Split Bregman Iterations on Regularized L^1 Total Variation Models

Marrick C. Neri
Institute of Mathematics
University of the Philippines-Diliman
mcneri@up.edu.ph

ABSTRACT

In this paper, regularized discrete versions of the L^1 total variation based image denoising model are solved using split Bregman iterations. The methods use inexact solutions which are effective in restoring images corrupted with impulse noise.

Keywords
Bregman method, image denoising, impulse noise

1. INTRODUCTION

Image denoising deals with the reconstruction of images that are corrupted with high energy oscillations known as noise. There are many mathematical types of noise, two of which are additive Gaussian noise and impulse noise. Gaussian noise follows a normal or Gaussian distribution. Impulse noise affects only some pixels of the image and can manifest as salt-and-pepper (SP) noise or random noise.

At present, there are many methods that efficiently remove Gaussian or impulse noise. Widely researched methods for image denoising are total variation (TV) based methods. In the presence of Gaussian noise, there is the Rudin-Osher-Fatemi (ROF) model [14]:

$$\min_{u\epsilon BV(\Omega)} \int_\Omega |\nabla u(x)|dx \;+\; \frac{\alpha}{2}\int_\Omega (u(x)-f(x))^2 dx \qquad (1)$$

on the strictly convex image domain $\Omega \subset \mathbb{R}^2$ with Lipschitz boundary, where the space of bounded variations BV is defined by

$$BV(\Omega) := \left\{ f \in L^1(\Omega) \;:\; \int_\Omega |\nabla f| < \infty \right\}$$

and

$$\int_\Omega |\nabla f|\, dx = \sup\left\{ \int_\Omega f \operatorname{div} v\, dx \,:\, v \in C^1_c(\Omega), |v|_{l^1} \leq 1 \right\}.$$

A good discussion on the solution of problem (1) can be found in [4]. Numerically, the ROF model is difficult to solve by gradient-based methods since the TV term is nondifferentiable. Usually, additional regularization is imposed on the model to make it sufficiently smooth. A recent method to solve the ROF model is the split Bregman algorithm [7, 11]. In this method, the l^1 and l^2 portions of the discretized functional are decoupled and the problem is reduced to a sequence of unconstrained optimization problems and Bregman updates. Numerical results demonstrate the Bregman method is fast and efficient in removing Gaussian noise.

It is known that in reconstructing images corrupted with impulse noise, the well-suited TV model is

$$\min_{u\epsilon BV(\Omega)} \int_\Omega |\nabla u(x)|dx \;+\; \alpha\int_\Omega |u(x)-f(x)|dx. \qquad (2)$$

See, e.g., [3, 9]. Problem (2) is an example of an L^1-regularized optimization problem. The general form for such problems is

$$\min_u |\Phi(u)| + H(u)$$

where $|\cdot|$ denotes the L^1-norm, and $|\Phi(u)|$ and $H(u)$ are convex functions. Compared with the ROF, there is more difficulty in solving problem (2) since both terms are nonsmooth. In [3], global smoothing is implemented on the problem and the resulting model was solved using a steepest-descent type approach. In [5], the problem was locally smoothed and solved by a primal-dual active set method. A decomposition strategy was proposed in [8] to handle problem (2) in its original form.

A review of methods in handling TV models for different noise types is in [13]. Recent hybrid algorithms that exploit the merge of numerical methods on the TV models and technological advances further increase the quality of restored images; see, e.g., [12].

In this paper, we formulate split Bregman algorithms to solve regularized versions of problem (2). These versions employ local or global smoothing. Instead of the Gauss-Seidel type method used in the split Bregman iteration for problem (1), we will use the conjugate gradient method (CGM). A numerical study on the implementation of the algorithm towards removing impulse noise ends the paper.

2. BREGMAN ITERATIVE ALGORITHM

The Bregman iterative algorithm was introduced in [11] as a method for solving constrained, convex, minimization problems of the form

$$\arg\min_u \; J(u) \; s.t. \; H(u) = 0, \tag{3}$$

where J and H are convex and nonnegative functions. This algorithm is similar to the augmented Lagrangian method. A key concept in the Bregman algorithm is the Bregman distance associated with the function J at the point v,

$$D_J^p(u,v) = J(u) - J(v) - \langle p, u - v \rangle,$$

where $p \in \partial J(v)$, the subdifferential of J at v. Essentially, the Bregman distance is the comparison of the value $J(u)$ with the tangent $J(v) + \langle\, p, u - v \,\rangle$ [6]. We have the following properties of the Bregman distance:

1. $D_J^p(u,u) = 0$
2. $D_J^p(u,v) \geq 0$
3. $D_J^p(u,v) \geq D_J^p(w,v)$ when w is on the line segment between u and v
4. $D_J^p(u,v) = -D_J^p(v,u)$.

Based on properties (1)-(3), the Bregman distance gives a measure of closeness between two points; but with property (4) and the fact that it does not satisfy the triangle inequality, D_J^p cannot be considered as distance in the usual sense.

The main Bregman iteration problem is defined as

$$u^{k+1} = \arg\min_u \; D_J^{p^k}(u, u^k) + \gamma H(u) \tag{4}$$

with Lagrange multiplier $\gamma > 0$, and $p^k \in \partial J(u^k)$ [6]. At each iteration the algorithm solves an unconstrained, convex minimization problem. The solution u^{k+1} satisfies the optimality condition

$$0 \in \; \partial(J(u^{k+1}) - J(u^k) - < p^k, u^{k+1} - u^k > + \gamma H(u^{k+1}))$$

which leads to the result

$$p^k - \gamma \nabla H(u^{k+1}) \; \in \; \partial J(u^{k+1}). \tag{5}$$

Since $p^{k+1} \in \partial J(u^{k+1})$, we can simply take $p^{k+1} = p^k - \gamma \nabla H(u^{k+1})$ assuming H is differentiable. The Bregman algorithm is presented as follows [6]:

Algorithm 1 Bregman Iterative Algorithm

```
• Initialize. Set $k = 0,\; u^0 = 0,\; p^0 \in \partial J(u^0)$
• while not convergent
      $u^{k+1} \;=\; \arg\min_u \; D_J^{p^k}(u, u^k) + \gamma H(u)$
      $p^{k+1} \;=\; p^k - \gamma \nabla H(u^{k+1})$
```

In principle, when $u^0 \neq 0$, one can start with a different value for p^0 as long as it is a subgradient of $J(u_0)$. However, the computations involved make these initial values impractical [11].

The Bregman algorithm aims to solve problem (3) by attaining a u such that $H(u)$ is minimized leading towards satisfying the constraint $H(u) = 0$. As the Bregman distance is minimized, the minimum of $J(u)$ is obtained. Formally stating these, we have the following convergence properties of the algorithm [2, 6, 7]:

Theorem 1. *Let J and H be convex, nonnegative functions where H is differentiable. If solutions to problem (4) exist, then*

1. *There is decrease in Bregman distance: $D_J^p(u, u^{k+1}) \leq D_J^p(u, u^k)$ at every iteration.*
2. *There is monotonic decrease in H, i.e., $H(u^{k+1}) \leq H(u^k)$.*
3. *The method converges to a minimizer u^* of H, with*
   $$H(u^k) \leq H(u^*) + \frac{J(u)}{k}$$
   and $k \to \infty$.

Constrained optimization problems of the form

$$\min_u \{J(u) \; : \; Au = b\} \tag{6}$$

can be solved efficiently by Bregman method, where $H(u) = Au - b$. The problem is first transformed to an unconstrained problem with a quadratic penalty term:

$$\min_u J(u) + \frac{\lambda}{2}\|Au - b\|_2^2 \tag{7}$$

where $\lambda \to \infty$ so that the original constraint is enforced. When A is a linear operator, the Bregman iteration is

$$\left.\begin{aligned} u^{k+1} &= \arg\min_u J(u) + \tfrac{\lambda}{2}\|Au - b^k\|_2^2 \\ b^{k+1} &= b^k + b - Au^k. \end{aligned}\right\} \tag{8}$$

where b^k, initialized at b, is the auxiliary variable responsible for adding the error in the constraint back to the right hand side [11, 16]. In practice, the exact value of u need not be solved for at each iteration, therefore resulting in quicker iterations. The error resulting from low accuracy in the solution is handled by the algorithm's error forgetting nature [16]: when u is inexactly solved for within a tolerance level, then it takes two iterations $k+1$ and $k+2$, with $k > m$ for some m, for the algorithm to reach a solution.

3. SPLIT BREGMAN METHOD FOR TV DENOISING

One of the best applications of Bregman iterations is on TV denoising. Suppose an image is corrupted with white noise. A popular model for reconstruction is the anisotropic problem

$$\min_{u \in \mathbb{R}^n} |\nabla_x u| + |\nabla_y u| + \frac{\alpha}{2}\|u - f\|_2^2 \tag{9}$$

where $|\cdot|$ is the l^1-norm, $f \in \mathbb{R}^n$, and ∇_x and ∇_y are finite difference matrices in the horizontal and vertical directions, respectively. Substituting d_x for $\nabla_x u$ and d_y for $\nabla_y u$, problem (9) becomes the constrained problem

$$\begin{aligned} \min \quad & |d_x| + |d_y| + \tfrac{\alpha}{2}\|u-f\|_2^2 \\ \text{s.t.} \quad & d_x = \nabla_x u \text{ and } d_y = \nabla_y u. \end{aligned}$$

As was done in problem (7), penalty function terms are added to the objective:

$$\min |d_x| + |d_y| + \frac{\alpha}{2}\|u-f\|_2^2 + \frac{\lambda}{2}\|d_x - \nabla_x u\|_2^2 + \frac{\lambda}{2}\|d_y - \nabla_y u\|_2^2.$$

The resulting Bregman iteration is

$$\begin{aligned} \left(u^{k+1}, d_x^{k+1}, d_x^{k+1}\right) = \quad & \arg\min_{u,d_x,d_y} |d_x| + |d_y| + \tfrac{\alpha}{2}\|u-f\|_2^2 \\ & + \tfrac{\lambda}{2}\|d_x - \nabla_x u - b_x^k\|_2^2 \\ & + \tfrac{\lambda}{2}\|d_y - \nabla_y u - b_y^k\|_2^2 \end{aligned} \tag{10}$$

$$\left.\begin{aligned} b_x^{k+1} &= b_x^k + \left(\nabla_x u^{k+1} - d_x^{k+1}\right) \\ b_y^{k+1} &= b_y^k + \left(\nabla_y u^{k+1} - d_y^{k+1}\right) \end{aligned}\right\}. \tag{11}$$

The main minimization problem results in an expression where the variables u and d are directly interacting. To solve problem (10), Goldstein and Osher [7] introduced the split Bregman algorithm. This technique is similar to the alternating direction method where the main problem is split into two subproblems, and minimized with respect to u and d separately. Firstly,

$$\begin{aligned} u^{k+1} = \quad & \arg\min_u \quad \tfrac{\alpha}{2}\|u-f\|_2^2 + \tfrac{\lambda}{2}\|d_x^k - \nabla_x u - b_x^k\|_2^2 \\ & + \tfrac{\lambda}{2}\|d_y^k - \nabla_y u - b_y^k\|_2^2. \end{aligned} \tag{12}$$

Then the second subproblem is solved:

$$\begin{aligned} d_x^{k+1} &= \arg\min_{d_x} |d_x| + \tfrac{\lambda}{2}\|d_x - \nabla_x u^{k+1} - b_x^k\|_2^2, \\ d_y^{k+1} &= \arg\min_{d_y} |d_y| + \tfrac{\lambda}{2}\|d_y - \nabla_y u^{k+1} - b_y^k\|_2^2. \end{aligned}$$

The solution to problem (12) satisfies the optimality condition

$$(\alpha I - \lambda\triangle)u^* = \alpha f + \lambda\nabla_x^\top \left(d_x^k - b_x^k\right) + \lambda\nabla_y^\top \left(d_y^k - b_y^k\right) \tag{13}$$

where $\triangle = -\nabla^\top\nabla$. A fast algorithm to get approximate solutions to system (13) is the Gauss-Seidel method:

$$\begin{aligned} \phi_{i,j}^k \quad = \quad & \tfrac{\lambda}{\alpha+4\lambda}\left(u_{i+1,j}^k + u_{i-1,j}^k + u_{i,j+1}^k + u_{i,j-1}^k\right. \\ & + d_{x,i-1,j}^k - d_{x,i,j}^k + d_{y,i,j-1}^k - d_{y,i,j}^k \\ & \left. - b_{x,i-1,j}^k + b_{x,i,j}^k - b_{y,i,j-1}^k + b_{y,i,j}^k\right) + \tfrac{\alpha}{\alpha+4\lambda} f_{i,j}, \\ u_{i,j}^{k+1} \quad = \quad & \phi_{i,j}^k. \end{aligned}$$

The solutions d_x^{k+1} and d_y^{k+1} can be obtained as

$$\left.\begin{aligned} d_x^{k+1} &= shrink\left(\nabla_x u^{k+1} + b_x^k, \tfrac{1}{\lambda}\right) \\ d_y^{k+1} &= shrink\left(\nabla_y u^{k+1} + b_y^k, \tfrac{1}{\lambda}\right) \end{aligned}\right\} \tag{14}$$

where for $\gamma \geq 0$,

$$shrink(v,\gamma) = \begin{cases} v-\gamma, & v \in (\gamma, \infty) \\ 0, & v \in [-\gamma, \gamma] \\ v+\gamma, & v \in (-\infty, -\gamma). \end{cases}$$

Another model for TV denoising is the isotropic model

$$\min_{u\in\mathbb{R}^n} \sum_i \sqrt{(\nabla_x u)_i^2 + (\nabla_y u)_i^2} + \frac{\alpha}{2}\|u-f\|_2^2 \tag{15}$$

Similar to the splitting of the l^1 and l^2 components of the anisotropic problem leading to problem (10), the split Bregman problem for problem (15) is

$$\begin{aligned} \min_{u,d_x,d_y} \|(d_x,d_y)\|_2 + \tfrac{\alpha}{2}\|u-f\|_2^2 + \tfrac{\lambda}{2}\|d_x - \nabla_x u - b_x\|_2^2 \\ + \tfrac{\lambda}{2}\|d_y - \nabla_y u - b_y\|_2^2 \end{aligned} \tag{16}$$

where $d_x \approx \nabla_x u$, $d_y \approx \nabla_y u$, and

$$\|(d_x,d_y)\|_2 = \sum_{i,j} \sqrt{d_{x,i,j}^2 + d_{y,i,j}^2}.$$

The iterative solution for u is the same as in the anisotropic problem, namely,

$$u_{i,j}^{k+1} = \phi_{i,j}^k.$$

The d-solution can be obtained using a generalized shrinkage formula [15]

$$\begin{aligned} d_x^{k+1} &= \tfrac{\nabla_x u^k + b_x^k}{s^k} \max\left(s^k - \tfrac{1}{\lambda}, 0\right) \\ d_y^{k+1} &= \tfrac{\nabla_y u^k + b_y^k}{s^k} \max\left(s^k - \tfrac{1}{\lambda}, 0\right) \end{aligned}$$

with

$$s^k = \sqrt{|\nabla_x u^k + b_x^k|^2 + |\nabla_y u^k + b_y^k|^2}.$$

4. SPLIT BREGMAN ITERATIONS FOR REGULARIZED L^1 TV MODEL

The TV denoising models with l^2 fidelity term worked particularly well in removing additive Gaussian noise. Numerical tests, e.g. in [7, 11] demonstrate the capability of Split Bregman iteration in producing good image reconstructions at a fast rate. A type of noise more common in applications is impulse noise, such as salt-and-pepper (SP) noise or random-valued (RV) noise. A model more suited to this type of noise is the L^1 TV model [3, 9]

$$\min_{u\epsilon BV(\Omega)} \int_\Omega |\nabla u(x)|dx \;+\; \alpha\int_\Omega |u(x) - f(x)|dx. \tag{17}$$

This model has the capability of recovering original image pixels, making it suitable for removing outlier or impulse noise [9]. However, the L^1-problem is nondifferentiable. Direct methods like the gradient method and the Newton method cannot be applied. Moreover, the problem is not strictly convex, and hence, the solution is not unique. The discrete anisotropic analogue of problem (19) is

$$\min_u |\nabla_x u| + |\nabla_y u| + \alpha\sum_i |u_i - f_i|.$$

Utilizing the Bregman method on this model results in the main minimization problem

$$\underset{u,d_x,d_y}{\arg\min} |d_x| + |d_y| + \alpha \sum_i |u_i - f_i| + \tfrac{\lambda}{2}\|d_x - \nabla_x u - b_x^k\|_2^2 + \tfrac{\lambda}{2}\|d_y - \nabla_y u - b_y^k\|_2^2. \tag{18}$$

As the d subproblem and b update do not deal with the model's fidelity term, their respective subproblem and update remain the same as in equations (11) and (14). The u subproblem, on the other hand, will be altered in the fidelity term. It becomes

$$\begin{aligned} u = \arg\min_u \alpha \sum_i |u_i - f_i| + \tfrac{\lambda}{2}\|d_x - \nabla_x u - b_x^k\|_2^2 \\ + \tfrac{\lambda}{2}\|d_y - \nabla_y u - b_y^k\|_2^2 \end{aligned} \tag{19}$$

This subproblem poses difficulty since the first term is non-differentiable. There are two approaches to remedy the fidelity term's non-differentiability, namely, global regularization and local regularization.

4.1 Global smoothing

Problem (19) can be sufficiently smoothened by introducing a small parameter ε to the fidelity term. The result is the globally regularized problem

$$\min_u J_G(u)$$

where

$$J_G(u) := \alpha \sum_i \sqrt{(u_i - f_i)^2 + \varepsilon} + \tfrac{\lambda}{2}\|d_x - \nabla_x u - b_x^k\|_2^2 + \tfrac{\lambda}{2}\|d_y - \nabla_y u - b_y^k\|_2^2. \tag{20}$$

Solving for the argument minimizer u is numerically difficult since u cannot be easily separated as it was in the model with l^2 fidelity term. We opt to solve for u through iterative means, specifically, the CGM. Differentiating J_G, we get

$$\frac{\partial}{\partial u_i} J_G(u) = \frac{\alpha(u_i - f_i)}{\sqrt{(u_i - f_i)^2 + \varepsilon}} + \lambda \left(\nabla_{xi}^\top((\nabla_x u)_i - d_{xi} + b_{xi}) + \nabla_{yi}^\top((\nabla_y u)_i - d_{yi} + b_{yi}) \right)$$

where for a matrix D, D_i is the i^{th} row. With u and f as column vectors, the gradient F_G is equal to

$$F_G = \alpha\psi(u-f) + \lambda \left(\nabla_x^\top (\nabla_x u - d + b) + \nabla_y^\top (\nabla_y u - d + b) \right) \tag{21}$$

where the diagonal matrix ψ has diagonal entries

$$[\psi_{ii}] = \frac{1}{\sqrt{(u_i - f_i)^2 + \varepsilon}}.$$

A sufficient smoothing of the u-subproblem can be made on kinks of the function, and not necessarily on the entire domain. This can be achieved via a local smoothing approach.

4.2 Local smoothing

When ε is close to zero, the globally regularized model could encounter ill-conditioning at kinks in the solution. An alternative approach is to use local regularization. A local smoothing of the l^1 model is made by replacing the fidelity term with a lower approximation, i.e., the u-subproblem becomes

$$\min_u J_L(u) := \alpha \sum_i \phi(u_i) + \alpha \sum_i |u_i - f_i| + \tfrac{\lambda}{2}\|d_x - \nabla_x u - b_x^k\|_2^2 + \tfrac{\lambda}{2}\|d_y - \nabla_y u - b_y^k\|_2^2 \tag{22}$$

where the Huber function ϕ is defined by

$$\phi(u_i) = \begin{cases} |u_i - f_i| - \frac{\gamma}{2}, & \text{if } |u_i - f_i| \ge \gamma \\ \frac{1}{2\gamma}|u_i - f_i|^2, & \text{otherwise} \end{cases},$$

for $\gamma > 0$. The index i is said to belong to the active set $\mathcal{A}$ if $|u_i - f_i| \ge \gamma$. It is well-known that ϕ is differentiable and that its minimizer equals that of the original fidelity term. The derivative of ϕ is given by

$$\frac{\partial}{\partial u_i}\phi(u_i) = \begin{cases} \sigma(u_i - f_i), & \text{if } |u_i - f_i| > \gamma \\ 0, & \text{if } |u_i - f_i| = \gamma \\ \frac{1}{\gamma}(u_i - f_i), & \text{otherwise} \end{cases} \tag{23}$$

with $\sigma(\cdot)$ as the sign function. The gradient of the locally smoothed J_L is

$$F_L = \alpha\Phi(u) + \lambda \left(\nabla_x^\top (\nabla_x u - d + b) + \nabla_y^\top (\nabla_y u - d + b) \right) \tag{24}$$

where $\Phi(u)_i = \partial\phi(u_i)$.

In both the globally and locally smoothed u-subproblems (20) and (22), we obtain the solution using CGM. We apply an inexact stepsize obtained using Armijo rule. We refer the reader to [1, 10] for details of this rule. The split Bregman algorithm with CGM for removing impulse noise is presented below. The method uses either derivatives (21) or (24) in the gradient evaluation of the CGM for the globally regularized problem or locally regularized problem, respectively.

Algorithm 2 Split Bregman method on L1TV denoising (SBL1)

```
while $\|u^k - u^{k-1}\|_2 > tol$ do
    Solve for $u^{k+1}$ by CGM using (21) or (24) in the gradient approximation
    Solve for $d^{k+1}$ using (14):
        $d_x^{k+1} = shrink\left(\nabla_x u^{k+1} + b_x^k, \frac{1}{\lambda}\right)$
        $d_y^{k+1} = shrink\left(\nabla_y u^{k+1} + b_y^k, \frac{1}{\lambda}\right)$
    Solve for $b^{k+1}$ using (11):
        $b_x^{k+1} = b_x^k + \left(\nabla_x u^{k+1} - d_x^{k+1}\right)$
        $b_y^{k+1} = b_y^k + \left(\nabla_y u^{k+1} - d_y^{k+1}\right)$
end while
```

In the next section, we discuss numerical results of the algorithm. We denote as SBGS the method using (21) and SBLS the method using (24).

5. NUMERICAL RESULTS

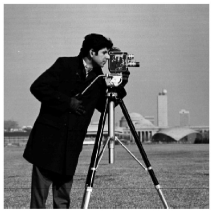
Numerical implementations of the split Bregman algorithm (SBL1) on two TV models for salt-and-pepper noise and random-valued noise were conducted. The method was coded in MATLAB, and run in a machine with a speed of 2.93 GHz and with 2 GB of RAM. For both SBGS and SBLS methods for SP noise, we initially used a maximum number of $c = 100$ CGM iterations, and $kmax = 20$ outer iterations of SBL1. We observed that within five to seven iterations, the reconstructions were already good enough, yet the method still took a long time to terminate. In later tests, we applied a decreasing number of CGM iterations, specifically, we set $c = \text{round}(M/\log(e+k))$, where k is the outer iteration number and $M >> 1$. In our tests, we set $M = 30$. To determine the efficiency of the method, we use the *peak signal-to-noise ratio* (PSNR) defined as

$$PSNR = 10\log_{10}\left(\frac{\sum_{ij} 255^2}{\sum_{ij}(\mathbf{f}_{ij} - \mathbf{u}_{ij})^2}\right)\text{dB} \qquad (25)$$

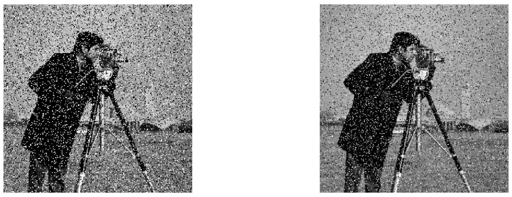
where $\mathbf{f}$ is the original image and $\mathbf{u}$ is the reconstructed image from the noisy image $\mathbf{f}$ on the 8-bit display. A higher PSNR signifies more noise is removed.

We also consider the image residual, which is the error of the reconstructed image from the original clean image. The smaller this value, the closer the reconstruction is to the original image.

For termination, we used the same stopping criteria in [7], i.e., when the condition $\|u^k - u^{k-1}\|/\|u^k\| < tolerance$ is met. In our tests, $tolerance = 10^{-6}$. The initial image solution is $u^0 = 0$.



(a)



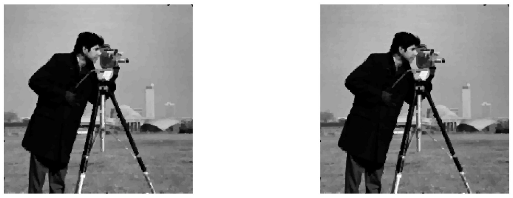
(b) SP noise

(c) RV noise

Figure 1: ***(a) original image; (b) and (c) are images with 20% SP noise and 20% RV noise, respectively***

Our first test image is the 256×256 Figure 1(a), with 20% SP noise (figure 1(b)). It was observed in [7] that good results are obtained by setting $\lambda = 2\alpha$. In this example, the parameter values we used are $\alpha = 1.75$ and $\lambda = 3.5$. For all test cases, we have $\gamma = 10^{-5}$, and the global regularization parameter is $\varepsilon = 10^{-5}$. Reconstructions of SBL1 using global and local smoothing are shown in figures 2(a) and 2(b) after at most 180 iterations, respectively. It is observed that the reconstructions are nearly the same in both methods as per PSNR values and image residual values (see Table 1).

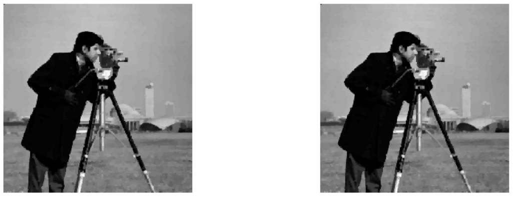
The image with 20% RV noise is shown in figure (1(c)). The parameter values used are $\alpha = 1.75$ and $\lambda = 3.5$ Denoised images are in figures 3(a) and 3(b). For both types of noise, the denoising capabilities of both regularization models are well displayed, with nearly equivalent PSNR values, as shown in Table 1. The main difference in the two methods is the computing time: it took longer for SBLS to converge than SBGS. This *slow* behavior of SBLS is symptomatic, largely due to the additional step of determining the indices in the active set $\mathcal{A}$.



(a) SBGS

(b) SBLS

Figure 2: Reconstruction of images with 20% SP noise using SBGS and SBLS, respectively



(a) SBGS

(b) SBLS

Figure 3: Reconstruction of images with 20% RV noise using SBGS and SBLS, respectively

	SP Noise			RV Noise		
	PSNR	Res	Time	PSNR	Res	Time
SBLS	25.46	13.63	79.19	25.87	13.03	78.16
SBGS	25.42	13.72	45.44	25.80	13.13	46.96

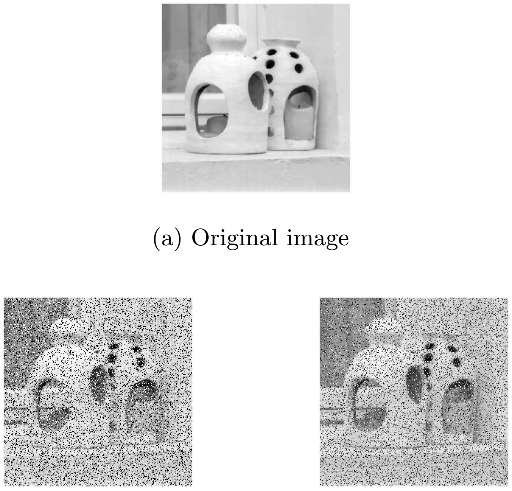
Table 1: ***Average running time (*****Time*****), resulting PSNR, and image residual (*****Res*****) of the split Bregman methods for global smoothing (SBGS) and local smoothing (SBLS) on the image with 20% noise***

The next example is a figure with 30% SP noise and 30% RV noise (see figures 4(b) and 4(c)). The parameter values are $\alpha = 1.55$ and $\lambda = 3.1$ in removing both SP and RV noise. The reconstructed images are figure 5(a) and figure 5(b) for SP noise, and figures 6(a) and 6(b) for RV noise. Table 2 shows PSNRs, image residuals, and computer running times of the methods for the different noises.

Next, we present PSNR plots for both locally regularized and globally regularized models for different values of α,

	SP Noise			RV Noise		
	PSNR	Res	Time	PSNR	Res	Time
SBLS	31.00	7.21	83.12	32.26	6.24	80.49
SBGS	30.79	7.39	51.26	31.95	6.46	52.84

Table 2: ***A comparison of resulting values of the split Bregman methods on the image with 30% noise***



(a) Original image

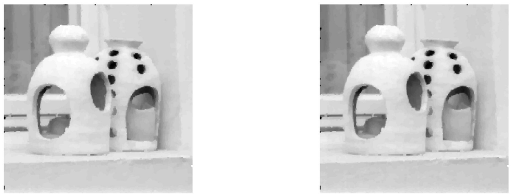
(b) SP noise

(c) RV noise

Figure 4: ***Images (b) and (c) are images with 30% SP noise and 30% RV noise, respectively***

specifically, α takes values from 1 to 3 with 0.05 increments. The test image has 20% salt-and-pepper noise. Figure 7(a) shows a quadratic trend in the PSNR values of the globally regularized model, with the highest PSNR value 26.417 achieved at $\alpha = 2$. The plot for PSNR values of the locally regularized model, in Figure 7(b), has a more zigzag pattern. The highest PSNR value 25.9345 occurs at $\alpha = 1.3$. Still, it can be observed that in general, the PSNR values tend to decrease beyond $\alpha = 1.3$.

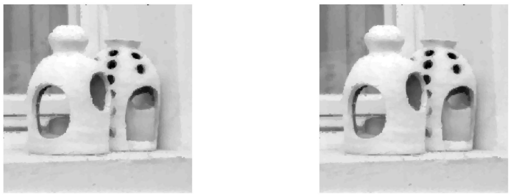
As mentioned earlier, the SBL1 method yields good results within a few outer iterations. It was observed in [7] that the split Bregman method does not need to solve the u-subproblem to full convergence. The method converges to a minimizer even if inexact iterative methods are used for each subproblem. This observation is verified by comparing results of SBL1 with 30 CGM iterations and with 50 CGM iterations. The test image with 10% SP noise is in figure (8(b)), and the restored images are figures 8(c) and 8(d).



(a) SBGS

(b) SBLS

Figure 5: ***Reconstructions of images with 30% SP noise***



(a) SBGS

(b) SBLS

Figure 6: ***Reconstructions of images with 30% RV noise***

Finally, we compared the performance of SBL1 on the regularized variation models with the primal-dual algorithm (PDA) in [5]. Note that the PDA solves a version of the l^1 variation model where both fidelity and TV terms are locally regularized. We apply the same regularizing parameter value of 10^{-5} for both terms; α is 1.5. In our tests, we generated eight noisy images of test image 1 (the cameraman) with the same level of noise, i.e., randomly chosen 20% of the pixels have salt-and-pepper noise or random valued noise. Since the PDA employs second-order information, naturally it would require less iterations to converge than first order methods such as the steepest descent method or conjugate gradient method. In the controlled experiments, we used 60 iterations for the PDA and the split Bregman methods. Table 3 shows average values of the methods for PSNR and running time.

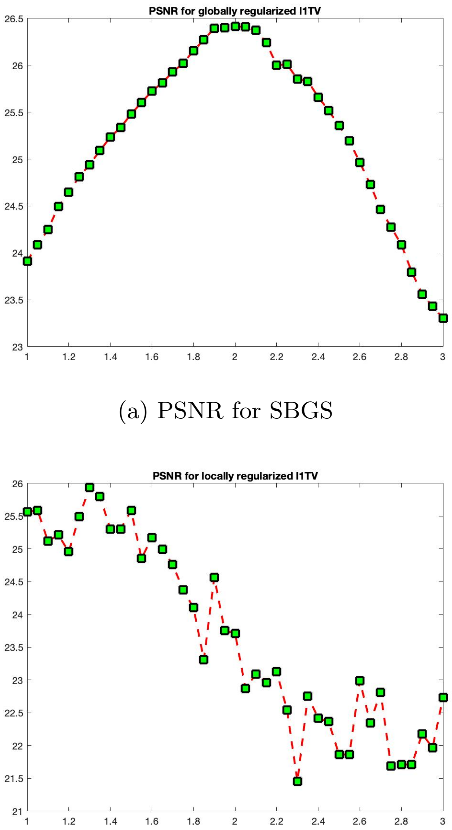
	SP Noise		RV Noise	
	PSNR	Time	PSNR	Time
SBLS	25.05	32.40	25.22	31.48
SBGS	25.00	21.60	25.17	20.59
PDA	25.40	45.41	25.79	45.04

Table 3: ***A comparison of PSNR values and running times of the split Bregman methods and PDA on an image with 20% noise***

In this case, the PDA gave the better PSNR values, but those of SBGS and SBLS are close enough at less time usage. These results demonstrate the efficiency of the Bregman methods in providing good image restorations. Figures 9(b)–9(d) show sample reconstructions of the two methods.

6. CONCLUSION AND RECOMMENDATIONS

In this paper, we implemented the split Bregman method on restoring images with impulse noise. Whereas in the work of Goldstein and Osher [7] the variation model has a l^2 fidelity term that is better suited for additive Gaussian noise, here we used a l^1 norm for restoring essential image features. Two regularized models for the fidelity term of the L^1 TV model were uitlized leading to two versions of the method. Using CGM in solving the resulting u-subproblems,



(a) PSNR for SBGS

(b) PSNR for SBLS

Figure 7: PSNR plots for different values of α. The vertical axis is the PSNR axis, and the horizontal axis is the α axis.

we numerically demonstrate the speed and efficiency of the split Bregman methods in removing salt-and-pepper noise and random-valued noise. The methods converged even with inexact solutions from the CGM.

A further extension of this research is to consider the application of a primal-dual scheme that employ the Bregman iterations on the l^1 regularized problem, as was done for the l^2 regularized problem in [17].

7. REFERENCES

[1] D. Bertsekas. Nonlinear Programming. *Athena Scientific, 2nd Edition*, 2003.

[2] J. Bush. Bregman Algorithms. University of California, Santa Barbara, 2011

[3] T. Chan and S. Esedoglu. Aspects of Total Variation Regularized L1 Function Approximation. *SIAM J Appl. Math.*, 65(5):1817–1837, 2005.

[4] R. Choksi, Y. v. Gennipy, and A. Oberman. Anisotropic Total Variation Regularized L^1-Approximation and Denoising/Deblurring of 2D Bar Codes. *arXiv:1007.1035 [math.OC]*, 2017

[5] Y. Dong, M. Hintermüller, and M. Neri. An Efficient Primal-Dual Method to Image Restoration. *SIAM J Imaging Sci.*, 2(4): 1168–1189, 2009.

[6] P. Getreuer. Rudin-Fatemi-Osher Total Variation Denoising using Split Bregman. *Image Processing On Line*, 2:74–95, 2012.

[7] T. Goldstein and S. Osher. The Split Bregman Method for L1 Regularized Problems. *SIAM J Imaging Sci.*, 2(2):323–343, 2009.

[8] M.C. Mojica. A VU-algorithm for L^1 total variation-based image denoising. *MS thesis, University of the Philippines-Diliman*, 2013

[9] M. Nikolova. Minimizers of cost-functions involving nonsmooth data-fidelity terms. Application to the processing of outliers. *SIAM J Numer. Anal.*, 40(3): 965–994, 2002

[10] J. Nocedal and S. Wright. Numerical Optimization. *Springer-Verlag New York, 2nd Edition*, 2006

[11] S. Osher, M. Burger, D. Goldfarb, J. Xu and W. Yin, An iterative regularization method for total variation-based image restoration. *Multiscale Model. Simul.*, 4(2): 460–489, 2006.

[12] T. Pock, M. Unger, D. Cremers, and H. Bischof. Fast and Exact Solution of Total Variation Models on the GPU. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops: 1–8, 2008

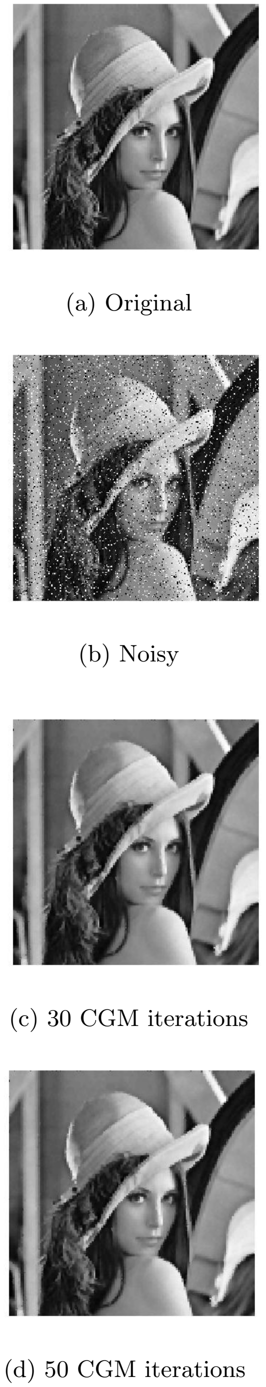
[13] P. Rodriguez. Total Variation Regularization Algorithms for Images Corrupted with Different Noise Models: A Review. *J Elec Comp Engg.*, 10: 1–16, 2013

[14] L. Rudin, S. Osher and E. Fatemi. Nonlinear Total Variation Based Noise Removal Algorithm *PHYSICA D*, 60: 259–268, 1992

[15] Y. Wang, W. Yin, and Y. Zhang. A fast algorithm for image deblurring with total variation regularization. *CAAM Technical Reports*, 2007.

[16] W. Yin and S. Osher. Error Forgetting of Bregman Iteration.*J Sci. Comput.*, 54(2–3): 684–695, 2013

[17] X. Zhang, M. Burger, and S. Osher. A unified primal-dual algorithm framework based on Bregman iteration. *J Sci Comput.*, 46(1): 20–46, 2011.



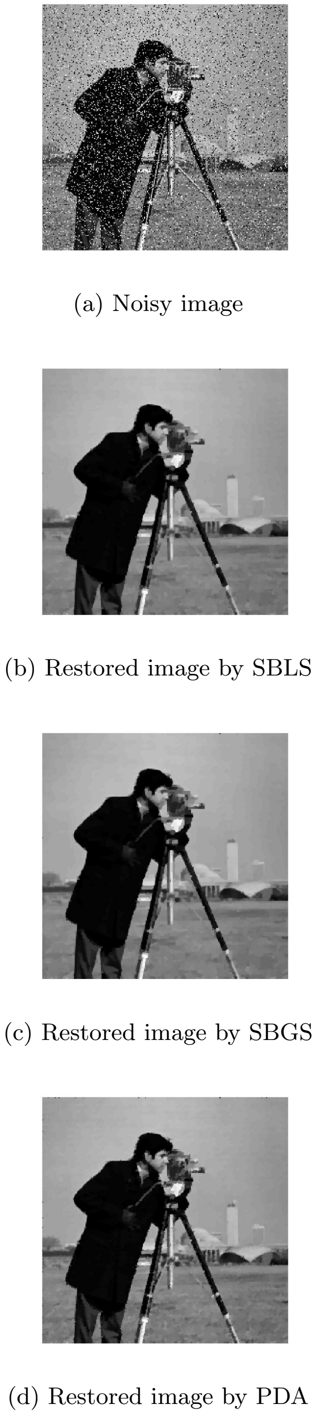
(a) Original

(b) Noisy

(c) 30 CGM iterations

(d) 50 CGM iterations

Figure 8: ***(a) Original image; (b) image with 10% SP noise; (c) and (d), restored image with 30 CGM iterations and 50 CGM iterations, respectively***



(a) Noisy image

(b) Restored image by SBLS

(c) Restored image by SBGS

(d) Restored image by PDA

Figure 9: Image reconstructions of different methods.