A variational EM framework with adaptive edge selection for blind motion deblurring

Liuge Yang and Hui Ji
Department of Mathematics, National University of Singapore, Singapore, 119076
yang_liuge@u.nus.edu and matjh@nus.edu.sg

Abstract

Blind motion deblurring is an important problem that receives enduring attention in last decade. Based on the observation that a good intermediate estimate of latent image for estimating motion-blur kernel is not necessarily the one closest to latent image, edge selection has proven itself a very powerful technique for achieving state-of-the-art performance in blind deblurring. This paper presented an interpretation of edge selection/reweighting in terms of variational Bayes inference, and therefore developed a novel variational expectation maximization (VEM) algorithm with built-in adaptive edge selection for blind deblurring. Together with a restart strategy for avoiding undesired local convergence, the proposed VEM method not only has a solid mathematical foundation but also noticeably outperformed the state-of-the-art methods on benchmark datasets.

1. Introduction

Motion blurring is a often-often type of image degradations. When there is a relative motion between the camera and the scene during exposure time, the resulting image will look blurry, known as *motion blurring* in digital photography. For example, one common cause of motion blurring is camera shake during exposure time. The so-called *blind motion deblurring* is then about recovering a clear image with sharp details from an input motion-blurred image.

This paper focuses on uniform motion blurring, i.e., the motion is nearly constant over the image. Uniform motion blurring happens when camera translates along image plane and the scene depth has small variations. Such blurring is often seen when taking a picture on targeted static object using mobile phone with 4X zoom or more. Uniform motion blurring can be modeled as a convolution process:

$$g = k \otimes f + n, \tag{1}$$

where the operator $\otimes$ denotes the discrete convolution operator, g denotes the given blurred image, f denotes the latent clear image, k denotes the unknown blur kernel determined by the motion, and n denotes noise.

Uniform motion deblurring is then about estimating the pair (k, f) from (1), which is an ill-posed problem with many solutions fitting(1) well. For instance, the trivial solution (δ, g) where δ denotes Delta function. To resolve such ambiguities, one has to impose certain priors on both kernel and latent image to constrain the space of feasible solutions. Such prior can be invoked either in the form of regularized variational models or in Bayesian framework, and they often come to the same solution. For example, an Maximum a posteriori (MAP) estimator is to find the maximum of

$$p(f, k|g) \propto p(g|f, k)p(f)p(k),$$

where $p(g|f, k)$ is likelihood function, and $p(f), p(k)$ are statistical priors of f and p. After applying a negative log, such an MAP estimator is equivalent to minimize the following regularized variational model:

$$\min_{f,k} \Phi(g - f \otimes k) + \Psi_1(f) + \Psi_2(k). \tag{2}$$

where $\Phi(\cdot)$ is fidelity term, and $\Psi_1(\cdot), \Psi_2(\cdot)$ regularize clear image and kernel. For example, one empirical statistics of natural images models image gradients as the samples drawn from i.i.d. Laplacian distribution. Then, an MAP estimator from such statistical prior is the same as the total variation, $\Psi_1(f) = \|\nabla f\|_1$, based regularization.

1.1. Discussions

The MAP estimator or the solution from variational model (2) usually takes an iterative procedure that alternatively updates f (or ∇f) and k. The main challenge in such an iterative procedure is how to prevent the sequence converges to suboptimal undesired local minimum or degenerate trivial solution ($k \approx \delta$). There have been an enduring research effort along this line, and many recent works are based on the following observation: the intermediate estimation of f (or ∇f) is for helping more accurate estimation of the kernel k, not the one as close to the truth. Once the estimation of the kernel is finalized, one then estimate an image that is closest to the truth. Thus, the prior from natural image statistics is not necessarily the optimal choice of the image prior used in estimating intermediate results.

One approach is modifying the MAP estimator such that the intermediate estimates of latent image are tuned for better estimating blur kernel. Some methods modify the regular sparsity-prompting norm in regularization methods, *e.g.* normalized ℓ_1-norm based regularization [19] and approximated ℓ_0-norm regularization [43]. Other methods introduce some heuristic procedure to modify the intermediate results such that the resulted one can lead the estimation of kernel toward the right direction. Many strategies have been proposed, including saliency edge/region selection [41, 14, 29] and edge filtering that removes weak details and enhancing salient edges [8].

Another approach is replacing MAP estimator by Variational Bayes (VB) methods [24, 10, 23, 22, 39, 1]. Different from the MAP estimator, VB methods pursue posteriori mean estimates for the kernel k such that the kernel is most likely with respect to the distribution of possible clear images. In addition, as summarized in [39, 18], VB based method will lead to an iterative re-weighting scheme that have the effect of promoting sparsity in image gradient domain. Wipf and Zhang [39] showed that when using Gaussian Scale Mixture (GSM) as the prior on image gradients, the VB based framework can be reformulated as an unconventional MAP framework with a join regularization term that depends on kernel, image gradients and noise level.

Both VB methods and edge selection based MAP estimators have their advantages and disadvantages. VB methods have their merits in several aspects, including rigorous mathematical interpretation, simpler implementation and better stability. However, their experimental performance is not state-of-the-art. Oppositely, edge selection/weighting based MAP methods depends on some heuristic strategy for edge selection/weighting, and some of them are among the top performers. Edge selection based methods tends to perform very well on the images with large blurring degree, but not so on the images with small blurring degree.

1.2. Our Contributions

Motivated by impressive performance gain of edge selection/reweighting for blind image deblurring, this paper aims at developing a mathematical foundation of edge selection/reweighting in the context of blind deblurring, from the viewpoint of VB inference. Thereafter a new edge re-weighting based deblurring method is presented in the framework of the VEM method.

The understanding of edge selection/reweighting in blind deblurring is based on the VEM method that alternatively estimates sharp image gradients and blur kernels. In the framework of the VEM method, instead of viewing the latent variable as the image gradient field of the latent image as existing VB methods, we interpret the latent variable as the image gradient that optimized for better estimation of blur kernel. The key idea of implementing such a latent variable is modeling the latent variable as a set of independent random variables whose standard deviations are regularized by the prior motivated from edge selection/reweighting. The outcome of such an approach leads to a new VEM method with built-in adaptive edge selection.

It is observed that the proposed VEM method sometimes suffers from the convergence to sub-optimal local minima, which indeed is also the issue of most VB methods. In the context of blind deblurring, a restart strategy is proposed for the VEM algorithm for effectively circumventing suboptimal local convergence, especially when the blurring degree is large. Together with the restarting strategy, the proposed VEM method for blind motion deblurring comes with a solid mathematical foundation, good stability to varying image contents, and superior deblurring performance. Extensive experiments on both synthesized dataset and real images showed that the proposed method outperformed most existing methods by a noticeable margin.

2. Related Work

There is abundant literature on blind deblurring using either only one image (*e.g.* [41, 8, 32, 10, 19, 23, 34, 43, 47, 25, 28, 30, 12, 45, 9, 11]) or using multiple images (*e.g.* [4, 31, 7, 46]). We only focus on single image blind motion deblurring that are very relevant to the proposed method.

Regularization methods. In the framework of MAP estimation, many types of regularizations have been developed in the past for blind motion deblurring. For example, Cai *et al.* [3, 5] proposed to regularize clear images by minimizing the ℓ_1-norm of its wavelet/framelet transform. In order to deal with the issue of the bias toward degenerate solution when only using ℓ_1-norm relating regularization, Krishnan *et al.* [19] proposed to replace it by the normalized ℓ_1-norm, i.e. $\|\cdot\|_1/\|\cdot\|_2$ on image gradients. In [43], a new sparsity-prompting function is proposed which approximates ℓ_0-norm of image gradients. For text images, Pan *et al.* [28] proposed to regularize the deblurring process by minimizing ℓ_0-norm of both intensity and image gradients. Instead of using image gradients, Sun *et al.* [34] introduced a set of image patch prior specifically tailored for image edges and corners for blind deblurring. Michaeli and Irani [25] proposed a different prior on image patches that exploits the recurrence of image patches in multiple scales.

Edge processing relating techniques. Based on the idea of processing intermediate estimations of clear image in order to better guide the estimation of blur kernel, Cho and Lee [8] presented a fast deblurring algorithm by iteratively estimating the blur kernel using the images which are the modified version of intermediate recovered images. The modification is done by first running shock filter on the recovered images and then only keep the edges are selected such that its histograms are orientation isotropic. Xu and Jia [41] proposed another edge selection strategy based on the observation that those edges whose span are smaller than the support of the kernel will lead the kernel estimation to the wrong direction. Based on a new metric on edge saliency, a map of salient edges are constructed in [41] to facilitate the estimation of blur kernel. In [29], a different definition of salient structure is proposed. Instead of using salient edges, Hu *et al.* [14] proposed to select image regions for kernel estimation. In [12], Gong *et al.* proposed an adaptive edge selection algorithm by introducing a binary gradient activation vector with sparse cardinality constrain in their optimization problem.

Variational Bayesian Frameworks. It is shown in Levin *et al.* [22] that many naive MAP approaches may fail because they mostly favor trivial solutions. The VB method that maximizing marginalized distributions has been proposed to replace MAP estimator to address this issue; see *e.g.* [10, 22, 23, 39, 1]. Fergus *et al.* [10] modeled images gradients using i.i.d. mixture of zero-mean Gaussians and used Miskin and MacKay's algorithm to iteratively update the estimations of image and kernel. Levin *et al.* [23] also modeled image derivatives using a mixture of zero-mean Gaussians. Different from Fergus *et al.* [10], they introduced a set of i.i.d. hidden variables to indicate the mixture component from which each image gradient arises, and adopted a VEM framework which makes use of the mean field approximation. Babacan *et al.* [1] presented a VB method using super-Gaussian image priors. Wipf and Zhang [39] analyzed both VB method and MAP method. They showed that the underlying cost functions used by VB framework with GSM prior can be reformulated as an unconventional MAP cost function with a joint regularization term depending on kernel, image gradient and the noise level, and the concavity of the regularization on image gradients is adaptively changing during the iterative optimization process.

Deep learning methods. In recent years, many deep learning based approaches have been proposed for blind motion deblurring. See *e.g.* [20, 26, 44, 35, 42, 27, 6, 40, 33].

3. Main Body

3.1. Preliminaries on Variational EM

Consider a probabilistic model involving observed variable y and latent variable z, parameterized by $\theta \in \Theta$. The Maximum Marginal Likelihood estimator of θ is given by

$$\theta^* = \mathrm{argmax}_{\theta\in\Theta}\, p(y;\theta) = \mathrm{argmax}_{\theta\in\Theta} \int p(y,z;\theta)dz.$$

Let q be any probability distribution on z such that $q(z) > 0$. Then by Jensen's inequality, we have the following lower bound of the marginal log-likelihood $\log p(y;\theta)$

$$\log p(y;\theta) \geq \int q(z) \log \frac{p(y,z;\theta)}{q(z)} dz.$$

Define $F(q,\theta) = \int q(z) \log \frac{p(y,z;\theta)}{q(z)} dz$. Instead of directly maximizing the marginal log-likelihood, the expectation maximization (EM) algorithm maximizes the lower bound $F(q,\theta)$:

$$(q^*,\theta^*) = \mathrm{argmax}_{q,\theta\in\Theta}\, F(q,\theta).$$

Compared with the standard EM, variational EM (VEM) method solves the optimization problem above by constraining q inside some family of distributions Q. This optimization problem is solved by alternatively maximizing the function $F(q,\theta)$ between $q(z) \in Q$ and $\theta \in \Theta$. For the t-th iteration,

1. *E-step*. Update $q(z)$ using θ^{t-1}:

$$\begin{aligned} q^t &= \mathrm{argmax}_{q\in Q}\, E_{q(z)}[\log \tfrac{p(z,y;\theta^{t-1})}{q(z)}] \\ &= \mathrm{argmin}_{q\in Q}\, \mathrm{KL}(q(z)||p(z|y;\theta^{t-1})). \end{aligned} \tag{3}$$

2. *M-step*. Update θ using q^t:

$$\theta^t = \mathrm{argmax}_{\theta\in\Theta}\, E_{q^t(z)}[\log p(z,y;\theta)], \tag{4}$$

where Θ denotes the feasible set of parameters, and $\mathrm{KL}(q||p)$ denotes the KL-divergence between q and p. See [2] for more details on EM or VEM

3.2. Problem Formulation in VEM

Estimating kernel in the domain of image gradients is usually more preferred, i.e. the kernel is estimated by

$$\nabla g = k \otimes \nabla f + \nabla n, \tag{5}$$

where $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})^\top$. As argued in [28, 41, 8], not all gradients in ∇g are helpful to kernel estimation. For example, it is shown in [41] that the gradients corresponding to image edges with small span could have negative impact on kernel estimation. Also, as proved in [18, 12], it is possible to get good kernel estimation even if only part of the image gradients are used in the kernel estimation. In other words, an approximation ∇z of the true image gradients ∇f could possibly do better when used for estimating the kernel k.

Let ∇z denote an image gradient field related to ∇f but is better tailored for the estimation of the kernel k. In this paper, we model ∇z as latent random variables drawn from Gaussian distribution with zero mean and invertible diagonal covariance matrix i.e.,

$$p(\nabla z) = \prod_{i=1}^{N} \mathcal{N}((\nabla z)_i | 0, \sigma_i^2),$$

where $\sigma_i \geq \tau$ for a constant τ. Certain prior need to be imposed on the covariance matrix Σ for constraining the space of distribution on ∇z. As observed in [18, 12], image gradients suitable for kernel estimation usually have large magnitude. Such a prior can be encoded as in the s.t.d. of the variable ∇z, since the random variable with zero mean and large s.t.d. is more likely to have the instance with large magnitude, if it has large s.t.d.. Also, only a small percentage of such image gradients should be sufficient for estimating k. Such observations motivates us to propose a cardinality prior on $\{\sigma_i\}_i$:

$$\#\{i : \sigma_i > \tau\} \leq M \quad (M \ll N)$$

where $\#$ denotes the set cardinality and M is a constant $(= \frac{N}{10}$ in our implementation).

In the context of VEM, we can reformulate the kernel estimation of blind deblurring as follows.

- Observed variable: $\nabla g \in R^N$.
- Latent variable: $\nabla z \in R^N$, that follows

$$p(\nabla z) = \prod_{i=1}^{N} \mathcal{N}((\nabla z)_i | 0, \sigma_i^2). \tag{6}$$

  And as usual, $p(\nabla g | \nabla z, k) = \mathcal{N}(\nabla g | k \otimes \nabla z, \tilde{\sigma}^2 \mathrm{I})$, where $\tilde{\sigma}$ is the noise level.
- Parameters: $\theta := [k, \theta_Z] \in \Theta$, where k denotes blur kernel and $\theta_Z := \{\sigma_i\}_{i=1}^N$ are the parameters of the distribution of ∇z. The feasible set Θ of the parameter is defined as

$$\Theta := \{(k, \theta_Z) : \sum_j k[j] = 1, k[j] \geq 0; \\ \sigma_i \geq \tau, \#\{i : \sigma_i > \tau\} \leq M\}. \tag{7}$$

- Variational approximation. We adopt the similar approximation as used by Levin *et al.* [23] such that Q is the set of Gaussian distributions with diagonal covariance matrix. However, different from [23], we restrict the covariance matrix to $\lambda \mathrm{I}$ with predefined constant λ:

$$Q := \{\mathcal{N}(\mu, \lambda \mathrm{I}) : \mu \in R^N\}. \tag{8}$$

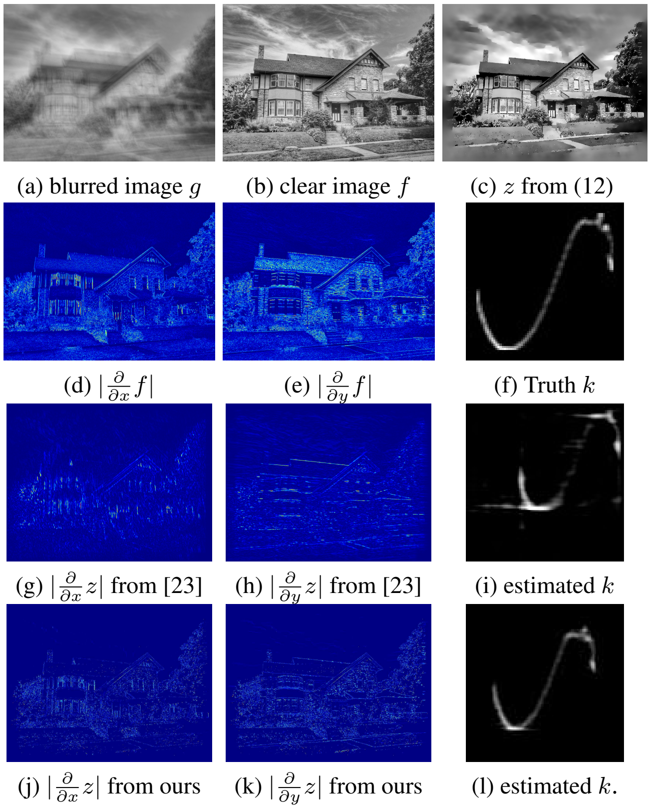
  Such a set is more computationally efficient yet does not decrease performance.

See Fig. 1 for an illustration of the difference between the latent variable ∇z from [23] and that from ours. The variable ∇z from ours are sparser than that from [23] and focus more on edges with large magnitude.

3.3. E-step

Provided an estimate $\theta^{(t)}$, the goal of E-step is to refine the estimation on q by solving

$$\mathrm{argmin}_{q \in Q} \mathrm{KL}(q(\nabla z) || p(\nabla z | \nabla g; \theta^{(t)})). \tag{9}$$


(a) blurred image g (b) clear image f (c) z from (12)

(d) $|\frac{\partial}{\partial x} f|$ (e) $|\frac{\partial}{\partial y} f|$ (f) Truth k

(g) $|\frac{\partial}{\partial x} z|$ from [23] (h) $|\frac{\partial}{\partial y} z|$ from [23] (i) estimated k

(j) $|\frac{\partial}{\partial x} z|$ from ours (k) $|\frac{\partial}{\partial y} z|$ from ours (l) estimated k.

Figure 1: Illustration of latent variable $\nabla z = [\frac{\partial}{\partial x} z, \frac{\partial}{\partial y} z]$. (a–b): input blurred image g and ground truth f; (c): z estimated using (12) of our algorithm; (d)–(f): Image gradient of ground truth image f and ground truth kernel k; (g)–(i): Latent variable ∇z from [23] and the resulting kernel in the last loop; (j)–(l): Latent variable ∇z from the proposed method and the resulting kernel in last loop. The kernel is estimated using (15).

Proposition 1. For $\theta^{(t)} = [k^{(t)}, \Sigma^{(t)}]$, the solution to the optimization problem (9) is

$$q^*(\nabla z) = \mathcal{N}(\nabla z | (\nabla z)^*, \lambda \mathrm{I}), \tag{10}$$

where $(\nabla z)^*$ is the minimizer of the following problem:

$$\min_{\nabla z} \|\nabla g - k^{(t)} \otimes \nabla z\|_2^2 + \tilde{\sigma}^2 \|(\Sigma^{(t)})^{-\frac{1}{2}} \nabla z\|_2^2. \tag{11}$$

Proof. See supplementary materials for the detailed proof. □

In the derivation of E-step above, the variable $(\nabla z)^*$ is estimated by assuming $\frac{\partial z}{\partial x}$ and $\frac{\partial z}{\partial y}$ are independent. Such an assumption ignored the existing correlation between $\frac{\partial z}{\partial x}$ and $\frac{\partial z}{\partial y}$. It is more stable to first estimate z and then calculate ∇z from it. Thus, we implement a modified version of E-step which first estimate z by:

$$\begin{aligned} z^{(t+1)} &= \arg\min_{z[j]\geq 0} \big(\lambda_0 ||g - k^{(t)} \otimes z||_2^2 \\ &+ ||\nabla g - k^{(t)} \otimes \nabla z||_2^2 + \tilde{\sigma}^2 ||(\Sigma^{(t)})^{-\frac{1}{2}} \nabla z||_2^2\big). \end{aligned} \tag{12}$$

Then assign $(\nabla z)^* = \nabla z^{(t+1)} := [\frac{\partial z^{(t+1)}}{\partial x}, \frac{\partial z^{(t+1)}}{\partial y}]^\top$.

3.4. M-step

Given $q^{(t+1)}$, the M-step is to update the estimate of parameters $\theta = [k, \Sigma]$ by solving the optimization problem

$$\theta^{(t+1)} = \text{argmax}_{\theta\in\Theta} E_{q^{(t+1)}(\nabla z)}[\log p(\nabla z, \nabla g; \theta)], \tag{13}$$

where $q^{(t+1)}(\nabla z)$ denote the output from the E-step above. First, we need to calculate the expectation and simplify the above problem. Let $h(\theta) = \log p(\nabla z, \nabla g; \theta)$. Then,

$$\begin{aligned} h(\theta) &= \log p(\nabla g|\nabla z; \theta) + \log p(\nabla z; \theta) \\ &= \log \mathcal{N}(\nabla g | k \otimes \nabla z, \tilde{\sigma}^2 \mathrm{I}) + \log \mathcal{N}(\nabla z | 0, \Sigma) \\ &= \log[\tfrac{1}{(2\pi\tilde{\sigma}^2)^{N/2}} \exp(-\tfrac{||\nabla g - k\otimes\nabla z||^2}{2\tilde{\sigma}^2})] \\ &\quad + \log[\tfrac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}} \exp(-\tfrac{1}{2}||\Sigma^{-\frac{1}{2}}\nabla z||^2)] \\ &= -\tfrac{1}{2\tilde{\sigma}^2}[||\nabla g - k \otimes \nabla z||^2 + \tilde{\sigma}^2 ||\Sigma^{-\frac{1}{2}} \nabla z||^2] \\ &\quad -N\log\tilde{\sigma} - \textstyle\sum_i \log \sigma_i - N\log(2\pi), \end{aligned}$$

where N is the dimensionality of ∇z. Then, by direct calculation, the optimization problem (13) is equivalent to

$$\begin{aligned} &\min_{\theta\in\Theta} \tfrac{1}{2\tilde{\sigma}^2}[||\nabla g - k \otimes \nabla z^{(t+1)}||_2^2 + \tilde{\sigma}^2 ||\Sigma^{-\frac{1}{2}} \nabla z^{(t+1)}||_2^2] \\ &\quad + \textstyle\sum_i \log \sigma_i + \tfrac{\lambda N}{2\tilde{\sigma}^2}||k||_2^2 + \tfrac{\lambda}{2}\sum_i \tfrac{1}{\sigma_i^2}, \end{aligned} \tag{14}$$

where Θ is defined in (7).

The optimization problem above is solved independently for k and Σ as follows. For k, by ignoring the irrelevant terms, we have:

$$k^* = \text{argmin}_{k\in\Theta} ||\nabla g - \nabla z^{(t+1)} \otimes k||_2^2 + \lambda N ||k||_2^2. \tag{15}$$

For Σ, let Λ denote the index set of the M largest entries of $|(\nabla z)_i|$. Then, we have the following:

Proposition 2. The solution to the problem (14) w.r.t. Σ is given by $\Sigma^* = \text{diag}((\sigma_1^*)^2, ..., (\sigma_N^*)^2)$, where

$$\sigma_i^* = \begin{cases} (|(\nabla z^{(t+1)})_i|^2 + \lambda)^{\frac{1}{2}} & \text{if } (|(\nabla z^{(t+1)})_i|^2 + \lambda)^{\frac{1}{2}} > \tau \\ & \quad \text{and } i \in \Lambda, \\ \tau & \text{otherwise.} \end{cases} \tag{16}$$

Proof. See supplementary materials for the detailed proof. □

3.5. Restarting technique for the VEM Method

Although it is well-known that VB based algorithms can effectively avoid trivial solutions ($k = \delta$) [23, 39], empirically we found that local convergence to other suboptimal solutions may occur especially when blurring degree is large. See Fig. 2 (d)–(g), the kernel estimation in the plain version of the proposed VEM method seem to be trapped in some local minima which is quite away from the truth shown in Fig. 2 (c). In other words, the VEM method can avoid trivial solution, but might be trapped in some local minima away from the truth. Such a phenomena might be caused by (1) the highly non-convex nature of the corresponding optimization problem, and (2) the fact that the update of the VEM method only guarantees that $F(q, \theta)$ is not decreasing.

One often used technique for circumventing such issue when solving a highly non-convex problem is to introduce some restarting strategy to allow the iteration jump out of the local maximum point. In this section, we propose a restarting strategy on the estimation of the parameter Σ. Recall that the latent variable ∇z can be viewed as an approximation to ∇f modified for better estimation of blur kernel. Thus, the restart strategy proposed in this paper is to restart the estimation of Σ using the available estimate on ∇f after a number of iterations. Let f^* denote the estimate of f using simple Tikhonov regularization method:

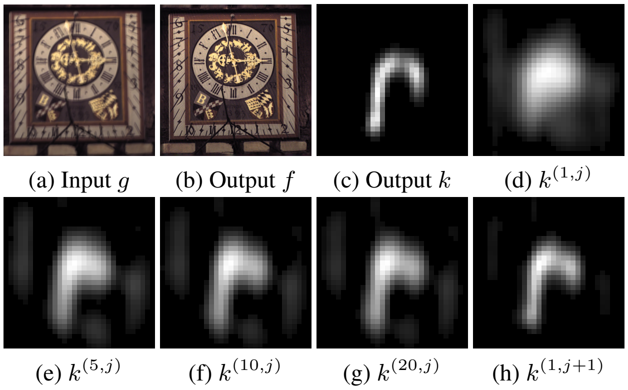
$$f^* = \text{argmin}_f ||g - k^* \otimes f||_2^2 + \lambda_1 ||\nabla f||_2^2. \tag{17}$$

where k^* is the most recent estimate on blur kernel and λ_1 is a constant ($= \frac{1}{400}$ in our implementation). Recall that in the statistical model of ∇z, from Proposition 2 we see that $\sigma_i \geq \tau$ is large only if its corresponding gradient is sufficiently large, and its value is mostly determined by the magnitude of the gradient. Thus, we define the restart of the diagonal Σ^* as follows. Let Λ^* denote the index set of M largest entries of $|\nabla f^*|$. Then, σ_i^* is defined by

$$\sigma_i^* = \begin{cases} |(\nabla f^*)_i| & \text{if } |(\nabla f^*)_i| > \tau \text{ and } i \in \Lambda^*, \\ \tau & \text{otherwise.} \end{cases} \tag{18}$$

Note that (17) uses a spatially uniform natural image prior, so what the restarting process essentially does is to select edges on natural latent image after every few VEM update, instead of continuously select edges from those images get form (12), which may only contain part of the edges as shown in Fig. 1(c).

See Fig. 2 for an illustration of how the restart can be more computationally efficient and guide the kernel estimation toward correct direction. It can be seen that after 5 iterations in inner loop, the estimate on k is nearly unchanged, i.e. $k^{(5,j)}, k^{(10,j)}, k^{(20,j)}$ are all similar. In contrast, if we use restarting after having $k^{(5,j)}$, the restarted estimate $k^{(1,j+1)}$ shown Fig. 2 (h) is clearly much closer to the truth. This indicates the effectiveness of restart. See Fig. 3 (d)—(g) and (h) for an illustration of how during one outer loop, the inner loops update the estimations of ∇z and its covariance matrix Σ. It can be seen that the iterations will yield a more sparse image gradient.



(a) Input g (b) Output f (c) Output k (d) $k^{(1,j)}$

(e) $k^{(5,j)}$ (f) $k^{(10,j)}$ (g) $k^{(20,j)}$ (h) $k^{(1,j+1)}$

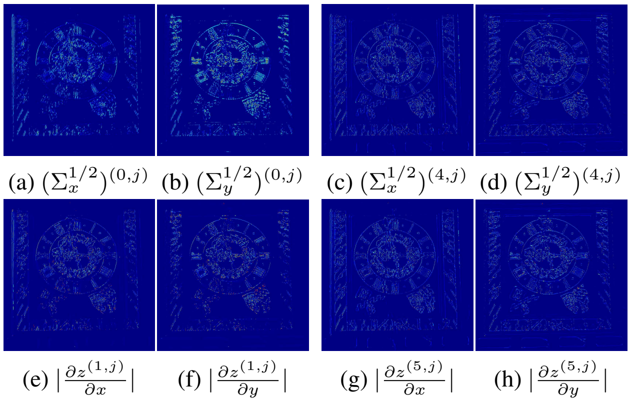
Figure 2: Illustration of how the restart helps the VEM algorithm to avoid local convergence to suboptimal solution. (a)–(c): blurred image g and the final output image f and final output kernel k; (d)–(g): the intermediate estimates of k in one inner loop of VEM; (h): the estimate of k after restarting the VEM only after 5 iterations in inner loop, i.e. the restarted estimate right after $k^{(5,j)}$. In this example $j = 2$. Input image is taken from the dataset in [17].

By including the restarting procedure on Σ in the VEM based alternating iteration, we have a VEM based approach with restart for estimating the motion-blur kernel. After sufficient number of iterations, we have an accurate estimation of the kernel, denoted by k^*. The recovery of clear image f becomes the classic non-blind deconvolution, which solve the linear problem: $g = k^* \otimes f + n$. There are several non-blind deblurring methods optimized for deblurring image using an estimated kernel; see *e.g*. [15, 16, 37]. For fair comparison, we also adopt the deblurring algorithm proposed in [37] which are used in several existing comparative studies on blind motion deblurring. See Algorithm 1 for the outline of the proposed method with restart.

4. Experiment

4.1. Important Implementation Details

In order to deal with large blurs, we adopt the common practice to take a coarse-to-fine estimation scheme, which assumes that the estimation of the kernel in the coarse scale is a good initialization to the kernel in the fine scale. At the coarsest scale, the kernel is initialized using 3×3 Gaussian kernel with $\sigma = \frac{3}{4}$ and call Alg. 1 to estimate the kernel. After that, the estimated kernel is up-sampled using bi-linear interpolation and served as the initialization for the kernel estimation in the finer scale. The set of images with coarse-to-fine scales are generated as follows. Starting with the input image, each image in the coarser scale is constructed by resizing the image in the current scale by half. The number of scales is determined by how many down-sampling is needed to resize the maximum kernel size down to 3×3.



(a) $(\Sigma_x^{1/2})^{(0,j)}$ (b) $(\Sigma_y^{1/2})^{(0,j)}$ (c) $(\Sigma_x^{1/2})^{(4,j)}$ (d) $(\Sigma_y^{1/2})^{(4,j)}$

(e) $|\frac{\partial z^{(1,j)}}{\partial x}|$ (f) $|\frac{\partial z^{(1,j)}}{\partial y}|$ (g) $|\frac{\partial z^{(5,j)}}{\partial x}|$ (h) $|\frac{\partial z^{(5,j)}}{\partial y}|$

Figure 3: Illustration of the updates of variable $|\nabla z|$ and its $\{\sigma_i\}_i$ in outer Iteration $j = 2$. (a)–(b): $\Sigma^{1/2}$ used in first iteration, (c)–(d): $\Sigma^{1/2}$ used in last iteration (5-th iteration); (e)–(h): the resulting estimate of ∇z using (12).

Algorithm 1 Outline of the VEM method

```
INPUT: blurred image $g$
OUTPUT: sharp image $f^*$, blur kernel $k^*$
%%% kernel estimation
Initialization: set initial kernel $k^{(0,0)}$.
for $j = 1, 2, \ldots, m$ do
    Restarting: define $\Sigma^{(0,j)}$ as described in Sec. 3.5.
    for $t = 1, 2, \ldots, n$ do
        E-step: define $\nabla z^{(t,j)}$ by (12)
        M-step: define $k^{(t,j)}$ by (15)
                define $\Sigma^{(t,j)}$ by (16)
    end for
    Set $k^{(0,j+1)} := k^{(n,j)}$.
end for
Set $k^* := k^{(n,m)}$.
%%% non-blind deblurring using $k^*$
Estimate $f^*$ using the method in [37].
```

For the experiments on the tested datasets, the image gradients are calculated using the difference operator $[-1, 1]$ and $[-1, 1]^\top$. The parameters are set uniformly as follows. The number of inner iterations $n = 5$, and the number of outer iterations $m = 12$. The constant M for cardinality constraint is set to be $\frac{N}{10}$ for both horizontal and vertical gradients, where N denotes the number of image pixels. For other parameters, $\tilde{\sigma} = 10^{-\frac{5}{2}}$, $\tau = 10^{-3}$ and $\lambda = 0.001/N$. The last step in Algorithm 1 calls the non-blind routine in [37] with iteration number= 100. (12), (17) and (15) are all quadratic programming problems. In our implementation, they are simply solved by first using Conjugate Gradient (CG) method as unconstrained problems, followed by a projection to their feasible sets.

	Fergus	Cho	Xu-10	Krishinan	Levin	Sun	Xu-13	Zhang	Zhong	Michaeli	Pan	Perron	Nah	Ours
	[10]	[8]	[41]	[19]	[23]	[34]	[43]	[46]	[47]	[25]	[28]	[30]	[26]	
man-made	14.10	16.11	19.56	15.67	18.02	19.30	17.87	16.93	17.32	17.32	17.33	17.53	15.63	**19.99**
natural	16.44	20.09	23.38	19.24	20.93	23.69	22.14	21.38	21.07	20.66	21.47	22.08	18.45	**24.33**
people	18.46	19.89	26.50	21.34	22.95	26.13	25.72	24.58	24.39	24.20	24.33	24.04	20.58	**27.22**
saturated	12.73	14.23	15.59	14.11	14.81	14.95	15.00	14.92	14.86	14.30	15.11	13.89	14.46	**17.04**
text	13.65	14.82	19.68	15.11	15.80	18.35	18.61	16.11	15.86	15.22	17.56	16.80	14.21	**20.35**
average	15.09	17.03	20.97	17.09	18.50	20.48	19.87	18.78	18.70	18.34	19.16	18.87	16.67	**21.79**

Table 1: Quantitative comparison on the synthetic uniform dataset in [21]. Performance is measured in average PSNR value. Different row denotes different category of images. The last row is the average PSNR value over the whole dataset.

	Whyte *et al.*	Hirsch *et al.*	Shan *et al.*	Krishnan *et al.*	Cho and Lee	Xu and Jia	Yue *et al.*	Gong *et al.*	Ours
	[38]	[13]	[32]	[19]	[8]	[41]	[45]	[12]	
Image 1	27.5475	26.7232	26.4253	26.8654	28.9093	29.4054	30.1340	30.3572	**31.7060**
Image 2	22.8696	22.5867	20.5950	21.7551	24.2727	25.4793	25.4749	25.5210	**26.3540**
Image 3	28.6112	26.4155	25.8819	26.6443	29.1973	29.3040	30.1777	**31.6577**	31.0048
Image 4	24.7065	23.5364	22.3954	22.8701	26.6064	26.7601	26.7661	27.4804	**27.9150**
Total Avg.	25.9337	24.8155	23.8244	24.5337	27.2464	27.7372	28.1158	28.7541	**29.2249**

Table 2: Quantitative comparison on Köhler dataset [17]. Performance is measured in average PSNR value.

4.2. Quantitative Evaluation

Synthetic dataset from Lai *et al.* [21]. In order to test our method on different types of images with different sizes of blurs, we adopt the recent benchmark dataset built by Lai *et al.* [21], which contains 100 blurry images divided into 5 categories. They are synthesized by using 4 different kernels with size ranging from 51×51 to 101×101, adding 1% Gaussian noise. Except the results from the deep learning method [26], the results of all other methods in Table 1 are obtained from [21]. We first downloaded the estimated kernels published online by [21], and then ran the same non-blind deblurring algorithm, Whyte *et al.* [37] with their code published online, to get the deblurred images for comparison. The results of [26] is obtained by using the trained model published by the authors.

See Table 1 for the comparison of the methods in terms of average PSNR value. It can be seen that our methods outperformed other methods in all categories, especially on the category of ”people” and saturated”, which indeed possess special characters of image edges. This shows the adaptive edge selection in the proposed framework is more robust than existing edge selection techniques *e.g*. [41, 8]. The comparison on SSIM [36] and the demonstration of some examples can be found in supplementary materials.

Köhler dataset [17]. We also tested the proposed method on those images whose motion blurring is not exactly uniform. Köhler dataset [17] is used for testing, which contains 48 real blurry images generated by convolving 4 latent sharp images with 12 blur kernels whose sizes range from 41×41 to 141×141. This dataset is generated by recording the samplings of the six dimensional camera motion. See Table 2 for the comparison in terms of PSNR value. The results of other methods are directly quoted [12, 45]. It can be seen that the proposed method overall noticeably outperform the compared methods.

4.3. Experiments on Real Images

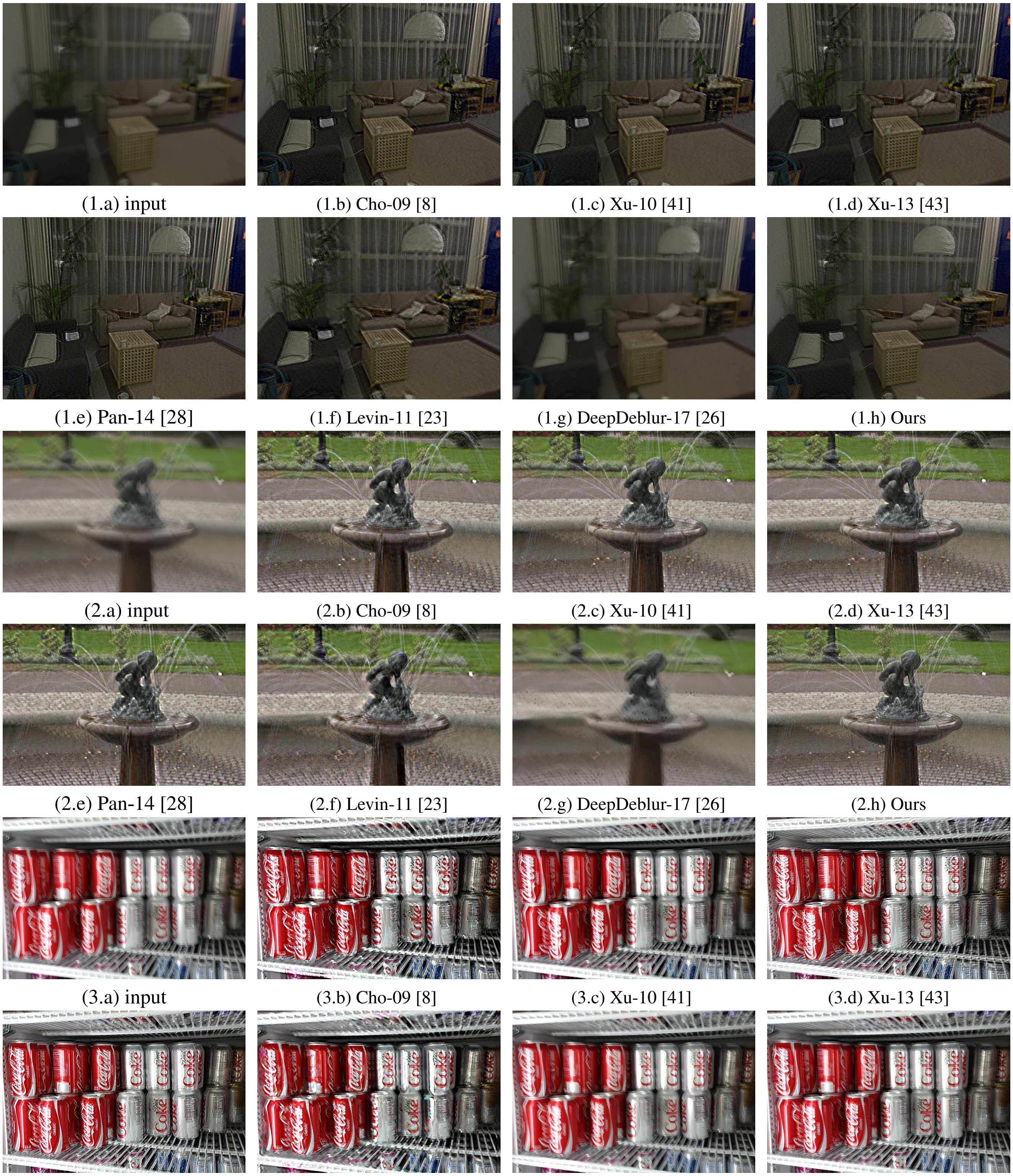
The proposed algorithm is also evaluated on real images summarized by Lai *et al*. [21]. We compared ours to the six representative methods with top performance, including two edge selection related methods: Cho *et al*. [8], Xu and Jia [41], two ℓ_0-norm regularization methods: Xu *et al*. [43] and Pan *et al*. [28], one VEM method: Levin *et al*. [23], and one deep learning method: [26]. The introduction to these methods can be found in Section 2. See Fig. 4 for visual inspection of the results. It can be seen that the results from the proposed method in general produce the results with better visual quality. The illustration of more results can be found in supplementary materials.

5. Conclusion

This paper revisited the powerful edge selection/reweighting technique used in blind motion deblurring from the perspective of VB inference. By building a mathematical foundation on edge selection, we develop a VEM method with strong motivation from edge selection/reweighting for blind motion deblurring. Together with a restart strategy, the proposed VEM method is easy to implement, stable to varying content, and provide state-of-the-art performance.

Acknowledgement.

This work was partially supported by Singapore MOE AcRF Grant MOE2017-T2-2-156.



(1.a) input (1.b) Cho-09 [8] (1.c) Xu-10 [41] (1.d) Xu-13 [43]

(1.e) Pan-14 [28] (1.f) Levin-11 [23] (1.g) DeepDeblur-17 [26] (1.h) Ours

(2.a) input (2.b) Cho-09 [8] (2.c) Xu-10 [41] (2.d) Xu-13 [43]

(2.e) Pan-14 [28] (2.f) Levin-11 [23] (2.g) DeepDeblur-17 [26] (2.h) Ours

(3.a) input (3.b) Cho-09 [8] (3.c) Xu-10 [41] (3.d) Xu-13 [43]

(3.e) Pan-14 [28] (3.f) Levin-11 [23] (3.g) DeepDeblur-17 [26] (3.h) Ours

Figure 4: Visual comparison of the results from different methods. Zoom-in for easier inspection

References

[1] S. D. Babacan, R. Molina, M. N. Do, and A. K. Katsaggelos. Bayesian blind deconvolution with general sparse image priors. In *ECCV*, pages 341–355, 2012.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. New York: Springer, 2007.

[3] J.-F. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring from a single image using sparse approximation. In *CVPR*, pages 104–111. IEEE, 2009.

[4] J.-F. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring using multiple images. *J. Comput. Physics*, 228(14):5057–5071, 2009.

[5] J.-F. Cai, H. Ji, C. Liu, and Z. Shen. Framelet-based blind motion deblurring from a single image. *IEEE Trans. Image Process.*, 21(2):562–572, 2012.

[6] A. Chakrabarti. A neural approach to blind motion deblurring. In *ECCV*, 2016.

[7] J. Chen, L. Yuan, C.-K. Tang, and L. Quan. Robust dual motion deblurring. In *CVPR*, 2008.

[8] S. Cho and S. Lee. Fast motion deblurring. *ACM TOG (Proc. SIG- GRAPH Asia)*, 28(5):145:1–145:8, 2009.

[9] T. S. Cho, S. Paris, B. K. P. Horn, and W. T. Freeman. Blur kernel estimation using the radon transform. In *CVPR*, pages 241–248, 2011.

[10] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM TOG (Proc. SIGGRAPH)*, 25(3):787–794, 2006.

[11] A. Goldstein and R. Fattal. Blur-kernel estimation from spectral irregularities. In *ECCV*, 2012.

[12] D. Gong, M. Tan, Y. Zhang, A. v. d. Hengel, and Q. Shi. Blind image deconvolution by automatic gradient activation. In *CVPR*, pages 1827–1836, June 2016.

[13] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf. Fast removal of non-uniform camera shake. In *ICCV*, pages 463–470. IEEE, 2011.

[14] Z. Hu and M.-H. Yang. Good regions to deblur. In *ECCV*, pages 59–72. Springer, 2012.

[15] H. Ji and K. Wang. Robust image deblurring with an inaccurate blur kernel. *IEEE Trans. Image Process.*, 21(4):1624–1634, 2012.

[16] H. Ji and K. Wang. A two-stage approach to blind spatially-varying motion deblurring. In *CVPR*, pages 73–80. IEEE, 2012.

[17] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In *ECCV*, pages 27–40, 2012.

[18] D. Krishnan, J. Bruna, and R. Fergus. Blind deconvolution with re-weighted sparsity promotion. *ArXiv e-prints*, 2013.

[19] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR*, 2011.

[20] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. *ArXiv e-prints*, 2017.

[21] W.-S. Lai, J.-B. Huang, Z. Hu, N. Ahuja, and M.-H. Yang. A comparative study for single image blind deblurring. In *CVPR*, 2016.

[22] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, pages 1964–1971, 2009.

[23] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *CVPR*, 2011.

[24] A. C. Likas and N. P. Galatsanos. A variational approach for bayesian blind image deconvolution. *IEEE Trans. Sig. Proc.*, 52(8):2222–2233, 2004.

[25] T. Michaeli and M. Irani. Blind deblurring using internal patch recurrence. In *ECCV*, 2014.

[26] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, July 2017.

[27] M. Noroozi, P. Chandramouli, and P. Favaro. Motion deblurring in the wild. In *GCPR*, 2017.

[28] J. Pan, Z. Hu, Z. Su, and M.-H. Yang. Deblurring text images via ℓ_0-regularized intensity and gradient prior. In *CVPR*, 2014.

[29] J. Pan, R. Liu, Z. Su, and X. Gu. Kernel estimation from salient structure for robust motion deblurring. *Signal Processing: Image Communication*, 28(9):1156–1170, 2013.

[30] D. Perrone and P. Favaro. Total variation blind deconvolution: The devil is in the details. In *CVPR*, 2014.

[31] A. Rav-Acha and S. Peleg. Two motion blurred images are better than one. *Pattern Recognition Letters*, 26:311–317, 2005.

[32] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Trans. Graph*, 27(3):73, 2008.

[33] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang. Deep video deblurring for hand-held cameras. In *CVPR*, pages 1279–1288, 2017.

[34] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *ICCP*, 2013.

[35] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, 2018.

[36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.*, 13:600 – 612, 05 2004.

[37] O. Whyte, J. Sivic, and A. Zisserman. Deblurring shaken and partially saturated images. *IJCV*, 110(2):185–201, 2014.

[38] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *IJCV*, 98(2):168–186, 2012.

[39] D. Wipf and H. Zhang. Revisiting bayesian blind deconvolution. *J. Mach. Learn. Res.*, 15:3775–3814, 2014.

[40] L. Xiao, J. Wang, W. Heidrich, and M. Hirsch. Learning high-order filters for efficient blind deconvolution of document photographs. In *ECCV*, 2016.

[41] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *ECCV*, 2010.

[42] L. Xu, J. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. *NIPS*, 2:1790–1798, 01 2014.

[43] L. Xu, S. Zheng, and J. Jia. Unnatural l_0 sparse representation for natural image deblurring. In *CVPR*, 2013.

[44] R. Yan and L. Shao. Blind image blur estimation via deep learning. *IEEE Trans. Image Process.*, 25(4), 2016.

[45] T. Yue, S. Cho, J. Wang, and Q. Dai. Hybrid image deblurring by fusing edge and power spectrum information. In *ECCV*, 2014.

[46] H. Zhang, D. Wipf, and Y. Zhang. Multi-image blind deblurring using a coupled adaptive sparse prior. In *CVPR*, 2013.

[47] L. Zhong, S. Cho, D. Metaxas, S. Paris, and J. Wang. Handling noise in single image deblurring using directional filters. In *CVPR*, 2013.