Total Variation Blind Deconvolution: The Devil is in the Details

Daniele Perrone
University of Bern
Bern, Switzerland
perrone@iam.unibe.ch

Paolo Favaro
University of Bern
Bern, Switzerland
paolo.favaro@iam.unibe.ch

Abstract

In this paper we study the problem of blind deconvolution. Our analysis is based on the algorithm of Chan and Wong [2] which popularized the use of sparse gradient priors via total variation. We use this algorithm because many methods in the literature are essentially adaptations of this framework. Such algorithm is an iterative alternating energy minimization where at each step either the sharp image or the blur function are reconstructed. Recent work of Levin et al. [14] showed that any algorithm that tries to minimize that same energy would fail, as the desired solution has a higher energy than the no-blur solution, where the sharp image is the blurry input and the blur is a Dirac delta. However, experimentally one can observe that Chan and Wong's algorithm converges to the desired solution even when initialized with the no-blur one. We provide both analysis and experiments to resolve this paradoxical conundrum. We find that both claims are right. The key to understanding how this is possible lies in the details of Chan and Wong's implementation and in how seemingly harmless choices result in dramatic effects. Our analysis reveals that the delayed scaling (normalization) in the iterative step of the blur kernel is fundamental to the convergence of the algorithm. This then results in a procedure that eludes the no-blur solution, despite it being a global minimum of the original energy. We introduce an adaptation of this algorithm and show that, in spite of its extreme simplicity, it is very robust and achieves a performance comparable to the state of the art.

Blind deconvolution is the problem of recovering a signal and a degradation kernel from their noisy convolution. This problem is found in diverse fields such as astronomical imaging, medical imaging, (audio) signal processing, and image processing. Yet, despite over three decades of research in the field (see [11] and references therein), the design of a principled, stable and robust algorithm that can handle real images remains a challenge. However, present-day progress has shown that recent models for sharp images and blur kernels, such as total variation [18], can yield remarkable results [7, 19, 4, 23, 13].

Many of these recent approaches are evolutions of the variational formulation [26]. A common element in these methods is the explicit use of priors for both blur and sharp image to encourage smoothness in the solution. Among these recent methods, *total variation* emerged as one of the most popular priors [2, 25]. Such popularity is probably due to its ability to encourage gradient sparsity, a property that can describe many signals of interest well [10].

However, recent work by Levin et al. [14] has shown that the joint optimization of both image and blur kernel can have the no-blur solution as its global minimum. That is to say, a wide selection of prior work in blind image deconvolution either is a local minimizer and, hence, requires a lucky initial guess, or it cannot depart too much from the no-blur solution. Nonetheless, several algorithms based on the joint optimization of blur and sharp image show good convergence behavior even when initialized with the no-blur solution [2, 19, 4, 23].

This incongruence called for an in-depth analysis of total variation blind deconvolution (TVBD). We find both experimentally and analytically that the analysis of Levin et al. [14] correctly points out that between the no-blur and the desired solution, the energy minimization favors the no-blur solution. However, we also find that the algorithm of Chan and Wong [2] converges to the desired solution, even when starting at the no-blur solution. We illustrate that the specific implementation of [2] does not minimize the originally defined energy. This algorithm separates some constraints from the gradient descent step and then applies them sequentially. When the cost functional is convex this alteration does not have a major impact. However, in blind deconvolution, where the cost functional is not convex, this completely changes the convergence behavior. Indeed, we show that if one imposed all the constraints simultaneously, as required, then the algorithm would never leave the no-blur solution independently of regularization. This analysis suggests that the scaling mechanism induces a novel type of image prior.

Our main contributions are: 1) We illustrate the behav-

ior of TVBD [2] both with carefully designed experiments and with a detailed analysis of the reconstruction of fundamental signals; 2) For a family of image priors we present a novel concise proof of the main result of [14], which showed how (a variant of) TVBD converges to the no-blur solution; 3) We strip TVBD of all recent improvements (such as filtering [19, 4, 14], blur kernel prior [2, 25], edge enhancement via shock filter [4, 23]) and clearly show the core elements that make it work; 4) We show how the use of proper boundary conditions can improve the results, and also apply the algorithm on current datasets and compare to the state of the art methods. Notwithstanding the simplicity of the algorithm, we obtain a comparable performance to the top performers.

1. Blur Model and Priors

Suppose that a blurry image f can be modeled by

$$f = k_0 * u_0 + n \tag{1}$$

where k_0 is a blur kernel, u_0 a sharp image, n noise and $k_0 * u_0$ denotes convolution between k_0 and u_0. Given only the blurry image, one might want to recover both the sharp image and the blur kernel. This task is called *blind deconvolution*. A classic approach to this problem is to solve the following regularized minimization

$$\min_{u,k} \|k * u - f\|_2^2 + \lambda J(u) + \gamma G(k) \tag{2}$$

where the first term enforces the convolutional blur model (data fitting), the functionals $J(u)$ and $G(k)$ are the smoothness priors for u and k (for example, Tikhonov regularizers [21] on the gradients), and λ and γ two nonnegative parameters that weigh their contribution. Furthermore, additional constraints on k, such as positivity of its entries and integration to 1, can be included. For any $\lambda > 0$ and $\gamma > 0$ the cost functional will not have as global solution neither the true solution $(k_0,\ u_0)$ nor the *no-blur solution* $(k = \delta,\ u = f)$, where δ denotes the Dirac delta. Indeed, eq. (2) will find an optimal tradeoff between the data fitting term and the regularization term. Nonetheless, one important aspect that we will discuss later on is that both the true solution $(k_0,\ u_0)$ and the *no-blur solution* make the data fitting term in eq. (2) equal to zero. Hence, we can compare their cost in the functional simply by evaluating the regularization terms. Notice also that the minimization objective in eq. (2) is non-convex, and, as shown in Fig. 2, has several local minima.

2. Prior work

A choice for the regularization terms proposed by You and Kaveh [26] is $J(u) = ||\nabla u||_2$ and $G(k) = ||k||_2$. Unfortunately, the $||.||_2$ norm is not able to model the sparse nature of common image and blur gradients and results in sharp images that are either oversmoothed or have ringing artifacts. Cho and Lee [4] and Xu and Jia [23] have reduced the generation of artifacts by using heuristics to select sharp edges.

An alternative to $||.||_2$ is the use of *total variation* (TV) [2, 25, 3, 9]. TV regularization was firstly introduced for image denoising in the seminal work of Rudin, Osher and Fatemi [18], and since then it has been applied successfully in many image processing applications.

You and Kaveh [25] and Chan and Wong [2] have proposed the use of TV regularization in blind deconvolution on both u and k. They also consider the following additional convex constraints to enhance the convergence of their algorithms

$$\|k\|_1 \doteq \int |k(\mathbf{x})| d\mathbf{x} = 1, \quad k(\mathbf{x}) \geq 0,\ u(\mathbf{x}) \geq 0 \tag{3}$$

where with $\mathbf{x}$ we denote either 1D or 2D coordinates. He *et al.* [9] have incorporated the above constraints in a variational model, claiming that this enhances the stability of the algorithm. However, despite these additional constraints, the cost function (2) still suffers from local minima.

A different approach is a strategy proposed by Wang *et al.* [22] that seeks for the desired local minimum by using downsampled reconstructed images as priors during the optimization in a multi-scale framework. Other methods use some variants of total variation that nonetheless share similar properties. Among these, the method of Xu and Jia [23] uses a Gaussian prior together with edge selection heuristics, and, recently, Xu *et al.* [24] have proposed an approximation of the L_0-norm as a sparsity prior. However, all above adaptations of TVBD work with a non-convex minimization problem.

Blind deconvolution has also been analyzed through a Bayesian framework, where one aims at maximizing the posterior distribution (MAP)

$$\arg\max_{u,k} p(u, k|f) = \arg\max_{u,k} p(f|u, k)p(u)p(k). \tag{4}$$

$p(f|u, k)$ models the noise affecting the blurry image; a typical choice is the Gaussian distribution [7, 13] or an exponential distribution [23]. $p(u)$ models the distribution of typical sharp images, and it is typically a heavy-tail distribution of the image gradients. $p(k)$ is the prior knowledge about the blur function, and that is typically a Gaussian distribution [23, 4], a sparsity-inducing distribution [7, 19] or a uniform distribution [13]. Under these assumptions on the conditional probability density functions $p(f|u, k)$, $p(u)$, and $p(k)$ and by computing the negative log-likelihood of the above cost, one can find the correspondence between eq. (2) and eq. (4).

Since under these assumptions the $MAP_{u,k}$ problem (4) is equivalent to problem (2), also the Bayesian approach suffers from local minima. Levin *et al.* [13] and Fergus *et*

al. [7] propose to address the problem by marginalizing over all possible sharp images u and thus solve the following MAP_k problem

$$\arg\max_k p(k|f) = \arg\max_k \int p(f|u,k)p(u)p(k)du. \quad (5)$$

Then they estimate u by solving a convex problem where k is given from the previous step. Since the right hand side of problem (5) is difficult to compute, in practice an approximation is used.

Levin *et al.* [14] recently argued that one should use the formulation (5) (MAP_k) rather than (4) ($MAP_{u,k}$). They have shown that, using a sparse-inducing prior for the image gradients and a uniform distribution for the blur, the $MAP_{u,k}$ approach favors the no-blur solution ($u = f,\ k = \delta$), for images blurred with a large blur. In addition, they have shown that, for sufficiently large images, the MAP_k approach converges to the true solution.

A recent work from Rameshan *et al.* [17] shows that the above conclusions are valid only when one uses a uniform prior for the blur kernel, and that with the use of a sparse prior the $MAP_{u,k}$ does not attain the no-blur solution.

3. Problem Formulation

By combining problem (2) with the constraints in eq. (3) and by setting $\gamma = 0$, we study the following minimization

$$\begin{array}{ll} \min_{u,k} & ||k * u - f||_2^2 + \lambda J(u) \\ \text{subject to} & k \geqslant 0, \quad \|k\|_1 = 1 \end{array} \quad (6)$$

where $J(u) = ||u||_{BV} \doteq \int ||\nabla u(\mathbf{x})||_2 d\mathbf{x}$ or $J(u) = ||u_x||_1 + ||u_y||_1$, with $\nabla u \doteq [u_x\ u_y]^T$ the gradient of u and $\mathbf{x} \doteq [x\ y]^T$, and $\|k\|_1$ corresponds to the L^1 norm in eq. (3). To keep the analysis simple we have stripped the formulation of all unnecessary improvements such as using a basis of filters in $J(u)$ [19, 4, 14], or performing some selection of regions by reweighing the data fitting term [23], or enhancing the edges of the blurry image f [23, 4]. Compared to previous methods, we also do not use any regularization on the blur kernel ($\gamma = 0$).

The formulation in eq. (6) involves the minimization of a constrained non-convex optimization problem. Also, notice that if (u, k) is a solution, then $(u(\mathbf{x}+\mathbf{c}), k(\mathbf{x}+\mathbf{c}))$ are solutions as well for any $\mathbf{c} \in \mathbf{R}^2$. If the additional constraints on k were not used, then the ambiguities would also include $(\alpha_1 u, \frac{1}{\alpha_1} k)$ for non zero α_1.

3.1. Analysis of Relevant Local Minima

Levin *et al.* [13] have shown that eq. (6) favors the no-blur solution (f, δ), when $J(u) = \int |u_x(\mathbf{x})|^\alpha + |u_y(\mathbf{x})|^\alpha d\mathbf{x}$, for any $\alpha > 0$ and either the true blur k_0 has a large support or $||k_0||_2^2 \ll 1$. In the following theorem we show that the above result is also true for any kind of blur kernels and for an image prior with $\alpha \geq 1$.

Theorem 3.1 *Let $J(u) = \|\nabla u\|_p \doteq \left(\int \|\nabla u(\mathbf{x})\|_p^p d\mathbf{x}\right)^{\frac{1}{p}}$, with $p \in [1, \infty]$, f be the noise-free input blurry image ($n = 0$) and u_0 the sharp image. Then,*

$$J(f) \leq J(u_0). \quad (7)$$

Proof. Because f is noise-free, $f = k_0 * u_0$; since the convolution and the gradient are linear operators, we have

$$J(f) = \|\nabla(k_0 * u_0)\|_p = \|k_0 * \nabla u_0\|_p \quad (8)$$

By applying Young's inequality [1] (see Theorem 3.9.4, pages 205-206) we have

$$J(f) = \|k_0 * \nabla u_0\|_p \leq \|k_0\|_1 \|\nabla u_0\|_p = \|\nabla u_0\|_p \doteq J(u_0) \quad (9)$$

since $\|k_0\|_1 = 1$. ∎

Since the first term (the data fitting term) in problem (6) is zero for both the no-blur solution (f, δ) and the true solution (u_0, k_0), Theorem 3.1 states that the no-blur solution has always a smaller, or at most equivalent, cost than the true solution. Notice that Theorem 3.1 is also valid for any $J(u) = \|\nabla u\|_p^r$ for any $r > 0$. Thus, it includes as special cases the Gaussian prior $J(u) = ||u||_{H^1}$, when $p = 2$, $r = 2$, and the anisotropic total variation prior $J(u) = \|u_x\|_1 + \|u_y\|_1$, when $p = 1$, $r = 1$.

Theorem 3.1 highlights a strong limitation of the formulation (6): The exact solution can not be retrieved when an iterative minimizer is initialized at the no-blur solution.

3.2. Exact Alternating Minimization

A solution to problem (6) can be readily found by an iterative algorithm that alternates between the estimation of the sharp image given the kernel and the estimation of the kernel given the sharp image. This approach, called *alternating minimization* (AM), requires the solution of an unconstrained convex problem in u

$$u^{t+1} \leftarrow \min_u ||k^t * u - f||_2^2 + \lambda J(u) \quad (10)$$

and a constrained convex problem in k

$$\begin{array}{lll} k^{t+1} \leftarrow & \min_k & ||k * u^t - f||_2^2 \\ & \text{subject to} & k \geqslant 0, \quad \|k\|_1 = 1. \end{array} \quad (11)$$

Unfortunately, as we show next, this algorithm suffers from the limitations brought out by Theorem 3.1, and without a careful initialization could get stuck on the no-blur solution.

3.3. Projected Alternating Minimization

To minimize problem (6) (with the additional kernel regularization term) Chan and Wong [2] suggest a variant of the AM algorithm that employs a gradient descent algorithm for each step and enforces the constraints on the blur kernel in

a subsequent step. The gradient descent results in the following update for the sharp image u at the t-th iteration

$$u^t \leftarrow u^t - \epsilon \left(k_-^t * (k^t * u^t - f) - \lambda \nabla \cdot \frac{\nabla u^t}{\|\nabla u^t\|_2} \right) \quad (12)$$

for some step $\epsilon > 0$ and where $k_-(\mathbf{x}) = k(-\mathbf{x})$. The above iteration is repeated until the difference between the updated and the previous estimate of u are below a certain threshold. The iteration on the blur kernel k is instead given by

$$k^t \leftarrow k^t - \epsilon \left(u_-^t * (k^t * u^t - f) \right) \quad (13)$$

where, similarly to the previous step, the above iteration is repeated until a convergence criterion, similar to the one on u, is satisfied. The last updated k^t is used to set $k^{t+1/3} \leftarrow k^t$.

Then, one applies the constraints on the blur k via a sequential projection, *i.e.*,

$$k^{t+2/3} \leftarrow \max\{k^{t+1/3}, 0\}, \qquad k^{t+1} \leftarrow \frac{k^{t+2/3}}{\|k^{t+2/3}\|_1}. \quad (14)$$

We call this iterative algorithm the *projected alternating minimization* (PAM). The choice of imposing the constraints sequentially rather than during the gradient descent on k seems a rather innocent and acceptable approximation of the correct procedure (AM). However, this is not the case and we will see that without this arrangement one would not achieve the desired solution.

3.4. Analysis of PAM

Our first claim is that this procedure does not minimize the original problem (6). To support this claim we start by showing some experimental evidence in Fig. 2. In this test we work on a 1D version of the problem. We blur a hat function with one blur of size 3 pixels, and we show the minimum of eq. (6) for all possible feasible blurs. Since the blur has only 3 nonnegative components and must add up to 1, we only have 2 free parameters bound between 0 and 1. Thus, we can produce a 2D plot of the minimum of the energy with respect to u as a function of these two parameters. The blue color denotes a small cost, while the red color denotes a large cost. The figures reveal three local minima at the corners, due to the 3 different shifted versions of the no-blur solution, and the local minimum at the true solution ($k_0 = [0.4, 0.3, 0.3]$) marked with a yellow triangle. We also show with white dots the path followed by k estimated via the PAM algorithm by starting from one of the no-blur solutions (upper-right corner). Clearly one can see that the PAM algorithm does not follow a minimizing path in the space of solutions of problem (6), and therefore does not minimize the same energy.

To further stress this point, we present two theoretical results for a class of 1D signals that highlight the crucial difference between the PAM algorithm and the AM algorithm. We look at the 1D case because analysis of the total variation solution of problem (6) is still fairly limited and a closed-form solution even for a restricted family of 2D signals is not available. Still, analysis in 1D can provide practical insights. The proof exploits recent work of Condat [5], Strong and Chan [20] and the *taut string algorithm* of Davies and Kovacs [6].

Theorem 3.2 *Let f be a 1D discrete noise-free signal, such that $f = k_0 * u_0$, where u_0 and k_0 are two unknown functions and $*$ is the circular convolution operator. Let us also constrain k_0 to be a blur of support equal to 3 pixels, and u_0 to be a step function*

$$u_0[x] = \begin{cases} -U & x \in [-L, -1] \\ U & x \in [0, L-1] \end{cases} \quad (15)$$

for some parameters U and L. We impose that $L \geq 2$ and $U > 0$. Then f will have the following form

$$f[x] = \begin{cases} \delta_1 - U & x = -L \\ -U & x \in [-L+1, -1] \\ \delta_0 - U & x = -1 \\ -\delta_1 + U & x = 0 \\ U & x \in [1, L-2] \\ -\delta_0 + U & x = L-1 \end{cases} \quad (16)$$

for some positive constants δ_0 and δ_1 that depend on the blur parameters. Then, there exists $\lambda \geq \max((L-1)\delta_0 - \delta_1, (L-1)\delta_1 - \delta_0)$ such that the PAM algorithm estimates the true blur $k = k_0$ in two steps, when starting from the no-blur solution (f, δ).

Proof. See [16]. ∎

3.5. Analysis of Exact Alternating Minimization

Theorem 3.2 shows how, for a restricted class of 1D blurs and sharp signals, the PAM algorithm converges to the desired solution in two steps. In the next theorem we show how, for the same class of signals, the exact AM algorithm (see sec. 3.2) can not leave the no-blur solution (f, δ).

Theorem 3.3 *Let f, u_0 and k_0 be the same as in Theorem 3.2. Then, for any $\max((L-1)\delta_0 - \delta_1, (L-1)\delta_1 - \delta_0) < \lambda < UL - \delta_0 - \delta_1$ the AM algorithm converges to the solution $k = \delta$. For $\lambda \geq UL - \delta_0 - \delta_1$ the AM algorithm is unstable.*

Proof. See [16]. ∎

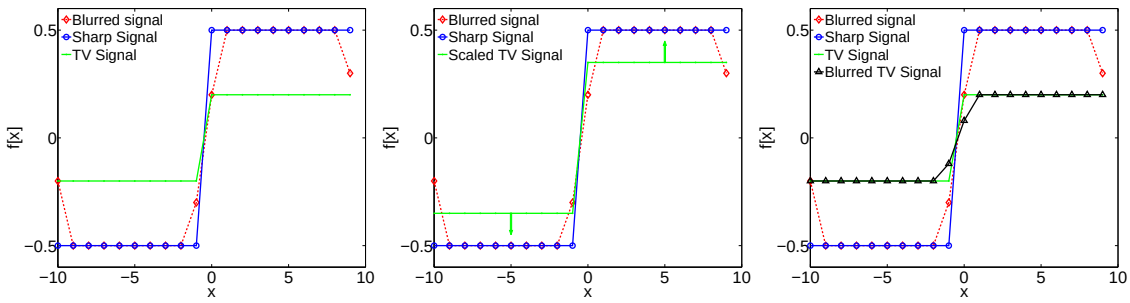


Figure 1. Illustration of Theorem 3.2 and Theorem 3.3 (it is recommended that images are viewed in color). The original step function is denoted by a solid-blue line. The TV signal (green-solid) is obtained by solving $\arg\min_u \|u - f\|_2^2 + \lambda J(u)$. In (a) we show how the first step of the PAM algorithm reduces the contrast of a blurred step function (red-dotted). In (b) we illustrate Theorem 3.2: In the second step of the PAM algorithm, estimating a blur kernel without a normalization constraint is equivalent to scaling the TV signal. In (c) we illustrate Theorem 3.3: If the constraints on the blur are enforced, any blur different from the Dirac delta increases the distance between the input blurry signal and the blurry TV signal (black-solid).

3.6. Discussion

Theorems (3.2) and (3.3) show that with 1D zero-mean step signals and no-blur initialization, for some values of λ PAM converges to the correct blur (and only in 2 steps) while AM does not. The step signal is fundamental to illustrate the behavior of both algorithms at edges in a 2D image. Extensions of the above results to blurs with a larger support and beyond step functions are possible and will be subject of future work.

In Fig. 3 we illustrate two further aspects of Theorem 3.2 (it is recommended that images are viewed in color): 1) the advantage of a non unitary normalization of k during the optimization step (which is a key feature of PAM) and 2) the need for a sufficiently large regularization parameter λ. In the top row images we set $\lambda = 0.1$. Then, we show the cost $\|k * u^1 - f\|_2^2$, with $u^1 = \arg\min_u \|u - f\|_2^2 + \lambda J(u)$, for all possible 1D blurs k with a 3-pixel support under the constraints $\|k\|_1 = 1,\ 1.5,\ 2.5$ respectively. This is the cost that PAM minimizes at the second step when initialized with $k = \delta$. Because k has three components and we fix its normalization, then we can illustrate the cost as a 2D function of $k[1]$ and $k[2]$ as in Fig. 2. However, as the normalization of k grows, the triangular domain of $k[1]$ and $k[2]$ increases as well. Since the optimization of the blur k is unconstrained, the optimal solution will be searched both within the domain and across normalization factors. Thanks to the color coding scheme, one can immediately see that the case of $\|k\| = 1$ achieves the global minimum, and hence the solution is the Dirac delta. However, as we set $\lambda = 1.5$ in the second row or $\lambda = 2.5$ in the bottom row, we can see a shift of the optimal value for non unitary blur normalization values and also for a shift of the global minimum to the desired solution (bottom-right plot).

4. Implementation

In Algorithm 1 we show the pseudo-code of our adaptation of TVBD. At each iteration we perform just one gradient descent step on u and on k because we experimentally noticed that this speeds up the convergence of the algorithm. If the blur is given as input, the algorithm can be adapted to a non-blind deconvolution algorithm by simply removing the update on the blur. We use the notation $\circ$ to denote the discrete convolution operator where the output is computed only on the valid region, *i.e.*, if $f = u \circ k$, with $k \in \mathbf{R}^{h\times w}$, $u \in \mathbf{R}^{m\times n}$, then we have $f \in \mathbf{R}^{(m-h+1)\times(n-w+1)}$. Notice that $k \circ u \neq u \circ k$ in general. Also, $u \circ k$ is not defined if the support of k is too large ($h > m+1$ and $w > n+1$). We use the notation $\bullet$ to denote the discrete convolution operator where the result is the full convolution region, *i.e.*, if $f = u \bullet k$, with $k \in \mathbf{R}^{h\times w}$, $u \in \mathbf{R}^{m\times n}$, then we have $f \in \mathbf{R}^{(m+h-1)\times(n+w-1)}$ with zero padding as boundary condition. As shown in Fig. 5 the proper use of $\bullet$ and $\circ$ as well as the correct choice of the domain of u and k improves the performance of the algorithm. In our algorithm we consider the domain of the sharp image u to be larger than the domain of the blurry image f. If $u \in \mathbf{R}^{m\times n}$, and

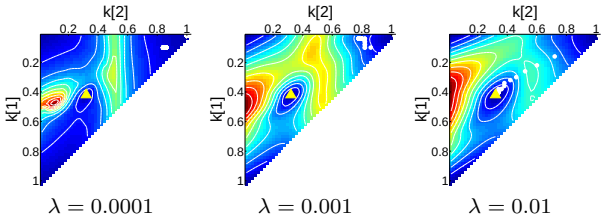


Figure 2. Illustration of Theorem 3.1 (it is recommended that images are viewed in color). In this example we show a 1D experiment where we blur a step function with $k_0 = [0.4; 0.3; 0.3]$. We visualize the cost function of eq. (6) for three different values of the parameter λ. Since the blur integrates to 1, only two of the three components are free to take values on a triangular domain (the upper-left triangle in each image). We denote with a yellow triangle the true blur k_0 and with white dots the intermediate blurs estimated during the minimization via the PAM algorithm. Blue pixels have lower values than the red pixels. Dirac delta blurs are located at the three corners of each triangle. At these locations, as well as at the true blur, there are local minima. Notice how the path of the estimated blur on the rightmost image ascends and then descends a hill in the cost functional.

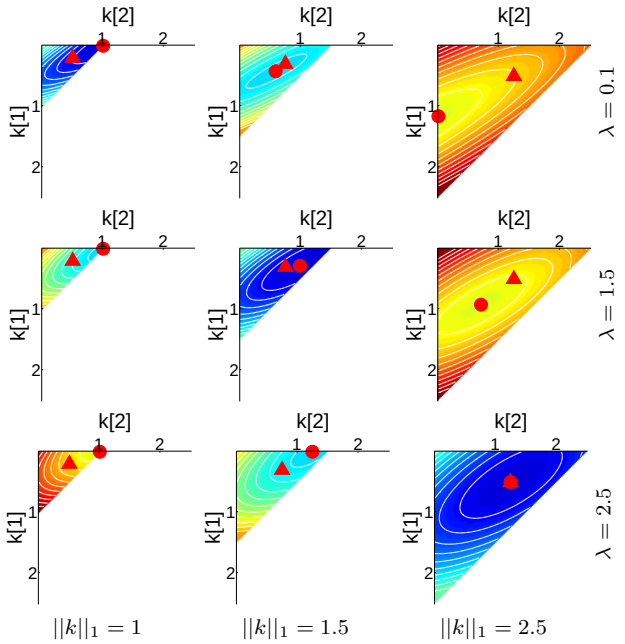


Figure 3. Illustration of Theorem 3.2 (it is recommended that images are viewed in color). Each row represents the visualization of the cost function for a particular value of the parameter λ. Each column shows the cost function for three different blur normalizations: $||k||_1 = 1,\ 1.5,$ and 2.5. We denote the scaled true blur $k_0 = [0.2,\ 0.5,\ 0.3]$ (with $||k||_1 = 1$) with a red triangle and with a red dot the cost function minimum. The color coding is such that: blue $<$ yellow $<$ red; each row shares the same color coding for cross comparison.

Data: f, size of blur, initial large λ, final λ_{min}
Result: u,k

$u_0 \leftarrow \text{pad}(f)$;
$k_0 \leftarrow \text{uniform}$;
while *not converged* **do**
  $u^{t+1} \leftarrow u^t - \epsilon_u \left(k_-^t \bullet (k^t \circ u^t - f) - \lambda \nabla \cdot \frac{\nabla u^t}{|\nabla u^t|} \right)$;
  $k^{t+1/3} \leftarrow k^t - \epsilon_k \left(u_-^{t+1} \circ (k^t \circ u^{t+1} - f) \right)$;
  $k^{t+2/3} \leftarrow \max\{k^{t+1/3}, 0\}$;
  $k^{t+1} \leftarrow \frac{k^{t+2/3}}{\|k^{t+2/3}\|_1}$;
  $\lambda \leftarrow \max\{0.99\lambda, \lambda_{min}\}$;
  $t \leftarrow t + 1$;
end
$u \leftarrow u^{t+1}$;
$k \leftarrow k^{t+1}$;

Algorithm 1: Blind Deconvolution Algorithm

$k \in \mathbf{R}^{h \times w}$, then we have $f \in \mathbf{R}^{(m-h+1)\times(n-w+1)}$. This choice introduces more variables to restore, but it does not need to make assumptions beyond the boundary of f. This results in a better reconstruction of u without ringing artifacts. From Theorem (3.2) we know that a big value for the parameter λ helps avoiding the no-blur solution, but in practice it also makes the estimated sharp image u too "cartooned". We found that iteratively reducing the value of λ as specified in Algorithm 1 helps getting closer to the true solution.

In the following paragraphs we highlight some other important features of Algorithm 1.

Boundary conditions. Typically, deblurring algorithms assume that the sharp image is smaller or equal in size to the blurry one. In this case, one has to make assumptions at the boundary in order to compute the convolution. For testing purposes we adapted our algorithm to support different boundary conditions by substituting the different discrete convolution operators described in the previous section with a convolution operator that gives in output an image with the same size of the input image. Commonly used boundary conditions in the literature are: *symmetric*, where the boundary of the image is mirrored to fill the additional frame around the image; *periodic*, where the image is padded with a periodic repetition of the boundary; *replicate*, where the borders continue with a constant value. We also used the *periodic* boundary condition after using the method proposed by Liu and Jia [15], that extends the size of the blurry image to make it periodic. In Fig. 5 we show a comparison on the dataset of [14] between our original approach and the adaptations with different boundary conditions. For each boundary condition we compute the cumulative histogram of the deconvolution error ratio across test examples, where the i-th bin counts the percentage of images for which the error ratio is smaller than i. The deconvolution error ratio, as defined in [14], measures the ratio between the SSD deconvolution error with the estimated and correct kernels. The implementations with the different boundary conditions perform worse than our free-boundary implementation, even if pre-processing the blurry image with the method of Liu and Jia [15] considerably improves the performance of the *periodic* boundary condition.

Filtered data fitting term. Recent algorithms estimate the blur by using filtered versions of u and f in the data fitting term (typically the gradients or the Laplacian). This choice might improve the estimation of the blur because it reduces the error at the boundary when using any of the previous approximations, but it might result also in a larger sensitivity to noise. In Fig. 6 we show how with the use of the filtered images for the blur estimation the performance of the *periodic* and *replicate* boundary conditions improves, while the others get slightly worse. Notice that our implementation still achieves better results than other boundary assumptions.

Pyramid scheme. While all the theory holds at the original input image size, to speed up the algorithm we also make use of a pyramid scheme, where we scale down the blurry image and the blur size until the latter is 3×3 pixels. We then launch our deblurring algorithm from the lowest scale, then upsample the results and use them as initialization for

the following scale. This procedure provides a significant speed up of the algorithm. On the smallest scale, we initialize our optimization from a uniform blur.

5. Experiments

We evaluated the performance of our algorithm on several images and compared with state-of-the-art algorithms. We provide our unoptimized Matlab implementation on our website[1]. Our blind deconvolution implementation processes images of 255×225 pixels with blurs of about 20×20 pixels in around 2-5 minutes, while our non-blind deconvolution algorithm takes about $10-30$ seconds.

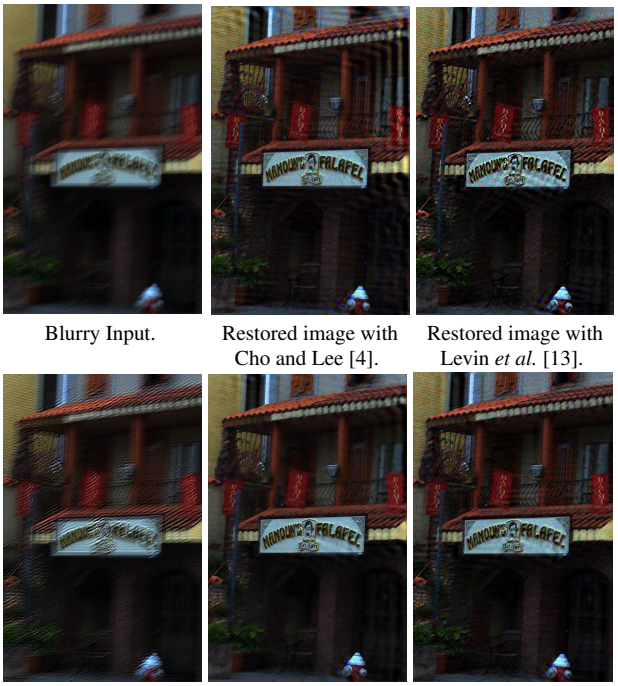
In our experiments we used the dataset from [14] in the same manner illustrated by the authors. For the whole test we used $\lambda_{min} = 0.0006$. We used the non-blind deconvolution algorithm from [12] with $\lambda = 0.0068$ and evaluated the ratios between the SSD deconvolution errors of the estimated and correct kernels. In Fig. 4 we show the cumulative histogram of the error ratio for Algorithm 1, the algorithm of Levin *et al.* [12], the algorithm of Fergus *et al.* [7] and the one of Cho and Lee [4]. Algorithm 1 performs on par with the one from Levin *et al.* [12], with a slightly higher number of restored images with small error ratio. In Fig. 7 we show some examples from dataset [14] and the relative errors.

In Fig. 8 we show a comparison between our method and the one proposed by Xu and Jia [23]. Their algorithm is able to restore sharp images even when the blur size is large by using an edge selection scheme that selects only large edges. This behavior is automatically minimicked by Algorithm 1 thanks to the TV prior. Also, in the presence of noise, Algorithm 1 performs visually on a par with the state-of-the-art algorithms as shown in Fig. 9.

6. Conclusions

In this paper we shed light on approaches to solve blind deconvolution. First, we confirmed that the problem formulation of total variation blind deconvolution as a maximum a priori in both sharp image and blur ($\text{MAP}_{u,k}$) is prone to local minima and, more importantly, does not favor the correct solution. Second, we also confirmed that the original implementation [2] of total variation blind deconvolution (PAM) can successfully achieve the desired solution. This discordance was clarified by dissecting PAM in its simplest steps. The analysis revealed that such algorithm does not minimize the original $\text{MAP}_{u,k}$ formulation. This analysis applies to a large number of methods solving $\text{MAP}_{u,k}$ as they might exploit the properties of PAM; moreover, it shows that there might be principled solutions to $\text{MAP}_{u,k}$. We believe that by further studying the behavior of the PAM algorithm one could arrive at novel useful formulations for blind deconvolution. Finally, we have showed that the PAM algorithm is neither worse nor better than the state of the art algorithms despite its simplicity.

[1] http://www.cvg.unibe.ch/dperrone/tvdb/



Blurry Input. | Restored image with Cho and Lee [4]. | Restored image with Levin *et al.* [13].

Restored image with Goldstein and Fattal [8]. | Restored image with Zhong *et al.* [27]. | Restored image with Algorithm 1.

Figure 9. Examples of blind-deconvolution restoration.

7. Acknowledgments

We would like to thank Martin Burger for his comments and suggestions.

References

[1] V. I. Bogachev. *Measure Theory I.* Berlin, Heidelberg, New York: Springer-Verlag, 2007.

[2] T. Chan and C.-K. Wong. Total variation blind deconvolution. *IEEE Transactions on Image Processing*, 1998.

[3] T. C. Chan and C. Wong. Convergence of the alternating minimization algorithm for blind deconvolution. *Linear Algebra and its Applications*, 316(13):259 – 285, 2000.

[4] S. Cho and S. Lee. Fast motion deblurring. *ACM Trans. Graph.*, 2009.

[5] L. Condat. A direct algorithm for 1-d total variation denoising. *Signal Processing Letters, IEEE*, 2013.

[6] P. L. Davies and A. Kovac. Local extremes, runs, strings and multiresolution. *The Annals of Statistics*, 2001.

[7] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graph.*, 2006.

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

[9] L. He, A. Marquina, and S. J. Osher. Blind deconvolution using tv regularization and bregman iteration. *Int. J. Imaging Systems and Technology*.

[10] J. Huang and D. Mumford. Statistics of natural images and models. pages 541 – 547, 1999.

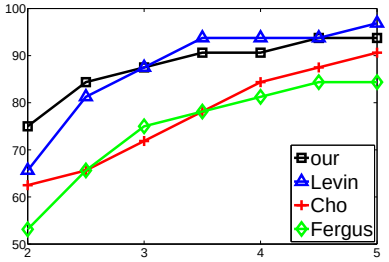


Figure 4. Comparison between Algorithm 1 and recent state-of-the-art algorithms.

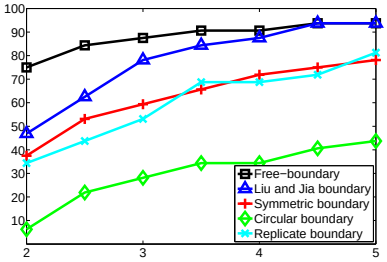


Figure 5. Comparison of Algorithm 1 with different boundary conditions.

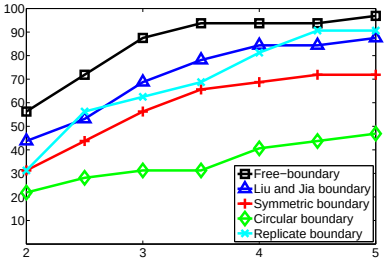
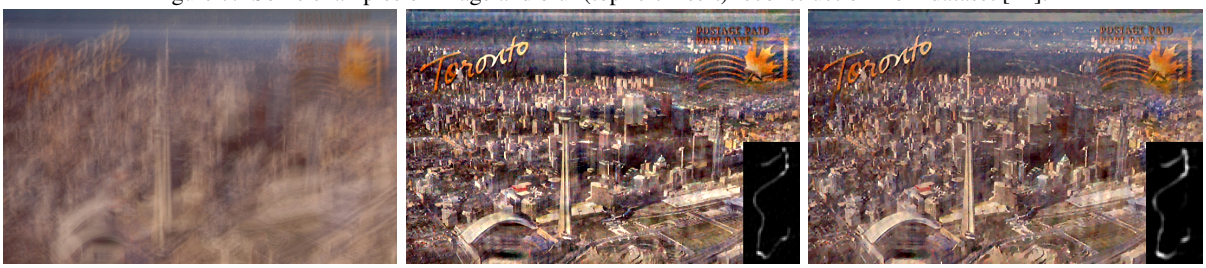


Figure 6. As in Fig. 5, but using a filtered version of the images for the blur estimation.

Our method err $= 2.04$ | Levin *et al.* [13] err $= 2.07$ | Cho *et al.* [4] err $= 4.01$ | Fergus *et al.* [7] err $= 10.4$ | Ground Truth err $= 1.00$

Figure 7. Some examples of image and blur (top-left insert) reconstruction from dataset [14].



Blurry Input. | Restored image and blur with Xu and Jia [23]. | Restored image and blur with Algorithm 1.

Figure 8. Example of blind-deconvolution image and blur (bottom-right insert) restoration.

[11] D. Kundur and D. Hatzinakos. Blind image deconvolution. *Signal Processing Magazine, IEEE*, 13(3):43–64, 1996.

[12] A. Levin, R. Fergus, F. Durand, and W. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.*, 26, 2007.

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

[14] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding blind deconvolution algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(12):2354–2367, 2011.

[15] R. Liu and J. Jia. Reducing boundary artifacts in image deconvolution. In *ICIP*, 2008.

[16] D. Perrone and P. Favaro. Total variation blind deconvolution: The devil is in the details. Technical Report - University of Bern, 2014.

[17] R. M. Rameshan, S. Chaudhuri, and R. Velmurugan. Joint map estimation for blind deconvolution: When does it work? ICVGIP, 2012.

[18] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physics D*, 60:259–268, 1992.

[19] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *ACM Trans. Graph.*, 2008.

[20] D. Strong and T. Chan. Edge-preserving and scale-dependent properties of total variation regularization. *Inverse Problems*, 19(6):S165, 2003.

[21] A. Tikhonov and V. Arsenin. *Solutions of ill-posed problems*. Vh Winston, 1977.

[22] C. Wang, L. Sun, Z. Chen, J. Zhang, and S. Yang. Multi-scale blind motion deblurring using local minimum. *Inverse Problems*, 26(1):015003, 2010.

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

[24] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *CVPR*, 2013.

[25] Y.-L. You and M. Kaveh. Anisotropic blind image restoration. In *ICIP 1996*, 1996.

[26] Y.-L. You and M. Kaveh. A regularization approach to joint blur identification and image restoration. *Image Processing, IEEE Transactions on*, 5(3):416–428, 1996.

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