IPOL IMAGE PROCESSING ON LINE

Published in Image Processing On Line on 2019–03–12.
Submitted on 2018–12–15, accepted on 2019–02–14.
ISSN 2105–1232 © 2019 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2019.243

Blind Image Deblurring using the ℓ_0 Gradient Prior

Jérémy Anger[1], Gabriele Facciolo[1], Mauricio Delbracio[2]

[1] CMLA, ENS Paris-Saclay, France ({anger,facciolo}@cmla.ens-cachan.fr)
[2] IIE, Universidad de la República, Uruguay (mdelbra@fing.edu.uy)

Communicated by Jean-Michel Morel and Miguel Colom *Demo edited by* Jérémy Anger

Abstract

Many blind image deblurring methods rely on unnatural image priors that are explicitly designed to restore salient image structures, necessary to estimate the blur kernel. In this article, we analyze the blur kernel estimation method introduced by Pan and Su in 2013 that uses an ℓ_0 prior on the gradient image. We present deconvolution results using the estimated blur kernels. Our experiments show the effectiveness of the method as well as some of its shortcomings.

Source Code

The C++ source code, the code documentation, and the online demo are accessible at the IPOL web page[1] of this article. Compilation and usage instruction are included in the `README.txt` file of the archive.

Keywords: deblurring; blind deconvolution; blur kernel estimation

1 Introduction

Blind image deblurring is an ill-posed image restoration problem which aims to restore a sharp image given a blurry one. Motion blur is a specific type of blur that occurs when there is relative motion between the camera and the target scene during the exposure time. When the camera focal length is large and there is no in-plane rotation, the image formation process can be expressed as the convolution between the sharp image u and a latent blur kernel k leading to

$$v = u * k + n, \tag{1}$$

where $*$ denotes the convolution[2], and n models acquisition noise. The goal of blind image deblurring is to restore the image u without knowing k before-hand. Most methods propose a two step restoration: first estimating the blur kernel k and then applying a non-blind deconvolution algorithm [6, 5, 11, 15]. While many methods assuming a uniform blur can be extended to a non-uniform model [27, 11, 9], this extension comes at the price of a non-negligible computational cost with, in general, only a minor quality improvement [13, 12].

[1] https://doi.org/10.5201/ipol.2019.243
[2] As explained in the following, we model blur as circular convolution for numerical convenience.

Notation. We use the discrete formalism for images. Let u be a digital image of size $M \times N$ defined on the grid $\Omega = \{0, \dots, M-1\} \times \{0, \dots, N-1\}$. The discrete Fourier transform of u, $\mathcal{F}(u) = \hat{u}$ is defined as

$$\hat{u}_{k,l} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} u_{m,n} e^{-2\pi i \left(\frac{mk}{M} + \frac{nl}{N}\right)}, \tag{2}$$

for $(k, l) \in \Omega$. The inverse discrete Fourier transform of $\hat{u} = \mathcal{F}(u)$ is defined as

$$\left[\mathcal{F}^{-1}(\hat{u})\right]_{m,n} = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} \hat{u}_{k,l} e^{2\pi i \left(\frac{mk}{M} + \frac{nl}{N}\right)}. \tag{3}$$

From these definitions, we have that $\mathcal{F}^{-1} \circ \mathcal{F}(u) = u$, and denoting by $*$ the circular convolution operator, we have the well-known convolution theorem: $u * k = \mathcal{F}^{-1}(\mathcal{F}(u) \cdot \mathcal{F}(k))$.

Discrete image gradients are denoted by $\nabla u = (\partial_x u, \partial_y u)$ and are computed using forward difference discretization of the derivatives. Let $d = [0, -1, 1]$, then, we have

$$\partial_x u = d * u \tag{4}$$
$$\partial_y u = d^T * u, \tag{5}$$

where d and d^T are considered 2D filters (horizontal and vertical respectively) for the convolution. The discrete derivatives are defined on Ω by periodic extension. We note $\nabla^c u$ the image gradient defined on Ω with constant extension (constant padding) instead of periodic extension. In the constant extension case we have, $(\partial_x^c u)_{M-1,n} = 0$ and $(\partial_y^c u)_{m,N-1} = 0$ for any image u defined on Ω.

The divergence of a field $\vec{w} = (w_x, w_y)$ is defined as $\operatorname{div} \vec{w} = -(\partial_x^* w_x + \partial_y^* w_y)$ where ∂_x^* and ∂_y^* are defined by

$$\partial_x^* u = \breve{d} * u \tag{6}$$
$$\partial_y^* u = \breve{d}^T * u, \tag{7}$$

and $\breve{d} = \mathsf{flip}(d) = [1, -1, 0]$. This definition of $-\partial_x^*$ corresponds to the backward difference, this leads to a centered difference scheme when computing $\operatorname{div} \nabla u$.

Method overview. In this paper, we describe and analyze the variational blind deconvolution method introduced by Pan et al. [16] and later extended in [15, 17]. This method proposes to estimate the blur kernel by iteratively solving the following variational problems

$$u^{(t+1)} = \arg\min_u \|u * k^{(t)} - v\|^2 + \lambda \|\nabla u\|_0 \tag{8}$$
$$k^{(t+1)} = \arg\min_k \|\nabla u^{(t+1)} * k - \nabla v\|^2 + \gamma \|k\|^2, \tag{9}$$

where

$$\|\nabla u\|_0 = \#\big\{(m,n) \in \Omega : |(\partial_x u)_{m,n}| + |(\partial_y u)_{m,n}| > 0\big\} \tag{10}$$

is the ℓ_0 norm of the gradient of u. The first step estimates a sharp image u given a previous estimation of the kernel k, using an ℓ_0 gradient prior on the image u. The second step estimates the blur kernel given the sharp and blurry images.

The ℓ_0 gradient prior guides the estimation towards an *unnaturally* sharp image, similarly to Xu et al. [31], and relates to the notion of unnatural image representation that is essential for blind

Method	Image prior	Sharp prediction	Kernel estimation	Structure map	Upscale
Fast L0 [16]	$\|\nabla u\|_0$	ADM	Fourier	Yes	
Text L0 [15]	$\|\nabla u\|_0 + \|u\|_0$	Quadratic penalty	Conjugate gradient	No	Kernel*
Dark L0 [17]	$\|\nabla u\|_0 + \|D(u)\|_0$	Quadratic penalty	Conjugate gradient	No	Kernel*
Our method	$\|\nabla u\|_0$	Quadratic penalty	Fourier	No	Image

Table 1: Comparison of methods based on the ℓ_0 gradient prior. (*) In [15] and [17], authors indicate that the upsampling from one scale to the next one is performed by upsampling the image. However, careful analysis of the source code indicates that it is the kernel that is upscaled.

deblurring methods. The unnaturalness comes from the fact that natural images contain textures. Hence, their gradient is not sparse and has a heavy tailed (typically Laplacian) distribution. In Section 2.2, we will show that the ℓ_0 prior includes an explicit gradient thresholding step resulting directly from the variational formulation, thus making a link to previous kernel estimation methods [5, 29] that required an edge selection heuristic. Table 1 offers a comparison of the ingredients present in the various kernel estimation methods based on the ℓ_0 gradient prior.

Paper organization. Section 2 contains a description of the method along with a discussion of modeling choices and the effect of its most relevant parameters. In Section 3, we experimentally explore the performance of the method with respect to different levels of noise and in real and synthetic setups. We conclude in Section 4.

2 Method

The alternate minimization described in (8) and (9) iterates between sharp image prediction and kernel estimation. In order to estimate large kernels, a multi-scale scheme is employed. Furthermore, at the start of the minimization the regularization weight λ is set to a high value in order to ensure restoration of the main edges and the removal of details. This allows to estimate a coarse kernel, which is then refined by lowering λ throughout the iterations similarly to Cho et al. [5] until the value λ_{min} is reached.

Algorithm 1 describes the general steps of the method. The method takes the blurry image as input, as well as the kernel size k_s as parameter. This size specifies an upper bound for the size of the actual blur kernel. At each scale, u and k are refined through $N_{\text{iter}} = 5$ iterations.

2.1 Preprocessing

Before iterating, the blurry image is preprocessed with three steps to increase speed and reduce artifacts:

1. First, the input image is converted to grayscale by averaging channels. Since we assume that the blur kernel is identical for each channel of the image, working in grayscale allows to reduce the noise by a factor $\sqrt{3}$ and reduces the computational cost.

2. The image is then cropped to an optimized size for the Fast Fourier Transform[3]. We observed that the removal of a few rows and/or columns of pixels can lead to a significant speed-up over the whole estimation. Note that since this crop is only applied at the finest scale, the optimality

[3]Optimized sizes are powers of 2, 3, 5 or 7 [24].

Algorithm 1: MultiscaleKernelEstimation

input : blurry image v, kernel size k_s, weights λ $(= 4 \cdot 10^{-3})$, $\lambda_{\min}$ $(= 10^{-4})$ and γ $(= 20)$
output: estimated kernel k

```
v = preprocess(v)                                   grayscale conversion, crop, edgetaper, normalize
for s from N_scales − 1 to 0 do
    v^(s) = downscale v to scale s
    if s == N_scales − 1 then
        u^(s,0) = v
    for t from 1 to N_iter do
        k^(y) = KernelEstimation(v^(s), u^(s,t−1), k_s, γ)        Algorithm 3
        u^(s,t) = SharpPrediction(v^(s), k^(t), λ)                Algorithm 2
        λ = max(λ/1.1, λ_min)                                     λ decay
    if s > 0 then
        u^(s−1,0) = upscale the sharp image u^(s) to scale s − 1
```

of the size might not be respected for coarser scales. Cho et al. [5] also use optimized sizes for the Fast Fourier Transform but by padding the image instead of cropping, which implies the introduction of invalid data.

3. Finally, an edge tapering procedure [21] is applied in order to reduce artifacts due to the image boundary discontinuities that arise since the circular boundary condition is not valid on real images. This tapering is performed by circularly blurring the image using a uniform kernel k_u and blending the result with the original image v, keeping the original image in the central part of the image domain and the re-blurred image $k_u * v$ near the boundary.

Additionally, the input image is normalized between 0 and 1.

2.2 Sharp Image Prediction

The sharp image prediction step consists in deconvolving the blurry image using the kernel k estimated from the previous iteration. Since the kernel contains errors, a non-blind deconvolution algorithm designed to match with natural image statistics is not adapted [28] and a specific regularization is needed so that the recovery of edges is favored against a blurry image. This is typically achieved using a gradient sparsifying prior (see Herley and Rickard [10] for a study on measures of sparsity).

Several kernel estimation methods, such as the classical work by Chan and Wong [4], use Total Variation (TV) to enforce a sparse gradient prior on the latent sharp image. Yet, Levin et al. [14], and Wipf et al. [28] have shown that an ℓ_p norm with p close to 0 is required in the case of deblurring in order to avoid the trivial solution ($u = v$ and $k = \delta$), as shown in Figure 1. Perrone and Favaro [20], in a careful and rigorous analysis of Chan and Wong TV deconvolution algorithm [4], revealed that the successfulness of this TV based algorithm lies in the original specific implementation. In fact, Perrone and Favaro show that if all the constrains are imposed simultaneously the TV blind deconvolution algorithm would never leave the no-blur trivial solution. This is due to the two opposing effects of blur: it reduces the gradient sparsity (which increases $\|\nabla \cdot\|_p^p$) but also reduces the variance of the image (which decreases $\|\nabla \cdot\|_p^p$). Hence, it is desirable that the image prior used during the sharp image prediction is less sensitive to the image variance and more sensitive to its sparsity, as is the case with the ℓ_0 norm. Note that the ℓ_0 gradient prior was originally introduced for image smoothing by Xu et al. [30].

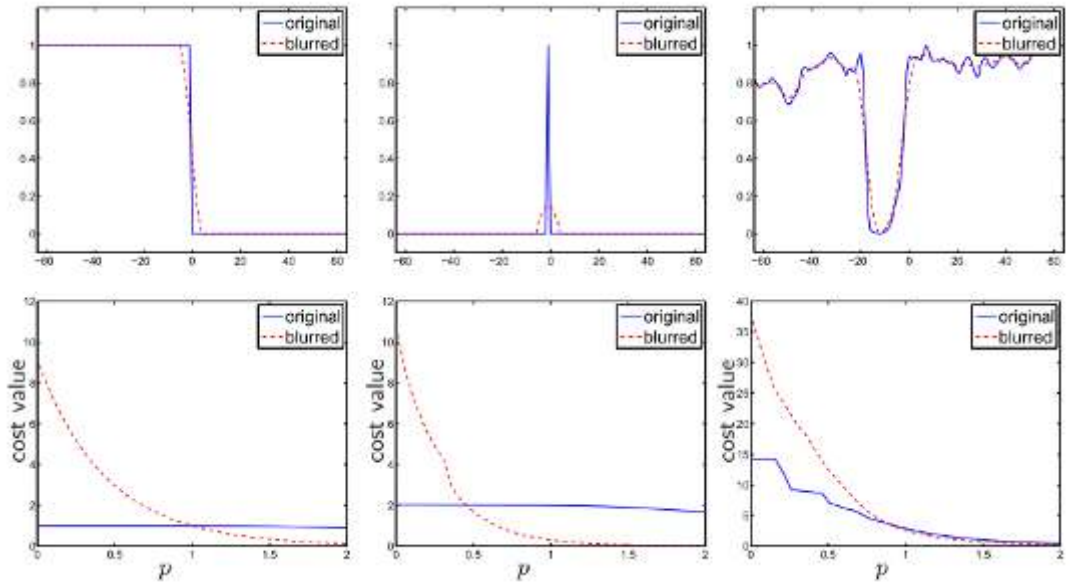


Figure 1: Illustration from Wipf and Zhang [28] with permission by the authors. The top row displays three 1D signals with sharp and blurry versions. The bottom row indicates the value of the cost function $\min_x \frac{1}{\lambda}\|k * x - y\|_2^2 + \sum_i |x_i|^p$ as a function of p.

Having chosen an image prior adapted for deblurring, we can model the sharp image prediction as the following variational problem

$$\arg\min_u \|u * k - v\|^2 + \lambda\|\nabla u\|_0. \tag{11}$$

Following Xu et al. [30], we introduce a splitting variable $\vec{g}$ and obtain the equivalent energy as β tends to infinity

$$\arg\min_{u,\vec{g}} \|u * k - v\|^2 + \beta\|\nabla u - \vec{g}\|^2 + \lambda\|\vec{g}\|_0. \tag{12}$$

Equation (12) can be solved using an iterative continuation scheme, increasing β at each iteration

$$\vec{g}^{(t+1)} = \arg\min_g \beta\|\nabla u^{(t)} - \vec{g}\|^2 + \lambda\|\vec{g}\|_0 \tag{13}$$

$$= T_{\sqrt{\lambda/\beta}}(\nabla u^{(t)}), \tag{14}$$

$$u^{(t+1)} = \arg\min_u \|u * k - v\|^2 + \beta\|\nabla u - \vec{g}^{(t+1)}\|^2 \tag{15}$$

$$= \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(k)}\mathcal{F}(v) + \beta\mathcal{F}(-\operatorname{div}\vec{g}^{(t+1)})}{|\mathcal{F}(k)|^2 + \beta(|\mathcal{F}(d)|^2 + |\mathcal{F}(d^T)|^2)}\right), \tag{16}$$

where $\overline{x}$ denotes the complex conjugate of x and T_α is the hard-thresholding operator defined for a vector field $\vec{w}$ as

$$T_\alpha(\vec{w})_{m,n} = \begin{cases} (0,0) & \text{if } |\vec{w}_{m,n}| < \alpha, \\ \vec{w}_{m,n} & \text{otherwise.} \end{cases} \tag{17}$$

Proof of the u update. We denote by $E(u)$ the cost function minimized in Equation (15)

$$E(u) = \|k * u - v\|^2 + \beta\|\nabla u - \vec{g}\|^2. \tag{18}$$

Algorithm 2: SharpPrediction

input : blurry image v, kernel k, regularization weight λ
output: estimated sharp image u
$\beta = 2\lambda$
$u^{(0)} = v$
$t = 1$
while $\beta < 10^5$ **do**
  $\vec{g}^{(t)} = T_{\sqrt{\lambda/\beta}}(\nabla u^{(t-1)})$ *Equation (14)*
  $u^{(t)} = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(k)}\mathcal{F}(v)+\beta\mathcal{F}(-\operatorname{div}\vec{g}^{(t)})}{|\mathcal{F}(k)|^2+\beta(|\mathcal{F}(d)|^2+|\mathcal{F}(d^T)|^2)}\right)$ *Equation (16)*
  $\beta = 2\beta$
  $t = t + 1$

Taking the gradient of E with respect to u, we obtain

$$\frac{\partial E}{\partial u} = 2\breve{k} * (k * u - v) + 2\beta\left(\partial_x^*(\partial_x u - \vec{g}_x) + \partial_y^*(\partial_y u - \vec{g}_y)\right) \tag{19}$$

$$= 2\breve{k} * k * u - 2\breve{k} * v + 2\beta(\partial_x^*\partial_x u + \partial_y^*\partial_y u) + 2\beta\operatorname{div} g. \tag{20}$$

Since E is quadratic, its minimum is attained when $\frac{\partial E}{\partial u} = 0$. Thus we have

$$\left(\breve{k} * k + \beta(\breve{d} * d + \breve{d}^T * d^T)\right) * u = \breve{k} * v - \beta\operatorname{div} g. \tag{21}$$

Since the convolution with k and the gradients assume circular boundary conditions, the linear system (21) can be expressed in the Fourier domain

$$\left(|\mathcal{F}(k)|^2 + \beta(|\mathcal{F}(d)|^2 + |\mathcal{F}(d^T)|^2)\right) \cdot \mathcal{F}(u) = \overline{\mathcal{F}(k)} \cdot \mathcal{F}(v) - \beta\mathcal{F}(\operatorname{div} g). \tag{22}$$

Thus we have

$$\mathcal{F}(u) = \frac{\overline{\mathcal{F}(k)} \cdot \mathcal{F}(v) - \beta\mathcal{F}(\operatorname{div} g)}{|\mathcal{F}(k)|^2 + \beta(|\mathcal{F}(d)|^2 + |\mathcal{F}(d^T)|^2)}. \tag{23}$$

$$\implies u = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(k)} \cdot \mathcal{F}(v) - \beta\mathcal{F}(\operatorname{div} g)}{|\mathcal{F}(k)|^2 + \beta(|\mathcal{F}(d)|^2 + |\mathcal{F}(d^T)|^2)}\right). \tag{24}$$

□

Proof of the $\vec{g}$ update [30]. Since the subproblem for $\vec{g}$ can be solved pixel-wise, the following problems are equivalent and the minimization reduces to an 1D optimization,

$$\left(\arg\min_{\vec{g}} \|\nabla u - \vec{g}\|_2^2 + \frac{\lambda}{\beta}\|\vec{g}\|_0\right)_i = \arg\min_{\vec{g}_i} |\nabla u_i - \vec{g}_i|^2 + \frac{\lambda}{\beta}|\vec{g}_i|^0. \tag{25}$$

This non-convex energy has two local minima: $\vec{g}_i = (0,0)$ and $\vec{g}_i = \nabla u_i$. Comparing the energy of both minima, we take the solution $\vec{g}_i = (0,0)$ when

$$|\nabla u_i - (0,0)|^2 + |(0,0)|_0 < |\nabla u_i - \nabla u_i|^2 + \frac{\lambda}{\beta}|\nabla u_i|^0. \tag{26}$$

That is, whenever

$$|\nabla u_i|^2 < \frac{\lambda}{\beta}. \tag{27}$$

Otherwise, we take $\vec{g}_i = \nabla u$, hence the energy minimizer is $\vec{g} = T_{\sqrt{\lambda/\beta}}(\nabla u)$. □

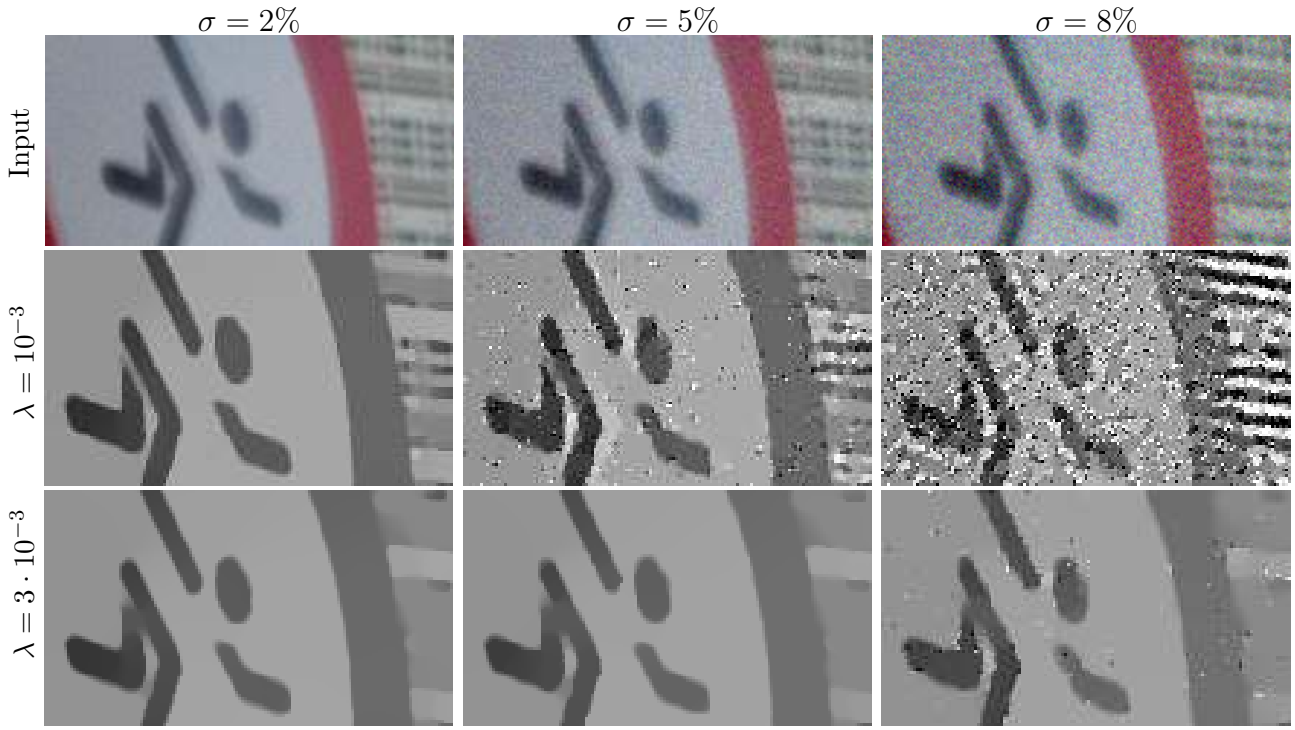


Figure 2: Sharp image prediction using the ground-truth kernel.

Algorithm 2 describes the iterative sharp image prediction procedure. Note that problem (11) is non-convex. Hence, we cannot expect to find a global minimum. Similarly, since the second step of this iterative prediction is the least square image reconstruction, it is unlikely that the gradients of the reconstructed image are still sparse in the sense of ℓ_0, since small slopes can be introduced by solving the linear system. However, the solution of these iterations can be sufficient to retrieve sharp edges for the kernel estimation.

Note that Equation (16) assumes circular boundary conditions for both the convolution and the gradient operator so that the system can be solved efficiently in the Fourier domain. A more accurate model would be to use unknown boundary conditions as in Almeida et al. [1]. However, we experimentally found that preprocessing the blurry image with an *edgetaper* step as explained in Section 2.1 was sufficient to avoid ringing at the boundaries while maintaining a fast convergence. As the goal of this step is to predict a sharp image, there is no need for a high quality image restoration.

Regularization weight λ. As shown by Equation (14) the parameter λ has an influence over the thresholding of the gradients. Depending on λ, details and noise can be removed, leaving only the most contrasted edges. If set too low, the ℓ_0 gradient prior will introduce spikes during the image prediction since they will have a small ℓ_0 cost while reducing the data-term, as shown in the central image in Figure 2. As λ is increased, more and more details are lost and the prediction becomes less noisy, as shown on the bottom row of the figure.

Implementation details. In our implementation of Algorithm 2, $\mathcal{F}(k)$, $\mathcal{F}(d)$ and $\mathcal{F}(d^T)$ are computed by zero-padding the corresponding filter up to the image size. Since k is fixed for a given estimation of u, $\overline{\mathcal{F}(k)}\mathcal{F}(v)$ and $|\mathcal{F}(k)|^2$ can be precomputed before iterating. Furthermore, $\mathcal{F}(v)$ and $|\mathcal{F}(d)|^2 + |\mathcal{F}(d^T)|^2$ can be precomputed for a given scale. Using these precomputations, the sharp image prediction only requires two FFTs per iteration.

2.3 Kernel Estimation

Once a sharp image has been estimated, it is possible to estimate the blur kernel by solving the linear problem

$$\arg\min_k \|\nabla u * k - \nabla v\|^2 + \gamma \|k\|^2. \tag{28}$$

Using a data term in a filtered domain, here the gradients in Equation (28), was observed to achieve better results than performing the estimation on intensities when considering circular boundary conditions for the convolution [5, 18, 19]. This fact can be explained by two observations: first, iterative solvers such as conjugate gradient descent benefit from working in a filtered domain since it allows a better conditioning of the system; indeed, Cho et al. [5] observed slower convergence rate when including the intensities in the data-term. Second, the errors that arise from the circular boundary condition for the convolution can be high since intensities on one side of the image can be very different from the other side. However this is not the case when working in the filtered domain where gradients have similar magnitude everywhere in the image [25]. This last point is true only if the filtered domain itself does not assume a circular boundary condition; hence to solve Equation (28) we use the gradients ∇^c with constant extension. Since the kernel is usually much smaller than the blurry image, the boundaries represent a small fraction of the data used to estimate the kernel. Thus, the errors introduced by the invalid boundary condition are small.

Assuming circular convolution, the variational problem (28) can be solved by a direct method using the discrete Fourier transform

$$k = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(\partial_x^c u)}\mathcal{F}(\partial_x^c v) + \overline{\mathcal{F}(\partial_y^c u)}\mathcal{F}(\partial_y^c v)}{|\mathcal{F}(\partial_x^c u)|^2 + |\mathcal{F}(\partial_y^c u)|^2 + \gamma}\right). \tag{29}$$

For this step, the kernel size parameter is used in order to retrieve only the part of the kernel we are interested in, thus discarding the noise in the rest of the estimation support.

Moreover, after solving (28) by (29), negative values of the kernel are set to zero and the kernel is normalized so that its values sum to one. Additionally, values below 5% of the maximum value of the kernel are set to 0 and connected components of the support of k with low intensity are filtered out. A connected component C is removed if $\sum_{(m,n)\in C} k_{m,n} < 0.1$. The kernel is then centered and renormalized.

Algorithm 3 describes the kernel estimation with these additional post-processing steps. The function crop_center extracts the central part of the image according to the kernel size k_s; for an image of size $M \times N$, this central part corresponds to $\{\frac{M-k_s}{2}, \dots, \frac{M+k_s}{2} - 1\} \times \{\frac{N-k_s}{2}, \dots, \frac{N+k_s}{2} - 1\}$. The function center_kernel shifts the kernel so that its center of mass, rounded to the nearest integer, is at the center of its support.

Regularization weight γ. The parameter γ penalizes large Fourier coefficients and tends to smooth kernels. If set too low, the estimation will be strongly affected by the noise in u. If set too high, the solution will be too smooth and kernel details will not be recovered. Figure 3 shows the kernel estimation step with different noise levels and γ values. The kernels were estimated between the blurry and noisy image and the ideal sharp image, computed using the ℓ_0 prediction step between the noise-less blurry image and the ground-truth kernel with $\lambda = 4 \cdot 10^{-3}$. For a given γ, as the noise increases, the estimation also gets noisier. If γ is increased, the estimation is less affected by noise but tends to be smoother. By default in our implementation, we set $\gamma = 20$.

Implementation detail. In Algorithm 3, since the blurry image does not change for a given scale, $\mathcal{F}(\partial_x v)$ and $\mathcal{F}(\partial_y v)$ can be precomputed so that only three FFTs are required per kernel estimation step.

Algorithm 3: KernelEstimation

input : blurry image v, estimated sharp image u, kernel size k_s, regularization weight γ
output: estimated kernel k

$k = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(\partial_x^c u)}\mathcal{F}(\partial_x^c v)+\overline{\mathcal{F}(\partial_y^c u)}\mathcal{F}(\partial_y^c v)}{|\mathcal{F}(\partial_x^c u)|^2+|\mathcal{F}(\partial_y^c u)|^2+\gamma}\right)$ *Equation (29)*

$k = \text{crop\_center}(k, k_s)$ *Impose support*

$k_{m,n} = \max(k_{m,n}, 0), \quad \forall (m,n)$ *Impose positivity*

$\text{threshold} = 0.05 \cdot \max_{m,n} k_{m,n}$

$k = T_{\text{threshold}}(k)$ *Remove low intensities*

$C =$ compute the set of the connected components of the support of k
forall the $c \in C$ **do**
  if $\sum_{(m,n)\in c} k_{m,n} < 0.1$ **then**
    forall the $(m,n) \in c$ **do**
      $k_{m,n} = 0$

$k = \text{center\_kernel}(k)$

$k_{m,n} = k_{m,n} / \sum_{m',n'} k_{m',n'}, \quad \forall (m,n)$ *Normalize*

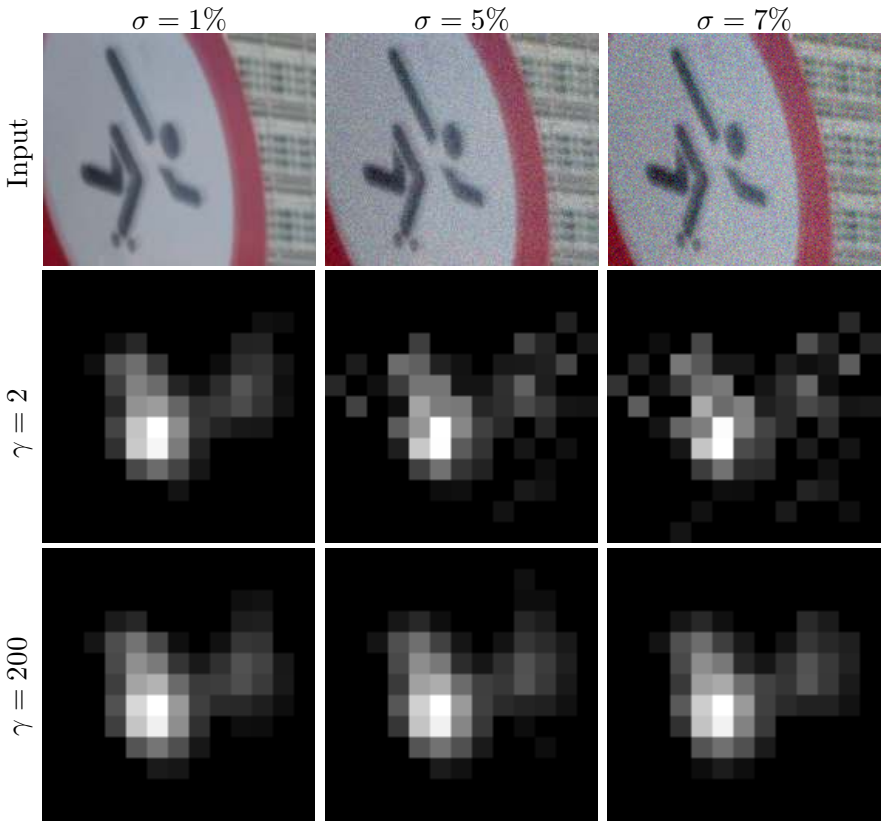
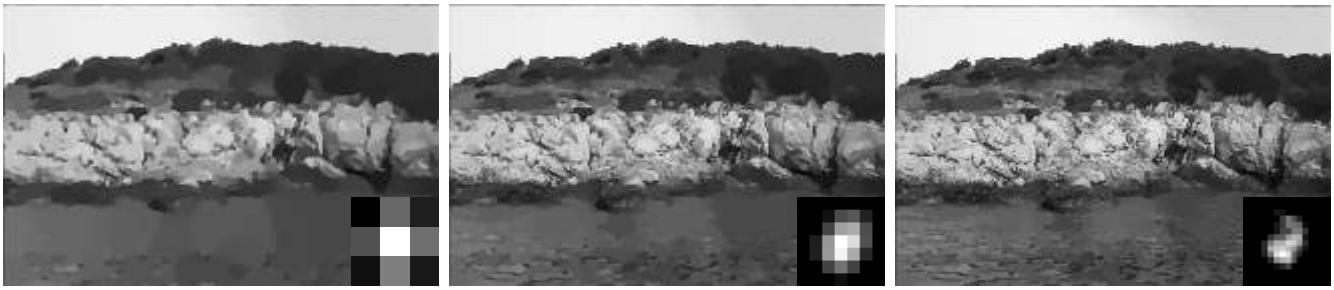


Figure 3: Kernel estimation results with different noise levels and γ values. The kernels were estimated between the blurry and noisy image (top row) and the ideal sharp image, obtained using the ℓ_0 prediction step between the noise-less blurry image and the ground-truth kernel with $\lambda = 4 \cdot 10^{-3}$.



(a) Scale 1. (b) Scale 2. (c) Scale 3.

Figure 4: Sharp image prediction and estimated kernel at the last iteration of each scale during the multiscale optimization.

2.4 Multiscale Procedure

Large kernels cannot be estimated by directly solving (8) and (9) at the finest scale without proper initialization of either the kernel k or the sharp image u. Indeed, at the finest scale, if the kernel is large the ℓ_0 gradient prior might not collapse blurry edges into a single sharp one. This issue is solved by starting the minimization at coarser scales before carrying on to finer scales, as described in Algorithm 1. A scaling ratio of 2 is used and the blurry image is downsampled until the kernel size is 3×3. For implementation convenience, at each scale, the kernel size is rounded to the next odd integer.

Once the kernel and the sharp image have been estimated at a low resolution scale, one of them needs to be up-sampled for the next scale. In Pan et al. [15], going from one scale to another is done by up-sampling the kernel. However, since motion blur kernels are non-smooth signals, up-sampling a kernel is not a well defined procedure. As images are smoother signals than kernels, the up-scaling of the sharp image is less problematic. Hence, we use the bicubic interpolation to up-sample the predicted sharp image to the next scale.

At each scale, the observed image v is down-scaled by a factor 2 using a Gaussian filtering with $\sigma = 0.8$ [22]. Figure 4 shows the sharp image prediction and kernel estimation at the end of each scale.

2.5 Non-Blind Deconvolution

As for most blind deblurring methods, the final deconvolution does not use the sharp image estimations since the image prior typically produces unnatural images. In the final step, the kernel is assumed to have been well estimated and a more natural prior is used. The well-known total variation prior is used [23], leading to the minimization problem

$$\arg\min_u \frac{\alpha}{2}\|u * k - v\|^2 + \|\nabla u\|_1. \tag{30}$$

In our implementation, this energy is minimized using the *split Bregman* algorithm [7]. The default value for the parameter α is 3000. Before deconvolution, the image is padded with symmetric extension and an edge taper is applied to reduce image boundary artifacts.

3 Experiments

In this section we evaluate the method under different conditions. We show results on a real image, on uniform and non-uniform synthetic datasets and on a very noisy case.

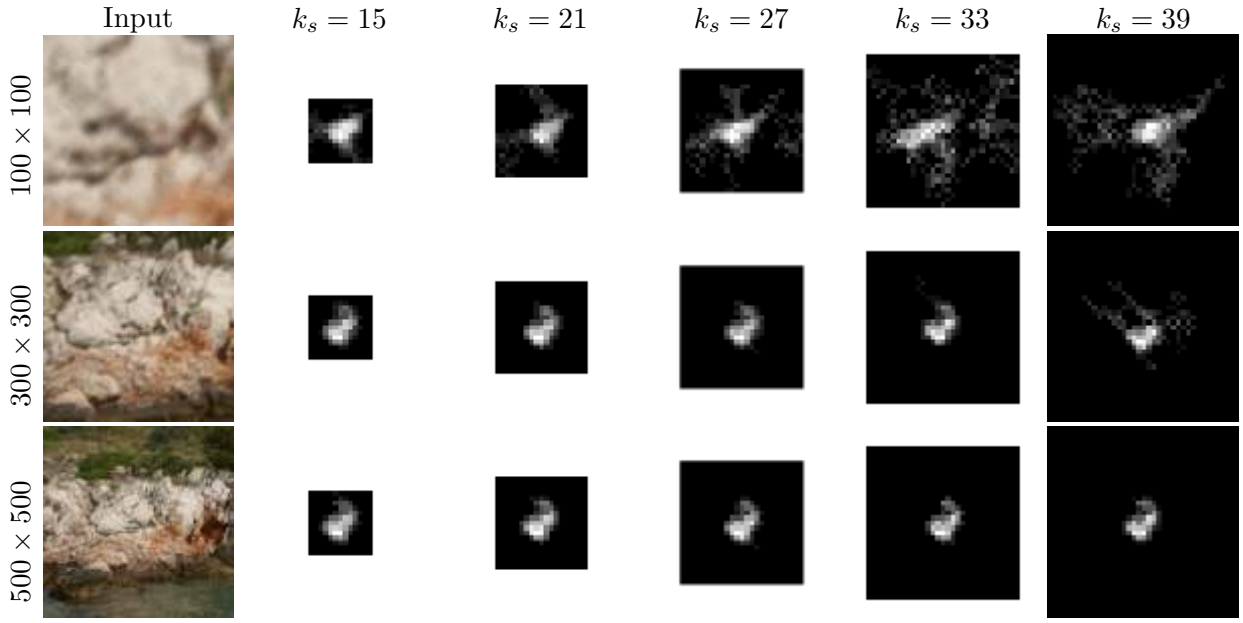


Figure 5: Kernel estimation depending on the image size and kernel support.

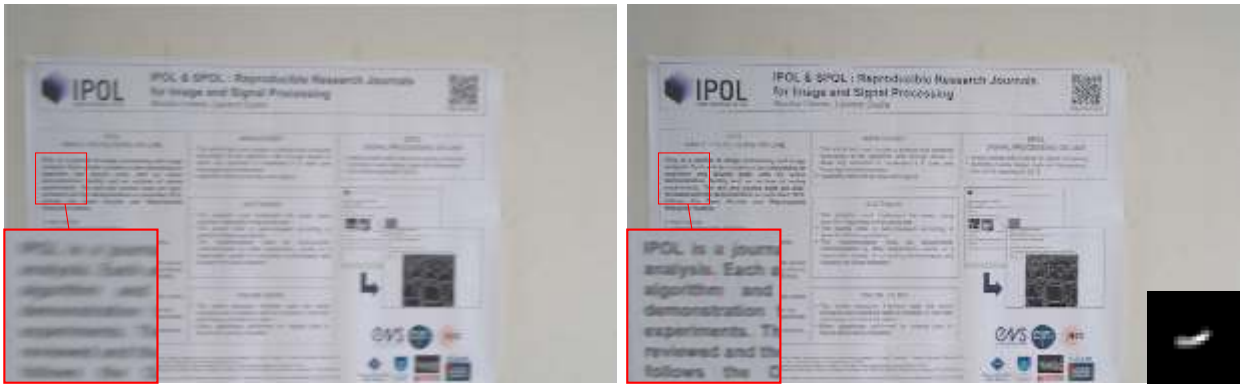


Figure 6: Result of the method on a real image.

Image and kernel size. Figure 5 shows the influence of the image size and the kernel support size for the estimation. If the image size is significantly larger than the kernel size, the estimation is accurate even if the kernel size is far from the ground-truth size. However, as the image size decreases, the kernel estimation becomes more ill-posed and an inaccurate kernel size leads to a noisier estimation.

Real image. Figure 6 shows the results of the method on a real image taken with a DxO ONE camera. While the deconvolution contains some color artifacts, the kernel is well estimated and allows to restore details. Kernel estimation and deconvolution were performed in linear space (before applying a gamma tonemap) and we set the TV regularization parameter to $\alpha = 1000$.

Uniform synthetic dataset. Figure 7 shows the kernel estimation and deconvolution results for a synthetic dataset with white Gaussian noise of standard deviation $\sigma = 1\%$. The third column shows the image deblurred using the ground-truth kernel while the last column shows the deblurring result using the estimated kernel. One can see that although the kernel estimation is not perfect, it is sufficient to deconvolve the image with a quality comparable to the one obtained by using the exact kernel.

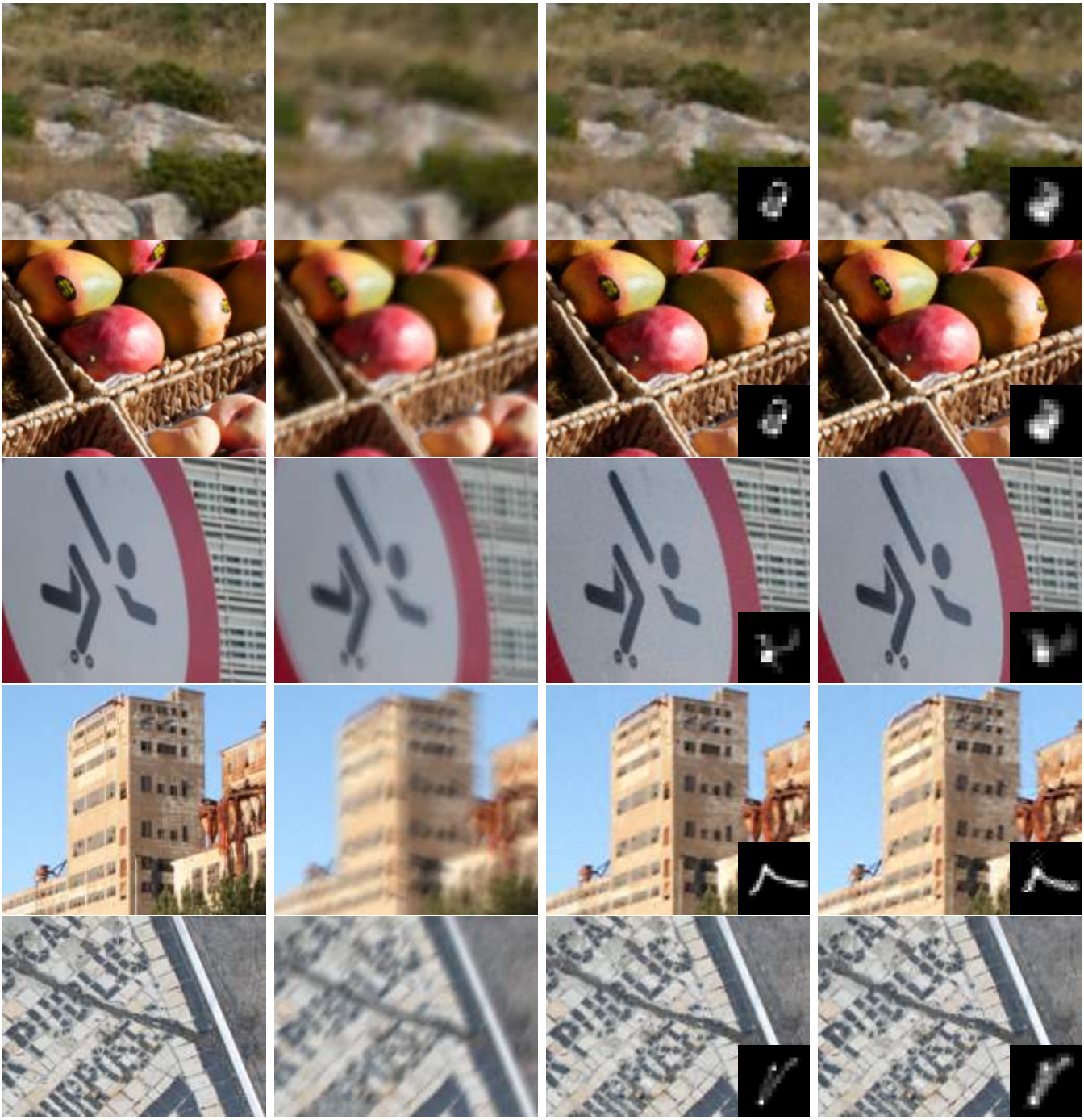


Figure 7: From left to right: original image, blurred image with Gaussian noise of standard deviation $\sigma = 1\%$, deblurred image with ground truth kernel, and deblurred image with estimated kernel. For the non-blind deconvolution, we set $\alpha = 1000$.

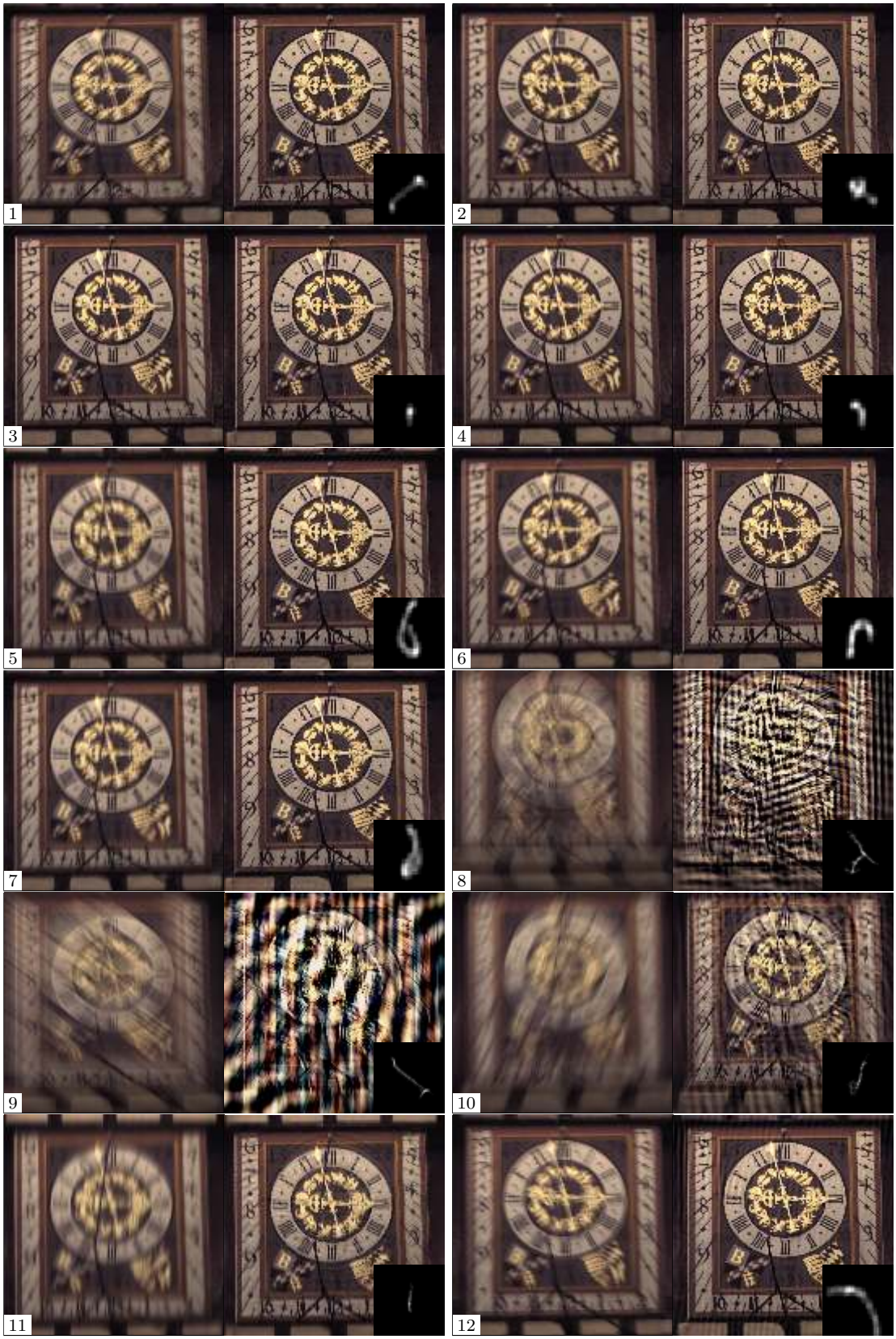


Figure 8: Results on the non-uniform dataset of Köhler et al. [12]. Each pair shows the blurry image with the deblurred one and the estimated kernel. We set all parameters to their default values except for the kernel size where images 1 to 7 and 12 are estimated with $k_s = 31$; images 8 to 11 with $k_s = 131$.

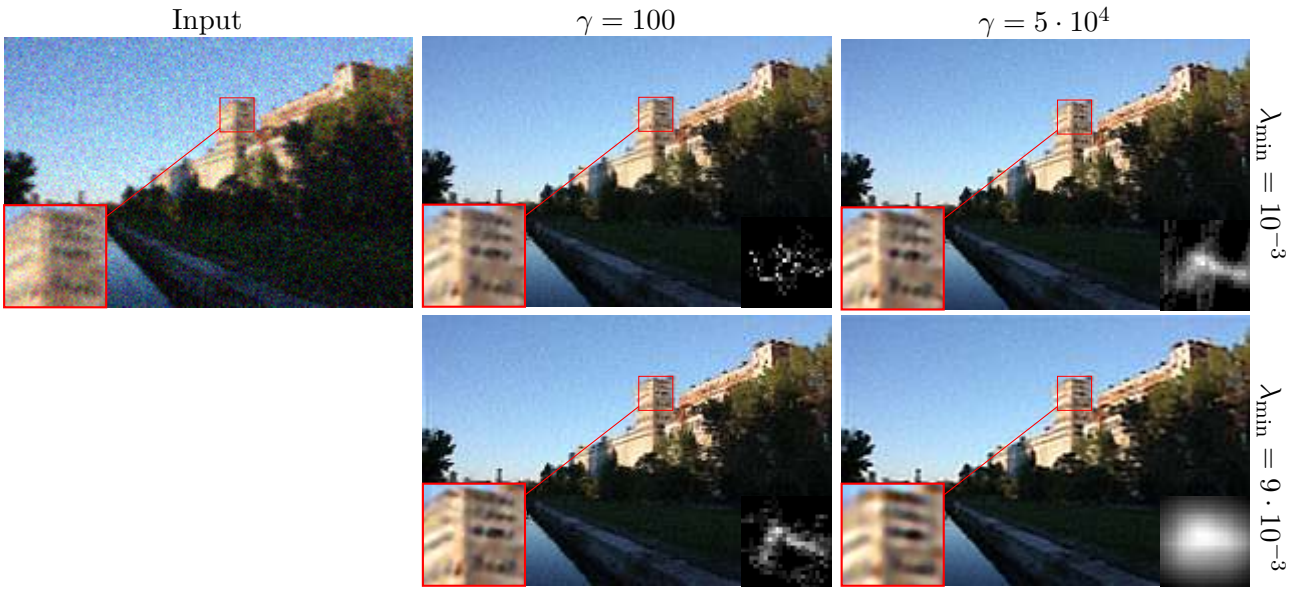


Figure 9: Blind deblurring under severe noise $\sigma = 10\%$. The regularization weight α for non-blind deconvolution was set to 50.

Non-uniform dataset. Figure 8 shows the pairs of input and deconvolution result on the non-uniform dataset of Köhler et al. [12]. We observe that large kernels are not well recovered. However, as noted in Lai et al. [13], the ℓ_0 gradient prior-based kernel estimation is quite robust to some degree of kernel non-uniformity, even if the non-blind deconvolution produces ringing near edges.

High noise level. Under high noise conditions (σ between 5% and 10%), kernel estimation becomes a very challenging problem [33, 26]. For the sharp image prediction step, the parameter $\lambda_{\min}$ allows to limit the decay of the ℓ_0 regularization weight and must be adjusted as the noise level increases. Similarly, the kernel estimation step can be adjusted through the regularization weight γ. We added white Gaussian noise of standard deviation $\sigma = 10\%$ to a synthetic blurry image and show the results in Figure 9 when varying the parameters $\lambda_{\min}$ and γ. We also reduced the data-term weight of the final non-blind deconvolution to $\alpha = 50$. We observe that both parameters $\lambda_{\min}$ and γ should be adjusted carefully to get a good result. Indeed increasing both simultaneously can lead to a performance decay, as shown in Figure 9.

Failure case. Figure 10 shows the result of the method on a real image that does not contain any prevalent edges, taken from Xu et al. [29]. The method was not able to correctly estimate the blur kernel since it is too large compared to the contrasted structures of the image.

Comparison with Goldstein et al. [8]. Goldstein et al. [8] propose the Spectral Irregularities kernel estimation (SP), which models natural images by a characteristic spectral decay corresponding to the spectral decay of edges. This model is violated when the image is blurry, allowing to estimate the modulus of the Fourier transform of the kernel. The spatial kernel is then recovered using a phase retrieval step. Conversely, the ℓ_0 gradient prior restores edges, hence both the modulus and the phase of the blurry image are used during the estimation. In short, for SP the latent sharp image needs not to contain any edges as long as the spectral decay model is respected. On the other hand, the ℓ_0 gradient prior assumes a similar model (since edges have the same spectral decay), but requires to have sparse gradients and be contrasted, thus edges to be in-phase.

To validate this hypothesis, we blurred two sharp images: the first one is a natural one, while the second is a random phase noise of the first one, thus its spectral decay is preserved but edges are

Figure 10: Failure case of the method on an image from Xu et al. [29]. Note that the described method was not able to correctly estimate the blur kernel since it is too large compared to the contrasted structures of the image.

not. We used the implementation from Anger et al. [2] to compare the method with the ℓ_0 kernel estimation. Figure 11 shows the estimated kernels and deconvolution results. We observe that ℓ_0 is more precise than Spectral Irregularities for a natural image but only recovers a coarse shape of the kernel for the random phase noise image. We also note that the method is more robust to noise, as shown in the middle row of Figure 11 where synthetic noise of standard deviation $\sigma = 1\%$ was added. Overall, the ℓ_0 gradient prior allows more precise kernel estimations for natural images than the Goldstein and Fattal method [8]. An interesting future research direction would be to combine the ℓ_0 and the power-spectrum decay priors into a single method [32].

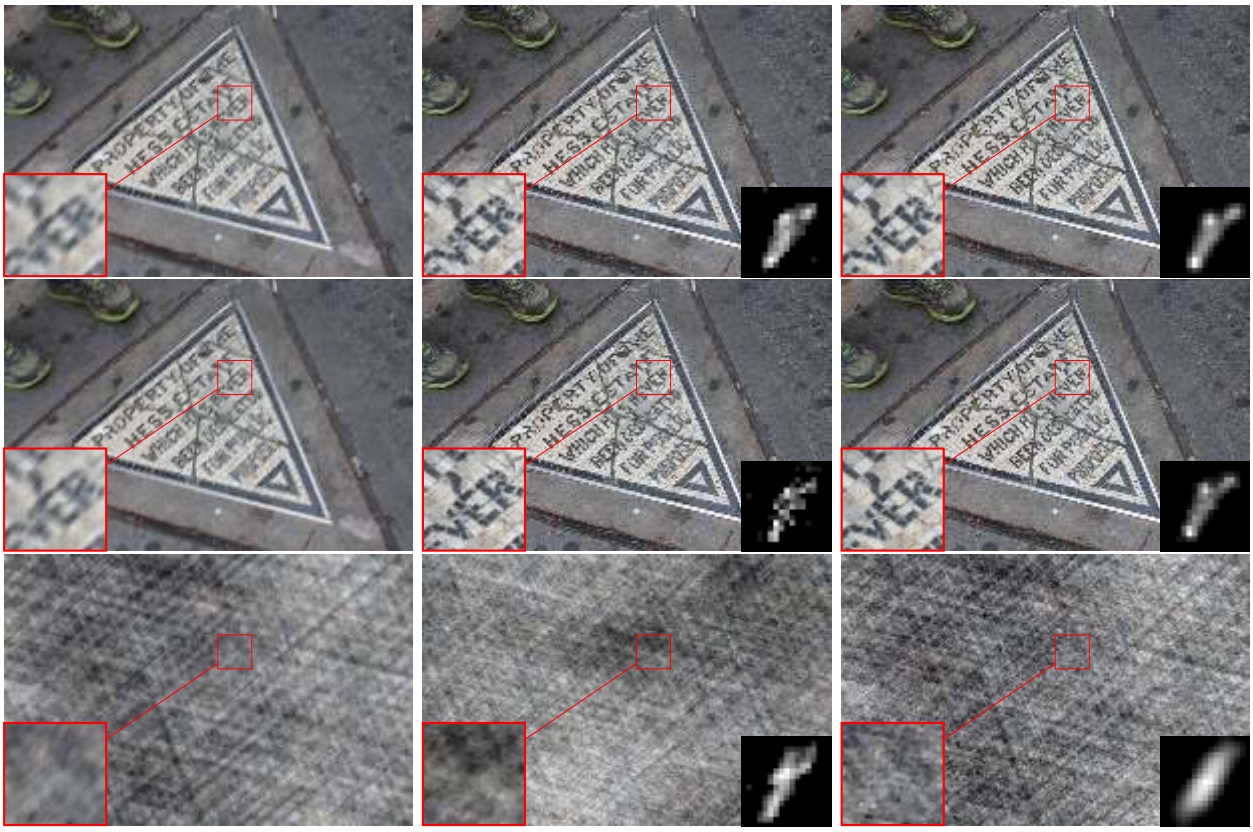


Figure 11: Comparison with the Goldstein and Fattal method [8]. From left to right: blurry image, deblurring results of Goldstein and Fattal, results of ℓ_0 gradient prior. The top row corresponds to a natural image with noise level $\sigma = 0\%$ and the middle row is the same but noise level $\sigma = 1\%$. The bottom row corresponds to a random phase noise of the latent sharp image of the top row.

4 Conclusion

In this paper we described a blind deblurring method based on the ℓ_0 gradient prior. The first step of the method is to estimate the blur kernel by alternating between a sharp image prediction using ℓ_0 and a kernel estimation. Once the kernel is estimated, a standard non-blind deconvolution method is used to obtain the deblurred image. This blur kernel estimation is carried out in a multi-scale manner.

While the method does not have convergence guarantees, we performed extensive experiments showing state-of-art performance for spatially uniform blur. Our results on a non-uniform dataset, on a synthetic example with severe noise and on real world images confirm the good performance of the method.

As a future work, we would like to extend this variational method to more realistic degradation scenarios commonly present in the image acquisition pipeline [3]. In particular, we would like to study if the kernel estimation step can benefit from a more accurate image formation model that incorporates saturation, quantization noise and a non-linear camera response function.

Acknowledgment

Work partly financed by Agencia Nacional de Investigación e Innovación (ANII, Uruguay) grant FCE_1_2017_135458; Office of Naval research grant N00014-17-1-2552, Programme ECOS Sud – UdelaR - Paris Descartes U17E04, DGA Astrid project “filmer la Terre” n°ANR-17-ASTR-0013-01, MENRT; DGA PhD scholarship jointly supported with FMJH.

Image Credits


Nicola Pierazzo


Köhler et al. [12]


Xu et al. [29]


Jérémy Anger

References

[1] M.S.C. Almeida and M. Figueiredo, *Deconvolving images with unknown boundaries using the alternating direction method of multipliers*, IEEE Transactions on Image Processing, 22 (2013), pp. 3074–3086. https://doi.org/10.1109/TIP.2013.2258354.

[2] J. Anger, G. Facciolo, and M. Delbracio, *Estimating an Image's Blur Kernel Using Natural Image Statistics, and Deblurring it: An Analysis of the Goldstein-Fattal Method*, Image Processing On Line, 8 (2018), pp. 282–304. https://doi.org/10.5201/ipol.2018.211.

[3] ——, *Modeling realistic degradations in non-blind deconvolution*, in Proceedings of IEEE International Conference on Image Processing (ICIP), IEEE, 2018, pp. 978–982. https://doi.org/10.1109/ICIP.2018.8451115.

[4] T.F. Chan and C-K. Wong, *Total variation blind deconvolution*, IEEE Transactions on Image Processing, 7 (1998), pp. 370–375. https://doi.org/10.1109/83.661187.

[5] S. Cho and S. Lee, *Fast motion deblurring*, ACM Transactions on Graphics, 28 (2009), p. 1. https://doi.org/10.1145/1618452.1618491.

[6] R. Fergus, B. Singh, A. Hertzmann, S.T. Roweis, and W.T. Freeman, *Removing camera shake from a single photograph*, ACM Transactions on Graphics, 25 (2006), p. 787. https://doi.org/10.1145/1141911.1141956.

[7] P. Getreuer, *Total Variation Deconvolution using Split Bregman*, Image Processing On Line, 2 (2012), pp. 158–174. https://doi.org/10.5201/ipol.2012.g-tvdc.

[8] A. Goldstein and R. Fattal, *Blur-Kernel Estimation from Spectral Irregularities*, in Proceedings of European Conference on Computer Vision (ECCV), vol. 7576 LNCS, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 622–635. https://doi.org/10.1007/978-3-642-33715-4_45.

[9] A. Gupta, N. Joshi, C.L. Zitnick, M. Cohen, and B. Curless, *Single Image Deblurring Using Motion Density Functions*, in Proceedings of European Conference on Computer Vision (ECCV), 2010, pp. 171–184. https://doi.org/10.1007/978-3-642-15549-9_13.

[10] C. Herley and P.C. Van Oorschot, *SoK: Science, Security, and the Elusive Goal of Security as a Scientific Pursuit*, in Proceedings of IEEE Symposium on Security and Privacy, 2017. https://doi.org/10.1109/SP.2017.38.

[11] M. Hirsch, C.J. Schuler, S. Harmeling, and B. Schölkopf, *Fast removal of non-uniform camera shake*, in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2011, pp. 463–470. https://doi.org/10.1109/ICCV.2011.6126276.

[12] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling, *Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7578 LNCS (2012), pp. 27–40. https://doi.org/10.1007/978-3-642-33786-4_3.

[13] W-S. Lai, Z. Hu, N. Ahuja, and M-H. Yang, *A Comparative Study for Single Image Blind Deblurring*, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. https://doi.org/10.1109/CVPR.2016.188.

[14] A. Levin, Y. Weiss, F. Durand, and W.T. Freeman, *Understanding blind deconvolution algorithms*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 33 (2011), pp. 2354–2367. https://doi.org/10.1109/TPAMI.2011.148.

[15] J. Pan, Z. Hu, Z. Su, and M-H. Yang, *Deblurring Text Images via L0-Regularized Intensity and Gradient Prior*, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 2901–2908. https://doi.org/10.1109/CVPR.2014.371.

[16] J. Pan and Z. Su, *Fast l0-regularized kernel estimation for robust motion deblurring*, IEEE Signal Processing Letters, 20 (2013), pp. 841–844. https://doi.org/10.1109/LSP.2013.2261986.

[17] J. Pan, D. Sun, H. Pfister, and M-H. Yang, *Blind Image Deblurring Using Dark Channel Prior*, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1628–1636. https://doi.org/10.1109/CVPR.2016.180.

[18] ——, *Deblurring Images via Dark Channel Prior*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 40 (2017), pp. 2315–2328. https://doi.org/10.1109/TPAMI.2017.2753804.

[19] D. Perrone and P. Favaro, *A Clearer Picture of Blind Deconvolution*, 2014. https://arxiv.org/pdf/1412.0251.pdf.

[20] ——, *Total Variation Blind Deconvolution: The Devil Is in the Details*, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2014), pp. 2909–2916. https://doi.org/10.1109/CVPR.2014.372.

[21] S.J. Reeves, *Fast image restoration without boundary artifacts*, IEEE Transactions on Image Processing, 14 (2005), pp. 1448–1453. https://doi.org/10.1109/TIP.2005.854474.

[22] I. Rey Otero and M. Delbracio, *Anatomy of the SIFT Method*, Image Processing On Line, 4 (2014), pp. 370–396. https://doi.org/10.5201/ipol.2014.82.

[23] L.I. Rudin and S. Osher, *Total variation based image restoration with free local constraints*, Proceedings of IEEE International Conference on Image Processing (ICIP), 1 (1994), pp. 31—-35 vol.1. https://doi.org/10.1109/icip.1994.413269.

[24] P.L. Søndergaard, *Next fast fft size*, tech. report, 2011. https://ltfat.github.io/notes/ltfatnote017.pdf.

[25] E. Reinhard T. Pouli, D.W. Cunningham, *Image Statistics and their Applications in Computer Graphics*, Eurographics State of the Art Report (STAR), (2010), pp. 83–112. http://www.erikreinhard.com/papers/eg2010_tania.pdf.

[26] Y-W. Tai and S. Lin, *Motion-aware noise filtering for deblurring of noisy and blurry images*, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), (2012), pp. 17–24. https://doi.org/10.1109/CVPR.2012.6247653.

[27] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, *Non-uniform deblurring for shaken images.*, International Journal of Computer Vision, 98 (2012), pp. 168–186. https://doi.org/10.1007/s11263-011-0502-7.

[28] D. Wipf and H. Zhang, *Revisiting Bayesian Blind Deconvolution*, CoRR, abs/1305.2 (2013), pp. 1–39. https://arxiv.org/abs/1305.2362.

[29] L. Xu and J. Jia, *Two-phase kernel estimation for robust motion deblurring*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6311 LNCS (2010), pp. 157–170. https://doi.org/10.1007/978-3-642-15549-9_12.

[30] L. Xu, C. Lu, Y. Xu, and J. Jia, *Image Smoothing via L0 Gradient Minimization*, ACM Transactions on Graphics, 30 (2011). https://doi.org/10.1145/2024156.2024208.

[31] L. Xu, S. Zheng, and J. Jia, *Unnatural L0 sparse representation for natural image deblurring*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 1107–1114. https://doi.org/10.1109/CVPR.2013.147.

[32] T. Yue, S. Cho, J. Wang, and Q. Dai, *Hybrid image deblurring by fusing edge and power spectrum information*, in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8695 LNCS, 2014, pp. 79–93. https://doi.org/10.1007/978-3-319-10584-0_6.

[33] L. Zhong, S. Cho, D. Metaxas, S. Paris, and J. Wang, *Handling noise in single image deblurring using directional filters*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 612–619. https://doi.org/10.1109/CVPR.2013.85.