

# EE 367 / CS 448I Computational Imaging and Display

## Notes: Image Deconvolution (lecture 6)

Gordon Wetzstein  
gordon.wetzstein@stanford.edu

This document serves as a supplement to the material discussed in lecture 6. The document is not meant to be a comprehensive review of image deconvolution, ADMM, or other techniques. It is supposed to be an intuitive introduction to the basic mathematical concepts of non-blind image deconvolution and outline efficient implementation strategies.

### 1 Image Formation

Given a 2D image  $x$  and a shift-invariant 2D convolution kernel or point spread function (PSF)  $c$ , a 2D image  $b$  is formed as

$$b = c * x + \eta. \quad (1)$$

Here,  $b$  is the measured image, which is usually blurry, i.e. in most imaging applications the PSF is an optical low-pass filter. The measurements are corrupted by an additive, signal-independent noise term  $\eta$ .

The convolution theorem states that Equation 1 can be similarly written as a multiplication in the Fourier domain:

$$b = \mathcal{F}^{-1} \{ \mathcal{F} \{c\} \cdot \mathcal{F} \{x\} \}, \quad (2)$$

where  $\cdot$  is the element-wise product. Note that Equations 1 and 2 are numerically (i.e. in Matlab) only equivalent when the convolution is performed with circular boundary conditions.

Deconvolution is the problem of finding an estimate  $\tilde{x}$  of the latent image from blurry, possibly also noisy, measurements  $b$ .

### 2 Inverse Filtering

The most straightforward approach of inverting Equation 2 is inverse filtering. For this purpose, a per-image-frequency division of the optical transfer function (OTF)  $\mathcal{F} \{c\}$  is performed as

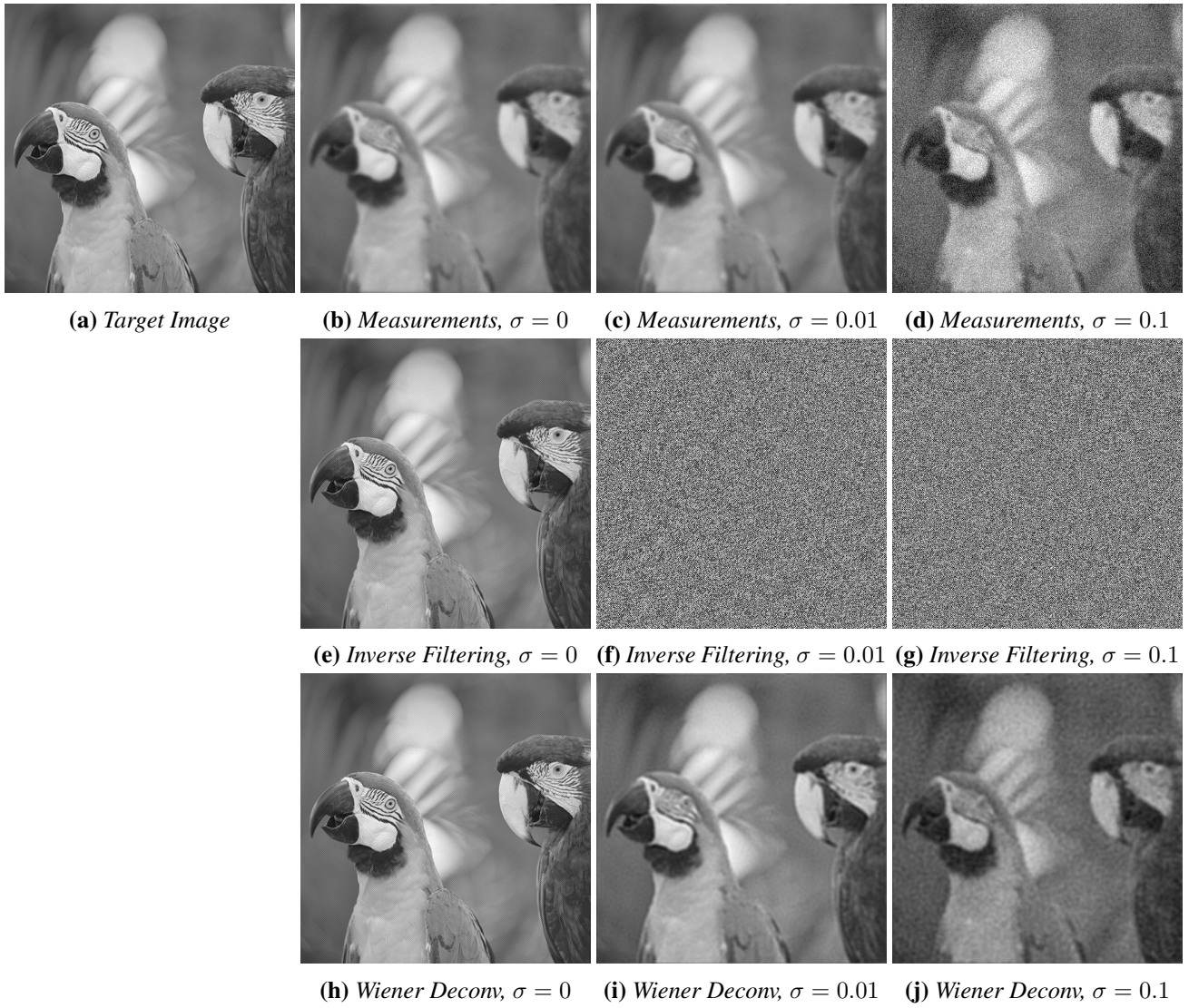
$$\tilde{x} = \mathcal{F}^{-1} \left\{ \frac{\mathcal{F} \{b\}}{\mathcal{F} \{c\}} \right\}. \quad (3)$$

Although inverse filtering is efficient, it is usually problematic when the values of  $\mathcal{F} \{c\}$  are small, i.e. the convolution kernel has zeros in the Fourier domain. Unfortunately, this is the case for most relevant point spread functions in imaging and optics. Divisions by zero or values close to zero will severely amplify measurement noise, as illustrated in Figures 1 and 2.

### 3 Wiener Deconvolution

The primary problem of inverse filtering is that measurement noise is ignored for the reconstruction. Wiener filtering applied to the deconvolution problem adds a damping factor to the inverse filter:

$$\tilde{x} = \mathcal{F}^{-1} \left\{ \frac{|\mathcal{F} \{c\}|^2}{|\mathcal{F} \{c\}|^2 + \frac{1}{SNR}} \cdot \frac{\mathcal{F} \{b\}}{\mathcal{F} \{c\}} \right\}, \quad (4)$$



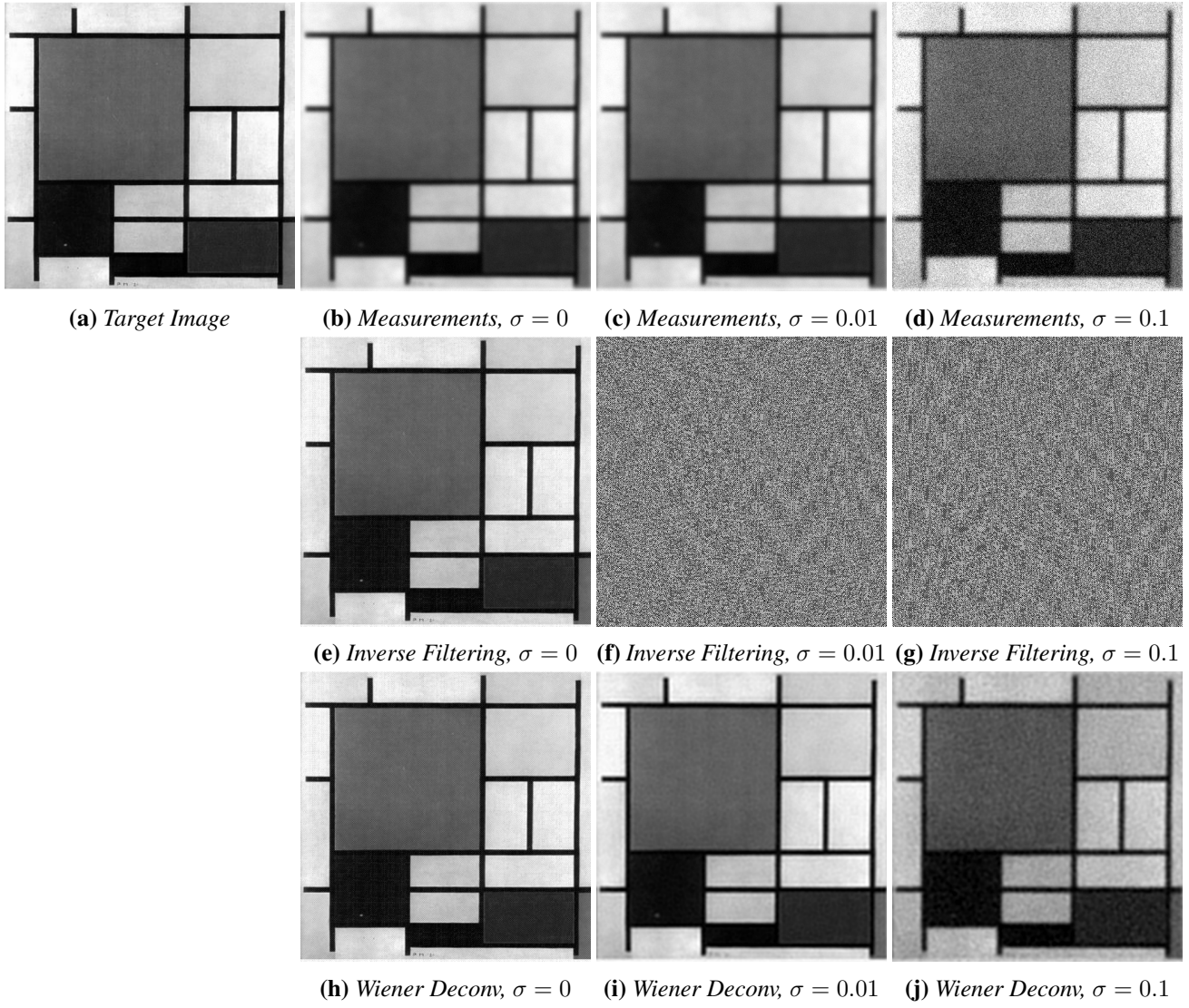
**Figure 1:** Inverse filtering and Wiener Deconvolution for birds image;  $\sigma$  is the standard deviation of the zero-mean i.i.d. Gaussian noise added to the corresponding measurements. The standard deviation of the Gaussian noise term is denoted as  $\sigma$ .

where SNR is the signal-to-noise ratio. If no noise is present in the measurements, the SNR is infinitely high. In that particular case, Wiener filtering is equivalent to inverse filtering. In all other cases, Equation 4 adds a per-frequency damping factor that requires the signal magnitude and the noise power spectral density for each frequency to be known. A common approximation to this is to choose the signal term as the mean image intensity and the noise term as the standard deviation of the (usually zero-mean Gaussian i.i.d.) noise distribution  $\eta$ .

Wiener deconvolution generally achieves acceptable results, as seen in Figures 1 and 2.

## 4 Regularized Deconvolution with ADMM

The primary problem for most deconvolution problems is that they are ill-conditioned. The optical low-pass filter removes high spatial image frequencies before the sensor measures them. Given a PSF or OTF of an optical system, it can be predicted which frequencies are lost. Such an ill-posed mathematical problem usually has infinitely many solutions that would result in the same measurements. Without any prior information that can be additionally



**Figure 2:** Inverse filtering and Wiener Deconvolution for art image;  $\sigma$  is the standard deviation of the zero-mean i.i.d. Gaussian noise added to the corresponding measurements. The standard deviation of the Gaussian noise term is denoted as  $\sigma$ .

imposed on the recovered solution, it is generally impossible to recover images that we would consider “good”. To remove noise and ringing artifacts in the deconvolved images, we will use a prior or regularizer for the estimate.

For this purpose, it is more convenient to vectorize Equation 5 and represent it as a matrix-vector multiplication

$$\mathbf{b} = \mathbf{C}\mathbf{x} + \eta, \quad (5)$$

where  $\mathbf{x} \in \mathbb{R}^N$  is a vector of unknown pixel values,  $\mathbf{b} \in \mathbb{R}^M$  are the vectorized measurements, and  $\mathbf{C} \in \mathbb{R}^{M \times N}$  is the convolution with kernel  $c$  expressed as a matrix-vector multiplication. The convolution matrix  $\mathbf{C}$  is a circulant Toeplitz matrix – its Eigenvalues are the Fourier transform of  $c$ .

A general formulation for the image reconstruction is

$$\underset{\{\mathbf{x}\}}{\text{minimize}} \quad \frac{1}{2} \|\mathbf{C}\mathbf{x} - \mathbf{b}\|_2^2 + \Gamma(\mathbf{x}), \quad (6)$$

where  $\Gamma(\mathbf{x})$  is the regularizer modeling prior knowledge of the latent image. Common priors include smoothness, sparseness, sparse gradients, non-local priors, and many others. In the context of this class, we will focus on an intuitive, yet powerful regularizer: total variation (TV) [Rudin et al. 1992]. For the anisotropic case, the regularizer is modeled as  $\Gamma(\mathbf{x}) = \lambda \|\mathbf{D}\mathbf{x}\|_1$ , with  $\mathbf{D} = [\mathbf{D}_x^T \ \mathbf{D}_y^T]^T$ .  $\mathbf{D} \in \mathbb{R}^{2M \times N}$  represents the finite differences approximation of the horizontal and vertical image gradients:

$$\mathbf{D}_x \mathbf{x} = \text{vec}(d_x * x), \quad d_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{D}_y \mathbf{x} = \text{vec}(d_y * x), \quad d_y = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad (7)$$

where the operator  $\text{vec}(\cdot)$  vectorizes a 2D image and  $d_x$  and  $d_y$  are the convolution kernels representing forward finite differences. The isotropic case is discussed in subsection 4.3. But in essence, both TV priors assume that the image gradients of  $x$  are sparse.

Using a TV prior, Equation 6 is convex but unfortunately the  $\ell_1$ -norm is not differentiable so solving the regularized linear system is not straightforward. Many different solutions exist, we focus on the alternating direction method of multipliers (ADMM) [Boyd et al. 2001]. ADMM is one of the most flexible tools for optimization-based image processing. For an in-depth introduction and overview of this method and related convex optimization concepts, please refer to the Stanford EE graduate course EE364a: Convex Optimization I.

In ADMM notation, the TV-regularized deconvolution problem, also known as Lasso, is formulated as

$$\underset{\{\mathbf{x}\}}{\text{minimize}} \quad \underbrace{\frac{1}{2} \|\mathbf{C}\mathbf{x} - \mathbf{b}\|_2^2}_{f(\mathbf{x})} + \underbrace{\lambda \|\mathbf{z}\|_1}_{g(\mathbf{z})} \quad (8)$$

$$\text{subject to } \mathbf{D}\mathbf{x} - \mathbf{z} = 0 \quad (9)$$

Note that  $\mathbf{z}$  is twice the size of  $\mathbf{x}$  because it contains the image gradients in  $x$  and  $y$ . Clearly, this formulation is equivalent to the original problem. ADMM splits the objective into a weighted sum of two independent functions  $f(x)$  and  $g(z)$  that are only linked through the constraints.

Following the general ADMM strategy, the Augmented Lagrangian of Equation 24 is formed as

$$L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^T (\mathbf{D}\mathbf{x} - \mathbf{z}) + \frac{\rho}{2} \|\mathbf{D}\mathbf{x} - \mathbf{z}\|_2^2 \quad (10)$$

As discussed in more detail in Chapter 3.1 of [Boyd et al. 2001], using the scaled form of the Augmented Lagrangian, the following iterative updates rules can be derived:

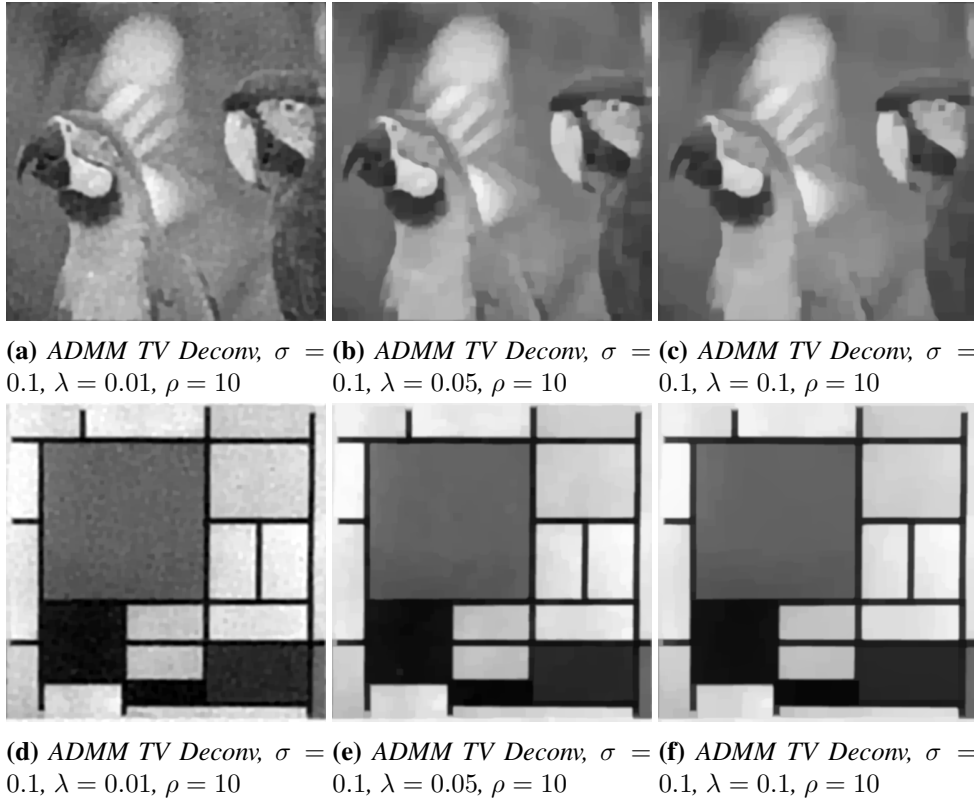
$$\begin{aligned} \mathbf{x} &\leftarrow \text{prox}_{f,\rho}(\mathbf{v}) = \arg \min_{\{\mathbf{x}\}} L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = \arg \min_{\{\mathbf{x}\}} f(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{D}\mathbf{x} - \mathbf{v}\|_2^2, \quad \mathbf{v} = \mathbf{z} - \mathbf{u} \\ \mathbf{z} &\leftarrow \text{prox}_{g,\rho}(\mathbf{v}) = \arg \min_{\{\mathbf{z}\}} L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = \arg \min_{\{\mathbf{z}\}} g(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{v} - \mathbf{z}\|_2^2, \quad \mathbf{v} = \mathbf{D}\mathbf{x} + \mathbf{u} \\ \mathbf{u} &\leftarrow \mathbf{u} + \mathbf{D}\mathbf{x} - \mathbf{z} \end{aligned} \quad (11)$$

where  $\mathbf{u} = (1/\rho)\mathbf{y}$ . The  $\mathbf{x}$  and  $\mathbf{z}$ -updates are performed with what is known as proximal operators  $\text{prox}_{\cdot,\rho} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ . The interested reader is referred to [Boyd et al. 2001] for more details.

#### 4.1 Efficient Implementation of $\mathbf{x}$ -Update

For the  $\mathbf{x}$ -update, we need to derive the proximal operator  $\text{prox}_{f,\rho}$ , which is the following a quadratic program:

$$\text{prox}_{f,\rho}(\mathbf{v}) = \arg \min_{\{\mathbf{x}\}} \frac{1}{2} \|\mathbf{C}\mathbf{x} - \mathbf{b}\|_2^2 + \frac{\rho}{2} \|\mathbf{D}\mathbf{x} - \mathbf{v}\|_2^2, \quad \mathbf{v} = \mathbf{z} - \mathbf{u} \quad (12)$$



**Figure 3:** ADMM deconvolution with anisotropic TV prior and varying  $\lambda$  parameters. All results are better than inverse filtering or Wiener deconvolution, but the choice of the regularization weight  $\lambda$  trades data fidelity of the measurements (i.e. noisy reconstructions) with confidence in the prior (i.e. “patchy” reconstructions). For images that exhibit sparse gradients, such as the artwork, this prior works very well, although more sophisticated priors may be required to adequately model natural images with more complex structures.

To make it easy to follow the derivation step-by-step, we write the objective function out as

$$\begin{aligned} & \frac{1}{2} (\mathbf{C}\mathbf{x} - \mathbf{b})^T (\mathbf{C}\mathbf{x} - \mathbf{b}) + \frac{\rho}{2} (\mathbf{D}\mathbf{x} - \mathbf{v})^T (\mathbf{D}\mathbf{x} - \mathbf{v}) \\ &= \frac{1}{2} (\mathbf{x}^T \mathbf{C}^T \mathbf{C} \mathbf{x} - 2\mathbf{x}^T \mathbf{C}^T \mathbf{b} + \mathbf{b}^T \mathbf{b}) + \frac{\rho}{2} (\mathbf{x}^T \mathbf{D}^T \mathbf{D} \mathbf{x} - 2\mathbf{x}^T \mathbf{D}^T \mathbf{v} + \mathbf{v}^T \mathbf{v}) \end{aligned} \quad (13)$$

The gradient of this expression (i.e. the objective function in Eq. 12) is

$$\mathbf{C}^T \mathbf{C} \mathbf{x} - \mathbf{C}^T \mathbf{b} + \rho \mathbf{D}^T \mathbf{D} \mathbf{x} - \rho \mathbf{D}^T \mathbf{v}, \quad (14)$$

which, equated to zero, results in the normal equations that allow us to derive an expression for estimating  $\tilde{\mathbf{x}}$  as

$$\tilde{\mathbf{x}} = (\mathbf{C}^T \mathbf{C} + \rho \mathbf{D}^T \mathbf{D})^{-1} (\mathbf{C}^T \mathbf{b} + \rho \mathbf{D}^T \mathbf{v}) \quad (15)$$

or using a large-scale, iterative method such as gradient descent, conjugate gradient, or the simultaneous algebraic reconstruction method (SART). For the specific case of 2D image deconvolution, the most efficient way of directly solving the normal equations is inverse filtering (Eq. 3).

To invert Equation 15 analytically using inverse filtering, we need to find expressions for all matrix-vector multiplications that allows us to express them as Fourier-domain multiplications. Both operations  $\mathbf{C}\mathbf{x}$  and  $\mathbf{D}\mathbf{x} =$

$[\mathbf{D}_x^T \quad \mathbf{D}_y^T]^T \mathbf{x}$  (and also their adjoint operations  $\mathbf{C}^T \mathbf{b}$  and  $\mathbf{D}^T \mathbf{v} = [\mathbf{D}_x^T \quad \mathbf{D}_y^T] \mathbf{v} = \mathbf{D}_x^T \mathbf{v}_1 + \mathbf{D}_y^T \mathbf{v}_2$ ) can be expressed as convolutions, i.e.  $c * x$  and  $d_{x/y} * x$ . Therefore, we can write operators for the matrix-vector multiplications in Equation 15 as

$$(\mathbf{C}^T \mathbf{C} + \rho \mathbf{D}^T \mathbf{D}) \Leftrightarrow \mathcal{F}^{-1} \{ \mathcal{F} \{c\}^* \cdot \mathcal{F} \{c\} + \rho (\mathcal{F} \{d_x\}^* \cdot \mathcal{F} \{d_x\} + \mathcal{F} \{d_y\}^* \cdot \mathcal{F} \{d_y\}) \} \} \quad (16)$$

$$(\mathbf{C}^T \mathbf{b} + \rho \mathbf{D}^T \mathbf{v}) \Leftrightarrow \mathcal{F}^{-1} \{ \mathcal{F} \{c\}^* \cdot \mathcal{F} \{b\} + \rho (\mathcal{F} \{d_x\}^* \cdot \mathcal{F} \{v_1\} + \mathcal{F} \{d_y\}^* \cdot \mathcal{F} \{v_2\}) \} \} \quad (17)$$

which gives rise to the inverse filtering proximal operator, which applies only operators (Fourier transform and element-wise multiplication or division) to solve Equation 12 as

$$\mathbf{prox}_{f,\rho}(\mathbf{v}) = \mathcal{F}^{-1} \left\{ \frac{\mathcal{F} \{c\}^* \cdot \mathcal{F} \{b\} + \rho (\mathcal{F} \{d_x\}^* \cdot \mathcal{F} \{v_1\} + \mathcal{F} \{d_y\}^* \cdot \mathcal{F} \{v_2\})}{\mathcal{F} \{c\}^* \cdot \mathcal{F} \{c\} + \rho (\mathcal{F} \{d_x\}^* \cdot \mathcal{F} \{d_x\} + \mathcal{F} \{d_y\}^* \cdot \mathcal{F} \{d_y\})} \right\}, \quad (18)$$

where  $\cdot$  is the element-wise product and. Just like inverse filtering, the  $\mathbf{x}$ -update itself may be unstable w.r.t. noise and zeros in the OTF, but embedded in the ADMM iterations this will not affect the resulting estimate of  $\mathbf{x}$ . Note that all parts of Equation 18 that do not depend on  $\mathbf{v}$  can be precomputed and do not have to be updated throughout the ADMM iterations.

## 4.2 Efficient Implementation of $\mathbf{z}$ -Update

In the  $\mathbf{z}$ -update, the  $\ell_1$ -norm is convex but not differentiable. Nevertheless, a closed form solution for the proximal operator exists, such that

$$\mathbf{prox}_{g,\rho}(\mathbf{v}) = \mathcal{S}_{\lambda/\rho}(\mathbf{v}) = \arg \min_{\{\mathbf{z}\}} \lambda \|\mathbf{z}\|_1 + \frac{\rho}{2} \|\mathbf{v} - \mathbf{z}\|_2^2, \quad (19)$$

with  $\mathbf{v} = \mathbf{D}\mathbf{x} + \mathbf{u}$  and  $\mathcal{S}_\kappa(\cdot)$  being the element-wise soft thresholding operator

$$\mathcal{S}_\kappa(v) = \begin{cases} v - \kappa & v > \kappa \\ 0 & |v| \leq \kappa \\ v + \kappa & v < -\kappa \end{cases} = (v - \kappa)_+ - (-v - \kappa) \quad (20)$$

that can be implemented very efficiently.

## 4.3 The Isotropic Case

Instead of the  $\ell_1$ -norm, the isotropic case uses the sum of the  $\ell_2$ -norms of the finite differences approximation of the horizontal and vertical image gradients as a regularizer. If we choose  $\mathbf{z} \in \mathbb{R}^{2 \times M}$  so that  $\mathbf{z} = [\mathbf{D}_x x \quad \mathbf{D}_y x]^T$ , we can use the  $\ell_{2,1}$ -norm to write the isotropic version of the regularizer as

$$\Gamma(\mathbf{x}) = \lambda \|\mathbf{z}\|_{2,1} \quad (21)$$

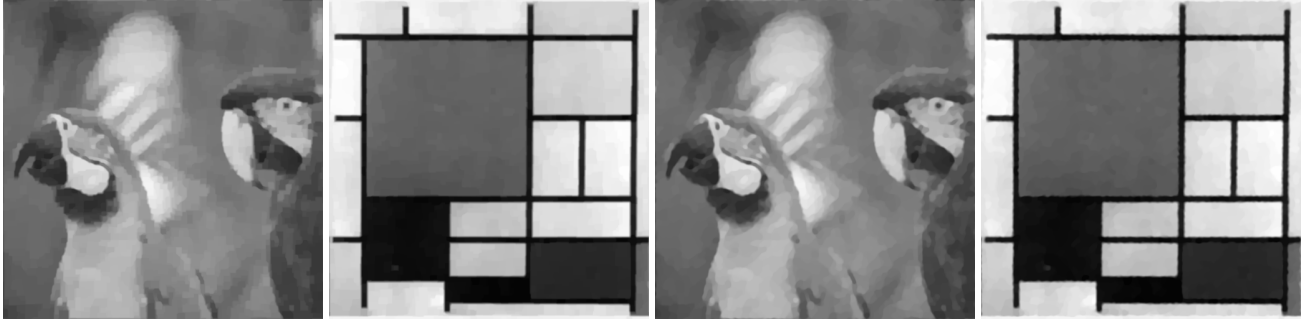
This is can be rewritten as a sum-of-norms regularizer, also known as the group lasso

$$\Gamma(\mathbf{x}) = \lambda \sum_{i=1}^M \left\| \begin{bmatrix} (\mathbf{D}_x x)_i \\ (\mathbf{D}_y x)_i \end{bmatrix} \right\|_2 \quad (22)$$

Using the same notation as in previous subsections, the deconvolution problem with an isotropic TV prior is formulated in ADMM notation as

$$\underset{\{x\}}{\text{minimize}} \quad \underbrace{\frac{1}{2} \|\mathbf{C}\mathbf{x} - \mathbf{b}\|_2^2}_{f(\mathbf{x})} + \lambda \underbrace{\sum_{i=1}^M \left\| \begin{bmatrix} z_i \\ z_{i+M} \end{bmatrix} \right\|_2}_{g(\mathbf{z})} \quad (23)$$

$$\text{subject to } \mathbf{D}\mathbf{x} - \mathbf{z} = 0 \quad (24)$$



(a) ADMM anisotropic TV, (b) ADMM anisotropic TV, (c) ADMM isotropic TV,  $\sigma = 0.1, \lambda = 0.05, \rho = 10$  (d) ADMM isotropic TV,  $\sigma = 0.1, \lambda = 0.05, \rho = 10$

**Figure 4:** ADMM deconvolution with anisotropic TV prior and isotropic TV prior. For the art picture the differences are barely noticeable, whereas for the birds the isotropic prior achieves slightly better results.

where  $z_i$  is the  $i$ -th element of  $\mathbf{z}$ . For  $1 \leq i \leq M$  it is meant to represent the finite differences approximation in horizontal direction,  $(\mathbf{D}_x x)_i$ , and for  $M + 1 \leq i \leq 2M$  the finite differences approximation in vertical direction,  $(\mathbf{D}_x y)_i$ . Notice that if we replace the  $\ell_2$ -norm in  $g(\mathbf{z})$  with the  $\ell_1$ -norm, then we get  $\sum_{i=1}^M \|(z_i, z_{i+M})\|_1$  which reduces to  $\|\mathbf{z}\|_1$  and we recover the anisotropic case.

Since only the regularization term  $g(\mathbf{z})$  changed as compared to the anisotropic case, the  $\mathbf{x}$ -update and  $\mathbf{u}$ -update rules for ADMM stay the same as above and only the  $\mathbf{z}$ -update changes to

$$\mathbf{z} \leftarrow \text{prox}_{g,\rho}(\mathbf{v}) = \arg \min_{\{\mathbf{z}\}} \lambda \sum_{i=1}^M \left\| \begin{bmatrix} z_i \\ z_{i+M} \end{bmatrix} \right\|_2 + \frac{\rho}{2} \|\mathbf{v} - \mathbf{z}\|_2^2, \quad \mathbf{v} = \mathbf{D}\mathbf{x} + \mathbf{u} \quad (25)$$

The corresponding proximal operator of  $g(\mathbf{z})$ , the group lasso, is block soft thresholding.

The  $\mathbf{z}$ -update rule then becomes

$$\begin{bmatrix} z_i \\ z_{i+M} \end{bmatrix} \leftarrow \mathcal{S}_{\lambda/\rho} \left( \begin{bmatrix} v_i \\ v_{i+M} \end{bmatrix} \right), \quad 1 \leq i \leq M \quad (26)$$

where  $\mathcal{S}_\kappa(\cdot)$  is the vector soft-thresholding operator

$$\mathcal{S}_\kappa(\mathbf{a}) = \left( 1 - \frac{\kappa}{\|\mathbf{a}\|_2} \right)_+ \mathbf{a}$$

#### 4.4 Pseudo Code and Additional Information

For additional information on implementation, please see the example code provided on the ADMM website <http://stanford.edu/~boyd/papers/admm/>.

Further, Algorithm 1 outlines pseudo code for the ADMM deconvolution with an anisotropic TV prior.

## References

BOYD, S., PARIKH, N., CHU, E., PELEATO, B., AND ECKSTEIN, J. 2001. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 3, 1, 1–122.

---

**Algorithm 1** ADMM for deconvolution with anisotropic TV prior

---

```
1: initialize  $\rho$  and  $\lambda$ 
2:  $\mathbf{x} = \text{zeros}(W, H)$ ;
3:  $\mathbf{z} = \text{zeros}(W, H, 2)$ ;
4:  $\mathbf{u} = \text{zeros}(W, H, 2)$ ;
5: for  $k = 1$  to  $\text{maxIters}$  do
6:    $\mathbf{x} = \text{prox}_{f,\rho}(\mathbf{z} - \mathbf{u}) = \arg \min_{\{\mathbf{x}\}} \frac{1}{2} \|\mathbf{C}\mathbf{x} - \mathbf{b}\|_2^2 + \frac{\rho}{2} \|\mathbf{D}\mathbf{x} - \mathbf{z} + \mathbf{u}\|_2^2$ 
7:    $\mathbf{z} = \text{prox}_{g,\rho}(\mathbf{D}\mathbf{x} + \mathbf{u}) = \mathcal{S}_{\lambda/\rho}(\mathbf{D}\mathbf{x} + \mathbf{u})$ 
8:    $\mathbf{u} = \mathbf{u} + \mathbf{D}\mathbf{x} - \mathbf{z}$ 
9: end for
```

---

RUDIN, L., OSHER, S., AND FATEMI, E. 1992. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena* 60, 14, 259 – 268.