Total Variation Deconvolution using Split Bregman

Pascal Getreuer

Yale University (pascal.getreuer@yale.edu)

Communicated by Jean-Michel Morel   *Demo edited by* Pascal Getreuer

Abstract

Deblurring is the inverse problem of restoring an image that has been blurred and possibly corrupted with noise. *Deconvolution* refers to the case where the blur to be removed is linear and shift-invariant so it may be expressed as a convolution of the image with a point spread function. Convolution corresponds in the Fourier domain to multiplication, and deconvolution is essentially Fourier division. The challenge is that since the multipliers are often small for high frequencies, direct division is unstable and plagued by noise present in the input image. Effective deconvolution requires a balance between frequency recovery and noise suppression.

Total variation (TV) regularization is a successful technique for achieving this balance in deblurring problems. It was introduced to image denoising by Rudin, Osher, and Fatemi [4] and then applied to deconvolution by Rudin and Osher [5]. In this article, we discuss TV-regularized deconvolution with Gaussian noise and its efficient solution using the split Bregman algorithm of Goldstein and Osher [16]. We show a straightforward extension for Laplace or Poisson noise and develop empirical estimates for the optimal value of the regularization parameter λ.

Source Code

ANSI C source code to produce the same results as the demo is accessible on the article web page https://doi.org/10.5201/ipol.2012.g-tvdc. Future software releases and updates will be posted at http://dev.ipol.im/~getreuer/code.

Keywords: image restoration, image deconvolution, total variation

1 Background

Suppose that an underlying image u is blurred by convolution with φ and corrupted with Gaussian noise η,

$$f = \varphi * u + \eta. \tag{1}$$

The deconvolution problem is, given f and φ, to estimate u. If the same imaging system that observed f is available to capture additional images, the blur kernel φ may be obtained by point spread estimation techniques, see for example the method by Delbracio, Musé, and Almansa [21].

If both the blur kernel φ and the underlying image u are unknown, the problem is called *blind deconvolution*. See for example Chan and Wong's TV-regularized blind deconvolution method [10] or the survey by Levin et al. [17]. *Linear deblurring* is the problem where the observed image f is related to the desired image u by

$$f(x) = (Ku)(x) + \eta(x) \tag{2}$$

where K is a linear operator. For example, K can describe blur which is spatially varying. *Nonlinear deblurring* refers to the case when K is a nonlinear operator. In this work, we restrict to the special case of *deconvolution*, where K is linear and shift-invariant.

1.1 Convolution with Boundary Handling

Before discussing the method itself, we address boundary handling. As deconvolution is the problem of inverting convolution, it is of central importance that convolution on a finite domain is clearly defined. In the continuous setting, convolution is defined as an integral over $\mathbb{R}^2$,

$$(\varphi * u)(x) := \int_{\mathbb{R}^2} \varphi(x - t)u(t)\, dt, \quad x \in \mathbb{R}^2, \tag{3}$$

and we assume that φ is specified on the whole domain $\mathbb{R}^2$. The image u, however, is in practice usually only available on a bounded rectangle. To deal with this problem, u can be extrapolated to $\mathbb{R}^2$ by a boundary extension operator E. Given u defined on an open set Ω, Eu is a function defined on $\mathbb{R}^2$ such that $Eu|_\Omega = u$. Convolution over Ω is then defined as $\varphi * Eu$.

Remark 1.1. *Finite-domain convolution is often defined with periodic extension ("convolution over the torus $\mathbb{T}^2$"), however, it is undesirable here since images are usually not periodic.*

Remark 1.2. *Another way to deal with the boundary handling problem is to normalize the convolution near the boundary*

$$(Ku)(x) := \int_\Omega k(x,t)u(t)\, dt, \quad k(x,t) = \frac{\varphi(x-t)}{\int_\Omega \varphi(x-s)\, ds}, \quad x \in \Omega. \tag{4}$$

This blur operator no longer describes a convolution. The denominator is always nonzero provided φ is nonnegative and if the integral of φ over any neighborhood of $(0,0)$ is positive.

For numerical implementation, convolution must be discretized. Let Z be a linear interpolation operator,

$$Z(u)(x) = \sum_{m,n \in \mathbb{Z}} u_{m,n} k(x_1 - m, x_2 - n), \quad x \in \mathbb{R}^2, \tag{5}$$

where $k(x)$ is the interpolation kernel, and define

$$\varphi_{i,j} := (\varphi * k)(i,j) = \int_{\mathbb{R}^2} \varphi(i - x_1, j - x_2) k(x)\, dx. \tag{6}$$

Interpreting u as an impulse train, $Z(u)$ can be written as $k * u$. By associativity of convolution,

$$\begin{aligned}
(\varphi * u)_{i,j} &:= \big(\varphi * Z(u)\big)(i,j) \\
&= \big(\varphi * (k * u)\big)(i,j) \\
&= \big((\varphi * k) * u\big)(i,j) \\
&= \sum_{m,n \in \mathbb{Z}} \varphi_{i-m,j-n} u_{m,n}.
\end{aligned} \tag{7}$$

The last line is the usual discrete convolution of $\varphi_{i,j}$ and $u_{i,j}$. For simplicity, we use nearest-neighbor interpolation for Z, where $\varphi_{i,j}$ is

$$\varphi_{i,j} = \int_{i-1/2}^{i+1/2} \int_{j-1/2}^{j+1/2} \varphi(x_1, x_2)\, dx_2\, dx_1. \tag{8}$$

We focus on the practical case where u is specified on a rectangle $u_{i,j}$, $i = 0, \ldots, N-1$, $j = 0, \ldots, N-1$, which may be viewed continuously as having a square domain $\Omega = [-\frac{1}{2}, N - \frac{1}{2}] \times [-\frac{1}{2}, N - \frac{1}{2}]$. We define E as the (half-sample) symmetric extension operator

$$Eu(x) := \begin{cases} u(x) & \text{if } x \in \Omega, \\ Eu(-1 - x_1, x_2) & \text{if } x_1 \leq -\frac{1}{2}, \\ Eu(2N - 1 - x_1, x_2) & \text{if } x_1 \geq N - \frac{1}{2}, \\ Eu(x_1, -1 - x_2) & \text{if } x_2 \leq -\frac{1}{2}, \\ Eu(x_1, 2N - 1 - x_2) & \text{if } x_2 \geq N - \frac{1}{2}. \end{cases} \tag{9}$$

The definition is recursive to account for the possibility that more than one reflection is needed, for example $Eu(2N, x_2) = Eu(-1, x_2) = u(0, x_2)$. The extended image is periodic with period $2N \times 2N$.

Due to the $2N \times 2N$ periodicity of Eu, this definition of discrete convolution can be stated in terms of the discrete Fourier transform (DFT), which is useful for computation. To eliminate wrap-around artifacts, we require that the kernel φ is compactly supported (or decays rapidly) within a rectangle no larger than $2N \times 2N$. Let Λ denote the $2N \times 2N$ grid $\{-N, \ldots, N-1\} \times \{-N, \ldots, N-1\}$ and let $\mathcal{F}$ denote the DFT on Λ. Then discrete convolution with symmetric boundaries is implemented through DFT transforms as

$$\varphi * u = \mathcal{F}^{-1}\big(\mathcal{F}(\varphi) \cdot \mathcal{F}(Eu)\big), \tag{10}$$

where $\cdot$ denotes pointwise multiplication. In other words, the procedure for DFT-based convolution is to construct Eu on Λ, compute its DFT, multiply with $\mathcal{F}(\varphi)$, and compute the inverse DFT. Since the spatial data is real, a real-to-complex FFT algorithm may be used to reduce the computational and memory costs by about a factor 2 compared to complex-to-complex FFTs.

Remark 1.3. *If Z is selected to be Fourier interpolation, then the DFT-based convolution is*

$$\varphi * u = \mathcal{F}^{-1}\big(\hat{\varphi} \cdot \mathcal{F}(Eu)\big), \tag{11}$$

where $\hat{\varphi}$ is the continuous-domain Fourier transform of φ.

1.2   DCT-based Convolution

If φ is even in both coordinates, $\varphi(x_1, x_2) = \varphi(\pm x_1, \pm x_2)$, then the discrete cosine transform (DCT) domain may be used instead of the DFT for greater computational efficiency. The data does not need to be padded in this case because symmetric boundaries are implied by the transforms, reducing the computational and memory costs by about a factor 4 compared to real-to-complex FFTs. Martucci [7] showed that convolution with (half-sample) symmetric boundary handling can be implemented through DCT transforms as

$$\varphi * u = \mathcal{C}_{2e}^{-1}\big(\mathcal{C}_{1e}(\varphi) \cdot \mathcal{C}_{2e}(u)\big), \tag{12}$$

where $\mathcal{C}_{1e}$ and $\mathcal{C}_{2e}$ are the DCT-I and DCT-II transforms of the same period lengths.

We describe DCT-based convolution in detail in one dimension. Convolution in higher dimensions is similar by tensor product. Let $x_0, \ldots, x_{N-1}$ be an input sequence with half-sample symmetric boundaries and let $h_N, \ldots, h_1, h_0, h_1, \ldots, h_N$ be a whole-sample symmetric filter. The convolution of x and h is defined as the length-$2N$ cyclic convolution of $x_0, \ldots, x_{N-1}, x_{N-1}, \ldots, x_0$ and $h_0, \ldots, h_N, h_{N-1}, \ldots, h_1$,

$$y_m = (h \overset{\text{sym}}{*} x)_m = \sum_{n=0}^{2N-1} h_n \, x_{m-n}, \tag{13}$$

where indices are modulo $2N$. This convolution is performed equivalently with DCT transforms as

$$y = \mathcal{C}_{2\text{e}}^{-1}\big(\mathcal{C}_{1\text{e}}(h) \cdot \mathcal{C}_{2\text{e}}(x)\big). \tag{14}$$

This formula can be implemented by the following steps:

1. Compute the DCT-I transform $\mathcal{C}_{1\text{e}}(h)$

$$H_k = h_0 + (-1)^k h_N + 2 \sum_{n=1}^{N-1} h_n \cos(\pi n k / N), \quad k = 0, \ldots, N$$

2. Compute the DCT-II transform $\mathcal{C}_{2\text{e}}(x)$

$$X_k = 2 \sum_{n=0}^{N-1} x_n \cos\big(\pi(n + \tfrac{1}{2})k/N\big), \quad k = 0, \ldots, N-1$$

3. Multiply the first N coefficients pointwise
   $Y_k = H_k \cdot X_k, \quad k = 0, \ldots, N-1$

4. Obtain the convolution in the spatial domain with $\mathcal{C}_{2\text{e}}^{-1}(Y)$

$$y_m = \mathcal{C}_{2\text{e}}^{-1}(Y) = \frac{1}{2N}\left(Y_0 + 2\sum_{k=1}^{N-1} Y_k \cos\big(\pi k(n + \tfrac{1}{2})/N\big)\right), \quad m = 0, \ldots, N-1$$

Note that although the period lengths are the same, the DCT-I transform is one sample longer ($k = 0, \ldots, N$) than the other two transforms ($k = 0, \ldots, N-1$) due to its whole-sample boundary symmetry. The transform definitions above are not orthogonal. Some authors define the DCT transforms with additional factors on certain coefficients to make them orthogonal.

These transforms can be performed efficiently with the FFTW library [13] using the real-to-real (r2r) interface. The codes `REDFT00` and `REDFT01` correspond respectively to the nonorthogonal DCT-I and DCT-II transforms as defined above. The inverse DCT-II transform is obtained using `REDFT10` and then dividing every element by $2N$.

2 Wiener Deconvolution

Wiener deconvolution [1] is a classical linear technique for deconvolution in the presence of noise. We review Wiener deconvolution in the discrete setting on a finite grid.

The observed image f is related to the desired $N \times N$ image u by

$$f = \varphi * u + \eta, \tag{15}$$

where φ is a known *point spread function*, η is additive noise, and $*$ denotes discrete convolution. It is assumed that u and η are uncorrelated. For boundary handling, u and η are symmetrically extended to size $2N \times 2N$ and periodized as described in the previous sections. The convolution becomes pointwise multiplication in the DFT domain of size $2N \times 2N$,

$$\mathcal{F}(Ef) = \mathcal{F}(\varphi) \cdot \mathcal{F}(Eu) + \mathcal{F}(E\eta). \tag{16}$$

We denote these quantities with ˆ as

$$\hat{f} = \hat{\varphi} \cdot \hat{u} + \hat{\eta}. \tag{17}$$

The Wiener strategy is to find a (noncausal and real-valued) linear filter w such that $w * f$ is an optimal estimate of u. The filter is selected to minimize the expected mean square error,

$$\underset{w}{\arg\min} \, \mathbb{E}\|w * f - u\|_2^2, \tag{18}$$

where $\mathbb{E}$ denotes expectation. Parseval's identity allows the objective function to be expressed in the DFT domain, where it decouples over frequency. So we may equivalently minimize in the DFT domain over $\hat{w} := \mathcal{F}(w)$,

$$\underset{\hat{w}}{\arg\min} \, \mathbb{E}\|\hat{w} \cdot \hat{f} - \hat{u}\|_2^2. \tag{19}$$

Differentiating by $\hat{w}$ and setting to zero yields the optimal filter,

$$2\mathbb{E}\big[(\hat{w} \cdot \hat{f} - \hat{u}) \cdot \hat{f}^*\big] = 0$$

$$\Rightarrow \quad \hat{w} = \frac{\mathbb{E}[\hat{u} \cdot \hat{f}^*]}{\mathbb{E}\big[|\hat{f}|^2\big]}, \tag{20}$$

where $(\cdot)^*$ denotes complex conjugation and the division is pointwise. The denominator $\mathbb{E}\big[|\hat{f}|^2\big]$ is the expected spectral density of f. The Wiener estimate of u is then obtained by pointwise multiplication,

$$\hat{u} \approx \hat{w} \cdot \hat{f}. \tag{21}$$

To write the filter in terms of φ, u, and η, we now substitute the degradation model $f = \varphi * u + \eta$:

$$\begin{aligned}
\hat{w} &= \frac{\mathbb{E}\big[\hat{u} \cdot (\hat{\varphi} \cdot \hat{u} + \hat{\eta})^*\big]}{\mathbb{E}\big[|\hat{\varphi} \cdot \hat{u} + \hat{\eta}|^2\big]} \\
&= \frac{\hat{\varphi}^* \cdot \mathbb{E}\big[|\hat{u}|^2\big] + \mathbb{E}[\hat{u} \cdot \hat{\eta}^*]}{|\hat{\varphi}|^2 \cdot \mathbb{E}\big[|\hat{u}|^2\big] + 2\mathrm{Re}\{\hat{\varphi}\} \cdot \mathbb{E}[\hat{u} \cdot \hat{\eta}^*] + \mathbb{E}\big[|\hat{\eta}|^2\big]} \\
&= \frac{\hat{\varphi}^* \cdot \mathbb{E}\big[|\hat{u}|^2\big]}{|\hat{\varphi}|^2 \cdot \mathbb{E}\big[|\hat{u}|^2\big] + \mathbb{E}\big[|\hat{\eta}|^2\big]}.
\end{aligned} \tag{22}$$

The $\mathbb{E}[\hat{u} \cdot \hat{\eta}^*]$ terms are zero by the assumption that u and η are uncorrelated. The optimal filter can be expressed in terms of φ and the noise-to-signal ratio (NSR),

$$\hat{w} = \frac{\hat{\varphi}^*}{|\hat{\varphi}|^2 + NSR}, \quad NSR := \frac{\mathbb{E}\big[|\hat{\eta}|^2\big]}{\mathbb{E}\big[|\hat{u}|^2\big]}. \tag{23}$$

Since φ, u, and η are real-valued, the above expression is conjugate symmetric and thus w is real-valued. For Gaussian noise of standard deviation σ, the numerator of the NSR is σ^2 for all frequencies. Ideally, u's expected spectral density $\mathbb{E}\big[|\hat{u}|^2\big]$ in the denominator of the NSR is available as well; we apply Wiener deconvolution this way in the examples to show the best possible linear result.

More realistically, $\mathbb{E}\big[|\hat{u}|^2\big]$ must be estimated. A simple estimate is

$$\begin{aligned}
\mathbb{E}\big[|\hat{f}|^2\big] &= |\hat{\phi}|^2 \cdot \mathbb{E}\big[|\hat{u}|^2\big] + |\hat{\eta}|^2 \\
\Rightarrow \quad \mathbb{E}\big[|\hat{u}|^2\big] &\approx |\hat{f}|^2 - \sigma^2.
\end{aligned} \tag{24}$$

A more sophisticated approach is the *iterative Wiener filter*: Wiener deconvolution produces an estimate of u, from which an updated expected spectral density is computed, and Wiener deconvolution is repeated. Hillery and Chin [11] analyze the iterative Wiener filter and propose a modification that ensures convergence.

3   TV Deconvolution

Given a blurred image with additive Gaussian noise, $f = Ku + \eta$, Rudin and Osher [5] proposed the following minimization problem to estimate the deblurred image u,

$$\arg\min_{u \in BV(\Omega)} \|u\|_{\mathrm{TV}(\Omega)} + \frac{\lambda}{2} \int_{\Omega} \big(f(x) - Ku(x)\big)^2 \, dx, \tag{25}$$

where λ is a positive parameter and K is a linear operator representing the blur, for example the convolution with boundary extension $Ku = \varphi * Eu$ or the spatially varying blur $Ku = \int k(x,t)u(t) \, dt$ operators described in the section on Convolution with Boundary Handling. The minimization problem is identical to the Rudin, Osher, and Fatemi denoising problem [4] except that here the second term penalizes $(f - Ku)^2$, while in the denoising case the second term penalizes $(f - u)^2$. TV-regularized denoising may be seen as a special case where K is the identity operator.

The solution u is found as the best function in the space of bounded variation (BV) images. A function u is in $BV(\Omega)$ if there exists a Radon measure Du such that

$$\int_{\Omega} u(x) \operatorname{div} \vec{g}(x) \, dx = -\int_{\Omega} \langle \vec{g}, Du(x) \rangle \quad \text{for all } \vec{g} \in C_c^1(\Omega, \mathbb{R}^2), \tag{26}$$

and the *total variation seminorm* of u is

$$\|u\|_{\mathrm{TV}(\Omega)} := \int_{\Omega} |Du| := \sup\left\{ \int_{\Omega} u \operatorname{div} \vec{g} \, dx : \vec{g} \in C_c^1(\Omega, \mathbb{R}^2), \sqrt{g_1^2 + g_2^2} \leq 1 \right\}. \tag{27}$$

When u is smooth, TV is equivalently the integral of its gradient magnitude,

$$\|u\|_{\mathrm{TV}(\Omega)} = \int_{\Omega} |\nabla u| \, dx. \tag{28}$$

Existence and uniqueness of the minimizer can be established under some conditions:

1. Ω is a bounded Lipschitz domain in $\mathbb{R}^2$.

2. The true image u is in $BV(\Omega)$.

3. The noisy image f is in $L^2(\Omega)$.

4. The blurring is a linear operator $K : L^1(\Omega) \to L^2(\Omega)$ that is bounded and injective, and K applied to a constant function reproduces the same constant, $K1 = 1$.

Provided that all of the above hold, the minimizer u of the TV-regularized deblurring problem exists and is unique [6, 8, 12]. A corollary of this result is that the mean of Ku must equal the mean of f.

There are many algorithms for solving the TV-regularized deconvolution minimization problem. Most recent algorithms use operator splitting [14, 15, 16, 18, 20]. In the next section, we discuss TV-regularized deconvolution with the split Bregman algorithm.

4   Split Bregman

We focus on the split Bregman algorithm of Goldstein and Osher [16] applied to deconvolution as developed in tvreg [19]. Due to the similarity between the TV denoising and deconvolution problems, the split Bregman algorithm for denoising [22] is readily adapted to deconvolution.

Our approach to discrete derivatives is to use one-sided differences and half-sample symmetric extension at points near the boundaries. In one dimension, we define the discrete derivative ∂ of a uniformly-sampled signal $f_0, f_1, \ldots, f_{N-1}$ as its forward difference,

$$\begin{pmatrix} \partial f_0 \\ \partial f_1 \\ \vdots \\ \partial f_{N-2} \\ \partial f_{N-1} \end{pmatrix} = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ & & & & 0 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_{N-2} \\ f_{N-1} \end{pmatrix}. \tag{29}$$

Since the half-sample symmetric extension is $2N$-periodic, the discrete gradient may also be viewed as a cyclic convolution of the reflected signal $(f_0, \ldots, f_{N-1}, f_{N-1}, \ldots, f_0)$ with the filter $h_{-1} = 1$, $h_0 = -1$, and h zero otherwise.

In two dimensions, we define the discrete gradient of an $N \times N$ image u using the discrete derivative as $\nabla u = (\partial_x u, \partial_y u)^T$, where the subscript on ∂ denotes the dimension along which the difference is applied. We define discrete divergence through the relationship $\text{div} := -\nabla^* = -\partial_x^* - \partial_y^*$, where the * superscript denotes operator adjoint, and the discrete Laplacian as $\Delta := \text{div}\,\nabla$. In the image interior, the discrete divergence is $\text{div}\, v = v_{i,j}^x - v_{i-1,j}^x + v_{i,j}^y - v_{i,j-1}^y$ and the discrete Laplacian is the 5-point formula $\Delta u = -4u_{i,j} + u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}$. Please refer to the section on discrete derivatives in the TV denoising article [22] for details.

Total variation is approximated by summing the vector magnitude $|\nabla u_{i,j}|$ over all pixels,

$$\|u\|_{\text{TV}(\Omega)} \approx \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |\nabla u_{i,j}|, \tag{30}$$

where ∇u is the discrete gradient developed above. Other discretizations of the TV seminorm are discussed in the TV denoising article [22]. The split Bregman algorithm is applied to solve the constrained minimization problem

$$\underset{d,u}{\arg\min} \sum_{i,j} |d_{i,j}| + \frac{\lambda}{2} \sum_{i,j} (f_{i,j} - Ku_{i,j})^2 \tag{31}$$

$$\text{subject to } d = \nabla u.$$

K denotes the discrete blurring operator, which is $Ku := \varphi * Eu$ or more generally is a linear operator on u. The auxiliary variable d is a vector field which is constrained to equal ∇u.

The Bregman iteration [2] method is applied to solve the constrained problem. In each iteration of the Bregman method, the following problem is solved:

$$\underset{d,u}{\arg\min} \sum_{i,j} |d_{i,j}| + \frac{\lambda}{2} \sum_{i,j} (f_{i,j} - Ku_{i,j})^2 + \frac{\gamma}{2} \sum_{i,j} |d_{i,j} - \nabla u_{i,j} - b_{i,j}|^2 \tag{32}$$

where b is a variable related to the Bregman iteration algorithm. The joint minimization over d and u is solved by an alternating direction method, in each step minimizing either d or u while keeping the other variable fixed.

d **subproblem**   With u fixed, the d subproblem is

$$\underset{d}{\arg\min} \sum_{i,j} |d_{i,j}| + \frac{\gamma}{2} \sum_{i,j} |d_{i,j} - \nabla u_{i,j} - b_{i,j}|^2. \tag{33}$$

This problem is solved as for TV denoising [22]:

$$d_{i,j} = \frac{\nabla u_{i,j} + b_{i,j}}{|\nabla u_{i,j} + b_{i,j}|} \max\{|\nabla u_{i,j} + b_{i,j}| - 1/\gamma, 0\}. \tag{34}$$

u **subproblem**   With d fixed, the u subproblem is

$$\arg\min_u \frac{\lambda}{2}\sum_{i,j}(f_{i,j} - Ku_{i,j})^2 + \frac{\gamma}{2}\sum_{i,j}|d_{i,j} - \nabla u_{i,j} - b_{i,j}|^2. \tag{35}$$

The optimal u satisfies the linear equation

$$\tfrac{\lambda}{\gamma}K^*Ku - \Delta u = \tfrac{\lambda}{\gamma}K^*f - \operatorname{div}(d - b), \tag{36}$$

where K^* denotes the adjoint of K. If K is a convolution with symmetric boundary handling, $Ku := \varphi * Eu$, then the problem may be efficiently solved in the discrete Fourier domain:

$$u = \mathcal{F}^{-1}\left[\frac{\tfrac{\lambda}{\gamma}\overline{\mathcal{F}(\varphi)}\cdot\mathcal{F}(Ef) - \mathcal{F}\big(E\operatorname{div}(d-b)\big)}{\tfrac{\lambda}{\gamma}|\mathcal{F}(\varphi)|^2 - \mathcal{F}(\Delta)}\right], \tag{37}$$

where E denotes symmetric extension to the $2N \times 2N$ grid $\{-N,\dots,N-1\}\times\{-N,\dots,N-1\}$ and $\mathcal{F}(\Delta)$ is the DFT of the 5-point Laplacian filter.

If φ is even in both coordinates, $\varphi(x_1, x_2) = \varphi(\pm x_1, \pm x_2)$, then the DCT domain may be used instead to reduce the computational cost. The u variable subproblem is solved as

$$u = \mathcal{C}_{2\mathrm{e}}^{-1}\left[\frac{\mathcal{C}_{2\mathrm{e}}\big(\tfrac{\lambda}{\gamma}\varphi * f - \operatorname{div}(d-b)\big)}{\mathcal{C}_{1\mathrm{e}}(\tfrac{\lambda}{\gamma}\varphi * \varphi - \Delta)}\right]. \tag{38}$$

where $\mathcal{C}_{1\mathrm{e}}$ and $\mathcal{C}_{2\mathrm{e}}$ are the DCT-I and DCT-II transforms of the same period lengths.

Enforcing the constraint   The auxiliary variable b is initialized to zero. To enforce the constraint $d = \nabla u$, the Bregman iteration algorithm updates b according to

$$b^{k+1} = b^k + \nabla u - d. \tag{39}$$

Bregman iteration ensures that the limiting solution satisfies the constraint exactly for any positive value γ. For good behavior in both subproblems, γ should be neither extremely large nor small [22]: the solution in the d subproblem is the vector shrinkage of $(\nabla u + b)$ by $1/\gamma$, which has greater effect for smaller γ, but in the u subproblem, the solution u satisfies

$$\lambda u - \gamma\Delta u = \lambda f - \gamma\operatorname{div}(d - b), \tag{40}$$

where spatial interaction is stronger for larger γ. Therefore, γ should be an intermediate value so that convergence is efficient in both subproblems. We fix $\gamma = 5$.

The overall algorithm is

Initialize $u = 0$, $d = 0$, $b = 0$
while $\|u_{cur} - u_{prev}\|_2 > Tol$ **do**
  Solve the d subproblem
  Solve the u subproblem
  $b = b + \nabla u - d$

Algorithm 1

where $Tol = \|f\|_2/1000$. To prevent excessive runtimes, we impose a maximum of 140 iterations in the implementation and online demo. For color images, the vectorial TV (VTV) [9] is used in place of TV,

$$\|u\|_{\mathrm{VTV}(\Omega)} := \int_\Omega\Big(\sum_{i\in\text{channels}}|\nabla u_i(x)|^2\Big)^{1/2}dx. \tag{41}$$

The grayscale algorithm extends directly to VTV-regularized deconvolution. In the color examples below, the blur operator K is applied separately to the red, green, and blue channels. It is possible with minor changes to the approach described here to deconvolve a color image where K has interchannel mixing, e.g.,

$$Ku = \begin{pmatrix} K_{RR} * u_R + K_{RG} * u_G + K_{RB} * u_B \\ K_{GR} * u_R + K_{GG} * u_G + K_{GB} * u_B \\ K_{BR} * u_R + K_{BG} * u_G + K_{BB} * u_B \end{pmatrix}. \tag{42}$$

5   TV Deconvolution with Laplace or Poisson Noise

Interpreted as a *maximum a posteriori* estimate, TV deconvolution can be adapted to other kinds of noise [19]. The minimization problem for general noise is

$$\underset{u \in BV(\Omega)}{\arg\min} \|u\|_{\mathrm{TV}(\Omega)} + \lambda \int_\Omega F\big(Ku(x), f(x)\big)\, dx, \tag{43}$$

where F specifies the noise model,

$$F\big(u(x), f(x)\big) = \begin{cases} |Ku(x) - f(x)| & \text{Laplace noise,} \\ Ku(x) - f(x)\log Ku(x) & \text{Poisson noise.} \end{cases} \tag{44}$$

A splitting with two auxiliary variables can be used to separate F from the derivative terms,

$$\underset{d,z,u}{\arg\min} \sum_{i,j}|d_{i,j}| + \lambda \sum_{i,j} F(z_{i,j}, f_{i,j}) \tag{45}$$
$$\text{subject to } d = \nabla u,\ z = Ku.$$

This is the same minimization used for denoising by the general-noise algorithm in the TV denoising article [22] except that $z = Ku$ is used here instead of $z = u$. Only minor changes to this method are needed to solve it. Bregman iteration is applied similarly, where in each iteration the problem

$$\underset{d,z,u}{\arg\min} \sum_{i,j}|d_{i,j}| + \lambda \sum_{i,j} F(z_{i,j}, f_{i,j})$$
$$+ \frac{\gamma_1}{2} \sum_{i,j}|d_{i,j} - \nabla u_{i,j} - b_{i,j}^1|^2\, dx + \frac{\gamma_2}{2} \sum_{i,j}(z_{i,j} - Ku_{i,j} - b_{i,j}^2)^2 \tag{46}$$

is solved. One variable d, z, or u is minimized at a time while keeping the other two fixed.

d **subproblem**   The d subproblem is solved as for TV denoising [22]:

$$d_{i,j} = \frac{\nabla u_{i,j} + b_{i,j}^1}{|\nabla u_{i,j} + b_{i,j}^1|} \max\big\{|\nabla u_{i,j} + b_{i,j}^1| - 1/\gamma_1, 0\big\}. \tag{47}$$

z **subproblem**   With d and u fixed, the z subproblem is

$$\underset{z}{\arg\min} \lambda \sum_{i,j} F(z_{i,j}, f_{i,j}) + \frac{\gamma_2}{2} \sum_{i,j}(z_{i,j} - Ku_{i,j} - b_{i,j}^2)^2. \tag{48}$$

The z subproblem is solved as for TV denoising [22] except that Ku replaces occurrences of u. The solution decouples over i, j and the optimal z satisfies

$$\lambda \partial_z F(z, f) + \gamma_2(z - Ku - b_2) = 0. \tag{49}$$

For the Laplace noise model with $F(z, f) = |z - f|$, the solution is

$$z_{i,j} = f_{i,j} + \text{sign } s_{i,j} \max\{|s_{i,j}| - \tfrac{\lambda}{\gamma_2}, 0\}, \tag{50}$$
$$s = Ku - f + b_2.$$

For the Poisson noise model with $F(z, f) = z - f \log z$, the solution is

$$z_{i,j} = s_{i,j}/2 + \sqrt{\left(s_{i,j}/2\right)^2 + \tfrac{\lambda}{\gamma_2} f_{i,j}}, \tag{51}$$
$$s = Ku - \tfrac{\lambda}{\gamma_2} + b_2.$$

u subproblem   With d and z fixed, the u subproblem is

$$\arg \min_u \frac{\gamma_1}{2} \sum_{i,j} |\nabla u_{i,j} - d_{i,j} + b_{i,j}^1|^2 + \frac{\gamma_2}{2} \sum_{i,j} (Ku_{i,j} - z_{i,j} + b_{i,j}^2)^2. \tag{52}$$

The optimal u satisfies

$$\tfrac{\gamma_2}{\gamma_1} K^* K u - \Delta u = \tfrac{\gamma_2}{\gamma_1} K^*(z - b_2) - \text{div}(d - b_1). \tag{53}$$

This equation has the same form as the u subproblem in the Gaussian case, where γ_2/γ_1 substitutes λ/γ and $(z - b_2)$ substitutes f. It can be solved in the Fourier or DCT domain as described previously.

The overall algorithm is

Initialize $u = 0$, $z = 0$, $b^2 = 0$, $d = 0$, $b^1 = 0$
while $\|u_{cur} - u_{prev}\|_2 > Tol$ **do**
    Solve the d subproblem
    Solve the u subproblem
    Solve the z subproblem
 $b^1 = b^1 + \nabla u - d$
    $b^2 = b^2 + u - z$

Algorithm 2

In the implementation, the default parameter values are $Tol = \|f\|_2/1000$, $\gamma_1 = 5$, $\gamma_2 = 8$.

6   Selection of λ

The method requires specifying the fidelity strength, λ, which balances the effects of deblurring and noise reduction. The optimal value of λ is indirectly related to the noise level and the kernel. The value of λ should increase as the noise level decreases or as the kernel becomes less concentrated. While there is no direct exact formula, the optimal λ values can be computed and tabulated over a range of noise levels and kernel sizes to obtain an empirical estimate.

We use a 480×480 crop of the boats image. Blurring is simulated with a disk-shaped kernel of radius $r \in \{1, 2, \ldots, 15\}$ and Gaussian noise is added of standard deviation $\sigma \in \{1, 2, \ldots, 10\}$. Brent's method [3] is used to find the λ value yielding the TV deconvolution result with the minimum mean squared error.
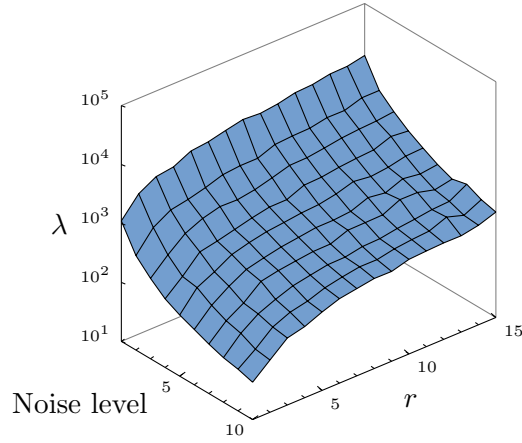
The result is a surface $\lambda(r, \sigma)$ specifying the optimal value for each combination of r and σ. The function is nearly linear along r and decays quickly along σ, suggesting an approximation of the form

$$\hat{\lambda}(r, \sigma) = r \left(\frac{c_1}{\sigma} + \frac{c_2}{\sigma^2} \right), \tag{54}$$

where the coefficients c_1, c_2 are selected for the mean squared error best fit. The average relative error between the approximation and the data is 0.11 and the maximum relative error is 0.38.
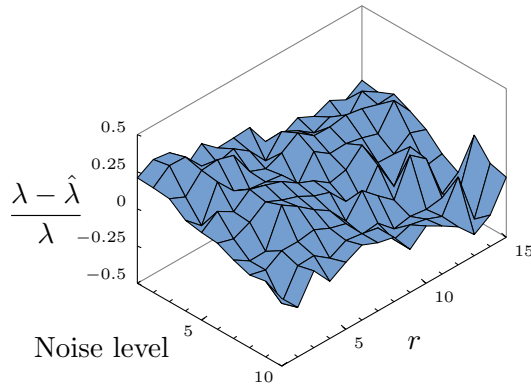
The optimal λ value depends not only on r and σ, but also on the image itself. We tested this by comparing the optimal λ values for different σ with fixed kernel radius $r = 2$ for eight different natural photographic images, including the 480×480 boats image, the peppers image, and Kodak suite images 5, 7, and 23. We also compare the optimal λ values for different r with fixed noise level $\sigma = 5$.
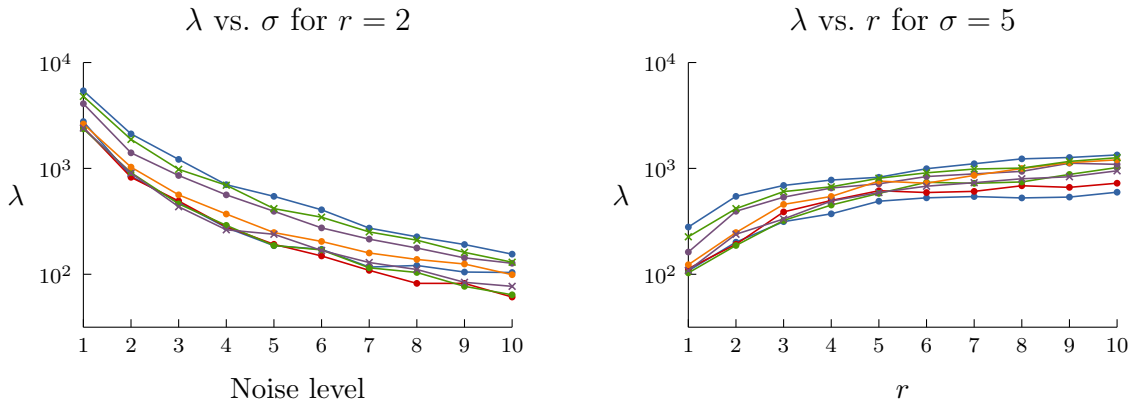
The comparison shows that the optimal λ does indeed vary depending on the image. Fortunately, λ decays at a consistent rate for all images as the noise level σ increases so that the optimal values

differ by a constant scale factor of up to about 2. Similarly, λ increases at roughly the same rate for all images as r increases with a maximum difference of about factor 2. So the estimate $\hat{\lambda}$ should be within an order of magnitude of the optimal value for most images over this range of kernel radius and noise level.

We perform the mean squared error best fit again using the optimal λ values over all eight images to obtain for disk-shaped φ

$$\hat{\lambda}(r, \sigma) = r \left(\frac{c_1}{\sigma} + \frac{c_2}{\sigma^2} \right), \qquad c_1 = 427.9, \ c_2 = 466.4. \tag{55}$$

Similarly, we repeat this process to estimate the optimal λ when φ is a Gaussian with standard deviation $r/2$,

$$\hat{\lambda}(r, \sigma) = r \left(\frac{c_1}{\sigma} + \frac{c_2}{\sigma^2} \right), \qquad c_1 = 117.0, \ c_2 = 4226.3. \tag{56}$$

These formulas are used in the online demo.

7 Examples

7.1 Comparison with Wiener Deconvolution

We first compare TV deconvolution with Wiener deconvolution [1] where the blurred image is simulated with a known kernel and noise is added. To show the best possible linear result, we apply Wiener deconvolution using the exact image's autocorrelation $|\hat{u}|^2$ in the construction of the filter $\hat{w}$. Note that in practice the autocorrelation is rarely available and must be estimated, so a realistic application of Wiener deconvolution would be worse.

The exact image is first convolved with kernel φ (shown below), a disk-shaped kernel of radius $r = 8$ pixels,

$$\varphi(x) = \begin{cases} \frac{1}{\pi r^2} & \text{if } |x| < r, \\ 0 & \text{otherwise.} \end{cases} \tag{57}$$

Gaussian noise is then added with standard deviation 0.01 relative to the intensity range $[0, 1]$. For both methods, the parameters are tuned to maximize the PSNR. TV deconvolution reconstructs sharper edges than Wiener deconvolution.
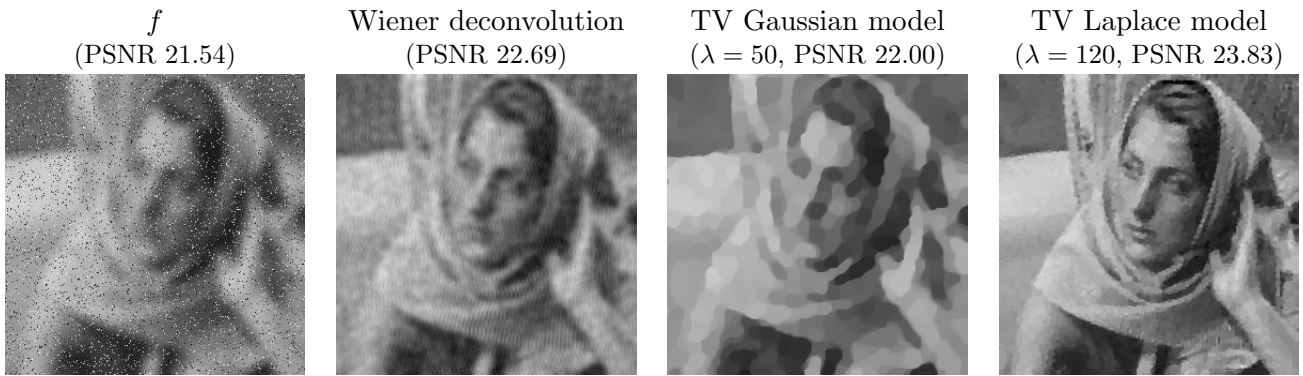
Exact f (PSNR 21.76)   Blur kernel φ

Wiener deconvolution   TV deconvolution
(PSNR 23.99)   ($\lambda = 1600$, PSNR 23.48)
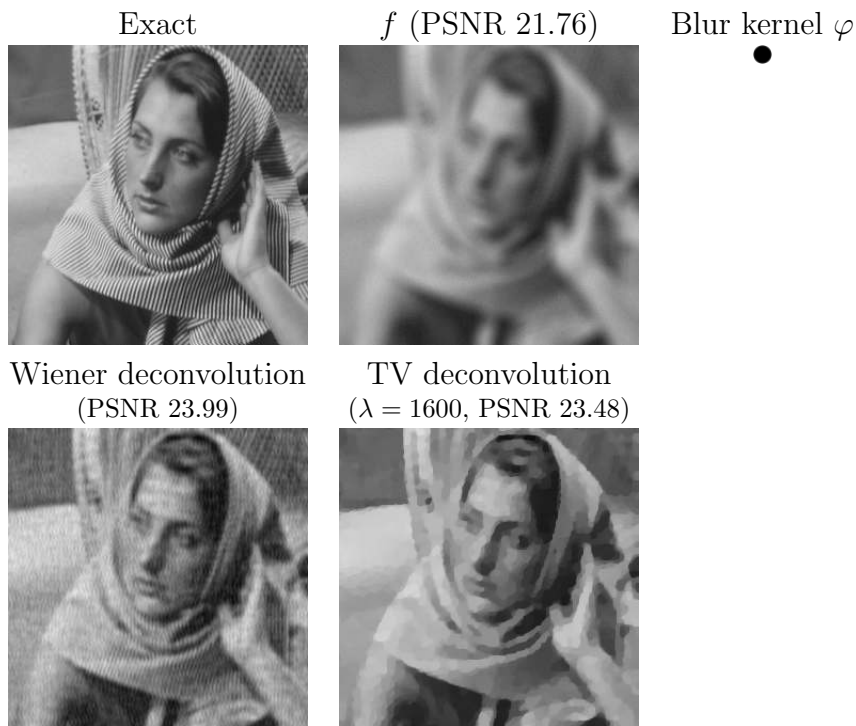
We repeat the previous experiment using a motion blur kernel. The kernel simulates a motion blur of length 20 pixels at angle 5°.

Exact f (PSNR 21.54)   Blur kernel φ

Wiener deconvolution   TV deconvolution
(PSNR 24.59)   ($\lambda = 1600$, PSNR 23.98)
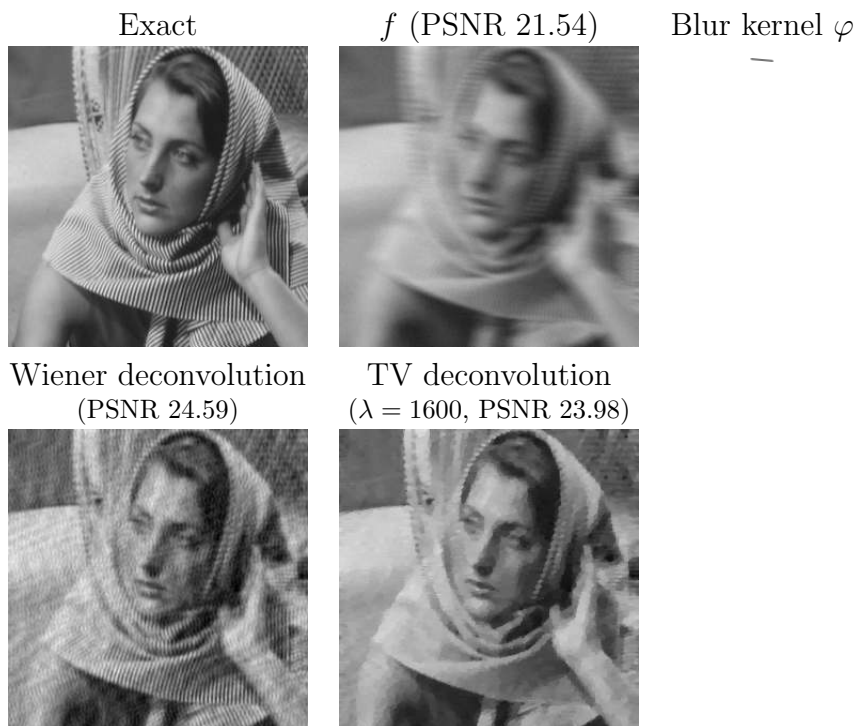
7.2   Deconvolution with Impulsive Noise

The noise model can significantly change the quality of the results. In this example, the image is blurred with a disk-shaped kernel with a radius of 7 pixels and corrupted with 10% impulsive noise, that is, each pixel has a 10% chance to be replaced with a uniform random value. The Laplace noise model is more effective on impulsive noise than the Gaussian noise model. To adapt Wiener deconvolution to impulsive noise, the exact autocorrelation of the noise is used in the NSR numerator.

| f	Wiener deconvolution	TV Gaussian model	TV Laplace model
(PSNR 21.54)	(PSNR 22.69)	($\lambda = 50$, PSNR 22.00)	($\lambda = 120$, PSNR 23.83)

7.3   Correcting an Out-of-Focus Photograph

TV-regularized deconvolution can be used to correct photographs that are out of focus. Provided that the objects in the frame are at approximately the same distance, the effect of the focus can be modeled as a convolution. The following demonstrates deconvolution on a real photograph. The Gaussian noise model is used and the blurring kernel is modeled as a disk of radius r.

The disk radius r and the fidelity weight λ need to be tuned to obtain good results. The effect of varying λ is shown below.

Input	$\lambda = 100, r = 2.3$	$\lambda = 10^3, r = 2.3$	$\lambda = 10^4, r = 2.3$
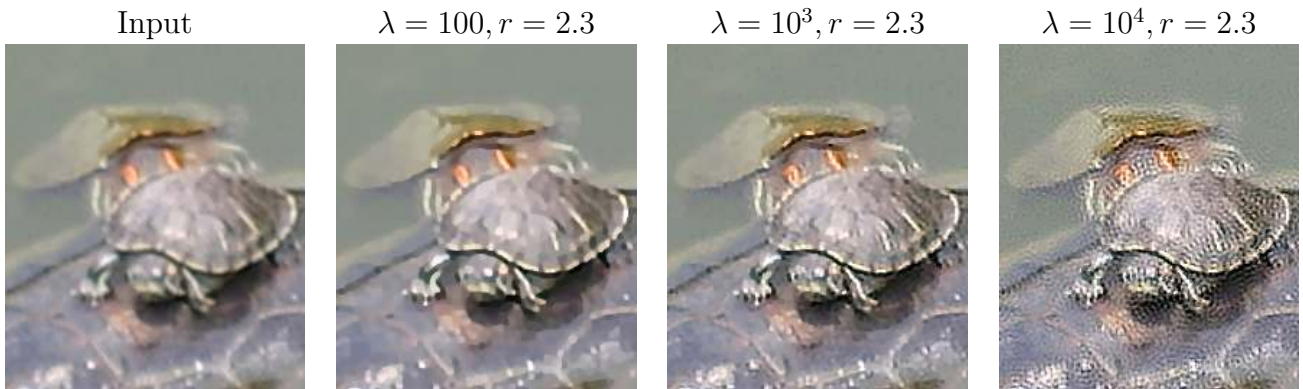
If λ is too small ($\lambda = 100$), then the result loses fine details. If λ is too large ($\lambda = 10^4$), then artifacts appear because the solution is not sufficiently regularized.

The next table shows the effect of varying the blur kernel's radius.

Input	$\lambda = 10^3, r = 1.6$	$\lambda = 10^3, r = 2.3$	$\lambda = 10^3, r = 3.0$
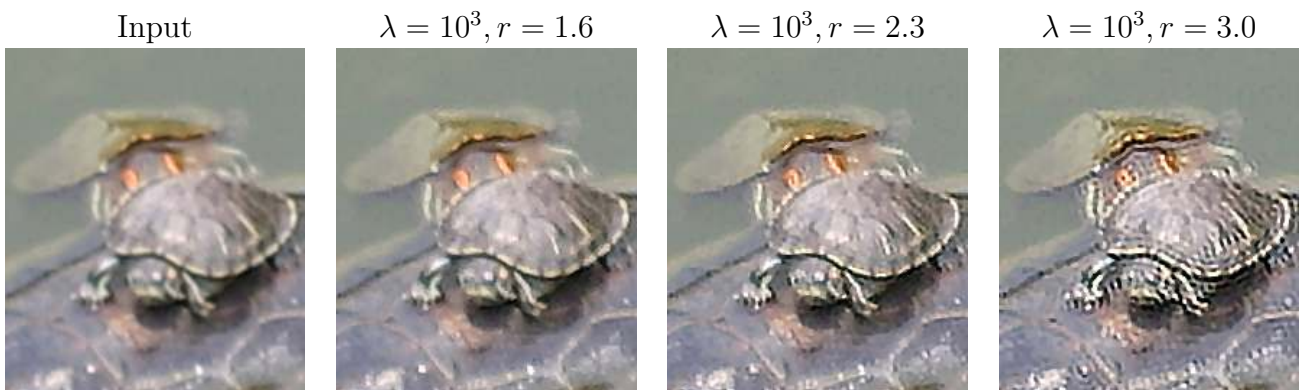
In these images, $\lambda = 10^3$ is fixed and r is varied. When r is too large, oscillation artifacts appear. It is visually better to underestimate rather than overestimate the support of the blur kernel.

7.4   Enhancement

TV-regularized deconvolution can both sharpen edges and remove noise, so it can be applied to perform light sharpening to enhance an image. Good parameters for enhancement are a concentrated kernel for φ and small λ. Here we use a Gaussian with standard deviation of 0.6 pixels and $\lambda = 352$ (corresponding to noise level 4.0 in the online demo) on an image that is slightly out of focus.

Input

TV deconvolution

Acknowledgments

Image Credits

Boats standard test image

Barbara standard test image

Pascal Getreuer, CC-BY

Gary M. Stolz, U.S. Fish and Wildlife Service, Public Domain (http://digitalmedia.fws.gov/cdm4/item_viewer.php?CISOROOT=/natdiglib&CISOPTR=2938)

References

[1] N. Wiener, "Extrapolation, Interpolation, and Smoothing of Stationary Time Series," *MIT Press*, Cambridge, MA, 1964.

[2] L.M. Bregman, "The relaxation method of nding the common points of convex sets and its application to the solution of problems in convex optimization," *USSR Computational Mathematics and Mathematical Physics*, 7:200–217, 1967. http://dx.doi.org/10.1016/0041-5553(67)90040-7

[3] R.P. Brent, "Algorithms for Minimization without Derivatives," *Englewood Cliffs*, NJ: Prentice-Hall, Chapter 5, 1973. ISBN: 0130223352.

[4] L.I. Rudin, S. Osher, E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992. http://dx.doi.org/10.1016/0167-2789(92)90242-F

[5] L.I. Rudin, S. Osher, "Total variation based image restoration with free local constraints," *Proceedings of the IEEE International Conference on Image Processing*, vol. 1, pp. 31–35, 1994. http://dx.doi.org/10.1109/ICIP.1994.413269

[6] R. Acar, C. R. Vogel, "Analysis of total variation penalty methods for ill-posed problems," *Inverse Problems*, vol. 10, pp. 1217–1229, 1994. http://dx.doi.org/10.1088/0266-5611/10/6/003

[7] S. Martucci, "Symmetric convolution and the discrete sine and cosine transforms," *IEEE Transactions on Signal Processing* SP-42, pp. 1038–1051, 1994. http://dx.doi.org/10.1109/78.295213

[8] A. Chambolle, P.L. Lions, "Image recovery via total variational minimization and related problems," *Numerische Mathematik*, vol. 76, pp. 167–188, 1997. http://dx.doi.org/10.1007/s002110050258

[9] P. Blomgren, T.F. Chan, "Color TV: total variation methods for restoration of vector-valued images," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 304–309, 1998. http://dx.doi.org/10.1109/83.661180

[10] T.F. Chan, C.-K. Wong, "Total variation blind deconvolution," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 370–375, 1998. http://dx.doi.org/10.1109/83.661187

[11] A.D. Hillery, R.T. Chin, "Iterative Wiener filters for image restoration," *IEEE Transactions on Signal Processing*, vol. 39, no. 8, pp. 1892–1899, 2002. http://dx.doi.org/10.1109/78.91161

[12] T.F. Chan, J. Shen, *Image Processing and Analysis: Variational, PDE, wavelet, and Stochastic Methods*, SIAM, 2005. http://www.math.ucla.edu/~imagers/chanshen.htm

[13] M. Frigo, S. G. Johnson, "The Design and Implementation of FFTW3," *Proceedings of the IEEE*, vol. 93, no. 2, pp. 216–231, 2005. http://dx.doi.org/10.1109/JPROC.2004.840301

[14] C. Chaux, P. L. Combettes, J.-C. Pesquet, V. R. Wajs, "A variational formulation for frame-based inverse problems," *Inverse Problems*, vol. 23, no. 4, 2007. http://dx.doi.org/10.1088/0266-5611/23/4/008

[15] Y. Wang, J. Yang, W. Yin, Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM Journal on Imaging Sciences*, vol. 1, no. 3, pp. 248–272, 2008. http://dx.doi.org/10.1137/080724265

[16] T. Goldstein, S. Osher, "The Split Bregman Method for L1-Regularized Problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009. http://dx.doi.org/10.1137/080725891

173

[17] A. Levin, Y. Weiss, F. Durand, W.T. Freeman, "Understanding and evaluating blind deconvolution algorithms," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1964–1971, 2009. http://dx.doi.org/10.1109/CVPR.2009.5206815

[18] A. Beck, M. Teboulle, "Fast Gradient-Based Algorithms for Constrained Total Variation Image Denoising and Deblurring Problems," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2419–2434, 2009. http://dx.doi.org/10.1109/TIP.2009.2028250

[19] P. Getreuer, "tvreg v2: Variational Imaging Methods for Denoising, Deconvolution, Inpainting, and Segmentation", 2010. http://www.mathworks.com/matlabcentral/fileexchange/29743-tvreg

[20] S. Becker, J. Bobin, E. J. Candès, "NESTA: A Fast and Accurate First-Order Method for Sparse Recovery," *SIAM Journal on Imaging Sciences*, vol. 4, no. 1, pp. 1–39, 2011. http://dx.doi.org/10.1137/090756855

[21] M. Delbracio, P. Musé, A. Almansa, "Non-parametric sub-pixel local point spread function estimation," *Image Processing On Line*, 2012. http://dx.doi.org/10.5201/ipol.2012.admm-nppsf

[22] P. Getreuer, "Rudin-Osher-Fatemi Total Variation Denoising using Split Bregman," *Image Processing On Line*, 2012. http://dx.doi.org/10.5201/ipol.2012.g-tvd