A CLASS OF FAST ALGORITHMS FOR TOTAL VARIATION IMAGE RESTORATION

Junfeng Yang
Wotao Yin
Yin Zhang
Yilun Wang

This work is produced by OpenStax-CNX and licensed under the Creative Commons Attribution License 2.0†

Abstract

This report summarizes work done as part of the Imaging and Optimization PFUG under Rice University's VIGRE program. VIGRE is a program of Vertically Integrated Grants for Research and Education in the Mathematical Sciences under the direction of the National Science Foundation. A PFUG is a group of Postdocs, Faculty, Undergraduates and Graduate students formed round the study of a common problem.

This module is based on the recent work of Junfeng Yang (jfyang2992@yahoo.com.cn) from Nanjing University and Wotao Yin, Yin Zhang, and Yilun Wang (wotao.yin, yzhang, yilun.wang@rice.edu) from Rice University.

In image formation, the observed images are usually blurred by optical instruments and/or transfer medium and contaminated by noise, which makes image restoration a classical problem in image processing. Among various variational deconvolution models, those based upon total variation (TV) are known to preserve edges and meanwhile remove unwanted fine details in an image and thus have attracted much research interests since the pioneer work by Rudin, Osher and Fatemi. However, TV based models are difficult to solve due to the nondifferentiability and the universal coupling of variables. In this module, we present, analyze and test a class of alternating minimization algorithms for reconstructing images from blurry and noisy observations with TV-like regularization. This class of algorithms are applicable to both single- and multi-channel images with either Gaussian or impulsive noise, and permit cross-channel blurs when the underlying image has more than one channels. Numerical results are given to demonstrate the effectiveness of the proposed algorithms.

1 Introduction

In electrical engineering and computer science, image processing refers to any form of signal processing in which the input is an image and the output can be either an image or a set of parameters related to the image. Generally, image processing includes image enhancement, restoration and reconstruction, edge and boundary detection, classification and segmentation, object recognition and identification, compression and communication, etc. Among them, image restoration is a classical problem and is generally a preprocessing method.

†http://creativecommons.org/licenses/by/2.0/
stage of higher level processing. In many applications, the measured images are degraded by blurs; e.g. the optical system in a camera lens may be out of focus, so that the incoming light is smeared out, and in astronomical imaging the incoming light in the telescope has been slightly bent by turbulence in the atmosphere. In addition, images that occur in practical applications inevitably suffer from noise, which arise from numerous sources such as radiation scatter from the surface before the image is sensed, electrical noise in the sensor or camera, transmission errors, and bit errors as the image is digitized, etc. In such situations, the image formation process is usually modeled by the following equation

\[ f(x) = (k * u)(x) + \omega(x), \quad x \in \Omega, \]  

where \( u(x) \) is an unknown clean image over a region \( \Omega \subset \mathbb{R}^2 \), “\(*\)” denotes the convolution operation, \( k(x) \), \( n(x) \) and \( f(x) \) are real-valued functions from \( \mathbb{R}^2 \) to \( \mathbb{R} \) representing, respectively, convolution kernel, additive noise, and the blurry and noisy observation. Usually, the convolution process neither absorbs nor generates optical energy, i.e., \( \int_{\Omega} k(x) \, dx = 1 \), and the additive noise has zero mean.

Deblurring or deconvolution aims to recover the unknown image \( u(x) \) from \( f(x) \) and \( k(x) \) based on ((1)). When \( k(x) \) is unknown or only an estimate of it is available, recovering \( u(x) \) from \( f(x) \) is called blind deconvolution. Throughout this module, we assume that \( k(x) \) is known and \( \omega(x) \) is either Gaussian or impulsive noise. When \( k(x) \) is equal to the Dirac delta, the recovery of \( u(x) \) becomes a pure denoising problem. In the rest of this section, we review the TV-based variational models for image restoration and introduce necessary notation for analysis.

1.1 Total Variation for Image Restoration

The TV regularization was first proposed by Rudin, Osher and Fatemi in [12] for image denoising, and then extended to image deblurring in [11]. The TV of \( u \) is defined as

\[ TV(u) = \int_{\Omega} \| \nabla u(x) \| \, dx. \]  

When \( \nabla u(x) \) does not exist, the TV is defined using a dual formulation [18], which is equivalent to ((2)) when \( u \) is differentiable. We point out that, in practical computation, discrete forms of regularization are always used where differential operators are replaced by certain finite difference operators. We refer TV regularization and its variants as TV-like regularization. In comparison to Tikhonov-like regularization, the homogeneous penalty on image smoothness in TV-like regularization can better preserve sharp edges and object boundaries that are usually the most important features to recover. Variational models with TV regularization and \( \ell_2 \) fidelity has been widely studied in image restoration; see e.g. [2], [3] and references therein. For \( \ell_1 \) fidelity with TV regularization, its geometric properties are analyzed in [4], [16], [17]. The superiority of TV over Tikhonov-like regularization was analyzed in [1], [5] for recovering images containing piecewise smooth objects.

Besides Tikhonov and TV-like regularization, there are other well studied regularizers in the literature, e.g. the Mumford-Shah regularization [9]. In this module, we concentrate on TV-like regularization. We derive fast algorithms, study their convergence, and examine their performance.

1.2 Discretization and Notation

As used before, we let \( \| \cdot \| \) be the 2-norm. In practice, we always discretize an image defined on \( \Omega \), and vectorize the two-dimensional digitalized image into a long one-dimensional vector. We assume that \( \Omega \) is a square region in \( \mathbb{R}^2 \). Specifically, we first discretize \( u(x) \) into a digital image represented by a matrix \( U \in \mathbb{R}^{n \times n} \). Then we vectorize \( U \) column by column into a vector \( u \in \mathbb{R}^{n^2} \), i.e.

\[ u_i = U_{pq}, \quad i = 1, ..., n^2, \]  

where \( u_i \) denotes the \( i \)-th component of \( u \), \( U_{pq} \) is the component of \( U \) at \( p \)-th row and \( q \)-th column, and \( p \) and \( q \) are determined by \( i = (q - 1) \cdot n + p \) and \( 1 \leq q \leq n \). Other quantities such as the convolution kernel
where in this case, \( \pi, \omega, f \in \mathbb{R}^{n^2} \) are all vectors representing, respectively, the discrete forms of the original image, additive noise and the blurry and noisy observation, and \( K \in \mathbb{R}^{n^2 \times n^2} \) is a convolution matrix representing the kernel \( k(x) \). The gradient \( \nabla u(x) \) is replaced by certain first-order finite difference at pixel \( i \). Let \( D_i \in \mathbb{R}^{2 \times n^2} \) be a first-order local finite difference matrix at pixel \( i \) in horizontal and vertical directions. E.g. when the forward finite difference is used, we have

\[
D_i u = \begin{pmatrix} u_{i+n} - u_i \\ u_{i+1} - u_i \end{pmatrix} \in \mathbb{R}^2,
\]

for \( i = 1, ..., n^2 \) (with certain boundary conditions assumed for \( i > n^2 - n \)). Then the discrete form of TV defined in ((2)) is given by

\[
TV(u) = \sum_{i=1}^{n^2} \| D_i u \|.
\]

We will refer to

\[
\min_u TV(u) + \frac{\mu}{2} \| Ku - f \|^2
\]

with discretized TV regularization ((6)) as TV/L^2. For impulsive noise, we replace the \( \ell_2 \) fidelity by \( \ell_1 \) fidelity and refer to the resulted problem as TV/L^1.

Now we introduce several more notation. For simplicity, we let \( \sum_i \) be the summation taken over all pixels. The two first-order global finite difference operators in horizontal and vertical directions are, respectively, denoted by \( D^{(1)} \) and \( D^{(2)} \) which are \( n^2 \)-by-\( n^2 \) matrices (boundary conditions are the same as those assumed on \( D_i \)). As such, it is worth noting that the two-row matrix \( D_i \) is formed by stacking the \( i \)th row of \( D^{(1)} \) on that of \( D^{(2)} \). For vectors \( v_1 \) and \( v_2 \), we let \( v = (v_1; v_2) \triangleq (v_1^\top, v_2^\top)^\top \), i.e. \( v \) is the vector formed by stacking \( v_1 \) on the top of \( v_2 \). Similarly, we let \( D = (D^{(1)}; D^{(2)}) = \left((D^{(1)})^\top, (D^{(2)})^\top\right)^\top \). Given a matrix \( T \), we let \( \text{diag}(T) \) be the vector containing the elements on the diagonal of \( T \), and \( \mathcal{F}(T) = \mathcal{F}T \mathcal{F}^{-1} \), where \( \mathcal{F} \in n^2 \times n^2 \) is the 2D discrete Fourier transform matrix.

1.3 Existing Methods

Since TV is nonsmooth, quite a few algorithms are based on smoothing the TV term and solving an approximation problem. The TV of \( u \) is usually replaced by

\[
TV_\varepsilon(u) = \sum_i \sqrt{\| D_i u \|^2 + \varepsilon},
\]

where \( \varepsilon > 0 \) is a small constant. Then the resulted approximate TV/L^2 problem is smooth and many optimization methods are available. Among others, the simplest method is the gradient descent method as was used in [12]. However, this method suffers slow convergence especially when the iterate point is close to the solution. Another important method is the linearized gradient method proposed in [13] for denoising and in [14] for deblurring. Both the gradient descent and the linearized gradient methods are globally and at best linearly convergent. To obtain super linear convergence, a primal-dual based Newton method was proposed in [13]. Both the linearized gradient method and this primal-dual method need to solve a large system of linear equations at each iteration. When \( \varepsilon \) is small and/or \( K \) becomes more ill-conditioned, the linear system becomes more and more difficult to solve. Another class of well-known methods for TV/L^2 are the iterative shrinkage/thresholding (IST) based methods [8]. For IST-based methods, a TV denoising
problem needs to be solved at each iteration. Also, in [6] the authors transformed the TV/L\(^2\) problem into a second order cone program and solved it by interior point method.

2 A New Alternating Minimization Algorithm

In this section, we derive a new algorithm for the TV/L\(^2\) problem

\[
\min_u \sum_i \| D_i u \| + \frac{\mu}{2} \| Ku - f \|^2.
\]

(9)

In ((9)), the fidelity term is quadratic with respect to \(u\). Moreover, \(K\) is a convolution matrix and thus can be easily diagonalized by fast transforms (with proper boundary conditions assumed on \(u\)). Therefore, the main difficulty in solving ((9)) is caused by the nondifferentiability and the universal coupling of variables of the TV term. Our algorithm is derived from the well-known variable-splitting and penalty techniques in optimization. First, we introduce an auxiliary variable \(w_i \in \mathbb{R}^2\) at pixel \(i\) to transfer \(D_i u\) out of the nondifferentiable term \(\| \cdot \|\). Then we penalize the difference between \(w_i\) and \(D_i u\) quadratically. As such, the auxiliary variables \(w_i\)'s are separable with respect to one another. For convenience, in the following we let \(w \triangleq [w_1, ..., w_{n^2}]\). The approximation model to ((9)) is given by

\[
\min_{w,u} \sum_i \| w_i \| + \frac{\beta}{2} \sum_i \| w_i - D_i u \|^2 + \frac{\mu}{2} \| Ku - f \|^2,
\]

(10)

where \(\beta \gg 0\) is a penalty parameter. It is well known that the solution of ((10)) converges to that of ((9)) as \(\beta \to \infty\). In the following, we concentrate on problem ((10)).

2.1 Basic Algorithm

The benefit of ((10)) is that while either one of the two variables \(u\) and \(w\) is fixed, minimizing the objective function with respect to the other has a closed-form formula that we will specify below. First, for a fixed \(u\), the first two terms in ((10)) are separable with respect to \(w_i\), and thus the minimization for \(w\) is equivalent to solving

\[
\min_{w_i} \| w_i \| + \frac{\beta}{2} \| w_i - D_i u \|^2, \quad i = 1, 2, ..., n^2.
\]

(11)

It is easy to verify that the unique solutions of ((11)) are

\[
w_i = \max\{ \| D_i u \| - \frac{1}{\beta} \| D_i u \|, 0 \}, \quad i = 1, ..., n^2,
\]

(12)

where the convention \(0 \cdot (0/0) = 0\) is followed. On the other hand, for a fixed \(w\), ((10)) is quadratic in \(u\) and the minimizer \(u\) is given by the normal equations

\[
(\sum_i D_i^T D_i + \frac{\mu}{\beta} K^T K) u = \sum_i D_i^T w_i + \frac{\mu}{\beta} K^T f.
\]

(13)

By noting the relation between \(D\) and \(D_i\) and a reordering of variables, ((13)) can be rewritten as

\[
(D^T D + \frac{\mu}{\beta} K^T K) u = D^T w + \frac{\mu}{\beta} K^T f,
\]

(14)

where

\[
w \triangleq \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \in \mathbb{R}^{2n^2} \text{ and } w_j \triangleq \begin{pmatrix} (w_1)_j \\ \vdots \\ (w_{n^2})_j \end{pmatrix}, \quad j = 1, 2.
\]

(15)
The normal equation ((14)) can also be solved easily provided that proper boundary conditions are assumed on $u$. Since both the finite difference operations and the convolution are not well-defined on the boundary of $u$, certain boundary assumptions are needed when solving ((14)). Under the periodic boundary conditions for $u$, i.e. the 2D image $u$ is treated as a periodic function in both horizontal and vertical directions, $D^{(1)}$, $D^{(2)}$ and $K$ are all block circulant matrices with circulant blocks; see e.g. [10], [7]. Therefore, the Hessian matrix on the left-hand side of ((14)) has a block circulant structure and thus can be diagonalized by the 2D discrete Fourier transform $F$, see e.g. [7]. Using the convolution theorem of Fourier transforms, the solution of ((14)) is given by

$$u = F^{-1} \left( \frac{F(D^\top w + \mu / \beta K^\top f)}{\text{diag}(F(D^\top D + \mu / \beta K^\top K))} \right), \quad (16)$$

where the division is implemented by componentwise. Since all quantities but $w$ are constant for given $\beta$, computing $u$ from ((16)) involves merely the finite difference operation on $w$ and two FFTs (including one inverse FFT), once the constant quantities are computed.

Since minimizing the objective function in ((10)) with respect to either $w$ or $u$ is computationally inexpensive, we solve ((10)) for a fixed $\beta$ by an alternating minimization scheme given below.

**Algorithm :**

- Input $f$, $K$ and $\mu > 0$. Given $\beta > 0$ and initialize $u = f$.
- While “not converged”, Do
  - a. Compute $w$ according to ((12)) for fixed $u$.
  - b. Compute $u$ according to ((16)) for fixed $w$ (or equivalently $w$).
- End Do

### 2.2 Optimality Conditions and Convergence Results

To present the convergence results of Algorithm "Basic Algorithm" (Section 2.1: Basic Algorithm) for a fixed $\beta$, we make the following weak assumption.

Assumption 1 $\mathcal{N}(K) \cap \mathcal{N}(D) = \{0\}$, where $\mathcal{N}(\cdot)$ represents the null space of a matrix.

Define

$$M = D^\top D + \frac{\mu}{\beta} K^\top K \quad \text{and} \quad T = DM^{-1}D^\top. \quad (17)$$

Furthermore, we will make use of the following two index sets:

$$L = \{i : \|D_i u^*\| < \frac{1}{2} \} \quad \text{and} \quad E = \{1, ..., n^2\} \setminus L. \quad (18)$$

Under Assumption 1, the proposed algorithm has the following convergence properties.

**Theorem 1** For any fixed $\beta > 0$, the sequence $\{ (w^k, u^k) \}$ generated by Algorithm "Basic Algorithm" (Section 2.1: Basic Algorithm) from any starting point $(w^0, u^0)$ converges to a solution $(w^*, u^*)$ of ((10)). Furthermore, the sequence satisfies

1. $\|w_{E}^{k+1} - w_{E}^*\| \leq \sqrt{\rho((T^2)_{EE})} \|w_{E}^k - w_{E}^*\| ; \quad (19)$

2. $\|u^{k+1} - u^*\|_M \leq \sqrt{\rho(T_{EE})} \|u^{k} - u^*\|_M ; \quad (20)$

for all $k$ sufficiently large, where $T_{EE} = [T_{i,j}]_{i,j \in E \cup (n^2 + E)}$ is a minor of $T$, $\|v\|_M = v^\top M v$ and $\rho(\cdot)$ is the spectral radius of its argument.

http://cnx.org/content/m19059/1.2/
3 Extensions to Multichannel Images and TV/L

The alternating minimization algorithm given in "A New Alternating Minimization Algorithm" (Section 2: A New Alternating Minimization Algorithm) can be extended to solve multichannel extension of ((9)) when the underlying image has more than one channels and TV/L\(^1\) when the additive noise is impulsive.

3.1 Multichannel image deconvolution

Let \( \pi = [\pi^{(1)}; \ldots; \pi^{(m)}] \in \mathbb{R}^{mn^2} \) be an \( m \)-channel image, where, for each \( j \), \( \pi^{(j)} \in \mathbb{R}^{n^2} \) represents the \( j \)th channel. An observation of \( \pi \) is modeled by ((14)), in which case \( f = [f^{(1)}; \ldots; f^{(m)}] \) and \( \omega = [\omega^{(1)}; \ldots; \omega^{(m)}] \) have the same size and the number of channels as \( \pi \), and \( K \) is a multichannel blurring operator of the form

\[
K = \begin{bmatrix}
K_{11} & K_{12} & \cdots & K_{1m} \\
K_{21} & K_{22} & \cdots & K_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
K_{m1} & K_{m2} & \cdots & K_{mm}
\end{bmatrix} \in \mathbb{R}^{mn^2 \times mn^2},
\]

(21)

where \( K_{ij} \in \mathbb{R}^{n^2 \times n^2} \), each diagonal submatrix \( K_{ii} \) defines the blurring operator within the \( i \)th channel, and each off-diagonal matrix \( K_{ij}, i \neq j \), defines how the \( j \)th channel affects the \( i \)th channel.

The multichannel extension of ((9)) is

\[
\min_u \sum_i \| (I_m \otimes D_i) u \| + \frac{\mu}{2} \| Ku - f \|^2,
\]

(22)

where \( I_m \) is the identity matrix of order \( m \), and "\( \otimes \)" is the Kronecker product. By introducing auxiliary variables \( w_i \in \mathbb{R}^{2n^2}, i = 1, \ldots, n^2 \), ((22)) is approximated by

\[
\min_{w,u} \sum_i \| w_i \| + \frac{\beta}{2} \sum_i \| w_i - (I_m \otimes D_i) u \|^2 + \frac{\mu}{2} \| Ku - f \|^2.
\]

(23)

For fixed \( u \), the minimizer function for \( w \) is given by ((12)) in which \( D_i u \) should be replaced by \( (I_m \otimes D_i) u \). On the other hand, for fixed \( w \), the minimization for \( u \) is a least squares problem which is equivalent to the normal equations

\[
(I_3 \otimes (D^T D) + \frac{\beta}{2} K^T K) u = (I_3 \otimes D)^T w + \frac{\mu}{2} K^T f,
\]

(24)

where \( w \) is a reordering of variables in a similar way as given in ((15)). Under the periodic boundary condition, ((24)) can be block diagonalized by FFTs and then solved by a low complexity Gaussian elimination method.

3.2 Deconvolution with Impulsive Noise

When the blurred image is corrupted by impulsive noise rather than Gaussian, we recover \( \pi \) as the minimizer of a TV/L\(^1\) problem. For simplicity, we again assume \( \pi \in \mathbb{R}^{n^2} \) is a single channel image and the extension to multichannel case can be similarly done as in "Multichannel image deconvolution" (Section 3.1: Multichannel image deconvolution). The TV/L\(^1\) problem is

\[
\min_u \sum_i \| D_i u \| + \mu \| Ku - f \|_1.
\]

(25)

Since the data-fidelity term is also not differentiable, in addition to \( w \), we introduce \( z \in \mathbb{R}^{n^2} \) and add a quadratic penalty term. The approximation problem to ((25)) is

\[
\min_{w,z,u} \sum_i \left( \| w_i \| + \frac{\beta}{2} \| w_i - D_i u \|^2 \right) + \mu \left( \| z \|_1 + \frac{\gamma}{2} \| z - (Ku - f) \|^2 \right),
\]

(26)
where $\beta, \gamma \gg 0$ are penalty parameters. For fixed $u$, the minimization for $w$ is the same as before, while the minimizer function for $z$ is given by the famous one-dimensional shrinkage:

$$z = \max\{|Ku - f| - \frac{1}{\lambda}, 0\} \cdot \text{sgn}(Ku - f).$$

(27)

On the other hand, for fixed $w$ and $z$, the minimization for $u$ is a least squares problem which is equivalent to the normal equations

$$\left(D^\top D + \frac{\mu \gamma}{\beta} K^\top K\right)u = D^\top w + \frac{\mu \gamma}{\beta} K^\top (f + z).$$

(28)

Similar to previous arguments, ((28)) can be easily solved by FFTs.

4 Experiments

In this section, we present the practical implementation and numerical results of the proposed algorithms. We used two images, Man (grayscale) and Lena (color) in our experiments, see Figure 1. The two images are widely used in the field of image processing because they contain nice mixture of detail, flat regions, shading area and texture.

![Test images: Man (left, 1024×1024) and Lena (right, 512×512).](http://cnx.org/content/m19059/1.2/)
the signal-to-noise ratio (SNR) defined by

\[ \text{SNR} = 10 \star \log_{10} \frac{\| \overline{u} - E(\overline{u}) \|^2}{\| \overline{u} - u \|^2}, \]  

(29)

where \( \overline{u} \) is the original image and \( E(\overline{u}) \) is the mean intensity value of \( \overline{u} \). All blurring effects were generated using the MATLAB function “imfilter” with periodic boundary conditions, and noise was added using “imnoise”. All the experiments were finished under Windows Vista Premium and MATLAB v7.6 (R2008a) running on a Lenovo laptop with an Intel Core 2 Duo CPU at 2 GHz and 2 GB of memory.

4.1 Practical Implementation

Generally, the quality of the restored image is expected to increase as \( \beta \) increases because the approximation problems become closer to the original ones. However, the alternating algorithms converge slowly when \( \beta \) is large, which is well-known for the class of penalty methods. An effective remedy is to gradually increase \( \beta \) from a small value to a pre-specified one. Figure 2 compares the different convergence behaviors of the proposed algorithm when with and without continuation, where we used Gaussian blur of size 11 and standard deviation 5 and added white Gaussian noise with mean zero and standard deviation \( 10^{-3} \).

![Figure 2: Continuation vs. no continuation](http://cnx.org/content/m19059/1.2/)

In this continuation framework, we compute a solution of an approximation problem which used a smaller beta, and use the solution to warm-start the next approximation problem corresponding to a bigger \( \beta \). As can be seen from Figure 2, with continuation on \( \beta \) the convergence is greatly sped up. In our experiments, we implemented the alternating minimization algorithms with continuation on \( \beta \), which we call the resulting algorithm “Fast Total Variation de-convolution” or FTVd, which, for TV/L2, the framework is given below. [FTVd]:

- Input \( f, K \) and \( \mu > 0 \). Given \( \beta_{\text{max}} > \beta_0 > 0 \).
- Initialize \( u = f, u_p = 0, \beta = \beta_0 \) and \( \varepsilon > 0 \).
- While \( \beta \leq \beta_{\text{max}} \), Do
  - Run Algorithm "Basic Algorithm" (Section 2.1: Basic Algorithm) until an optimality condition is met.
b. $\beta \leftarrow 2 \ast \beta$.

- End Do

**Figure 3:** SNRs of images recovered from (1) for different $\beta$. 
Generally, it is difficult to determine how large $\beta$ is sufficient to generate a solution that is close to be a solution of the original problems. In practice, we observed that the SNR values of recovered images from the
approximation problems are stabilized once $\beta$ reached a reasonably large value. To see this, we plot the SNR values of restored images corresponding to $\beta = 2^0, 2^1, \cdots, 2^{18}$ in Figure 3. In this experiment, we used the same blur and noise as we used in the testing of continuation. As can be seen from Figure 3, the SNR values on both images essentially remain constant for $\beta \geq 2^7$. This suggests that $\beta$ need not to be excessively large from a practical point of view. In our experiments, we set $\beta_0 = 1$ and $\beta_{\text{max}} = 2^7$ in Algorithm "Practical Implementation" (Section 4.1: Practical Implementation). For each $\beta$, the inner iteration was stopped once an optimality condition is satisfied. For TV/L$_1$ problems, we also implement continuation on $\gamma$, and used similar settings as used in TV/L$_2$.

### 4.2 Recovered Results

In this subsection, we present results recovered from TV/L$_2$ and TV/L$_1$ problems including ((9)), ((25)) and their multichannel extensions. We tested various of blurs with different levels of Gaussian noise and impulsive noise. Here we merely present serval test results. Figure 4 gives two examples of blurry and noisy images and the recovered ones, where the blurred images are corrupted by Gaussian noise, while Figure 5 gives the recovered results where the blurred images are corrupted by random-valued noise. For TV/L$_1$ problems, we set $\gamma = 2^{15}$ and $\beta = 2^{10}$ in the approximation model and implemented continuation on both $\beta$ and $\gamma$. 

http://cnx.org/content/m19059/1.2/
Figure 5: Results recovered from TV/L1. Image Lena is blurred by a cross-channel kernel and corrupted by 40% (left) and 50% (right) random-valued noise. The top row contains the blurry and noisy observations and the bottom row shows the results recovered by FTVd.
5 Concluding Remarks

We proposed, analyzed and tested an alternating algorithm FTVd which for solving the TV/$L^2$ problem. This algorithm was extended to solve the TV/$L^1$ model and their multichannel extensions by incorporating an extension of TV. Cross-channel blurs are permitted when the underlying image has more than one channels. We established strong convergence results for the algorithms and validated a continuation scheme. Numerical results are given to demonstrate the feasibility and efficiency of the proposed algorithms.

6 Acknowledgements

This Connexions module describes work conducted as part of Rice University’s VIGRE program, supported by National Science Foundation grant DMS-0739420.

References


