Normalized Blind Deconvolution

Meiguang Jin[1][0000−0003−3796−2310], Stefan Roth[2][0000−0001−9002−9832], and
Paolo Favaro[1][0000−0003−3546−8247]

[1] University of Bern, Switzerland
[2] TU Darmstadt, Germany
{jin, paolo.favaro}@inf.unibe.ch
stefan.roth@visinf.tu-darmstadt.de

Abstract. We introduce a family of novel approaches to single-image blind deconvolution, *i.e.*, the problem of recovering a sharp image and a blur kernel from a single blurry input. This problem is highly ill-posed, because infinite (image, blur) pairs produce the same blurry image. Most research effort has been devoted to the design of priors for natural images and blur kernels, which can drastically prune the set of possible solutions. Unfortunately, these priors are usually not sufficient to favor the sharp solution. In this paper we address this issue by looking at a much less studied aspect: the relative scale ambiguity between the sharp image and the blur. Most prior work eliminates this ambiguity by fixing the L^1 norm of the blur kernel. In principle, however, this choice is arbitrary. We show that a careful design of the blur normalization yields a blind deconvolution formulation with remarkable accuracy and robustness to noise. Specifically, we show that using the Frobenius norm to fix the scale ambiguity enables convex image priors, such as the total variation, to achieve state-of-the-art results on both synthetic and real datasets.

1 Introduction

The removal of blur in images has seen substantial progress in the past few decades with a wealth of efficient algorithms available today [7, 24, 36]. Advances have been made by exploring different energy formulations [5, 24, 44] as well as image priors [28, 39, 43, 45, 52] in both Bayesian [2, 9, 23, 47] and deterministic frameworks [7, 30, 31]. Advanced image formation models [14, 40, 48] allow going beyond stationary blur, and, recently, a variety of deep learning approaches [4, 15, 16, 25, 26, 35, 38, 41] have been proposed.

In its simplest form, blind deconvolution describes a blurry image y as the convolution between a latent blur k and a latent sharp image x. Because of the convolutional image formation model, there is an inherent scale ambiguity between these two unknowns. That is, one could multiply the blur k by a scale factor $s > 0$ and the image x by the reciprocal factor $1/s$ to obtain the same blurry image y. To remove this ambiguity, it is common to impose that $|k|_1 = 1$. However, since the scale is arbitrary, so is this choice of normalization.[3] Indeed,

[3] Blind deconvolution is a mathematical problem with a corresponding physical problem (image deblurring). From the mathematical point of view, there is an ambiguous

we could use any other norm in principle, such as the L^2 or Frobenius norm, and require $|k|_2 = 1$. More generally, we could apply an arbitrary p-norm and constrain $|k|_p = 1$, where $|\cdot|_p$ denotes the L^p norm. To the best of our knowledge, very little attention has been paid to the scale ambiguity between blur and image. We show that this choice matters much more than has been apparent, as it can significantly affect the performance of blind deconvolution algorithms. As we demonstrate later on, a more appropriate normalization can enable simple, convex image priors, such as total variation, reach state-of-the-art (SotA) image quality, which was previously possible only with more complex, non-convex priors [29, 46]. Moreover, our scheme provably allows avoiding many difficulties that have hampered the practical implementation of SotA methods. This includes hand-tuned schedules for varying the amount of regularization across the iterations, approximating operators [29], or gradient steps.

Contributions: We would like to emphasize that we neither propose a novel image prior nor a novel blur prior. Rather, we *(1)* introduce a formulation of blind deconvolution with a novel, general **constraint** to fix the **scale ambiguity**; *(2)* provide a mathematical proof that shows for the first time a condition under which the sharp image is preferred over the blurry input image **even with classic convex image priors**;[4] *(3)* show that the proposed scale constraint **automatically** changes the amount of image regularization **across iterations**, avoiding custom-designed tuning typically used in most algorithms; *(4)* introduce two new algorithms that achieve **SotA** in terms of both accuracy and robustness.

2 Prior Work

Although blind deconvolution has been studied for more than two decades, it still remains a challenging task due to its ill-posed nature. Therefore, most methods for solving this problem differ in their regularization technique.

A common practice is to employ a regularization term for the sharp image x that encourages the sparsity of its gradients. This follows from the study of natural image statistics [37]. As a convex sparsity-inducing prior, the total variation (TV), introduced by Rudin, Osher, and Fatemi [32] for denoising, has emerged as an effective choice also in blind deconvolution [5]. Despite the success of the TV prior in practice, Levin *et al.* [22] show that it favors blurry solutions. Perrone

scale between the blur kernel and the sharp image, and L^1 normalization, as with any other arbitrary norm, is one possible choice to fix the scale of the blur kernel. However, as a model of a physical system, the blur kernel corresponds to the point spread function (PSF) of the camera lens(es), and in this case physics indicates that normalization is through the L^1 norm [3]. Therefore, we first solve the mathematical problem with an arbitrary norm, and then we map the final solution to a physically valid PSF by normalizing it in terms of L^1. This way we benefit from the mathematical freedom while ensuring physical validity in the end.

[4] Please note that prior work showed how convex image priors together with an L^1 constraint on the kernel *do not favor* the sharp image as a solution [22, 30].

and Favaro [30] find that post normalization in the alternating minimization algorithm is responsible for the empirical success. Cho and Lee [8] show that under certain conditions, the MAP energy function can favor the correct solution. Beyond convex image priors, a wide variety of effective, non-convex methods has been proposed [7, 36, 44]. Wipf and Zhang [42] re-examine the common belief that the regularization term should encourage natural image statistics. They argue, instead, that the image prior should simply focus on discriminating between sharp and blurry images. Ideally this would be achieved with an L^0 regularizer, but the resulting objective is NP-hard. As a substitute, they suggest a logarithmic prior, which Babacan *et al.* [2] adopt with success. Different L^0 norm approximations have also been proposed [20, 28, 45]. Patch-based methods [24, 39] sidestep classical regularizers and have shown impressive performance. However, they are usually computationally demanding due to searching for similar patches across scales [24] or finding sharp patches in an external dictionary [39].

Recently, due to the astounding success of deep learning, neural network-based motion deblurring approaches have been proposed [4, 35]. [4] estimates Fourier coefficients of a deconvolutional filter to be applied to an input patch and shows remarkable reconstruction quality and speed. The main limitation of current learning-based approaches is that they do not generalize well to large blurs. Conventional approaches have remained superior in this regard. In this paper we also provide a theoretical result, hence focus on the simplest formulation of blind deblurring, where a blurry image is the result of a convolution. While already allowing practical applications, real blur can be much more complex. Great progress has been made on handling real blur, such as for camera shake [9, 13, 38, 40], dynamic scenes [16–18, 25, 26], or object motion [11, 27, 34].

So far most approaches have focused on the design of image priors. Still, a variety of methods have also considered blur priors based on an L^1 or L^2 penalty [4, 6, 7, 20, 24, 39, 44, 45] to either encourage the estimated blur kernel to be sparse or to discourage the trivial solution (*i.e.*, the estimated sharp image equals the blurry input). The recent work of Zhang *et al.* [49] is most closely related to ours. They also impose a unit Frobenius norm constraint on the blur kernel. Their analysis requires an explicit solution of the latent image given the blur kernel, which is challenging to obtain and requires approximations (*e.g.*, blur sparsity). In contrast, our analysis does not require an explicit solution; we present a novel proof for the TV prior that considers a family of normalizations, and we include a positivity constraint on the blur. Finally, we show that the scale ambiguity can be fixed so that even simple and efficient convex image priors yield SotA results outperforming [49], while provably avoiding the trivial solution.

3 Blind Deconvolution

A classic way to cast blind deconvolution [5] is to optimize

$$\min_{k,x} |y - k * x|_2^2 + \lambda |\nabla x|_2 \qquad \text{subject to } |k|_1 = 1, \quad k \geq 0, \tag{1}$$

where $\lambda > 0$ is a regularization parameter, $k \geq 0$ enforces element-wise nonnegativity, and $|\nabla x|_2 \triangleq \sum_{i,j} |\nabla x_{ij}|_2$ denotes the discretized TV [32]. This formulation appeared with different modifications in several works [5, 30, 44]. One of its favorable properties is that the problem is convex in the blur k when the sharp image x is fixed and, vice versa, it is convex in x when fixing k. To see this, note that the constraints $|k|_1 = 1$ and $k \geq 0$ are equivalent to $\sum_i k_i = 1$ and $k \geq 0$. Thus, convergence analysis and several computationally efficient algorithms, such as alternating minimization, are available for this family of problems [12]. Unfortunately, solving the above formulation **globally** is known not to work [22, 30], *i.e.*, it fails at recovering a sharp image. Specifically, the degenerate solution $(k{=}\delta,\ x{=}y)$ yields a smaller energy than the true solution.

We now show that introducing a different scale normalization constraint allows transforming such a formulation into one that succeeds in recovering a sharp latent image. We then introduce two algorithms: one is used to study different scale normalizations, while the other is computationally efficient. Both achieve SotA results on standard synthetic and real datasets.

3.1 L^p scale normalization constraint

As noted above, fixing the scale constraint via $|k|_1 = 1$ is an arbitrary choice due to the inherent scale ambiguity of the blind deblurring problem. Thus, let us consider the more general scale normalization $|k|_p = 1$, where $|\cdot|_p$ denotes the L^p norm. Our formulation with the L^p constraint then becomes

$$\min_{w,z} |y - w * z|_2^2 + \lambda |\nabla z|_2 \qquad \text{subject to } |w|_p = 1, \quad w \geq 0, \tag{2}$$

where for notational distinction from Problem (1), we have used w and z to denote the blur kernel and latent image, respectively. This formulation is not desirable as is, because it is not convex when solving for the blur kernel w due to the L^p constraint. However, we now show how to transform this formulation so that all alternating minimization steps involve convex optimization. First, let us relate the pair (k, x) to (w, z) via $k \triangleq {}^{w}\!/\!_{|w|_1}$ and $x \triangleq |w|_1 z$. Thus, $w{*}z = k{*}x$ and $|k|_p = {}^{|w|_p}\!/\!_{|w|_1}$. With the constraints on w in Problem (2) we have $|k|_p = {}^{1}\!/\!_{|w|_1}$. The above definitions plus constraints on w in Problem (2) are then equivalent to the new constraints $k \geq 0$, $|k|_1 = 1$, $w = {}^{k}\!/\!_{|k|_p}$, $z = |k|_p x$. Since the relations between k and w and between x and z are 1-to-1, we obtain

$$\min_{k,x} |y - k * x|_2^2 + \lambda |k|_p |\nabla x|_2 \qquad \text{subject to } |k|_1 = 1, \quad k \geq 0, \tag{3}$$

which is now convex in k for $p \geq 1$ and a fixed x.

Remark 1. Problem (3) is almost identical to the classic formulation of Problem (1), except for the **modified regularization term** $\lambda |k|_p |\nabla x|_2$. The weight λ is now scaled by the L^p norm of the blur k. When the blur k is close to a Dirac δ, the regularization will be the highest, and when the blur k is close to uniform, the regularization will be the lowest. It is thus clear that the proposed normalization is not equivalent to the classic L^1 case when $p > 1$.

Remark 2. We have transformed Problem (2) with the mappings $|k|_p = {}^{|w|_p}\!/\!_{|w|_1}$ and $x \triangleq |w|_1 z$ into Problem (3). Thus, the latent blur k in Problem (3) is always estimated as a valid PSF as in Problem (1).

The first question is then whether a choice of $p > 1$ brings any improvements to the set of solutions in the blind deconvolution formulation above. In the following proposition we show that this is indeed the case for $p \geq 2$.

Proposition 1. *Assume the gradients of the true sharp image x to be i.i.d. zero-mean Gaussian and the true blur kernel k to have finite support. Given a blurry image $y = k * x$ with the true blur k, Problem (3) then favors with high probability the true blur/image pair (k, x) over the trivial no-blur pair (δ, y) for $p \geq 2$.*

Proof. Both solutions make the data term $|y - k * x|_2^2 = 0$ and satisfy the constraints. The only term left in the objective function is therefore the joint prior $|k|_p |\nabla x|_2$ (note that the regularization parameter $\lambda > 0$ can be ignored). Therefore, we need to show that $|\delta|_p |\nabla y|_2 \geq |k|_p |\nabla x|_2$. **Blur term $|k|_p$:** This term can be left as is. **Blur term $|\delta|_p$:** We have that $|\delta|_p = 1$, $\forall p \in Z$. **Sharp image prior $|\nabla x|_2$:** Let us define $\nabla x_{ij} \doteq [\mathbf{u}_{ij}\ \mathbf{v}_{ij}]^\top$. Because of the assumptions on the gradient of a sharp image, we have that $\mathbf{u}_{ij}, \mathbf{v}_{ij} \sim \mathcal{N}(0, \sigma^2)$. Then, we obtain $|\nabla x_{ij}|_2 = \sqrt{\mathbf{u}_{ij}^2 + \mathbf{v}_{ij}^2} \sim \mathcal{X}_2$, where $\mathcal{X}_2$ denotes the Chi-squared distribution with two degrees of freedom. Its mean is $\mu_{\mathcal{X}} = \sqrt{\pi/2}\sigma$ and its variance $\sigma_{\mathcal{X}}^2 = (2 - \pi/2)\sigma^2$. Note that, because there are only two degrees of freedom and the Gaussian variables have zero mean, this is also equivalent to the Rayleigh distribution. Since we need to evaluate $|\nabla x|_2 \doteq \sum_{i=1}^N \sum_{j=1}^M |\nabla x_{ij}|_2$, we then need to compute the sum of MN independent Chi-squared (or Rayleigh) random variables. By using Chebyshev's inequality, we can write for any $\xi > 0$ $P\left(|{}^1/_{MN}|\nabla x|_2 - \mu_{\mathcal{X}}| < \xi\right) \geq 1 - {}^{\sigma_{\mathcal{X}}^2}/_{MN\xi^2}$. Therefore, for a sufficiently large MN the approximation

$$|\nabla x|_2 \simeq MN\sqrt{\tfrac{\pi}{2}}\sigma \tag{4}$$

will hold with very high probability.

Blurry image prior $|\nabla y|_2$: Let us define $|\nabla y_{ij}|_2 \doteq \sqrt{(k * \mathbf{u})_{i,j}^2 + (k * \mathbf{v})_{i,j}^2}$ so that $|\nabla y|_2 = \sum_{i,j} |\nabla y_{ij}|_2$. One can see that each $(k * \mathbf{u})_{i,j} \doteq \sum_{m,n} k_{m,n} \mathbf{u}_{i-m,j-n}$ is a zero-mean Gaussian with variance $\sigma^2 |k|_2^2$. Thus, we also obtain that $|\nabla y_{ij}|_2$ is a Chi-squared distributed random variable, but with mean $\hat{\mu} = \sqrt{\pi/2}\sigma|k|_2$ and variance $\hat{\sigma}^2 = (2 - \pi/2)\sigma^2|k|_2^2$. The sum over the pixels (i, j) now needs additional care, because neighboring terms may not be independent. We use the assumption of a finite $W \times H$ support of k. Thus, we know that $|\nabla y_{ij}|_2$ is independent from $|\nabla y_{i+|\Omega|,j}|_2$. This suggests that we split the sum so that $\sum_{i,j} |\nabla y_{ij}|_2 = \sum_{p=1}^W \sum_{q=1}^H \sum_{i,j} |\nabla y_{p+Wi,q+Hj}|_2$. Then, by using the approximation of Eq. (4) we have $\sum_{i,j} |\nabla y_{p+Wi,q+Hj}|_2 \simeq \frac{MN}{WH}\sqrt{\frac{\pi}{2}}\sigma|k|_2$ and finally $|\nabla y|_2 \simeq \sum_{p=1}^W \sum_{q=1}^H \frac{MN}{WH}\sqrt{\frac{\pi}{2}}\sigma|k|_2 = MN\sqrt{\frac{\pi}{2}}\sigma|k|_2$. By putting all together, we have that $|\delta|_p|\nabla y|_2 \simeq MN\sqrt{\frac{\pi}{2}}\sigma|k|_2 \geq MN\sqrt{\frac{\pi}{2}}\sigma|k|_p \simeq |k|_p|\nabla x|_2$.This boils down to $|k|_2 \geq |k|_p$, which is true for $p \geq 2$. We can therefore conclude that $|\nabla y|_2 \geq |k|_2|\nabla x|_2 \geq |k|_p|\nabla x|_2$ for any $p \geq 2$ with high probability when MN is large enough. □

This result shows that by changing the scale normalization constraint, we have transformed Problem (1) with a trivial global solution into Problem (3), where the trivial solution is no longer preferred over the true one. Note that the optimal solution will be given as a trade-off between data term and image prior. To fully assess the impact of our scale normalization constraint, we propose two algorithms to minimize Problem (3) and test them on standard datasets.

3.2 Frank-Wolfe (`FW`) algorithm

Because of Proposition 1, we can now attempt to solve Problem (3) **globally**. We propose to use an alternating minimization method and run each alternating step until convergence. By denoting the iteration index as $t \geq 0$, we first update the latent image via

$$x^t = \arg\min_x |y - k^t * x|_2^2 + \lambda |k^t|_p |\nabla x|_2 \tag{5}$$

with the L-BFGS-B algorithm [50] with box constraints (imposing $x \geq 0$ and $x \leq 1$ at every pixel). Then, we solve

$$k^{t+1} = \arg\min_k |y - k * x^t|_2^2 + \lambda |k|_p |\nabla x^t|_2 \qquad \text{subject to } |k|_1 = 1, \quad k \geq 0 \tag{6}$$

with the Frank-Wolfe algorithm (`FW`) [10]. The algorithm is initialized with $k^0 = \delta$. We are not aware of any prior blind deconvolution method that solves each step of the alternating minimization problem above, particularly Eq. (6), **without approximations**. Note that in the `FW` algorithm we do not adapt the regularization weight λ with the iteration index t. We have not found prior methods capable of converging to the correct kernel when the latent blur is initialized with the Dirac δ and the regularization parameter is fixed. This is indeed not possible if the global minimum of the objective is the degenerate solution $(k = \delta, x = y)$. Note that changing the regularization during the iterations in a heuristic way means changing the original cost function that one intends to minimize.

3.3 Post-normalization (`PN`) algorithm

As we will see in the experiments, the `FW` algorithm tends to converge slowly (although still faster than many other SotA methods). As a more efficient implementation, we consider the method of Chan and Wong [5, 30]. We thus derive an alternating gradient descent method. We minimize

$$E[x, k] \triangleq |y - k * x|_2^2 + \lambda |k|_p |\nabla x|_2 \tag{7}$$

with respect to x and k by alternating between

$$x^t = x^{t-1} - \epsilon_x \nabla_x E[x^{t-1}, k^{t-1}] \tag{8}$$

and

$$\hat{k}^t = k^{t-1} - \epsilon_k \nabla_k E[x^t, k^{t-1}], \quad \tilde{k}^t = \max\{0, \hat{k}^t\} \quad \text{and} \quad k^t = {}^{\tilde{k}^t}/_{|\tilde{k}^t|_1} \tag{9}$$

with step sizes $\epsilon_x, \epsilon_k > 0$. This method, termed PN, is an approximate solver since the sequential updates of the blur do not correspond to an alternating minimization [30]. Nonetheless, when applied to Problem (1), it has been shown to perform well (albeit not with SotA results). In our experiments we show that the new L^2 blur normalization of Problem (3) can boost its performance dramatically. As with the FW algorithm, we initialize the blur with $k^0 = \delta$.

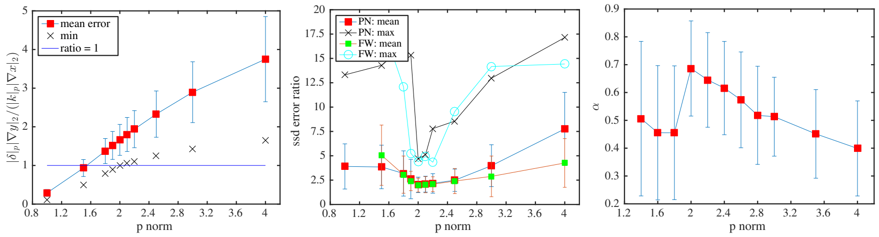
Remark 3. The prior $\lambda|k|_p|\nabla x|_2$ in Problem (3) has a very important role during sharp image estimation. Since the blur is fixed during the estimation of x, we can consider $\lambda|k|_p$ as a regularization weight. This weight changes as k is updated, in turn affecting the amount of regularization of the sharp image. Because we initialize the blur with a Dirac δ, the initial regularization is the highest possible. Then, as the blur gradually moves away from the Dirac δ, the amount of regularization decreases. This annealing of the regularization weight is similar to the heuristic schedule used in other methods [2, 23, 30, 31]. However, note that in our case we obtain this desirable effect in a rigorous fashion from the objective (and by defining the blur initialization). Also note that **our scale constraint is different from adding an L^2 penalty on the blur as used in prior work** [6, 7, 24, 39, 44]. In our case the blur update is affected by the TV regularizer, thus adjusting the amount of blur regularization across iterations.

Remark 4. FW vs PN. FW and PN are both blind deconvolution algorithms. FW is an exact optimization in the sense that it optimizes exactly the original objective. PN instead uses an approximate algorithm to minimize the original objective. Hence, in the following section we use FW to demonstrate the theory and the accuracy. In the experimental section, we mainly use PN as it converges about 3 times faster than FW without losing much accuracy.

4 Scale Normalization Analysis

Our first experiment is to validate Proposition 1 empirically when gradient statistics are non-Gaussian. We use the whole BSDS dataset [1] (500 images) and randomly generate 100 different blurs as in [4] to yield triplets (blurry image y, sharp image x, blur kernel k). We do not add any noise such that every triplet satisfies exactly the convolutional model. We then compute the prior ratio $\frac{|\delta|_p|\nabla y|_2}{|k|_p|\nabla x|_2}$ to analyze how often the prior favors the true sharp image (*i.e.*, the ratio is above 1). Fig. 1a shows the mean and standard deviation of the prior ratio as well as its minimum value (*i.e.*, the worst case) over the whole BSDS dataset for different values of p. We observe that for $p \geq 2$ the prior always favors the estimation of the sharp image (on all samples, and in particular, also in the worst case) as indicated by Proposition 1 for Gaussian image statistics.

Next, we explore different choices of p (between 1 and 4) for both the FW and PN algorithms. We randomly pick 12 images from the BSDS dataset [1], and combine them with 6 blurs from [33] to generate synthetic blurry images. We compute the Sum of Squared Difference (SSD) ratio on the estimated latent sharp image as proposed by [23]. We set $\epsilon_x = 0.005$ and $\epsilon_k = 0.001$. λ is tuned for

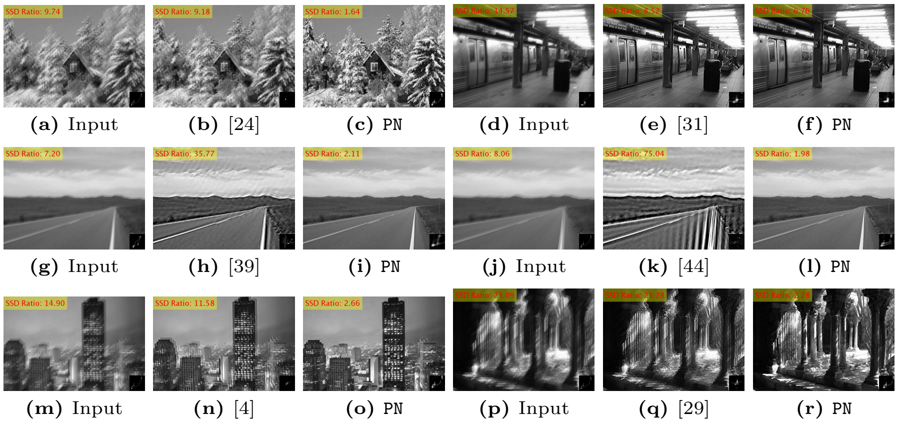


(a) **(b)** **(c)**

Fig. 1: (a) Evaluation of image prior ratios $|\delta|_p|\nabla y|_2/|k|_p|\nabla x|_2$ over triplets (blurry image y, sharp image x, blur kernel k) on the BSDS dataset [1]. The plot shows the mean, standard deviation, and the minimum value of the prior ratio for different values of p. The image prior favors sharp images when the computed prior ratio is above 1. This is always the case with $p \geq 2$. **(b)** Evaluation of different scale normalizations on 12 BSDS images [1] for 6 blurs [33]. We plot the mean and standard deviation of the SSD error ratio of the `FW` and `PN` methods for different values of p. The smallest SSD error ratio is achieved by $p = 2$. **(c)** Evaluation of the blur bias due to the p-norm. For different values of p, we plot the weight $\alpha \in [0, 1]$ (*i.e.*, the coefficient of the convex combination $k_\alpha = \alpha k^* + (1 - \alpha)k_{\text{unif}}$, see text) for which k_α has minimal MSE. As p increases beyond 2, α decreases and the estimated blur moves towards being uniform. This shows a clear bias that tends to smooth the blur kernel and therefore results in oversharpened image reconstructions.

each p-norm to obtain the best performance; the blur is initialized as a Dirac δ. We omit the results for the `FW` method with $p = 1$, since it gets stuck at the trivial solution. Note that the `FW` method converges more slowly than `PN`. The results of both algorithms are shown in Fig. 1b. We observe that $p = 2$ performs best among the different L^p norms.

Our last evaluation is to show that large p values tend to introduce a bias towards uniform blur. To that end, we combine an image from the BSDS dataset with a blur k^*, and then solve the blind deconvolution problem to estimate a blur kernel k for different p values. After obtaining results for different p values, we first spatially align the estimated blurs to the ground truth k^*, and then find the best match within a set of example kernels. To measure the bias, we generate the example set from convex combinations of the true blur k^* and the uniform blur k_{unif}, *i.e.*, $k_\alpha = \alpha k^* + (1 - \alpha)k_{\text{unif}}$ for a few $\alpha \in [0, 1]$ values. We then search for the optimal α for each p, such that k_α has the minimum mean square error (MSE) compared to the corresponding estimated blur. We repeat this experiment on 56 different image/blur pairs and plot the mean and standard deviation of different α weights for different p values in Fig. 1c. We observe that the estimated blur moves increasingly towards the uniform blur as p increases beyond 2, thus showing the unwanted bias. Looking at the resulting images, this bias initially results in an oversharpened latent image and then in artifacts when p is sufficiently large. For $p < 2$ we also observe more bias, stemming from instabilities of algorithm for such p-norms, as they do not consistently favor the true solution over the trivial one (Fig. 1a).

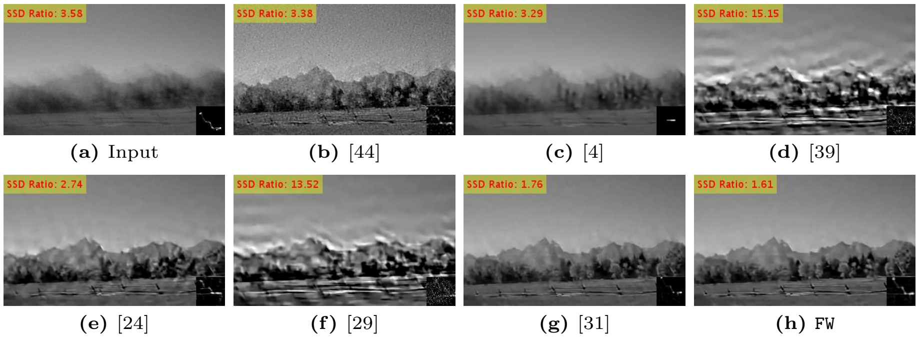


(a) Input **(b)** [24] **(c)** PN **(d)** Input **(e)** [31] **(f)** PN

(g) Input **(h)** [39] **(i)** PN **(j)** Input **(k)** [44] **(l)** PN

(m) Input **(n)** [4] **(o)** PN **(p)** Input **(q)** [29] **(r)** PN

Fig. 2: Visual comparison of the results obtained from the top performers on the SUN dataset [39]. We pick the worst-case result for each top performer and show the corresponding output of our PN method. Our worst-case result is shown in (f).



(a) Input **(b)** [44] **(c)** [4] **(d)** [39]

(e) [24] **(f)** [29] **(g)** [31] **(h)** FW

Fig. 3: One example image from the BSDS dataset [1] tested on different methods.

Considering that a larger p yields more bias (Fig. 1c) and that $p = 2$ is the smallest value for which the prior consistently favors the true solution (Fig. 1a) may explain why the best performing choice in practice is $p = 2$ (see Fig. 1b).

5 Experiments

We test our algorithms on the standard SUN deblurring dataset [39], containing 80 images of average size 1024×768 pixels, which have been synthetically blurred with the 8 blurs from [22] with 1% white Gaussian noise added. The evaluation uses the Sum of Squared Difference (SSD) ratio of [23]. We estimate kernels with various approaches, including ours, and use EPLL [51] to obtain final latent sharp images. Michaeli & Irani [24] pointed out that results with an error ratio below 5 can be considered visually pleasing. We follow this principle, consider an error ratio above 5 as a failure case, and count the number of failure cases for each method to quantify its robustness. In all our evaluations we achieve

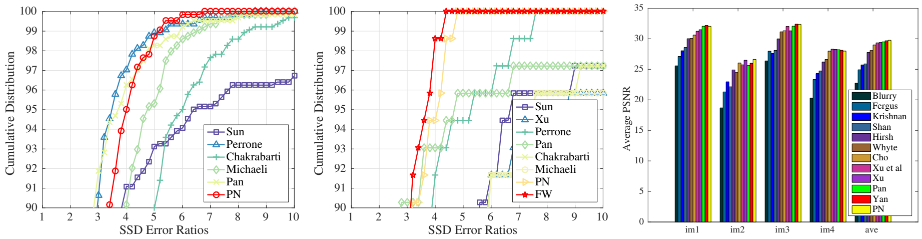


Fig. 4: Left: Cumulative distribution of the SSD error ratios on the whole SUN dataset [39]. Middle: Cumulative distribution of the SSD error ratios on the small BSDS dataset [1]. Right: Quantitative results on the Köhler *et al.* [19] dataset.

Table 1: Quantitative comparison on the entire SUN dataset [39] (640 blurry images).

Method	mean error ratio	maximum error ratio	failure cases
Cho & Lee [7]	9.198	113.491	224
Krishnan *et al.* [20]	12.015	142.668	475
Levin *et al.* [23]	6.695	44.171	357
Sun *et al.* [39]	2.581	35.765	44
Xu & Jia [44]	3.817	75.036	98
Perrone & Favaro [31]	2.114	8.517	**7**
Chakrabarti [4]	3.062	11.576	64
Michaeli & Irani [24]	2.617	9.185	30
Pan *et al.* [29]	**1.914**	23.279	11
`PN`	2.299	**6.764**	8
`FW`	2.195	**6.213**	8

SotA performance in terms of accuracy, worst case errors, and robustness to noise, except for one case where we achieve second place. This highlights the importance of normalization in blind deconvolution.

Fig. 2 shows some visual comparisons between our proposed `PN` method and other top-performing algorithms on the SUN dataset [39]. Results for other methods are collected from the corresponding authors' web page. We pick the worst input image for each algorithm and show the corresponding result obtained with our `PN` algorithm. It can be seen that the proposed algorithm succeeds in most cases. The worst input image for our algorithm is the same as for [31] (Fig. 2f). While the SSD ratio is above 5 in this case, our result is still visually pleasant.

Robustness. To better understand the robustness properties of our algorithm, we show the cumulative distribution of SSD error ratios across the whole SUN dataset [39] in Fig. 4 (left). We observe that our algorithm is on par with other SotA methods. More importantly, we see that our algorithm saturates to 100% faster than the other methods, since our algorithm's worst SSD error ratio is smaller than the others'. Moreover, most of our failure cases have an SSD error ratio below 6. There is only one failure case above 6, shown visually in Fig. 2f. Additionally, in Table 1 we show the mean error ratio, maximum error ratio, and number of failure cases for all methods on the SUN dataset [39]. Our proposed

Table 2: Quantitative comparison on the small BSDS dataset [1] (72 blurry images).

Method	mean error ratio	maximum error ratio	failure cases
Sun *et al.* [39]	2.648	15.152	12
Xu & Jia [44]	3.645	22.272	13
Perrone & Favaro [31]	2.093	7.493	4
Chakrabarti [4]	3.768	11.809	9
Michaeli & Irani [24]	3.458	23.001	14
Pan *et al.* [29]	2.058	13.516	3
Yan *et al.* [46]	2.022	12.237	3
L^1 normalization	2.211	7.821	3
weight decay (heuristic)	2.591	8.762	2
L^2 blur prior (classic)	2.487	7.953	4
`PN`	2.011	4.676	**0**
`FW`	**1.983**	**4.387**	**0**
Quantitative comparison on the full BSDS dataset [1] (3000 blurry images)			
Pan *et al.* [29]	2.956	68.976	325
`PN`	**2.067**	**24.091**	**94**

methods take second and third place in terms of mean error ratio and number of failure cases, but are at the first and second place for maximum error ratio, which highlights their *robustness*. Moreover, our methods require very few tuning parameters (and no adjustment of the regularization weight λ across iterations).

Challenging blurs. Although the 8 blurs from [22] look realistic, the blur sizes only range from 13×13 to 27×27, which is limited. Hence, to additionally evaluate the robustness for all algorithms, we create a small dataset composed of 12 images randomly picked from the BSDS dataset [1]. We collect 6 blurs from [33], which are also recorded realistically. Blur sizes range from 21×21 to 41×41. Hence, our dataset contains 72 blurry images to which we have added 1% zero-mean white Gaussian noise as in the SUN dataset [39]. We estimate blur kernels by running each algorithm and then use EPLL [51] to obtain the final latent sharp image. Fig. 3 shows one visual comparison between all methods. We observe that due to the lack of strong edges, many methods fail to estimate the correct blur. Nonetheless, the `PN` algorithm is robust enough to provide a good blur estimate and then to restore a pleasant latent sharp image. Quantitative results are given in Table 2 and show that our approach *consistently outperforms* all previous methods in all three metrics (mean error ratio, maximum error ratio, and the number of failure cases). Our `FW` and `PN` algorithms succeed on all 72 images. Fig. 4 (middle) shows the cumulative distributions of the SSD error ratio for all competitors. We observe that the `FW` and `PN` algorithms perform very well. In 90% of all cases, our `FW` and `PN` algorithms obtain an error ratio below 3. We also evaluated only the `PN` method and the SotA method [29] on the full BSDS dataset (3000 blurry images from 500 sharp images with 6 blur kernels). Table 2 shows that the benefit of our approach on the full BSDS is even

Table 3: Quantitative comparison (normalized L^2 error) of SotA approaches on Levin's dataset [22] (32 blurry images) with additive noise (*original* means no noise is added).

Method	original	1%	2%	3%	4%	5%
Xu & Jia [44]	3.959	4.132	4.920	5.232	5.623	5.785
Pan *et al.* [29]	3.795	4.008	4.658	4.958	5.325	5.368
Yan *et al.* [46]	3.790	3.901	4.562	4.946	5.278	5.342
PN	**3.788**	**3.815**	**3.911**	**4.098**	**4.188**	**4.202**

Table 4: Quantitative comparison (PSNR) of SotA approaches on Köhler's dataset [19] (48 blurry images) under additive noise (*original* means no noise is added).

Method	original	1%	2%	3%	4%	5%
Xu & Jia [44]	29.475	26.602	25.202	21.523	18.785	17.239
Pan *et al.* [29]	29.551	27.883	26.812	22.116	19.152	18.021
Yan *et al.* [46]	29.595	27.921	26.855	22.143	19.166	18.100
PN	**29.613**	**29.113**	**27.890**	**25.156**	**22.892**	**21.544**

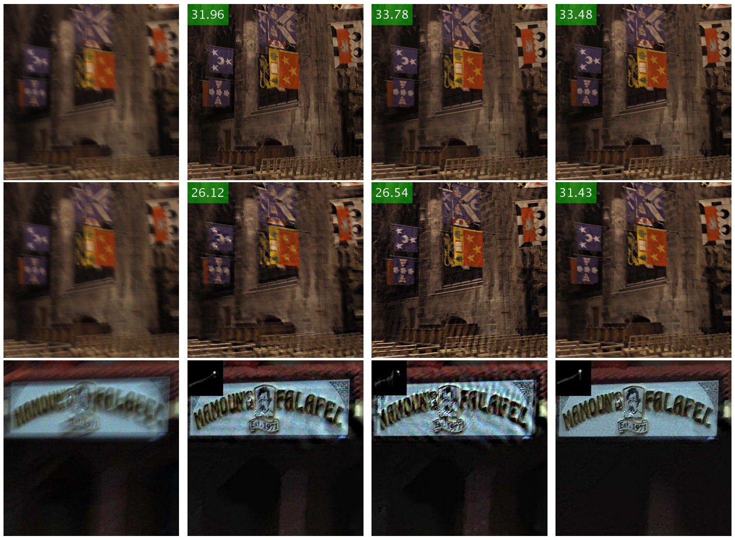
more pronounced than on the smaller subset. For both datasets, we consistently use the same parameter $\lambda = 0.016$ and apply a multi-scale pyramid scheme [24, 29, 31, 39, 46] to speed up convergence of both our PN and FW algorithms. We run 500 and 80 iterations at each scale for the PN and the FW algorithms, respectively.

L^2 vs L^1 blur normalization. To see the effectiveness of L^2 normalization, we compare the proposed Problem (3) with the classic Problem (1) on the BSDS test set. For Problem (1) we employ the same regularization weight scheme used in prior work [29, 31, 46]. By starting with a strong regularization, these methods avoid the trivial solution ($k = \delta$). Nonetheless, we observe that Problem (1) has an SSD ratio of 2.21 (0.2 worse than our PN) and three failure examples, which are shown in the second block of Table 2. Therefore, our L^2 blur normalization scheme outperforms the standard L^1 normalization.

Principled vs heuristic weight decay. As an additional test, we modify the L^1 model to more closely mimic the regularization of Problem (2). Specifically, we artificially scale the regularization weight λ with the L^2 norm of the blur kernel. However, in this modification the blur estimation is carried as usual with Problem (1) (with post normalization). The SSD ratio of this modified algorithm on the BSDS dataset is 2.59, which is almost 0.6 worse than that of Problem (2). Additionally, it has 2 failure cases as shown in the second block of Table 2. This demonstrates empirically that our exact formulation in Problem (2) is better than a heuristic weight decay rule.

Comparison to additive L^2 norm blur prior. We add an explicit L^2 blur regularization term to the objective function in Problem (1) as in SotA approaches [24, 29, 39, 44, 46]. This modification has an SSD error ratio of 2.48 and 4 failure examples, which we report in the second block of Table 2.

Köhler's dataset (real data). We also evaluate our methods on the full dataset of Köhler *et al.* [19]. Quantitative results and one visual example are



(a) Blurry input **(b)** Xu & Jia [44] **(c)** Pan *et al.* [29] **(d)** PN

Fig. 5: Visual comparison of results from [44, 29] compared to the PN method on two real images from [19] and [21]. The first two rows show the deblurred results of an original image from [19] and its version with 2% added noise, respectively.

shown in Fig. 4 (right) and Fig. 6. Although this dataset contains blur from camera shake, the FW and PN algorithms yield visually pleasant estimates. Note that both algorithms were not explicitly designed for such non-uniform blur.

Noise sensitivity. Our numerical analysis in Sec. 4 is based on noise-free images as in [29]. It is important to note that all experiments shown in Sec. 5 are conducted on datasets (SUN and BSDS) in which 1% zero-mean Gaussian noise was added (Table 1 and 2). To quantitatively evaluate the noise sensitivity of our method, we modify Levin's [22] and Köhler's [19] datasets by adding $1\%, 2\%, 3\%, 4\%$, and 5% white Gaussian noise. Quantitative comparisons are shown in Table 3 and 4. Blur kernel estimation errors are measured as the average of the normalized L^2 error $|\hat{k} - k|_2/|k|_2$, where $\hat{k}$ is the estimate and k is the ground truth. Notice that a quantitative comparison on the original dataset (no added noise) is also shown in Fig. 4 (right). Additionally, two visual comparisons on the original and 1% noise level images are shown in the 1st and 2nd rows of Fig. 5. One visual comparison on a real noisy image is shown on the third row of Fig. 5. We observe that our approach performs better than the SotA methods [29, 44, 46] and is overall also less sensitive to noise.

Comparison to [49]. Finally, we compare to the recent L^2 blur normalization approach of Zhang *et al.* [49]. Specifically, we compare on the dataset of Levin *et al.* [22], containing 32 real blurry images, for which the authors of [49] have supplied results. We find that the average of the normalized L^2 error of our kernel estimates is significantly better than in [49] (3.788 vs 5.452). In fact, our results consistently outperform [49] on each of the 32 estimates.

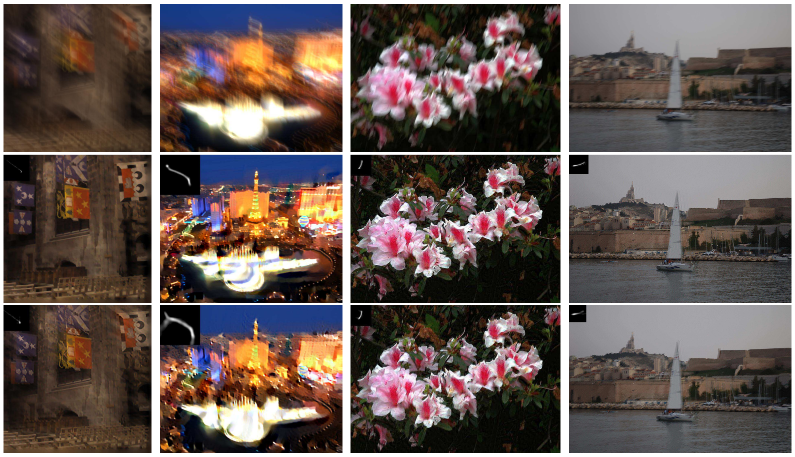


Fig. 6: Worst-case failure examples of our PN method on real images with camera shake from [19]. Top: blurry inputs. Middle: Results from [29]. Bottom: Results with PN.

Limitations. One current limitation of the proposed approach is speed. To process a 400 × 400 blurry image with kernel size 41 × 41, our unoptimized Matlab code on our setup (Intel Core i7-3635QM, 16G RAM) for the PN and FW algorithms estimates the blur kernels in around 2 minutes and 6 minutes, respectively, whereas C++ implemented methods [44] only take less than 10 seconds. Still, ours are approximately 3× and 10× faster than other recent competitive methods [29, 31]. Moreover, due to the convexity of each alternating iteration, we believe that the computational performance can be further improved. In Fig. 6 we show some of our worst failure cases on real images and compare our results with those of the SotA approach from [29]. The first two rows show results in the case of large blur and saturation. In these cases, our results are not as accurate as those of [29]. Nonetheless, the reconstruction artifacts of our PN algorithm are still acceptable, which demonstrates its robustness to noise and saturation.

6 Conclusion

We have introduced a novel scale normalization technique for blind deblurring based on L^p norms and shown both analytically and numerically that the choice $p=2$ avoids trivial solutions that have challenged prior work. We demonstrated that our scale normalization can be interpreted as a rescaled classical objective in which the regularizer is adaptively weighted by the norm of the blur kernel. The resulting method is conceptually simple, obviates common heuristic adaptations of the regularization, and experiments on different datasets show SotA image reconstruction accuracy and a very high degree of robustness.

Acknowledgements. MJ and PF acknowledge support from the Swiss National Science Foundation on project 200021_153324.

References

1. Arbelaez, P., Maire, M., Fowlkes, C.C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE TPAMI **33**(5), 898–916 (2011)
2. Babacan, S.D., Molina, R., Do, M.N., Katsaggelos, A.K.: Bayesian blind deconvolution with general sparse image priors. In: ECCV (2012)
3. Born, M.: Principles of optics - electromagnetic theory of propagation, interference and diffraction of light (7. ed.). Cambridge University Press (1999)
4. Chakrabarti, A.: A neural approach to blind motion deblurring. In: ECCV (2016)
5. Chan, T.F., Wong, C.: Total variation blind deconvolution. IEEE TIP (1998)
6. Chaudhuri, S., Velmurugan, R., Rameshan, R.M.: Blind image deconvolution – Methods and convergence. Springer (2014)
7. Cho, S., Lee, S.: Fast motion deblurring. ACM Trans. Graph. (2009)
8. Cho, S., Lee, S.: Convergence analysis of map based blur kernel estimation. In: ICCV (2017)
9. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. ACM Trans. Graph. **25**(3), 787–794 (2006)
10. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Naval Research Logistics Quarterly (1956)
11. Gast, J., Sellent, A., Roth, S.: Parametric object motion from blur. In: CVPR (2016)
12. Gorski, J., Pfeuffer, F., Klamroth, K.: Biconvex sets and optimization with biconvex functions: A survey and extensions. Math. Meth. of OR (2007)
13. Hirsch, M., Schuler, C.J., Harmeling, S., Schölkopf, B.: Fast removal of non-uniform camera shake. In: ICCV (2011)
14. Hirsch, M., Sra, S., Schölkopf, B., Harmeling, S.: Efficient filter flow for space-variant multiframe blind deconvolution. In: CVPR (2010)
15. Jin, M., Hirsch, M., Favaro, P.: Learning face deblurring fast and wide. In: CVPR Workshops (2018)
16. Jin, M., Meishvili, G., Favaro, P.: Learning to extract a video sequence from a single motion-blurred image. In: CVPR (2018)
17. Kim, T.H., Ahn, B., Lee, K.M.: Dynamic scene deblurring. In: ICCV (2013)
18. Kim, T.H., Lee, K.M.: Segmentation-free dynamic scene deblurring. In: CVPR (2014)
19. Köhler, R., Hirsch, M., Mohler, B.J., Schölkopf, B., Harmeling, S.: Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In: ECCV (2012)
20. Krishnan, D., Tay, T., Fergus, R.: Blind deconvolution using a normalized sparsity measure. In: CVPR (2011)
21. Lai, W., Huang, J., Hu, Z., Ahuja, N., Yang, M.: A comparative study for single image blind deblurring. In: CVPR (2016)
22. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding and evaluating blind deconvolution algorithms. In: CVPR (2009)
23. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Efficient marginal likelihood optimization in blind deconvolution. In: CVPR (2011)
24. Michaeli, T., Irani, M.: Blind deblurring using internal patch recurrence. In: ECCV (2014)
25. Nah, S., Kim, T.H., Lee, K.M.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: CVPR (2017)

26. Noroozi, M., Chandramouli, P., Favaro, P.: Motion deblurring in the wild. In: GCPR (2017)
27. Pan, J., Hu, Z., Su, Z., Lee, H., Yang, M.: Soft-segmentation guided object motion deblurring. In: CVPR (2016)
28. Pan, J., Hu, Z., Su, Z., Yang, M.: Deblurring text images via L0-regularized intensity and gradient prior. In: CVPR (2014)
29. Pan, J., Sun, D., Pfister, H., Yang, M.H.: Blind image deblurring using dark channel prior. In: CVPR (2016)
30. Perrone, D., Favaro, P.: Total variation blind deconvolution: The devil is in the details. In: CVPR (2014)
31. Perrone, D., Favaro, P.: A logarithmic image prior for blind deconvolution. IJCV (2016)
32. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D (1992)
33. Schelten, K., Nowozin, S., Jancsary, J., Rother, C., Roth, S.: Interleaved regression tree field cascades for blind image deconvolution. In: WACV (2015)
34. Schelten, K., Roth, S.: Localized image blur removal through non-parametric kernel estimation. In: ICPR (2014)
35. Schuler, C.J., Hirsch, M., Harmeling, S., Schölkopf, B.: Learning to deblur. IEEE TPAMI. **38**(7), 1439–1451 (2016)
36. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. ACM Trans. Graph. **27**(3) (2008)
37. Srivastava, A., Lee, A.B., Simoncelli, E.P., Zhu, S.: On advances in statistical modeling of natural images. Journal of Mathematical Imaging and Vision (2003)
38. Sun, J., Cao, W., Xu, Z., Ponce, J.: Learning a convolutional neural network for non-uniform motion blur removal. In: CVPR (2015)
39. Sun, L., Cho, S., Wang, J., Hays, J.: Edge-based blur kernel estimation using patch priors. In: ICCP (2013)
40. Whyte, O., Sivic, J., Zisserman, A., Ponce, J.: Non-uniform deblurring for shaken images. IJCV (2012)
41. Wieschollek, P., Hirsch, M., Lensch, H.P.A., Schölkopf, B.: End-to-end learning for image burst deblurring. ACCV (2016)
42. Wipf, D., Zhang, H.: Revisiting Bayesian blind deconvolution. J. Mach. Learn. Res. (2014)
43. Xiao, L., Wang, J., Heidrich, W., Hirsch, M.: Learning high-order filters for efficient blind deconvolution of document photographs. In: ECCV (2016)
44. Xu, L., Jia, J.: Two-phase kernel estimation for robust motion deblurring. In: ECCV (2010)
45. Xu, L., Zheng, S., Jia, J.: Unnatural L0 sparse representation for natural image deblurring. In: CVPR (2013)
46. Yan, Y., Ren, W., Guo, Y., Wang, R., Cao, X.: Image deblurring via extreme channels prior. In: CVPR (2017)
47. Zhang, H., Wipf, D.P., Zhang, Y.: Multi-image blind deblurring using a coupled adaptive sparse prior. In: CVPR (2013)
48. Zhang, H., Yang, J.: Intra-frame deblurring by leveraging inter-frame camera motion. In: CVPR (2015)
49. Zhang, Y., Lau, Y., Kuo, H.W., Cheung, S., Pasupathy, A., Wright, J.: On the global geometry of sphere-constrained sparse blind deconvolution. In: CVPR (2017)
50. Zhu, C., Byrd, R.H., Lu, P., Nocedal, J.: L-BFGS-B – Fortran subroutines for large-scale bound constrained optimization. ACM Trans. Math. Software (1994)

51. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: ICCV (2011)
52. Zuo, W., Ren, D., Gu, S., Lin, L., Zhang, L.: Discriminative learning of iteration-wise priors for blind deconvolution. In: CVPR (2015)