Title
Intensity-constrained total variation regularization for image denoising and deblurring

Author
Swenson, Daniel

Publication Date
2011

Peer reviewed|Thesis/dissertation

University of California, Merced

Capstone Project

Intensity-Constrained Total Variation Regularization for Image Denoising and Deblurring

Daniel Swenson

A technical report submitted
in partial fulfillment of the requirements for the degree of

Master of Science in Applied Mathematics

May, 2011

University of California, Merced
Graduate Division

This is to certify that I have examined a copy of a technical report by

Daniel Swenson

and found it satisfactory in all respects, and that any and all revisions required by the examining committee have been made.

Research Advisor: _____
Roummel Marcia

Reading Committee: _____
Arnold Kim

Applied Mathematics Graduate Studies Chair: _____
Boaz Ilan

_____
May 4, 2011

ABSTRACT OF THE CAPSTONE PROJECT

Intensity-Constrained Total Variation Regularization for Image Denoising and Deblurring

by

Daniel Swenson
May 2011

University of California, Merced

Abstract

Many problems in digital image processing are problems in *digital image restoration*. In other words, we have a noisy, incomplete, or otherwise corrupted image, and our goal is to estimate the original image from the corrupted image by solving an appropriate optimization problem. Typically, in order to restore such corrupted images, we solve an optimization problem that penalizes the "roughness" (and consequently, promotes the smoothness) of our estimate of the original image. Often, we may have knowledge about some properties of the original image that our estimate may not possess. If we impose these properties explicitly as constraints in our optimization problem, then we may be able to significantly improve our estimate of the original image. Our goal in this technical report is to examine what benefits are incurred when we impose an intensity constraint in the context of *total variation regularization*, which is a common image restoration tool that works by encouraging smoothness of the restored image. We will provide analytical and numerical justification that the intensity constraint does not significantly affect the outcome of total variation regularization when applied to denoising and deblurring problems. Consequently, we can achieve the same image restoration results at a lower computational cost, because we do not have to solve projection subproblems to enforce the intensity constraint. In the future, we would like to see if our projection algorithm improves the results of common tasks in digital image *enhancement*, such as image sharpening.

1 Introduction

Many problems in digital image processing are problems in *digital image restoration*. In other words, we have a noisy, incomplete, or otherwise corrupted image, and we are trying to estimate the original image from the corrupted image by solving an appropriate optimization problem. Typically, we solve an optimization problem that penalizes the "roughness" (and consequently, promotes the smoothness) of our estimate of the original image, because natural images tend to be smoother than corrupted images. Often, we may have additional knowledge about some properties the original image that our estimate of the original image may or may not possess. For example, 8-bit black-and-white digital images have pixel values that lie between 0 and 255. This is not always honored by our estimate of the true image. Also, we may sometimes know or have a reasonable guess regarding what the overall intensity of the true image might be. Again, this may or may not be the same as the overall intensity of our estimate of the true image. If we impose the above properties explicitly as constraints in our optimization problem, then we may be able to significantly improve our estimate of the original image. Our goal in this technical report is to examine what happens when we impose the above constraints to perform image denoising and deblurring using *total variation regularization*, a common image restoration tool that we will describe in detail shortly.

2 Problem Formulation

The digital image restoration problems that we will consider in this technical report will be *linear inverse problems*, that is, problems of the form

$$y = Au + \eta, \tag{1}$$

where $u \in \mathbb{R}^n$ is an image containing n pixels, $A \in \mathbb{R}^{n \times n}$ is a linear operator, $\eta \in \mathbb{R}^n$ is a noise term (which we will take to be zero-mean and Gaussian), and $y \in \mathbb{R}^n$ represents the noisy and/or corrupted image that we observe. The problem then becomes finding a reasonable estimate $\hat{u}$ for u by solving an appropriate optimization problem. Many problems in digital signal processing, including denoising, deblurring, and other problems, can be modeled as specific examples of the general problem (1). The naive approach to solving the problem (1), namely, computing

$$\hat{u} = A^{-1}y = u + A^{-1}\eta$$

yields an estimate $\hat{u}$ that is not useful if A is ill-conditioned, because in that case the "inverted noise" term $A^{-1}\eta$ dominates the estimate (see [8]). We may have general knowledge about the statistical properties of η, but we will not know specifically what η is, so we cannot simply remove it from the image. Consequently, we must solve a suitable optimization problem to obtain a reasonable estimate of u.

3 Optimization Problem

Many approaches to solving the problem (1) involve finding

$$\hat{u} = \arg\min_{u \in \mathbb{R}^n} \frac{1}{2}||y - Au||_2^2 + \tau \text{pen}(u) \tag{2}$$

where $\frac{1}{2}||y - Au||_2^2$ is a term that ensures that the argument u that minimizes (2) is not "too far"

from our observed data y, pen(u) is some function that is designed to promote certain desirable properties in the image (often referred to as a *regularization* term), and τ is a positive parameter that determines how heavily regularized the resulting image will be. The function pen(u) tends to be a function that penalizes some measure of "roughness" in the image. The problem (2) is also sometimes expressed in the constrained form

$$\min_u \text{pen}(u) \quad \text{subject to } ||y - Au||_2^2 \leq ||\eta||_2^2. \tag{3}$$

A classical image processing technique that follows the above framework for obtaining an estimate $\hat{u}$ for u in (1) (as described in [5] and [7]), involves solving an optimization problem of the form

$$\min_u ||\Gamma u||_2^2 \quad \text{subject to } ||y - Au||_2^2 \leq ||\eta||_2^2 \tag{4}$$

where Γ is chosen to be the discrete Laplacian operator. This problem, then, yields an estimate $\hat{u}$ that is as smooth as it can be (in the sense that Γ measures smoothness) subject to the constraint that our estimate $\hat{u}$ satisfies $||y - A\hat{u}||_2^2 \leq ||\eta||_2^2$. (This requires us to have an estimate of $||\eta||_2^2$, which we may or may not have, depending on the situation at hand.) This problem (4) is convenient in that the objective function and the constraint are both smooth, and the solution can be computed efficiently by using an iterative method in the Fourier domain (see [7] for details). According to [8], however, approaches such as the one described above do not do as good of a job of preserving details or edges that are present in u as the approach that we will use in this technical report.

Advances in bases available for digital image processing and in optimization techniques have made it viable to solve optimization problems that preserve the edges of the original image u or preserve details in u in ways that the solution to (4) does not. Interest in this field has grown recently with the advent of *compressed sensing*[1]. Compressed sensing is the process of acquiring a signal in an "undersampled" manner and reconstructing the signal utilizing the prior knowledge that it has a sparse representation in some basis (or dictionary). (For a good overview of compressed sensing, see [14].) The resulting optimization problems are of the form

$$\min_\theta \frac{1}{2}||y - AW\theta||_2^2 + \tau||\theta||_1 \tag{5}$$

or alternatively in the constrained form

$$\min_\theta ||\theta||_1 \quad \text{subject to } ||y - AW\theta||_2^2 \leq ||\eta||_2^2 \tag{6}$$

where τ is a non-negative parameter, and W is a *sparsity-inducing* representation - that is, one in which we can represent most of the energy of u using only a few coefficients. If we denote the solution to (5) by $\hat{\theta}$, then $\hat{u} = W\hat{\theta}$. The formulation (5) is more often used in practice since, as an unconstrained problem, it is easier to solve. Solutions to either formulation of this problem tend to promote the sparsity of $\hat{\theta}$. When using bases such as *wavelet* bases, the problems (5) and (6) promote piecewise-smooth reconstructions $\hat{u}$ (see [13]). For a treatment of wavelets in an image processing context, we refer the reader to [7]. For a more general treatment of wavelets the reader is encouraged to consult [10] or [13]. Software that solves either the formulation (5) or the formulation (6) includes [3], [4], [6], [15], and many other software packages.

[1]Note that the image restoration problems considered in this technical report are not of compressed sensing type.

4 Total Variation

Total variation regularization, first introduced by Rudin, Osher, and Fatemi in [12], is an alternative penalty which induces sparsity in an image's *gradient* by minimizing the change between adjacent pixels. Specifically, let $u_{i,j}$ denote the pixel in the ith row and jth column of an $m \times n$ image u, let h and v signify the horizontal and vertical gradients of u, and define the operators

$$D_{h;ij}u = \begin{cases} u_{i+1,j} - u_{i,j}, & i \leq m \\ 0 & i = m \end{cases} \quad D_{v;ij}u = \begin{cases} u_{i,j+1} - u_{i,j}, & j \leq n \\ 0 & j = n \end{cases}$$

and

$$D_{ij}u = \begin{pmatrix} D_{h;ij}u \\ D_{v;ij}u \end{pmatrix}$$

$D_{ij}u$ can be interpreted as a discrete gradient of the digital image u. The total variation of u is the sum of the magnitudes of this discrete gradient at every point:

$$TV(u) = \sum_{i,j} \sqrt{(D_{h;ij}u)^2 + (D_{v;ij}u)^2} = \sum_{i,j} ||D_{ij}u||_2. \tag{7}$$

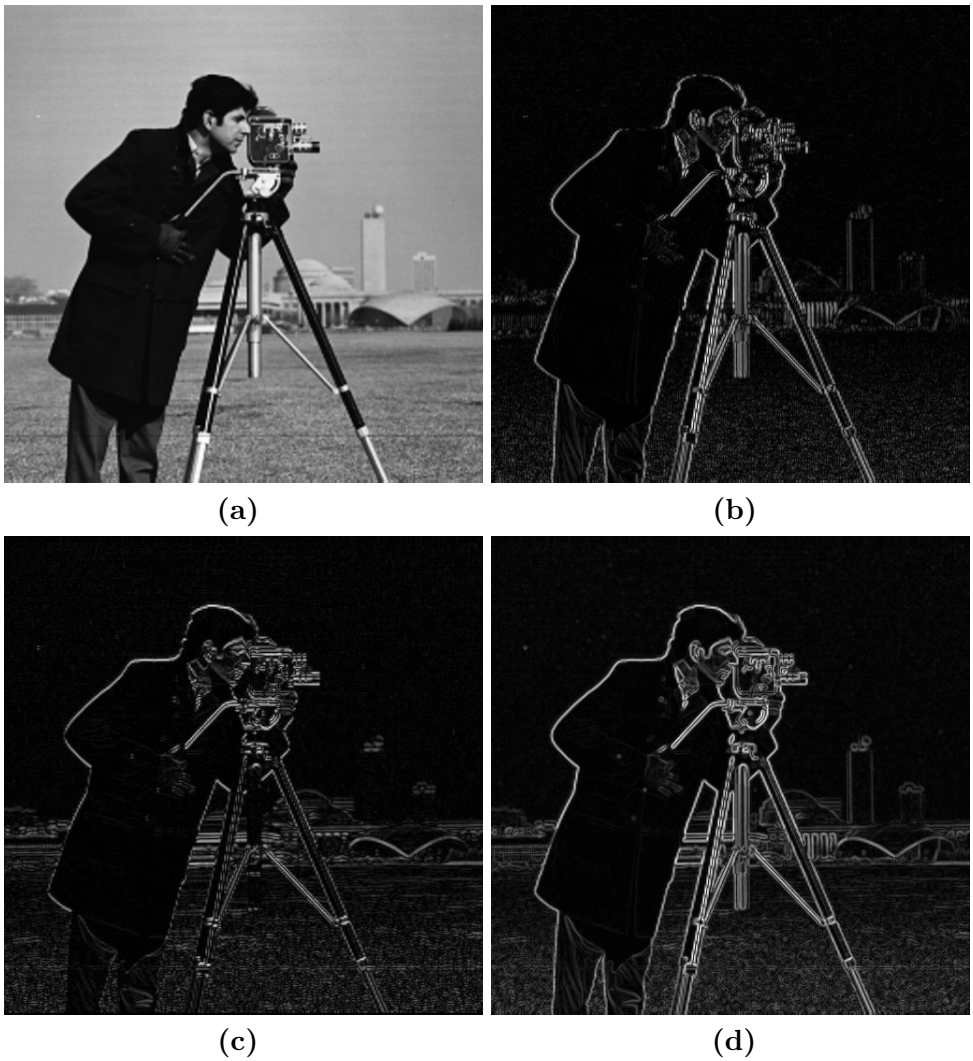
(This expression is also sometimes called the *isotropic* total variation of u.) For comparison, let us consider the related functional

$$J(u) = \sum_{i,j} ((D_{h;ij}u)^2 + (D_{v;ij}u)^2) = \sum_{i,j} ||D_{ij}u||_2^2.$$

Here, following the exposition in [8], we will give an intuition regarding the benefits of using (7) rather than the more traditional functional $J(u)$ for image regularization. The expression $J(u)$ is easier to work with due to the absence of the square roots (see [8]), but the presence of the square root acting on each term in the summation in (7) is beneficial since it de-emphasizes large changes (namely, edges) in u. Another form of total variation, the so-called *anisotropic* total variation of u, is given by

$$TV_{\ell_1}(u) = \sum_{i,j} (|D_{h;ij}u| + |D_{v;ij}u|) = \sum_{i,j} ||D_{ij}u||_1. \tag{8}$$

Here, the absolute value functions de-emphasize large changes in u compared to the squaring operations in $J(u)$, which again means that (8) penalizes the edges of u less severely than $J(u)$ does. Regarding the differences between (7) and (8), note that if we use the isotropic total variation (7) in our optimization problems, there is no preference for smoothing in any particular direction (i.e., the image is smoothed in all directions), but if we use the anisotropic total variation (8), there is a preference for smoothing in the horizontal and vertical directions. To illustrate our terminology, we include a picture on the following page of the standard "cameraman" test image, its horizontal gradient $D_{h;ij}u$, its vertical gradient $D_{v;ij}u$, and the magnitude of the gradient $||D_{ij}u||_2$ (see Figure 1). The isotropic total variation $TV(u)$ of the test image is the sum of all of the pixel values of the image $||D_{ij}u||_2$ in the lower-right-hand corner.



(a) **(b)**

(c) **(d)**

Figure 1: **(a)** The cameraman test image. **(b)** Its horizontal gradient $D_{h;ij}u$. **(c)** Its vertical gradient $D_{v;ij}u$. **(d)** The magnitude $||D_{ij}u||_2$ of the gradient. Note the strong vertical edges of the town captured by $D_{h;ij}u$ in (b) and the horizontal rooftop edges of the building captured by $D_{v;ij}u$ in (c).

We can use total variation regularization to estimate our original image u given y, A, and an estimate of $||\eta||_2^2$, by solving the optimization problem

$$\min_u \frac{1}{2}||y - Au||_2^2 + \tau TV(u) \tag{9}$$

for an appropriate value of $\tau > 0$. Total variation regularization is also sometimes augmented with constraints. A common constrained total variation regularization problem is

$$\min_u \quad \frac{1}{2}||y - Au||_2^2 + \tau TV(u) \tag{10a}$$

$$\text{subject to } a \leq u \leq b \tag{10b}$$

where $a, b \in \mathbb{R}^n$ and $a \leq u \leq b$ should be understood as a component-wise inequality.

5 Algorithm for Total Variation Regularization

A mathematical approach that is often used for total variation regularization is Beck and Teboulle's Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [1]. FISTA was designed to efficiently solve the problem (6) by alternately employing a total variation regularization algorithm to denoise the image and (if $A \neq I$) taking steps in the direction of the minus gradient of $\frac{1}{2}||y - Au||_2^2$ to ensure data fidelity. Between each data fidelity step, a few denoising steps are taken. Let u_d denote the image generated after any particular denoising step. After each iteration of the denoising algorithm, the resulting image u_d is projected onto the feasible set defined by (10b), so that at the end of each iteration of the denoising algorithm, we have that the projected image u_p is given by

$$u_p = \arg\min_u \frac{1}{2}||u - u_d||_2^2 \text{ subject to } a \leq u \leq b$$

which is easily accomplished by setting

$$(u_p)_{i,j} = \begin{cases} a & \text{if } (u_d)_{i,j} < a, \\ (u_d)_{i,j} & \text{if } a \leq (u_d)_{i,j} \leq b, \\ b & \text{if } (u_d)_{i,j} > b. \end{cases}$$

The next denoising iteration is then initialized with u_p. In this way FISTA ensures that the solution to (6) satisfies the constraints (10b). FISTA can be modified to perform total variation regularization subject to arbitrary constraints if the user supplies a projection algorithm that projects u_d onto the desired constraints. We will develop such a projection approach in the following section.

6 Intensity-Constrained Total Variation Regularization

We are interested in solving the following problem:

$$\begin{aligned} \min_u \quad & \frac{1}{2}||Au - y||_2^2 + \tau||u||_{T.V.} && \text{(11a)} \\ \text{subject to} \quad & -u \leq 0, && \text{(11b)} \\ & u - 255\mathbb{1} \leq 0, && \text{(11c)} \\ & ||u||_1 - I_0 = 0. && \text{(11d)} \end{aligned}$$

where $\mathbb{1}$ is a vector of ones. The constraints reflect the fact that our 8-bit images have pixel values between 0 and 255, and that the true image's intensity I_0 is known. The algorithm FISTA can solve this problem if we supply it with an algorithm that solves the projection subproblem

$$\begin{aligned} \min_u \quad & \frac{1}{2}||u_d - u||_2^2 && \text{(12a)} \\ \text{subject to} \quad & -u \leq 0, && \text{(12b)} \\ & u - 255\mathbb{1} \leq 0, && \text{(12c)} \\ & ||u||_1 - I_0 = 0 && \text{(12d)} \end{aligned}$$

(Note that (11d) reduces to (12d) because of (11b).) One naive attempt at solving the problem (12) involves projecting u_d first onto the constraints (12b) and (12c), and then projecting the result onto the constraint (12d). However, in projecting onto (12d) last, it is quite likely that we will violate (12c). Similarly, if we project u_d onto (12b), (12d), and then (12c), we are quite likely to violate (12d). Therefore, projecting onto each constraint individually will not yield the answer to (12). Consequently, to compute the solution of (12), we will solve the corresponding *Lagrange dual problem*, which incorporates the above constraints into the objective function and replaces them with simpler (bound) constraints. We will appeal to standard results (see [2]) that show that the solution of the dual problem yields the solution u^* to the primal problem (12). The *Lagrangian* $\mathcal{L}(u, \lambda, \xi, \psi)$ corresponding to (12) is

$$\mathcal{L}(u, \lambda, \xi, \psi) = \frac{1}{2}||u_d - u||_2^2 - \lambda^T u + \xi^T(u - 255\mathbb{1}) + \psi(\mathbb{1}^T u - I_0) \tag{13}$$

with $\lambda, \xi \in \mathbb{R}^n$ and $\psi \in \mathbb{R}$, where we have partitioned the Lagrange multipliers corresponding to the inequality constraints from (12) into λ and ξ for convenience. To get the Lagrange dual function corresponding to the Lagrangian (13), we set $\frac{\partial \mathcal{L}}{\partial u} = 0$, which yields

$$u^* = u_d + \lambda - \xi - \psi\mathbb{1}. \tag{14}$$

Substituting the above expression for u^* in for u in (13) we get the Lagrange dual problem corresponding to the primal problem (12), which is

$$\max_{\lambda,\xi,\psi} \quad g(\lambda, \xi, \psi) \tag{15a}$$

$$\text{subject to } \lambda \geq 0, \xi \geq 0 \tag{15b}$$

where

$$g(\lambda, \xi, \psi) = -\frac{1}{2}n\psi^2 - \frac{1}{2}\lambda^T\lambda - \frac{1}{2}\xi^T\xi - \lambda^T u_d + \xi^T u_d - 255\mathbb{1}^T\xi + \psi(\mathbb{1}^T u_d) + \psi(\mathbb{1}^T\lambda) - \psi(\mathbb{1}^T\xi) - \psi I_0.$$

where n is the number of pixels in u. The maximum value of the objective function in (12) is equal to the minimum value of the objective function in (7), and the solution u^* to (7) is given by (14). This is guaranteed by the fact that our primal problem (12) satisfies *Slater's condition*. For a problem such as ours where the constraints and objective function are convex and all of the inequality constraints are of the form $r_i^T u - s_i \leq 0$, Slater's condition says that if the optimization problem in question has a non-empty feasible set, then the maximum value of (12) is in fact equal to the minimum value of (7), and that the solution to (12) yields the solution to (7) (in our case, through (14)). To see that this holds for our problem, consider the image $u = \frac{I_0\mathbb{1}}{n}$. This satisfies all of our equality and inequality constraints (unless an absurd choice for I_0 is made that makes our projection subproblem infeasible, such as some $I_0 \leq 0$ or some $I_0 \geq 255n$). Consequently, the feasible set of our projection subproblem (12) is nonempty, so our problem satisfies Slater's condition. Therefore, since Slater's condition is satisfied for our problem, the solution to the Lagrangian dual problem (16) will yield the solution to our primal problem (13) through the expression (14). We will use an iterative refinement algorithm similar to that of [9] to solve (15). To derive the procedure, we will

first set $\nabla_\lambda g = 0$, $\nabla_\xi g = 0$, and $\frac{\partial g}{\partial \eta} = 0$. This yields the following expressions:

$$\begin{aligned}\lambda &= -z + \psi\mathbb{1},\\ \xi &= z + (-255 - \psi)\mathbb{1},\\ \psi &= \frac{\mathbb{1}^T(z + \lambda - \xi) - I_0}{n}.\end{aligned}$$

However, we have also the inequality constraints (15b). Consequently, we will initialize our algorithm with some initial $\lambda^{(0)}$ and some initial $\xi^{(0)}$, and then we will perform the following steps (where $[\cdot]_+$ denotes the *positive-part* operator):

$$\begin{aligned}\psi^{(i)} &= \frac{\mathbb{1}^T(z + \lambda^{(i-1)} - \xi^{(i-1)}) - I_0}{\mathbb{1}^T\mathbb{1}},\\ \lambda^{(i)} &= [-(z - \psi^{(i)}\mathbb{1})]_+,\\ \xi^{(i)} &= [(z - \psi^{(i)}\mathbb{1}) - 255\mathbb{1}]_+.\end{aligned}$$

Note that certainly $g(\lambda^{(i-1)}, \xi^{(i-1)}, \psi^{(i-1)}) \leq g(\lambda^{(i-1)}, \xi^{(i-1)}, \psi^{(i)})$ for all i, since $\psi = \psi^{(i)}$ maximizes $g(\lambda^{(i-1)}, \xi^{(i-1)}, \psi)$. Also, the above expressions for $\lambda = \lambda^{(i)}$ and $\xi = \xi^{(i)}$ can be regarded as attempts to maximize $g(\lambda, \xi^{(i-1)}, \psi^{(i)})$ and $g(\lambda^{(i)}, \xi, \psi^{(i)})$, respectively. Examination of g reveals that these maximization problems are of the form

$$\max_x \; -\frac{1}{2}x^T x + x^T b \quad \text{subject to } x \geq 0 \tag{16}$$

where $b \in \mathbb{R}$ is given. The solution of (16) *without* the constraint $x \geq 0$ is simply $x = b$. The above terms for $\lambda^{(i)}$ and $\xi^{(i)}$ can then be regarded as finding the vector closest to $x = b$ for which $x \geq 0$, that is, finding

$$\min_x \frac{1}{2}||x - b||_2^2 \quad \text{subject to } x \geq 0.$$

But

$$\frac{1}{2}||x - b||_2^2 = \frac{1}{2}x^T x - x^T b + \frac{1}{2}b^T b$$

and so, dropping the term $\frac{1}{2}b^T b$ that does not depend on x, the above optimization problem reduces to

$$\min_x \frac{1}{2}x^T x - x^T b \quad \text{subject to } x \geq 0.$$

which is equivalent to the optimization problem (16). So, we see that $\lambda = \lambda^{(i)}$ and $\xi = \xi^{(i)}$ maximize $g(\lambda, \xi^{(i-1)}, \psi^{(i)})$ and $g(\lambda^{(i)}, \xi, \psi^{(i)})$ respectively, subject to the constraints $\lambda^{(i)} \geq 0$ and $\xi^{(i)} \geq 0$. Consequently, $g(\lambda^{(i-1)}, \xi^{(i-1)}, \psi^{(i)}) \leq g(\lambda^{(i)}, \xi^{(i-1)}, \psi^{(i)})$ and $g(\lambda^{(i-1)}, \xi^{(i)}, \psi^{(i)}) \leq g(\lambda^{(i)}, \xi^{(i)}, \psi^{(i)})$ for all i. These montonicity results, combined with the fact that the value of $g(\lambda, \xi, \psi)$ for any λ, ξ, and ψ is bounded above by the minimum value of the objective function in the primal problem (13), allows us to conclude that our iterative refinement algorithm is convergent in a general sense. Moreover, we will see in a moment that we are able to employ stopping criteria in our algorithm in such a way that ensures that the output of our algorithm is arbitrarily close to the solution of (16). To select such a robust set of stopping criteria for our algorithm, let us consider the first-order necessary conditions, or so-called *KKT conditions* (see [11]), for u^* to be a solution to (13):

$$u^* = u_d + \lambda - \xi - \psi\mathbb{1}, \tag{17a}$$
$$\mathbb{1}^T u^* - I_0 = 0, \tag{17b}$$
$$-u_j^* \leq 0, \tag{17c}$$
$$u_j^* - 255 \leq 0, \tag{17d}$$
$$\lambda_j \geq 0, \tag{17e}$$
$$\xi_j \geq 0, \tag{17f}$$
$$\lambda_j u_j^* = 0, \tag{17g}$$
$$\xi_j(u_j^* - 255) = 0. \tag{17h}$$

where $j = 1, 2, ..., n$. Note that (17a), (17e) and (17f) are satisfied at each iterate of our algorithm. The conditions (17c) and (17d) are also satisfied at each iterate of our algorithm. To see this, let us first consider the bound constraints (12b) and (12c). We see that they will be satisfied at each iterate within the projection algorithm:

$$u^{(i)} = z - \psi^{(i)}\mathbb{1} + \lambda^{(i)} - \xi^{(i)} = (z - \psi^{(i)}\mathbb{1}) + [-(z - \psi^{(i)}\mathbb{1})]_+ - [(z - \psi^{(i)}\mathbb{1}) - 255\mathbb{1}]_+.$$

It is readily seen from this last expression that the net effect of our procedure is to set $u^{(i)}$ equal to the parts of $z - \psi^{(i)}\mathbb{1}$ that are greater than zero and less than 255. Therefore, the conditions (17c) and (17d) are satisfied at each iterate. Regarding the condition (17b), we can ensure that it is satisfied at the kth iterate by taking as a stopping criterion that

$$\frac{|\mathbb{1}^T u^{(k)} - I_0|}{I_0}$$

is less than or equal to a user-specified tolerance (we chose 10^{-4}). Regarding (17g) and (17h), since we know that (17c), (17d), (17e), and (17f) hold at each iterate, we know that for $j = 1, 2, ..., n$ and at each iterate i,

$$-\lambda_j^{(i)} u_j^{(i)} \leq 0,$$
$$\xi_j^{(i)}(u_j^{(i)} - 255) \leq 0.$$

So, certainly

$$-\lambda_j^{(i)} u_j^{(i)} + \xi_j^{(i)}(u_j^{(i)} - 255) \leq 0,$$

and consequently (17g) and (17h) are satisfied at the kth iterate if

$$-(\lambda^{(k)})^T u^{(k)} + (\xi^{(k)})^T(u^{(k)} - 255\mathbb{1}) = 0.$$

By considering (13), (14), (15a), and the fact that we are already enforcing (17b) in our algorithm, we see that the above condition amounts to

$$\frac{|\frac{1}{2}||u_d - u^{(k)}||_2^2 - g(\lambda^{(k)}, \xi^{(k)}, \psi^{(k)})|}{\frac{1}{2}||u_d - u^{(k)}||_2^2}$$

being less than some user-specified tolerance at the kth iterate. We have chosen to enforce this as an additional stopping criterion (with 10^{-4} as our tolerance). Consequently, with our choice of stopping criteria, our algorithm generates at the kth iterate estimates of the primal and dual variables that satisfy the first-order necessary conditions (17a)-(17h) to within user-specified tolerances. To see that $u^{(k)}$ is a solution to (13), we note that since the Hessian with respect to u of the Lagrangian (13) is a multiple of the identity matrix, it is positive definite everywhere. Consequently, the image $u^{(k)}$ found by our algorithm satisfies the *second-order sufficient conditions* (see [11]) to be a solution to (13). In fact, since (12a) is convex everywhere, $u^{(k)}$ is in fact the unique solution to (13), to within user-specified tolerances.

On a technical note, after we solve each projection subproblem, we retain the values of the Lagrange multiplers λ, ξ, and ψ so that we can use them as our initializations for the Lagrange multipliers at the beginning of the next projection subproblem. Consequently, in the following experiments, it typically takes only two or three iterations of our projection algorithm before our stopping criteria are satisfied.

7 Numerical Results

We will use FISTA combined with the projection algorithm described above to solve one denoising problem and one deblurring problem. We will have three results for each of these problems. First, we will solve the problem with only the bound constraints on the pixels. Next, we will take the results from the bound-constrained solution and project it onto the intensity constraint to see if an improvement is yielded. (By "the intensity constraint", we mean $||\hat{u}||_1 = I_0 = ||u||_1$, or in other words, that our reconstructed images have the same intensity as the original, uncorrupted image.) Finally, we will impose the intensity constraint at each denoising step that is taken by FISTA. For both experiments, we will use the standard "cameraman" test image (see Figure 2a). For each experiment, the τ values on the horizontal axis of each graph are the values of the regularization parameter, and the "relative error" on the vertical access is $\frac{||\hat{u}(\tau)-u||_2}{||u||_2}$, where $\hat{u}(\tau)$ is our estimate of the true image that corresponds to a particular value of τ, and u is the original, true cameraman image. We choose values of τ ranging from $\frac{1}{2}$ to 40 in increments of $\frac{1}{2}$. We will first display the corrupted image, and the images corresponding to the best values of the relative error $\frac{||u(\tau)-u||_2}{||u||_2}$ for each of the three cases. Then, we will display a graph of $\frac{||u(\tau)-u||_2}{||u||_2}$ against τ for each of the three cases. We will use the isotropic total variation (7) in this section - numerical results using (8) are similar.

a. Denoising. First we will perform a denoising experiment, where our cameraman test image has been corrupted with zero-mean Gaussian noise with standard deviation 30. We have chosen to stop iterating when the relative change in the image is less than 10^{-4}. The optimal τ value in all cases was $\tau = 26$. The images resulting from regularizing at the optimal τ value can be seen in Figure 2. We see that the image reconstructions where the intensity constraint (11d) is imposed are not significantly different than the image reconstruction where we do not impose (11d). Also, if we examine the relative error of each of our estimates as a function of τ (Figure 3), we see that the relative error for any given value of τ is not meaningfully different when we impose the intensity constraint (11d). All of these results would lead us to conclude that the intensity constraint (11d) does not seem to meaningfully improve the results of total variation denoising. Consequently, we can perform total variation denoising more quickly by not imposing (11d) and still achieve a very similar result. In fact, solving (11a) for the optimal τ with only the bound constraints (11b) and (11c) only takes 33 seconds on our machine (a Dell Inspiron 1545 laptop with a 2GHz dual-core processor and 4GB of RAM), while solving (11a) with (11b), (11c), and the intensity constraint

(a) **(b)**

(c) **(d)** **(e)**

Figure 2: **(a)** The cameraman test image. **(b)** The noisy cameraman image (relative error was .2250963). **(c)** Result when regularizing with bound constraints (relative error was .0616943). **(d)** Result of projecting (c) onto the intensity constraint (relative error was .0616930). **(e)** Result when regularizing with bound and intensity constraints (relative error was .0616930). Note that the image reconstructions (c), (d), and (e) look nearly identical.

(11d) for the same τ value takes 51 seconds.

b. Deblurring. Typically, blurring is modeled in digital image processing as the *convolution* of an image with a point-spread function (PSF), which represents the type of blurring being modeled. Typically, the point-spread function f is of odd size $m \times n$ with $m = 2a + 1$ and $n = 2b + 1$, and is much smaller than the image w with which it will be convolved (see [7]). In such a case, the elements of the convolution $f * w$ are given by

$$(f * w)_{i,j} = \sum_{s=-a}^{s=a} \sum_{t=-b}^{t=b} f_{s,t} w_{i-s,j-t}.$$

where i and j range over all of the pixels in the image w. For this experiment, we will convolve our cameraman test image with the "out-of-focus" blur kernel from [8] which has a point-spread function p where each element is given by

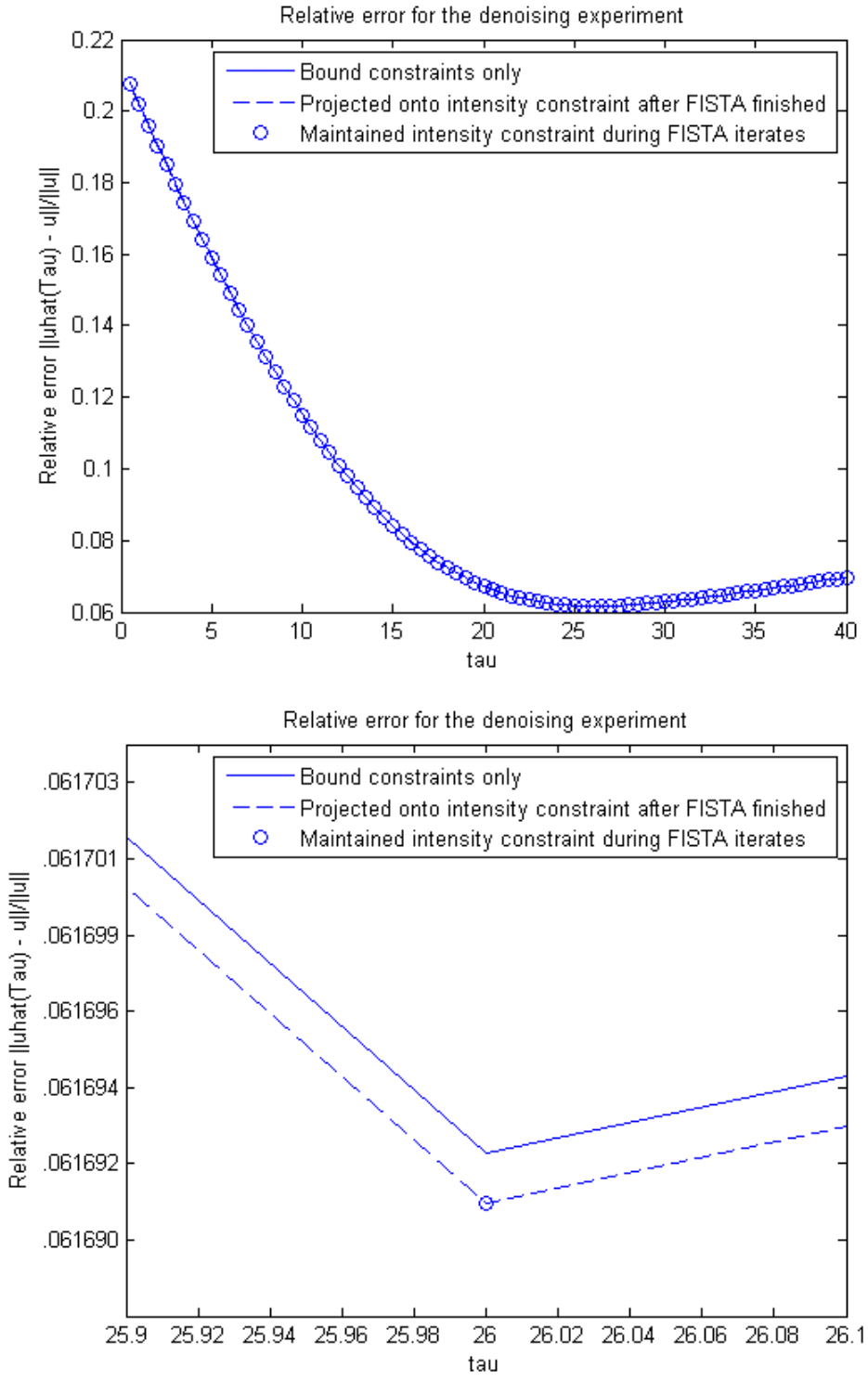


Figure 3: Relative error of our estimates as a function of τ. At the larger scale (top graph), the three curves are indistinguishable, and if we zoom in on any particular part of the graph, we see (bottom graph) that the estimates involving the intensity constraint yield only very slightly less error than that involving only the bound constraints.

$$p_{i,j} = \begin{cases} \frac{1}{s} & \text{if } (i-k)^2 + (j-l)^2 \leq r^2, \\ 0 & \text{elsewhere,} \end{cases} \tag{18}$$

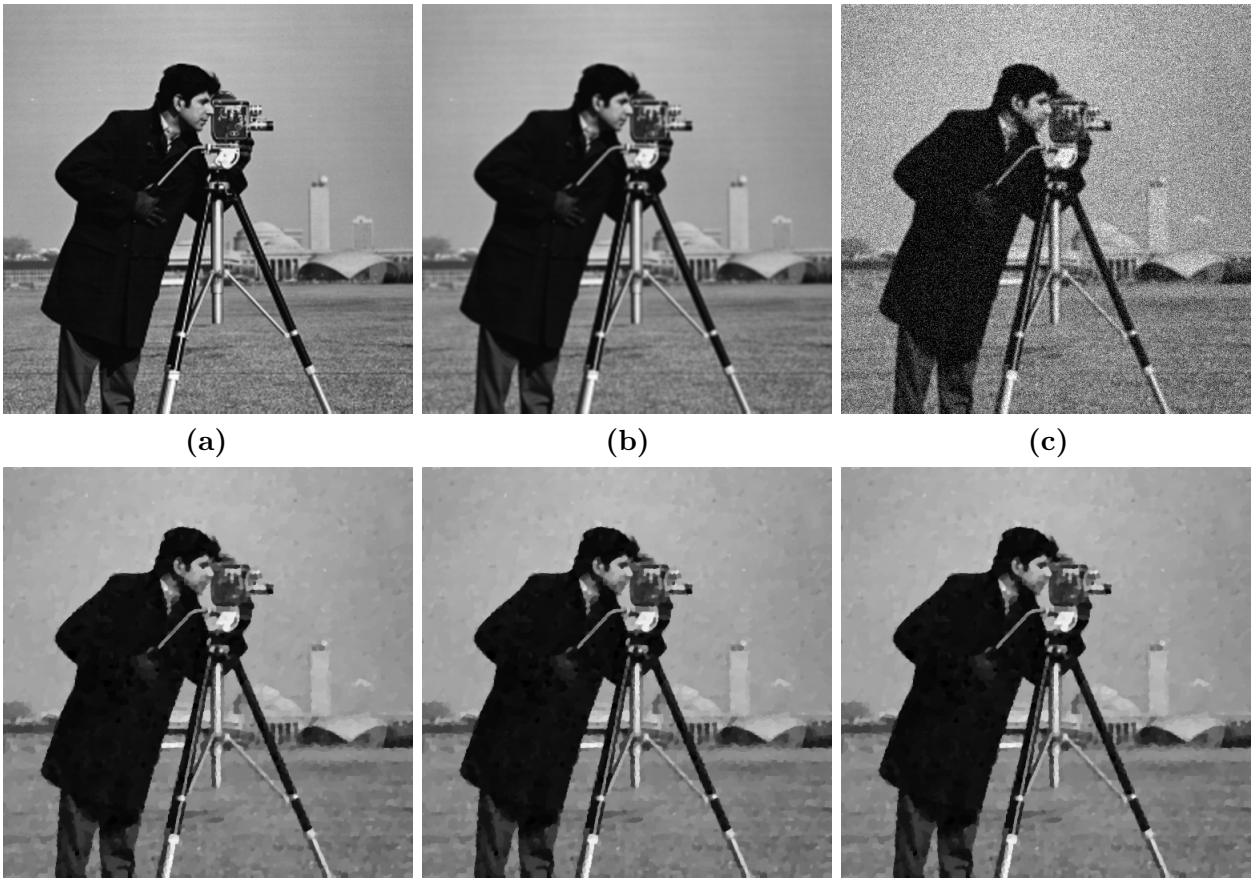
where (k, l) is the center of the point-spread function, r is the radius of the blur, and s is the number of non-zero elements in the point-spread function. We will use this PSF with $r = 2$ and $(k, l) = (3, 3)$, so that we have a symmetric PSF of size 5×5, that is,

$$p = \frac{1}{13} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

So the matrix-vector product Au corresponds to the convolution of the PSF p with the image u, namely, the convolution $p * u$ has elements given by

$$(p * u)_{i,j} = \sum_{s=-2}^{s=2} \sum_{t=-2}^{t=2} p_{s,t} u_{i-s,j-t}. \tag{19}$$

where i and j range over all of the pixels in u. (On a technical note, we need to extend the image in some way to account for what happens when the term $u_{i-s,j-t}$ contains the values of pixels from "outside the image." We have chosen to adopt (standard) "reflexive" boundary conditions, which amounts to mirror-reflecting the image across each of its boundaries before computing (19) - see [8].) In addition to blurring u, we will also add zero-mean Gaussian noise η of standard deviation 30 to the resulting blurred image Au, so that our problem $y = Au + \eta$ now corresponds to estimating our original test image from a noisy, blurry image y (see Figure 4b). Regarding our implementation of deblurring, we will again use FISTA to perform total variation regularization to estimate u, taking values of τ ranging from $\frac{1}{2}$ to 40 in increments of $\frac{1}{2}$. We have again chosen to stop iterating when the relative change in the image is less than 10^{-4}, and we are solving the denoising subproblems using the same stopping criterion. The optimal τ value was $\tau = 13.5$ in all three cases. The images resulting from regularizing at the optimal τ value can be seen in Figure 4. As with the denoising experiment, we see that the image reconstructions where the intensity constraint (11d) is imposed are not significantly different than the image reconstruction where we do not impose (11d). Again, we also see that if we examine the relative error of each of our estimates as a function of τ (Figure 5), the relative error for any given value of τ is not meaningfully different when we impose the intensity constraint (11d). All of this would lead us to conclude that the intensity constraint (11d) does not seem to meaningfully improve the results of total variation deblurring either. Consequently, as with total variation denoising, we can perform total variation deblurring more quickly by not imposing (11d) and still achieve a very similar result. In this case, solving (11a) for the optimal τ with only the bound constraints (11b) and (11c) only takes 174 seconds on our machine, while solving (11a) with (11b), (11c), and the intensity constraint (11d) for the same τ value takes 250 seconds.



(a) **(b)** **(c)**

(d) **(e)** **(f)**

Figure 4: **(a)** The cameraman test image. **(b)** The blurred cameraman test image yielded by performing the convolution (19) on (a). **(c)** The blurred, noisy cameraman image (relative error was .2294614). **(d)** Result when regularizing with bound constraints (relative error was .0716356). **(e)** Result of projecting **(b)** onto the intensity constraint (relative error was .0716343). **(f)** Result when regularizing with bound and intensity constraints (relative error was .0716343). Again, the image reconstructions (d), (e), and (f) look nearly identical.

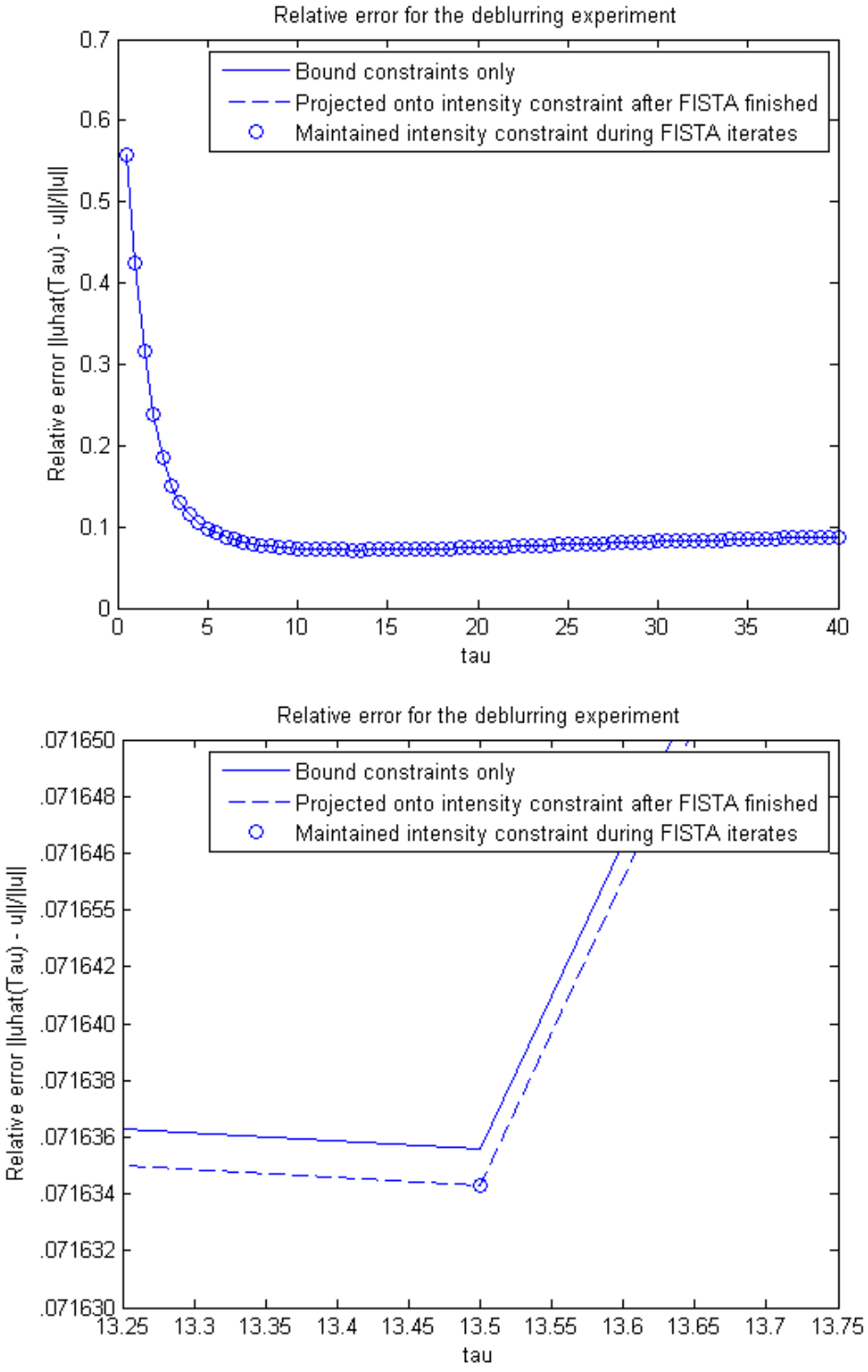


Figure 5: Relative error of our estimates as a function of τ. Again, at the larger scale (top graph), the three curves are indistinguishable, and if we zoom in on any particular part of the graph, we again see (bottom graph) that the estimates involving the intensity constraint yield only very nominally less error than that involving only the bound constraints.

8 Analysis of Numerical Results

To analyze our numerical results, let us consider that one way of solving the problem (9) (letting $f(u) = \frac{1}{2}||y - Au||_2^2 + \tau TV(u)$) is by the following "gradient descent" method:

Initialize with some $u = u^{(0)}$.
do "until converged":
$u^{(k+1)} = u^{(k)} - \alpha \nabla f(u^{(k)}) = u^{(k)} - \alpha(-A^T y + A^T A u^{(k)} + \tau \nabla TV(u^{(k)})$
end do

where $\alpha > 0$ is a scalar chosen at each iterate to satisfy certain conditions that ensure that the iterative scheme above yields the solution to (9) (for examples of such conditions, see [11]).
In appendix A, we prove that for any u,

$$\mathbb{1}^T(\nabla TV(u)) = \sum_{i,j}(\nabla TV(u))_{i,j} = 0. \tag{20}$$

Denoting the solution to (9) by u^*, we will use (20) to prove that if the conditions

$$A = A^T, \tag{21a}$$
$$\mathbb{1}^T A u = \mathbb{1}^T u \ \forall u \tag{21b}$$

are satisfied for our linear operator A, then we have

$$\mathbb{1}^T u^* = \mathbb{1}^T y. \tag{22}$$

In the denoising experiment, $A = I$, so certainly (21a) and (21b) hold. Also, since the Gaussian noise is zero-mean, we have that

$$\begin{aligned} \mathbb{1}^T y &= \mathbb{1}^T A u + \mathbb{1}^T \eta \\ &= \mathbb{1}^T A u + 0 \\ &= \mathbb{1}^T u. \end{aligned}$$

As we discuss in appendix B, our linear operator A in the deblurring experiment also has the properties (21a) and (21b) if we adopt periodic or reflexive boundary conditions. Now to prove (22), suppose that we are solving (9) with A satisfying (21a) and (21b), and that we initialize our algorithm with $u_0 = y$. The first iteration looks like

$$\begin{aligned} u^{(1)} &= u^{(0)} - \alpha(-A^T y + A^T A u^{(0)} + \tau \nabla TV(u^{(0)}) \\ &= A^T y - \alpha(-A^T y + A^T A y + \nabla TV(y)) \end{aligned}$$

and consequently

$$\begin{aligned}
\mathbb{1}^T u^{(1)} &= \mathbb{1}^T A^T y - \alpha(-\mathbb{1}^T A^T y + \mathbb{1}^T A^T A y + \mathbb{1}^T T V(y)) \\
&= \mathbb{1}^T y + \alpha \mathbb{1}^T y - \alpha \mathbb{1}^T A y + 0 \\
&= \mathbb{1}^T y + \alpha \mathbb{1}^T y - \alpha \mathbb{1}^T y \\
&= \mathbb{1}^T y
\end{aligned}$$

because of (20), (21a) and (21b). This forms the *base case* for induction. Now we suppose that $\mathbb{1}^T u^{(k)} = \mathbb{1}^T y$ at some iteration k, and we consider

$$\begin{aligned}
\mathbb{1}^T u^{(k+1)} &= \mathbb{1}^T u^{(k)} - \alpha \mathbb{1}^T(-A^T y + u^{(k)} + \tau \nabla T V(u^{(k)}) \\
&= \mathbb{1}^T y + \alpha \mathbb{1}^T A^T y - \alpha \mathbb{1}^T u^{(k)} + \alpha \tau \mathbb{1}^T \nabla T V(u^{(k)}) \\
&= \mathbb{1}^T y + \alpha \mathbb{1}^T y - \alpha \mathbb{1}^T y + 0 \\
&= \mathbb{1}^T y.
\end{aligned}$$

by our induction hypothesis and because of (20), (21a) and (21b). So, by induction, we have shown that $\mathbb{1}^T u^{(k)} = \mathbb{1}^T y$ for any k and any values of τ and α. We emphasize that, although the algorithm that we are using from [1] to solve (9) does not use the above gradient descent algorithm, the above reasoning still leads us to the conclusion that the solution to (9) has the property (22) for any value of τ in cases where (21a) and (21b) are true, regardless of how the solution to (9) was obtained. Moreover, in these cases, this property should still be observed even if the algorithm is initialized with an initialization other than $u^{(0)} = y$.

Now, the problem (9) is our problem (12) without the constraints (11b), (11c), (11d). The above analysis does not hold exactly if we impose the constraints (11b) and (11c), but it does suggest that the constraint (11d) will *almost* hold in cases where (21a) and (21b) are true, even if we do not impose (11d) explicitly. In fact, if we consult a graph of the sums of the pixel values of the reconstructed images in the denoising problem, first solving the unconstrained problem (9), and then solving (11a) with the bound constraints (11b) and (11c), we see the following:

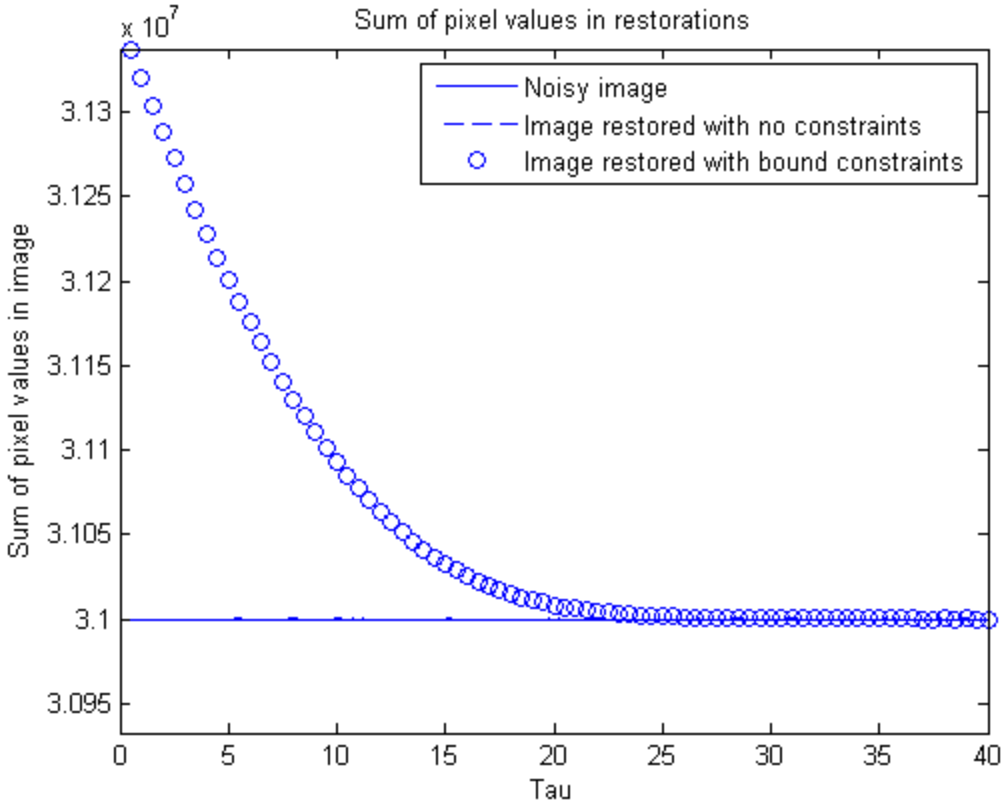


Figure 6: A graph of the sum of the pixel values of the reconstructions as a function of τ. The dashed line denoting the result of the unconstrained minimization problem is indistinguishable from the sum of the pixel values of the noisy image. The result involving bound constraints at first only has approximately the desired sum of pixel values, but as τ increases, the bound constraints tend to be satisfied automatically by the total variation regularization, and the sum of pixel values is as desired.

This would seem to confirm our expectations that the constraint (11d) holds exactly if we solve the unconstrained problem (9), and that it will almost hold in cases where (21a) and (21b) are true, even if we do not impose (11d) explicitly.

If we examine the analogous results for the deblurring problem, we see the following:

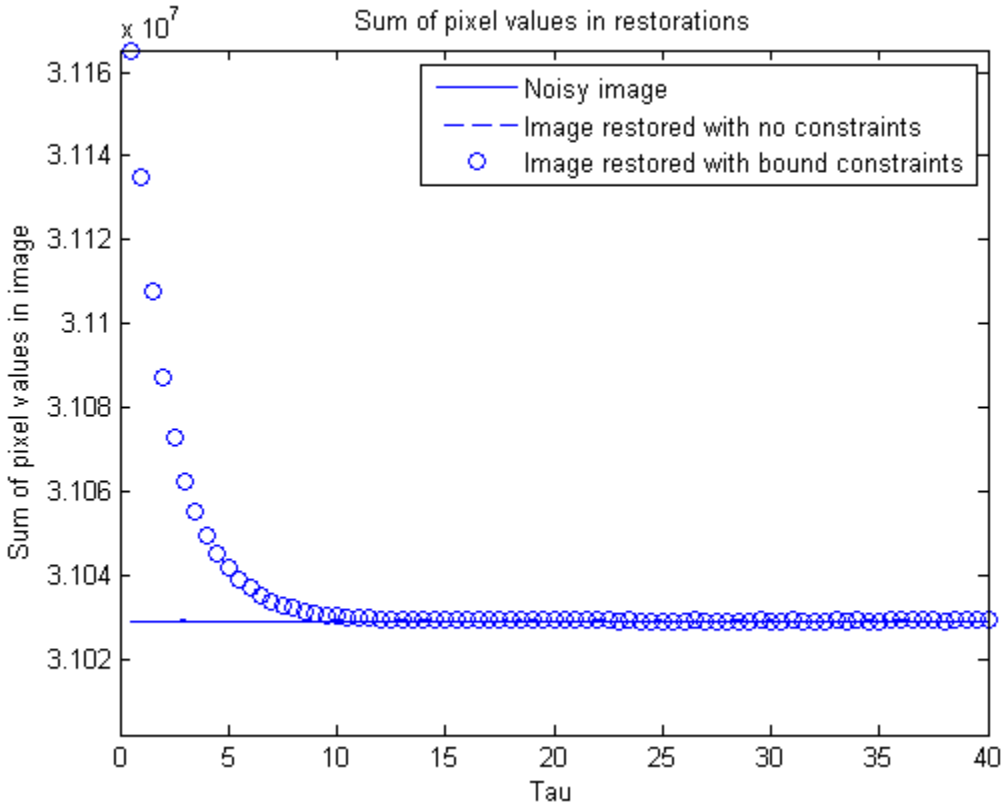


Figure 7: A graph of the sum of the pixel values of the reconstructions as a function of τ. Again, the dashed line denoting the result of the unconstrained minimization problem is indistinguishable from the sum of the pixel values of the noisy image. Again, the result involving bound constraints at first only has approximately the desired sum of pixel values, but as τ increases, the bound constraints tend to be satisfied automatically by the total variation regularization, and the sum of pixel values is as desired.

This is consistent with the conclusions that we drew previously from examining the corresponding graph from the denoising results. This would explain why the intensity constraint (11d) does not improve the results of total variation denoising or deblurring. However, we emphasize that the property (22) should not be expected for general linear operators A. Our analysis only shows that it holds for A for which (21a) and (21b) are true.

9 Conclusions and Future Work

We have shown both analytically and empirically that total variation regularization preserves the intensity of the original image u in denoising and deblurring problems. As a result, the intensity constraint (11d) does not improve the results of total variation regularization in these cases. This is fortunate, since these problems can be solved more quickly if (11d) does not need to be enforced. Potential future work includes assessing whether the projection algorithm that solves (12) could be used to improve the results of image enhancement techniques such as image sharpening.

References

[1] A. Beck and M. Teboulle, *Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems*, IEEE Trans. Image Process., 18 (2009), pp. 2419–2434.

[2] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, 2004.

[3] E. Candes and J. Romberg, *l1-magic: Recovery of sparse signals via convex programming*, October 2005. http://www.acm.caltech.edu/l1magic/downloads/l1magic.pdf.

[4] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM Journal on Scientific Computing, 20 (1998). http://dx.doi.org/10.1137/S1064827596304010.

[5] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, 2010.

[6] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, *Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems*, IEEE Journal of Selected Topics in Signal Processing: Special Issue on Convex Optimization Methods for Signal Processing, September 12, 2007.

[7] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Pearson Prentice Hall, 3rd ed., 2008.

[8] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring images*, vol. 3 of Fundamentals of Algorithms, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006. Matrices, spectra, and filtering.

[9] Z. Harmany, D. Thompson, M. Willett, and R. Marcia, *Gradient projection for linearly constrained convex optimization in sparse signal recovery*, IEEE International Conference on Image Processing, 2010.

[10] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 3rd ed., 2009.

[11] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer Series in Operations Research, Springer-Verlag, New York, 1999.

[12] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, (1992).

[13] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellsley-Cambridge Press, 2nd ed., 1997.

[14] E. van den Berg and M. P. Friedlander, *An introduction to compressive sampling*, IEEE Signal Processing Magazine, March 2008.

[15] E. van den Berg and M. P. Friedlander, *SPGL1: A solver for large-scale sparse reconstruction*, June 2007. http://www.cs.ubc.ca/labs/scl/spgl1.

Appendix A

Theorem. Let the total variation $TV(u)$ of a $p \times q$ digital image be defined by (7). Let us define $\nabla TV(u) : \mathbb{R}^{mn} \to \mathbb{R}^{mn}$ as having entries given by

$$(\nabla TV(u))_{i,j} = \frac{\partial}{\partial u_{i,j}}(TV(u)).$$

Then

$$\mathbb{1}^T(\nabla TV(u)) = \sum_{i,j}(\nabla TV(u))_{i,j} = 0$$

for any image u.

Proof. Let us consider the 3×3 digital image

$$u = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} \\ u_{2,1} & u_{2,2} & u_{2,3} \\ u_{3,1} & u_{3,2} & u_{3,3} \end{bmatrix}.$$

The isotropic total variation (7) for the above digital image u is

$$\begin{aligned} TV(u) = &\sqrt{(u_{1,1}-u_{1,2})^2 + (u_{1,1}-u_{2,1})^2} + \sqrt{(u_{1,2}-u_{1,3})^2 + (u_{1,2}-u_{2,2})^2} + |u_{1,3}-u_{2,3}| \\ &+ \sqrt{(u_{2,1}-u_{2,2})^2 + (u_{2,1}-u_{3,1})^2} + \sqrt{(u_{2,2}-u_{2,3})^2 + (u_{2,2}-u_{3,2})^2} + |u_{2,3}-u_{3,3}| \\ &+ |u_{3,1}-u_{3,2}| + |u_{3,2}-u_{3,3}| \end{aligned}$$

where the terms in absolute value that only involve the edge pixels come from the conditions for $i = m$ and $j = n$ in (7), and the fact that $\sqrt{a^2} = |a|$ for any real number a. Now if we compute all of the terms in $\nabla TV(u)$, we get

$$
\begin{aligned}
(\nabla TV(u))_{1,1} =& \frac{(u_{1,1}-u_{1,2})+(u_{1,1}-u_{2,1})}{\sqrt{(u_{1,1}-u_{1,2})^2+(u_{1,1}-u_{2,1})^2}},\\
(\nabla TV(u))_{2,1} =& \frac{(u_{2,1}-u_{2,2})+(u_{2,1}-u_{3,1})}{\sqrt{(u_{2,1}-u_{2,2})^2+(u_{2,1}-u_{3,1})^2}} - \frac{(u_{1,1}-u_{2,1})}{\sqrt{(u_{1,1}-u_{1,2})^2+(u_{1,1}-u_{2,1})^2}},\\
(\nabla TV(u))_{3,1} =& -\frac{(u_{2,1}-u_{3,1})}{\sqrt{(u_{2,1}-u_{2,2})^2+(u_{2,1}-u_{3,1})^2}} + \operatorname{sgn}(u_{3,1}-u_{3,2}),\\
(\nabla TV(u))_{1,2} =& -\frac{(u_{1,1}-u_{1,2})}{\sqrt{(u_{1,1}-u_{1,2})^2+(u_{1,1}-u_{2,1})^2}} + \frac{(u_{1,2}-u_{1,3})+(u_{1,2}-u_{2,2})}{\sqrt{(u_{1,2}-u_{1,3})^2+(u_{1,2}-u_{2,2})^2}},\\
(\nabla TV(u))_{2,2} =& -\frac{(u_{1,2}-u_{2,2})}{\sqrt{(u_{1,2}-u_{1,3})^2+(u_{1,2}-u_{2,2})^2}} - \frac{(u_{2,1}-u_{2,2})}{\sqrt{(u_{2,1}-u_{2,2})^2+(u_{2,1}-u_{3,1})^2}},\\
&+ \frac{(u_{2,2}-u_{2,3})+(u_{2,2}-u_{3,2})}{\sqrt{(u_{2,2}-u_{2,3})^2+(u_{2,2}-u_{3,2})^2}},\\
(\nabla TV(u))_{3,2} =& -\frac{(u_{2,2}-u_{3,2})}{\sqrt{(u_{2,2}-u_{2,3})^2+(u_{2,2}-u_{3,2})^2}} - \operatorname{sgn}(u_{3,1}-u_{3,2}) + \operatorname{sgn}(u_{3,2}-u_{3,3}),\\
(\nabla TV(u))_{1,3} =& -\frac{(u_{1,2}-u_{1,3})}{\sqrt{(u_{1,2}-u_{1,3})^2+(u_{1,2}-u_{2,2})^2}} + \operatorname{sgn}(u_{1,3}-u_{2,3}),\\
(\nabla TV(u))_{2,3} =& -\operatorname{sgn}(u_{1,3}-u_{2,3}) - \frac{(u_{2,2}-u_{2,3})}{\sqrt{(u_{2,2}-u_{2,3})^2+(u_{2,2}-u_{3,2})^2}} + \operatorname{sgn}(u_{2,3}-u_{3,3}),\\
(\nabla TV(u))_{3,3} =& -\operatorname{sgn}(u_{2,3}-u_{3,3}) - \operatorname{sgn}(u_{3,2}-u_{3,3})
\end{aligned}
$$

where

$$
\operatorname{sgn}(x) = \begin{cases} 1 & \text{if } x > 0, \\ -1 & \text{if } x < 0, \end{cases}
$$

and we take $\operatorname{sgn}(0) = 0$. Our claim is that

$$
\mathbb{1}^T(\nabla TV(u)) = \sum_{i,j}(TV(u))_{i,j} = 0.
$$

To see this, let us first consider terms in $(\nabla TV(u))$ that involve $u_{1,1}$. We see that only $(\nabla TV(u))_{1,1}$, $(\nabla TV(u))_{2,1}$, and $(\nabla TV(u))_{1,2}$ contain terms that involve $u_{1,1}$, and that when we compute $(\nabla TV(u))_{1,1}+(\nabla TV(u))_{2,1}+(\nabla TV(u))_{1,2}$, the terms containing $u_{1,1}$ are

$$
\frac{2u_{1,1}}{\sqrt{(u_{1,1}-u_{1,2})^2+(u_{1,1}-u_{2,1})^2}} - \frac{(u_{1,1})}{\sqrt{(u_{1,1}-u_{1,2})^2+(u_{1,1}-u_{2,1})^2}} - \frac{(u_{1,1})}{\sqrt{(u_{1,1}-u_{1,2})^2+(u_{1,1}-u_{2,1})^2}} = 0.
$$

Clearly this is true regardless of whether the dimensions of u are 3×3 or $m \times n$. So, all terms involving $u_{1,1}$ vanish. The same thing happens to terms involving $u_{1,3}$, $u_{3,1}$, and $u_{3,3}$, with the computation looking essentially the same as that for terms involving $u_{1,1}$. That is, the terms from $(\nabla TV(u))_{1,3}$ that involve $u_{1,3}$ cancel with those in $(\nabla TV(u))_{1,2}$ and $(\nabla TV(u))_{2,3}$ that contain

$u_{1,3}$ (and no other entries in $\nabla TV(u)$ contain terms with $u_{1,3}$ in them), and so forth. Clearly the computations would be similar for an image of size $m \times n$ rather than 3×3, with $(\nabla TV(u))_{1,3}$ being replaced by $(\nabla TV(u))_{1,n}$, $(\nabla TV(u))_{3,3}$ being replaced by $(\nabla TV(u))_{m,n}$, and so forth. Now if we consider all terms in $\mathbb{1}^T(\nabla TV(u))$ that involve $u_{2,2}$, we see that these are the terms from $(\nabla TV(u))_{2,2}$, $(\nabla TV(u))_{2,1}$, $(\nabla TV(u))_{1,2}$, $(\nabla TV(u))_{3,2}$, and $(\nabla TV(u))_{2,3}$ - namely, the pixels in $\nabla TV(u)$ that are adjacent to $(\nabla TV(u))_{2,2}$, including $(\nabla TV(u))_{2,2}$ itself. The addition of these terms yields

$$\frac{4u_{2,2}}{\sqrt{(u_{1,2}-u_{1,3})^2}} - \frac{u_{2,2}}{\sqrt{(u_{1,2}-u_{1,3})^2}} - \frac{u_{2,2}}{\sqrt{(u_{1,2}-u_{1,3})^2}} - \frac{u_{2,2}}{\sqrt{(u_{1,2}-u_{1,3})^2}} - \frac{u_{2,2}}{\sqrt{(u_{1,2}-u_{1,3})^2}} = 0.$$

For any interior pixel $u_{i,j}$ in an $m \times n$ digital image, the addition of all terms involving $u_{i,j}$ in $\mathbb{1}^T(\nabla TV(u))$ looks the same as the addition of the terms involving $u_{2,2}$ for $\nabla TV(u)$ for our 3×3 image, and also sums to zero. If we now look at all the terms in $\mathbb{1}^T(\nabla TV(u))$ that involve $u_{2,1}$, we see that only the terms in $(\nabla TV(u))_{2,1}$, $(\nabla TV(u))_{2,2}$, $(\nabla TV(u))_{1,1}$, and $(\nabla TV(u))_{3,1}$ involve $u_{2,1}$, and that adding these terms yields

$$\frac{3u_{2,1}}{\sqrt{(u_{2,1}-u_{2,2})^2+(u_{2,1}-u_{3,1})^2}} - \frac{3u_{2,1}}{\sqrt{(u_{2,1}-u_{2,2})^2+(u_{2,1}-u_{3,1})^2}} = 0.$$

For any pixel $u_{i,j}$ in an $m \times n$ digital image that is on a boundary, but that is not a corner pixel ($u_{1,1}$, $u_{m,1}$, $u_{1,n}$, or $u_{m,n}$), the addition of all terms involving $u_{i,j}$ in $\mathbb{1}^T(\nabla TV(u))$ looks the same as the addition of the terms involving $u_{2,1}$ (or $u_{1,2}$ or $u_{2,3}$ or $u_{3,2}$) for $\nabla TV(u)$ for our 3×3 image, and also sums to zero. This covers all of the terms in $\nabla TV(u)$, giving us the desired result,

$$\mathbb{1}^T(\nabla TV(u)) = \sum_{i,j}(TV(u))_{i,j} = 0,$$

for u of arbitrary size in addition to our 3×3 example. Similar computations show that we can also say that $\mathbb{1}^T(\nabla TV_{\ell_1}(u))$ for the anisotropic total variation (8) (if we adopt the convention $\text{sgn}(0) = 0$ as we did above).

There is also a corollary to the above theorem which is of considerable practical importance. The above definition of $\nabla TV(u)$ involves division by zero if the image u contains regions where the pixels' intensity values do not change. (The anisotropic total variation (8), which does not have this problem, does not need such an ϵ to be introduced.) In practice, using a slightly regularized total variation functional $TV(u;\epsilon)$ (below) circumvents this problem. The proof of the corollary is extremely similar to that of the theorem.

Corollary. If we define

$$TV(u;\epsilon) = \sum_{i,j}(\sqrt{(D_{h;ij}u)^2 + (D_{v;ij}u)^2 + \epsilon^2} - \epsilon),$$

and we define $\nabla TV(u;\epsilon) : \mathbb{R}^{mn} \to \mathbb{R}^{mn}$ as having entries given by

$$(\nabla TV(u;\epsilon))_{i,j} = \frac{\partial}{\partial u_{i,j}}(TV(u;\epsilon)),$$

then

$$\mathbb{1}^T(\nabla TV(u;\epsilon)) = \sum_{i,j}(\nabla TV(u;\epsilon))_{i,j} = 0$$

for any image u and any $\epsilon > 0$.

Appendix B

In this section, we argue that our convolution $Pu = p * u$ described by (19) has the property that, if periodic or reflexive boundary conditions (see [8]) are adopted for the convolution, then for any image u,

$$Pu = P^T u,$$

and

$$\mathbb{1}^T Pu = \sum_{i,j} (Pu)_{i,j} = \sum_{i,j} u_{i,j} = \mathbb{1}^T u.$$

To give intuition regarding why the above are true, let us consider the convolution of a one-dimensional signal $x \in \mathbb{R}^8$ and the 5×1 filter $p = \frac{1}{5}[1\ 1\ 1\ 1\ 1]^T$, which is a one-dimensional version of our point-spread function (18). We can represent the convolution of p and x as a matrix-vector product Px. If we adopt reflexive boundary conditions, then the matrix P is

$$P = \frac{1}{5}\begin{bmatrix} 2 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 2 \end{bmatrix}.$$

The sum along each column is 1, signifying that $\mathbb{1}^T Px = \mathbb{1}^T x$. If we instead adopt periodic boundary conditions, we see that

$$P = \frac{1}{5}\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

The sum along each column is again 1, signifying that again $\mathbb{1}^T Px = \mathbb{1}^T x$. Note also that $P = P^T$ for each of the above matrices. For digital images, the computations in the convolution (19) are more involved, so we will use an example with the 3×3 version of the point-spread function (18), namely,

$$p = \frac{1}{5}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

In §4.3 of [8], the authors state that any two-dimensional point-spread function that exhibits horizontal and vertical symmetry (as ours does) will lead to a matrix representation P that satisfies $P = P^T$. Therefore, we will focus on showing that P satisfies $\mathbb{1}^T Pu = \mathbb{1}^T u$. To see this, let us consider a 3×3 image

$$u = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} \\ u_{2,1} & u_{2,2} & u_{2,3} \\ u_{3,1} & u_{3,2} & u_{3,3} \end{bmatrix}.$$

and its 5×5 reflexive extension

$$u_r = \begin{bmatrix} u_{1,1} & u_{1,1} & u_{1,2} & u_{1,3} & u_{1,3} \\ u_{1,1} & u_{1,1} & u_{1,2} & u_{1,3} & u_{1,3} \\ u_{2,1} & u_{2,1} & u_{2,2} & u_{2,3} & u_{2,3} \\ u_{3,1} & u_{3,1} & u_{3,2} & u_{3,3} & u_{3,3} \\ u_{3,1} & u_{3,1} & u_{3,2} & u_{3,3} & u_{3,3} \end{bmatrix}.$$

The convolution $Pu = p * u$ is then given by convolving p with the 3×3 submatrix consisting of the interior pixels of u_r. Specifically, we have that

$$Pu = \frac{1}{5}\begin{bmatrix} 3u_{1,1} + u_{1,2} + u_{2,1} & 2u_{1,2} + u_{1,1} + u_{1,3} + u_{2,2} & 3u_{1,3} + u_{1,2} + u_{2,3} \\ 2u_{2,1} + u_{1,1} + u_{2,2} + u_{3,1} & u_{1,2} + u_{2,1} + u_{2,2} + u_{2,3} + u_{3,2} & u_{1,3} + u_{2,2} + 2u_{2,3} + u_{3,3} \\ 3u_{3,1} + u_{2,1} + u_{3,2} & u_{2,2} + u_{3,1} + 2u_{3,2} + u_{3,3} & u_{2,3} + u_{3,2} + 3u_{3,3} \end{bmatrix}.$$

If we compute

$$\mathbb{1}^T Pu = \sum_{i=1}^{3}\sum_{j=1}^{3}(Pu)_{i,j},$$

we see that

$$\sum_{i=1}^{3}\sum_{j=1}^{3}(Pu)_{i,j} = \sum_{i=1}^{3}\sum_{j=1}^{3}u_{i,j} = \mathbb{1}^T u.$$

If we examine $(Pu)_{2,2}$ and its adjacent entries $(Pu)_{1,2}$, $(Pu)_{2,1}$, $(Pu)_{2,3}$, and $(Pu)_{3,2}$, we see that they each contain $\frac{1}{5}u_{2,2}$. This is the reason that the pixel $u_{2,2}$ is counted exactly once in $\mathbb{1}^T Pu$. If p were convolved with a digital image that was larger than u, then for each interior pixel $u_{i,j}$, the term $\frac{1}{5}u_{i,j}$ would appear in $(Pu)_{i,j}$ and its adjacent entries $(Pu)_{i-1,j}$, $(Pu)_{i,j-1}$, $(Pu)_{i,j+1}$, and $(Pu)_{i+1,j}$, as we have just seen. Consequently, this is the reason that pixels in the interior of u are

counted exactly once in $\mathbb{1}^T Pu$. We also see that the reflexive boundary conditions ensure that the pixels on the boundary of u are counted exactly once in $\mathbb{1}^T Pu$, for similar reasons as in the one-dimensional case. It can also be shown that $\mathbb{1}^T Pu = \mathbb{1}^T u$ if we adopt periodic boundary conditions for the convolution Pu instead of reflexive boundary conditions. Finally, the 5×5 version of the PSF (18) also satisfies $\mathbb{1}^T Pu = \mathbb{1}^T u$, for the same reasons that the above 3×3 PSF does.