Geometric Applications of the Split Bregman Method: Segmentation and Surface Reconstruction

Tom Goldstein · Xavier Bresson · Stanley Osher

Abstract Variational models for image segmentation have many applications, but can be slow to compute. Recently, globally convex segmentation models have been introduced which are very reliable, but contain TV-regularizers, making them difficult to compute. The previously introduced Split Bregman method is a technique for fast minimization of L1 regularized functionals, and has been applied to denoising and compressed sensing problems. By applying the Split Bregman concept to image segmentation problems, we build fast solvers which can out-perform more conventional schemes, such as duality based methods and graph-cuts. The convex segmentation schemes also substantially outperform conventional level set methods, such as the Chan-Vese level set-based segmentation algorithm. We also consider the related problem of surface reconstruction from unorganized data points, which is used for constructing level set representations in 3 dimensions. The primary purpose of this paper is to examine the effectiveness of "Split Bregman" techniques for solving these problems, and to compare this scheme with more conventional methods.

1 Introduction and Motivation

Segmentation and object extraction is one of most important tasks in image processing and computer vision. Many of the most general and effective segmentation methods can be written as variational/PDE based models. This category of variational models has been shown to be very effective in many applications, especially in the processing and analysis of medical images [29, 33, 46]. While there are many disparate approaches to image segmentation, this paper will focus on recently proposed methods which can be cast in the form of totally convex optimization problems. This convexity property allows segmentations to be computed using fast elliptic solvers. For other state-of-the-art approaches that do not fall

---

Dedicated to the memory of David Gottlieb.

T. Goldstein (✉) · X. Bresson · S. Osher
Department of Mathematics, UCLA, Box 951555, Los Angeles, CA 90095-1555, USA
e-mail: tomgold2@gmail.com

into this framework, see [13, 24, 44]. Several different variational frameworks for image segmentation have been proposed, most of which fall into one of two categories.

The first such category we will discuss is the geodesic active contour (GAC)/snakes model. Originally proposed in [30], snakes based segmentation identifies objects using an edge detector function, which takes on small values near boundaries (e.g. where the image gradient is large) and large values (typically near unity) where the image is smooth. Following the GAC method of Caselles, Kimmel, and Sapiro [14], this is accomplished using a curve evolution procedure by solving

$$\min_{C} \int_{C} g(\nabla f) ds, \tag{1}$$

where C represents the closed boundary curve, ∇f is the image gradient. The function g is the non-negative edge detector function. Models of this type have been studied in the context of segmentation and feature extraction by Kichenassamy et al. [46]. One common choice for the edge detector is

$$g(\xi) = \frac{1}{1 + \beta |\xi|^2}, \tag{2}$$

where β is a parameter that determines the detail level of the segmentation. It has been found that, by identifying a curve lying along edges in an image, model (1) extracts relevant semantic features from an image [14].

The calculus of variations then allows us to find the Euler-Lagrange equation for (1) and the minimization is carried out using the following curvature-minimizing gradient flow:

$$C_t = (g\kappa - \langle \nabla g, N \rangle) N, \tag{3}$$

where κ represents the curvature and N is the unit normal to the curve C [14].

Various numerical schemes have been proposed for computing this gradient flow. One of the most simple and versatile schemes is the level set method of Osher and Sethian [36, 37, 41]. In this method, the curve C is represented using an implicit level set function, ϕ. The curve evolution is then solved using the flow

$$\phi_t = \left(g \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} - \langle \nabla g, \nabla \phi \rangle \right) |\nabla \phi|. \tag{4}$$

Models of the form (1) have also been used for the related problem of reconstructing a surface from unorganized data, where g is replaced by the distance to the unorganized data set. Given a set of points $\{\mathbf{x}_i\}$ lying on some smooth surface, we wish to reconstruct an approximation of that surface. This problem is very difficult because it is, in general, ill-posed. Also, any solution to this problem must be able to handle a wide range of topological and geometric surface configurations. Explicit representation methods such as NURBS [39] rely on a parametrization of the surface, which requires a significant amount of a priori knowledge of the surface topology. Methods relying on triangularizations and Voronoi diagrams are limited to reconstructing piecewise linear surface, and become impractical if dynamic surface evolution is required [3, 4, 7]. A very nice approach that does not suffer from these difficulties is to use an implicit level-set representation [28, 35, 51, 52]. This can be accomplished by defining g to be the distance function of the set $\{\mathbf{x}_i\}$, and then minimizing the corresponding energy (1). As a result, this modified GAC model computes a surface that lies along the minimal contours of this distance function (i.e. a surface that passes close to the points $\{\mathbf{x}_i\}$). This model has been applied to 3 dimensional data in [51, 52].

The second approach to segmentation that we shall consider is a technique based on the Mumford Shah model [34]. In this model, an image is segmented by finding the best approximation of the image as a piecewise smooth function. One of the simplest and most successful formulations of this model is the "active contours without edges" (ACWE) model proposed by Chan and Vese [18]. The ACWE model seeks to approximate an image by a function taking on two values via the following optimization problem

$$\min_{\Omega, c_1, c_2} Per(\Omega) + \mu \int_{\Omega} (c_1 - f)^2 + \mu \int_{\Omega^c} (c_2 - f)^2, \tag{5}$$

where Ω is the extracted subset of the image, and $c_1, c_2 \in R$ represent the mean intensity inside and outside of the segmented region, respectively [18]. The length term $Per(\Omega)$ serves as a regularizer, which ensures that the curve $C = \partial\Omega$ has a well behaved boundary. By varying the parameter μ, the user can control the strength of the regularization, and thus the level of detail that is to be segmented. To compute the optimal region, Ω, Chan and Vese chose a level set representation. The energy (5) is then minimized using an alternating scheme in which the optimal curve C is computed, after which the mean intensities are updated. Using a level set function, u, and a regularized Heaviside function, H_ϵ, the optimal curve can be computed using the gradient flow [18]

$$\partial_t u = H'_\epsilon(u) \left(\nabla \cdot \frac{\nabla u}{|\nabla u|} - \mu \left((c_1 - f)^2 - (c_2 - f)^2 \right) \right), \tag{6}$$

for fixed c_1, and c_2.

Unfortunately, both the snakes and ACWE approach to segmentation suffer from substantial difficulties because neither model is convex. If one examines the snakes model (1), one will observe that the global minimum of this energy occurs when the curve C collapses into a single point (in this case the energy is zero). Therefore, any meaningful solution to this problem will only be a local minima. The ACWE model performs a minimization over all two-valued functions, which do not form a convex collection. As a result, the minimization problem (5) is non-convex and may have local minima. Indeed, it has been observed that the results of both of these models depend in a crucial way on the initial guess that is used. Furthermore, the level set evolution methods can sometimes get "stuck" at undesirable local minima. Note that, unlike the GAC/snakes approach, the ACWE model does have a meaningful global minimizer, although we have no efficient method for finding it in general.

To resolve the problems associated with non-convex models, several convex models for image segmentation have been proposed. The first of these models was originally inspired by Chan Esedoglu and Nikolova [20], and hybridizes the piecewise constant Mumford Shah model [34] with the globally convex ROF denoising model [40]. The second model we consider is a hybrid between the GAC model [14, 30] and the ROF model [40], which allows for fast implicit surface evolution.

In this manuscript, we consider fast numerical methods for computing the minimizers of these global models. We begin by introducing two convex image segmentation models, both based on total variation regularized energies. We then introduce the class of Split Bregman methods for L1 regularized problems, and explain how this methodology can be applied to segmentation problems. Finally, we present numerical examples showing the efficiency of the Split Bregman method for these problems, and comparisons with other numerical schemes.

1.1 Notation

When discussing discretized problems, we shall frequently use vector norms to avoid cumbersome summation. Consider a grid function, f_α, defined for all α in some index set Γ (for an $M \times N$ image, we take $\Gamma = \{(i,j) \in Z \times Z : 0 \le i \le M, 0 \le i \le N\}$). We shall use the following norm and inner product notations:

$$|f|_1 = \sum_{\alpha \in \Omega} |f_\alpha|, \qquad \|f\| = \left(\sum_{\alpha \in \Omega} |f_\alpha|^2 \right)^{\frac{1}{2}}, \qquad \langle f, g \rangle = \sum_{\alpha \in \Omega} f_\alpha \cdot g_\alpha.$$

We will also use "∇" in the discrete context to denote the first order discrete gradient operator and BV norms as follows:

$$(\nabla f)_{i,j} = (f_{i+1,j} - f_{i,j}, f_{i,j+1} - f_{i,j}), \qquad |\nabla f|_1 = \sum_{(i,j) \in \Omega} \|(\nabla f)_{i,j}\|.$$

In some circumstances, we wish to consider grid functions that are vector-valued at each pixel. For the sake of clarity, we shall use the "arrow" superscript to denote such vector-valued quantities. For example, we may write $\vec{d} = \nabla u$ to emphasize that the value of $\vec{d}$ at each grid location is an ordered pair.

When working with L1 minimization problems, we will make frequent use of the "shrink" operator. This operator has been used frequently in the wavelet literature [22], and was first adapted for use in L1 optimization in [45]. The shrink operator is defined at each point $\alpha \in \Gamma$ as follows

$$shrink(\vec{z}, \lambda)_\alpha = \max\{\|\vec{z}_\alpha\| - \lambda, 0\} \frac{\vec{z}_\alpha}{\|\vec{z}_\alpha\|}.$$

We also use the "weighted" shrink operator associated with the function g:

$$shrink_g(\vec{z}, \lambda)_\alpha = \max\{\|\vec{z}_\alpha\| - \lambda/g_\alpha, 0\} \frac{\vec{z}_\alpha}{\|\vec{z}_\alpha\|}.$$

Finally, the notation A^T will be used to denote the adjoint of the linear operator A.

2 Convex Methods for Image Segmentation

In [20], Chan et al. eliminate difficulties associated with these non-convex models by proposing an approach to segmentation that is based on convex energies. This globally convex segmentation (GCS) method is both easier to handle numerically, and is more reliable because it does not get "stuck" at local minima. The GCS formulation is based on the observation that the steady state solution of the gradient flow (6) coincides with the steady state of the simplified flow

$$\partial_t u = \left(\nabla \cdot \frac{\nabla u}{|\nabla u|} - \mu \left((c_1 - f)^2 - (c_2 - f)^2 \right) \right).$$

This simplified flow represents the gradient descent for minimizing the energy

$$E(u) = |\nabla u|_1 + \mu \langle u, r \rangle \tag{7}$$

where $r = (c_1 - f)^2 - (c_2 - f)^2$.

Because the energy (7) is only homogeneous of degree 1, it has no unique global minimizer (this should be expected, as curves do not have unique level set representations). To make the global minima well defined, we must constrain the solution to lie e.g. in the interval [0, 1]. This results in the optimization problem

$$\min_{0 \le u \le 1} |\nabla u|_1 + \mu \langle u, r \rangle. \tag{8}$$

Once this optimization problem is solved, the segmented region is found by thresholding the level set function to get

$$\Omega = \{x : u(x) > \alpha\}, \tag{9}$$

for some $\alpha \in (0, 1)$.

In [20], the uniqueness of solutions to (8) is examined. There, it is found that, with probability 1, the indicator function, χ_Ω, obtained through this thresholding procedure is a global minimizer of the energy (7). Note that, if we take $u = \chi_\Omega$, then the energies (5) and (8) are equivalent, and so the region obtained through this procedure is a global minimizer of (5) as well. Furthermore, in the continuous case, the results of this thresholding procedure are independent of the value of α, up to a set of measure zero. For a more precise statement of these results, we refer the reader to [20]. More theoretical results on this class of scheme for global minimization are presented by Burger [12].

As we have described it thus far, the GCS segmentation procedure is merely a convexification of the ACWE approach. In [11] Bresson et al. modify the energy (7) to incorporate information from an edge detector, and get results that are very similar to those obtained through the classical snakes/GAC model. This is accomplished using the weighted *TV* norm

$$TV_g(u) = \int g|\nabla u| = |\nabla u|_g.$$

By replacing the standard *TV* norm with this weighted version, we make the model more likely to favor segmentation along curves where the edge detector function is minimal. In this sense, the GCS model is an elegant hybridization of the GAC/snakes model with the ACWE model.

In summary, the GCS approach to segmentation proceeds as follows

1: **while** "not converged" **do**
2:     Define $r^k = (c_1^k - f)^2 - (c_2^k - f)^2$
3:     Solve $u^k = \min_{0 \le u \le 1} |\nabla u|_g + \mu \langle u, r^k \rangle$
4:     Find $\Omega^k = \{x : u^k(x) > \alpha\}$
5:     Update $c_1^{k+1} = \int_{\Omega^k} f \, dx$, and $c_2^{k+1} = \int_{(\Omega^k)^c} f \, dx$
6: **end while**

The numerical bottleneck of this segmentation algorithm is computation of the minimizer (8), i.e. step 3 of the above algorithm. In [20], the authors propose to enforce the inequality constraint using an exact penalty function. They solve the unconstrained problem

$$\min_u |\nabla u|_g + \mu \langle u, r \rangle + \alpha \nu(u), \tag{10}$$

where $\nu(u) = \max\{0, 2|u - 0.5| - 1\}$. The authors of [20] show that, for sufficiently large α, this penalty function will exactly enforce the constraint. The main drawback of the formulation (10) is that the penalty function being used is non-differentiable. To handle this problem,

the authors of [20] propose to regularize the penalty function, rather than to exactly enforce the inequality constraint.

In [11] the authors use a splitting/regularization approach to minimize (8). They minimize

$$\min_{u, 0 \leq v \leq 1} |\nabla u|_g + \mu \langle u, r \rangle + \frac{1}{2\theta} \|u - v\|^2,$$

where the right-most term enforces $u \approx v$ for sufficiently small θ. Minimization with respect to u corresponds to solving the ROF model, which is done using a gradient projection method [5, 15, 19], and the minimization for v can be solved using an explicit formula. However, this scheme slows down as the accuracy increases (i.e. as $\theta \to 0$). Also, the approximate enforcement of the constraint $u \approx v$ has the effect of regularizing the model.

The disadvantage of these regularized schemes is that they "smear" the values of u near the boundaries of objects. This makes the results more dependent on the value of the cutoff parameter, α, and can eliminate fine segmentation details (see Fig. 5 for an example of this).

In the next section, we show how the GAC energy can be minimized using a Split Bregman approach. In addition to being able to minimize this energy without the use of regularization, the proposed method has the advantage of being a much more efficient solver for (8).

3 A Convex Formulation of Snakes/GAC

In this section, we show how the snakes/GAC model can be well approximated using the ROF functional. Conventional techniques for evolving the GAC contour rely on explicit discretizations of the gradient flow (4). Because of the vanishing denominator in (4), explicit methods require regularization, and suffer from stiff time step restrictions [35, 37]. An efficient alternative is to minimize this energy using a non-regularized implicit scheme, as introduced by Almgren, Taylor and Wang [2]. This scheme evolves the GAC contour by solving a non-convex variational problem at each time step. In [16], Chambolle introduced an implicit scheme that evolves the contour using a sequence of convex variational problems involving the ROF functional. The derivation presented here follows the approach presented in [10], which adapts Chambolle's mean curvature motion to the GAC energy.

Given some initial curve, C, let d_C be its representation as a signed distance function. Consider the convex minimization problem

$$u^* = \min_u |\nabla u|_g + \frac{1}{2h} \|u - d_C\|^2. \tag{11}$$

Now, consider the new curve $C' = \partial\{x : u^*(x) < 0\}$. In words, u^* is the level set representation of C'.

We will show that this process of obtaining C' from C is equivalent to applying a discretized version of (3) to the initial curve C. The Euler-Lagrange equation for (11) is

$$-g\nabla \cdot \frac{\nabla u}{|\nabla u|} - \left\langle \nabla g, \frac{\nabla u}{|\nabla u|} \right\rangle + \frac{1}{h}(u - d_C) = 0.$$

If we choose $x \in C'$, then $u^*(x) = 0$ and

$$d_C(x) = -h \left[g\nabla \cdot \frac{\nabla u}{|\nabla u|} + \left\langle \nabla g, \frac{\nabla u}{|\nabla u|} \right\rangle \right] \tag{12}$$

$$= -h \left[g\kappa + \langle \nabla g, N \rangle \right](x) \tag{13}$$

where we have used κ and N to denote the curvature and normal to C', respectively. We now denote by x_0 the projection of x onto the curve C. We now have

$$x = x_0 - h\left[g\kappa + \langle \nabla g, N \rangle\right] N(x_0),$$

which is the equivalent to one implicit time step in the evolution of (3). This result suggests Algorithm 1 for the evolution of the snakes/GAC model.

Algorithm 1 ROF based snakes/GAC

1: **while** $\|u^k - u^{k-1}\| > \epsilon$ **do**
2:     Define $u^{k+1} = \min_u |\nabla u|_g + \frac{1}{2h}\|u - d^k\|^2$
3:         $d^{k+1} = SDF(u^{k+1})$
4: **end while**

Here, we use $SDF(u)$ to denotes the signed distance function obtained by re-initializing u [1, 35, 43] . For a more detailed discussion of this approach, and some theoretical results, see [10, 16].

The advantages of this formulation are two fold: First, the ROF-based formulation of the snakes model allows one to use much larger time steps than the standard explicit discretizations, and does not suffer from the strict time step restrictions of regularized methods. Second, this method can be easily and efficiently implemented using a fast ROF solver. Below, we will discuss the implementation of this snakes model using a fast Split Bregman ROF solver.

4 The Split Bregman Method: A General l1 Minimization Technique

The Split Bregman method is a technique for solving general L1-regularized problems of the form

$$\arg\min_u |\Phi u|_1 + \frac{\mu}{2}\|Au - f\|^2, \tag{14}$$

where Φ and A are (possibly singular) linear operators. For example, choosing $\Phi = \nabla$ and $A = I$ yields the ROF model [40]. There is a large literature on techniques for solving (14). Many techniques approach the problem by either solving a regularized form of (14) directly, or by attacking the differentiable "dual formulation" of the problem, which requires the enforcement of linear inequality constraints.

In [26], the Split Bregman method was introduced for solving (14). This method has the advantage that it does not require regularization, continuation, or the enforcement of inequality constraints. Furthermore, the technique has been shown to be an extremely efficient solver for L1 regularized denoising problems, as well as a large class of problems from compressed sensing.

The Split Bregman method works by "de-coupling" the L1 and L2 terms in (14), using a splitting originally introduced in [50]. Rather than solve (14) directly, we introduce the auxiliary variable $\vec{d} \leftarrow \Phi u$. The problem (14) then becomes

$$\arg\min_u |\vec{d}|_1 + \mu\|Au - f\|^2 \quad \text{such that } \vec{d} = \Phi u. \tag{15}$$

To solve this constrained problem, we convert it to an unconstrained problem by introducing a quadratic penalty function:

$$\arg\min_{u,\vec{d}} |\vec{d}|_1 + \mu\|Au - f\|^2 + \frac{\lambda}{2}\|\vec{d} - \Phi u\|^2. \tag{16}$$

This formulation of the problem is very advantageous because the unconstrained problem (16) can be solved using a simple alternating minimization scheme. The first step of this alternating scheme is to minimize with respect to u. This is a differentiable optimization problem and the solution is obtained by solving

$$(\mu A^T A - \lambda \Delta)u^* = \mu A^T f + \lambda \Phi \vec{d}. \tag{17}$$

We next minimize (16) with respect to $\vec{d}$. This optimization problem is element-wise decoupled. In the work [45], it is shown that the solution can be written explicitly as

$$\vec{d}^* = shrink(\Phi u, 1/\lambda).$$

Note that the quadratic penalty function in (16) only approximately enforces the constraint $\vec{d} = \Phi u$. We wish to enforce this constraint exactly. A standard approach to this problem is to use a continuation scheme: solve (16) with an increasing sequence of penalty parameters, $\lambda_1 < \lambda_2 < \cdots < \lambda_n$. This continuation approach was introduced and applied to deconvolution problems in [50].

This continuation scheme suffers from two deficiencies. First, as $\lambda \to \infty$, so does the condition number of the system (17). This causes most iterative solvers for the system (17) to stall. Also, it has been shown in [50] that for large enough λ, the convergence rate of the alternating minimization scheme becomes arbitrarily slow.

To avoid these difficulties, the Split Bregman approach uses a fixed value for λ, and enforces the constraint $\vec{d} = \Phi u$ using a Bregman iteration technique [9, 26]. The details of Bregman iteration will not be discussed here. Rather we refer the reader to the works [27, 38, 49]. Alternately, Bregman iteration techniques can also be viewed as a special case of the method of multipliers, which is discussed in detail in [6]. An in-depth description of the application of this technique to the Split Bregman method can be found in [26].

To apply Bregman iteration to problem (16), we add a vector, $\vec{b}^k$, inside of the quadratic penalty function. We then solve a sequence of unconstrained problems defined by

$$(u^k, \vec{d}^k) = \arg\min_{u,\vec{d}} |\vec{d}|_1 + \mu\|Au - f\|^2 + \lambda\|\vec{d} - \Phi u - \vec{b}^k\|^2, \tag{18}$$

$$\vec{b}^{k+1} = \vec{b}^k + \Phi u^k - \vec{d}^k. \tag{19}$$

After the alternating minimization scheme approximately solves each unconstrained problem, the Bregman vector is updated using the rule (19). This rule is the analog of the "adding back the noise" technique, which has been used to enhance image denoising [38].

In [26, 38], it is shown that (under sufficient assumptions) this algorithm converges in the sense that, as $k \to \infty$, we have $\|\vec{d} - \Phi u\| \to 0$ and $\|u^k - u^*\| \to 0$ where u^* is some solution to (14). Furthermore, in [42] Setzer demonstrates the equivalence between split Bregman schemes and an alternating-direction-implicit scheme on the dual form of the problem (14), proving convergence in the case where optimizations sub-problems are solved exactly. Similar results are proved using the classical augmented Lagrangian framework in [23].

We have observed that, in general, an exact solution to the unconstrained problem (18) is not necessary. Rather, an approximate solution can be used. Usually, this approximation

is obtained using only one iteration of the alternating minimization procedure and/or an inexact solution to the system (17). For this reason, the iterates $(u^k, \vec{d}^k)$ can be obtained very fast for most applications. This observation is in agreement with the previous literature on multiplier methods for differentiable problems. The use of inexact solvers in the context of differentiable problems is discussed in detail in [6], where conditions are presented under which multiplier methods can be guaranteed to converge. A discussion of inexact solvers in the context of Bregman methods for L1 regularized problems can be found in [47].

5 Split Bregman Methods for ROF

One of the simplest applications of the Split Bregman method is for ROF denoising [26, 38]. We give a brief overview of this technique here, and refer the reader to [26] for more detail. In this image restoration model, the goal is to recover a denoised image, u, from a noisy image, f. The ROF model accomplishes this by solving an optimization problem of the form:

$$u^* = \arg\min_u \|u\|_{BV} + \frac{\mu}{2}\|u - f\|^2 = \arg\min_u |\nabla u|_1 + \frac{\mu}{2}\|u - f\|^2. \tag{20}$$

Note that this problem is of the form (14), when the operator A is taken to be the identity.

To apply the Split Bregman approach to this optimization problem, we make the substitutions $\vec{d} \leftarrow \nabla u$. Because this problem is defined on a two dimensional domain, $\vec{d} = (d_x, d_y)$ is vector valued at each pixel [26]. To approximately enforce these equality constraints, we add quadratic penalty functions. This results in the unconstrained problem

$$(u^*, \vec{d}^*) = \arg\min_{u, \vec{d}} |\vec{d}|_1 + \frac{\mu}{2}\|u - f\|^2 + \frac{\lambda}{2}\|\vec{d} - \nabla u\|^2. \tag{21}$$

We now wish to exactly enforce the equality constraints $\vec{d} = \nabla u$. For this purpose, we apply Bregman iteration to the unconstrained problem (21). This results in the following sequence of optimization problems:

1: **while** not converged **do**
2:    $(u^k, \vec{d}) = \arg\min_{u, \vec{d}} |\vec{d}|_1 + \frac{\mu}{2}\|u - f\|^2 + \frac{\lambda}{2}\|\vec{d} - \nabla u - \vec{b}^k\|^2$
3:    $\vec{b}^{k+1} = \vec{b}^k + \nabla u - \vec{d}^k$
4: **end while**

As described above, it is not necessary to solve the minimization problem in step 2 of the above scheme exactly. Rather, we generate an approximate solution using the alternating minimization scheme. For this purpose, we first minimize with respect to u. The optimal value of u can be obtained by solving the system

$$(\mu I - \lambda \Delta)u = \mu f + \lambda \nabla \cdot (\vec{b} - \vec{d}). \tag{22}$$

To approximately solve this system, we choose a fast, iterative solve. Because the system is strictly diagonally dominant, the most natural choice is a Gauss-Seidel solver. The next step in the alternating minimization scheme is to minimize with respect to $\vec{d}$. This can be done explicitly using the vector-valued shrinkage operator

$$\vec{d}^{k+1} = shrink(\nabla u + \vec{b}, \lambda). \tag{23}$$

When we put all of these pieces together, we get the following very simple, yet efficient algorithm.

Algorithm 2 Split Bregman ROF

1: **while** $\|u^{k+1} - u^k\| > \epsilon$ **do**
2:     $u^{k+1} = GS_{ROF}(f, \vec{d}^k, \vec{b}^k)$
3:     $\vec{d}^{k+1} = shrink(\nabla u^k + \vec{b}^k, \lambda)$
4:     $\vec{b}^{k+1} = \vec{b}^k + \nabla u^{k+1} - \vec{d}^{k+1}$
5: **end while**

We have used $GS_{ROF}(f, \vec{d}, \vec{b})$ to denote one iteration of the Gauss-Seidel method applied to the system (22). An explicit formula for the iteration, as well as more implementation details of this method, is given in [26]. In the results section of the paper, time trials are also presented, and the efficiency of this approach is discussed.

6 The Split Bregman Method Applied to Globally Convex Segmentation

In this section, we discuss the application of the Split Bregman method to the GCS problem, and prove some elementary convergence results. As described in Algorithm 2, the convexified segmentation can be reduced to a sequence of problems of the form

$$\min_{0 \leq u \leq 1} |\nabla u|_g + \mu \langle u, r \rangle, \tag{24}$$

where $r = (f - c_1)^2 - (f - c_2)^2$.

We wish to solve this problem efficiently without the use of regularization [11, 20]. To this end, we will apply the Split Bregman method. Just as was done for the ROF model, we introduce the auxiliary variable, $\vec{d} \leftarrow \nabla u$. To weakly enforce the resulting equality constraint, we add a quadratic penalty function. We get the following unconstrained problem

$$(u^*, \vec{d}^*) = \arg \min_{0 \leq u \leq 1, \vec{d}} |\vec{d}|_g + \mu \langle u, r \rangle + \frac{\lambda}{2} \|\vec{d} - \nabla u\|^2. \tag{25}$$

In order to strictly enforce the constraint $\vec{d} = \nabla u$, we apply Bregman iteration to the problem, just as was done for ROF. The resulting sequence of optimization problems is

$$(u^{k+1}, \vec{d}^{k+1}) = \arg \min_{0 \leq u \leq 1, \vec{d}} |\vec{d}|_g + \mu \langle u, r \rangle + \frac{\lambda}{2} \|\vec{d} - \nabla u - \vec{b}^k\|^2, \tag{26}$$

$$\vec{b}^{k+1} = \vec{b}^k + \nabla u^k - \vec{d}^k. \tag{27}$$

As described above, we will solve the problem (26) using the alternating minimization scheme. We begin by considering the minimization of (26) with respect to u. As was observed for the ROF problem, the algorithm converges very quickly even when an approximate solution is used. We will obtain an approximate solution using a Gauss-Seidel method. This observation is in agreement with the previous literature on multiplier methods for differentiable problems. The use of inexact solvers in the context of differentiable problems is discussed in detail in [6], where conditions are presented under which multiplier methods can be guaranteed to converge. A discussion of inexact solvers in the context of Bregman methods for L1 regularized problems can be found in [47].

To derive a closed-form expression for the element-wise minimizer, we begin with the Euler-Lagrange equation of (25). The optimal value of u must satisfy this Euler-Lagrange equation at every point where the constraint is not active:

$$\Delta u = \frac{\mu}{\lambda} r + \nabla \cdot (\vec{d} - \vec{b}), \quad \text{whenever } 0 < u < 1. \tag{28}$$

Next, consider the problem of minimizing (25) with respect to $u_{i,j}$ while all other elements of u remain constant. Ignoring the constraints, this energy is quadratic in $u_{i,j}$, and the minimum of this quadratic function is found by solving (28) for $u_{i,j}$. If the solution to this equation lies in the interval [0, 1] then this global minimizer coincides with the minimizer of the constrained problem. If the solution lies outside of this interval, then the energy is strictly monotonic inside [0, 1], (because the energy is quadratic) and the minimizer lies at the endpoint closest to the unconstrained minimizer. We have now arrived at the following element-wise minimization formula:

$$\alpha_{i,j} = d^x_{i-1,j} - d^x_{i,j} - b^x_{i-1,j} + b^x_{i,j} + d^y_{i,j-1} - d^y_{i,j} - b^y_{i,j-1} + b^y_{i,j}, \tag{29}$$

$$\beta_{i,j} = \frac{1}{4}(u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - \frac{\mu}{\lambda} r + \alpha_{i,j}), \tag{30}$$

$$u_{i,j} = \max\{\min\{\beta_{i,j}, 1\}, 0\}. \tag{31}$$

Minimization with respect to $\vec{d}$ is performed using the following formula:

$$\vec{d}^{k+1} = shrink_g(\vec{b}^k + \nabla u^{k+1}, \lambda).$$

When this Split Bregman minimization scheme is placed into Algorithm 2, we get the following scheme for segmentation.

Algorithm 3 Split Bregman for GCS

1: **while** $\|u^{k+1} - u^k\| > \epsilon$ **do**
2:     Define $r^k = (c^k_1 - f)^2 - (c^k_2 - f)^2$
3:     $u^{k+1} = GS_{\text{GCS}}(r^k, \vec{d}^k, \vec{b}^k)$
4:     $\vec{d}^{k+1} = shrink_g(\nabla u^{k+1} + \vec{b}^k, \lambda)$
5:     $\vec{b}^{k+1} = \vec{b}^k + \nabla u^{k+1} - \vec{d}^{k+1}$
6:     Find $\Omega^k = \{x : u^k(x) > \mu\}$
7:     Update $c^{k+1}_1 = \int_{\Omega^k} f \, dx$, and $c^{k+1}_2 = \int_{(\Omega^k)^c} f \, dx$
8: **end while**

Here, we have used $GS_{\text{GCS}}(r^k, \vec{d}^k, \vec{b}^k)$ to denote one sweep of the Gauss-Seidel formula (29).

7 Numerical Results

In the following discussion, we will explore the efficacy of the Split Bregman method for the applications discussed above by applying the techniques to several sample problems. All Split Bregman algorithms were implemented in C, and compiled/called through Matlab using a "mex" interface and the GCC compiler. All time trails were done on a 3 GHz, Intel

Core 2 Duo processor. All codes used to generate these results are available for download from the authors' websites.

7.1 Region-Based Segmentation

To demonstrate the efficiency of the Split-Bregman algorithm for GCS, the algorithm was applied to a variety of test problems. We begin by comparing the GCS algorithm to ACWE using a standard brain segmentation problem (Fig. 1). This comparison was done using a Matlab implementation of ACWE. The ACWE energy was minimized using a gradient descent method, with time steps adaptively chosen using a backtracking line search. For all the test problems presented here, the GCS model was minimized using the above Split Bregman formulation, with $\lambda = 0.5$.

Our first example demonstrates the advantages of a convex segmentation model, such as GCS. Because the ACWE algorithm relies on a non-convex energy (5), gradient descent methods for this problem converge to local minima, resulting in segmentations that depend strongly on initialization. Two different initialization states, and the resulting segmentations, are presented in (Fig. 1). The GCS method, in contrast, produces a segmentation with very
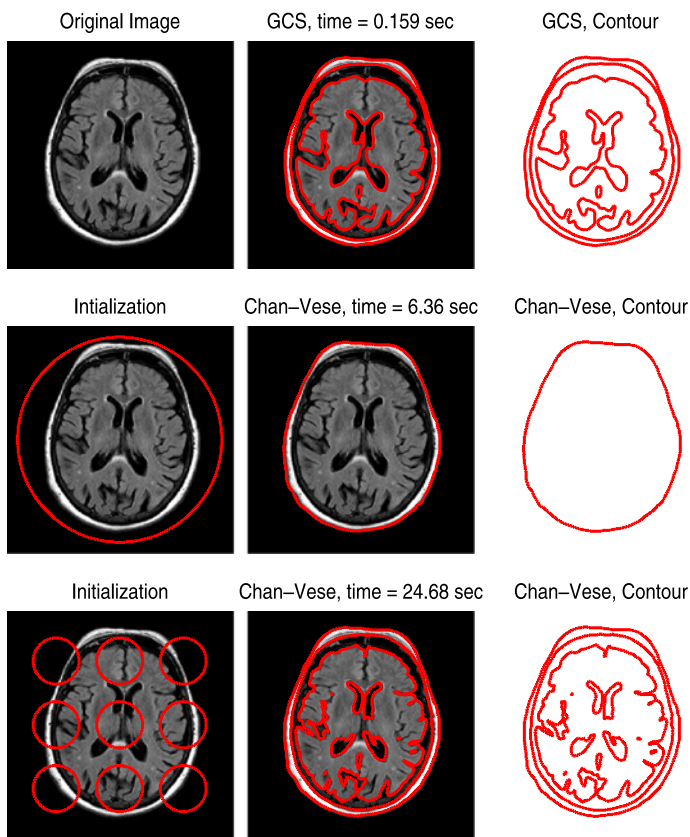

Fig. 1 Comparison of GCS to Chan-Vese. (*Top*) The results of GCS are independent of initialization. (*Center/Bottom*) Results of ACWE segmentation with 2 different initializations. Image size is 256×256
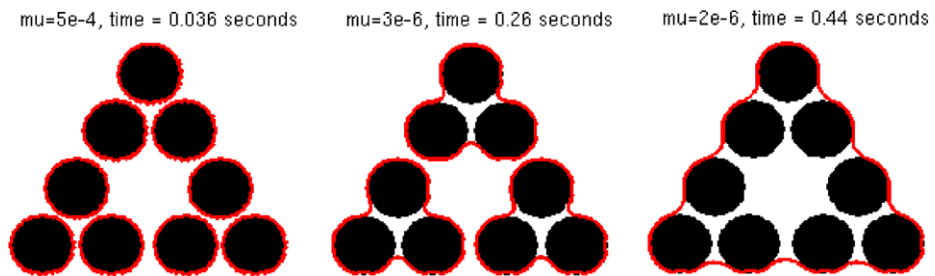
Fig. 2 Segmentation of a multi-scale test image using the GCS method. Note how the fidelity parameter, μ, allows the user to control the scale of the segmentation. Image size is 256×256

smooth contours, and does not depend on initialization. Note also the Split Bregman algorithm allows us to compute the GCS minimizer in only 0.16 seconds. The ACWE minimizer was much slower, taking over 20 seconds for some initializations. The image size is 256 by 256.

Our second test problem demonstrates how parameters allows the user to control the scale of the segmentation. Figure 2 shows segmentation results for a 256 by 256 image at 3 different levels of coarseness. At the finest level, each circle in the image is recognized as its own feature. As the GCS parameter decreases, the segmentation becomes coarser—first grouping image features together into clusters, and finally recognizing all circles as one feature. Finally, note how the Split Bregman method slows down considerably as the segmentation becomes more coarse. This is largely due the fact that the Gauss-Seidel iteration converges more slowly for small values of the fidelity parameter. However, even for the most coarse segmentation, computation took only 0.44 seconds, which is quite reasonable for such a difficult segmentation problem. Note that the GCS model inherits this level of control from the ACWE model [18] which also contains a scale parameter.

We now examine how the incorporation of an edge detector enhances segmentation detail using examples from medical imaging. We first consider a brain segmentation problem, Fig. 3. Unlike Fig. 1, this brain image has many fine scale, narrow details. Without the use of the edge detector, the regularization prevents the algorithm from recognizing these details. The use of the edge detector results in much more detailed segmentation while still maintaining smooth contours, and heavy regularization away from the edges.

The edge detector can also be used to avoid problems caused by non-constant regions, as demonstrated in Fig. 4. In this test problem, the standard GCS model breaks the main image feature into two sections because of the heavy intensity gradient in this image. The edge detector causes the segmentation to break along feature edges. Other approaches to confront this problem have been proposed using, Laplacian-based edge detectors [31], but this approach has the advantage of convexity.

Finally, we evaluate the Split Bregman scheme by direct comparison to another method: the gradient projection method originally proposed in [11]. This method works by splitting the functional (24) into two parts, one of which looks like the standard ROF energy, which can subsequently be minimized using a dual gradient projection method [5, 15, 19] . Note that this method does not solve the original minimization (24) exactly, but rather a regularized version of the problem. In Fig. 5, we display the level set function, u, computed by each algorithm as a minimizer of 24. A segmentation is produced by thresholding this LSF, as in equation (9). In addition to the LSF's, the histogram of each segmentation is displayed in Fig. 5. Note that, because of the regularization, the results computed by the gradient projection show many small scale features that were not entirely removed by the segmentation.
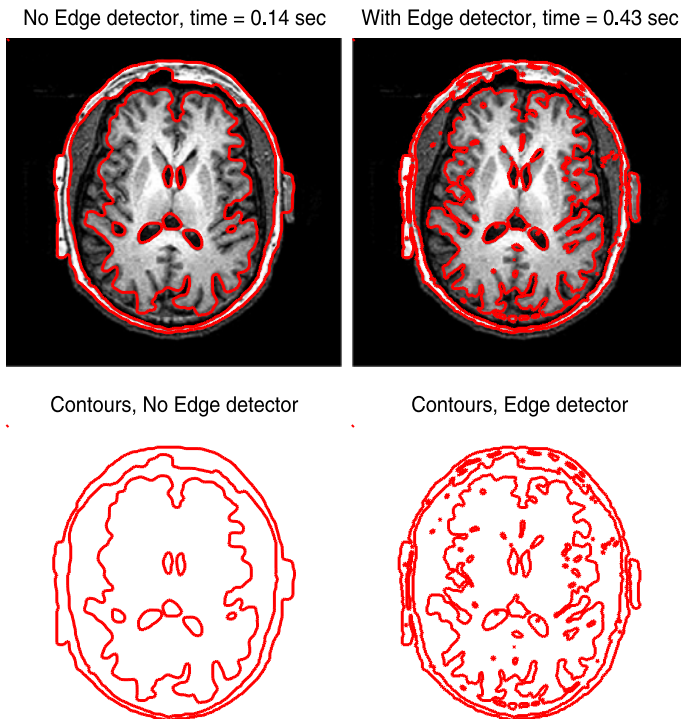
No Edge detector, time = 0.14 sec   With Edge detector, time = 0.43 sec

Contours, No Edge detector   Contours, Edge detector

Fig. 3 The results of GCS are enhanced by using an edge detector function. (*Left*) Segmentation with no edge detector. (*Right*) Results using an edge detector, and a weighted TV regularizer. The edge detector allows the segmentation to recognize more fine-scale feature without changing the regularization parameter. Image size is 218×218

Also, as can be seen in the histograms, a much higher fraction of the pixels in the SDF are either 0, or 1 when the results are computed using the Split Bregman method. Because of this, the Split Bregman results are much less dependent of the choice of the threshold, α.

7.2 GAC/Snakes

We will examine several application of the GAC model. We first consider a standard segmentation problem using an edge detector (Fig. 6). The edge detector function was gradient based, and of the form (2), with $\beta = 1$. The time step parameter μ was chosen so that the time required to compute each weighted ROF minimization and SDF re-initialization was approximately equal. Because the implicit method allows such long time steps, we were able to compute this segmentation using only 15 time steps.

Another application of the GAC model (1) is the reconstruction of a surface from unorganized data points [51]. Given a set of points $\{\mathbf{x}_i\}$ lying on some surface, we wish to reconstruct an approximation of that surface. This can be accomplished using time steps of the form (11) where the weight function g is defined to be the distance function of the set $\{\mathbf{x}_i\}$. As a result, this modified GAC model computes a surface that lies along the minimal contours of this distance function.

We begin our examination of this algorithm using 2D model problems. While these model problems are unrealistically simple, they demonstrate the effect of surface geometry on convergence speed. We apply the algorithm to two data sets in Fig. 7. Note that for
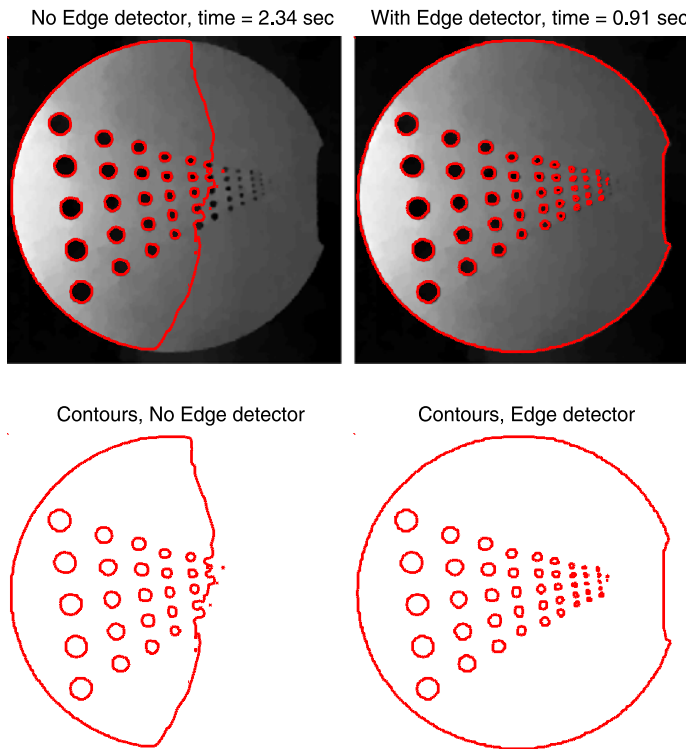
No Edge detector, time = 2.34 sec   With Edge detector, time = 0.91 sec

Contours, No Edge detector   Contours, Edge detector

Fig. 4 Segmentation applied to a non-piecewise constant image. (*Left*) Segmentation of an MR image without edge detector. (*Right*) Edge detector enhanced segmentation. Image size is 371 × 392

the convex shape in Fig. 7, the surface reconstruction took only 3 time steps, whereas 9 time steps were required to segment the "star" shape. The reason for this is that the curve evolution tends to slow down when it reaches non-convex portions of the surface because evolution into these regions increases the perimeter of the curve, and thus this gradient of the energy (1) is not as steep here. Note that a similar effect occurs in Fig. 6, where the evolution slows down as the curve approaches the wings of the plane.

We now consider the more realistic case of 3d data. We apply the surface reconstruction method to three well known 3d data sets: the "Buddha," "cow" and "bunny." The unorganized data points are shown in Fig. 8. To build an implicit representation for these surfaces, an initial guess was first chosen using the initialization method of [51]. The surface was evolved using a 3d implementation of Algorithm 2 for weighted ROF. Re-initialization was performed using a fast-marching method [1, 35, 43].

Running times and number of iterations required for convergence are shown in Table 1. All times are reported in seconds. The reconstructed surfaces are shown in Fig. 9. Note that, much like in the 2d case, more iterations are required to resolve shapes with deeper convexities, such as the "Buddha" data set. However, even for this difficult test problem, only 17 iterations were required for convergence. Even thought the ROF energy was minimized over a 3-dimensional space, the computing times for these data sets are quite manageable (under 4 minutes for "Buddha" and under 30 seconds for the "bunny"). It should be noted
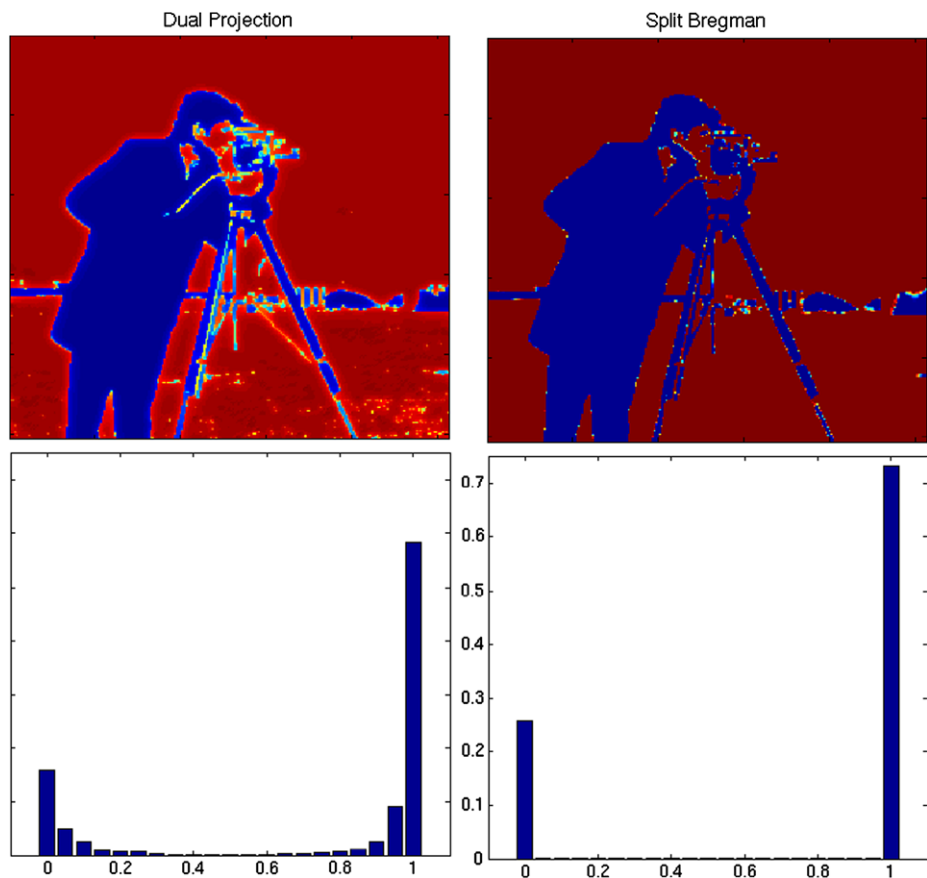
Fig. 5 Level set representations of the GCS minimizer. (*Left*) Minimization using the regularized dual projection algorithm [11]. (*Right*) The non-regularized solution obtained via Split Bregman. Note that the use of regularization results in artifacts around small features that make results more dependent on the threshold parameter

that these results could be further accelerated by limiting the ROF computation to a narrow band around the surface.

As a final example, we compare the Split Bregman scheme to the well known graph-cut algorithm for motion by mean curvature [17, 21]. In this approach, motion by mean curvature is accomplished using time steps of the form (11), where the weighted ROF energy is minimized using graph cuts, as computed using a preflow-push based Algorithm [25, 48]. Note that graph cuts cannot compute the isotropic energy minimizer [8, 32], and thus this technique computes a motion by "anisotropic curvature" [17]. Results are depicted in Fig. 10, where contours represent the evolution after 0, 3, 6, and 9 time steps. Note that the Split Bregman is slightly more efficient than the graph-cuts based solver. Also, the Split Bregman method gives us the option of computing the isotropic curvature motion; something that the anisotropic graph cuts solver does not.
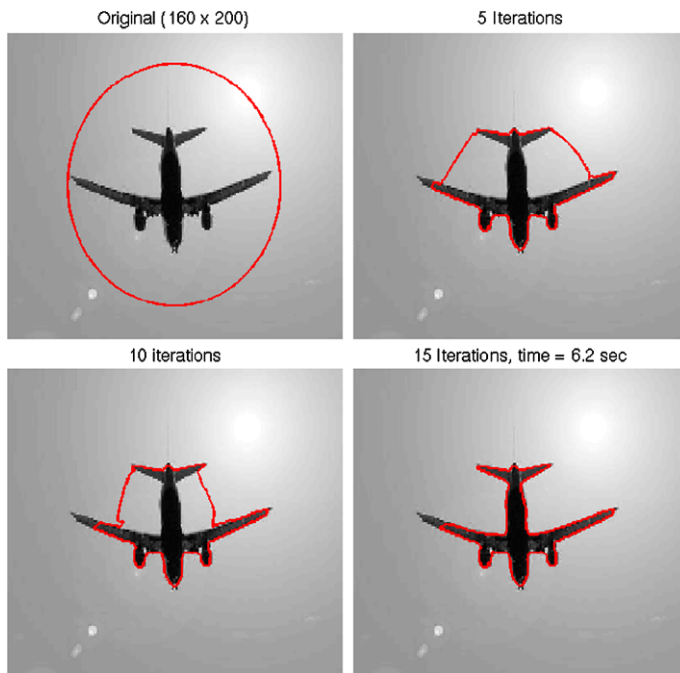
Fig. 6 Segmentation of a test image using GAC/snakes. Results are shown at 4 different stages of the segmentation
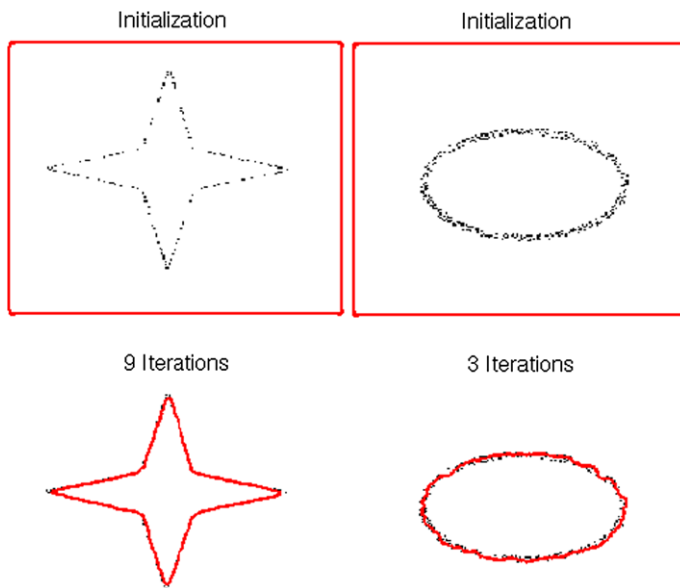

Fig. 7 GAC/snakes segmentation of two different test images. Note that more time steps are needed to segment the non-convex regions on the "star" shape
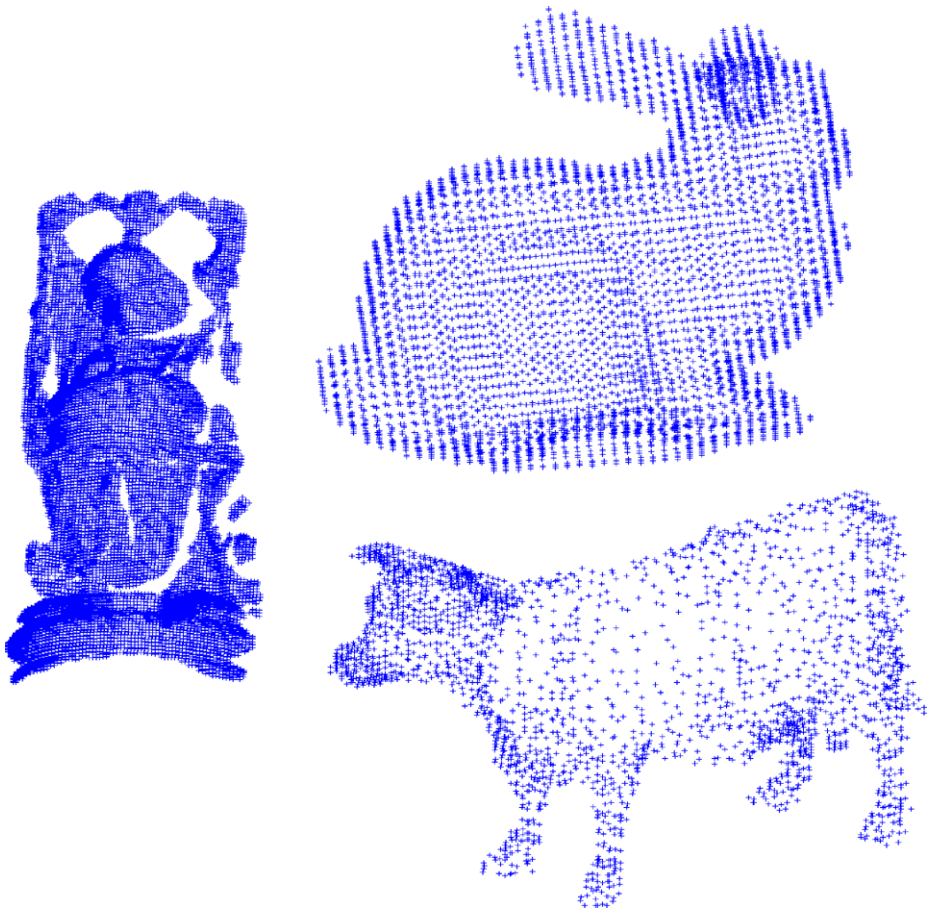
Fig. 8 Three sets of unorganized points. These data sets are the inputs for the surface reconstruction algorithm in 3 dimensions

Table 1 Results for 3d surface reconstruction

Data set	Buddha	Bunny	Cow
Num. iterations	17	13	15
Num. data points	16285	13995	2872
Dimensions	$93 \times 205 \times 84$	$76 \times 76 \times 66$	$53 \times 119 \times 81$
Time for ROF	11.6	2	2.9
Time for redistance	2.2	0.2	0.4

8 Conclusions

Variational Segmentation models have an important place in computer vision, and have been applied extensively in medical image, tracking, and robotics applications. Because the GCS and implicit GAC models only require the solution of convex problems, segmentations can
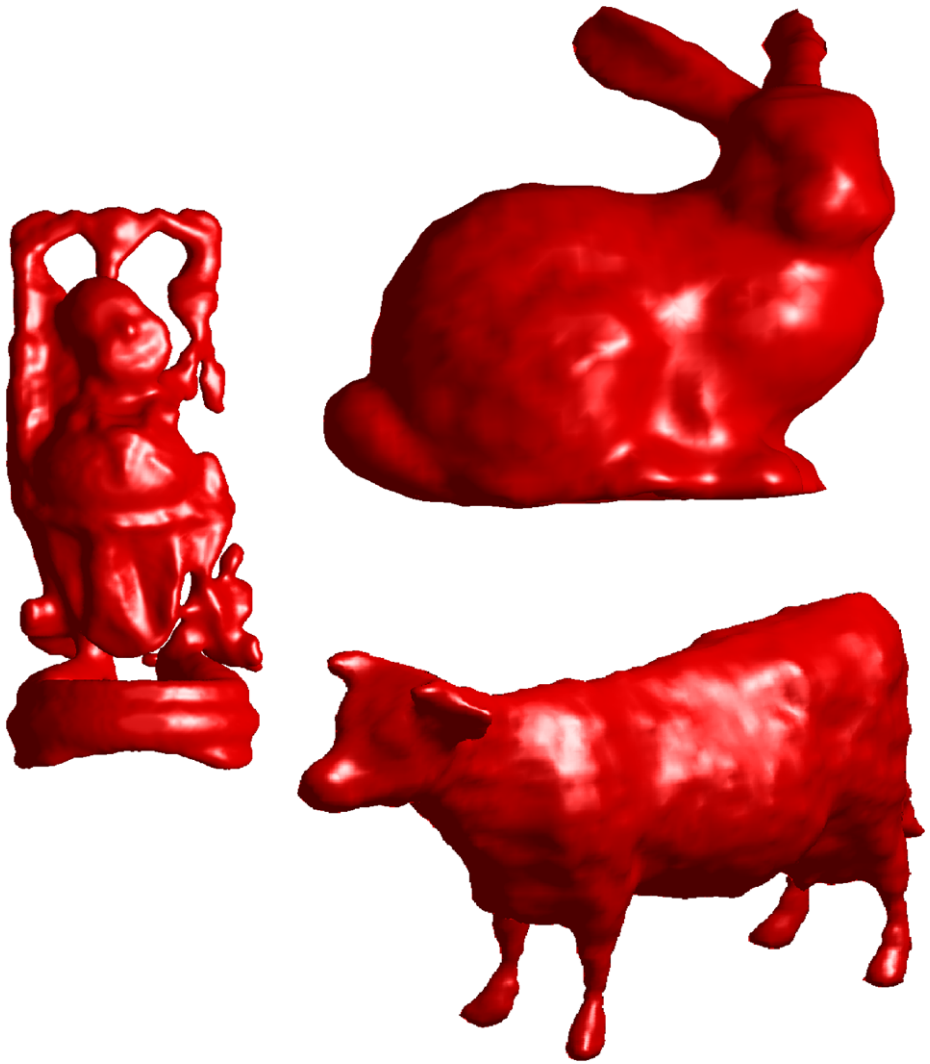
Fig. 9 Implicit surfaces generated by the 3d reconstruction algorithm

be computed reliably with minimal user input. In the case of the GCS method, results are completely independent of initialization, making this technique very attractive for applications which require full automation.

However, for many applications, particularly when images must be processed in "real time," the most important consideration when choosing a segmentation algorithm is speed. Because most PDE-based segmentation models require the solution of a non-linear, non-differentiable elliptic problem, this class of methods can be very slow to compute if conventional numerical schemes (especially those involving regularization) are used. A very important result of the convexity of the GCS and implicit GAC models is that it opens the door for fast minimization methods.
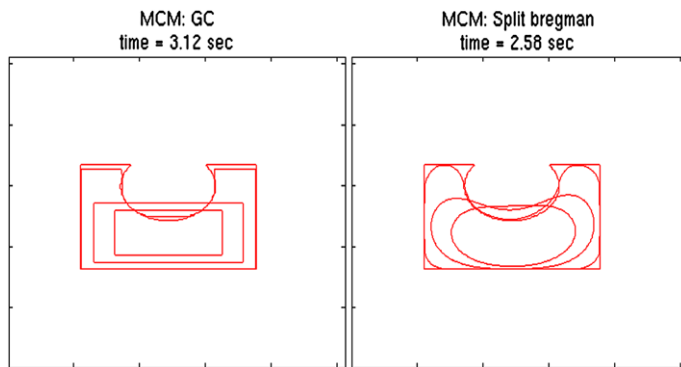
Fig. 10 Evolution of shapes by Mean-Curvature motion (MCM). Contours represent the results after 0, 3, 6, and 9 time steps. (*Left*) MCM computed via graph cuts. Graph-cuts techniques evolve curves by "anisotropic curvature," resulting in sharp angles and cusps. (*Right*) MCM computed via Split Bregman. This curve evolution is isotropic

In this paper, we first gave a brief introduction to image segmentation using convex energies. We then gave an introduction to the category of Split Bregman methods for minimizing L1 regularized energies. Because the segmentation models considered here rely on TV regularized energies, they are easily minimized using a Split Bregman approach. Finally, we presented numerical examples demonstrating the quality of segmentations produced using convex models, as well as the efficiency of these methods when computed using a Split Bregman scheme.

For the mean curvature motion problem considered here, the Split Bregman approach was found to be more efficient than other proposed methods, including techniques based on graph-cuts. In addition to its speed, the Split Bregman approach has the advantage that it can solve isotropic segmentation problems (rather than the anisotropic approximations employed by graph-cut strategies), and does not require regularization.

Future research will focus on accelerating the Split Bregman scheme in the case of small fidelity parameters, allowing for faster coarse segmentation of large images, and faster evolution of the GAC contour. Also, adaptive time stepping for GAC will allow for faster segmentation of non-convex regions.

References

1. Adalsteinsson, D., Sethian, J.: A fast level set method for propagating interfaces. J. Comput. Phys. **118**, 269–277 (1995)
2. Almgren, F., Taylor, J.E., Wang, L.: Curvature-driven flows: a variational approach. SIAM J. Control Optim. **31**(2), 387–438 (1993)

3. Amenta, N., Bern, M.: Surface reconstruction by Voronoi filtering. In: SCG'98: Proceedings of the Fourteenth Annual Symposium on Computational Geometry, pp. 39–48. ACM, New York (1998)
4. Amenta, N., Bern, M., Kamvysselis, M.: A new Voronoi-based surface reconstruction algorithm. In: SIGGRAPH'98: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, pp. 415–421. ACM, New York (1998)
5. Aujol, J.-F., Chambolle, A.: Dual norms and image decomposition models. Int. J. Comput. Vision **63**(1), 85–104 (2005)
6. Bertsekas, D.: Constrained Optimization and Lagrange Multiplier Methods. Academic Press, San Diego (1996)
7. Boissonnat, J.-D.: Geometric structures for three-dimensional shape representation. ACM Trans. Graph. **3**(4), 266–286 (1984)
8. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. **23**, 1222–1239 (2001)
9. Bregman, L.: 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. Math. Phys. **7**, 200–217 (1967)
10. Bresson, X., Chan, T.: Active contours based on chambolle's mean curvature motion. In: IEEE International Conference on Image Processing, pp. 33–36 (2007)
11. Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J.-P., Osher, S.: Fast global minimization of the active contour/snake model. J. Math. Imaging Vis. **28**, 151–167 (2007)
12. Burger, M., Hintermuller, M.: Projected gradient flows for bv/level set relaxation. UCLA CAM technical report, 05-40 (2005)
13. Carson, C., Belongie, S., Greenspan, H., Malik, J.: Blobworld: Image segmentation using expectation-maximization and its application to image querying. IEEE Trans. Pattern Anal. Mach. Intell. **24**, 1026–1038 (1999)
14. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: IEEE International Conference on Computer Vision, p. 694 (1995)
15. Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Imaging Vis. **20**(1–2), 89–97 (2004)
16. Chambolle, A.: An algorithm for mean curvature motion. Interfaces Free Bound. **6**(2), 195–218 (2004)
17. Chambolle, A., Darbon, J.: On total variation minimization and surface evolution using parametric maximum flows. UCLA CAM report 08-19 (2008)
18. Chan, T.F., Vese, L.: Active contours without edges. IEEE Trans. Image Process. **10**, 266–277 (2001)
19. Chan, T.F., Golub, G.H., Mulet, P.: A nonlinear primal-dual method for total variation-based image restoration. SIAM J. Sci. Comput. **20**, 1964–1977 (1999)
20. Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. SIAM J. Appl. Math. **66**, 1932–1648 (2006)
21. Darbon, J., Sigelle, M.: A fast and exact algorithm for total variation minimization. IbPRIA 2005 **3522**(1), 351–359 (2005)
22. Donoho, D.L., Johnstone, I.M.: Adapting to unknown smoothness via wavelet shrinkage J. Am. Stat. Assoc. **90**(432), 1200–1224 (1995)
23. Esser, E.: Applications of Lagrangian-based alternating direction methods and connections to split Bregman. UCLA CAM technical report, 09-31 (2009)
24. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. Int. J. Comput. Vis. **59**(2), 167–181 (2004)
25. Goldfarb, D., Yin, W.: Parametric maximum flow algorithms for fast total variation minimization. CAAM technical report, TR07-09 (2008)
26. Goldstein, T., Osher, S.: The split Bregman method for l1 regularized problems. UCLA CAM report 08-29 (2008)
27. He, L., Chang, T.-C., Osher, S.: Mr image reconstruction from sparse radial samples by using iterative refinement procedures. In: Proceedings of the 13th Annual Meeting of ISMRM, p. 696 (2006)
28. Hoppe, H., Derose, T., Duchamp, T., Mcdonald, J., Stuetzle, W.: Surface reconstruction from unorganized points. Comput. Graph. **26**(2), 71–78 (1992)
29. Jonasson, L., Bresson, X., Hagmann, P., Cuisenaire, O., Meuli, R., Thiran, J.-P.: White matter fiber tract segmentation in dt-mri using geometric flows. Med. Image Anal. **9**(9), 223–236 (2005)
30. Kass, W., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. J. Comput. Vis. **1**(4), 312–331 (2004)
31. Kimmel, R., Bruckstein, A.M.: Regularized Laplacian zero crossings as optimal edge integrators. Int. J. Comput. Vis. **53**, 225–243 (2001)
32. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell., pp. 147–159 (2004)

33. Malladi, R., Kimmel, R., Adalsteinsson, D., Sapiro, G., Caselles, V., Sethian, J.A.: A geometric approach to segmentation and analysis of 3d medical images. In: MMBIA'96: Proceedings of the 1996 Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA'96), Washington, DC, USA, p. 244. IEEE Comput. Soc., Los Alamitos (1996)
34. Mumford, D., Shah, J.: Optimal approximation by piecewise smooth functions and associated variational problems. Commun. Pure Appl. Math. **42**, 577–685 (1989)
35. Osher, S., Fedkiw, R.: Level Set Methods and Dynamic Implicit Surfaces. Springer, Berlin (2003)
36. Osher, S., Fedkiw, R.P.: Level set methods. Technical report, in Imaging, Vision and Graphics (2003)
37. Osher, S., Sethian, J.A.: Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. J. Comput. Phys. **79**, 12–49 (1988)
38. Osher, S., Burger, M., Goldfarb, D., Xu, J., Yin, W.: An iterative regularization method for total variation-based image restoration. MMS **4**, 460–489 (2005)
39. Rogers, D.F.: An Introduction to NURBS: With Historical Perspective. Morgan Kaufmann, San Mateo (2001)
40. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60**, 259–268 (1992)
41. Sethian, J.A.: Level set methods and fast marching methods: Evolving. In: Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science. Cambridge University Press, Cambridge (1999)
42. Setzer, S.: Split Bregman algorithm, Douglas-Rachford splitting and frame shrinkage. In: Proceedings of the Second International Conference on Scale Space Methods and Variational Methods in Computer Vision (2009)
43. Sussman, M., Smereka, P., Osher, S.: A level set approach for computing solutions to incompressible two-phase flow. J. Comput. Phys. **114**(1), 146–159 (1994)
44. Tschirren, J., Hoffman, E.A., McLennan, G., Sonka, M.: Intrathoracic airway trees: segmentation and airway morphology analysis from low-dose ct scans. IEEE Trans. Med. Imag. **24**, 1529–1539 (2005)
45. Wang, Y., Yin, W., Zhang, Y.: A fast algorithm for image deblurring with total variation regularization. CAAM technical reports (2007)
46. Yezzi, A., Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A.: A geometric snake model for segmentation of medical imagery. IEEE Trans. Med. Imag. **16**(2), 199–209 (1997)
47. Yin, W.: Analysis and generalizations of the linearized Bregman method. UCLA CAM technical report, 09-42 (2009)
48. Yin, W.: Pgc: A preflow-push based graph-cut solver. Version 2.32
49. Yin, W., Osher, S., Goldfarb, D., Darbon, J.: Bregman iterative algorithms for l1-minimization with applications to compressed sensing. SIAM J. Imag. Sci. **1**, 142–168 (2008)
50. Wang, Y., Yang, J., Yin, W., Zhang, Y.: A new alternating minimization algorithm for total variation image reconstruction. SIAM J. Imag. Sci. **1**(3), 248–272 (2008)
51. Zhao, H.-K., Osher, S., Merriman, B., Kang, M.: Implicit and nonparametric shape reconstruction from unorganized data using a variational level set method. Comput. Vis. Image Underst. **80**(3), 295–314 (2000)
52. Zhao, H.-K., Osher, S., Fedkiw, R.: Fast surface reconstruction using the level set method. In: VLSM'01: Proceedings of the IEEE Workshop on Variational and Level Set Methods (VLSM'01), Washington, DC, USA, p. 194. IEEE Comput. Soc., Los Alamitos (2001)