

# A Fast Variational Method for Surface Reconstruction from Sets of Scattered Points

Jian Ye, Xavier Bresson, Tom Goldstein, Stanley Osher\*

January 4, 2010

## Abstract

This paper introduces an efficient method based on total variation-based functionals to solve the reconstruction problem of closed surfaces from a set of unorganized points in  $\mathbb{R}^2$  and  $\mathbb{R}^3$ . Recent developments in optimization have provided fast, accurate, and robust algorithms to solve total variation-based problems in imaging. We propose to use the efficient split-Bregman method to solve the problem of surface reconstruction without tuning any parameter and without using any additional information such as surface normals. Our surface reconstruction algorithm can handle dense and sparse collections of data points, while being robust to the presence of outliers. Our experiments show that our algorithm is fast and can recover surface details while using no more and often less information than related methods.

## 1 Introduction

Surface reconstruction is a highly challenging problem because sets of scattered points lack ordering information, connectivity, and may be noise contaminated. There exist two kinds of surface representations; explicit representations and implicit representations. Explicit surface representations track the exact locations of points on a surface, while implicit surface representations embed surfaces as a level set of a scalar-valued function. More

---

\*Authors are with the Department of Mathematics, University of California, Los Angeles, USA. Email: {ye,xbresson,tgoldst,sjo}@math.ucla.edu. J. Ye is supported by NSF DMS 0914561, X. Bresson by ONR N00014-03-1-0071, T Goldstein by ONR N00014-07-1-0810 and S. Osher by an ARO MURI subcontract from the University of South Carolina.

precisely, explicit representations include parametric surfaces s.a. [3, 4] and triangulated surfaces such as [6, 7, 8, 9, 10]. Implicit representations include e.g. [13, 14, 15, 16, 17, 18]. In this paper, we focus on implicit representations because they easily handle arbitrary and dynamically changing topology. Also, recent fast convex optimization techniques for implicit representations can be used.

One of the earliest successful implicit surface reconstruction methods was introduced by Zhao *et al.* in [15, 16]. Given a set of unorganized points,  $\{x_i\}$ , the reconstructed surface is the minimizing solution of the weighted length/area energy:

$$E(\Gamma) = \int_{\Gamma} d(x) d\Gamma, \quad (1)$$

where  $\Gamma$  is a curve in  $\mathbb{R}^2$  or a surface in  $\mathbb{R}^3$ ,  $d\Gamma$  is the arc length element or the surface parametrization, and  $d \in \mathbb{R}_+$  is the distance function of the set of points. This distance function satisfies the eikonal equation

$$\begin{aligned} |\nabla d(x)| &= 1 & x \in \Omega \setminus \{x_i\} \\ d(x_i) &= 0, \end{aligned} \quad (2)$$

and can be efficiently computed using the fast sweeping method of [19, 20].

The gradient flow of energy (1) is:

$$\partial_t C = (d\kappa - \langle \nabla d, \mathcal{N} \rangle) \mathcal{N} \quad (3)$$

where  $t$  is an artificial time parameter. The variables  $\kappa$  and  $\mathcal{N}$  are respectively the curvature and the normal to the surface  $\Gamma$ . If the surface  $\Gamma$  is represented by the zero level set of a function  $\phi$ , then the gradient flow becomes:

$$\partial_t \Phi = |\nabla \phi| \nabla \cdot \left[ d \frac{\nabla \phi}{|\nabla \phi|} \right] = |\nabla \phi| \left[ \langle \nabla d, \frac{\nabla \phi}{|\nabla \phi|} \rangle + d \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \right], \quad (4)$$

where  $\nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}$  is the curvature of level sets of  $\phi$ . The above PDE-based model provides accurate reconstructed surfaces (see [15] and [16] for more details). However, solving the PDE (4) is time consuming because the gradient flow requires very small time steps according to the CFL condition[1] to ensure a stable evolution of the surface. Furthermore, the energy functional (1) is non-convex, making results sensitive to the initial condition. In order to avoid non-meaningful local minimizers, the initial surface must lie close to

the true surface.

There exist alternative methods for evolving the surface  $\Gamma$  with the flow (3). Almgren et al. introduced in [21] an implicit scheme to evolve the contour. In [22], Chambolle modified Almgren's method to evolve the contour under the mean curvature motion using a sequence of convex optimization problems involving the Rudin-Osher-Fatemi (ROF) model. Based on [22], Bresson and Chan showed in [23] that the following iterative weighted ROF model can implicitly solve the gradient flow (3):

$$u^{k+1} = \arg \min_u \int_{\Omega} d|\nabla u| + \frac{1}{2h}(u - d_{\Gamma^k})^2, \quad (5)$$

where  $\Omega$  is the image domain,  $d$  is the distance function of the set of points,  $d_{\Gamma^k}$  is the signed distance function for the evolving surface  $\Gamma$  at iteration  $k$ , and  $h$  is the time step. In [18], the authors apply the split-Bregman method to iteratively minimize (5). Since split-Bregman efficiently solves the optimization problem (5), the iterative method for the surface reconstruction problem in [18] is faster than Zhao et.al.'s minimal surface model. However, the method in [18] is iterative, which means that several weighted ROF problems need to be solved. Also, the signed distance function  $d_{\Gamma^{k+1}}$  must be recomputed at each iteration.

In [17], Lempitsky and Boykov introduced a discrete optimization method to reconstruct surfaces implicitly. The proposed algorithm uses surface normals as the input and maximizes the number of weakly oriented data points contained in the surface while minimizing the surface area. This objective corresponds to minimize the energy function defined as:

$$\min_{\Gamma} \lambda \int_{\Gamma} d\Gamma - \int_{\Gamma} \langle \vec{n}_p, \mathcal{N} \rangle d\Gamma \quad (6)$$

where  $\lambda > 0$  controls the regularity of the surface,  $\vec{n}_p = \sum_i \rho_i(p) \vec{n}_i, \forall p \in \mathbb{R}^3$  and  $\vec{n}_i$  are input normals, and  $\rho_i(p)$  is a Gaussian whose variance depends on the density and the noise level of the data. Lempitsky and Boykov applied the parametric max flow/min cut (a.k.a. graph-cuts) to solve the optimization problem (6) with an implicit surface representation.

In this paper, we introduce an algorithm for surface reconstruction that only requires a set of unorganized points as input (e.g. surface normals are not required as inputs). Our algorithm consists of three steps. First, an

estimation of the surface will be determined using the eikonal equation and a convex formulation of the weighted Chan-Vese model [25, 26]. Second, surface normals will be approximated by denoising the surface estimation with the weighted ROF model. Finally, the Bregman iterative method [29] will be used to accurately reconstruct the surface, while removing noise and outliers. Our approach offers several advantages. The proposed algorithm is not only fast and accurate, it is also easy to code in 2D and 3D. In addition, numerical schemes are isotropic, sub-pixel accurate and the memory requirement in 3D is low. Our approach does not need extra information such as surface normals. Finally, the algorithm is robust to noise and outliers.

## 2 Proposed Method

In this section, we will describe the three steps of the surface reconstruction algorithm. The first step is to estimate the reconstructed surface based on the eikonal equation and the Chan-Vese model. The second step is to estimate the surface normals based on the weighted ROF model. The last step is to do the surface reconstruction using Bregman iteration. The computational cost for each step is low, so the overall algorithm is fast.

### 2.1 Step 1: Estimation of the Reconstructed Surface Based on the Eikonal Equation and the Chan-Vese Model

The objective is to compute an estimation of the surface using the set of scattered points. To estimate the surface, we will apply a convex formulation of the segmentation model of Chan and Vese [12] to a two-value image whose edges are located along the set of points. This two-value image is determined as follows. We observe that the gradient  $\nabla d$  of the distance function of the data points is equivalent to a vector flow pointing toward the data points. Hence, the function  $d$  can be considered as an edge detector function. A typical choice of an edge detector for an image  $f$  is:

$$g(x) = \frac{1}{\epsilon + |\nabla f(x)|^p}, \quad p > 0. \quad (7)$$

Because  $d$  is an edge detector we can write  $g = d + \epsilon$  and we can approximate the corresponding image  $f$  to this edge detector by solving the eikonal equation:

$$|\nabla f| = \frac{1}{g^{1/p} + \epsilon} \quad (8)$$

The boundary condition applied here is  $f = 0$  on the far field boundary. Figures 1(a-b) show the solution  $f$  of (8) for  $p = 1$  and  $p = \frac{1}{2}$ . As expected, the images look like a binary-valued function whose edges are located along the white spikes which are the data points. In our experiments, we pick  $p = \frac{1}{2}$  which produces approximately binary images for convex and non-convex shapes. We also take  $\epsilon = 1$  for all experiments.

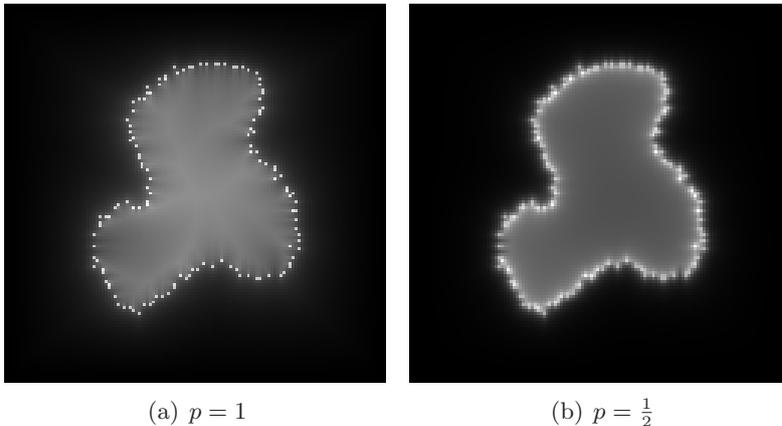


Figure 1: Solutions  $f$  of the eikonal equation (8).

Given the binary image  $f$  computed by (8), we would like to compute an implicit representation of the surface. This problem is equivalent to a binary image segmentation problem. It can be solved with the Chan and Vese model [12], which is the two-phase piecewise approximation of the Mumford and Shah energy[2]. The Chan-Vese model is defined as a variational model:

$$\min_{D, c_1, c_2} Per(D) + \mu \int_D (c_1 - f)^2 + \mu \int_{D^c} (c_2 - f)^2, \quad (9)$$

where  $D$  is a subset of the image domain  $\Omega$ ,  $D^c = \Omega \setminus D$ ,  $Per$  is the perimeter,  $\mu > 0$ , and  $c_1, c_2 \in \mathbb{R}$  are respectively the mean intensity values inside and outside the region  $D$ . Chan and Vese used the level set method to find a minimizer of (9). However, the level set method is sensitive to the initial contour position of  $D$  because the corresponding energy is non-convex. Besides, the numerical implementation of the level set method is relatively slow. To overcome the shortcomings of the non-convex Chan-Vese model, Chan *et al.* introduced in [25] a convex formulation of their model. The

authors propose to minimize the convex energy

$$\min_{0 \leq u \leq 1} \int_{\Omega} |\nabla u| + \mu \langle u, r \rangle, \quad (10)$$

where  $\int_{\Omega} |\nabla u|$  is the total variation of  $u$ ,  $\langle v_1, v_2 \rangle = \int_{\Omega} v_1 v_2$ , and  $r = (c_1 - f)^2 - (c_2 - f)^2$ . Chan *et al.* showed that for any minimizer of (10), any thresholding of the minimizer produces a minimizer of the original problem (9). In [26], Bresson *et al.* extended the convex formulation of the Chan-Vese model to get an elegant hybrid model which combines both the Chan-Vese model and GAC/snakes model:

$$\min_{0 \leq u \leq 1} \int_{\Omega} g |\nabla u| + \mu \langle u, r \rangle \quad (11)$$

where  $r = (c_1 - f)^2 - (c_2 - f)^2$ , and  $g$  is an edge detector. Bresson *et al.* also improved the speed of optimization of [25] by a factor 5 – 10 using the projection algorithm of Chambolle [22].

As we said above, the surface reconstruction problem is analogous to an image segmentation problem. Indeed, the binary image  $f$  computed with the eikonal equation will be segmented using the model (11) to give an implicit representation of the surface. We call  $u_1$  the solution of (11) when the function  $f$  is the solution of the eikonal equation and the edge detector is chosen to be  $g = d + \epsilon$ , where  $\epsilon > 0$  ensures a better stability of the numerical scheme. The model (11) is minimized using the efficient split-Bregman algorithm of Goldstein and Osher [27]. In [18], it is shown that the split-Bregman algorithm can solve the hybrid model extremely quickly. For example, a C implementation of the algorithm segments the classic test image ‘‘Cameraman’’ with  $256 \times 256$  grid points in less than 0.1 seconds on a standard laptop. The split-Bregman algorithm is thus applied to find a good estimate of the surface to reconstruct. We summarize the split-Bregman algorithm below (see [18] for more details):

- 1: **while**  $\|u^{k+1} - u^k\| > \epsilon$  **do**
- 2: Define  $r^k = (c_1^k - f)^2 - (c_2^k - f)^2$
- 3:  $u^{k+1} = GS_{hybrid}(r^k, \vec{d}^k, \vec{b}^k)$
- 4:  $\vec{d}^{k+1} = shrink_g(\nabla u^{k+1} + \vec{b}^k, \mu)$
- 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  $c_1^{k+1} = \int_{\Omega_k} f dx$ , and  $c_2^{k+1} = \int_{\Omega_k^c} f dx$

## 8: end while

Figure (2) presents a good estimation of the surface from the given set of scattered points. However, the estimation lacks smoothness because it has some “dimple” effects. These artifacts are the result of spikes in the solution  $f$  of the eikonal equation. In the next step, we will apply a shape denoising process to remove the artifacts on the contour and get a good estimate of the surface normals.

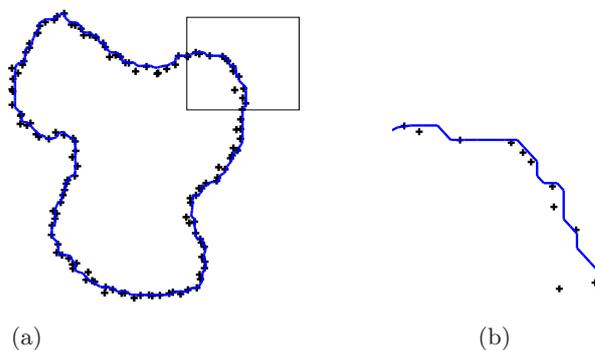


Figure 2: Estimation of the reconstructed surface using the model (11). The dark dots are the data points and the blue contour is the 0.5-level set of the solution  $u_1$  of (11). Fig. (b) presents an enlarged view of the rectangular region located in Fig. (a).

## 2.2 Step 2: Estimation of Surface Normals Based on the Weighted ROF Model

As discussed above, the model (11) gives a coarse approximation to the desired surface. We can obtain a “smoother” surface by using the coarse approximation to estimate the surface normals, and then using these normals to reconstruct the final surface. However, before the model (11) can be used for normal estimation, we must smooth out the “dimples” that arise from the spikes in the eikonal equation. We do this using a shape denoising method for implicit surface representations. From the result  $u_1$  of the convex model

(11), we obtain a smoothed surface  $u_2$ , using the convex model

$$u_2 = \min_{0 \leq u \leq 1} \int_{\Omega} d|\nabla u| + \frac{\lambda}{2}(u - u_1)^2, \quad (12)$$

which is well known as the weighted ROF (WROF) model. Note that  $d$  denotes the signed distance function of the scattered points, as defined in (2). Figure (3) shows that the WROF model has removed the artifacts produced by the first step of our algorithm, without decreasing too much the accuracy of the estimated surface.

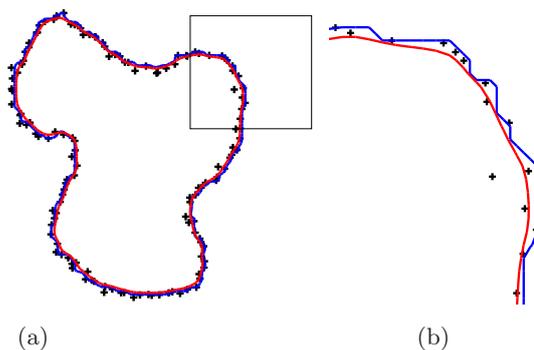
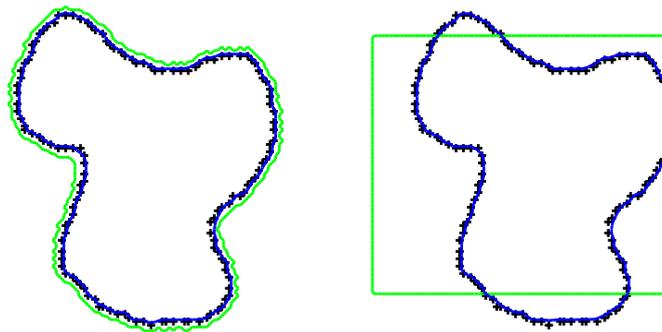


Figure 3: Shape smoothing using the weighted ROF model. The dark dots are the data points, the blue contour is the 0.5-level set of the solution  $u_1$  of (11) and the red contour is the 0.5-level set of the solution  $u_2$  of (12). Fig. (b) presents an enlarged view of the rectangular region located in Fig. (a).

We have observed that the solution  $u_2$  of the shape denoising model (12) is relatively insensitive to the exact choice of the function  $f$  used in the data part of the model (11). The choice of the edge detector in (11) is more critical. For example, we may replace  $f$  by its super level set function  $\bar{f} = Thresh_{\alpha}(f)$  or an indicator function of a rectangle as shown by the green contour in Figure 4. Here  $Thresh_{\alpha}(f) = 1$  if  $f \geq \alpha$ ;  $Thresh_{\alpha}(f) = 0$  otherwise. We can see that the final results are almost identical. When the function  $f$  is replaced by a binary function, it also allows us to speed up the minimization scheme by choosing a large fidelity constant. This is more useful for 3D computations. In addition, the super level set function does not have any spikes inherited from the solution of the eikonal equation.



(a) Super level set function.      (b) Indicator function of a rectangle

Figure 4: Solution  $u_2$  with two different choices of  $f$  used in the data part of (11). The dark dots are the data points, the green contour is the 0.5-level set of the function  $f$  used in the data part of (11), and the blue contour is the 0.5-level set of the solution  $u_2$  of (12). Fig. (a) defines the function  $f$  used in the data part of (11) as the super level set function of the solution of the eikonal equation. Fig. (b) defines  $f$  as an indicator function of a rectangle. Final results are almost identical, but the result in Fig. (a) was obtained more quickly.

Once we have obtained the denoised solution  $u_2$  of (12), we can use the result to estimate the surface normals at the data points using the following non-linear difference operator:

$$\vec{n}_p = \frac{\nabla u_2}{|\nabla u_2|} \quad (13)$$

Although step 2 of our algorithm refines the jagged approximation of the surface produced in step 1 by smoothing the contours. However, the results of steps 1 and 2 are not robust with respect to outliers, and may contain artifacts when extremely sparse data is used as input.

The final step of the reconstruction algorithm will make the reconstruction more robust to both outliers and sparse data by applying a Bregman iterative scheme.

### 2.3 Step 3: Surface Reconstruction using Bregman Iteration

In this section, we apply Bregman iteration to the results of step 2 in order to remove the effects of artifacts and outliers. Recall that we obtained the approximation  $u_2$  by smoothing the approximation  $u_1$ . The smoothness of  $u_2$  allows the surface normals to be accurately estimated using (13). These surface normals can then be used to obtain a new approximation to the surface by solving the following optimization problem, which is the continuous analogue of the discrete model of Lempitsky and Boykov [17].

$$u_3 = \arg \min_{0 \leq u \leq 1} \int_{\Omega} |\nabla u| - \lambda \langle \vec{n}_p, \nabla u \rangle, \quad (14)$$

where  $\vec{n}_p$  are the estimated normal to the data points. Using integration by parts, the above energy can be written

$$u_3 = \arg \min_{0 \leq u \leq 1} \int_{\Omega} |\nabla u| + \lambda \langle \text{div } \vec{n}_p, u \rangle \quad (15)$$

where  $\text{div } \vec{n}_p$  is the flux at each point of the surface  $\Gamma$ .

Inspired by [17], we compute the flux at a point  $Q$  as follows  $\text{div } \vec{n}_q = \sum_p \frac{1}{\sqrt{2\pi}\sigma} e^{-|\vec{x}_{PQ}|^2/2\sigma^2} \cdot \langle \vec{x}_{PQ}, \vec{n}_p \rangle$ , where  $\vec{x}_{PQ}$  is the vector centered at each neighboring data point  $P$  and pointed to the points  $Q$  inside a ball of radius  $r$  centered at  $P$ . In the calculation of the flux, we take into account both the distance to the data point and the angle between the surface normal and the vector from the data point to the neighboring points. Note that the flux at each data point is a signed function, which is negative inside the

surface and positive outside the surface. Figure 5(a) shows the semi-dense field of surface normals surrounding the data points. Figure 5(b) presents the flux at the data points. The split-Bregman algorithm is applied to solve (15) and the solution  $u_3$  will be used as the initial surface in the Bregman iterative scheme.

We now remove the artifacts from the approximation  $u_3$  using a Bregman iterative scheme. Schemes of this type were originally introduced in [29] to denoise images while preserving edges and image contrast. The Bregman method is defined by the following iterative scheme  $k = 4, 5, 6, \dots$

$$\begin{aligned} u^k &= \arg \min_u \int_{\Omega} |\nabla u| + \lambda \|v^{k-1} - u\|_2^2 \\ v^k &= v^{k-1} - u^k \end{aligned} \quad (16)$$

starting from  $v^3 = SDF(u_3)$ . We have found the choice  $\lambda = 0.5$  to be effective for an extremely wide range of data sets.

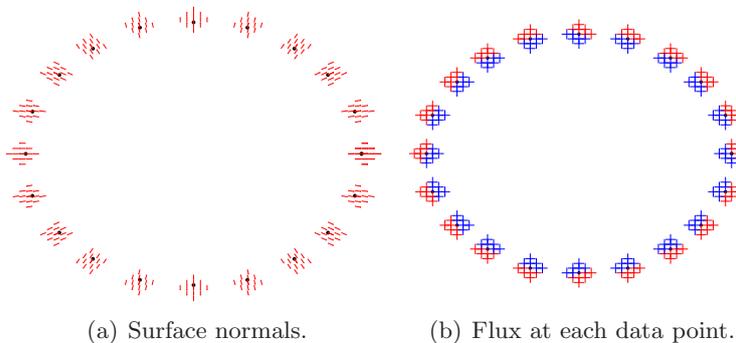


Figure 5: Figure (a) presents the semi-dense field of surface normals  $\vec{n}_p$ . Figure (b) presents the flux at each data point. The red crosses correspond to positive values of the divergence of flux  $div \vec{n}_p$ , and the blue crosses correspond to negative values.

We do not start the Bregman iterative scheme directly from the binary function  $z = u_3$ , but from a signed distance function of  $u_3$ . Indeed, the original Bregman method[28] was designed to denoise image intensities and normals of image level sets. The signed distance function better fits the Bregman framework since signed distance function gives a better representation of the interpolated surface than binary representation. Figure 6 presents

the evolution of the zero level set of  $u^k$  for  $k = 4, 5, 6$ . As expected, the first iteration produces a very smooth surface and successive iterations converge more and more to the green surface corresponding to the zero level set of  $u_3$ .

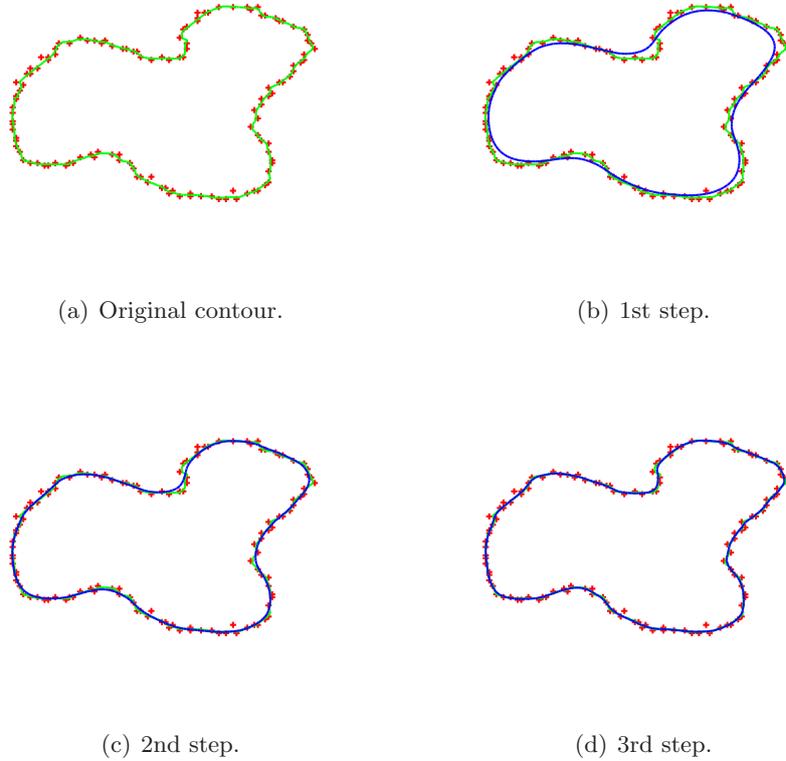


Figure 6: Bregman iterative scheme for surface reconstruction. The green contour in Figure (a) presents the zero level set of  $u_4$ . The blue contour in Figures (b-d) represent the successive Bregman iteration for  $k = 4, 5, 6$ .

We now show the importance of using the Bregman method in the challenging case of sparse data and the presence of outliers. Figure 7(a) presents the data points and the solution  $f$  of the eikonal equation (8). Figure 7(b) shows the result  $u_1$  computed by the model (11). Figure 7(c) is the result  $u_3$  given by the model (15). Observe the presence of an outlier at the bottom right of the contour. Figures 7(d-f) show the first three steps of the

Bregman iterative method. In the last figure, the reconstructed surface is smooth and accurate. The outlier is removed and all the major geometric features are recovered.

## 2.4 Algorithm

Here is the summary of our algorithm.

---

### Algorithm 1 Surface Reconstruction Algorithm

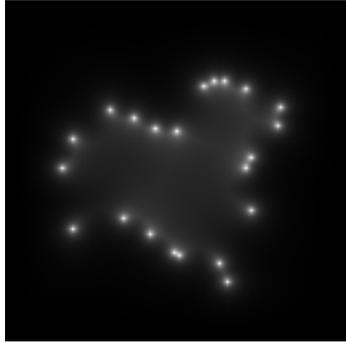
---

- 1: Step 1:
    - 1.1 Compute the distance function of the set of points using the eikonal equation (2).
    - 1.2 Compute the image  $f$  with the eikonal equation (8).
    - 1.3 Replace  $f$  by  $\tilde{f} = Thresh(f)$  and use the segmentation model (11) to find an initial estimate of the surface.
  - 2: Step 2:
    - 2.1 Use the WROF model (5) to regularize the geometry of the surface.
    - 2.2 Estimate the surface normals from the solution of the WROF model.
  - 3: Step 3:
    - 3.1 Calculate the divergence of normal field.
    - 3.2 Compute the initial function for Iterative Bregman scheme using (14).
    - 3.3 Iterate the Bregman scheme (16).
- 

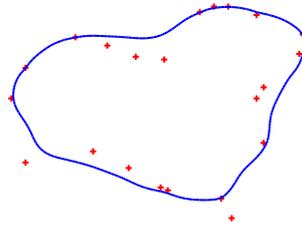
## 3 Numerical Results

In this section, we will present several three-dimensional surface reconstruction results. Figure 8 shows the different steps to reconstruct the “bunny” surface. Figure 9 also reconstructs the bunny surface, but with a higher resolution. We also show the reconstructions of the “cow” surface (Figure 10), the “Buddha” surface (Figure 11), the “dragon” surface (Figure 12), the “armadillo” surface (Figure 13), and the “skeleton hands” surface (Figure 14). Figure 15 shows the reconstruction of the “church” surface.

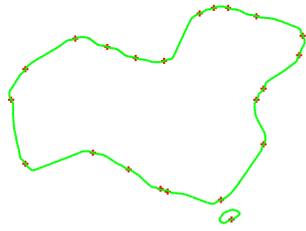
This last test shows the limitations of our reconstruction method. The result is not fully satisfactory because our method is designed to reconstruct closed surfaces. The surface in Figure 15 is not completely closed, which limits the applicability of the level set representation. In all experiments for 3D real data, the parameters are the same, i.e.  $\mu = 0.01$  for (11) and



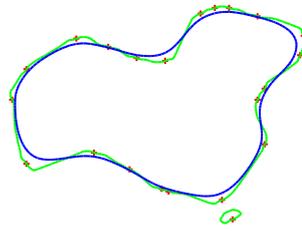
(a) Solution  $f$  from the eikonal equation (8)



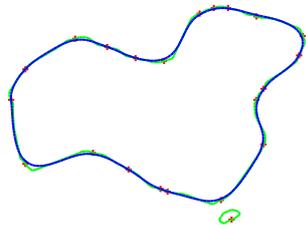
(b) Solution  $u_1$  from (11).



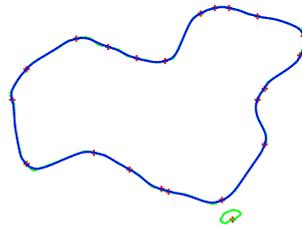
(c) Solution  $u_3$  from (15).



(d) 1st Bregman iteration.



(e) 2nd Bregman iteration.



(f) 3rd Bregman iteration.

Figure 7: Surface reconstruction for sparse data and in the presence of outliers. Fig. (a) presents the data points and the function  $f$  solution of the eikonal equation (8). Fig. (b) shows the result  $u_1$  computed by the model (11). Fig. (c) is the result  $u_3$  given by the model (15). Figs (d-f) show the first three steps of the Bregman iterative method. Observe that the final result is accurate, without outliers.

$\lambda = 0.5$  for (15). Mesh sizes and computing times are reported for all of these examples in table 1.

Table 1: Computational times (for the whole algorithm)

data set	size	time (in seconds)
bunny	71x71x60	13.01
bunny	141x140x114	198.25
cow	53x119x81	22.41
Buddha	71x143x71	35.55
Buddha	121x264x121	242.17
dragon	221x162x110	214.25
armadillo	141x164x130	255.69
skeleton hand	221x161x90	150.27

## 4 Conclusion

Several attempts have been made to apply variational methods to reconstruction problems. These methods have been proven accurate and robust to noise, but they have been limited by speed. In this paper, we have significantly improved the reconstruction speed of these models by proposing a convex model that can be efficiently minimized using the split-Bregman algorithm. Furthermore, our algorithm only requires a set of unorganized data points as input. Finally, we have shown that our method can deal with dense data, sparse data, and the presence of outliers.

## References

- [1] R. Courant, K. Friedrichs and H. Lewy, *On the partial difference equations of mathematical physics*, IBM Journal, pp. 215-234, 1967.
- [2] D. Mumford and J. Shah, *Optimal approximation by piecewise smooth functions and associated variational problems*, Commun. Pure Appl. Math, v.42, pp. 577-685, 1989.
- [3] L. Piegl and W. Tiller, *The NURBS book*, Springer-Verlag, Berlin, Germany, 1996.

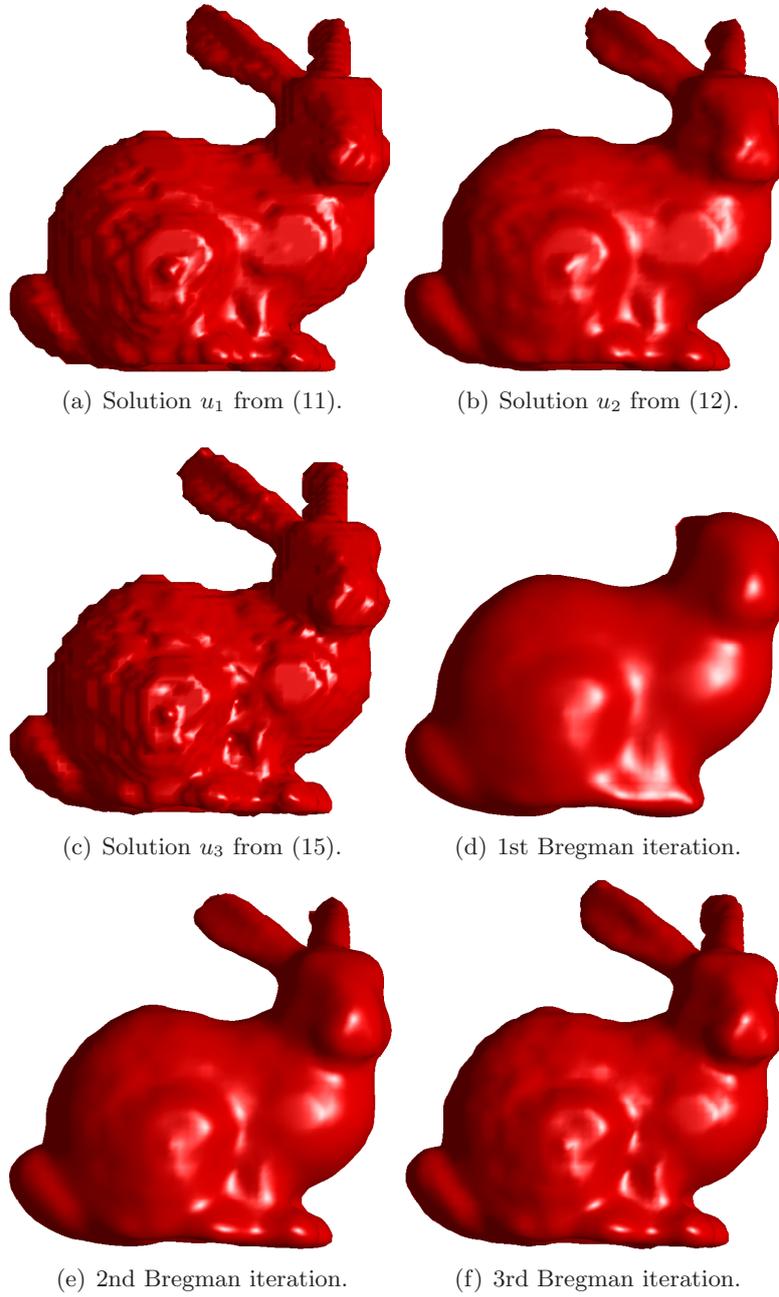


Figure 8: Surface reconstruction of “bunny”. Mesh size is  $71 \times 71 \times 60$ .

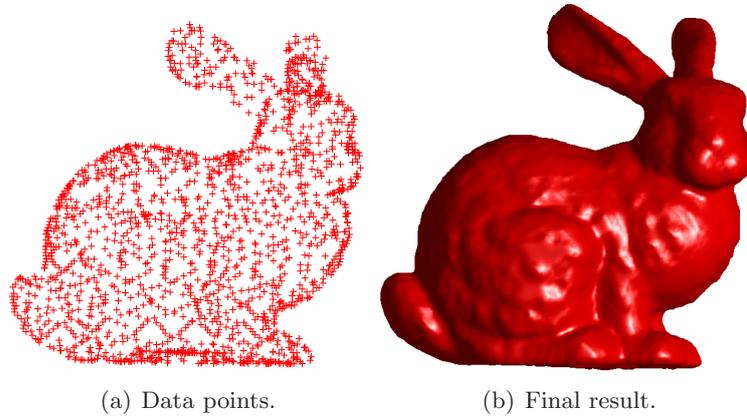


Figure 9: Surface reconstruction of “bunny”. Mesh size is  $141 \times 140 \times 114$ .

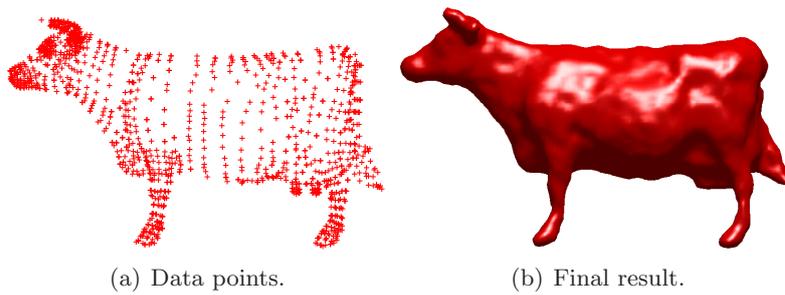


Figure 10: Surface reconstruction of “cow”. Mesh size is  $53 \times 119 \times 81$ .

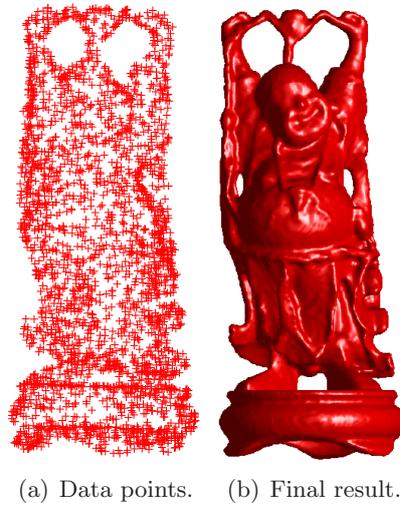


Figure 11: Surface reconstruction of “Buddha”. Mesh size is  $121 \times 264 \times 121$ .

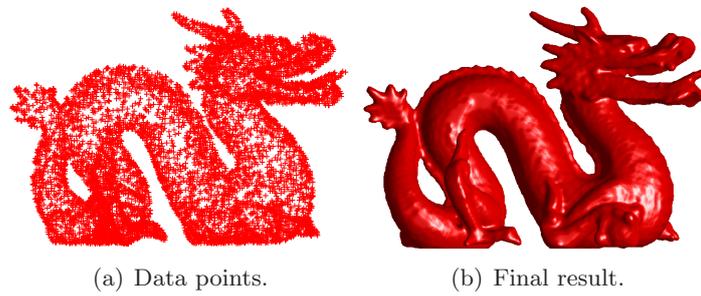


Figure 12: Surface reconstruction of “dragon”. Mesh size is  $221 \times 162 \times 110$ .

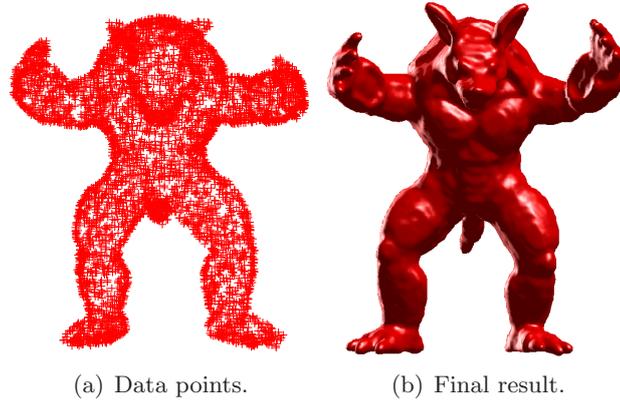


Figure 13: Surface reconstruction of “armadillo”. Mesh size is  $141 \times 164 \times 130$ .

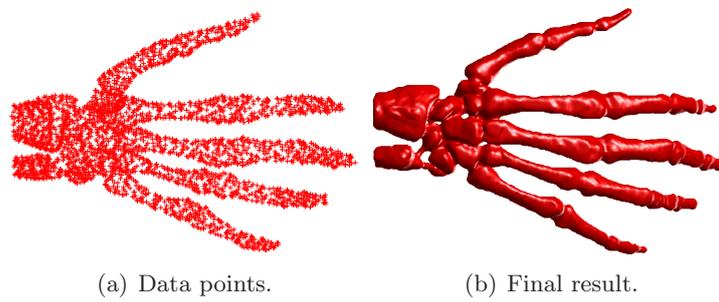


Figure 14: Surface reconstruction of “skeleton hand”. Mesh size is  $221 \times 161 \times 90$ .

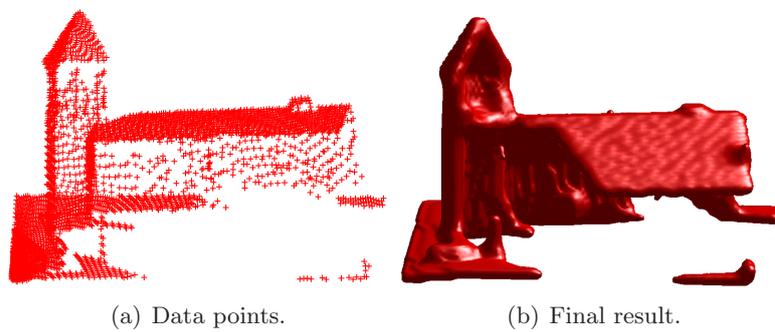


Figure 15: Surface reconstruction of “church”. Mesh size is  $151 \times 124 \times 107$ .

- [4] D.F. Rogers, *An Introduction to NURBS*, Morgan Kaufmann, 2003.
- [5] N. Amenta, M. Bern, and M. Kamvysselis, *A new Voronoi-based surface reconstruction algorithm*, Proc. SIGGRAPH 98, pp. 415-421, 1998.
- [6] N. Amenta, M. Bern, and D. Eppstein, *The crust and the  $\beta$ -skeleton: combinatorial curve reconstruction*, Graphical Models and Image Processing, v.60(2), pp. 125-135, 1998.
- [7] N. Amenta, M. Bern, and M. Kamvysselis, *A new Voronoi-based surface reconstruction algorithm*, Proc. SIGGRAPH98, pp. 415-421, 1998.
- [8] J.D. Boissonnat, *Geometric structures for three dimensional shape reconstruction*, ACM Trans. Graphics 3, pp. 266-286, 1984.
- [9] H. Edelsbrunner, *Shape reconstruction with Delaunay complex*, Springer-Verlag, Proc. of LATIN98: Theoretical Informatics, v.1380, pp.119-132, 1998.
- [10] H. Edelsbrunner and E. P. Mucke, *Three dimensional  $\alpha$  shapes*, ACM Trans. Graphics 13, pp.43-72, 1994.
- [11] B. Curless and M. Levoy, *A volumetric method for building complex models from range images*, SIGGRAPH96 Proceedings, pp. 303-312, 1996.
- [12] Tony F. Chan and Luminita A. Vese, *Active Contours without Edges*, IEEE Transactions on Image Processing, v.10, pp.266-277, 2001.
- [13] D.F. Rogers, *An introduction to NURBS: with historical perspective*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.
- [14] H. Hoppe, T. Deroose, T. Duchamp, J. Mcdonald, and W. Stuetzle. *Surface reconstruction from unorganized points*, Computer Graphics, 26(2), pp.71-78, 1992.
- [15] H.K. Zhao, S. Osher, B. Merriman and M. Kang. *Implicit, Nonparametric Shape Reconstruction from Unorganized Points Using A Variational Level Set Method*, Computer Vision and Image Understanding, v.80, pp.295-319, 1998.
- [16] H.K. Zhao, S. Osher and R. Fedkiw. *Fast surface reconstruction using the level set method*, Proceedings. IEEE workshop, pp. 194-201, 2001.

- [17] V. Lempitsky and Y. Boykov, *Global Optimization for Shape Fitting*, CVPR, 2007.
- [18] T. Goldstein, X. Bresson and S. Osher, *Geometric Applications of the Split Bregman Method: Segmentation and Surface Reconstruction*, CAM09-06, 2009.
- [19] Y.R. Tsai, *Rapid and accurate computation of the distance function using grids*, J.Comp.Phys., v.178, no. 1, pp. 175-195, 2002.
- [20] H.K. Zhao, *A Fast Sweeping Method for Eikonal Equations*, Mathematics of Computation, v.74, pp. 603-627, 2005.
- [21] Fred Almgren, Jean E. Taylor, and Lihe Wang, *Curvature-driven flows: a variational approach*, SIAM J. Control Optim., v.31(2), pp.387-438, 1993.
- [22] Antonin Chambolle, *An algorithm for total variation minimization and applications*. J. Math. Imaging Vis., v.20(1-2), pp. 89-97, 2004.
- [23] Xavier Bresson and Tony Chan, *Active contours based on chambolle's mean curvature motion*, IEEE International Conference on Image Processing, pp.33-36, 2007.
- [24] D. Mumford and J. Shah. *Optimal approximation by piecewise smooth functions and associated variational problems*, Comm. Pure Appl. Math., v.42, pp. 577-685, 1989.
- [25] T. Chan, S. Esedoglu, and M. Nikolova, *Algorithms for finding global minimizers of image segmentation and denoising models*, SIAM Journal on Applied Mathematics, 2004.
- [26] X. Bresson, S. Esedoglu, P. Vandergheynst, J. Thiran, S. Osher, *Fast Global Minimization of the Active Contour/Snake Model*, Journal of Mathematical Imaging and Vision, v.28, pp.151-167, 2007.
- [27] T. Goldstein, S. Osher, *The split bregman method for L1 regularized problems*, SIAM Journal on Imaging Sciences, v.2(2), pp.323-343, 2009.
- [28] M. Burger, S. Osher, J.J. Xu and G. Gilboa, *Nonlinear Inverse Scale Space Methods for Image Restoration*, Communications in Mathematical Sciences, pp.2006, 2005.

- [29] S. Osher, M. Burger, D. Goldfarb, J. Xu and W. Yin, *An iterative regularization method for total variation-based image restoration*, Multiscale Modeling and Simulation, v.4, pp.460-489, 2005.
- [30] Y. Boykov and V. Kolmogorov, *Computing geodesics and minimal surfaces via graph cuts*, ICCV03(I), pp. 26, 2003.