

The Split Bregman Method for L1-Regularized Problems*

Tom Goldstein[†] and Stanley Osher[†]

Abstract. The class of L1-regularized optimization problems has received much attention recently because of the introduction of “compressed sensing,” which allows images and signals to be reconstructed from small amounts of data. Despite this recent attention, many L1-regularized problems still remain difficult to solve, or require techniques that are very problem-specific. In this paper, we show that Bregman iteration can be used to solve a wide variety of constrained optimization problems. Using this technique, we propose a “split Bregman” method, which can solve a very broad class of L1-regularized problems. We apply this technique to the Rudin–Osher–Fatemi functional for image denoising and to a compressed sensing problem that arises in magnetic resonance imaging.

Key words. constrained optimization, L1-regularization, compressed sensing, total variation denoising

AMS subject classification. 65K05

DOI. 10.1137/080725891

1. Introduction. The category of L1-regularized problems includes many important problems in engineering, computer science, and imaging science. The general form for such problems is

$$(1.1) \quad \min_u |\Phi(u)| + H(u),$$

where $|\cdot|$ denotes the L1-norm and both $|\Phi(u)|$ and $H(u)$ are convex functions. Many important problems in imaging science (and other computational areas) can be posed as L1-regularized optimization problems. Some common examples of this include the following:

$$(1.2) \quad \text{“TV/ROF denoising”}: \min_u \|u\|_{BV} + \frac{\mu}{2} \|u - f\|_2^2,$$

$$(1.3) \quad \text{“basis pursuit/compressed sensing”}: \min_u J(u) + \frac{\mu}{2} \|Au - f\|_2^2,$$

where $J(u)$ is some regularizing functional, usually in the form of a bounded variation (BV) or Besov norm.

The Rudin–Osher–Fatemi (ROF) functional (1.2), despite its simple form, has proved to be very difficult to minimize by conventional methods. Total variation (TV) based image restoration was first introduced in [23]. In that paper, the authors proposed minimizing this energy using a gradient projection method. While this approach is simple, the nonlinearity and

*Received by the editors June 2, 2008; accepted for publication (in revised form) November 4, 2008; published electronically April 1, 2009. This research was supported by the National Science Foundation's GRFP program, as well as by ONR grant N000140710810 and the Department of Defense.

<http://www.siam.org/journals/siims/2-2/72589.html>

[†]Department of Mathematics, University of California at Los Angeles, Los Angeles, CA 90095 (tomgoldstein@gmail.com, sjo@math.ucla.edu).

poor conditioning of the problem make this approach very slow. Several authors have proposed improved time-stepping schemes that result in better performance, such as those presented in [26, 16]. A more efficient class of solvers are those based on Newton's method. One such algorithm was presented in [9], in which the preconditioned conjugate gradient method was used to invert the Hessian at each step. A somewhat more efficient implementation of a second-order method was proposed by Vogel in [25], where an algebraic multigrid preconditioner was used to accelerate the method.

Still, the most efficient solver for the ROF problem was proposed by Darbon and Sigelle in [10]. In this work, the ROF functional was approximated using an anisotropic BV norm. It was shown that, using this formulation, the resulting problem could be solved very quickly using graph cuts [3].

Problems of the form (1.3) have received a lot of attention recently because of the introduction of compressed sensing (CS) techniques, which allow high resolution images and signals to be reconstructed from a small amount of data [7, 6, 11]. This formulation has been shown to be useful in many areas, including medical imaging [17, 18], radar [1], and other signal processing applications. CS is based on the idea that a signal can be reconstructed from a very small number of measurements, provided that these measurements are obtained in the correct basis. The particular application of CS which we will focus on is magnetic resonance imaging (MRI). The goal of sparse MRI is to solve

$$\min_u J(u) \text{ such that } \|\mathcal{F}u - f\|_2 = 0.$$

Here, f represents the so-called CS data, which consists of samples of the Fourier transform of the unknown image. \mathcal{F} comprises a subset of the rows of a Fourier matrix, u represents the unknown image that we wish to reconstruct, and $J(\cdot)$ is a properly chosen L1-regularization term. The unconstrained formulation for this problem was introduced in [14], where a Bregman iterative approach [4] was used to obtain solutions to “denoising” problems of the form

$$(1.4) \quad \min_u J(u) \text{ such that } \|Fu - f\|_2 < \sigma.$$

Because of the presence of an L1-regularization term, optimization problems of the form (1.4) are still very difficult to solve. Several authors have applied classical optimization schemes, such as interior point methods, to problems of this form. In [15], CS problems were reformulated as quadratic programming problems which were then solved using the code “l1_ls,” which is claimed to be one of the most efficient solvers for general CS problems. Another notable interior point approach is the code “l1-magic,” which formulates a CS problem as a second-order cone program and enforces inequality constraints using a logarithmic barrier potential [7].

In the special case where the CS problem can be written in the form

$$(1.5) \quad \min_u |u| \text{ such that } \|Au - f\|_2 < \sigma,$$

one can use a relatively new class of methods that reduce the CS problem to a set of simpler problems using linearization. The first of these methods is the “fixed point continuation”

(FPC) method introduced in [13] by Hale, Yin, and Zhang. This method solves the unconstrained problem

$$(1.6) \quad \min_u |u| + \frac{\mu}{2} \|Au - f\|_2^2$$

by iteratively performing gradient descent steps.

By applying the Bregman iteration scheme in [22, 14], it is also possible to solve the constrained problem (1.5) using FPC/Bregman. This is done by iteratively solving the unconstrained problem (1.6) and then modifying the value of f used in the next iteration.

Rather than solving the unconstrained problem and then performing a Bregman update separately, these two steps were elegantly combined in the linearized Bregman algorithm [30]. Linearized Bregman solves the problem (1.5) by iteratively solving

$$\begin{aligned} v^{k+1} &= v^k + A^T(f - Au^k), \\ u^{k+1} &= \delta * \text{shrink}(v^{k+1}, 1/\mu) \end{aligned}$$

for $k = 1, 2, \dots, n$. The algorithm is terminated when the denoising constraint in (1.5) is met. It was shown in [5, 21] that u^n will be a suitable approximation to the solution of (1.5) provided that appropriate values are chosen for the parameters μ and δ .

While the FPC and linearized Bregman algorithms are extremely efficient, they can solve only problems which can be put into the form (1.6). Because the gradient operator is not invertible, these algorithms cannot be used to solve problems involving the BV norm. Also, these schemes cannot solve optimization problems involving multiple L1-regularization terms. For these reasons, it is difficult to apply the FPC and linearized Bregman methods to image processing problems. For example, it has been noted by many authors [17, 24] that optimal MRI reconstruction is obtained using a combination of the BV norm and the Besov norm, $B_{1,1}$. In this case, this problem can be written as

$$\min_u \|u\|_{BV} + \|u\|_H + \frac{\mu}{2} \|\mathcal{F}u - f\|_2^2,$$

where $\|\cdot\|_{BV} = \|\nabla u\|_1$ is the BV norm and $\|u\|_H = \|Hu\|_1$ denotes the Besov norm with respect to the Haar wavelet transform. We will later address this particular form of the CS problem.

In this paper, we will present a general technique that can be used to solve most common L1-regularized problems efficiently. Furthermore, the optimization scheme we employ can be generalized to solve a very broad range of equality-constrained optimization problems, some of which may be difficult to solve by existing techniques.

The contents of this paper are organized as follows: We begin with a brief discussion of constrained optimization problems and classical penalty function methods. We then review the concept of Bregman iteration and use it to derive a general principle for solving constrained optimization problems. We will then define the “split Bregman” method and show how it can be used to solve the general L1-regularized optimization problem (1.1). Finally, we will apply the split Bregman technique to TV denoising and CS problems to demonstrate its efficiency. The latter may involve BV, Besov, or multiple regularizers.

1.1. Constrained optimization problems. Consider a convex energy functional, E , and some linear function, $A : R^n \rightarrow R^m$. We wish to solve the generalized constrained optimization problem

$$(1.7) \quad \min_u E(u) \text{ such that } Au = b.$$

This problem can be very difficult to solve directly if E is nondifferentiable.

In order to make (1.7) simpler to solve, we wish to convert it into an unconstrained optimization problem. One common method for doing this is to use a penalty function/continuation method, which approximates (1.7) by a problem of the form

$$(1.8) \quad \min_u E(u) + \frac{\lambda_k}{2} \|Au - b\|_2^2,$$

where $\lambda_1 < \lambda_2 < \dots < \lambda_N$ is an increasing sequence of penalty function weights [2, 20]. In order to enforce that $H(u) \approx 0$, we must choose λ_N to be extremely large.

Unfortunately, for many problems, choosing a large value for λ makes (1.8) extremely difficult to solve numerically. We often wish to solve (1.8) by a Newton-type method, which requires us to invert the Hessian of the objective function. However, as $\lambda_k \rightarrow \infty$, the condition number of the Hessian approaches infinity, making it impractical to use fast iterative methods (such as conjugate gradient or Gauss–Seidel methods) [2, 20]. Also, for many applications, λ_k must be increased in very small steps, making the method less efficient.

In the next section, we will show that Bregman iteration can be used to reduce (1.7) to a short sequence of unconstrained problems. In this sense, Bregman iteration is an alternative to conventional penalty function methods.

2. Bregman iteration. Bregman iteration is a concept that originated in functional analysis for finding extrema of convex functionals [4]. Bregman iteration was first used in image processing by Osher et al. in [22], where it was applied to the ROF model for TV denoising. Bregman iteration has also been applied to solve the basis pursuit problem in [30, 5, 21], and was subsequently applied to medical imaging problems in [14]. Rather than focus on specific applications, we will present here a general formulation of this technique.

We begin with the concept of “Bregman distance.” The Bregman distance associated with a convex function E at the point v is

$$D_E^p(u, v) = E(u) - E(v) - \langle p, u - v \rangle,$$

where p is in the subgradient of E at v . Clearly, this is not a distance in the usual sense because it is not in general symmetric. However, it does measure closeness in the sense that $D_E^p(u, v) \geq 0$, and $D_E^p(u, v) \geq D_E^p(w, v)$ for w on the line segment between u and v .

Again, consider two convex energy functionals, E and H , defined over R^n with $\min_{u \in R^n} H(u) = 0$. The associated unconstrained minimization problem is

$$(2.1) \quad \min_u E(u) + \lambda H(u).$$

We can modify this problem by iteratively solving

$$(2.2) \quad u^{k+1} = \min_u D_E^p(u, u^k) + \lambda H(u)$$

$$(2.3) \quad = \min_u E(u) - \langle p^k, u - u^k \rangle + \lambda H(u),$$

as was suggested by Bregman in [4].

For simplicity, we will assume that H is differentiable. In this case, we have that $0 \in \partial(D_E^p(u, u^k) + \lambda H(u))$ where this subdifferential is evaluated at u^{k+1} . Since $p^{k+1} \in \partial E(u^{k+1})$ at this location, we have that

$$p^{k+1} = p^k - \nabla H(u^{k+1}).$$

In [22], the authors analyze the convergence of Bregman iterative schemes. In particular, it is shown that, under fairly weak assumptions on E and H , $H(u^k) \rightarrow 0$ as $k \rightarrow \infty$.

Two particular convergence results from [22] are especially relevant here, and so we restate them.

Theorem 2.1. *Assume that E and H are convex functionals, and that H is differentiable. We also assume that solutions to the subproblems in (2.2) exist. We then have*

$$(1) \text{ monotonic decrease in } H: H(u^{k+1}) \leq H(u^k),$$

$$(2) \text{ convergence to a minimizer of } H: H(u^k) \leq H(u^*) + J(u^*)/k.$$

In addition to these convergence results, Bregman iteration has several nice denoising properties which are discussed and proved in [22] and [14].

2.1. Constrained optimization via Bregman iteration. In this section, we present a method for solving a wide variety of constrained optimization problems without using continuation. We first show that Bregman iteration can be used to solve the constrained problem (1.7). We will then discuss a simplified form of Bregman iteration, which is equivalent to “adding the noise back” as is done with ROF denoising [22]. Finally, we will discuss the convergence properties of this method.

We wish to solve

$$(2.4) \quad \min_u E(u) \text{ such that } Au = b$$

for some linear operator A and vector b . To apply formula (2.1), we make this into an unconstrained problem using a quadratic penalty function:

$$(2.5) \quad \min_u E(u) + \frac{\lambda}{2} \|Au - b\|_2^2.$$

For small λ , the penalty function does not accurately enforce the constraint. The conventional solution to this problem is to let $\lambda \rightarrow \infty$. Rather, we apply the Bregman iteration (2.2) and iteratively minimize:

$$(2.6) \quad u^{k+1} = \min_u D_E^p(u, u^k) + \frac{\lambda}{2} \|Au - b\|_2^2$$

$$(2.7) \quad = \min_u E(u) - \langle p^k, u - u^k \rangle + \frac{\lambda}{2} \|Au - b\|_2^2,$$

$$(2.8) \quad p^{k+1} = p^k - \lambda A^T (Au^{k+1} - b).$$

Bregman iterations of this form were considered in [30] and [22]. Here, it is shown that, when A is linear, this seemingly complicated iteration is equivalent to the simplified method:

$$(2.9) \quad u^{k+1} = \min_u E(u) + \frac{\lambda}{2} \|Au - b^k\|_2^2,$$

$$(2.10) \quad b^{k+1} = b^k + b - Au^k.$$

In other words, we simply add the error in the constraint back to the right-hand side. This is the analogue of “adding back the noise” in the ROF model for TV denoising [22].

Because of the equivalence of (2.6)–(2.8) and (2.9)–(2.10), and the convergence results of Theorem 2.1, we have that

$$(2.11) \quad \lim_{k \rightarrow \infty} Au^k = b,$$

where convergence is in the 2-norm sense. In other words, for large k , the iterates u^k satisfy the constraint condition to an arbitrarily high degree of accuracy.

We now need to show that a solution, u^* , of $Au = b$ obtained through (2.9)–(2.10) is indeed a solution to the original constrained problem (2.4). Note that Bregman iteration was used to solve a constrained optimization problem in [30]. In this paper, the authors assume a specific form for the operator A . Here, we broaden this result and present a very simple proof.

Theorem 2.2. *Let $H : R^n \rightarrow R$ be convex. Let $A : R^n \rightarrow R^m$ be linear. Consider the algorithm (2.9)–(2.10). Suppose that some iterate, u^* , satisfies $Au^* = b$. Then u^* is a solution to the original constrained problem (2.4).*

Proof. Let u^* and b^* be such that $Au^* = b$ and

$$(2.12) \quad u^* = \min_u E(u) + \frac{\lambda}{2} \|Au - b^*\|_2^2.$$

Let \hat{u} be a true solution to (2.4). Then $Au^* = b = A\hat{u}$, which implies that

$$(2.13) \quad \|Au^* - b^*\|_2^2 = \|A\hat{u} - b^*\|_2^2.$$

Because u^* satisfies (2.12), we have

$$(2.14) \quad E(u^*) + \frac{\lambda}{2} \|Au^* - b^*\|_2^2 \leq E(\hat{u}) + \frac{\lambda}{2} \|A\hat{u} - b^*\|_2^2.$$

Finally, note that (2.13) and (2.14) together imply

$$E(u^*) \leq E(\hat{u}).$$

Because \hat{u} satisfies the original optimization problem, this inequality can be sharpened to an equality, showing that u^* solves (2.4). ■

This shows that, provided that (2.9)–(2.10) converges in the sense of (2.11), the iterates u^k will get arbitrarily close to a solution of the original constrained problem (1.7). Note the generality of the above theorem. The proof does not explicitly use the linearity of A , and in fact this condition is not required for the theorem to hold. The application of this method to problems in which A is not linear will be a subject of further research.

2.2. Advantages of Bregman iteration. This Bregman iteration technique has several advantages over traditional penalty function/continuation methods. First, Bregman iteration converges very quickly when applied to certain types of objective functions, especially for problems where E contains an L1-regularization term. For an explanation of why this is true, see the appendix. When Bregman iteration converges quickly, we need to solve only a small number of unconstrained problems.

The second (and perhaps most significant) advantage of Bregman iteration over continuation methods is that the value of λ in (2.1) remains constant. We can therefore choose a value for λ that minimizes the condition number of the subproblems, resulting in fast convergence for iterative optimization methods such as Newton or Gauss–Seidel.

Bregman iteration also avoids the problem of numerical instabilities that occur as $\lambda \rightarrow \infty$ when using continuation methods.

3. Split Bregman—A better formulation for L1-regularized problems. We will now apply the Bregman framework to solve the general L1-regularized optimization problem (1.1). In the discussion that follows, we shall assume $H(\cdot)$ and $|\Phi(\cdot)|$ to be convex functionals. We shall also assume $\Phi(\cdot)$ to be differentiable.

The key to our method is that we will “decouple” the L1 and L2 portions of the energy in (1.1). This split formulation follows that proposed in [27], where a similar technique is applied to L1-regularized deconvolutions. Rather than considering (1.1), we will consider the problem

$$(3.1) \quad \min_{u,d} |d| + H(u) \text{ such that } d = \Phi(u).$$

This problem is clearly equivalent to (1.1). To solve this problem, first convert it into an unconstrained problem:

$$(3.2) \quad \min_{u,d} |d| + H(u) + \frac{\lambda}{2} \|d - \Phi(u)\|_2^2.$$

This is where our method departs from that of [27]. If we let $E(u, d) = |d| + H(u)$ and define $A(u, d) = d - \Phi(u)$, then we can see that (3.2) is simply an application of (2.5). To enforce the constraint condition we now plug this problem into the above Bregman formulation (2.6)–(2.8):

$$(3.3) \quad (u^{k+1}, d^{k+1}) = \min_{u,d} D_E^p(u, u^k, d, d^k) + \frac{\lambda}{2} \|d - \Phi(u)\|_2^2$$

$$(3.4) \quad = \min_{u,d} E(u, d) - \langle p_u^k, u - u^k \rangle - \langle p_d^k, d - d^k \rangle + \frac{\lambda}{2} \|d - \Phi(u)\|_2^2,$$

$$(3.5) \quad p_u^{k+1} = p_u^k - \lambda(\nabla \Phi)^T(\Phi u^{k+1} - d^{k+1}),$$

$$(3.6) \quad p_d^{k+1} = p_d^k - \lambda(d^{k+1} - \Phi u^{k+1}).$$

When we apply the simplification presented in (2.9)–(2.10), we get the following elegant two-phase algorithm:

Split Bregman Iteration

$$(3.7) \quad (u^{k+1}, d^{k+1}) = \min_{u,d} |d| + H(u) + \frac{\lambda}{2} \|d - \Phi(u) - b^k\|_2^2,$$

$$(3.8) \quad b^{k+1} = b^k + (\Phi(u^{k+1}) - d^{k+1}).$$

We have reduced the L1-regularized problem (1.1) to a sequence of unconstrained optimization problems and Bregman updates. It may not be immediately clear why this algorithm is so effective. We will see in the next section that this formulation of the problem is much easier to compute than the conventional formulation for an L1-regularized problem.

3.1. Iterative minimization. In order to implement the algorithm (3.7), we must be able to solve the problem

$$(3.9) \quad (u^{k+1}, d^{k+1}) = \min_{u,d} |d| + H(u) + \frac{\lambda}{2} \|d - \Phi(u) - b^k\|_2^2.$$

Because of the way that we have “split” the L1 and L2 components of this functional, we can perform this minimization efficiently by iteratively minimizing with respect to u and d separately. The two steps we must perform are

$$\text{Step 1: } u^{k+1} = \min_u H(u) + \frac{\lambda}{2} \|d^k - \Phi(u) - b^k\|_2^2,$$

$$\text{Step 2: } d^{k+1} = \min_d |d| + \frac{\lambda}{2} \|d - \Phi(u^{k+1}) - b^k\|_2^2.$$

The speed of the Bregman splitting method is largely dependent on how quickly we can solve each of these two subproblems.

To solve Step 1, note that because we have “decoupled” u from the L1 portion of the problem, the optimization problem that we must solve for u^k is now differentiable. We can thus use a wide variety of optimization techniques to solve this problem. The particular method used to solve this optimization problem depends on the exact nature of H , but for many common problems either Gauss–Seidel or Fourier transform methods can be used. For rare problems in which Φ has little structure, a few steps of a conjugate gradient method can be used to approximately solve this problem.

In Step 2 of the above algorithm, there is no coupling between elements of d . We can explicitly compute the optimal value of d using shrinkage operators. We simply compute

$$d_j^{k+1} = \text{shrink}(\Phi(u)_j + b_j^k, 1/\lambda),$$

where

$$(3.10) \quad \text{shrink}(x, \gamma) = \frac{x}{|x|} * \max(|x| - \gamma, 0).$$

This shrinkage is extremely fast and requires only a few operations per element of d^{k+1} .

3.2. Implementation of the proposed algorithm. When we place the iterative minimization scheme into the process described in (3.7), we get the following:

Generalized Split Bregman Algorithm

```

While  $\|u^k - u^{k-1}\|_2 > tol$ 
  For  $n = 1$  to  $N$ 
     $u^{k+1} = \min_u H(u) + \frac{\lambda}{2} \|d^k - \Phi(u) - b^k\|_2^2$ 
     $d^{k+1} = \min_d |d| + \frac{\lambda}{2} \|d - \Phi(u^{k+1}) - b^k\|_2^2$ 
  end
   $b^{k+1} = b^k + (\Phi(u^{k+1}) - d^{k+1})$ 
end

```

We have found that it is not desirable to solve the first subproblem in (3.7) to full convergence. Intuitively, the reason for this is that if the error in our solution for this subproblem is small compared to $\|b^k - b^*\|_2$, then this extra precision will be “wasted” when the Bregman parameter is updated. In fact, we have found empirically that for many applications optimal efficiency is obtained when only one iteration of the inner loop is performed (i.e., $N = 1$ in the above algorithm). Even when we solve for u^{k+1} only approximately (e.g., by using a few steps of an iterative method), the above algorithm still converges.

To understand why the split Bregman algorithm is so robust to numerical imprecision, we must examine the results of Theorem 2.2. Using this theorem, it is easy to show that any fixed point of the split Bregman algorithm is indeed a minimizer of the original constrained problem (3.1), even if we use inexact iterative methods for each subproblem. Let (u^*, b^*) be a fixed point of (3.7)–(3.8) that also satisfies (3.7). The fixed point satisfies $b^* = b^* + \Phi u^* - d^*$, which implies that $d^* = \Phi u^*$. This result, together with (3.7), satisfies the conditions of Theorem 2.2, which shows that (u^*, b^*) is a solution of the constrained problem (3.1).

4. Applications. We will now illustrate how to use the split Bregman framework by discussing several applications.

4.1. TV denoising. TV denoising is considered to be one of the best denoising models, but also one of the hardest to compute. In this section, we will show how the split Bregman technique can be used to solve this problem in a way that is not only simple but also extremely efficient. Furthermore, this model can solve the isotropic TV minimization problem (a superior TV model which cannot be solved using the popular graph cuts method). We will treat this as a two-dimensional problem to demonstrate that the split Bregman method applies to problems with more than one L1-regularization term.

We begin by addressing the anisotropic problem

$$(4.1) \quad \min_u |\nabla_x u| + |\nabla_y u| + \frac{\mu}{2} \|u - f\|_2^2.$$

To apply Bregman splitting, we first replace $\nabla_x u$ by d_x and $\nabla_y u$ by d_y . This yields the constrained problem

$$\min_u |d_x| + |d_y| + \frac{\mu}{2} \|u - f\|_2^2 \text{ such that } d_x = \nabla_x u \text{ and } d_y = \nabla_y u.$$

To weakly enforce the constraints in this formulation, we add penalty function terms as was done in (3.2). This yields

$$\min_{d_x, d_y, u} |d_x| + |d_y| + \frac{\mu}{2} \|u - f\|_2^2 + \frac{\lambda}{2} \|d_x - \nabla_x u\|_2^2 + \frac{\lambda}{2} \|d_y - \nabla_y u\|_2^2.$$

Finally, we strictly enforce the constraints by applying the Bregman iteration (3.7) to get

$$\min_{d_x, d_y, u} |d_x| + |d_y| + \frac{\mu}{2} \|u - f\|_2^2 + \frac{\lambda}{2} \|d_x - \nabla_x u - b_x^k\|_2^2 + \frac{\lambda}{2} \|d_y - \nabla_y u - b_y^k\|_2^2,$$

where the proper values of b_x^k and b_y^k are chosen through Bregman iteration.

To solve this minimization problem, we apply the generalized split Bregman algorithm described in section 3.2, which requires us to solve the subproblem

$$u^{k+1} = \min_u \frac{\mu}{2} \|u - f\|_2^2 + \frac{\lambda}{2} \|d_x^k - \nabla_x u - b_x^k\|_2^2 + \frac{\lambda}{2} \|d_y^k - \nabla_y u - b_y^k\|_2^2,$$

which has the optimality condition

$$(4.2) \quad (\mu I - \lambda \Delta) u^{k+1} = \mu f + \lambda \nabla_x^T (d_x^k - b_x^k) + \lambda \nabla_y^T (d_y^k - b_y^k).$$

In order to achieve optimal efficiency, we wish to use a fast iterative algorithm to get approximate solutions to this system. Because the system is strictly diagonally dominant, the most natural choice is the Gauss–Seidel method. The Gauss–Seidel solution to this problem can be written componentwise as $u_{i,j}^{k+1} = G_{i,j}^k$, where

$$G_{i,j}^k = \frac{\lambda}{\mu + 4\lambda} (u_{i+1,j}^k + u_{i-1,j}^k + u_{i,j+1}^k + u_{i,j-1}^k + d_{x,i-1,j}^k - d_{x,i+1,j}^k + d_{y,i,j-1}^k - d_{y,i,j}^k - b_{x,i-1,j}^k + b_{x,i+1,j}^k - b_{y,i,j-1}^k + b_{y,i,j}^k) + \frac{\mu}{\mu + 4\lambda} f_{i,j}.$$

Using this solver, the split Bregman algorithm is written as follows:

Split Bregman Anisotropic TV Denoising

Initialize: $u^0 = f$, and $d_x^0 = d_y^0 = b_x^0 = b_y^0 = 0$

While $\|u^k - u^{k-1}\|_2 > tol$

$u^{k+1} = G^k$

$d_x^{k+1} = shrink(\nabla_x u^{k+1} + b_x^k, 1/\lambda)$

$d_y^{k+1} = shrink(\nabla_y u^{k+1} + b_y^k, 1/\lambda)$

$b_x^{k+1} = b_x^k + (\nabla_x u^{k+1} - d_x^{k+1})$

$b_y^{k+1} = b_y^k + (\nabla_y u^{k+1} - d_y^{k+1})$

end

Note that the “for” loop in the generalized split Bregman algorithm is absent here. We have found that this algorithm attains optimal efficiency when this loop is executed only once per iteration, and have therefore removed the loop for clarity. It may be necessary to include this loop in applications where high precision results are needed.

The isotropic TV model can also be minimized using the split Bregman technique. In this case, we wish to solve

$$\min_u \sum_i \sqrt{(\nabla_x u)_i^2 + (\nabla_y u)_i^2} + \frac{\mu}{2} \|u - f\|_2^2.$$

Just as we did for the anisotropic problem, we will split the L1 and L2 components of the problem by setting $d_x \approx \nabla_x u$ and $d_y \approx \nabla_y u$. The split Bregman formulation of the problem then becomes

$$\min_{u, d_x, d_y} \|(d_x, d_y)\|_2 + \frac{\mu}{2} \|u - f\|_2^2 + \frac{\lambda}{2} \|d_x - \nabla_x u - b_x\|_2^2 + \frac{\lambda}{2} \|d_y - \nabla_y u - b_y\|_2^2,$$

where

$$(4.3) \quad \|(d_x, d_y)\|_2 = \sum_{i,j} \sqrt{d_{x,i,j}^2 + d_{y,i,j}^2}.$$

Note that the d_x and d_y variables do not decouple as they did in the anisotropic case. This changes the way in which we must treat these variables. In order to apply the iterative minimization procedure to this problem, we must solve the subproblem

$$(d_x^{k+1}, d_y^{k+1}) = \min_{d_x, d_y} \|(d_x, d_y)\|_2 + \frac{\lambda}{2} \|d_x - \nabla_x u - b_x\|_2^2 + \frac{\lambda}{2} \|d_y - \nabla_y u - b_y\|_2^2.$$

Despite the fact that the variables d_x and d_y do not decouple as they did in the anisotropic case, we can still explicitly solve the minimization problem for (d_x^{k+1}, d_y^{k+1}) using a generalized shrinkage formula [27]:

$$(4.4) \quad d_x^{k+1} = \max \left(s^k - \frac{1}{\lambda}, 0 \right) \frac{\nabla_x u^k + b_x^k}{s^k},$$

$$(4.5) \quad d_y^{k+1} = \max \left(s^k - \frac{1}{\lambda}, 0 \right) \frac{\nabla_y u^k + b_y^k}{s^k},$$

where

$$(4.6) \quad s^k = \sqrt{|\nabla_x u^k + b_x^k|^2 + |\nabla_y u^k + b_y^k|^2}.$$

If we apply Bregman iteration to this problem, we get the minimization algorithm for the isotropic TV functional:

Split Bregman Isotropic TV Denoising

Initialize: $u^0 = f$, and $d_x^0 = d_y^0 = b_x^0 = b_y^0 = 0$

While $\|u^k - u^{k-1}\|_2 > tol$

$$u^{k+1} = G_{i,j}^k$$

$$d_x^{k+1} = \max(s^k - \frac{1}{\lambda}, 0) \frac{\nabla_x u^k + b_x^k}{s^k}$$

$$d_y^{k+1} = \max(s^k - \frac{1}{\lambda}, 0) \frac{\nabla_y u^k + b_y^k}{s^k}$$

$$b_x^{k+1} = b_x^k + (\nabla_x u^{k+1} - d_x^{k+1})$$

$$b_y^{k+1} = b_y^k + (\nabla_y u^{k+1} - d_y^{k+1})$$

end

where s^k is defined above by (4.6).

4.2. Fast CS for image reconstruction. CS is an emerging area of medical imaging, and many people predict that it will someday be a commonplace tool for radiologists. Consequently, fast algorithms for this problem are extremely desirable.

The exact formulation of the CS optimization problem depends somewhat on the application being considered. For demonstration purposes, we shall focus on the application of CS for sparse MRI. We choose this application not only because of the great success of CS in this field, but also because the difficulties of this problem allow us to demonstrate the versatility of the split Bregman method.

The general form for the sparse MRI reconstruction problem is presented and discussed in [17, 14, 15]. We choose to write this problem in the form

$$(4.7) \quad \min_u J(u) \text{ such that } \|R\mathcal{F}u - f\|_2^2 < \sigma^2,$$

where \mathcal{F} represents the Fourier transform matrix, f represents the observed “ k -space” data, and σ represents the variance of the signal noise. The matrix R represents a “row selector” matrix, which comprises a subset of the rows of an identity matrix. Also, $J(u)$ represents some L1-regularization term.

In [30] it was shown that, using a Bregman iteration technique, problem (4.7) could be reduced to a sequence of unconstrained problems of the form

$$(4.8) \quad u^{k+1} = \min_u J(u) + \frac{\mu}{2} \|R\mathcal{F}u - f^k\|_2^2,$$

$$(4.9) \quad f^{k+1} = f^k + f - R\mathcal{F}u^{k+1}.$$

It is this unconstrained problem that we wish to solve using the split Bregman technique.

To be concrete about the regularization term that we are using, we will now choose a specific form for $J(\cdot)$. Several authors have observed that superior image reconstructions occur when a hybrid of TV and Besov regularizers is used. Following these authors, we choose $J(u) = \|u\|_{BV} + \|u\|_{B_{1,1}} = |\nabla u| + |Wu|$, where W represents the discrete Haar orthogonal wavelet transform. While it may seem that the BV and Haar regularizers are very similar, we have found that the inclusion of multiple regularizers helps to ensure accurate reconstruction of smooth images. Since the inclusion of such “overcomplete” transforms has little effect on the convergence rate of the split Bregman algorithm, and both transforms can be evaluated quickly, there is little additional cost to including such regularizers.

Note that u can now take on complex values, and so we must be precise about our notation. In the following, we have $|v| = \|v\|_1 = \sum_i \sqrt{v_i^H v_i}$, where v_i^H denotes the Hermitian transpose of the vector v_i . Using this definition of the L1-norm, we may define $\|v\|_{BV} = |\nabla v| = \sum_i \sqrt{|\nabla_x v|^2 + |\nabla_y v|^2}$.

To apply the split Bregman method to this problem, we first make the replacements $w \leftarrow Wu$, $d_x \leftarrow \nabla_x u$, and $d_y \leftarrow \nabla_y u$. The split formulation of the problem then becomes

$$\begin{aligned} \min_{u, d_x, d_y, w} & \| (d_x, d_y) \|_2 + |w| + \frac{\mu}{2} \|R\mathcal{F}u - f\|_2^2 + \frac{\lambda}{2} \|d_x - \nabla_x u - b_x\|_2^2 \\ & + \frac{\lambda}{2} \|d_y - \nabla_y u - b_y\|_2^2 + \frac{\gamma}{2} \|w - Wu - b_w\|_2^2, \end{aligned}$$

where we have used the short-hand notation

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

which is the complex analogue of (4.3). Note that d_x , d_y , b_x , and b_y are now complex-valued, and are obtained by applying the difference operator to the real and complex parts of u separately.

We then decompose this into subproblems using the generalized split Bregman algorithm of section 3.2. We may use the generalized shrinkage formula (4.4)–(4.6) to solve for the optimal values of d_x and d_y , and the standard shrinkage formula (3.10) to solve for the optimal value of w . To find the optimal value of u , we must solve the optimization subproblem

$$u^{k+1} = \min_u \frac{\mu}{2} \|\mathcal{F}u - f\|_2^2 + \frac{\lambda}{2} \|d_x^k - \nabla_x u - b_x^k\|_2^2 + \frac{\lambda}{2} \|d_y^k - \nabla_y u - b_y^k\|_2^2 + \frac{\gamma}{2} \|w^k - Wu - b_w^k\|_2^2.$$

Because this subproblem is differentiable, optimality conditions for u^{k+1} are easily derived. By differentiating with respect to u and setting the result equal to zero, we get the update rule

$$(\mu \mathcal{F}^T R^T R \mathcal{F} + \lambda \nabla_x^T \nabla_x + \lambda \nabla_y^T \nabla_y + \gamma W^T W) u^{k+1} = rhs^k,$$

where

$$rhs^k = \mu \mathcal{F}^T R f + \lambda \nabla_x^T (d_x^k - b_x) + \lambda \nabla_y^T (d_y^k - b_y) + \gamma W^T (w - b_w)$$

represents the right-hand side in the above equation.

We now take advantage of the identities $\nabla^T \nabla = -\Delta$, $W^T W = I$, and $\mathcal{F}^T = \mathcal{F}^{-1}$ to get

$$(\mu \mathcal{F}^T R^T R \mathcal{F} - \lambda \Delta + \gamma I) u^{k+1} = rhs^k.$$

Therefore, the system that must be inverted to solve for u^{k+1} is circulant. We can thus write the system as $\mathcal{F}^{-1} K \mathcal{F}$, where K is the diagonal operator

$$K = (\mu R^T R - \lambda \mathcal{F} \Delta \mathcal{F}^{-1} + \gamma I).$$

Because of the circulant structure of this system, we can solve for the optimal value of u^{k+1} using only two Fourier transforms.

When we put all of these elements together, we get the following algorithm:

Unconstrained CS Optimization Algorithm

Initialize: $u^0 = \mathcal{F}^{-1} f$, and $d_x^0 = d_y^0 = w^0 = b_x^0 = b_y^0 = b_w^0 = 0$

While $\|u^k - u^{k-1}\|_2 > tol$

$$u^{k+1} = \mathcal{F}^{-1} K^{-1} \mathcal{F} rhs^k$$

$$d_x^{k+1} = \max(s^k - \frac{1}{\lambda}, 0) \frac{\nabla_x u^k + b_x^k}{s^k}$$

$$d_y^{k+1} = \max(s^k - \frac{1}{\lambda}, 0) \frac{\nabla_y u^k + b_y^k}{s^k}$$

$$w^{k+1} = shrink(Wu^{k+1} + b_w^k, \frac{1}{\gamma})$$

$$b_x^{k+1} = b_x^k + (\nabla_x u^{k+1} - d_x^{k+1})$$

$$b_y^{k+1} = b_y^k + (\nabla_y u^{k+1} - d_y^{k+1})$$

$$b_w^{k+1} = b_w^k + (Wu^{k+1} - w^{k+1})$$

end

where s^k is defined in (4.6).

Note that this algorithm solves only the unconstrained CS problem (4.8). To solve the constrained problem (4.7), we must replace f by f^k in the above algorithm. After approximately solving each unconstrained problem, we must apply the Bregman update rule

$f^{k+1} = f^k + f - R\mathcal{F}u^{k+1}$. When we embed the unconstrained algorithm inside of this outer Bregman update, we get the following algorithm:

Constrained CS Optimization Algorithm

Initialize: $u^0 = \mathcal{F}^{-1}f$, and $d_x^0 = d_y^0 = w^0 = b_x^0 = b_y^0 = b_w^0 = 0$

While $\|R\mathcal{F}u^k - f\|_2^2 > \sigma^2$

For $i = 1$ to N

$$u^{k+1} = \mathcal{F}^{-1}K^{-1}\mathcal{F} r h s^k$$

$$d_x^{k+1} = \max(s^k - \frac{1}{\lambda}, 0) \frac{\nabla_x u^k + b_x^k}{s^k}$$

$$d_y^{k+1} = \max(s^k - \frac{1}{\lambda}, 0) \frac{\nabla_y u^k + b_y^k}{s^k}$$

$$w^{k+1} = shrink(Wu^{k+1} + b_w^k, \frac{1}{\gamma})$$

$$b_x^{k+1} = b_x^k + (\nabla_x u^{k+1} - d_x^{k+1})$$

$$b_y^{k+1} = b_y^k + (\nabla_y u^{k+1} - d_y^{k+1})$$

$$b_w^{k+1} = b_w^k + (Wu^{k+1} - w^{k+1})$$

end

$$f^{k+1} = f^k + f - R\mathcal{F}u^{k+1}$$

end

The speed of this algorithm will largely depend on how many times the outer “while” loop needs to be executed. When parameter values are properly chosen, it has been found that the outer loop of this algorithm needs to be executed only a small number of times. Also, for imaging applications it is not necessary to solve each unconstrained subproblem entirely to numerical precision. As a result, this algorithm is very fast when properly chosen parameter values are used.

5. Numerical results.

5.1. TV denoising results. We will now examine the efficiency of the split Bregman approach using time trials. The split Bregman algorithm was implemented in C++ and compiled on a UNIX platform using the g++ compiler. Time trials were generated on an Intel Core 2 Duo desktop PC (E6850, 3.00 GHz).

We tested our method on two images: The first was a 256×256 synthetic image of two overlapping squares. The second was a 512×512 representation of the famous test image “Lena.” Both images were contaminated with noise ($\sigma = 15$). Denoising parameters were $\mu = 0.05$, $\lambda = 0.1$ for both images. In general, we have found that choosing $\lambda = 2\mu$ usually results in good convergence. Iterations were terminated when the condition $\|u^k - u^{k-1}\|/\|u^k\| < 5 \cdot 10^{-3}$ was met. Results of time trials for the isotropic ROF algorithm are shown in Table 1. For comparison, we also report the computation time of a graph cuts based solver [29, 12], which minimizes an anisotropic TV functional using either a 4-point stencil or a less anisotropic

Table 1
ROF computation times (sec).

Image	Split Bregman	Graph cuts (4-point)	Graph cuts (16-point)
256×256 blocks	0.0732	0.214	0.468
512×512 Lena	0.2412	0.709	1.51

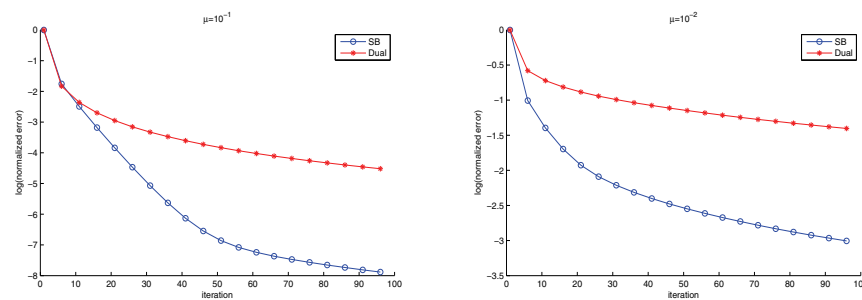


Figure 1. Error vs. iteration number for the split Bregman ROF minimization algorithm and for an iterative method based on the dual problem [8]. The error at iteration k is defined as $\|u^k - u^*\|$, where u^k is the approximation at iteration k , and u^* is the exact solution. Convergence results are for the test image “Lena” with Gaussian noise ($\sigma = 15$). Left: Results for $\mu = 0.1$. Right: Results for $\mu = 0.01$.

16-point stencil. Note that in addition to outperforming the graph cuts algorithm in terms of speed, the split Bregman method can minimize the isotropic functional and is thus less likely to introduce artifacts into the image.

The convergence speed of the split Bregman method can also be observed in Figure 1, where we have plotted the error vs. iteration number for the test image “Lena” as described above. To compare our method to another unregularized iterative method, we have also plotted the convergence curve for the gradient descent algorithm [8], which is based on the dual formulation of the ROF functional. We have found that both methods converge quickly for the first several iterations, but then slow down as the exact solution is reached. This stalling effect is, however, considerably less severe for the split Bregman method.

In addition to being fast, the split Bregman method has the advantage that, before convergence is reached, the intermediate images are smooth. Most of the image noise is eliminated during the first 10 iterations of the algorithm, and subsequent iterations serve to flatten “staircases.” For both the geometric image and “Lena,” almost 50 iterations of the split Bregman algorithm were needed to reach convergence within 0.5 intensity units per pixel. However, because intermediate results of the algorithm are smooth, it is extremely difficult to discern between the results after 10 iterations and the results after full convergence. This is illustrated using the image of Lena in Figure 2. This property is also illustrated in Figure 3, where we show the test image, as well as its cross sections, at various levels of convergence.

Note that the speed of this algorithm depends strongly on the fact that we have used Bregman iteration, rather than continuation, to enforce a constraint. Had we followed a more conventional path and let $\lambda \rightarrow \infty$, the problem (4.2) would have become ill-conditioned, and the Gauss–Seidel method would have stalled.

5.2. CS results. We now discuss the efficiency of the split Bregman CS algorithm using BV and/or $B_{1,1}$ as regularizers. We tested the split Bregman algorithm on two sample images, one synthetic and one real. Two different formulations of the CS problem were tested. First, the CS reconstruction was performed using only a Besov regularizer. Next, we reconstructed images using the hybrid Besov + BV algorithm discussed above. In the case where only the Besov regularizer is used, the optimization problem can be put into a particularly simple form, which allows it to be solvable by the linearized Bregman method. In this case, we compare the



Figure 2. Split Bregman ROF denoising of Lena. Top left: Original image. Top right: Noise contaminated with $\sigma = 25$. Bottom left: Denoised with 10 iterations of split Bregman. Bottom right: Denoised with 50 iterations of split Bregman.

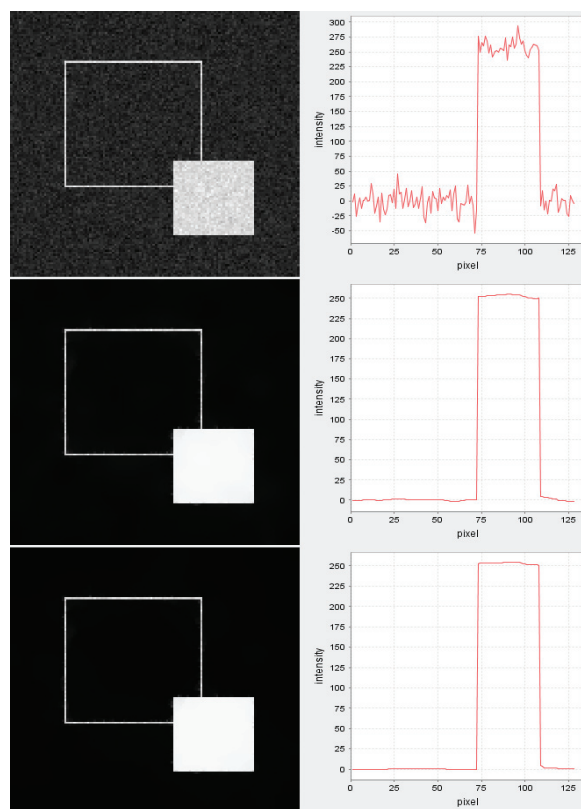


Figure 3. Split Bregman ROF denoising of the geometric test image. A cross section of each image is displayed on the right. Top: Noisy image with $\sigma = 15$. Middle: Denoised with 10 iterations of split Bregman. Bottom: Denoised with 50 iterations of split Bregman.

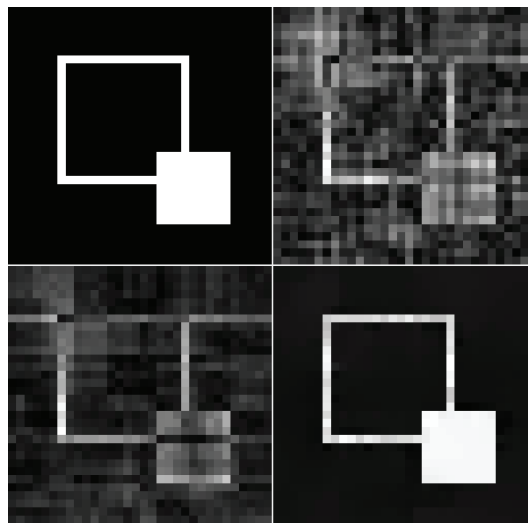


Figure 4. Synthetic image recovered from 50% of k -space data. The k -space data was contaminated with noise ($\sigma = 15$) prior to recovery. Top left: Original image reconstructed using the full k -space. Top right: Image reconstructed using 50% of k -space data. Reconstruction was done using the conventional method, which fills in missing samples with zeros. Bottom left: CS reconstruction using only the Besov regularizer. Bottom right: CS results using the hybrid TV+Besov regularizer. Note the superiority of the results when the hybrid regularizer is used.

Table 2

Iteration counts for CS algorithms.

Algorithm	Geometric	Geometric, $\sigma = 25$	MRI phantom
Linearized, Besov	367	35	51
Split, Besov	10	5	40
Split, Besov + BV	17	10	35

split Bregman algorithm to the linearized Bregman algorithm with “kicking” [21] for efficiency. Results of these reconstructions are shown in Figure 4.

In Table 2, we present the number of iterations of each algorithm required to reach the convergence criteria for each task described below. Note that with both the split Bregman and the linearized Bregman algorithm, two Fourier transforms are required per iteration. Because the Fourier transform is by far the most time-consuming step in each algorithm, we found that both algorithms require a similar amount of time to execute one iteration.

Example 1. We generated a sample synthetic image with dimensions 128×128 and pixel intensities ranging from 0 to 255. The image consists of two overlapping geometric shapes and has an extremely sparse gradient and Haar wavelet transform. The synthetic image has signal in both its real and imaginary components; however, for display purposes we show the absolute value of the image. For our first test, no noise was added to the CS data. The objective of this test was to recover the exact image using only 50% of the k -space data (sampled at random). The algorithm was run until $\|\mathcal{F}u - f^k\|_2/128^2 < 10^{-3}$. For the second test, the CS data was contaminated with noise ($\sigma = 25$), and each algorithm was run until the stopping criterion $\|\mathcal{F}u - f^k\|_2/128^2 < \sigma$ was met.

Example 2. To demonstrate the effectiveness of the split Bregman method in medical imaging, we tested the algorithm on real MRI data. For this purpose, we used a cross-sectional image of a saline phantom. In the original image acquisition, the entire k -space was sampled. To generate the CS data, we randomly and uniformly selected 30% of the k -space samples. Note that, while ideal for CS, this type of sampling is not practical for most MRI applications. This is because most pulse sequences acquire k -space data in some sort of geometric pattern, such as a spiral [19, 24, 17]. Because the focus of this paper is on numerics, and not on the details of image acquisition, we choose uniform random sampling for simplicity.

Several results from Table 2 are particularly noteworthy. First, we found that the linearized Bregman algorithm was fairly efficient for large σ , but converged very slowly for $\sigma = 10^{-3}$. The reason for this is that the linearized Bregman algorithm slowed down considerably as the method approached convergence. Also, to attain this high level of accuracy, fairly extreme values for the method parameters had to be chosen, which resulted in slow performance. For all tasks involving noisy data, the split Bregman method was comparable to, or even faster than, linearized Bregman. One reason for this is that the split Bregman method is very tolerant of large values of μ in (4.8). Because we were able to choose large values for this parameter, the number of outer “while” loops of the split Bregman CS algorithm was very small (8 or fewer for problems involving the phantom test image). Also, the inner “for” loop had to be executed only 5 times between each outer loop iteration. These convergence results are demonstrated using the MR phantom in Figure 5.

Note that the most significant advantage of the split Bregman method over other CS methods is its versatility. The split Bregman method can minimize energies involving the BV norm, which most CS algorithms (such as linearized Bregman and FPC) cannot. Also, the split Bregman method can solve optimization problems involving multiple regularization terms.

6. Conclusion. In this paper, we introduce the “split Bregman” framework for solving L1-regularized optimization problems. By applying the method to image denoising and CS problems, we showed that this method is a very efficient solver for many problems that are difficult to solve by other means. Besides its speed, our algorithm has several advantages: Because the split Bregman algorithm makes extensive use of Gauss–Seidel and Fourier transform methods, it is easily parallelizable. Also, it has a relatively small memory footprint compared to second-order methods that require explicit representations of the Hessian matrix. Both of these characteristics make split Bregman a practical algorithm for large-scale problems. Finally, the method is easy to code.

Appendix. Fast convergence of Bregman iteration. Here we give an informal but intuitive explanation of why Bregman iteration converges so quickly for L1 based problems. This is a generalization of what was done in [28], where Bregman iteration was applied to soft thresholding for wavelet denoising.

Consider the optimization problem

$$\min_u |u|_\epsilon \text{ such that } Au = f,$$

where

$$|u|_\epsilon = \sum_i \sqrt{u_j^2 + \epsilon}$$

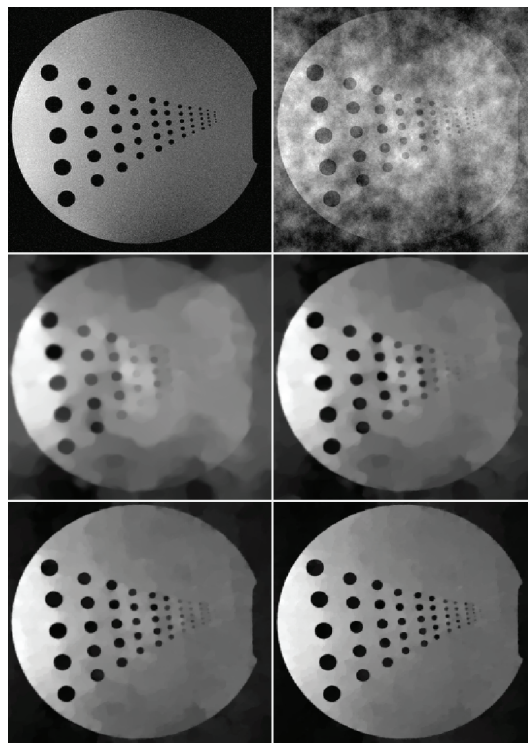


Figure 5. Split Bregman CS reconstruction of an MR image using 30% of the k -space data. Top left: Original image reconstructed using the full k -space. Top right: Image reconstructed using 30% of k -space data. Reconstruction was done using the conventional method, which fills in missing samples with zeros. Middle left: Results of split Bregman algorithm after 10 inner iterations (2 outer iterations). Middle right: Results of split Bregman algorithm after 20 inner iterations (4 outer iterations). Bottom left: Results after 30 inner iterations. Bottom right: Optimal results obtained after 40 inner iterations.

is a “smoothed out” variant of the L1-norm. We will need this smoothness to perform the analysis below. The Bregman update rule for this problem is

$$(A.1) \quad p^{k+1} - p^k + A^T(Au^{k+1} - f) = 0.$$

Now, because $J(\cdot) = |\cdot|_\epsilon$ is convex, and p^k and p^{k+1} are gradients of this functional at u^k and u^{k+1} , the mean value theorem tells us that

$$p^{k+1} - p^k = D^{k+\frac{1}{2}}(u^{k+1} - u^k),$$

where $D^{k+\frac{1}{2}}$ is a diagonal matrix such that

$$D_{i,i}^{k+\frac{1}{2}} = \epsilon \left((u_i^{k+\frac{1}{2}})^2 + \epsilon \right)^{-3/2}$$

for some $u^{k+\frac{1}{2}}$ between u^k and u^{k+1} .

Applying this formula to (A.1) yields

$$D^{k+\frac{1}{2}}(u^{k+1} - u^k) + A^T(Au^{k+1} - f) = 0.$$

We now let $Q^{k+\frac{1}{2}} = (D^{k+\frac{1}{2}})^{-1}$, multiply by $AQ^{k+\frac{1}{2}}$, and rearrange to get

$$(A.2) \quad Au^{k+1} - f = \left(I + AQ^{k+\frac{1}{2}}A^T\right)^{-1} \left(Au^k - f\right).$$

This equation gives us some insight into why Bregman iteration behaves as it does. When $u_i^{k+\frac{1}{2}}$ is large compared to ϵ , which occurs at “spikes” (or edges in the BV case), $Q_{i,i}^{k+\frac{1}{2}}$ is large, and u_i^k converges rapidly. Small values of u^k take more iterations to settle down. An important result of this is that for problems with shocks or edges, Bregman iteration puts these features in the right place almost immediately, unlike the continuation based method [27]. Resolving discontinuities is usually the most difficult part of any imaging task, and Bregman iteration is extremely well suited for this task.

Acknowledgment. We thank Jie Zheng for his helpful discussions regarding MR image processing.

REFERENCES

- [1] R. BARANIUK AND P. STEEGHS, *Compressive radar imaging*, in Proceedings of the 2007 IEEE Radar Conference, 2007, pp. 128–133.
- [2] S. BOYD AND L. VANDENBERGHE, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.
- [3] Y. BOYKOV, O. VEKSLER, AND R. ZABIH, *Fast approximate energy minimization via graph cuts*, IEEE Trans. Pattern Anal. Mach. Intell., 23 (2005), pp. 1222–1239.
- [4] L. BREGMAN, *The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex optimization*, USSR Comput. Math. and Math. Phys., 7 (1967), pp. 200–217.
- [5] J. F. CAI, S. OSHER, AND Z. SHEN, *Linearized Bregman iterations for compressed sensing*, Math. Comp., to appear.
- [6] E. J. CANDÉS AND J. ROMBERG, *Signal recovery from random projections*, in Computational Imaging III, Proc. SPIE 5674, International Society for Optical Engineering, Bellingham, WA, 2005, pp. 76–86.
- [7] E. J. CANDÉS, J. ROMBERG, AND T. TAO, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Trans. Inform. Theory, 52 (2006), pp. 489–509.
- [8] A. CHAMBOLLE, *Total variation minimization and a class of binary MRF models*, in Energy Minimization Methods in Computer Vision and Pattern Recognition, Springer-Verlag, Berlin, 2005, pp. 136–152.
- [9] T. F. CHAN, G. H. GOLUB, AND P. MULET, *A nonlinear primal-dual method for total variation-based image restoration*, SIAM J. Sci. Comput., 20 (1999), pp. 1964–1977.
- [10] J. DARBON AND M. SIGELLE, *A fast and exact algorithm for total variation minimization*, in IbPRIA, Lecture Notes in Comput. Sci. 3522, Springer-Verlag, Berlin, 2005, pp. 351–359.
- [11] D. DONOHO, *Compressed sensing*, IEEE Trans. Inform. Theory, 52 (2006), pp. 1289–1306.
- [12] D. GOLDFARB AND W. YIN, *Parametric Maximum Flow Algorithms for Fast Total Variation Minimization*, CAAM Technical Report TR07-09, Rice University, Houston, 2008.
- [13] E. HALE, W. YIN, AND Y. ZHANG, *A Fixed-Point Continuation Method for l_1 -Regularized Minimization with Applications to Compressed Sensing*, CAAM Technical Report TR07-07, Rice University, Houston, 2007.
- [14] L. HE, T.-C. CHANG, AND S. OSHER, *MR image reconstruction from sparse radial samples by using iterative refinement procedures*, in Proceedings of the 13th Annual Meeting of ISMRM, 2005, p. 696.
- [15] S. KIM, K. KOH, M. LUSTIG, S. BOYD, AND D. GERINVESKY, *A Method for Large-Scale l_1 -Regularized Least Squares Problems with Applications in Signal Processing and Statistics*, Technical report, Department of Electrical Engineering, Stanford University, Palo Alto, CA, 2007.

- [16] Y. LI AND F. SANTOSA, *An Affine Scaling Algorithm for Minimizing Total Variation in Image Enhancement*, Technical report TR94-1470, Center for Theory and Simulation in Science and Engineering, Cornell University, Ithaca, NY, 1994.
- [17] M. LUSTIG, D. DONOHO, AND J. PAULY, *Sparse MRI: The application of compressed sensing for rapid MR imaging*, Magn. Reson. Med., 58 (2007), pp. 1182–1195.
- [18] M. LUSTIG, J. H. LEE, D. L. DONOHO, AND J. M. PAULY, *Faster imaging with randomly perturbed undersampled spirals and L_1 reconstruction*, in Proceedings of the 13th Scientific Meeting of the ISMRM, 2005.
- [19] G. L. MARSEILLE, R. DE BEER, M. FUDERER, A. F. MEHLKOPF, AND D. VAN ORMONDT, *Nonuniform phase-encode distributions for MRI scan-time reduction*, J. Magn. Reson., 111 (1996), pp. 70–75.
- [20] J. NOCEDAL AND S. WRIGHT, *Numerical Optimization*, 2nd ed., Springer-Verlag, New York, 2006.
- [21] A. OSHER, Y. MAO, B. DONG, AND W. YIN, *Fast Linearized Bregman Iterations for Compressed Sensing and Sparse Denoising*, UCLA CAM Report, 08-37, University of California at Los Angeles, Los Angeles, 2008.
- [22] 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 (2005), pp. 460–489.
- [23] L. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
- [24] J. TRZASKO, A. MANDUCA, AND E. BORISCH, *Sparse MRI reconstruction via multiscale L_0 -continuation*, in Proceedings of the 14th IEEE/SP Workshop on Statistical Signal Processing, 2007, pp. 176–180.
- [25] C. VOGEL, *A multigrid method for total variation-based image denoising*, in Computation and Control, IV, Progr. Systems Control Theory 20, Birkhäuser-Boston, Boston, 1995, pp. 323–331.
- [26] C. R. VOGEL AND M. E. OMAN, *Iterative methods for total variation denoising*, SIAM J. Sci. Comput., 17 (1996), pp. 227–238.
- [27] Y. WANG, W. YIN, AND Y. ZHANG, *A Fast Algorithm for Image Deblurring with Total Variation Regularization*, CAAM Technical Report TR07-10, Rice University, Houston, 2007.
- [28] J. XU AND S. OSHER, *Iterative regularization and nonlinear inverse scale space applied to wavelet-based denoising*, IEEE Trans. Image Process., 16 (2007), pp. 534–544.
- [29] W. YIN, *PGC: A Preflow-Push Based Graph-Cut Solver*, Version 2.32, <http://www.caam.rice.edu/~optimization/L1/pgc/>.
- [30] W. YIN, S. OSHER, D. GOLDFARB, AND J. DARBON, *Bregman iterative algorithms for ℓ_1 -minimization with applications to compressed sensing*, SIAM J. Imaging Sci., 1 (2008), pp. 143–168.