Published in Image Processing On Line on 2018–09–26.
Submitted on 2017–05–30, accepted on 2018–09–07.
ISSN 2105–1232 © 2018 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.2018.211

Estimating an Image's Blur Kernel Using Natural Image Statistics, and Deblurring it: An Analysis of the Goldstein-Fattal Method

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)

Abstract

Despite the significant improvement in image quality resulting from improvement in optical sensors and general electronics, camera shake blur significantly undermines the quality of hand-held photographs. In this work, we present a detailed description and implementation of the blur kernel estimation algorithm introduced by Goldstein and Fattal in 2012. Unlike most methods that attempt to solve an inverse problem through a variational formulation (e.g. through a Maximum A Posteriori estimation), this method directly estimates the blur kernel by modeling statistical irregularities in the power spectrum of blurred natural images. The adopted mathematical model extends the well-known power-law by contemplating the presence of dominant strong edges in particular directions. The blur kernel is retrieved from an estimation of its power spectrum, by solving a phase retrieval problem using additional constraints associated with the particular nature of camera shake blur kernels (e.g. non-negativity and small spatial support). Although the algorithm is conceptually simple, its numerical implementation presents several challenges. This work contributes to a detailed anatomy of the Goldstein and Fattal method, its algorithmic description, and its parameters.

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: camera-shake; blur kernel estimation; spectral irregularities; Fourier

[1] https://doi.org/10.5201/ipol.2018.211

1 Introduction

The principle of photography lies in accumulating light in the camera sensor (digital or analog) traveling through the camera aperture during a predefined exposure time. Unfortunately, when the camera shakes (e.g. due to the tremor of the photographer hands), the acquired photograph exhibits blur since arriving photons are spread over nearby pixels. Reducing camera shake blur is one of the main bottlenecks for improving final image quality.

This problem is generally addressed by modeling blur as a linear operator acting on the underlying sharp image. If we assume that the acting blur is the same all over the image this boils down to a (shift-invariant) convolution

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

where v is the acquired image, u the latent sharp image, h the blur kernel, and n is the image noise.

Blind image deconvolution aims at recovering both the original image u and the blur kernel h from the blurry and noisy observation v. To address this severely ill-posed problem most techniques use a Maximum a Posteriori (MAP) framework to incorporate additional prior information about both u and h. For instance, the sparsity of natural image derivatives is often enforced during the reconstruction of u, either by means of a regularization term or by explicit edge identification (see [17] for a recent survey). The methods based on the MAP framework usually alternate between the estimation of u and h progressively refining them, or propose a variational Bayesian strategy that marginalizes over the image space [18].

An alternative approach consists in directly recovering the kernel h from the blurry image v without computing u in the process [19, 10, 7]. The main idea is to estimate the kernel h from the anomalies that the spectrum of the blurry image v shows with respect to the canonical behavior of natural sharp images. This canonical behavior can be seen as a statistical version of the sparsity of image derivative priors used in the MAP framework. Image derivatives of a non-blurry image are typically weakly correlated, so its autocorrelation should be close to a delta function. Yitzhaky et al. [19] directly estimate the power spectrum (PS) of the kernel in the sensor movement direction from the 1D autocorrelation of the directional image derivative. Hu et al. [10] propose to use an eight-point Laplacian filter for whitening the image spectrum so that the covariance matrix of the image patches provides an estimate of the 2D kernel PS. This assumption will be further described and extended to an improved image model in Section 2 but we give now an intuition behind this family of kernel estimation methods. Given a whitening filter d such that $|\hat{d}(\boldsymbol{\xi})|^2 = \|\boldsymbol{\xi}\|^2$, where $\hat{\cdot}$ represents the Fourier transform, and assuming the image follows the isotropic model $|\hat{u}(\boldsymbol{\xi})|^2 = c \cdot \|\boldsymbol{\xi}\|^{-2}$, we have

$$\begin{aligned} |\widehat{d * v}(\boldsymbol{\xi})|^2 &= |\hat{d}(\boldsymbol{\xi})|^2 \cdot |\hat{v}(\boldsymbol{\xi})|^2 \\ &= \|\boldsymbol{\xi}\|^2 \cdot c \cdot \|\boldsymbol{\xi}\|^{-2} \cdot |\hat{h}(\boldsymbol{\xi})|^2 \\ &= c \cdot |\hat{h}(\boldsymbol{\xi})|^2, \qquad \forall \boldsymbol{\xi} \neq (0,0). \end{aligned} \tag{2}$$

The blur kernel can then be computed from the estimated kernel power spectrum using a phase retrieval algorithm that estimates the Fourier phase by enforcing additional restrictions on the kernel, such as non-negativity and compact support.

Goldstein and Fattal [7] notice that in a typical natural image the presence of many strong edges in particular directions breaks the commonly assumed isotropic power-law decay [4]. The authors propose a refined power-law that accounts for this observation and makes the model (2) more accurate. This refinement leads in many scenarios to an improvement in the estimation of the blur kernel power spectrum. The authors also introduce an adaptation of the Relaxed Averaged

Alternating Reflections (RAAR) phase retrieval algorithm [13], which improves its convergence speed and its ability to resolve ambiguities.

Unlike methods that rely on the presence and identification of well-contrasted edges in images, due to its statistical formulation, this approach copes well with images containing under-resolved texture or foliage clutter, which are abundant in outdoor scenes.

Although the algorithm is conceptually simple, being based on clean mathematical and physical assumptions, its numerical implementation presents several challenges and many important details need to be fully specified. This work contributes to a detailed anatomy of the Goldstein and Fattal blur kernel estimation algorithm, and includes an exhaustive evaluation with synthetic examples. We also propose an implementation that aims to be faithful to the original one. However, we simplified some steps, and these modifications are highlighted throughout the paper.

The rest of the paper is organized as follows. The refined power-law model along with the estimation of the power spectrum of the kernel as proposed by Goldstein and Fattal in [7] is presented in Section 2. Then, the algorithmic description of the method is detailed in Section 3. Section 4 presents some experiments and we finally conclude in Section 5.

2 From Natural Image Model to Blur Kernel Estimation

In this section we show how a simple model derived from statistics of natural images can lead to an estimation of the blur kernel. First, we introduce mathematical tools used throughout the section, including the autocorrelation, the shear projection and their relation to Fourier analysis. We then review a simple, yet powerful, natural image model based on an anisotropic power-law for the power spectrum. Furthermore, we show how whitening the image reveals the blur kernel spectrum up to some multiplicative coefficients. Using three physically-based assumptions on blur kernels, the autocorrelation of the shear projection of the kernel can be estimated. Finally, we show how to reconstruct the power spectrum of the kernel from these autocorrelation signals.

2.1 Notation and Mathematical Tools from Fourier Analysis

Let $\mathbb{R}^2$ be the set of pairs of real numbers $\boldsymbol{x} = (x, y)$. Let us denote by $L^1(\mathbb{R}^2)$ the set of integrable functions on $\mathbb{R}^2$, $L^2(\mathbb{R}^2)$ the set of square integrable functions on $\mathbb{R}^2$. We denote by $\mathcal{BL}(\mathbb{R}^2)$ the set of $L^2(\mathbb{R}^2)$ functions that are band-limited inside $[-\pi, \pi]^2$. We will denote by $\mathcal{BL}^1(\mathbb{R}^2) = L^1(\mathbb{R}^2) \cap \mathcal{BL}(\mathbb{R}^2)$.

In order to present the theory behind the Goldstein-Fattal method, we will assume that all continuous images are integrable band-limited functions, that is, they belong to $\mathcal{BL}^1(\mathbb{R}^2)$. The optical lens present in any camera motivates the band-limited assumption, while the integrable assumption is reasonable since we can assume that the energy outside the observable window is negligible. In this particular setting we have the following results, that are standard results from Fourier Analysis (see, for example, [2]).

Definition 1. *Let u be an image defined in $L^1(\mathbb{R}^2)$, the Fourier Transform $\mathcal{F}(u)$ of u is defined as,*

$$\hat{u}(\boldsymbol{\xi}) = \iint_{\mathbb{R}^2} u(\boldsymbol{x}) e^{-i\boldsymbol{x}\cdot\boldsymbol{\xi}} d\boldsymbol{x}, \tag{3}$$

for $\boldsymbol{\xi} = (\xi_1, \xi_2) \in \mathbb{R}^2$.

The Inverse Fourier Transform $\mathcal{F}^{-1}(\hat{u})$ of $\hat{u} \in L^1(\mathbb{R}^2)$ is defined as

$$u(\boldsymbol{x}) = \frac{1}{(2\pi)^2} \iint_{\mathbb{R}^2} \hat{u}(\boldsymbol{\xi}) e^{i\boldsymbol{x}\cdot\boldsymbol{\xi}} d\boldsymbol{\xi}, \tag{4}$$

for $\boldsymbol{x} = (x, y) \in \mathbb{R}^2$, *and satisfies* $\mathcal{F}^{-1}(\mathcal{F}(u)) = u$ *if* u *and* $F(u)$ *are both in* $L^1(\mathbb{R}^2)$.

As mentioned in the introduction (Equation (1)), blur can be modeled as a convolution between a sharp (the underlying generally unknown) image $u \in \mathcal{BL}^1(\mathbb{R}^2)$ and a blur kernel $h \in \mathcal{BL}^1(\mathbb{R}^2)$. Let us introduce a basic result regarding the convolution between two images.

Definition 2 (Convolution). *The convolution between two 2D images* $u, h \in L^1(\mathbb{R}^2)$ *is defined as*

$$(u * h)(\boldsymbol{x}) = \iint_{\mathbb{R}^2} u(\boldsymbol{x} - \boldsymbol{s})h(\boldsymbol{s})d\boldsymbol{s}. \tag{5}$$

Proposition 1 (Convolution in Fourier). *Let* u *and* h *be two images defined in* $L^1(\mathbb{R}^2)$, *then we have*

$$\mathcal{F}(u * h) = \mathcal{F}(u) \cdot \mathcal{F}(h). \tag{6}$$

Definition 3 (Autocorrelation 1D). *We define the 1D autocorrelation of* $w \in L^1(\mathbb{R})$ *as*

$$R(w)(\tau) = \int_{\mathbb{R}} w(x)w(x + \tau)dx. \tag{7}$$

Proposition 2. *Let* $w \in L^1(\mathbb{R})$ *be a 1D signal. The relation between the Fourier transform of the autocorrelation of* w *and the Fourier transform of* w *is given by*

$$\mathcal{F}(R(w))(\xi) = |\mathcal{F}(w)(\xi)|^2, \quad \forall \xi \in \mathbb{R}. \tag{8}$$

Let us introduce the shear projection operation whose utility will become evident later.

Definition 4 (Shear projection). *Let* $u \in L^1(\mathbb{R}^2)$ *be an image, the vertical shear projection of* u *with parameter* $\alpha_\phi = -\tan\phi$ *is defined as*

$$P_\phi(u)(x) = \int_{\mathbb{R}} u(x + \alpha_\phi y, y)dy. \tag{9}$$

The horizontal shear projection is defined in a similar manner with $\alpha_\phi = -1/\tan\phi$, *and the similar formula holds,*

$$P^h_\phi(u)(y) = \int_{\mathbb{R}} u(x, y + \alpha_\phi x)dx. \tag{10}$$

An illustration of the horizontal shear projection operator is shown in Figure 1.

Proposition 3. *Let* $u \in L^1(\mathbb{R}^2)$ *be an image and* $\phi \in [-\frac{\pi}{2}, \frac{\pi}{2})$ *an angle. Then the Fourier coefficients of the shear projection of* u *at angle* ϕ *are related to* u *by the following equality:*

$$\widehat{P_\phi(u)}(\xi) = \hat{u}(\xi, -\alpha_\phi\xi). \tag{11}$$

Proof.

$$\begin{aligned}
\widehat{P_\phi(u)}(\xi) &= \int_{\mathbb{R}} P_\phi(u)(x)\, e^{-ix\cdot\xi}dx & (12)\\
&= \int_{\mathbb{R}}\int_{\mathbb{R}} u(x + \alpha_\phi y, y)dy\, e^{-ix\cdot\xi}dx & (13)\\
&= \int_{\mathbb{R}}\int_{\mathbb{R}} u(x', y)e^{-i(x'\xi - y\alpha_\phi\xi)}dydx' & (14)\\
&= \hat{u}(\xi, -\alpha_\phi\xi). & (15)
\end{aligned}$$

□

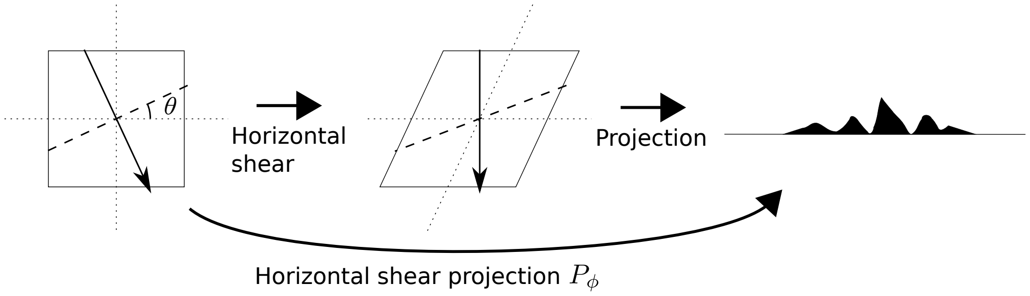


Figure 1: The shear projection operator P_ϕ is the composition of the shear and the vertical projection.

2.2 Blur Kernel Assumptions

We now introduce three assumptions about the kernel that are derived from its physical properties: non-negativity, (approximately) compact support, and unit area,

(i) $h(\boldsymbol{x}) \geq 0,\ \forall \boldsymbol{x} \in \mathbb{R}^2$ (non-negativity),

(ii) $\iint_{\mathbb{R}^2 \setminus [-K,K]^2} h(\boldsymbol{x})\ d\boldsymbol{x} < \delta$, δ close to 0 (approx. compact support),

(iii) $\iint h(\boldsymbol{x})\ d\boldsymbol{x} = 1$ (unit area).

One key observation is that the unit area assumption can be expressed in terms of shear projections. Let $\phi \in [0, \pi)$, then

$$\int_{\mathbb{R}} P_\phi(h)(x)dx = \int_{\mathbb{R}} \int_{\mathbb{R}} h(x + \alpha_\phi y, y)dydx \tag{16}$$

$$= \int_{\mathbb{R}} \int_{\mathbb{R}} h(x', y)dx'dy \tag{17}$$

$$= 1. \tag{18}$$

A similar observation can be made for the autocorrelation of the projection, namely

$$\int_{\mathbb{R}} R(P_\phi(h))(x)dx = 1. \tag{19}$$

The importance of these assumptions will become clear later.

2.3 The Anisotropic Power Spectrum Image Model

Modeling natural images, which are assumed to be sharp, is essential to distinguish between sharp and blurry images. Furthermore, by studying the discrepancy of a blurry image with respect to a given model, one can extract information of the blur kernel. In this section, we review a simple model based on the power spectrum of natural images.

A powerful yet over-simplistic assumption when modeling natural images is to assume that for frequencies larger than a certain threshold ϵ its power spectrum follows an isotropic power-decay law

$$|\hat{u}(\boldsymbol{\xi})|^2 \approx c \cdot \|\boldsymbol{\xi}\|^{-\beta}, \quad \forall \boldsymbol{\xi} \in \mathbb{R}^2 \text{ s.t. } \|\boldsymbol{\xi}\| > \epsilon, \tag{20}$$

with $c > 0$ and ϵ a very small positive scalar. In what follows we use a similar power-decay model

$$|\hat{u}(\boldsymbol{\xi})|^2 \approx (\epsilon + c^{-1} \cdot \|\boldsymbol{\xi}\|^{\beta})^{-1}, \quad \forall \boldsymbol{\xi} \in \mathbb{R}^2, \tag{21}$$

that has the advantage of being valid over the entire domain while keeping the same behavior on high frequencies.

The usual choice of $\beta = 2$ (a quadratic decay of the power spectrum) is what we would expect if the relative contrast energy of the image were scale invariant (i.e. independent of the viewing distance). This hypothesis holds for many types of images, for instance those following the dead leaves model, or containing ideal step edges [14].

Although this model has proven accurate for certain image types, in the presence of long edges or elongated structures at particular orientations the rotation invariance assumption does not hold. As a matter of fact, Torralba and Oliva [16] study a generalization that contemplates a particular anisotropic behavior. Given $\boldsymbol{\xi} \neq (0, 0)$, we denote by $\boldsymbol{n}_\theta = \frac{\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|}$ the unit length vector $\boldsymbol{n}_\theta$ pointing to the orientation θ given by $\boldsymbol{\xi}$. Let ξ be such as $\xi\boldsymbol{n}_\theta = \boldsymbol{\xi}$. Then we have the anisotropic model,

$$|\hat{u}(\xi\boldsymbol{n}_\theta)|^2 \approx (\epsilon + c_\theta^{-1} \cdot |\xi|^{\beta_\theta})^{-1}, \tag{22}$$

where $c_\theta > 0$ is a constant amplitude factor for each orientation θ and β_θ is a constant frequency exponent for each orientation.

In many scenarios, where there are strong edges at particular orientations, the exponent can be assumed to be $\beta_\theta \approx 2$, and thus the power spectral model simplifies to

$$|\hat{u}(\xi\boldsymbol{n}_\theta)|^2 \approx (\epsilon + c_\theta^{-1} \cdot |\xi|^2)^{-1}. \tag{23}$$

Note that the factors c_θ have no influence on the DC coefficient ($\xi = 0$), which is coherent since this value is shared by all slices.

2.4 Autocorrelation of Kernel Projections from Spectral Irregularities

The anisotropic model introduced above indicates a power-law decay of the power spectrum of natural images. The operation of whitening an image consists in filtering it so that its power spectrum is white (i.e. flat). Since we want to reveal the blur kernel, we cannot whiten a blurry image with respect to its own power spectrum. However, exploiting the natural image model (23), we can filter the image so that its spectrum should be white if the image were sharp. The resulting irregularities thus corresponds to the blur kernel.

In this section, we show how to filter an image according to the natural image model and prove that the result corresponds to the power spectrum of the kernel up to constant factors per angle.

Let $u \in \mathcal{BL}^1(\mathbb{R}^2)$ and we denote by $D_\phi u(\boldsymbol{x})$ the continuous directional derivative of $u(\boldsymbol{x})$ along the direction given by $\boldsymbol{n}_\phi = (\cos\phi, \sin\phi)$. This can be rewritten as

$$\begin{aligned} D_\phi u(\boldsymbol{x}) &= \frac{\partial u}{\partial x}(x,y)\cos\phi + \frac{\partial u}{\partial y}(x,y)\sin\phi && (24)\\ &= \iint_{[-\pi,\pi]^2} \hat{u}(\xi_1,\xi_2)\left(\frac{\partial}{\partial x}e^{i(x\xi_1+y\xi_2)}\cos\phi + \frac{\partial}{\partial y}e^{i(x\xi_1+y\xi_2)}\sin\phi\right)d\xi_1 d\xi_2 && (25)\\ & && (26)\\ &= \iint_{[-\pi,\pi]^2} \hat{u}(\boldsymbol{\xi})i(\boldsymbol{\xi}\cdot\boldsymbol{n}_\phi)e^{i(\boldsymbol{x}\cdot\boldsymbol{\xi})}d\boldsymbol{\xi}. && (27) \end{aligned}$$

Thus, we define the continuous directional derivative operator as the filter $d_\phi \in \mathcal{BL}^1(\mathbb{R}^2)$ having Fourier coefficients

$$(\hat{d_\phi})(\boldsymbol{\xi}) = i\boldsymbol{\xi}\cdot\boldsymbol{n}_\phi, \quad \forall\boldsymbol{\xi} \in [-\pi,\pi]^2. \tag{28}$$

If we restrict to the frequencies $\boldsymbol{\xi} = \xi \boldsymbol{n}_\theta$ along the slice $\theta = \phi$, given by the orientation of the directional derivative, we obtain the following power spectrum for the derivative

$$\left|\hat{d}_\phi(\xi \boldsymbol{n}_\phi)\right|^2 = |\xi|^2. \tag{29}$$

Remark 1 (Whitening)**.** *Let u and h be an image and a blur kernel both in $\mathcal{BL}^1(\mathbb{R}^2)$ and let $v = u * h$ represent the observed blurry image. Then, given a direction θ the coefficients of the directional derivative over the slice θ satisfy*

$$\begin{aligned}
\left|(\widehat{D_\theta v})(\xi \boldsymbol{n}_\theta)\right|^2 &= |\hat{d}_\theta(\xi \boldsymbol{n}_\theta)|^2 \cdot |\hat{v}(\xi \boldsymbol{n}_\theta)|^2 && (30)\\
&= |\xi|^2 \cdot |\hat{u}(\xi \boldsymbol{n}_\theta)|^2 \cdot |\hat{h}(\xi \boldsymbol{n}_\theta)|^2 && (31)\\
&= \frac{|\xi|^2}{(\epsilon + c_\theta^{-1} \cdot |\xi|^2)} \cdot |\hat{h}(\xi \boldsymbol{n}_\theta)|^2 && (32)\\
&= c_\theta \cdot |\hat{h}(\xi \boldsymbol{n}_\theta)|^2 \cdot \left(1 - \frac{\epsilon c_\theta}{\epsilon c_\theta + |\xi|^2}\right) && (33)\\
&= c_\theta \cdot |\hat{h}(\xi \boldsymbol{n}_\theta)|^2 - c_\theta \cdot |\hat{h}(\xi \boldsymbol{n}_\theta)|^2 \frac{\epsilon c_\theta}{\epsilon c_\theta + |\xi|^2} && (34)\\
&= c_\theta \cdot |\hat{h}(\xi \boldsymbol{n}_\theta)|^2 + \widehat{r_\theta^\epsilon}(|\xi|), && (35)
\end{aligned}$$

where we have defined

$$\widehat{r_\theta^\epsilon}(|\xi|) = -c_\theta \cdot |\hat{h}(\xi \boldsymbol{n}_\theta)|^2 \frac{\epsilon c_\theta}{\epsilon c_\theta + |\xi|^2}. \tag{36}$$

Equation (35) can be written using projections according to Equations (8) and (11)

$$\mathcal{F}(R(P_\theta(D_\theta v)))(\xi) = c_\theta \cdot \mathcal{F}(R(P_\theta(h)))(\xi) + \widehat{r_\theta^\epsilon}\left(\frac{|\xi|}{\cos\theta}\right), \tag{37}$$

or equivalently in the spatial domain

$$R(P_\theta(D_\theta v))(x) = c_\theta \cdot R(P_\theta(h))(x) + |\cos\theta| \cdot r_\theta^\epsilon(x). \tag{38}$$

In Appendix A we show that the term $r_\theta^\epsilon(x)$ has a very small derivative. In what follows we will assume that it is almost constant within the support of $R(P_\theta(h))$,

$$R(P_\theta(D_\theta v))(x) = c_\theta \cdot R(P_\theta(h))(x) + \mu_\theta. \tag{39}$$

2.5 Kernel Power Spectrum Reconstruction

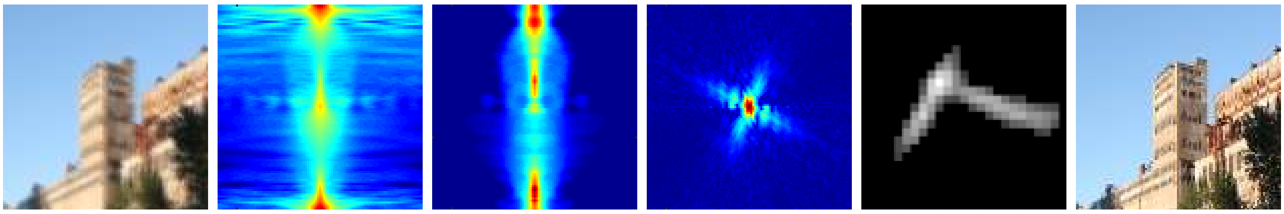
In Section 2.2, we formulated three realistic assumptions on the kernel. Assumptions (i) and (ii), imply that the kernel should be always non-negative and zero outsize a compact set. Thus, we can substract the minimum value of $R(P_\theta(D_\theta v))(x)$, to recover $c_\theta \cdot R(P_\theta(h))(x)$. Then, we need to get rid of the nuisance parameter c_θ.

The value of c_θ can be determined using assumption (iii) in Section 2.2. Indeed, knowing that the projection of the autocorrelation of the kernel has area one, $c_\theta \cdot R(P_\theta(h))(x)$ can be normalized so as its area is one, and thus compensate for c_θ. In short, from assumptions (i), (ii), and (iii) in Section 2.2 it is possible to recover $R(P_\theta(h))(x)$ from $R(P_\theta(D_\theta v))(x)$.

Given the autocorrelations $R(P_\theta(h))$ for a given θ, the power spectrum of the kernel can be reconstructed (up to the discretization of the θ angle grid). Indeed, from Equation (11) we have

$$\widehat{R(P_\theta(h))}(\xi\cos\theta) = |\widehat{P_\theta(h)}(\xi\cos\theta)|^2 = |\hat{h}(\xi \boldsymbol{n}_\theta)|^2. \tag{40}$$

Thus, $|\hat{h}(\xi \boldsymbol{n}_\theta)|^2$ can be retrieved from $\widehat{R(P_\theta(h))}$.



(a) v (b) $R(P_\theta(D_\theta v))$ (c) $R(P_\theta(h))$ (d) $|H|$ (e) h (f) Deblurred

Figure 2: Overview of the kernel estimation process. (a) Input blurry image, (b) the 1D auto-correlation functions of the differentiated projections of the blurry image. Each row corresponds to a projection along a different θ. (c) Recovered correlations using the estimated c_θ and μ_θ. The power spectrum recovered is shown in (d). The kernel computed by the phase retrieval algorithm is shown in (e) and finally the deblurred image in (f).

Influence of noise. A development equivalent to (35) for the noisy case shows that the whitening boosts the noise in high frequencies

$$\mathcal{F}(R(P_\theta(D_\theta(v+n))))(\xi\cos\theta) = c_\theta \left|\hat{h}(\xi \boldsymbol{n}_\theta)\right|^2 + \hat{r}^{\epsilon}_{\theta}(\xi\cos\theta) + |\xi|^2 |\hat{n}(\xi\boldsymbol{n}_\theta)|^2. \tag{41}$$

Thus the projection does not result in a reduction of the noise compared to other methods based on a 2D whitening [10].

The next section describes a procedure to retrieve the kernel from a blurry image based on this model, including the retrieval of the phase of the kernel.

3 Numerical Algorithm for Blur Kernel Estimation

A high level description of the Goldstein and Fattal method is presented in Algorithm 1, and the intermediate results for a given example are depicted on Figure 2. The first step computes the 1D autocorrelation of the projections of the whitened image for a set of angles. This is detailed in Section 3.1. An initial support of the projections is then estimated, which allows to begin an iterative process. This process estimates the power spectrum of the kernel, recovers a kernel using a phase retrieval algorithm, and reestimates the kernel support. These steps are detailed in Section 3.2.

Discrete considerations. In this section, continuous images are interpolated versions of their discrete counterpart such that $u_{k,l} = u(k,l)$, $\forall (k,l) \in \Omega$ where $\Omega = \{-M,\dots,M\} \times \{-N,\dots,N\}$. The continuous images are assumed to be band-limited and sampled without alias. Since compactness cannot be enforced on both the spatial support and the frequency support, we assume that the energy of the interpolated image outside the support Ω is negligible. As such, the theory described in Section 2 applies.

However, the discrete version of the method aims to recover a discrete power spectrum of the kernel, not a continuous one. This implies an implicit periodization in the spatial domain and motivates the use of the Discrete Fourier Transform instead of the Fourier Transform. Since the spectrum is defined on a finite discrete grid, the power spectrum is estimated for a finite number of coefficients and so the number of orientations θ is finite as well. We note the discrete Fourier transform of h as $H_{m,n} = \mathcal{F}(h)_{m,n}$.

In what follows, we will assume, without loss of generality, that the kernel support is a square of size $M_h \times M_h$. Notwithstanding, for the numerical computations of the power spectrum of the kernel, we will assume a larger support, that is $rM_h \times rM_h$, with $r = 4$ in our implementation.

Algorithm 1: Kernel estimation from blurry image (entry point).

```
input  : blurred image $v$, approximate kernel size $p \times p$, compensation factor $\alpha$
output: blur kernel $h$
Sec. 3.1 {
    $\mathcal{A}$ = ComputeProjectionAngleSet($p$)                                   Equation (42)
    $\{R(P_\theta(D_\theta v))\}$ = ComputeProjectionsAutocorrelation($v, \mathcal{A}, p, \alpha$)   Algorithm 2
Sec. 3.2 {
    $\{s_\theta\}$ = InitialSupportEstimation($\{R(P_\theta(D_\theta v))\}, \mathcal{A}$)           Algorithm 3
    for $i$ from 1 to $N_{outer}$ do
        $|H|^2$ = EstimatePowerSpectrum($\{R(P_\theta(D_\theta v))\}, \{s_\theta\}$)          Algorithm 4
        $h$ = RetrievePhase($|H|^2, p$)                                             Algorithm 6
        $\{s_\theta\}$ = ReestimateSupport($h, \mathcal{A}$)
```

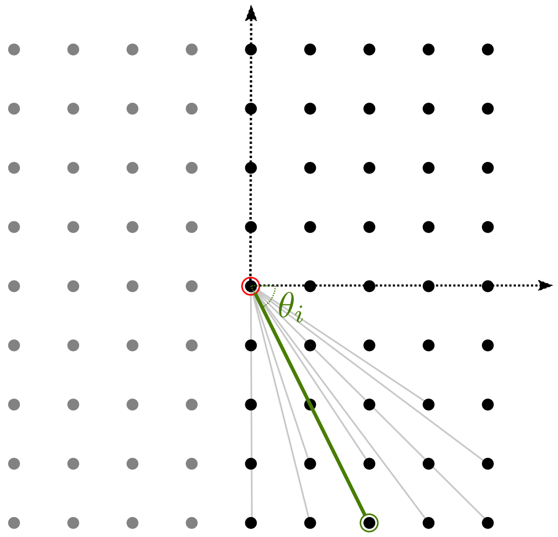


Figure 3: For a given support, projection angles are selected such that at least one slice passes exactly through each sample of the power spectrum.

3.1 Computing the 1D Autocorrelations

As shown by Equation (39), given an orientation θ, the autocorrelation of the projection of the filtered blurry image along θ gives an estimate of the autocorrelation of the projection of the kernel. This estimate can then be used to reconstruct the power spectrum of the kernel.

The first step of the method is to compute the angle set $\mathcal{A}$ along which the image will be projected. Then, the image is whitened with a discrete differential filter derived from the continuous derivative filter D_θ. Using the angle set $\mathcal{A}$, the directional derivative of the image is projected for each orientation and the $1D$ autocorrelations of these projections are computed. Additionally, a compensation filter is applied on the autocorrelations to account for the intrinsic blur of the camera. This will be explained further in this section.

Each of these steps are detailed in what follows.

Projections angle set. Let us define the set of projection angles $\mathcal{A}$ such that at least one slice passes exactly through each pixel of the kernel frequency domain (see Figure 3), that is,

$$\mathcal{A} = \Big\{\theta \in (-{}^{\pi}/_{2}, {}^{\pi}/_{2}] : \tan\theta = {}^{j}/_{i}, (i,j) \in \{-\frac{r}{2}M_h, \dots, \frac{r}{2}M_h\} \times \{-\frac{r}{2}M_h, \dots, \frac{r}{2}M_h\}\Big\}. \tag{42}$$

In our implementation and in Figure 2 the angle set is sorted decreasingly from $\pi/2$ to $-\pi/2$.

The projection angles θ are measured with respect to the horizontal axis and the rotation center is the image center.

Whitening and projections. The first step towards the reconstruction of the power spectrum of the kernel is to whiten the blurry image using the filters D_θ and project it with respect to multiple angles. Noting that the operators D_θ can be constructed from an horizontal and a vertical filter (see Equation (24)), we can implement the filtering using the relation

$$D_\theta v = v^x \cos\theta + v^y \sin\theta, \tag{43}$$

where v^x and v^y denote the horizontal and vertical discrete derivatives of the blurry image computed using, as in Goldstein and Fattal [7], the 1D filter $d = [3, -32, 168, -672, 0, 672, -168, 32, -3]/840$. The projections of the blurry image are performed using a shear projection operator using nearest neighbor interpolation for efficiency, as detailed in Algorithm 2. For $\theta \in [-\pi/4, \pi/4]$, the shear projection is performed horizontally; the vertical shear is used for the other orientations. Computing the projections is the only step of the method that works on the entire input image; the following steps work on a reduced domain, which is proportional to the kernel size.

Autocorrelation of projections. Once the image projections are obtained, the discrete 1D autocorrelation limited to a window of size rM_h is computed by convolving each projection with its mirrored version. A zero boundary condition is used for this convolution.

Compensation filter. In the implementation of Goldstein et al. [7] a post-processing compensation filter is applied on the autocorrelations in order to account for the fact that the whitening procedure might not produce a perfect delta even on sharp images. This filter can be seen as a sharpening filter which reduces the effect of the camera intrinsic blur and results in a sharper kernel. This filtering is performed by deconvolving each 1D autocorrelation by the following symmetric discrete kernel

$$h_k = \frac{1}{Z}(|k|+1)^{-\alpha}, \quad k \in [-\frac{r}{2}M_h, \frac{r}{2}M_h], \tag{44}$$

where $\alpha \in [1, \infty)$ is a parameter and $Z = \sum_k (|k|+1)^{-\alpha}$ is a normalization factor. The deconvolution is performed using the conjugate gradient method [9] with replicate boundary condition. If one value of the resulting autocorrelation at a maximum distance of 2 pixels of the center is negative, the deconvolution is considered erroneous and the original autocorrelation is kept.

3.2 Kernel Estimation from Projections

Before reconstructing the power spectrum of the kernel, one needs to adjust the computed autocorrelation to account for the c_θ and μ_θ parameters of the anisotropic model. Estimating c_θ is described in the following sections and requires an estimation of the support of the autocorrelation s_θ. As the image might not respect exactly the anisotropic power law model, an iterative scheme is used to successively refine the support (see Algorithm 1). During the iteration, the power spectrum $|\hat{h}|^2$ is reconstructed from $R(P_\theta(D_\theta v))$ and its associated phase is estimated. In turn, this kernel allows to reestimate the kernel support s_θ. In the following paragraphs, we describe each step in detail.

Initial support estimation. The initial support estimation consists in finding the minima in the autocorrelation of projections $R(P_\theta(D_\theta v))$. For each angle, we set s_θ to the positive index of the minimum of the autocorrelation. A Lipschitz continuity constraint is then imposed on the resulting

```
Algorithm 2: ComputeProjectionsAutocorrelation
  input : blurred image $v$ defined on $\Omega$, angle sampling set $\mathcal{A}$, approximate kernel size $p \times p$,
          compensation factor $\alpha$
  output: autocorrelation of projections $R(P_\theta(D_\theta v))$ for $\theta \in \mathcal{A}$
  // Whitening
  $d = [3, -32, 168, -672, 0, 672, -168, 32, -3]/840$
  $v^x = d * v$                                         Horizontal discrete derivative
  $v^y = d^T * v$                                       Vertical discrete derivative
  // Projection
  foreach $\theta \in \mathcal{A}$ do
      $q = 0$
      foreach $l \in [-N_v, N_v]$ do
          foreach $k \in [-M_v, M_v]$ do
              if $\theta \in [-\pi/4, \pi/4]$ then
                  offset $= x + y \tan\theta$                    Horizontal shear
              else
                  offset $= y + \frac{x}{\tan\theta}$              Vertical shear
              $w = (v^x)_{k,l} \cos\theta + (v^y)_{k,l} \sin\theta$          Equation (43)
              $q_{\mathtt{round}(\text{offset})} = q_{\mathtt{round}(\text{offset})} + w$    Nearest neighbor accumulation
      // Autocorrelation and compensation filter
      $R(P_\theta(D_\theta v)) = \mathtt{autocorrelation}(q, p)$
      foreach $k \in [-M_v, M_v]$ do
          $h_k = \frac{(|k|+1)^{-\alpha}}{\sum_{k'} (|k'|+1)^{-\alpha}}$
      solve for $R(P_\theta(D_\theta v))'$: $h * R(P_\theta(D_\theta v))' = R(P_\theta(D_\theta v))$   Conjugate gradient descent
```

support to ensure that it does not vary too much on a local neighborhood. This constraint is imposed using the following formula

$$\forall \theta_i, \theta_j,\ \ s_{\theta_i} = \min(s_{\theta_i}, s_{\theta_j} + \kappa|i - j|), \tag{45}$$

with $\kappa = 2/70$, which results in Algorithm 3. An illustration of this constraint is shown in Figure 4.

```
Algorithm 3: InitialSupportEstimation
  input : autocorrelation of projections $R(P_\theta(D_\theta v))$ of size $rM_h$, angle set $\mathcal{A}$, maximum slope
          $\kappa$ $(= 2/70)$
  output: support $s_\theta$
  foreach $\theta \in \mathcal{A}$ do
      $s'_\theta = \arg\min_{k>0}\ R(P_\theta(D_\theta v))_k$          Index of the minimum value
      $s_\theta = rM_h$
  foreach $\theta_i \in \mathcal{A}$ do
      if $s'_{\theta_i} < s_{\theta_i}$ then
          $s_{\theta_i} = s'_{\theta_i}$
          foreach $\theta_j \in \mathcal{A}$ do
              $s_{\theta_j} = \min(s_{\theta_j}, s'_{\theta_i} + \kappa|i - j|)$     Propagate the constraint (45) to neighbours
```

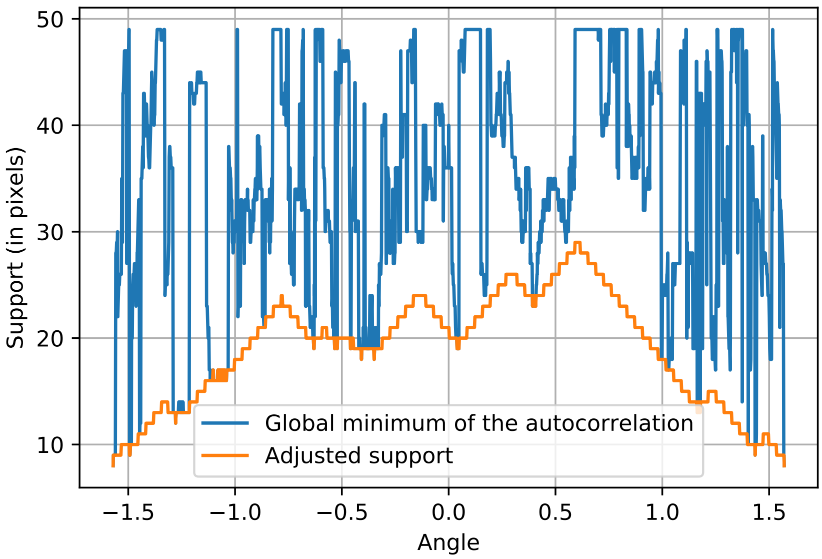


Figure 4: Effect of the Lipschitz continuity constraint for the initial support estimation. The blue plot indicates the index of the minimum value of the autocorrelation for every angle. The orange curve corresponds to the adjusted support, which is used as initial support for the method.

Power spectrum reconstruction. Once the autocorrelations of the projections and the estimation of their support are computed, the power spectrum of the kernel can be estimated.

The first step is to estimate the autocorrelation of the projection of the kernel. This uses the three blur kernel assumptions in order to compensate for c_θ and μ_θ as described in Section 2.5. However, instead of defining μ_θ as the minimum value of the autocorrelation, it is defined as the value $R(P_\theta(D_\theta v))(s_\theta)$, to account for noise and model inaccuracies. The discrete 2D power spectrum can then be reconstructed from samples of the discrete Fourier transform of the autocorrelations. Algorithm 4 describes how to extract the power spectrum from the autocorrelation of projections $R(P_\theta(D_\theta v))$ and the estimated support s_θ.

Figures 2b and 2c indicate the resulting $R(P_\theta(h))$ estimated from $R(P_\theta(D_\theta v))$ for each angle θ of the angle set. In order to reduce the noise during the estimation of $R(P_\theta(h))$, a 1D median filter of size $2\sqrt{|\mathcal{A}|}$ is applied on the autocorrelation across the angles. Figure 2c was processed with the median filtering, applied independently on each column.

Phase retrieval. Reconstructing the kernel from its spectral representation requires both its power spectrum $|H|^2$ and its discrete Fourier transform phase $\phi(H)$. Phase retrieval algorithms [5, 3, 13] allow to reconstruct the Fourier phase from the power spectrum and a few assumptions about the signal. In the particular context of blur kernels, the reconstructed signal should be non-negative, of limited support, and should sum to one.

The error-reduction algorithm [5] and its further improvements [3, 1] consists in iteratively enforcing the given power spectrum in the Fourier domain and then applying spatial domain constraints to retrieve a non-negative support-limited signal. The initial phase is set to random values.

In their work, Goldstein et al. [7] modify the Relaxed Averaged Alternating Reflections algorithm originally proposed by Luke [13] to be less sensitive to inaccurate input. This modification consists in considering the power spectrum as an attraction point instead of a hard constraint. Algorithm 5 describes this method.

Since the phase retrieval algorithm is not guaranteed to converge to the correct solution, the method estimates $N_{\text{tries}} = 30$ phases $\phi^{(i)}(H)$ as described in Algorithm 6. Since the phase retrieval

```
Algorithm 4: EstimatePowerSpectrum
  input  : autocorrelation of projections $R(P_\theta(D_\theta v))$, estimated support $s_\theta$
  output: kernel power spectrum $|H|^2$
  // Compensate $\mu_\theta$ and $c_\theta$
  foreach $\theta \in A$ do
      $\mu = R(P_\theta(D_\theta v))(s_\theta)$
      $R(P_\theta(h))_k = \begin{cases} \max(0, R(P_\theta(D_\theta v))_k - \mu) & \text{if } k \in [-s_\theta, s_\theta] \\ 0 & \text{otherwise} \end{cases}$
      $R(P_\theta(h))_k = \frac{R(P_\theta(h))_k}{\sum_k R(P_\theta(h))_k}$
  // Filter the autocorrelations and reconstruct the power spectrum
  median filter on $R(P_\theta(h))$ across angles
  reconstruct $|H|^2$ from the samples of $\mathcal{F}(R(P_\theta(h)))$
```

```
Algorithm 5: SinglePhaseRetrieval
  input  : discrete kernel spectrum magnitude $|H|$ of size $rM_h \times rM_h$, kernel size $s$
  output: kernel $h$ of size $s \times s$
  $\alpha = 0.95$
  $\beta_0 = 0.75$
  initialize a Hermitian symmetric random phase $\phi(G)_{m,n}$
  $g = \mathcal{F}^{-1}(|H| \cdot e^{i \cdot \phi(\hat{g})})$                    Associate the magnitude and the random phase
  for $m$ from $0$ to $N_{inner} - 1$ do
      $\beta = \beta_0 + (1 - \beta_0)(1 - \exp(-(m/7)^3))$
      $g' = \mathcal{F}^{-1}((\alpha \cdot |H| + (1 - \alpha) \cdot |G|) \cdot e^{i \cdot \phi(G)})$    Enforce the spectrum magnitude constraint
      $\Omega = \{(k,l) : 2 \cdot g'_{k,l} < g_{k,l}\} \cup \{(k,l) : (k,l) \notin [0,s]^2\}$     Invalid values of the kernel
      $g_{k,l} = \begin{cases} \beta \cdot g_{k,l} + (1 - 2\beta) \cdot g'_{k,l} & \text{if } (k,l) \in \Omega \\ g'_{k,l} & \text{otherwise} \end{cases}$    Enforce the support and positivity constraints
  $h_{k,l} = \texttt{max}(g'_{k,l}, 0), \quad \forall (k,l) \in [0,s]^2$
  $k = {}^{k}/{\sum_{k,l} k_{k,l}}$                                          Normalize the kernel
  $h = \texttt{thresholding}(h, {}^{1}/{255})$                               Simple kernel denoising
  $k = {}^{k}/{\sum_{k,l} k_{k,l}}$                                          Re-normalize the kernel
```

starts with random values for each sample, it is important to draw enough samples to avoid converging to bad local minima. The reflection of each resulting kernel is also considered, as the autocorrelation is a symmetric operation. Since the phase retrieval problem is shift invariant, kernels are centered around their respective centroid before any deconvolution.

To select the best kernel from all the candidates, a patch of the blurry image is extracted and deconvolved using each candidate. The most likely kernel is selected as the one that minimizes the ℓ_1/ℓ_2 score computed on the gradients of the deconvolved patch. This score can be seen as a normalized sparsity measure [11] and decreases as the sharpness increases. The split Bregman total variation method, described in Section 4, is used to deconvolve the patch.

Kernel support re-estimation. Obtaining an estimation of the kernel h allows to reevaluate the support of its autocorrelation. The kernel is projected for each angle of $\mathcal{A}$ and the autocorrelation of the projections are computed using valid boundary conditions. Once the kernel has been projected, the support is set to the largest k such that $R(P_\theta(h))_k > 0.05 \cdot \max(R(P_\theta(h))$. This refined support

Algorithm 6: `PhaseRetrieval`

```
input  : kernel power spectrum $|H|^2$, kernel size $s$, blurred image $v$
output: kernel $h$ of size $s \times s$
$P$ = extract a high variance $150 \times 150$ patch of $v$ (10 random window samples)
for $j$ from 1 to $N_{tries}$ do
    $h^{(j)} = \texttt{SinglePhaseRetrieval}(|H|, s)$                              Algorithm 5
    $d^{(j)} = \texttt{deconv}(P, h^{(j)})$
    $c^{(j)} = \frac{\sum_{\boldsymbol{x}} |\nabla d_{\boldsymbol{x}}^{(j)}|}{\sqrt{\sum_{\boldsymbol{x}} |\nabla d_{\boldsymbol{x}}^{(j)}|^2}}$          $\nabla d$ computed by finite differences
    $h^{(j+N_{\text{tries}})} = \texttt{reflect}(h^{(j)})$
    $d^{(j+N_{\text{tries}})} = \texttt{deconv}(P, h^{(j+N_{\text{tries}})})$          Evaluate the mirror kernel
    $c^{(j+N\text{tries})} = \frac{\sum_{\boldsymbol{x}} |\nabla d_{\boldsymbol{x}}^{(j+N\text{tries})}|}{\sqrt{\sum_{\boldsymbol{x}} |\nabla d_{\boldsymbol{x}}^{(j+N\text{tries})}|^2}}$
return the kernel $h^{(j)}$ with lowest cost $c^{(j)}$
```

allows to retrieve a more accurate power spectrum and the associated phase in the next iteration. Note that in the implementation of Goldstein and Fattal [7], a Radon transform was used to project the kernel for this step and requires a stretching coefficient to transpose the Radon transform support to the shear projection support. Our implementation uses the same shear projection operator P_θ that is used to project the blurry image.

4 Experiments

In order to evaluate the performance of the kernel estimation, we built a dataset of five blurry images synthetized from five natural images using kernels from Levin et al. [12]. We also used the blurry images of Goldstein and Fattal [7]. Unless otherwise specified, the default parameters were: $N_{\text{outer}} = 3$, $N_{\text{tries}} = 30$, $N_{\text{inner}} = 300$ and $\alpha = 2.1$. These are the values given by the authors of the method [7]. The kernel size, which is a parameter of Algorithm 1, was set to the ground truth kernel size.

Figure 5 shows results of the kernel estimation. Although the exact shape is never completely recovered, the obtained kernels are consistent with the ground truth. The relevance of the estimated kernels can be evaluated through the deconvolution of the blurry images, as shown in Figure 6. These results are obtained with a split Bregman deconvolution algorithm [8, 6] which solves the following variational problem

$$\arg\min_u \frac{1}{2}\|u * h - v\|_2^2 + \lambda \|\nabla u\|_1. \tag{46}$$

In order to mitigate boundary discontinuities which produce artifacts during the deconvolution, the blurry image v is preprocessed with an edge tapering procedure [15]. This tapering is performed by circularly blurring the image using the estimated kernel and blending the two images in order to keep the original image v in the central part of the image domain while keeping the re-blurred image $h * v$ around the boundary.

Parameters of the method. The parameter N_{outer} in Algorithm 1 controls the number of iterations of the method. With only one iteration, the kernel support is not refined and the resulting kernel is often inaccurate. Through our experiments, we concluded that two iterations can be enough to obtain a plausible kernel and more than three iterations do not improve the method performance. This indicates that the support converges quickly to a good estimate.

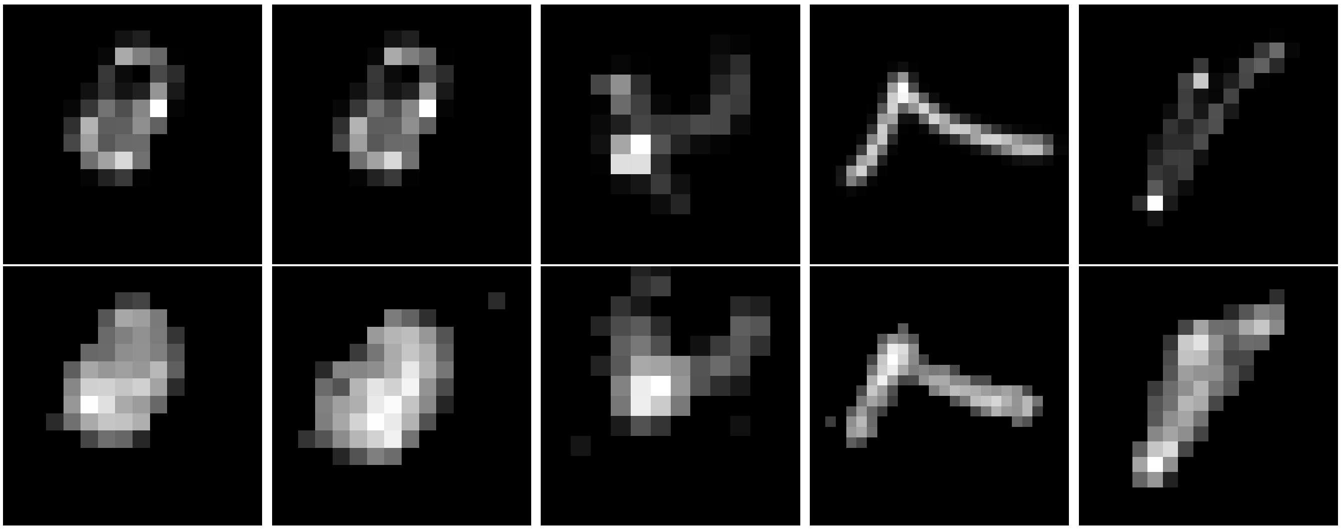


Figure 5: First row: ground truth kernels. Second row: estimated kernels. Images normalized between 0 and 1 for display purposes.

Similarly, the parameter N_{inner} in Algorithm 5, which controls the number of iterations for a single phase retrieval sampling, can be set to a lower value (e.g. 100) without degrading too much the result. This is explained by the fast convergence of the phase retrieval method, even if the random initialization can lead to a local minimum.

The parameter N_{tries} in Algorithm 6 controls how many random phases should be sampled. Because the phase retrieval starts with random values for each sample, it is important to draw enough samples to avoid converging to a local minimum. Figure 7 shows the effect of the parameter N_{tries}.

The input kernel size should be chosen sufficiently large to assure that the kernel is included in the specified domain. However, setting it too large results in a poorer and more time-consuming estimation.

The compensation filter, as described in Section 3.1, controls the sharpness of the estimated kernel. While a fine tuned value allows to retrieve a sharp kernel, we observed that an excessive compensation ($\alpha \approx 1$) may results in incorrect estimations (see Figure 8). In the dataset of Goldstein et al. [7], the images were simulated from sharp ones so the compensation filter has to be close to the identity ($\alpha \to \infty$) which is equivalent to disabling it. Choosing $\alpha = 1$ for this dataset leads to a degradation of the results (Figure 8a). On the other hand, our dataset (Figure 6) was generated from images that contain a small amount of intrinsic blur. For this reason, the compensation filter has to be set to $\alpha \approx 1$ in order to obtain sharp kernels as illustrated in Figure 8d.

Noise robustness. Figure 9 illustrates the influence of the noise on the kernel estimation and deconvolution. The performance starts to decrease significantly for σ larger than 1% of the image dynamic range if the regularization is not adjusted. This fact is demonstrated in Figure 10 which plots the results of the metric $\|I_k - I_{k_{gt}}\|^2$ for various noise levels and a fixed data term weight $\lambda = 2000$, where I_k and $I_{k_{gt}}$ are respectively the deconvolved image with the estimated kernel and with the ground truth kernel.

For higher noise levels, increasing the regularization weight achieves better deconvolution results even when the kernels are not properly estimated. Thus, even if the kernel estimation step is not robust to noise, the deconvolved image may still be acceptable thanks to the regularization of the final deconvolution. Note that during the kernel estimation, after the phase retrieval step, kernels are evaluated by deconvolving a small window of the input image. This deconvolution also requires

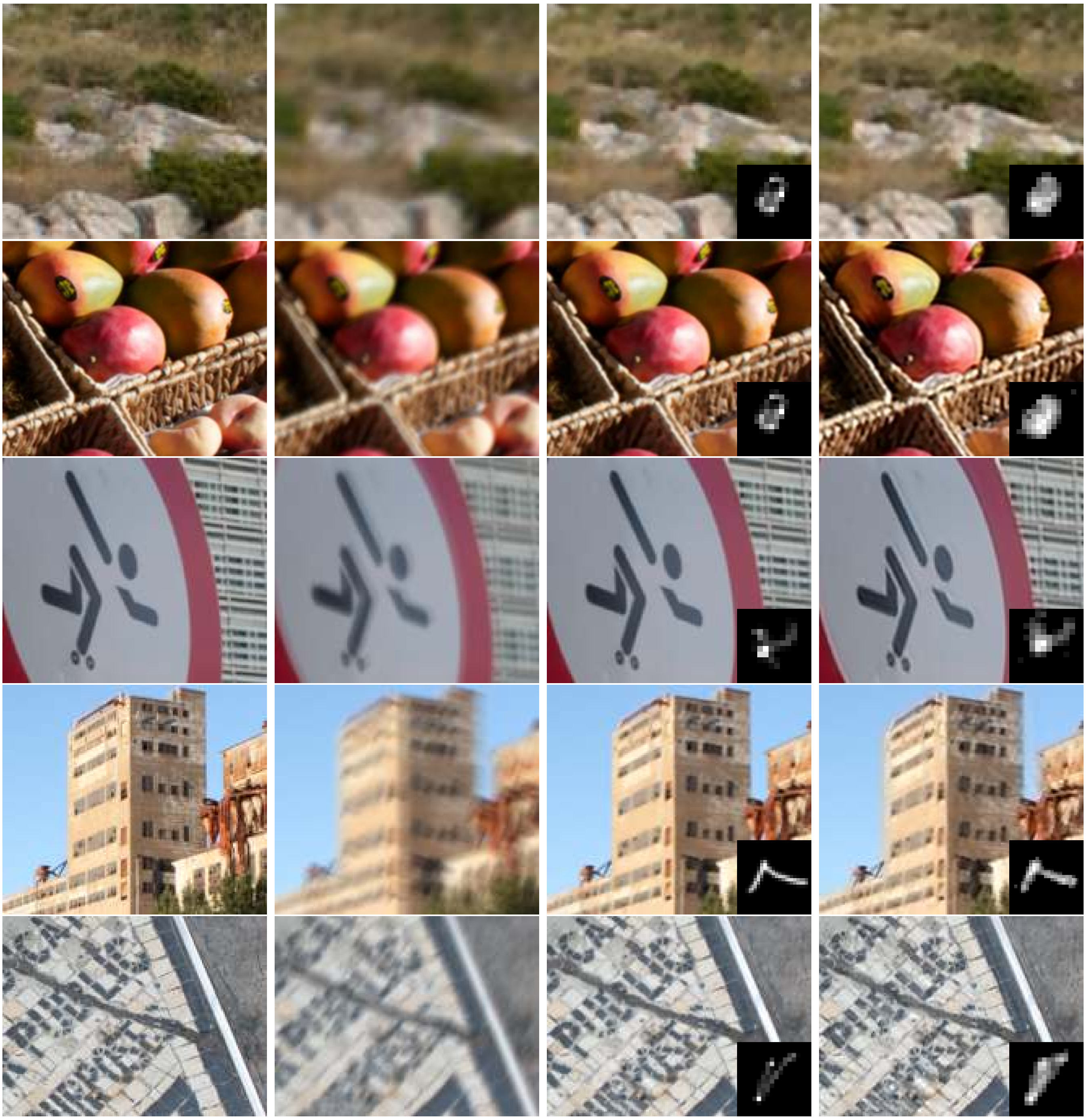


Figure 6: From left to right: original image, blurred image, deblurred image with ground truth kernel, and deblurred image with estimated kernel. Deconvolution results are obtained using the split Bregman algorithm [8].

a regularization. This parameter is of less influence than the final deconvolution one, and we set it to the same value.

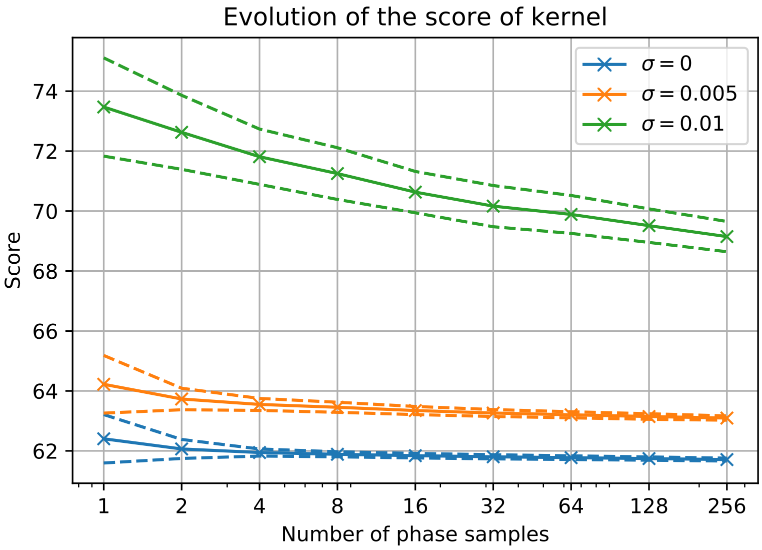
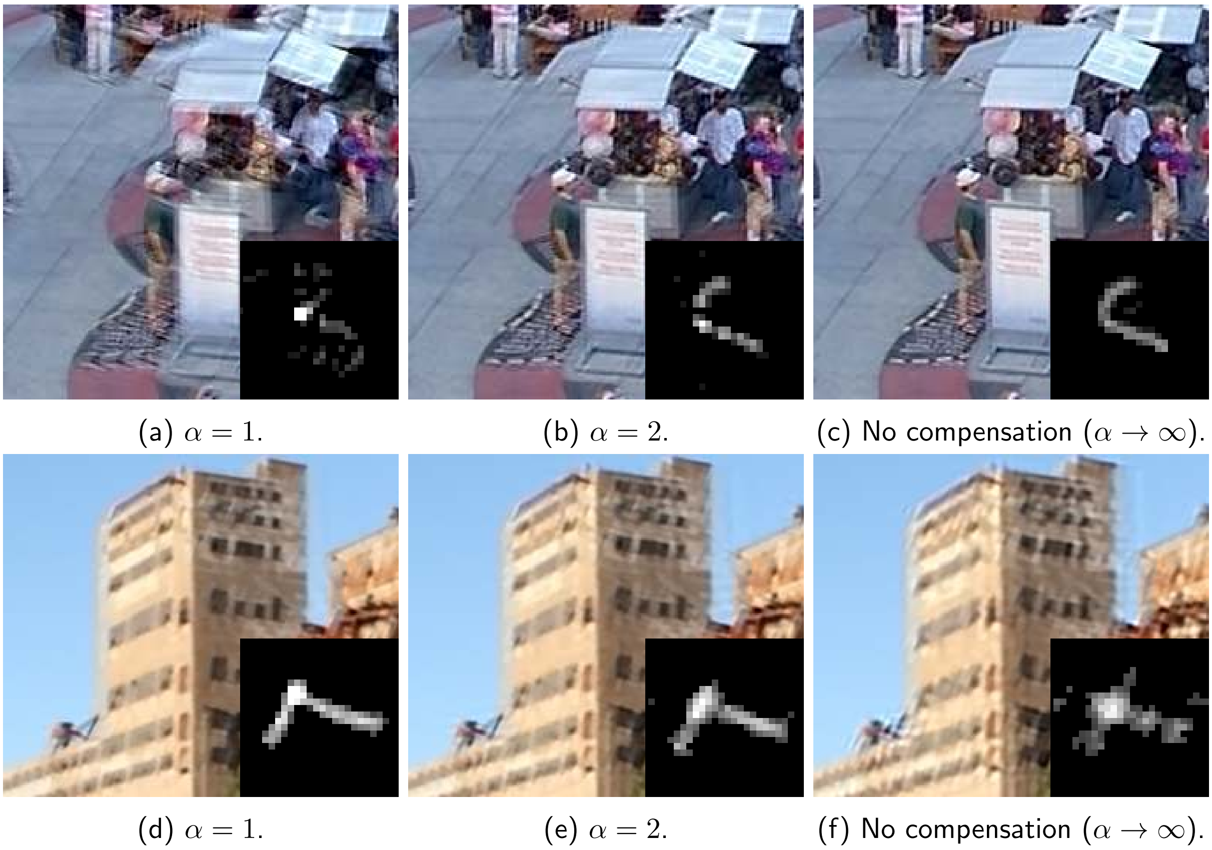


Figure 7: Evolution of the kernel score for different number of phase samples per run at three different noise levels. Each point corresponds to the mean score of 100 runs and the dash lines represents the mean +/− standard deviation.



(a) $\alpha = 1$.

(b) $\alpha = 2$.

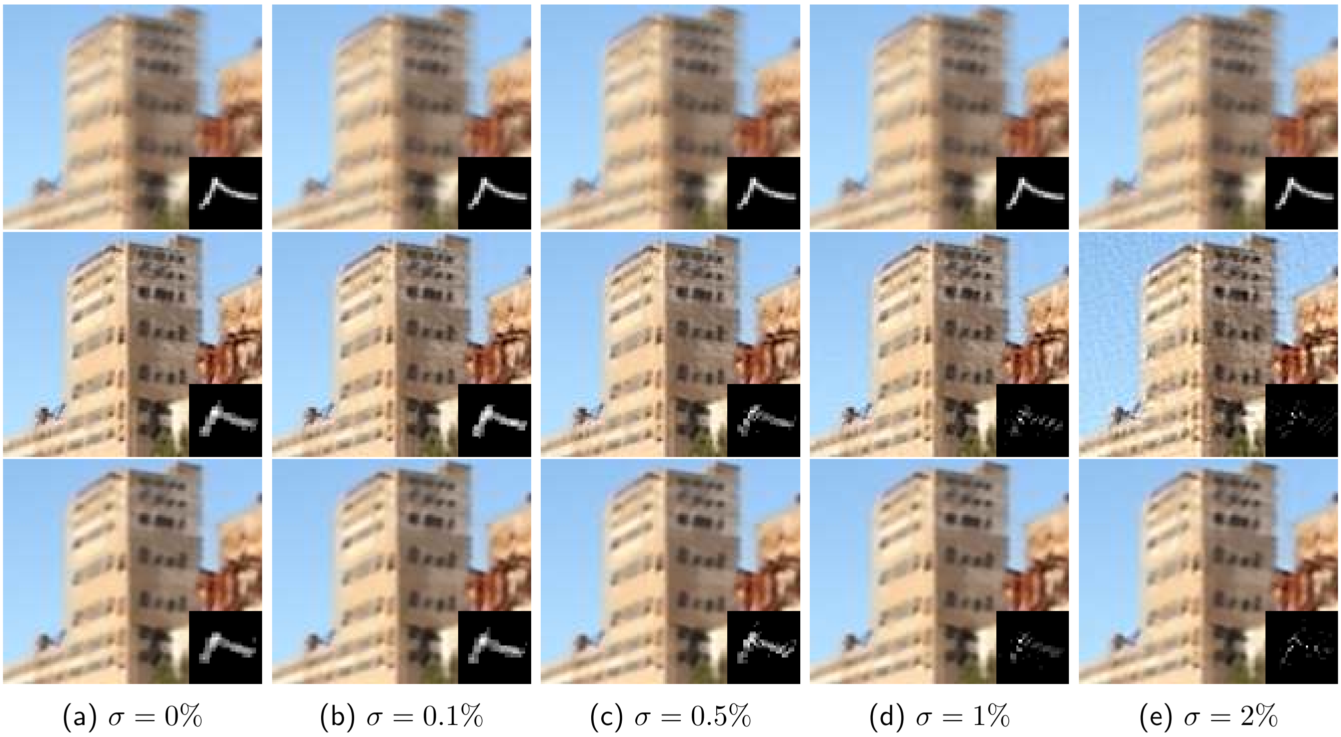
(c) No compensation ($\alpha \to \infty$).

(d) $\alpha = 1$.

(e) $\alpha = 2$.

(f) No compensation ($\alpha \to \infty$).

Figure 8: Effect of the compensation filter on the estimated kernel. The first row shows deconvolution results computed on an image without intrinsic blur, while the image used in the second row has some intrinsic blur so the compensation filter permits to improve the result.



(a) $\sigma = 0\%$ (b) $\sigma = 0.1\%$ (c) $\sigma = 0.5\%$ (d) $\sigma = 1\%$ (e) $\sigma = 2\%$

Figure 9: Influence of the noise in the kernel estimation. Top row: noisy blurred input with ground truth kernel, middle row: estimated kernel and deconvolution with high data term weight ($\lambda = 2000$), bottom row: estimated kernel and deconvolution with low data term weight ($\lambda = 200$).

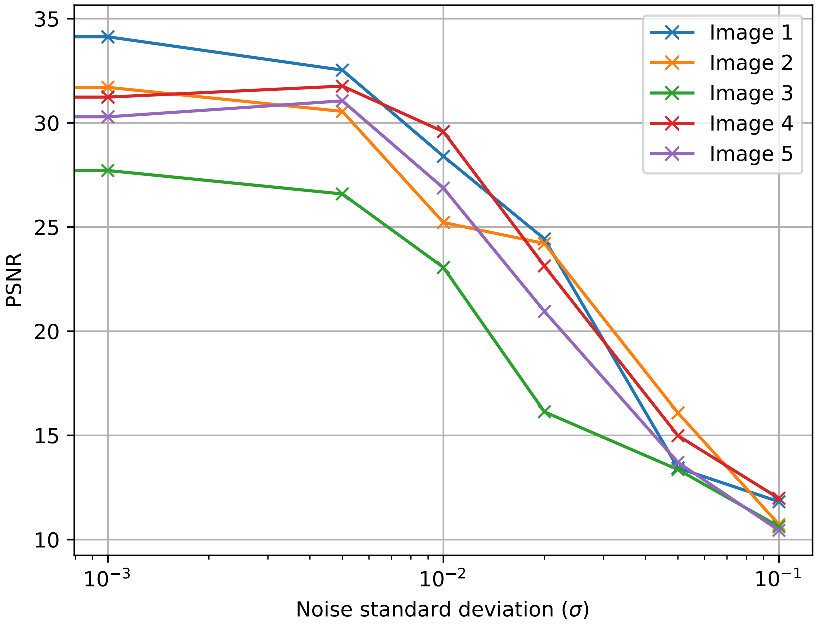


Figure 10: Influence of the noise in the final deblurred image.

Comparison to the original implementation. Figure 11 shows kernel estimation and deconvolution results on the dataset of Goldstein et al. [7], comparing their implementation with the one proposed in this paper. We notice that the estimated kernels and the deconvolution results are almost identical.

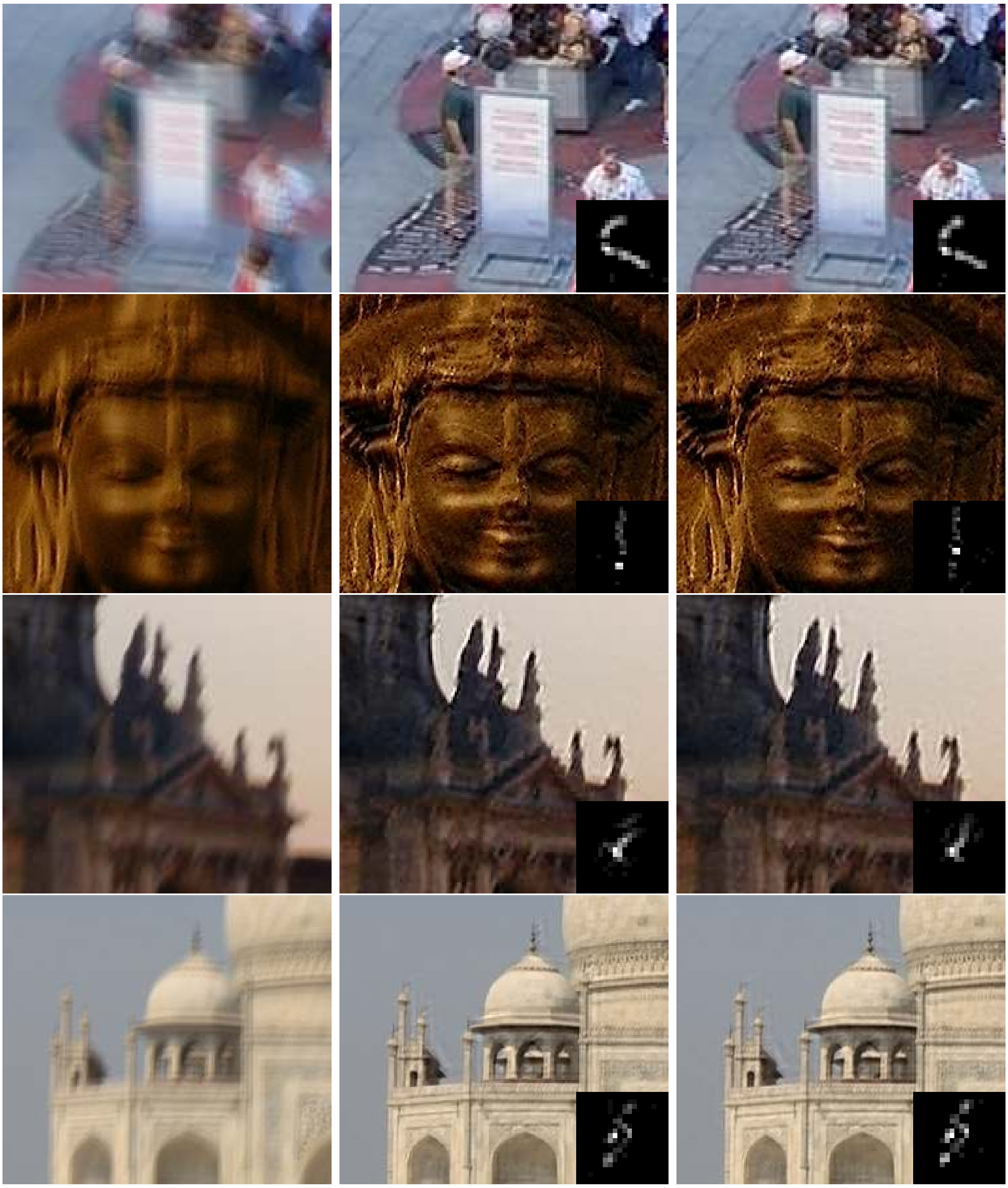


Figure 11: From left to right: blurry image, deblurred image using the implementation by Goldstein and Fattal [7], and deblurred image using our implementation.

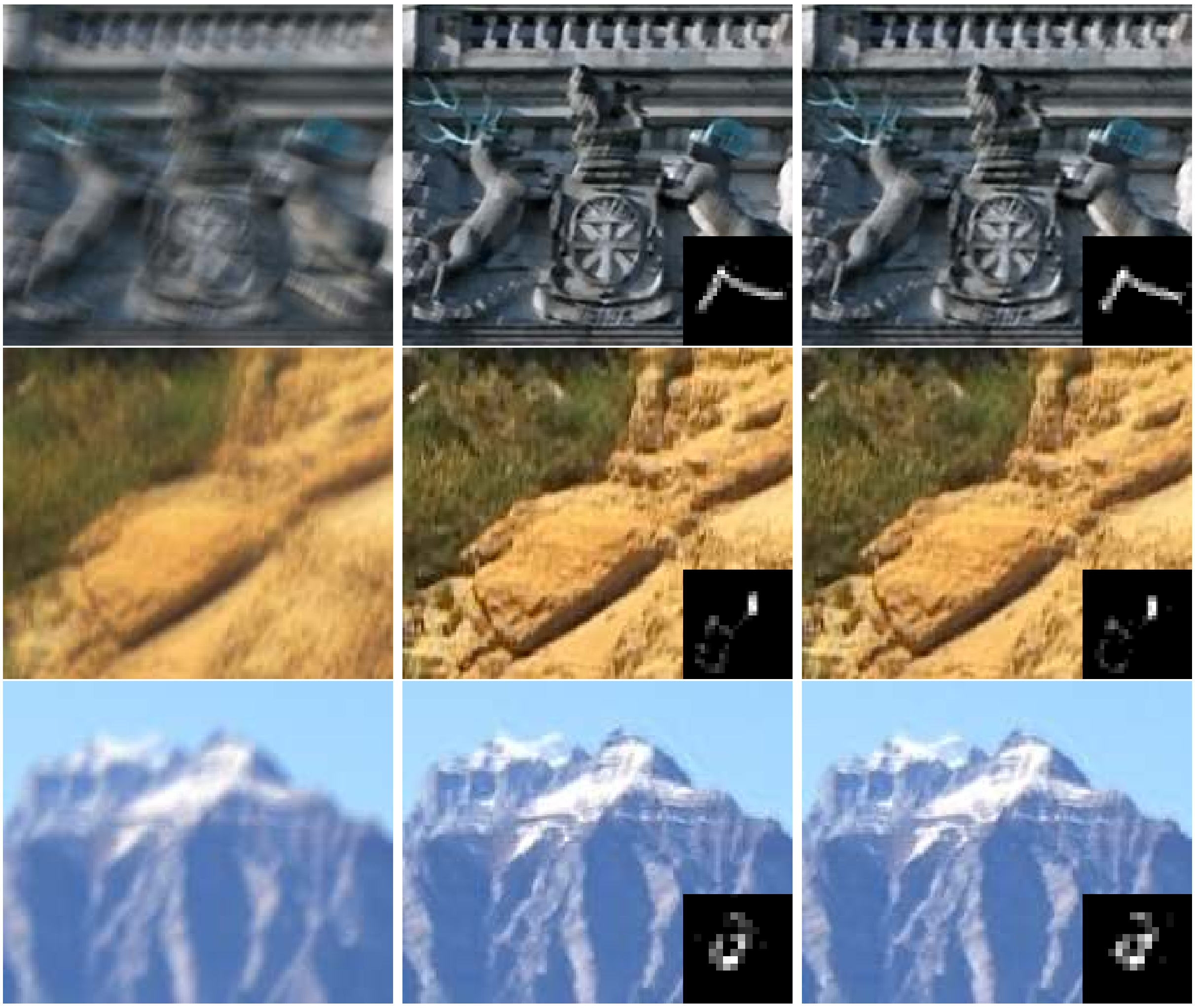


Figure 11 (cont.): From left to right: blurry image, deblurred image using the implementation by Goldstein and Fattal [7], and deblurred image using our implementation.

5 Conclusions

Estimating the blur kernel of an image is the first step towards its deblurring. In this paper, we detailed a kernel estimation method proposed by Goldstein and Fattal published in 2012 which estimates the kernel by modeling statistical irregularities exhibited in the power spectrum of blurred images. By using an anisotropic power-law model to characterize natural images, the method is capable of reconstructing the power spectrum of the underlying blur kernel. Then, a phase retrieval algorithm is used to obtain the kernel in the spatial domain, which can be used to deconvolve the observed blurry image.

We detailed the mathematical model used by the method and described the proposed numerical algorithm and its parameters. Each key component of the method, including the numerical optimizations, are explained in this paper and a parallel C++ implementation is provided.

The method is able to estimate complex motion blur kernels from slightly noisy images. Under more severe conditions, the estimated kernel only keeps the general shape of the true kernel. Nevertheless, strong regularization during the deconvolution step mitigates the kernel estimation errors. Furthermore, even though large input images are required because of the nature of the algorithm, the computational cost scales linearly with the size of the kernel. Our implementation is able to handle high definition images in reasonable time.

A Approximation of $r_\theta^\epsilon(x)$

Let us analyze the term $\hat{r}_\theta^\epsilon(|\xi|)$. Its 1D inverse Fourier transform is

$$r_\theta^\epsilon(x) = c_\theta \cdot R(P_\theta(h))(x) * \left(\frac{\sqrt{\epsilon c_\theta}}{2} e^{-\sqrt{\epsilon c_\theta}|x|} \right). \tag{47}$$

In what follows we show that the derivative of $r_\theta^\epsilon(x)$ is arbitrary small so this term can be approximated by a constant value μ_θ. Let us define $f(x) = c_\theta \cdot R(P_\theta(h))(x)$ and $\psi_a(x) = \frac{a}{2} e^{-a|x|}$, where $a = \sqrt{c_\theta \epsilon}$. Then,

$$\begin{aligned} r(x) &= f * \psi_a(x) && (48) \\ &= \int_{-\infty}^{\infty} f(s) \psi_a(x-s) ds && (49) \\ &= \frac{a}{2} \int_{-\infty}^{x} f(s) e^{-a(x-s)} ds + \frac{a}{2} \int_{x}^{+\infty} f(s) e^{-a(s-x)} ds. && (50) \end{aligned}$$

From Leibniz's differentiation rule, we have:

$$\begin{aligned} |r'(x)| &= \left| -\frac{a^2}{2} \int_{-\infty}^{x} f(s) e^{-a(x-s)} ds + \frac{a^2}{2} \int_{x}^{+\infty} f(s) e^{-a(s-x)} ds \right| && (51) \\ &\leq \frac{a^2 e^{-ax}}{2} \int_{-x}^{\infty} |f(s)| e^{-as} ds + \frac{a^2 e^{ax}}{2} \int_{x}^{+\infty} |f(s)| e^{-as} ds && (52) \\ &\leq \frac{a^2 \|f\|_\infty e^{-ax}}{2} \int_{-x}^{\infty} e^{-as} ds + \frac{a^2 \|f\|_\infty e^{ax}}{2} \int_{x}^{+\infty} e^{-as} ds && (53) \\ &\leq \frac{a \|f\|_\infty e^{-ax}}{2} e^{ax} + \frac{a \|f\|_\infty e^{ax}}{2} e^{-ax} && (54) \\ &\leq a \|f\|_\infty, \quad \forall x \in \mathbb{R}. && (55) \end{aligned}$$

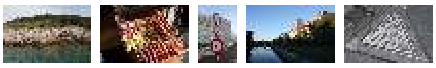
Where we have assumed that $f(s) = f(-s)$.

Thus, if ϵ is very small then $r'(x)$ is also small, so $r(x)$ is almost constant.

Acknowledgment

We thank Jean-Michel Morel, Thibaud Briand and Enric Meinhardt-Llopis for fruitful comments and discussions. Work partly financed by BPI France and Région Ile-de-France FUI 18 Plein Phare, Agencia Nacional de Investigación e Innovación (ANII, Uruguay) grant FCE_1_2017_135458; Office of Naval research grant N00014-17-1-0023, Programme ECOS Sud – UdelaR - Paris Descartes U17E04, n°ANR-12-ASTR-0035 DGA project; DGA PhD scholarship jointly supported with FMJH.

Image Credits



Nicola Pierazzo

Goldstein and Fattal [7]

References

[1] H.H. Bauschke, P.L. Combettes, and D. R. Luke, *Hybrid projection-reflection method for phase retrieval*, Journal of the Optical Society of America A, 20 (2003), p. 1025. https://doi.org/10.1364/JOSAA.20.001025.

[2] J-M. Bony, *Cours d'analyse: théorie des distributions et analyse de Fourier*, Editions Ecole Polytechnique, 2001.

[3] V. Elser, *Phase retrieval by iterated projections*, Journal of the Optical Society of America A, 20 (2003), pp. 40–55. https://doi.org/10.1364/JOSAA.20.000040.

[4] D.J. Field, *Relations between the statistics of natural images and the response properties of cortical cells*, Journal of the Optical Society of America A, 4 (1987), pp. 2379–2394. https://doi.org/10.1364/JOSAA.4.002379.

[5] J. R. Fienup, *Phase retrieval algorithms: a comparison*, Applied Optics, 21 (1982), pp. 2758–2769. https://doi.org/10.1364/AO.21.002758.

[6] 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.

[7] A. Goldstein and R. Fattal, *Blur-kernel estimation from spectral irregularities*, in European Conference on Computer Vision (ECCV), Springer, 2012, pp. 622–635. https://doi.org/10.1007/978-3-642-33715-4_45.

[8] T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 323–343. https://doi.org/10.1137/080725891.

[9] M.R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, 1952. https://doi.org/10.6028/jres.049.044.

[10] W. Hu, J. Xue, and N. Zheng, *PSF estimation via gradient domain correlation*, IEEE Transactions on Image Processing, 21 (2012), pp. 386–392. https://doi.org/10.1109/TIP.2011.2160073.

[11] D. Krishnan, T. Tay, and R. Fergus, *Blind deconvolution using a normalized sparsity measure*, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011. https://doi.org/10.1109/CVPR.2011.5995521,.

[12] A. Levin, Y. Weiss, F. Durand, and W.T Freeman, *Understanding and evaluating blind deconvolution algorithms*, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009. https://doi.org/10.1109/CVPR.2009.5206815.

[13] D.R. Luke, *Relaxed averaged alternating reflections for diffraction imaging*, Inverse Problems, 21 (2005), pp. 37–50. https://doi.org/10.1088/0266-5611/21/1/004.

[14] T. Pouli, D.W. Cunningham, and E. Reinhard, *Image statistics and their applications in computer graphics*, in Eurographics (STARs), 2010, pp. 83–112.

[15] 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.

[16] A. Torralba and A. Oliva, *Statistics of natural image categories*, Network: computation in neural systems, 14 (2003), pp. 391–412.

[17] R. Wang and D. Tao, *Recent progress in image deblurring*, arXiv preprint arXiv:1409.6838, (2014).

[18] D. P Wipf and H. Zhang, *Revisiting Bayesian blind deconvolution*, Journal of Machine Learning Research, 15 (2014), pp. 3595–3634.

[19] Y. Yitzhaky, I. Mor, A. Lantzman, and N.S. Kopeika, *Direct method for restoration of motion-blurred images*, Journal of the Optical Society of America A, 15 (1998), pp. 1512–1519. https://doi.org/10.1364/JOSAA.15.001512.