Proximal Methods for Image Deblurring

Yair Daon

I describe a method of deblurring images using proximal operators. For lack of time, I implement only a part of it and show results.

1 Problem Description

I follow, somewhat loosely, a deblurring method presented in [1]. The goal is to recover a high resolution image $\theta \in \mathbb{R}^{N\times N}$ from a blurred and noisy observation $y \in \mathbb{R}^{N\times N}$. This obeserved image y is obtained from the original image θ by a noisy convolution as follows. Let $H : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ be the blurring operator. It acts as $H\theta = \theta * K$, where $K \in \mathbb{R}^{N\times N}$ is a convolution kernel and the convolution is taken with periodic boundary conditions. The blurred image is further corrupted by gaussian noise, which is $\mathcal{N}(0, \sigma^2)$, additive and independent for every pixel. Thus $y \sim \mathcal{N}(H\theta, \sigma^2 I)$, where $I : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ is the identity operator.

2 Hierarchical Model and ∇_d

This problem is ill posed and needs regularization. The regularization method comes from the following bayesian hierarchical model suggested by [2].

$$
\begin{aligned}
f(y|\theta) &= \frac{1}{(2\pi\sigma^2)^{\frac{N^2}{2}}} \exp(-\frac{1}{2\sigma^2}||H\theta - y||_2^2) \text{ (likelihood)}\\
\pi(\theta|\alpha) &= \frac{1}{Z(\alpha)} \exp(-\alpha||\nabla_d\theta||_2) \text{ (prior)}\\
\pi(\alpha) &= e^{-\alpha}\mathbf{1}_{\mathbb{R}_+}(\alpha) \text{ (hyperprior).}
\end{aligned}
$$

$\nabla_d\theta$ is the discrete gradient. [3] defines it slightly differently that below (I just use a periodic boundary). $(\nabla_d\theta)_{ij} = ((\nabla_d\theta)^1_{ij}, (\nabla_d\theta)^2_{ij})$ with

$$
\begin{aligned}
(\nabla_d\theta)^1_{ij} &= \theta_{i+1 \text{ mod } N,j} - \theta_{ij}\\
(\nabla_d\theta)^2_{ij} &= \theta_{i,j+1 \text{ mod } N} - \theta_{ij}.
\end{aligned}
$$

In the following discussion, I drop the $\mod$ term when referring to ∇_d. In [1], the norm on the gradient in the prior is taken to be "the $l_1 - l_2$ composite norm". Here I use the 2-norm for the gradient, which is the one used in [2, 3].

2.1 The posterior

We seek the MAP estimator of $p(\theta|y)$. First we calculate the posterior $p(\theta|y)$. The following discussion follows the one in [2, section 4.1] with the simplification that, in their notation, we take $\alpha = \beta = 1$. First, we need the marginal $\pi(\theta)$.

$$\begin{aligned}
\pi(\theta) &= \int_0^\infty \pi(\theta|\alpha)\pi(\alpha)d\alpha \\
&= \int_0^\infty \pi(\theta|\alpha)e^{-\alpha}d\alpha \\
&= \int_0^\infty \frac{1}{Z(\alpha)}\exp(-\alpha||\nabla_d\theta||_2)e^{-\alpha}d\alpha
\end{aligned}$$

Now we have to calculate the normalization constant. Explanations follow.

$$\begin{aligned}
Z(\alpha) &= \int_{\mathbb{R}^{N\times N}} \exp(-\alpha||\nabla_d\theta||_2)d\theta \\
&= \int_{\mathbb{R}^{N\times N}} \exp(-\alpha\sum_{i,j=1}^{N-1}\sqrt{(\theta_{i+1,j}-\theta_{ij})^2+(\theta_{i,j+1}-\theta_{ij})^2})d\theta \\
&\approx [\int_{\mathbb{R}^2} \exp(-\alpha\sqrt{u^2+v^2})dudv]^{N^2} \\
&= (\frac{2\pi}{\alpha^2})^{N^2} \\
&\approx C\alpha^{\eta N^2}.
\end{aligned}$$

A few notes: The first approximation and builds on an assumption from [2] that the graditents of different pixels are independent. In the last line, C is an irrelevant constant and $\eta = 2$. Since independence does not hold, [2] use different values of η for better performance ($\eta = 0.4$, specifically) and [1] uses $\eta = 1$. So $\eta = 1$ from now on. Now that we have the normalizing constant, we can conclude:

$$
\begin{aligned}
\pi(\theta) &\approx \frac{1}{C}\int_0^\infty \alpha^{N^2}\exp(-\alpha(||\nabla_d\theta||_2+1))d\alpha \\
&= \frac{1}{C(||\nabla_d\theta||_2+1)}\int_0^\infty \alpha^{N^2}(||\nabla_d\theta||_2+1)\exp(-\alpha(||\nabla_d\theta||_2+1))d\alpha \\
&= \frac{1}{C(||\nabla_d\theta||_2+1)}\cdot\frac{N^2!}{(||\nabla_d\theta||_1+1)^{N^2}} \\
&\propto (||\nabla_d\theta||_2+1)^{-(N^2+1)},
\end{aligned}
$$

which is equation (24) from [2] (with $\alpha = \beta = 1$ by their notation). Putting the pieces together, we see:

$$
\begin{aligned}
p(\theta|y) &\propto f(y|\theta)\pi(\theta) \\
&\propto \exp[-\frac{1}{2\sigma^2}||H\theta - y||_2^2 - (N^2+1)\log(||\nabla_d\theta||_2+1)]
\end{aligned}
$$

and so

$$
\theta_{\text{MAP}} = \arg\min_\theta \frac{1}{2\sigma^2}||H\theta - y||_2^2 + (N^2+1)\log(||\nabla_d\theta||_2+1),
$$

which is the maximization problem (21) from [1] except that we use the 2-norm and not the composite 1,2-norm and our N^2 is denoted there by n.

2.2 Convex Majorants

The above minimization problem is not convex - for once, log is concave. This is circumvented in [2] by taking a sequence of convex majorants. Consider the problem of finding $\hat{\theta} \in \arg\min_x L(\theta)$, for some L. Carrying out the majorization-minimization approach consists of finding a bound $Q(\theta;\theta') \geq L(\theta), \forall\theta,\theta'$ with equality for $\theta = \theta'$ and then iterating $\theta^{(t+1)} := \arg\min_\theta Q(\theta;\theta^{(t)})$. This iteration is monotone:

$$
\begin{aligned}
L(\theta^{(t+1)}) &= L(\theta^{(t+1)}) - Q(\theta^{(t+1)};\theta^{(t)}) + Q(\theta^{(t+1)};\theta^{(t)}) \\
&\leq Q(\theta^{(t+1)};\theta^{(t)}) \text{ by } Q \geq L \\
&\leq Q(\theta^{(t)};\theta^{(t)}) \text{ by definition of } \theta^{(t+1)} \\
&= L(\theta^{(t)}) \text{ by the equality condition above.}
\end{aligned}
$$

Here I follow this approach. Define

$$
L(\theta) := \frac{1}{2\sigma^2}||H\theta - y||_2^2 + (N^2+1)\log(||\nabla_d\theta||_2+1).
$$

Now we seek a majorant Q. Note that $\forall z, z_0 > 0$ the following inequality holds:

$$\log z \leq \log z_0 + \frac{z - z_0}{z_0},$$

with equality iff $z = z_0$.
Use this inequality with $z = ||\nabla_d \theta||_2 + 1, z_0 = ||\nabla_d \theta^{(t)}||_2 + 1$ to observe

$$\begin{aligned}
\log(||\nabla_d \theta||_2 + 1) &\leq \log(||\nabla_d \theta^{(t)}||_2 + 1) + \frac{||\nabla_d \theta||_2 + 1 - (||\nabla_d \theta^{(t)}||_2 + 1)}{||\nabla_d \theta^{(t)}||_2 + 1} \\
&= C(\theta^{(t)}) + \frac{||\nabla_d \theta||_2}{||\nabla_d \theta^{(t)}||_2 + 1}.
\end{aligned}$$

Denote $\alpha^{(t)} := (N^2 + 1)(||\nabla_d \theta^{(t)}||_2 + 1)^{-1}$. Then

$$\begin{aligned}
L(\theta) &= \frac{1}{2\sigma^2}||H\theta - y||_2^2 + (N^2 + 1)\log(||\nabla_d \theta||_2 + 1) \\
&\leq \frac{1}{2\sigma^2}||H\theta - y||_2^2 + \alpha^{(t)}||\nabla_d \theta||_2 + C(\theta^{(t)}) \\
&=: Q(\theta; \theta^{(t)}).
\end{aligned}$$

Thus, we find the next approximation by:

$$\begin{aligned}
\theta^{(t+1)} &:= \arg\min_{\theta} Q(\theta, \theta^{(t)}) \\
&= \arg\min_{\theta} \frac{1}{2\sigma^2}||H\theta - y||_2^2 + \alpha^{(t)}||\nabla_d \theta||_2
\end{aligned}$$

We omit with $C(\theta^{(t)})$ since it clearly does not change the minimizer. This is the problem that (should be) denoted by (22) in [1]. There, one of the authors made a typo and uses a different $\alpha^{(t)}$. I confirmed with the author that the above is correct.
Now that we have a convex optimization problem, we'd like to use Douglas Rachford. Define

$$\begin{aligned}
g(\theta) &:= \frac{1}{2\sigma^2}||H\theta - y||_2^2 \\
h(\theta) &:= \alpha^{(t)}||\nabla_d \theta||_2.
\end{aligned}$$

We need to find the proximal maps.

3 Calculating Prox$_h$

This section follows [3]. Let $J(\theta) := ||\nabla_d\theta||_2 = \sum_{i,j} |(\nabla_d \theta_{ij})|$. Note that $J(\lambda\theta) = \lambda J(\theta)$ for $\lambda \geq 0$ and also $J \geq 0$. If $\exists \theta_0$ s.t. $\langle \phi, \theta_0 \rangle - J(\theta_0) > 0$, then $\langle \lambda\theta_0, \phi \rangle - J(\lambda\theta_0) \to \infty$ as $\lambda \to \infty$. Thus we may easily conclude,

$$\begin{aligned} J^*(\phi) : &= \sup_\theta \langle \phi, \theta \rangle - J(\theta) \\ &= \sup_\theta \sum_{i,j=1}^N \phi_{ij}\theta_{ij} - J(\theta) \\ &= \begin{cases} 0 & \phi \in K \\ \infty & \phi \notin K. \end{cases} \end{aligned}$$

K is convex since J^* is. Since J is convex lsc, we observe that

$$\begin{aligned} J(\theta) &= J^{**}(\theta) \\ &= \sup_\phi \langle \phi, \theta \rangle - J^*(\phi) \\ &= \sup_{\phi \in K} \langle \phi, \theta \rangle \end{aligned}$$

By Cauchy Schwarz (and its equality condition)

$$\begin{aligned} J(\theta) &= \sum_{ij} |\nabla_d \theta| \\ &= \sup_{|p_{ij}| \leq 1} \sum_{ij} (\nabla_d\theta)^1_{ij} p^1_{ij} + (\nabla_d\theta)^2_{ij} p^2_{ij} \\ &= \sup_{|p_{ij}| \leq 1} \langle \nabla_d \theta, p \rangle \\ &= \sup_{|p_{ij}| \leq 1} \langle \theta, \nabla_d^* p \rangle \end{aligned}$$

with the obvious definition of an inner product. If we denote div $:= -\nabla_d^*$, the negative adjoint of the discrete gradient operator, then we may easily observe $K = \{\text{div} p : |p_{ij}| \leq 1 \ \forall 1 \leq i, j, \leq N\}$. I won't write the expression for div here but it is extremely simple because of the periodic boundary. We may now turn to deriving an algorithm for the proximity mapping.

$$
\begin{aligned}
p &= \mathrm{Prox}_{\lambda J}(\theta) \\
&= \arg\min_x \frac{1}{2}||x-\theta||^2 + \lambda J(x) \\
\Leftrightarrow 0 &\in \frac{p-\theta}{\lambda} + \partial J(p) \\
\Leftrightarrow \frac{\theta-p}{\lambda} &\in \partial J(p) \\
\Leftrightarrow p &\in \partial J^*(\frac{\theta-p}{\lambda}) \\
\Leftrightarrow 0 &\in \frac{\theta-p}{\lambda} - \frac{\theta}{\lambda} + \frac{1}{\lambda}\partial J^*(\frac{\theta-p}{\lambda}).
\end{aligned}
$$

Denote $w := \frac{\theta-p}{\lambda}$. We conclude that w minimizes $\frac{1}{2}||w - \frac{\theta}{\lambda}||^2 + \frac{1}{\lambda}J^*(w)$. Since J^* is the characteristic funcition of K, we deduce $w = P_K(\frac{\theta}{\lambda})$. Recalling $p = \mathrm{Prox}_{\lambda J}(\theta)$ and rearranging:

$$\mathrm{Prox}_{\lambda J}(\theta) = \theta - P_{\lambda K}(\theta).$$

Finding the projection amounts to finiding the minimizer

$$P_{\lambda K}(\theta) = \arg\min_{|p_{ij}|-1\leq 0} ||\lambda \mathrm{div} p - \theta||^2.$$

Recall that div $= -\nabla_d^*$, by definition and that it is merely a linear operator. The Karush Kuhn Tucker conditions yield the existence of a Lagrange multiplier μ_{ij} corresponding to every inequality constraint $|p_{ij}| - 1 \leq 0$. For these and for a minimum, it holds that $\forall i, j$:

$$
\begin{aligned}
-\nabla_d(\lambda \mathrm{div} p - \theta)_{ij} + \mu_{ij} p_{ij} &= 0 \\
|p_{ij}|^2 - 1 &\leq 0 \\
\mu_{ij} &\geq 0 \\
\mu_{ij}(|p_{ij}|^2 - 1) &= 0.
\end{aligned}
$$

Thus, if $\mu_{ij} = 0$ then also $-\nabla_d(\lambda \mathrm{div} p - \theta)_{ij} = 0$. If $\mu_{ij} > 0$ then $|p_{ij}| = 1$ and so $|\nabla_d(\lambda \mathrm{div} p - \theta)_{ij}| = \mu_{ij}$. Consequently,

$$|\nabla_d(\lambda \mathrm{div} p - \theta)_{ij}| = \mu_{ij}, \ \forall i, j.$$

Then a minimum will satisfy

$$\nabla_d(\mathrm{div} p - \frac{\theta}{\lambda})_{ij} = |\nabla_d(\mathrm{div} p - \frac{\theta}{\lambda})_{ij}| p_{ij}.$$

Let $\tau > 0$. The following iteration is reasonable at least because the minimum is a fixed point.

$$p_{ij}^{n+1} = p_{ij}^n + \tau[\nabla_d(\mathrm{div}p^n - \frac{\theta}{\lambda})_{ij} - |\nabla_d(\mathrm{div}p^n - \frac{\theta}{\lambda})_{ij}|p_{ij}^{n+1},$$

which is equivalent to:

$$p_{ij}^{n+1} = \frac{p_{ij}^n + \tau\nabla_d(\mathrm{div}p^n - \frac{\theta}{\lambda})_{ij}}{1 + \tau|\nabla_d(\mathrm{div}p^n - \frac{\theta}{\lambda})_{ij}|}.$$

Chambolle proves this converges for $0 < \tau \leq \frac{1}{8}$ and states that $\tau = \frac{1}{4}$ is optimal.

3.1 Results for Chambolle's algorithm

The proposed algorithm runs extremely fast. It takes merely a few seconds on my CIMS machine.

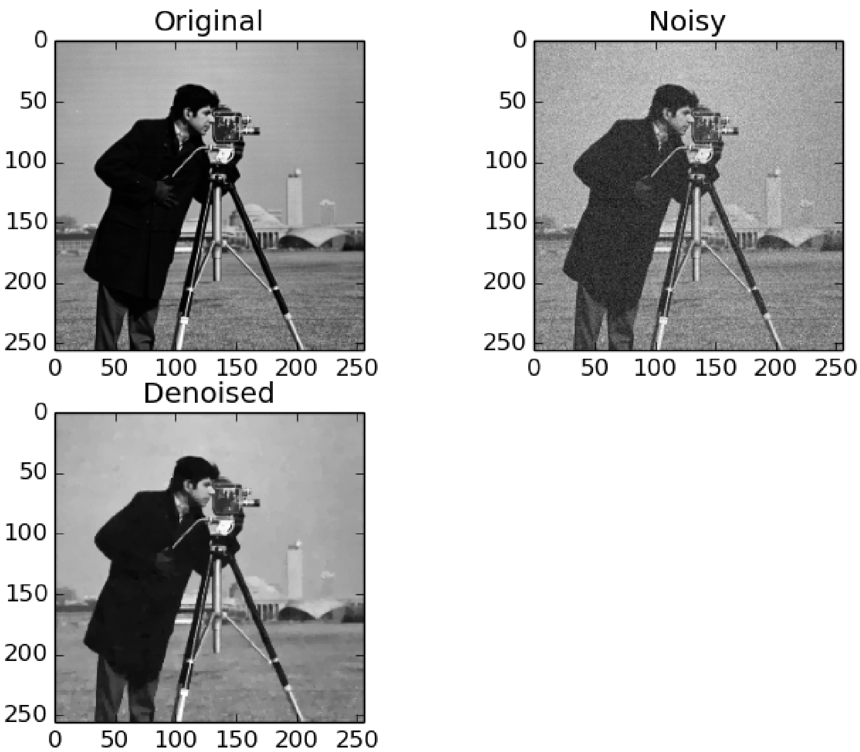

Figure 3.1: Performance of Chambolle's algorithm. Noise amplitude is $\sigma = 15$.

4 Calculating ∇g

Lets do a calculation. In this section we define, for our kernel $\bar{K}_{i-m,j-n} = K_{m-i,n-j}$, $\forall m, n$.

$$\begin{aligned}
\langle Hu, v\rangle &= \langle K * u, v\rangle \\
&= \sum_{ij}\sum_{mn} K_{i-m,j-n} u_{mn} v_{ij} \\
&= \sum_{mn} u_{mn} \sum_{ij} K_{i-m,j-n} v_{ij} \\
&= \sum_{ij} u_{ij} \sum_{mn} K_{m-i,n-j} v_{mn} \\
&= \sum_{ij} u_{ij} \sum_{mn} \bar{K}_{i-m,j-n} v_{mn} \\
&= \sum_{ij} u_{ij} (\bar{K} * v)_{ij} \\
&= \langle u, \bar{K} * v\rangle \\
&= \langle u, H^* v\rangle,
\end{aligned}$$

Specifically, if $m \equiv n \equiv 0 \mod N$ we have $\bar{K}_{i,j} = K_{N-i,N-j}$. Recalling that $K_{ij} := \frac{1}{(2m+1)^2}\mathbf{1}_{\{0\le i,j\le 2m\}}$, we arrive at

$$\begin{aligned}
\bar{K}_{ij} &= \frac{1}{(2m+1)^2}\mathbf{1}_{\{0\le N-i,N-j\le 2m\}} \\
&= \frac{1}{(2m+1)^2}\mathbf{1}_{\{-N\le -i,-j\le 2m-N\}} \\
&= \frac{1}{(2m+1)^2}\mathbf{1}_{\{N-2m\le i,j\le N\}}
\end{aligned}$$

We may concolude that the gradient ∇ (wrt to each pixel, not the discrete gradient) is:

$$\begin{aligned}
\nabla \frac{1}{2\sigma^2}||Hu-y||^2 &= \frac{1}{\sigma^2} H^*(Hu-y) \\
&= \frac{1}{\sigma^2}\bar{K} * (K * u - y) \\
&= \frac{1}{\sigma^2} K * (K * u - y),
\end{aligned}$$

which can be very easily implemented using FFT. If we want to use forward-backward, we must estimate the Lipschitz constant of the gradient:

$$
\begin{aligned}
||\nabla g(u) - \nabla g(v)|| &= ||\frac{1}{\sigma^2}H^*(Hu-y) - \frac{1}{\sigma^2}H^*(Hv-y)|| \\
&= \frac{1}{\sigma^2}||H^*H|| \cdot ||u-v|| \\
&\leq \frac{1}{\sigma^2}||H||^2||u-v|| \\
&= \frac{1}{\sigma^2}[\sup_{||w||=1}||Hw||]^2||u-v|| \\
&= \frac{1}{\sigma^2}[\sup_{||\hat{w}||=1}||\hat{K}\cdot\hat{w}||]^2||u-v|| \text{ (FT)} \\
&= \frac{1}{\sigma^2}[\sup_{||\hat{w}||=1}|\langle\hat{K},\hat{w}\rangle|]^2||u-v|| \\
&\leq \frac{1}{\sigma^2}||\hat{K}||^2||u-v|| \text{ (CS)} \\
&\leq \frac{1}{\sigma^2}||K||^2||u-v|| \\
&= \frac{1}{\sigma^2}\frac{1}{(2m+1)^4}(2m+1)^4||u-v|| \\
&= \frac{1}{\sigma^2}||u-v||.
\end{aligned}
$$

And so, in forward-backward terminology, $\beta = \sigma^2$. We will thus take $\gamma := \beta = \sigma^2$. We see that $\delta := \min\{1, \beta/\gamma\} + \frac{1}{2} = \frac{3}{2}$. Thus we may take $\lambda_n \equiv 1$.

5 Putting the pieces together

I used Forward Backward to find $\theta^{(t+1)}$ in every step. Here is the algorithm I implement. Recall that

$$
\begin{aligned}
g(\theta) &= \frac{1}{2\sigma^2}||H\theta - y||_2^2, \\
\gamma h(\theta) &= \sigma^2\alpha^{(t)}||\nabla_d\theta||_2, \\
\alpha^{(t)} &= (N^2+1)(||\nabla_d\theta^{(t)}||_2 + 1)^{-1}.
\end{aligned}
$$

I took, as noted above, $\gamma = \sigma^2, \lambda_n \equiv 1$ in the forward backward algorithm.

5.1 Final Results

In figure 5.1 below I show results for one forward backward cycle with α set to be the "true" α (based on the original image). Then, in figure 5.2, I show result for the entire framework (including the majorisation-minimization steps). The noise level is taken to be $sigma = 0.05$. The Blurring kernel is uniform 5×5.

Algorithm 1 MM and FwdBckwd

Set $\theta \leftarrow y$ (the corrupted image).
for $t = 1, 2, 3, ...$ **do**
  $\alpha = (N^2 + 1)(||\nabla_d \theta||_2 + 1)^{-1}$.
  for $n = 1, 2, 3, ...$ **do**
    $z \leftarrow x - \gamma \nabla g(x)$.
    $x \leftarrow \text{Prox}_{\gamma h}(z)$.
  $\theta \leftarrow x$.
return

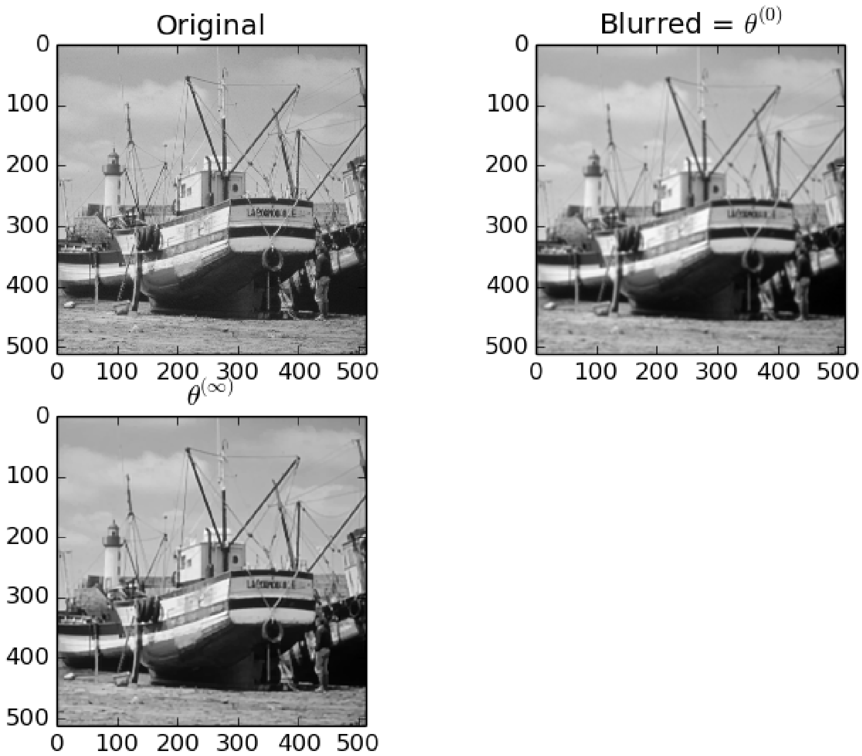

Figure 5.1: Performance of one F-B cycle with the true α.

References

[1] Peter J Green, Krzysztof Łatuszyński, Marcelo Pereyra, and Christian P Robert. Bayesian computation: a perspective on the current state, and sampling backwards and forwards. *arXiv preprint arXiv:1502.01148*, 2015.

[2] João P Oliveira, José M Bioucas-Dias, and Mário AT Figueiredo. Adaptive total vari-

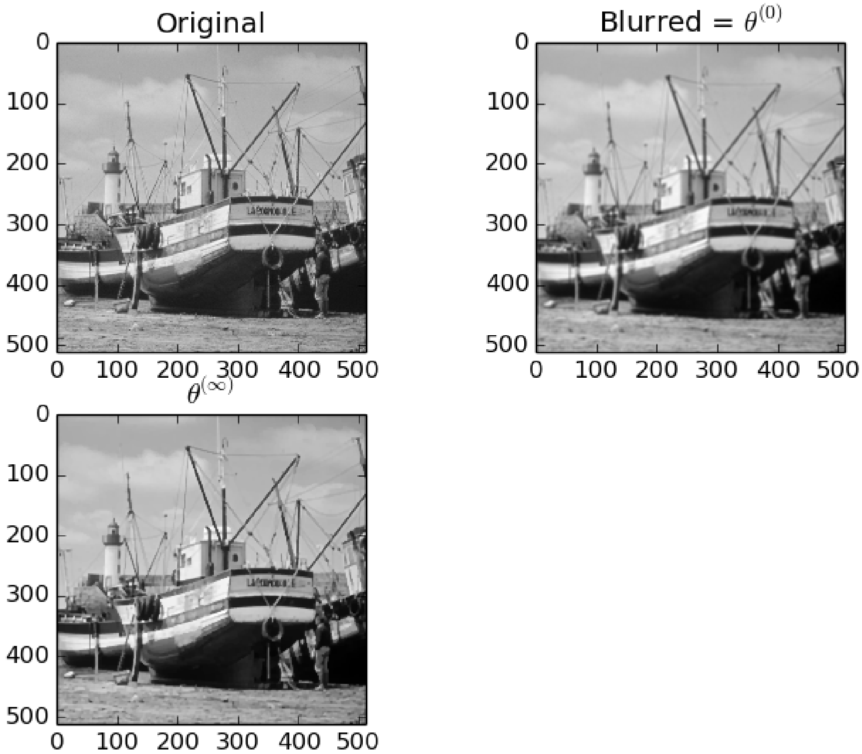

Figure 5.2: Performance of the entire framework.

ation image deblurring: a majorization–minimization approach. *Signal Processing*, 89(9):1683–1693, 2009.

[3] Antonin Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97, 2004.