complex motion blur PSF from a single image [1, 3, 6]. The success arises in part from the employment of sparse priors and the multi-scale framework. Fergus *et al.* [1] used a zero-mean Mixture of Gaussian to fit the heavy-tailed natural image prior. A variational Bayesian framework was employed. Shan *et al.* [3] also exploited the sparse priors for both the latent image and blur kernel. Deblurring is achieved through an alternating-minimization scheme. Cai *et al.* [6] introduced a framelet and curvelet system to obtain the sparse representation for kernels and images. Levin *et al.* [13] showed that common MAP methods involving estimating both the image and kernel likely fail because they favor the trivial solution. Special attention such as edge re-weighting is probably the remedy. It is notable that using sparse priors usually result in non-convex objective functions, encumbering efficient optimization.

Another group of methods [4, 5, 7] do not use sparse priors, but instead employ an explicit edge prediction step for the PSF estimation. Specifically, Joshi *et al.* [4] predicted sharp edges by first locating step edges and then propagating

[1] `http://www.cse.cuhk.edu.hk/~leojia/projects/robust_deblur/index.html`

the local intensity extrema towards the edge. This method was used to handle complex PSFs with a multi-scale scheme [7]. Cho and Lee [5] adopted bilateral filtering together with shock filtering to predict sharp edges. These methods impose simple Gaussian priors, which avail to construct quick solvers. These priors however cannot capture the sparse nature of the PSF and image structures, which occasionally make the estimates noisy and dense.

Non-blind deconvolution. Given a known blur PSF, the process of restoring an unblurred image is referred to as non-blind deconvolution. Early work such as Richardson-Lucy (RL) or Weiner filtering is known as sensitive to noise. Yuan *et al.* [14] proposed a progressive multi-scale refinement scheme based on an edge-preserving bilateral Richardson-Lucy (BRL) method. Total Variation regularizer (also referred to as Laplacian prior) [9], heavy-tailed natural image priors [1, 3] and Hyper-Laplacian priors [15–18] were also extensively studied.

To suppress noise, Bar *et al.* [19] used the ℓ_1 fidelity term together with a Mumford-Shah regularizer to reject impulse noise. Joshi *et al.* [20] incorporated a local two-color prior to suppress noise. These methods used the iterative re-weighted least square to solve the nonlinear optimization problem, which inevitably involves intensive computation. In this paper, we developed a fast TV-ℓ_1 deconvolution method based on half-quadratic splitting [16, 18], to efficiently reject outliers and preserve structures.

2 Two-Phase Sparse Kernel Estimation

By convention, the blur process is modeled as

$$\mathrm{B} = \mathrm{I} \otimes k + \varepsilon,$$

where I is the latent image, k is the blur kernel, ε is the image noise, $\otimes$ denotes convolution and B is the observed blur image. In this section, we introduce a two-phase method for PSF estimation. The first stage aims to efficiently compute a coarse version of the kernel without enforcing much sparsity. In the second phase, although non-convex optimization is employed, with the initial kernel estimate propagated from stage one, no significant computation is required to produce the final result.

2.1 Phase One: Kernel Initialization

In the first step, we estimate the blur kernel in a multi-scale setting. High efficiency can be yielded as we use the Gaussian priors where closed-form solutions exist. The algorithm is sketched in Alg. 1. with three main steps – that is, sharp edge construction, kernel estimation, and coarse image restoration.

In the first place, like other motion deblurring methods, we filter the image and predict salient edges to guide the kernel initialization. We use Gaussian

Algorithm 1. Kernel Initialization

INPUT: Blur image B and an all-zero kernel (size h × h)
Build an image pyramid with level index $\{1, 2, \cdots, n\}$.
for $l = 1$ to n **do**
  Compute gradient confidence r for all pixels (Eq. (2)).
  for $i = 1$ to m (m is the number of iterations) **do**
    (a) Select edges $\nabla \mathrm{I}^s$ for kernel estimation based on confidence r (Eq. (4)).
    (b) Estimate kernel with the Gaussian prior (Eq. (6)).
    (c) Estimate the latent image I^l with the spatial prior (Eq. (8)), and update $\tau_s \leftarrow \tau_s/1.1$, $\tau_r \leftarrow \tau_r/1.1$.
  end for
  Upscale image $\mathrm{I}^{l+1} \leftarrow \mathrm{I}^l \uparrow$.
end for
OUTPUT: Kernel estimate k^0 and sharp edge gradient map $\nabla \mathrm{I}^s$

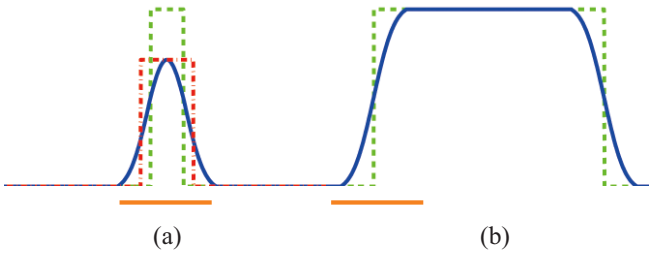


Fig. 1. Ambiguity in motion deblurring. Two latent signals (green dashed lines) in (a) and (b) are blurred (shown in blue) with the same Gaussian kernel. In (a), the blurred signal is not total-variation preserving, making the kernel estimation ambiguous. In fact, the red curve is more likely the latent signal than the green one in a common optimization process. The bottom orange lines indicate the input kernel width.

filtering to pre-smooth the image and then solve the following shock filtering PDE problem [10] to construct significant step edges:

$$\partial \mathrm{I}/\partial t = -\mathrm{sign}(\Delta \mathrm{I})\|\nabla \mathrm{I}\|, \qquad \mathrm{I}_0 = G_\sigma \otimes \mathrm{I}_{input}, \tag{1}$$

where $\nabla \mathrm{I} = (\mathrm{I}_x, \mathrm{I}_y)'$ and $\Delta \mathrm{I} = \mathrm{I}_x^2 \mathrm{I}_{xx} + 2\mathrm{I}_x \mathrm{I}_y \mathrm{I}_{xy} + \mathrm{I}_y^2 \mathrm{I}_{yy}$ are the first- and second-order spatial derivatives respectively. I_0 denotes the Gaussian smoothed input image, which serves as an initial input for iteratively updating $\partial \mathrm{I}/\partial t$.

Selective Edge Map for Kernel Estimation. Insignificant edges make PSF estimation vulnerable to noise, as discussed in [3–5, 13]. We however observe a different connection between image edges and the quality of kernel estimation – that is, salient edges do not always improve kernel estimation; on the contrary, *if the scale of an object is smaller than that of the blur kernel, the edge information could damage kernel estimation.*

We give an example in Figure 1. Two step signals (the green dashed lines) in (a) and (b) are blurred with a large PSF. The observed blur signals are shown

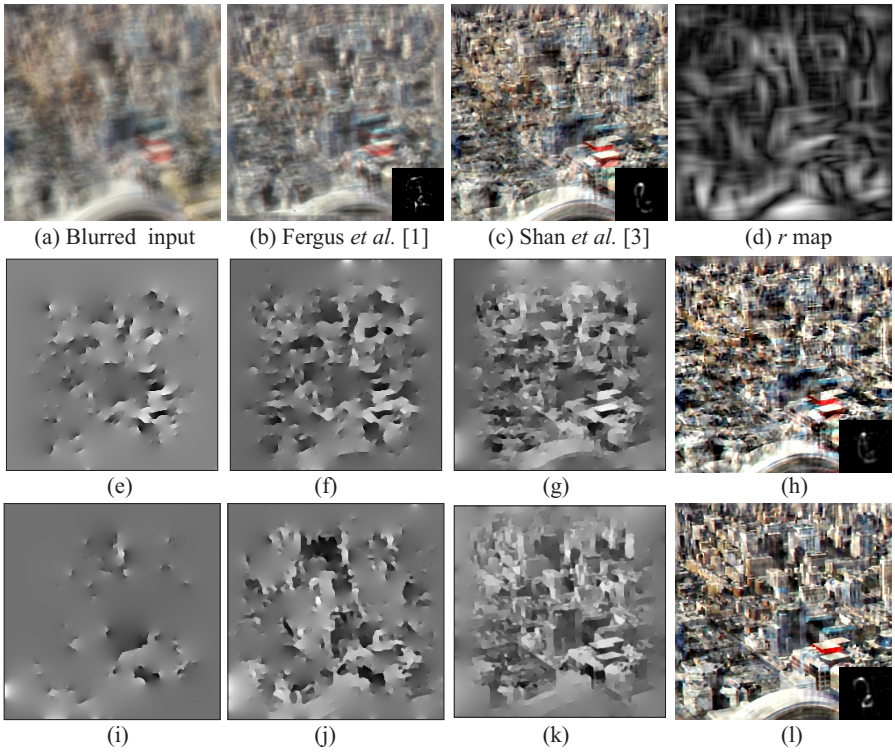


Fig. 2. Image structure influence in kernel estimation. (a) Blurred image. (b) Result of Fergus *et al.* [1]. (c) Result of Shan *et al.* [3]. (d) r map (by Eq. (2)). (e)-(g) $\nabla \mathrm{I}^s$ maps, visualized using Poisson reconstruction, in the 1st, 2nd and 7th iterations without considering r. (h) Deblurring result not using the r map. (i)-(k) $\nabla \mathrm{I}^s$ maps computed according to Eq. (4). (l) Our final result. The blur PSF is of size 45×45.

in blue. Because the left signal is horizontally narrow, the blur process lowers its height in (a), yielding ambiguity in the latent signal restoration. Specifically, motion blur methods imposing sparse priors on the gradient map of the latent image [1, 3] will favor the red dashed line in computing the unblurred signal because this version presents smaller gradient magnitudes. Moreover, the red signal preserves the total variation better than the green one. So it is also a more appropriate solution for the group of methods using sharp edge prediction (including shock filtering and the method of [4]). This example shows that if image structure magnitude significantly changes after blur, the corresponding edge information could mistake kernel estimation.

In comparison, the larger-scale object shown in Figure 1(b) can yield stable kernel estimation because it is wider than the kernel, preserving the total variation of the latent signal along its edges.

Figure 2 shows an image example. The blurred input (shown in (a)) contains rich edge information along many small-scale objects. The results of Fergus *et*

al. [1] (b) and Shan *et al.* [3] (c) are computed by extensively hand-tuning parameters. However, the correct kernel estimate still cannot be found, primarily due to the aforementioned small structure problem.

We propose a new criterion for selecting informative edges for kernel estimation. The new metric to measure the usefulness of gradients is defined as

$$r(x) = \frac{\|\sum_{y\in N_{\mathrm{h}}(x)} \nabla \mathrm{B}(y)\|}{\sum_{y\in N_{\mathrm{h}}(x)} \|\nabla \mathrm{B}(y)\| + 0.5}, \tag{2}$$

where B denotes the blurred image and $N_{\mathrm{h}}(x)$ is a h × h window centered at pixel x. 0.5 is to prevent producing a large r in flat regions. The signed $\nabla \mathrm{B}(y)$ for narrow objects (spikes) will mostly cancel out in $\|\sum_{y\in N_{\mathrm{h}}(x)} \nabla \mathrm{B}(y)\|$. $\sum_{y\in N_{\mathrm{h}}(x)} \|\nabla \mathrm{B}(y)\|$ is the sum of the absolute gradient magnitudes in $N_{\mathrm{h}}(x)$, which estimates how strong the image structure is in the window. A small r implies that either spikes or a flat region is involved, which causes neutralizing many gradient components. Figure 2(d) shows the computed r map.

We then rule out pixels belonging to small r-value windows using a mask

$$\mathrm{M} = H(r - \tau_r), \tag{3}$$

where $H(\cdot)$ is the Heaviside step function, outputting zeros for negative values and ones otherwise. τ_r is a threshold. The final selected edges for kernel estimation are determined as

$$\nabla \mathrm{I}^s = \nabla \tilde{\mathrm{I}} \cdot H(\mathrm{M}\|\nabla \tilde{\mathrm{I}}\|_2 - \tau_s), \tag{4}$$

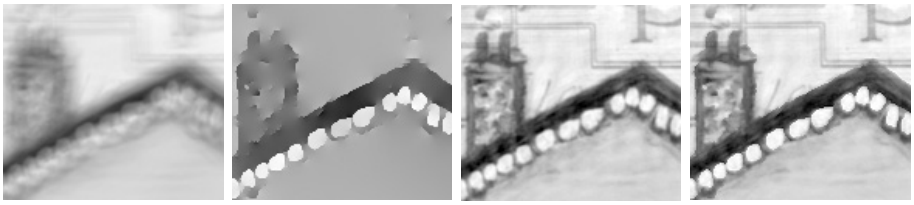
where $\tilde{\mathrm{I}}$ denotes the shock filtered image and τ_s is a threshold of the gradient magnitude. Eq. (4) excludes part of the gradients, depending jointly on the magnitude $\|\nabla \tilde{\mathrm{I}}\|_2$ and the prior mask M. This selection process reduces ambiguity in the following kernel estimation.

Figures 2(e)-(g) and (i)-(k) illustrate the computed $\nabla \mathrm{I}^s$ maps in different iterations without and with the edge selection operation. The comparison shows that including more edges do not necessarily benefit kernel estimation. Optimization could be misled especially in the first a few iterations. So an image edge selection process is vital to reduce the confusion.

To allow for inferring subtle structures during kernel refinement, we decrease the values of τ_r and τ_s in iterations (divided by 1.1 in each pass), to include more and more edges. So the maps in (g) and (k) contain similar amount of edges. But the quality notably differs. The method to compute the final results shown in (h) and (l) is detailed further below.

Fast Kernel Estimation. With the critical edge selection, initial kernel estimation can be accomplished quickly. We define the objective function with a Gaussian regularizer as

$$\mathrm{E}(k) = \|\nabla \mathrm{I}^s \otimes k - \nabla \mathrm{B}\|^2 + \gamma \|k\|^2, \tag{5}$$



(a) Blurred input (b) $\nabla \mathrm{I}^s$ map (c) Gaussian prior (d) Our spatial prior

Fig. 3. Comparison of results using the sparse $\|\nabla \mathrm{I}\|^2$ and spatial $\|\nabla \mathrm{I} - \nabla \mathrm{I}^s\|^2$ priors. The spatial prior makes the result in (d) preserve more sharp edges.

where γ is a weight. Based on the Parseval's theorem, we perform FFTs on all variables and set the derivative w.r.t. k to zero. The closed-form solution is given by

$$k = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(\partial_x \mathrm{I}^s)}\mathcal{F}(\partial_x \mathrm{B}) + \overline{\mathcal{F}(\partial_y \mathrm{I}^s)}\mathcal{F}(\partial_y \mathrm{B})}{\mathcal{F}(\partial_x \mathrm{I}^s)^2 + \mathcal{F}(\partial_y \mathrm{I}^s)^2 + \gamma}\right), \tag{6}$$

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

Coarse Image Estimation with a Spatial Prior We use the predicted sharp edge gradient $\nabla \mathrm{I}^s$ as a spatial prior to guide the recovery of a coarse version of the latent image. The objective function is

$$\mathrm{E}(\mathrm{I}) = \|\mathrm{I} \otimes k - \mathrm{B}\|^2 + \lambda \|\nabla \mathrm{I} - \nabla \mathrm{I}^s\|^2, \tag{7}$$

where the new spatial prior $\|\nabla \mathrm{I} - \nabla \mathrm{I}^s\|^2$ does not blindly enforce small gradients near strong edges and thus allows for a sharp recovery even with the Gaussian regularizer. The closed-form solution exists. With a few algebraic operations in the frequency domain, we obtain

$$\mathrm{I} = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(k)}\mathcal{F}(\mathrm{B}) + \lambda(\overline{\mathcal{F}(\partial_x)}\mathcal{F}(\mathrm{I}^s_x) + \overline{\mathcal{F}(\partial_y)}\mathcal{F}(\mathrm{I}^s_y))}{\overline{\mathcal{F}(k)}\mathcal{F}(k) + \lambda(\overline{\mathcal{F}(\partial_x)}\mathcal{F}(\partial_x) + \overline{\mathcal{F}(\partial_y)}\mathcal{F}(\partial_y))}\right). \tag{8}$$

Figure 3 compares the deconvolution results using the spatial and Gaussian priors respectively (the latter is usually written as $\|\nabla \mathrm{I}\|^2$). The regularization weight $\lambda = 2e^{-3}$. The image shown in Figure 3(d) contains well preserved sharp edges.

2.2 Phase Two: ISD-Based Kernel Refinement

To obtain sparse PSFs, previous methods [1, 3, 5, 21] apply hard or hysteresis thresholding to the kernel estimates. These operations however ignore the inherent blur structure, possibly degrading the kernel quality. One example is shown

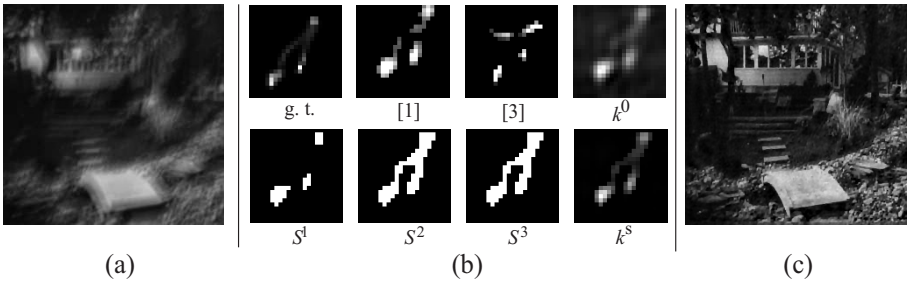


Fig. 4. Sparse Kernel Refinement. (a) A blurred image [13]. (b) Kernels. The top row shows respectively the ground truth kernel, the kernel estimates of Fergus *et al.* [1], Shan *et al.* [3], and of our method in phase one. k^s in the bottom row is our final result after kernel refinement. S^1-S^3 show the iteratively detected support regions by the ISD method. (c) Our restored image using k^s.

in Figure 4(b), where only keeping the large-value elements apparently cannot preserve the subtle structure of the motion PSF.

We solve this problem using an iterative support detection (ISD) method that can ensure the deblurring quality while removing noise. The idea is to iteratively secure the PSF elements with large values by relaxing the regularization penalty. So these elements will not be significantly affected by regularization in the next-round kernel refinement. This strategy was shown in [22] capable of correcting imperfect estimates and converging quickly.

ISD is an iterative method. At the beginning of each iteration, previously estimated kernel k^i is used to form a partial support; that is, large-value elements are put into a set S^{i+1} and all others belong to the set $\overline{S^{i+1}}$. S^{i+1} is constructed as

$$S^{i+1} \leftarrow \{j : k^i_j > \epsilon^s\}, \tag{9}$$

where j indexes the elements in k^i and ϵ^s is a positive number, evolving in iterations, to form the partial support. We configure ϵ^s by applying the "first significant jump" rule [22]. Briefly speaking, we sort all elements in k^i in an ascending order w.r.t. their values and compute the differences d_0, $d_1 \cdots$ between each two nearby elements. Then we exam these differences sequentially starting from the head d_0 and search for the first element, d_j for example, that satisfies $d_j > \|k^i\|_\infty/(2\mathrm{h} \cdot i)$, where h is the kernel width and $\|k^i\|_\infty$ returns the largest value in k^i. We then assign the kernel value in position j to ϵ^s. More details are presented in [22]. Examples of the detected support are shown in the bottom row of Figure 4(b). The elements within each S will be less penalized in optimization, resulting in an adaptive kernel refinement process.

We then minimize

$$\mathrm{E}(k) = \frac{1}{2}\|\nabla \mathrm{I}^s \otimes k - \nabla \mathrm{B}\|^2 + \gamma \sum_{j \in \overline{S^{i+1}}} |k_j| \tag{10}$$

Algorithm 2. ISD-based Kernel Refinement

INPUT: Initial kernel k^0, ∇B, and $\nabla \mathrm{I}^s$ (output of Algorithm 1)
Initialize the partial support $\overline{S^0}$ on k^0 (Eq. (9)).
repeat
  Solve for k^i by minimizing Eq. (10).
  Update $\overline{S}$ (Eq. (9)).
  $i \leftarrow i + 1$.
until $\frac{\|k^{i+1} - k^i\|}{\|k^i\|} \leq \epsilon_k$ ($\epsilon_k = 1e^{-3}$ empirically)
OUTPUT: Kernel estimate k^s

for PSF refinement. The difference between this function and those used in [3, 6] is on the definition of the regularization terms. Thresholding applies softly in our function through adaptive regularization, which allows the energy to concentrate on significant values and thus automatically maintains PSF sparsity, faithful to the deblurring process. The algorithm is outlined in Alg. 2..

To minimize Eq. (10) with the partial support, we employed the iterative reweighed least square (IRLS) method. By writing convolution as matrix multiplication, the latent image I, kernel k, and blur input B are correspondingly expressed as matrix A, vector V_k, and vector V_B. Eq. (10) is then minimized by iteratively solving linear equations w.r.t. V_k. In the t-th pass, the corresponding linear equation is expressed as

$$[\mathrm{A}^T\mathrm{A} + \gamma \mathrm{diag}(V_{\bar{S}} \Psi^{-1})] V_\mathrm{k}^t = \mathrm{A}^T V_\mathrm{B}, \tag{11}$$

where A^T denotes the transposed version of A and $V_{\bar{S}}$ is the vector form of $\overline{S}$. Ψ is defined as $\Psi = \max(\|V_\mathrm{k}^{t-1}\|_1, 1e^{-5})$, which is the weight related to the kernel estimate from the previous iteration. diag$(\cdot)$ produces a diagonal matrix from the input vector. Eq. (11) can be solved by the conjugate gradient method in each pass (we alternatively apply the matrix division operation in Matlab). As PSFs have small size compared to images, the computation is very fast.

Our final kernel result k^s is shown in Figure 4(b). It maintains many small-value elements; meanwhile, the structure is appropriately sparse. Optimization in this phase converges in only a few iterations. Figure 4(c) shows our restored image using the computed PSF. It contains correctly reconstructed textures and small edges, verifying the quality of the kernel estimate.

3 Fast TV-ℓ_1 Deconvolution

Assuming the data fitting costs following a Gaussian distribution is not a good way to go in many cases. It possibly makes results vulnerable to outliers, as demonstrated in many literatures. To achieve high robustness, we propose a TV-ℓ_1 model in deconvolution, which is written as

$$\mathrm{E}(\mathrm{I}) = \|\mathrm{I} \otimes k - \mathrm{B}\| + \lambda \|\nabla \mathrm{I}\|. \tag{12}$$

Algorithm 3. Robust Deconvolution

INPUT: Blurred image B and the estimated kernel k^s
Edge taping in Matlab
$\mathrm{I} \leftarrow \mathrm{B}$, $\beta \leftarrow \beta_0$.
repeat
  Solve for v using Eq. (18)
  $\theta \leftarrow \theta_0$
  repeat
    Solve for w using Eq. (17)
    Solve for I in the frequency domain using Eq. (15)
    $\theta \leftarrow \theta/2$
  until $\theta < \theta_{\min}$
  $\beta \leftarrow \beta/2$
until $\beta < \beta_{\min}$
OUTPUT: Deblurred image I

It contains non-linear penalties for both the data and regularization terms. We propose solving it using an efficient alternating minimization method, based on a half-quadratic splitting for ℓ_1 minimization [16, 18].

For each pixel, we introduce a variable v to equal the measure $\mathrm{I} \otimes k - \mathrm{B}$. We also denote by $w = (w_x, w_y)$ image gradients in two directions. The use of these auxiliary variables leads to a modified objective function

$$\mathrm{E}(\mathrm{I}, w, v) = \frac{1}{2\beta}\|\mathrm{I} \otimes k - \mathrm{B} - v\|^2 + \frac{1}{2\theta}\|\nabla \mathrm{I} - w\|_2^2 + \|v\| + \lambda\|w\|, \tag{13}$$

where the first two terms are used to ensure the similarity between the measures and the corresponding auxiliary variables. When $\beta \to 0$ and $\theta \to 0$, the solution of Eq. (13) approaches that of Eq. (12).

With the adjusted formulation, Eq. (13) can now be solved by an efficient Alternating Minimization (AM) method, where the solver iterates among solving I, w, and v independently by fixing other variables. w and v are initialized to zeros.

In each iteration, we first compute I given the initial or estimated w and v by minimizing

$$\mathrm{E}(\mathrm{I}; w, v) = \|\mathrm{I} \otimes k - \mathrm{B} - v\|^2 + \frac{\beta}{\theta}\|\nabla \mathrm{I} - w\|_2^2. \tag{14}$$

Eq. (14) is equivalent to Eq. (13) after removing constants. As a quadratic function, Eq. (14) bears a closed form solution in minimization according to the Parseval's theorem after the Fourier transform. The optimal I is written as

$$\mathcal{F}(\mathrm{I}) = \frac{\overline{\mathcal{F}(k)}\mathcal{F}(B+v) + \beta/\theta(\overline{\mathcal{F}(\partial_x)}\mathcal{F}(w_x) + \overline{\mathcal{F}(\partial_y)}\mathcal{F}(w_y))}{\overline{\mathcal{F}(k)}\mathcal{F}(k) + \beta/\theta(\overline{\mathcal{F}(\partial_x)}\mathcal{F}(\partial_x) + \overline{\mathcal{F}(\partial_y)}\mathcal{F}(\partial_y))}. \tag{15}$$

The notations are the same as those in Eq. (6).

(a) Noisy input (b) [15] (c) [17] (d) Ours

Fig. 5. Deconvolution result comparison. The blurred images in the top and bottom rows are with Gaussian and impulse noise respectively.

In solving for w and v given the I estimate, because w and v are not coupled with each other in the objective function (they belong to different terms), their optimization is independent. Two separate objective functions are thus yielded:

$$\begin{cases} \mathrm{E(w;I)} = \frac{1}{2}\|w - \nabla \mathrm{I}\|_2^2 + \theta\lambda\|w\|_2 \\ \mathrm{E(v;I)} = \frac{1}{2}\|v - (\mathrm{I} \otimes k - \mathrm{B})\|^2 + \beta\|v\| \end{cases} \tag{16}$$

Each objective function in Eq. (16) categorizes to a single-variable optimization problem because the variables for different pixels are not spatially coupled. The optimal solutions for all w_xs can be derived according to the shrinkage formula:

$$w_x = \frac{\partial_x \mathrm{I}}{\|\nabla \mathrm{I}\|_2} \max(\|\nabla \mathrm{I}\|_2 - \theta\lambda, 0). \tag{17}$$

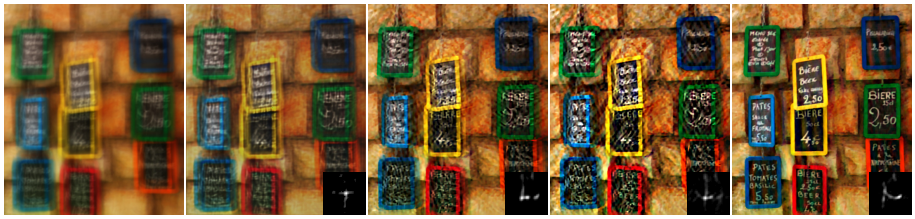
Here, isotropic TV regularizer is used – that is, $\|\nabla \mathrm{I}\|_2 = \sqrt{(\partial_x \mathrm{I})^2 + (\partial_y \mathrm{I})^2}$. w_y can be computed similarly using the above method.

Computing v can be even simpler because it is an one-dimensional shrinkage:

$$v = \mathrm{sign}(\mathrm{I} \otimes k - \mathrm{B}) \max(\|\mathrm{I} \otimes k - \mathrm{B}\| - \beta, 0), \tag{18}$$

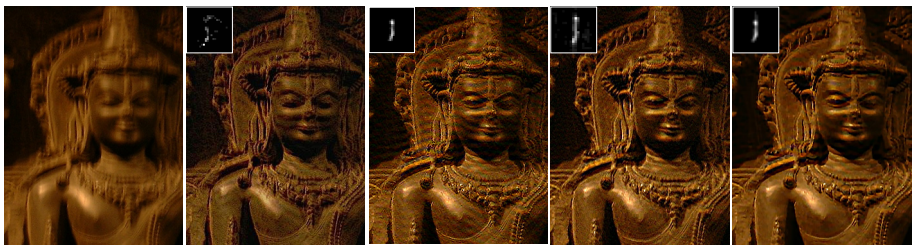
where β and θ are two small positive values to enforce the similarity between the auxiliary variables and the respective terms. To further speed up the optimization, we employ the warm-start scheme [3, 16]. It first sets large penalties (β and θ in our algorithm) and gradually decreases them in iterations. The details are shown in Alg. 3.. We empirically set $\beta_0 = 1$, $\theta_0 = \lambda^{-1}$, and $\beta_{\min} = \theta_{\min} = 0.01$.

Figure 5 shows examples where the blurred images are with Gaussian and impulse noise respectively. The TV-ℓ_1 model performs comparably to other state-of-the-art deconvolution methods under the Gaussian noise. When significant impulse-like sensor noise exists, it works even better. In terms of the computation time, the methods of [15] and [17] spend 3 minutes and 1.5 seconds respectively to produce the results in Figure 5 with the provided implementation while our deconvolution algorithm, albeit using the highly non-linear function, uses 6s in Matlab. All methods deconvolve three color channels independently.



(a) Input (b) [1] (c) [3] (d) no M (e) with M

Fig. 6. Small objects such as the characters and thin frames are contained in the image. They greatly increase the difficulty of motion deblurring. (d)-(e) show our results using and not using the M map. The blur kernel is of size 51×51.



(a) Input (b) [1] (c) [3] (d) [5] (e) Ours

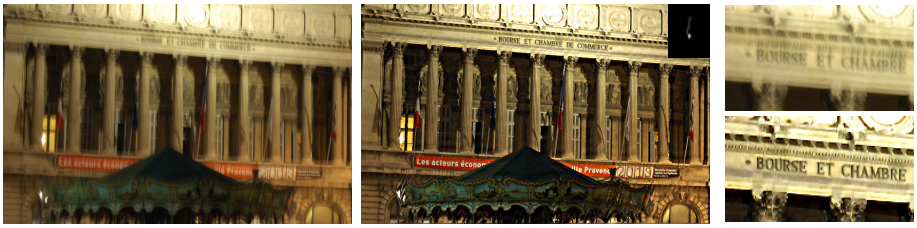
Fig. 7. Comparison of state-of-the-art deblurring methods

4 More Experimental Results

We experimented with several challenging examples where the images are blurred with large kernels. Our method generally allows using the default or automatically adapted parameter values. In the kernel estimation, we adaptively set the initial values of τ_r and τ_s, using the method of [5]. Specifically, the directions of image gradient are initially quantized into four groups. τ_s is set to guarantee that at least $2\sqrt{P_k}$ pixels participate in kernel estimation in each group, where P_k is the total number of pixels in kernel k. τ_r is similarly determined by allowing at least $0.5\sqrt{P_I P_k}$ pixels to be selected in each group. P_I is the total number of pixels in the input image. In the coarse kernel estimation phase, we set $\lambda = 2e^{-3}$ and $\gamma = 10$ to resist noise. In the kernel refinement, we set $\gamma = 1$. λ in the final image deconvolution is set to $2e^{-2}$.

Our two-phase kernel estimation is efficient because we put the non-convex optimization into the second phase. Our Matlab implementation spends about 25 seconds to estimate a 25×25 kernel from an 800×600 image with an Intel Core2Quad CPU@2.40G. The coarse kernel estimation uses 12s in the multi-scale framework while the kernel refinement spends 13s as it is performed only in the finest image scale.

In Figure 6(a), we show an example that contains many small but structurally-salient objects, such as the characters, which make high quality kernel estimation



(a) (b) (c)

Fig. 8. One more example. (a) Blurred image. (b) Our result. (c) Close-ups.

very challenging. The results (shown in (b) and (c)) of two other methods contain several visual artifacts due to imperfect kernel estimation. (d) shows our result without performing edge selection. Compared to the image shown in (e), its quality is lower, indicating the importance of incorporating the gradient mask M in defining the objective function.

Figure 7 shows another example with comparisons with three other blind deconvolution methods. The kernel estimates of Fergus *et al.*[1] and Shan *et al.*[3] are seemingly too sparse, due to the final hard thresholding operation. The restored image is therefore not very sharp. The deblurring result of Cho and Lee [5] contains some noise. Our restored image using Alg. 3. is shown in (e). We have also experimented with several other natural image examples. Figure 8 shows one taken under dim light. More of them are included in our supplementary file downloadable from the project website.

5 Concluding Remarks

We have presented a novel motion deblurring method and have made a number of contributions. We observed that motion deblurring could fail when considerable strong and yet narrow structures exist in the latent image and proposed an effective mask computation algorithm to adaptively select useful edges for kernel estimation. The ISD-based kernel refinement further improves the result quality with adaptive regularization. The final deconvolution step uses a ℓ_1 data term that is robust to noise. It is solved with a new iterative optimization scheme. We have extensively tested our algorithm, and found that it is able to deblur images with very large blur kernels, thanks to the use of the selective edge map.

Acknowledgements

The work described in this paper was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Project No. 413110) and CUHK Direct Grant (No. 2050450).

References

1. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. ACM Trans. Graph. 25, 787–794 (2006)
2. Jia, J.: Single image motion deblurring using transparency. In: CVPR (2007)
3. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. ACM Trans. Graph. 27 (2008)
4. Joshi, N., Szeliski, R., Kriegman, D.J.: Psf estimation using sharp edge prediction. In: CVPR (2008)
5. Cho, S., Lee, S.: Fast motion deblurring. ACM Trans. Graph. 28 (2009)
6. Cai, J.F., Ji, H., Liu, C., Shen, Z.: Blind motion deblurring from a single image using sparse approximation. In: CVPR, pp. 104–111 (2009)
7. Joshi, N.: Enhancing photographs using content-specific image priors. PhD thesis, University of California, San Diego (2008)
8. You, Y., Kaveh, M.: Blind image restoration by anisotropic regularization. IEEE Transactions on Image Processing 8, 396–407 (1999)
9. Chan, T., Wong, C.: Total variation blind deconvolution. IEEE Transactions on Image Processing 7, 370–375 (1998)
10. Osher, S., Rudin, L.: Feature-oriented image enhancement using shock filters. SIAM Journal on Numerical Analysis 27, 919–940 (1990)
11. Alvarez, L., Mazorra, L.: Signal and image restoration using shock filters and anisotropic diffusion. SIAM J. Numer. Anal. 31 (1994)
12. Gilboa, G., Sochen, N.A., Zeevi, Y.Y.: Regularized shock filters and complex diffusion. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 399–413. Springer, Heidelberg (2002)
13. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding and evaluating blind deconvolution algorithms. In: CVPR (2009)
14. Yuan, L., Sun, J., Quan, L., Shum, H.Y.: Progressive inter-scale and intra-scale non-blind image deconvolution. ACM Trans. Graph. 27 (2008)
15. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. ACM Trans. Graph. 26, 70 (2007)
16. Wang, Y., Yang, J., Yin, W., Zhang, Y.: A new alternating minimization algorithm for total variation image reconstruction. SIAM Journal on Imaging Sciences 1, 248–272 (2008)
17. Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-laplacian priors. In: NIPS (2009)
18. Yang, J., Zhang, Y., Yin, W.: An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise. SIAM J. Sci. Comput. 31, 2842–2865 (2009)
19. Bar, L., Sochen, N., Kiryati, N.: Image deblurring in the presence of salt-and-pepper noise. In: International Conference on Scale Space and PDE methods in Computer Vision, pp. 107–118 (2005)
20. Joshi, N., Zitnick, C.L., Szeliski, R., Kriegman, D.J.: Image deblurring and denoising using color priors. In: CVPR, pp. 1550–1557 (2009)
21. Yuan, L., Sun, J., Quan, L., Shum, H.Y.: Image deblurring with blurred/noisy image pairs. ACM Trans. Graph. 26, 1 (2007)
22. Wang, Y., Yin, W.: Compressed Sensing via Iterative Support Detection. CAAM Technical Report TR09-30 (2009)