Noise-Blind Image Deblurring

Meiguang Jin
University of Bern
Switzerland

Stefan Roth
TU Darmstadt
Germany

Paolo Favaro
University of Bern
Switzerland

Abstract

We present a novel approach to noise-blind deblurring, the problem of deblurring an image with known blur, but unknown noise level. We introduce an efficient and robust solution based on a Bayesian framework using a smooth generalization of the 0-1 loss. A novel bound allows the calculation of very high-dimensional integrals in closed form. It avoids the degeneracy of Maximum a-Posteriori (MAP) estimates and leads to an effective noise-adaptive scheme. Moreover, we drastically accelerate our algorithm by using Majorization Minimization (MM) without introducing any approximation or boundary artifacts. We further speed up convergence by turning our algorithm into a neural network termed GradNet, which is highly parallelizable and can be efficiently trained. We demonstrate that our noise-blind formulation can be integrated with different priors and significantly improves existing deblurring algorithms in the noise-blind and *in the known-noise case. Furthermore, GradNet leads to state-of-the-art performance across different noise levels, while retaining high computational efficiency.*

1. Introduction

Non-blind image deblurring has been studied extensively in the literature. Its principal assumption is that the blur kernel affecting the image is known ahead of time. While this may seem limiting, the blur may be known from the design of the imaging system [14] or can be estimated through other modalities, *e.g.*, inertial sensors [12]. Moreover, the vast majority of blind deblurring algorithms have a non-blind subcomponent [15], alternating between kernel estimation and non-blind deblurring.

Even if the blur kernel is known, image deblurring is still difficult due to the loss of high-frequency information and the sensor noise. Moreover, noise cannot be avoided even with the best image sensors. Although we might theoretically calibrate the noise level for each camera and each ISO level, this quickly becomes infeasible in practice.

One approach to address this issue is to use a separate noise estimator to tune a deblurring algorithm that assumes known noise. For example [13, 24, 25, 37] focus on the scenario where the noise level is known or user specified. Discriminative approaches [24, 25] are even custom-trained for specific noise levels; we would need to train and store a deblurring method for each noise level, which is not practical. A key challenge of a separate noise estimation step is that most noise estimation algorithms [6, 8, 17, 19, 35] are designed for non-blurry input. An exception is [36], which is able to estimate noise levels from blurry images. As we show in the experiments, the combination of noise estimation with subsequent deblurring can be suboptimal both in accuracy and in execution time.

Therefore, we aim at estimating both the noise level and a sharp image from a single noisy and blurred image, a problem that we call *noise-blind image deblurring*. There has been very little work on noise-blind deblurring so far. Schmidt *et al*. [26] propose a Bayesian framework to deal with the noise-blind case. Nevertheless, their sampling-based technique is computationally very intensive, thus impractical for high-resolution images. In fact, computational efficiency is a challenge even in the known noise case; only very few fast and effective approaches exist [13, 25, 30].

In this paper, we propose an approach to noise-blind deblurring based on a noise-adaptive formulation derived from Bayesian principles. More specifically, instead of using the common 0-1 loss, which yields the well-known Maximum a-Posteriori (MAP) estimation, we use a smooth Gaussian utility function. We treat noise as a parameter that can be integrated into the data term of the energy function. As a consequence our formulation is noise-adaptive, and tuning for different noise levels is no longer needed. Moreover, we majorize the energy function, such that FFT-based preconditioning can be applied, which speeds up the execution process significantly, but also avoids artifacts from circular boundary assumptions [25, 30]. We combine the above formulations and derive a convolutional neural network, which we call *GradNet*, that can solve the noise-blind image deblurring problem with very high computational efficiency. Each block of layers in GradNet implements a gradient descent step. Thus, the training of such network is the optimization of a gradient descent algorithm [2, 7]. We can also

interpret GradNet as a generalization of the diffusion network of [4] to the noise-blind deblurring problem, where we integrated our noise adaptivity and FFT-based preconditioning. Hence, our network is also highly parallelizable and well-suited for computation on GPUs, which makes inference very fast, yet achieves very high image quality.

Our work makes a number of contributions: *(1)* The proposed noise-adaptive formulation is conceptually simple and easy to calculate, with little computational cost; *(2)* it is easily integrated into existing image restoration frameworks, and even with a simple total variation prior it can already achieve high levels of image quality; *(3)* the automatic weighting between data term and prior can yield significant benefits even in the known-noise case (0.3–0.7dB on average); *(4)* our noise-adaptive formulation is also able to deal with colored (spatially correlated) noise (see supplementary material); *(5)* FFT-based preconditioning makes solving the non-blind deblurring problem much faster; *(6)* our trainable network GradNet makes inference even faster, yet outperforms the state of the art especially for large blur kernels.

2. Related Work

Non-blind deblurring is used not only when kernels are known [14], but also in blind deblurring [5, 9, 15, 16, 20, 22, 28, 29, 32, 34] to restore the final sharp images.

Most non-blind deblurring approaches can be divided into two classes, either based on iterative energy minimization [13, 14, 30, 37] or discriminative learning [24, 25, 27, 33]. Wang *et al*. [30] rely on total variation and use half-quadratic techniques to speed up optimization. Krishnan and Fergus [13] similarly combine high-quality results with fast execution. Levin *et al*. [14] formulate a more advanced prior using second-order derivative filters. Zoran and Weiss [37] use a Gaussian mixture prior, which is learned from a set of natural images. This approach (EPLL) has been widely used in blind deblurring for estimating the final sharp image owing to its high-quality restoration results. However, all these methods need to be well tuned according to the noise level at test time. On the other hand, Schuler *et al*. [27] propose a two-step approach, which first uses a regularized inversion of the blur in the Fourier domain and then removes the noise in the second step. Schmidt and Roth [25] propose shrinkage fields, a discriminatively trained network architecture, which is very efficient. However, it suffers from boundary artifacts due to its circular boundary assumption and is not noise adaptive. Schmidt *et al*. [24] propose a Gaussian conditional random field framework, where parameters are predicted using learned regression trees [11]. Xu *et al*. [33] design a CNN to handle saturation and nonlinearities of the model. However, these learning approaches are designed/trained for a specific noise level and not robust to other noise levels. Bayesian deblurring [26] is an exception, which is able to integrate non-blind deblurring and noise estimation with a Bayesian minimum mean squared error estimate. However, this approach is computationally inefficient; scaling it to larger images is prohibitively slow.

An intuitive way to deal with noise-blind deblurring is first to estimate noise [6, 8, 17, 19, 35] and then apply existing non-blind deblurring algorithms. Donoho *et al*. [8] propose a mean absolute deviation (MAD) framework to infer noise levels from the wavelet coefficients at the highest resolution. Zlokolica *et al*. [35] extend the MAD framework to video noise estimation. Liu *et al*. [17] estimate an upper bound on the noise level from a single image based on a piecewise smooth image prior. De Stefano *et al*. [6] explore the relationship between kurtosis values and image noise in a wavelet-based approach. Liu *et al*. [19] apply principal component analysis to selected patches to estimate the noise level. However, none of these methods explicitly deals with the case where the image is also blurry. The work of Zoran and Weiss [37] is an exception, which exploits the connection between kurtosis values and image noise levels. Their work can also estimate the noise level under image blur.

3. Bayesian Noise-Blind Image Deblurring

Let $\bar{x}$ represent an unknown sharp image and k a given blur kernel with non-negative values integrating to 1. We assume that the observed blurry image y is formed[1] as

$$y = k * \bar{x} + n, \qquad n \sim \mathcal{N}(0, \bar{\sigma}_n), \tag{1}$$

where n is Gaussian zero-mean noise with unknown standard deviation $\bar{\sigma}_n$. Alternatively, we can rewrite the image formation via the Toeplitz matrix K of the blur kernel k as $y = K\bar{x} + n$, where we rearranged the sharp and blurry images, as well as the noise n into column vectors. We consider $\bar{\sigma}_n$ a parameter, which is equivalent to assuming that it follows a yet unknown Dirac delta distribution. We aim to recover both $\bar{x}$ and $\bar{\sigma}_n$ given y and k.

To that end we first define a *loss function* L between the true $(\bar{x}, \bar{\sigma}_n)$ and the estimate (x, σ_n), given observation y and kernel k. Formally speaking, the loss function maps $(y, k, \bar{x}, \bar{\sigma}_n, x, \sigma_n)$ to $[0, \infty)$. For notational simplicity, we drop y, k as they are fixed. Also, since σ_n and $\bar{\sigma}_n$ are parameters, *i.e*. modeled by Dirac delta distributions, they are forced to be equal (for reasonable loss functions). We can thus directly substitute $\bar{\sigma}_n$ with σ_n and omit the parameters σ_n and $\bar{\sigma}_n$ in the definition of the loss function.

In our formulation we consider Bayes' risk

$$E_{\bar{x},y;\sigma_n}[L(\bar{x}, x)] = \int L(\bar{x}, x) p(\bar{x}, y; \sigma_n)\, d\bar{x}\, dy, \tag{2}$$

and define the estimator $(\tilde{x}, \tilde{\sigma}_n)$ via

$$(\tilde{x}, \tilde{\sigma}_n) = \arg\min_{x, \sigma_n} E_{\bar{x},y;\sigma_n}[L(\bar{x}, x)]. \tag{3}$$

[1] $*$ is a 'valid' convolution, *i.e*., the output y is smaller than the input x.

A common choice is the 0-1 loss

$$L(\bar{x}, x) = 1 - \delta(\bar{x} - x), \tag{4}$$

which leads to the Maximum-a-Posteriori (MAP) problem

$$(\tilde{x}, \tilde{\sigma}_n) = \arg\max_{x,\sigma_n} p(y|x;\sigma_n)p(x). \tag{5}$$

Here, the joint probability $p(x, y; \sigma_n) = p(y|x;\sigma_n)p(x)$ is written as product of likelihood and prior. Now, let us consider the denoising case ($k = 1$). The log-likelihood is given as

$$\log p(y|x;\sigma_n) = -\tfrac{1}{2\sigma_n^2}|y - x|^2 - M \log \sigma_n + \text{const}, \tag{6}$$

where M is the number of pixels in x and the constant is due to the partition function. The MAP solution becomes

$$\arg\min_{x,\sigma_n} \tfrac{1}{2\sigma_n^2}|y - x|^2 + M \log \sigma_n - \log p(x). \tag{7}$$

By setting the first derivative w.r.t. σ_n to 0, we have

$$\sigma_n^2 = \tfrac{1}{M}|y - x|^2, \tag{8}$$

which is the well-known variance sample estimator. We plug this closed form solution into Eq. (7) and obtain

$$\tilde{x} = \arg\min_{x} \tfrac{M}{2} \log |y - x|^2 - \log p(x). \tag{9}$$

The solution to Eq. (9) is $\tilde{x} = y$, since the first term tends to $-\infty$ while the second term will be typically finite at $x = y$. This solution, however, is undesirable as it performs no denoising. To address this failure, we introduce a different loss function and a novel lower bound.

4. Beyond Maximum a-Posteriori

To avoid the degenerate solution of Eq. (9) we introduce a family of loss functions that does not drastically penalize small errors of the estimators. Let us define the loss function as $L(\bar{x}, x) = 1 - G(\bar{x}, x)$, where we call G the *utility function*,[2] and impose that $G(\bar{x}, x) \geq 0$ and $\int G(\bar{x}, x)\, d\bar{x} = 1$. For example, we can choose a Gaussian density with partition function Z and variance σ^2:

$$G(\bar{x}, x) = \tfrac{1}{Z} \exp\left[-\tfrac{|\bar{x}-x|^2}{2\sigma^2}\right]. \tag{10}$$

This family of smooth loss functions generalizes the 0-1 loss, which is its limit case as $\sigma \to 0$. We then obtain

$$E_{\bar{x},y;\sigma_n}[L(\bar{x}, x)] = 1 - E_{\bar{x},y;\sigma_n}[G(\bar{x}, x)], \tag{11}$$

and the minimization of Bayes' risk

$$\arg\min_{x,\sigma_n} E_{\bar{x},y;\sigma_n}[L(\bar{x}, x)] = \arg\max_{x,\sigma_n} E_{\bar{x},y;\sigma_n}[G(\bar{x}, x)] \tag{12}$$

becomes the maximization of *Bayes' utility* (BU). More explicitly, we have

$$\begin{aligned}\arg\max_{x,\sigma_n} E_{\bar{x},y;\sigma_n}[G(\bar{x}, x)] &= \arg\max_{x,\sigma_n} \log E_{\bar{x},y;\sigma_n}[G(\bar{x}, x)] \\ &= \arg\max_{x,\sigma_n} \log \int G(\bar{x}, x)p(\bar{x}, y;\sigma_n)\, d\bar{x}.\end{aligned} \tag{13}$$

Because of Jensen's inequality and since $\log$ is concave, the logarithm of BU has a lower bound (right hand side)

$$\log \int G(\bar{x}, x)p(\bar{x}, y;\sigma_n)\, d\bar{x} \geq \int G(\bar{x}, x) \log p(\bar{x}, y;\sigma_n)\, d\bar{x} \tag{14}$$

and therefore

$$\begin{aligned}\max_{x,\sigma_n} \log \int G(\bar{x}, x)p(\bar{x}, y;\sigma_n)\, d\bar{x} \geq \\ \max_{x,\sigma_n} \int G(\bar{x}, x) \log p(\bar{x}, y;\sigma_n)\, d\bar{x}.\end{aligned} \tag{15}$$

The advantage of the above lower bound to BU is that it can be computed in closed form – despite the high-dimensional integral – whenever $\log p(\bar{x}, y;\sigma_n) = \log p(y|\bar{x};\sigma_n) + \log p(\bar{x})$ takes simple forms (*e.g.*, linear or quadratic).

Data fidelity term. Let us now consider the deblurring problem. We start by considering the log-likelihood

$$\log\ p(y|\bar{x};\sigma_n) = -\tfrac{|y-k*\bar{x}|^2}{2\sigma_n^2} - N \log \sigma_n + \text{const}, \tag{16}$$

where N is the number of pixels of the blurry image y. By plugging Eq. (16) into the right hand side of Eq. (14) the contribution of the log-likelihood to the bound becomes

$$\begin{aligned}&\int G(\bar{x}, x) \log p(y|\bar{x};\sigma_n)\, d\bar{x} \\ &\quad= -\int \tfrac{1}{Z} e^{-\frac{|\bar{x}-x|^2}{2\sigma^2}} \tfrac{|y-k*\bar{x}|^2}{2\sigma_n^2}\, d\bar{x} - N \log \sigma_n + \text{const} \\ &\quad= -\tfrac{|y-k*x|^2}{2\sigma_n^2} - M\tfrac{\sigma^2}{2\sigma_n^2}|k|^2 - N \log \sigma_n + \text{const}.\end{aligned} \tag{17}$$

Image priors. We consider a product of type-1 Gumbel density functions [21] of the squared norm of image filter responses as image prior. A broad enveloping Gaussian ensures the distribution to be proper. This prior takes the form

$$\log p(\bar{x}) = -\tfrac{|\bar{x}|^2}{2\sigma_0^2} + \sum_{ijk} w_{ij} \exp\left[-\tfrac{|F_{ik}\bar{x}-\mu_j|^2}{2\sigma_j^2}\right] + \text{const}, \tag{18}$$

[2] Notice that the two constraints on G are irrelevant (in the vast majority of instances) as far as Bayes' risk minimization is concerned. Positivity can be achieved by adding a constant to the loss function and normalization can be achieved by scaling the whole cost by a positive constant. Both of these modifications to Bayes' risk will not affect the minimizer (as long as the loss function is bounded from below).

where F_i are Toeplitz matrices representing the filters, F_{ik} yields the k^{th} entry of the output and is therefore a row vector with M pixels; μ_j and σ_j are parameters. σ_0^2 is chosen to be a large constant. Later we will see that this prior has connections to common priors based on products of Gaussian mixtures [26, 37]. The weights w_{ij} must be positive, but do not have to sum to 1 here. Other priors, such as total variation, are discussed in the supplementary material.

By constraining the filters to $|F_{ik}|_2 = 1$, we obtain the contribution of the log-prior to the bound in Eq. (14)

$$\int G(\bar{x}, x) \log p(\bar{x})\, d\bar{x} = \tag{19}$$
$$-\tfrac{|x|^2}{2\sigma_0^2} - \sum_{ijk} \hat{w}_{ij} \exp\left[-\tfrac{|F_{ik}x-\mu_j|^2}{2(\sigma^2+\sigma_j^2)}\right] + \text{const},$$

where, for ease of notation, we define

$$\hat{w}_{ij} = -w_{ij} \exp\left[-\mu_j^2 \tfrac{\sigma^2}{\sigma_j^2(\sigma^2+\sigma_j^2)}\right]. \tag{20}$$

Notice that when $\sigma \to 0$ the bound collapses to Eq. (18).

Image deblurring. Finally, we can put all the terms together and solve the maximization of the lower bound

$$\arg\max_{x,\sigma_n} \int G(\bar{x}, x) \log p(\bar{x}, y; \sigma_n)\, d\bar{x} \tag{21}$$

to BU as the following problem:

$$(\hat{x}, \hat{\sigma}_n) = \arg\min_{x,\sigma_n} \tfrac{|y-k*x|^2+M\sigma^2|k|^2}{2\sigma_n^2} + N \log \sigma_n$$
$$+ \tfrac{|x|^2}{2\sigma_0^2} + \sum_{ijk} \hat{w}_{ij} \exp\left[-\tfrac{|F_{ik}x-\mu_j|^2}{2(\sigma^2+\sigma_j^2)}\right]. \tag{22}$$

We can now solve explicitly for σ_n and obtain

$$\sigma_n^2 = \tfrac{1}{N}\left[|y-k*x|^2 + M\sigma^2|k|^2\right]. \tag{23}$$

This closed form can be incorporated in Eq. (22) and yields

$$\boxed{\hat{x} = \arg\min_x U[x] \doteq \arg\min_x \tfrac{N}{2} \log\left[|y-k*x|^2 + M\sigma^2|k|^2\right] + \tfrac{|x|^2}{2\sigma_0^2} + \sum_{ijk} \hat{w}_{ij} \exp\left[-\tfrac{|F_{ik}x-\mu_j|^2}{2(\sigma^2+\sigma_j^2)}\right].} \tag{24}$$

We point out that this formulation does not lead to degenerate solutions in the case of denoising *or* deblurring. In fact with denoising ($k = 1$), Eq. (24) is not minimized at $x = y$. In the more general noise-blind deblurring formulation, we can explicitly obtain the gradient descent iteration

$$x^{\tau+1} = x^\tau - \alpha \nabla_x U[x^\tau] \tag{25}$$
$$\nabla_x U[x^\tau] = \lambda^\tau K^\top (Kx^\tau - y) + \tfrac{x^\tau}{\sigma_0^2} - \sum_{ik} F_{ik}^\top \phi_i(F_{ik}x^\tau)$$
$$\phi_i(z) = \sum_j \hat{w}_{ij} \exp\left[-\tfrac{|z-\mu_j|^2}{2(\sigma^2+\sigma_j^2)}\right] \tfrac{z-\mu_j}{\sigma^2+\sigma_j^2},$$

for some small step $\alpha > 0$, where x^τ denotes the solution at gradient descent iteration τ and $\lambda^\tau = \frac{N}{|y-Kx^\tau|^2+M\sigma^2|k|^2}$.

Discussion. The alternative Bayesian approach by Schmidt *et al.* [26] instead directly minimizes the Bayesian minimum mean squared error (MMSE)

$$\hat{x} = \arg\min_x \int |\bar{x} - x|^2 p(\bar{x}, \bar{\sigma}_n | y)\, d\bar{x}\, d\bar{\sigma}_n. \tag{26}$$

This very high-dimensional integration is then solved via Gibbs sampling, but it is computationally intensive. In contrast, in our case the form of the utility function and the proposed lower bound allow a simple analytical solution. Notice that while we focus on Gaussian utility functions, other choices (of probability density functions) lead to similar closed form solutions. The utility function G has a regularizing effect on both the noise estimates through λ^τ and the image prior filters. When $\sigma \gg 1$ then λ^τ is biased towards larger noise estimates and the image prior tends to flatten more the filter responses while fixing the coefficients $\hat{w}_{ij}$ to $-w_{ij} \exp[-\mu_j^2/\sigma_j^2]$ (see Eq. 20).

Notice also that $\nabla_x U[x^\tau]$ is similar to the gradient of a standard least squares estimation with some prior $p(x)$:

$$U_{L_2}[x] \doteq \tfrac{\lambda}{2}|y-k*x|^2 - \log p(x)$$
$$\nabla_x U_{L_2}[x^\tau] = \lambda K^\top (Kx^\tau - y) - \tfrac{p'(x^\tau)}{p(x^\tau)}. \tag{27}$$

The main difference is that in Eq. (25) the parameter λ^τ changes during each iteration τ and thus adaptively determines the amount of regularization. Instead, λ is constant in the minimization of U_{L_2}. As shown later, our adaptive λ^τ yields a better solution than any choice of a fixed λ.

5. Exact Preconditioning

We now describe an alternative method to the gradient descent iteration of Eq. (25), which minimizes the problem in Eq. (24) more efficiently while not introducing any approximation. We use the Majorization Minimization (MM) technique [10]. MM defines an iteration much like gradient descent, but such that every step is easy to compute and still provably minimizes the original cost, here Eq. (24). We first define a *surrogate function* $\psi(x|x^\tau)$, where x^τ is the solution at iteration τ, such that $\forall x$ we have $\psi(x|x^\tau) \geq U[x]$, and $\psi(x^\tau|x^\tau) = U[x^\tau]$. We split the construction of ψ into two surrogate functions ψ_1 and ψ_2, *i.e.*, $\psi(x|x^\tau) = \psi_1(x|x^\tau) + \psi_2(x|x^\tau)$, each of which will be a surrogate function to one of the terms in Eq. (24), *i.e.*, $\forall x$

$$\psi_1(x|x^\tau) \geq \tfrac{N}{2} \log\left[|y-k*x|^2 + M\sigma^2|k|^2\right]$$
$$\psi_2(x|x^\tau) \geq \tfrac{|x|^2}{2\sigma_0^2} + \sum_{ijk} \hat{w}_{ij} \exp\left[-\tfrac{|F_{ik}x-\mu_j|^2}{2(\sigma^2+\sigma_j^2)}\right]. \tag{28}$$

Data term. The logarithm in the first term in Eq. (24) is concave and thus we can use a first-order Taylor expansion

as upper bound. Furthermore, we add a quadratic term with the Toeplitz matrix H corresponding to the periodic circular convolution with kernel k. By using the matrix notation K for the blur k and λ^τ, we have

$$\begin{aligned}\psi_1(x|x^\tau) = &\left[\lambda^\tau K^\top (Kx^\tau - y)\right]^\top (x - x^\tau) \\ &+ (x - x^\tau)^\top \lambda^\tau \tfrac{H^\top H}{2}(x - x^\tau) + \text{const}(x^\tau).\end{aligned} \tag{29}$$

Image prior. In the second term we use a quadratic upper bound instead:

$$\begin{aligned}\psi_2(x|x^\tau) = &\text{const}(x^\tau) + \left(\tfrac{x^\tau}{\sigma_0^2} - \textstyle\sum_{ik} F_{ik}^\top \phi_i(F_{ik}x^\tau)\right)^\top (x - x^\tau) \\ &+ \tfrac{\gamma}{2}(x - x^\tau)^\top \left(\delta + \textstyle\sum_{ik} F_{ik}^\top F_{ik}\right)(x - x^\tau),\end{aligned} \tag{30}$$

where $\gamma = \max_i 2\sum_j \frac{|\hat{w}_{ij}|}{\sigma^2 + \sigma_j^2}$ and $\delta = \frac{1}{\gamma\sigma_0^2}$.

Preconditioning. Now we can minimize $\psi(x|x^\tau)$ with respect to x by setting its gradient to 0:

$$\begin{aligned}\nabla_x \psi(x|x^\tau) = &\lambda^\tau K^\top (Kx^\tau - y) + \lambda^\tau H^\top H(x - x^\tau) \\ &+ \tfrac{x^\tau}{\sigma_0^2} - \textstyle\sum_{ik} F_{ik}^\top \phi_i(F_{ik}x^\tau) \\ &+ \gamma\left(\delta + \textstyle\sum_{ik} F_{ik}^\top F_{ik}\right)(x - x^\tau) = 0.\end{aligned} \tag{31}$$

Since this is a linear system, we arrive at the iteration

$$\boxed{\begin{aligned} x^{\tau+1} &= x^\tau - \Lambda \nabla_x U[x^\tau] \\ \Lambda^{-1} &= \lambda^\tau H^\top H + \gamma\left(\delta I + \textstyle\sum_{ik} F_{ik}^\top F_{ik}\right), \end{aligned}} \tag{32}$$

which is a modification of the previous gradient descent in Eq. (25) via preconditioning. Since preconditioning with positive semidefinite matrices maintains the convergence of gradient descent, we can also substitute the filters F_i in the preconditioner with the corresponding periodic circular convolution Toeplitz matrices B_i and obtain our algorithm

$$\boxed{\begin{aligned} x^{\tau+1} &= x^\tau - \Lambda \nabla_x U[x^\tau] \\ \Lambda^{-1} &= \lambda^\tau H^\top H + \gamma \textstyle\sum_{ik} B_{ik}^\top B_{ik} + \frac{1}{\sigma_0^2} I \\ \nabla_x U[x^\tau] &= \lambda^\tau K^\top (Kx^\tau - y) \\ &\quad + \tfrac{x^\tau}{\sigma_0^2} - \textstyle\sum_{ik} F_{ik}^\top \phi_i(F_{ik}x^\tau) \\ \phi_i(z) &= \textstyle\sum_j \hat{w}_{ij} \exp\left[-\frac{|z-\mu_j|^2}{2(\sigma^2+\sigma_j^2)}\right] \frac{z-\mu_j}{\sigma^2+\sigma_j^2}. \end{aligned}} \tag{33}$$

This preconditioner can be computed very efficiently via the fast Fourier transform (FFT) [25]. Notice that our derivation ensures convergence to a local minimum of the *original* cost $U[x]$, and thus unlike [25] it does not suffer from artifacts due to the periodic boundary assumptions of H and B_{ik}. In other words, circular convolutions are only used in the preconditioner, but not in the cost and its gradient, where valid convolutions are applied.

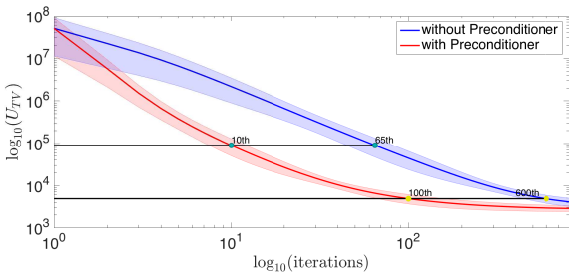


Figure 1. Cost U_{TV} with and without preconditioning. The proposed preconditioning (Eq. 32) leads to much faster convergence.

Figure 1 shows the average cost U_{TV} ($\pm$ the standard deviation), where we used the TV prior in Eq. (33), against the iteration time over 32 images from the dataset of Levin *et al.* [16] with and without preconditioning (omitting noise-adaptivity and noise). Notice how preconditioning accelerates convergence between 6 and 6.5 times.

Discussions. We now point out some fundamental differences and similarities between two previous methods [4, 25] and Eq. (33). First, we turn to the cascade of shrinkage fields (CSF) of Schmidt and Roth [25]. Even though not originally derived in this way, based on Sec. 3 we can rewrite Eq. (10) in [25] as a gradient descent step with preconditioning (with $\sigma_0^2 \to \infty$)

$$\begin{aligned} x^{\tau+1} &= x^\tau - \Lambda^{-1} \nabla_x U_{\text{SF}}[x^\tau] \\ \nabla_x U_{\text{SF}}[x^\tau] &= \lambda H^\top (Hx^\tau - y) - \textstyle\sum_{ik} B_{ik}^\top \phi_i^{\text{SF}}(B_{ik}x^\tau) \\ \phi_i^{\text{SF}}(z) &= z - \textstyle\sum_j \pi_{ij} \exp\left[-\gamma_{\text{SF}}|z - \mu_j|^2\right]. \end{aligned} \tag{34}$$

The main differences to our approach are *(1)* the missing noise adaptivity term λ^τ, *(2)* the use of Toeplitz matrices H and B_{ik} in the definition of the gradient leading to artifacts in shrinkage fields due to circular boundary conditions, and *(3)* in the definition of ϕ_i^{SF}, which we interpret as an approximation of the gradient of the negative log of an image prior. Based on our derivation, the above iteration can be seen as the minimizer of the following image prior (*c.f.* Eq.18)

$$\log p(x) = -\textstyle\sum_{ijk}\left[\frac{|F_{ik}x|^2}{2} + \hat{\pi}_{ij} e^{-\gamma_{\text{SF}}|F_{ik}x - \mu_j|^2}\right], \tag{35}$$

for some $\hat{\pi}_{ij}$ and where we used the difference of two Gaussians to approximate terms $\exp\left[-\gamma_{\text{SF}}|z - \mu_j|^2\right](z - \mu_j)$ in the radial basis functions (RBF) expansion in ϕ_i^{SF}.

In the case of trainable nonlinear reaction diffusion (TNRD) of Chen and Pock [4] with an RBF influence function we can also rewrite their Eq. (3) in our formalism as

$$\begin{aligned} x^{\tau+1} &= x^\tau - \Delta^\tau \nabla_x E_{\text{TN}}[x^\tau] \\ \nabla_x E_{\text{TN}}[x^\tau] &= \lambda K^\top (Kx^\tau - y) - \textstyle\sum_{ik} F_{ik}^\top \phi_i^{\text{TN}}(F_{ik}x^\tau) \\ \phi_i^{\text{TN}}(z) &= -\textstyle\sum_j \hat{w}_{ij} \exp\left[-\frac{|z-\mu_j|^2}{\gamma_j}\right]. \end{aligned} \tag{36}$$

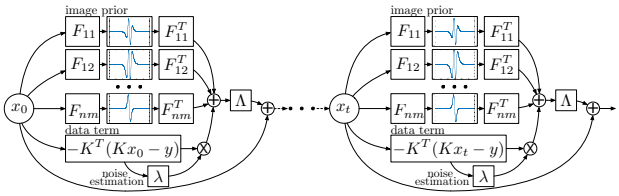


Figure 2. The GradNet architecture.

The main differences to our approach are *(1)* the lack of noise adaptivity, *(2)* the lack of preconditioning, and *(3)* in the definition of ϕ_i^{TN}. By using similar approximations as in the previous comparison, Eq. (36) can be seen as the approximate minimizer of the following image prior

$$\log p(x) = -\sum_{ijk} \hat{w}_{ij} \exp\left[-\frac{|F_{ik}x-\mu_j|^2}{\gamma_j}\right]. \qquad (37)$$

Notice that the above methods were derived based on a mixture of Gaussians prior. Our derivation above shows a link between that prior and the type-1 Gumbel prior (Eq. 18).

6. The GradNet Architecture

We implement the gradient descent from Eq. (33) as a neural network architecture so that the filters and all the unknown parameters can be learned directly from data. An illustration of the network, which we call *GradNet* is shown in Fig. 2. Our GradNet is trained in a supervised manner, *i.e.*, we predefine Q-tuples of training samples $\{y_q, k_q, x_q^{\text{GT}}\}_{q=1}^Q$, where y_q is a noisy blurry image, k_q is the corresponding blur kernel, and x_q^{GT} is the latent sharp image. We use 57 RBF functions and fix $\sigma_j = 10$, $\mu_j \in [-280 : 10 : 280]$ and $\sigma_0 = 10^5$. A GradNet with S stages learns model parameters $\Theta = \{\gamma^\tau, \sigma, f_i^\tau, \hat{w}_{ij}^\tau\}_{\tau=1,\dots,S}$, which include regularization tradeoff γ^τ, σ in the noise-adaptivity λ_q^τ, linear filters f_i^τ (2D kernel of F_i), and coefficients $\hat{w}_{ij}^\tau$ by minimizing the following loss function

$$\min_\Theta L(\Theta) = \min_\Theta \sum_{q=1}^Q \tfrac{1}{2}\left|C_q^S(x_q^S - x_q^{\text{GT}})\right|_2^2, \qquad (38)$$

$$\text{s.t.} \begin{cases} x_q^{\tau+1} = x_q^\tau - \Lambda \nabla_x U[x_q^\tau], \quad \tau \in [0,\dots,S-1] \\ \Lambda^{-1} = \lambda_q^\tau {H_q}^\top H_q + \frac{I}{\sigma_0^2} + \gamma^\tau \sum_{ik} B_{ik}^{\tau\top} B_{ik}^\tau, \end{cases}$$

where C_q^S is an operator that selects only the valid part of the latent image and we initialize x_q^0 by a 3-fold edge tapering of y_q. Recall that B_i^τ is the Toeplitz matrix for circular convolution with filter f_i^τ. Additionally, instead of learning arbitrary filters, we define each kernel as $f_i^\tau = \frac{\sum_d \alpha_{id}^\tau t_d}{|\alpha_i^\tau|_2}$, where $\{t_1, \dots, t_{48}\}$ is a Discrete Cosine Transform (DCT) basis, so that $|f_i^\tau|_2 = 1$ and they are zero-mean. In $\nabla_x U$ we consider functions $\phi_i(z) = \sum_j \hat{w}_{ij} \exp\left[-\frac{|z-\mu_j|^2}{2(\sigma^2+\sigma_j^2)}\right]$, since we found experimentally that they yield the same performance as the functions ϕ_i defined in Eq. (33), but are faster to train. More details, including the backpropagation, are reported in the supplementary material.

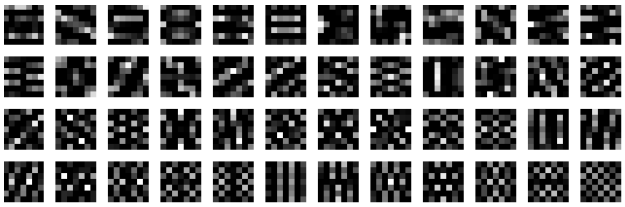


Figure 3. 48 learned filters from the 5th stage of GradNet.

Figure 4. 5 large kernels [23] that are tested with images from [1].

7. Experiments

Training. We choose $S = 7$ stages and train the network with a greedy + joint training scheme. First, we greedily train each of the 7 stages one after the other. Afterwards the network is finetuned jointly. In the first 4 stages, we simply use 4 pre-defined pairwise filters as they give a good trade-off between image deblurring accuracy and computational cost. Hence, only the regularization parameter and nonlinear functions are trained. From the 5th to 7th stage, we use 48 filters each of size 7×7. At each stage, we use 400 training images from the Berkeley segmentation dataset [1] without cropping. Since real blur data is limited, we synthetically generated motion blur kernels with size 27×27 using [3]. We add different amounts of white Gaussian noise, $\sigma \in \{2.55, 3.875, 5.1, 6.375, 7.65, 8.925, 10.2\}$, to the blurry images. For each stage different image blurs are used to avoid overfitting. We optimize using 150 iterations of limited memory BFGS [18]. Greedy training takes 1.5 days and joint training takes half a day with one Titan X GPU. The code, trained model, dataset and other supplemental material will be available on the authors' webpage. Figure 3 shows the 48 filters of the 5th stage. Most filters resemble directional derivative filters, similar to those learned by the diffusion network [4]. Figure 2 shows three representative non-linear functions in the GradNet architecture, again similar to those learned in the diffusion network.

Noise-blind deblurring. To thoroughly study our noise-adaptive approach as well as GradNet, we experiment with three different datasets. First, we use the popular datasets of Levin *et al.* [16] and Sun *et al.* [29] to assess performance with different image scales. These two datasets contain 32 test images (255×255) and 640 test images (roughly 700×900), where 8 different blur kernels from [16] are used. As the amount of blur from these kernels is somewhat limited, we furthermore test a more challenging setting. We randomly select 10 images from the Berkeley dataset [1] and test with 5 large blurs from [23]. The blurs have different sizes from 29 to 37 pixels, see Fig. 4. Note that training and test sets do not overlap, neither in images nor kernels.

Method	$\sigma \rightarrow$	2.55	5.10	7.65	10.20
FD [13] (non-blind)		30.03	28.40	27.32	26.52
RTF [24] ($\sigma = 2.55$)		**32.36**	26.34	21.43	17.33
CSF [25] (non-blind)		29.85	28.13	27.28	26.70
TNRD [4] (non-blind)		28.88	28.10	–	–
TV-L_2 (non-blind)		30.87	28.43	27.59	26.51
EPLL [37] (non-blind)		32.03	29.79	28.31	27.20
EPLL [37] + NE [36]		31.86	29.77	28.28	27.16
EPLL [37] + NA		32.16	**30.25**	**28.96**	**27.85**
TV-L_2 + NA		31.05	29.14	28.03	27.16
BD [26]		30.42	28.77	27.91	27.29
GradNet 7S		31.43	28.88	27.55	26.96

Table 1. Average PSNR (dB) on 32 test images from [16].

Method	$\sigma \rightarrow$	2.55	5.10	7.65	10.20
FD [13] (non-blind)		30.79	28.90	27.86	27.14
EPLL [37] (non-blind)		32.05	29.60	28.25	27.34
CSF [25] (non-blind)		30.88	28.60	27.65	26.97
TNRD [4] (non-blind)		30.03	28.79	–	–
EPLL [37] + NE		32.02	29.60	28.25	27.34
EPLL [37] + NA		**32.18**	**30.08**	**28.77**	**27.81**
TV-L_2 + NA		30.07	28.59	27.60	26.89
GradNet 7S		31.75	29.31	28.04	27.54

Table 2. Average PSNR (dB) on 640 test images from [29].

We blur the test images and add 1%, 2%, 3%, and 4% noise (*i.e.*, $\sigma = 2.55,\ 5.10,\ 7.65,\ 10.20$). Additionally, we quantized the intensities of the noisy blurry observations to 8-bit to make them more realistic. All results are measured using the PSNR (see supplementary material for SSIM [31]).

Table 1 shows the performance on the dataset of Levin *et al*. [16]. Algorithms are divided into three classes: noise non-blind (top), noise estimation + non-blind (middle), and noise-blind (bottom). The non-blind experiments comprise 6 approaches: Fast Deconvolution [13] (FD) and TV-L_2 are well tuned for each noise level and EPLL [37] is tested with known ground truth noise level. For CSF [25], we use the official code to train different models for each noise level. We strictly follow the greedy + joint training mechanism to obtain the best performing model for each noise level. Since there is no available deblurring code for TNRD [4], we modified our code by removing preconditioning and noise adaptivity and then trained for two exemplary noise levels, thus ensuring best performance per noise level. For Regression Tree Fields (RTF) [24], we use the only available pre-trained model ($\sigma = 2.55$). We observe that RTFs only perform well for the noise level on which they are trained. For other noise levels, the performance drops significantly. Notice that our noise-blind method GradNet 7S performs better than CSF and our implementation of TNRD, which are non-blind and custom-trained for each noise level.

Method	$\sigma \rightarrow$	2.55	5.10	7.65	10.20
FD [13] (non-blind)		24.44	23.24	22.64	22.07
EPLL [37] (non-blind)		25.38	23.53	22.54	21.91
RTF [24] ($\sigma = 2.55$)		**25.70**	23.45	19.83	16.94
CSF [25] (non-blind)		24.73	23.61	22.88	22.44
TNRD [4] (non-blind)		24.17	23.76	–	–
EPLL [37] + NE [36]		25.36	23.53	22.55	21.90
EPLL [37] + NA		25.57	23.90	22.91	22.27
TV-L_2 + NA		24.61	23.65	22.90	22.34
GradNet 7S		25.57	**24.23**	**23.46**	**22.94**

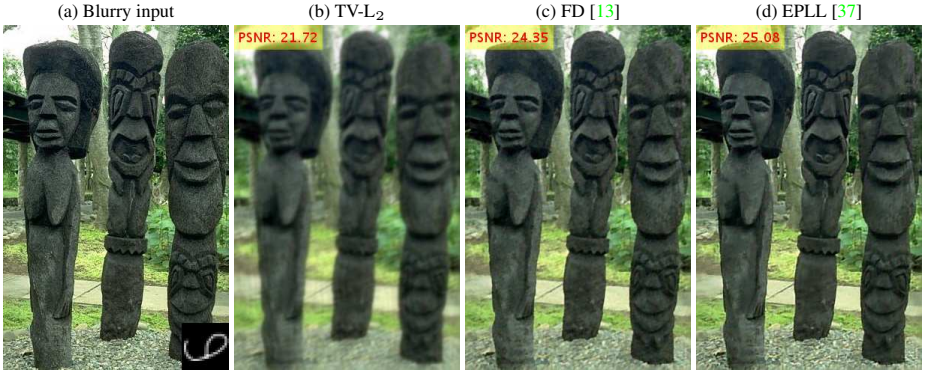
Table 3. Average PSNR (dB) on 50 test images from the Berkeley segmentation dataset [1] with large blurs (Fig. 4).

Method	size $\rightarrow$	128^2	256^2	512^2	1024^2	2048^2
FD [13]		0.05s	0.08s	0.13s	0.53s	2.3s
CSF [25]		0.06s	0.11s	0.28s	1.35s	5.44s
EPLL [37]		13s	54s	185s	860s	>1h
TV-L_2		0.26s	0.86s	2.8s	17.2s	63s
BD [26]		7min	26min	40min	>1h	–
FD [13] + NE [36]		0.35s	0.50s	0.99s	3.74s	15.8s
CSF [25] + NE [36]		0.36s	0.53s	1.14s	4.56s	19.8s
GradNet 7S		0.07s	0.24s	0.78s	3.62s	14.8s

Table 4. Execution time for different algorithms. All methods are based on Matlab implementations and tested on the same platform (Intel Core i7, quad-core at 2.4GHz).

To assess the effect of pre-estimating the noise level (NE), we use the approach of [36] and use the estimated noise level to adapt EPLL. Finally, the noise-blind experiments rely on 5 different settings: First, we extend two widely used non-blind techniques, EPLL and TV-L_2, to the noise-blind case using our noise-adaptive (NA) formulation (see supplementary material for details). We find that our noise-adaptive formulation not only enables existing techniques to deal with the noise-blind case. Importantly, the results also compare favorably to the known-noise case. For EPLL our noise-adaptive formalism improves the performance significantly by 0.3–0.7dB over the non-blind setting despite the fact that we solve a more challenging problem. This is because the optimal λ does not depend just on image noise, but more generally on a combination of image noise and approximations made by the image prior. Put differently, a prior captures the statistics of a whole set of images, which is not necessarily the best choice for a specific image. Our adaptive λ^τ based on the regularized image residual (Eq. 23) addresses this and outperforms any fixed λ. For TV-L_2 the improvement is equally significant, with improvements in the same range. Additionally, we show the result of Bayesian deblurring (BD) [26] and our GradNet. While GradNet does not quite reach the performance level of EPLL + NA, it is 2 orders of magnitude faster.

(a) Blurry input (b) TV-L_2 (c) FD [13] (d) EPLL [37]



(e) Ground Truth (f) GradNet 1 stage (g) GradNet 4 stage (h) GradNet 7 stage

Figure 5. Results for 1% noise case. PSNR results are also shown at the top left corner of the estimated images. *Best viewed on screen.*

Table 2 shows the performance on the dataset of Sun *et al*. [29]. We omit Bayesian deblurring [26] here, since it does not scale well to large images. Results show that our noise-adaptive approaches compare favorably to state-of-the-art priors (EPLL) with separate noise estimation.

Results in Table 3 show that our GradNet is more robust to large blurs [23] and outperforms EPLL at all noise levels, on average by 0.75dB. We posit that discriminative training may enable GradNet to better cope with this challenging setting. We also combined EPLL with our noise-adaptive formulation, which again improves the performance.

In all three experimental settings, our noise-adaptive approach consistently improves the performance of existing priors, even compared to the non-blind case, which means that our noise adaptation is robust to image scales, noise levels, and blur kernels. Figs. 5 shows qualitative and quantitative results with 1% noise. Compared to competing methods, GradNet handles boundaries better and also restores the ground part of the image more faithfully.

Execution time. Table 4 shows a comparison of execution times. We see that GradNet scales well to large images. Although FD and CSF are fast, noise estimation is quite slow, which is a bottleneck for further efficiency improvements. However, we are free of this issue, since our approach automatically adapts to the noise level. Another potential benefit of our model is that it is highly parallelizable and well-suitable for computation on the GPU. [4] has shown that by going from CPU to GPU, their approach can be sped up around 100 times. Since we are using a similar architecture, we believe that our network can also enjoy a significant GPU speed-up. We leave this as future work.

8. Conclusion

Noise is an unavoidable image degradation that must be accounted for in image restoration and in particular in image deblurring. We focused on the practical case where a full characterization of noise is not available and must be estimated. We showed that a direct application of MAP leads to a degenerate solution and proposed instead to substitute the 0-1 loss with a more general family of smooth loss functions. While using general loss functions may lead to infeasible high-dimensional integrals or computationally intensive methods, we derive simple bounds that can be computed analytically in closed form. This leads to a novel method for noise-adaptive deblurring, which can be efficiently implemented as a neural network. The noise adaptation leads to significant performance boosts in the noise-blind *and* known-noise case. The efficient GradNet yields state-of-the-art performance even with large blurs.

Acknowledgements. MJ and PF acknowledge support from the Swiss National Science Foundation on project 200021_153324. SR was supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Grant Agreement No. 307942.

References

[1] P. Arbelaez, M. Maire, C. C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE TPAMI*, 33(5):898–916, 2011. 6, 7

[2] A. Barbu. Training an active random field for real-time image denoising. *IEEE TIP*, 18(11):2451–2462, Nov. 2009. 1

[3] G. Boracchi and A. Foi. Modeling the performance of image restoration from motion blur. *IEEE TIP*, 21(8):3502–3517, 2012. 6

[4] Y. Chen and T. Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *arXiv:1508.02848*, 2015. 2, 5, 6, 7, 8

[5] S. Cho and S. Lee. Fast motion deblurring. *ACM Trans. Graph.*, 28(5):145:1–145:8, 2009. 2

[6] A. De Stefano, P. R. White, and W. B. Collis. Training methods for image noise level estimation on wavelet components. *EURASIP J. Adv. Sig. Proc.*, 2004(16):2400–2407, 2004. 1, 2

[7] J. Domke. Generic methods for optimization-based modeling. In *AISTATS*, 2012. 1

[8] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81:425–455, 1994. 1, 2

[9] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graph.*, 25(3):787–794, 2006. 2

[10] D. R. Hunter and K. Lange. A tutorial on MM algorithms. *Amer. Statist*, pages 30–37, 2004. 4

[11] J. Jancsary, S. Nowozin, and C. Rother. Loss-specific training of non-parametric image restoration models: A new state of the art. In *ECCV*, 2012. 2

[12] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. *ACM Trans. Graph.*, 29(4):30:1–30:9, 2010. 1

[13] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-Laplacian priors. In *NIPS*, 2009. 1, 2, 7, 8

[14] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.*, 26(3):70, 2007. 1, 2

[15] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, 2009. 1, 2

[16] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *CVPR*, 2011. 2, 5, 6, 7

[17] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang. Noise estimation from a single image. In *CVPR*, 2006. 1, 2

[18] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Program.*, 45(1–3):503–528, 1989. 6

[19] X. Liu, M. Tanaka, and M. Okutomi. Single-image noise level estimation for blind denoising. *IEEE TIP*, 22(12):5226–5237, 2013. 1, 2

[20] T. Michaeli and M. Irani. Blind deblurring using internal patch recurrence. In *ECCV*, 2014. 2

[21] G. Papandreou and A. Yuille. Perturb-and-MAP random fields: Using discrete optimization to learn and sample from energy models. In *ICCV*, pages 193–200, 2011. 3

[22] D. Perrone and P. Favaro. Total variation blind deconvolution: The devil is in the details. In *CVPR*, 2014. 2

[23] K. Schelten, S. Nowozin, J. Jancsary, C. Rother, and S. Roth. Interleaved regression tree field cascades for blind image deconvolution. In *WACV*, 2015. 6, 8

[24] U. Schmidt, J. Jancsary, S. Nowozin, S. Roth, and C. Rother. Cascades of regression tree fields for image restoration. *IEEE TPAMI*, 38(4):677–689, 2016. 1, 2, 7

[25] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *CVPR*, 2014. 1, 2, 5, 7

[26] U. Schmidt, K. Schelten, and S. Roth. Bayesian deblurring with integrated noise estimation. In *CVPR*, 2011. 1, 2, 4, 7, 8

[27] C. J. Schuler, H. C. Burger, S. Harmeling, and B. Schölkopf. A machine learning approach for non-blind image deconvolution. In *CVPR*, 2013. 2

[28] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Trans. Graph.*, 27(3), 2008. 2

[29] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *ICCP*, 2013. 2, 6, 7, 8

[30] Y. Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for Total Variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008. 1, 2

[31] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 7

[32] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *ECCV*, 2010. 2

[33] L. Xu, J. S. J. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *NIPS*, 2014. 2

[34] L. Yuan, J. Sun, L. Quan, and H. Shum. Image deblurring with blurred/noisy image pairs. *ACM Trans. Graph.*, 26(3):1, 2007. 2

[35] V. Zlokolica, A. Piurica, and W. Philips. Noise estimation for video processing based on spatial-temporal gradient histograms. *IEEE Signal Processing Letters*, 13(6):337–340, June 2006. 1, 2

[36] D. Zoran and Y. Weiss. Scale invariance and noise in natural images. In *ICCV*, 2009. 1, 7

[37] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, 2011. 1, 2, 4, 7, 8