An Efficient Algorithm for Level Set Method Preserving Distance Function

Virginia Estellers[⋆], Dominique Zosso[⋆], Rongjie Lai[†], Jean-Philippe Thiran[⋆], Stanley Osher[◇], Xavier Bresson[○]

[⋆] Department of Electrical Engineering, Ecole Polytechnique Fédérale de Lausanne
[†] Department of Mathematics, University of Southern California, Los Angeles
[◇] Department of Mathematics, University of California, Los Angeles
[○] Department of Computer Science, City University of Hong Kong

September 10, 2011

Abstract

The level set method [31] is a popular technique for tracking moving interfaces in several disciplines including computer vision and fluid dynamics. However, despite its high flexibility, the original level set method is limited by two important numerical issues. Firstly, the level set method does not implicitly preserve the level set function as a distance function, which is necessary to estimate accurately geometric features s.a. the curvature or the contour normal. Secondly, the level set algorithm is slow because the time step is limited by the standard CFL condition, which is also essential to the numerical stability of the iterative scheme. Recent advances with graph cut methods [4, 3] and continuous convex relaxation methods [7, 5, 16] provide powerful alternatives to the level set method for image processing problems because they are fast, accurate and guaranteed to find the global minimizer independently to the initialization. These recent techniques use binary functions to represent the contour rather than distance functions, which are usually considered for the level set method. However, the binary function cannot provide the distance information, which can be essential for some applications s.a. the surface reconstruction problem from scattered points and the cortex segmentation problem in medical imaging. In this paper, we propose a fast algorithm to preserve distance functions in level set methods. Our algorithm is inspired by recent efficient ℓ^1 optimization techniques, which will provide an efficient and easy to implement algorithm. It is interesting to note that our algorithm is not limited by the CFL condition and it naturally preserves the level set function as a distance function during the evolution, which avoids the classical re-distancing problem in level set methods. We apply the proposed algorithm to carry out image segmentation, where our methods proves to be 5 to 6 times faster than standard distance preserving level set techniques. We also present two applications where preserving a distance function is essential. Nonetheless, our method stays generic and can be applied to any level set methods that require the distance information.

1 Introduction

In the last twenty years, the level set method (LSM) of Osher and Sethian [31] has become a popular numerical technique for tracking moving interfaces in computational geometry, fluid mechanics, computer graphics, computer vision and material sciences. The main reasons of its success are the high flexibility of this method to adapt to different problems, the ability to deal with changes of topology (contour breaking and merging) without any extra functions and the guarantee of the existence of solutions in the class of viscosity partial differential equations (PDEs). Moreover, extensive numerical algorithms based on Hamilton-Jacobi equations have been developed, accurately handling shocks and providing stable numerical schemas.

The key idea of the LSM is to implicitly represent a contour or interface as the zero level set of a higher dimensional function, called the level set function (LSF), and formulate the evolution of the contour through the evolution of the level set function. For closed contours, signed distance functions (SDFs) were originally adopted to represent level set functions because they directly provide stability and accuracy to the LSM [29, 30]. In the last years, however, new approaches have proposed to use binary functions, rather than distance functions, to represent the contour. This change of representation allows to use fast and convex optimization techniques with graph cut methods [4, 3] and convex relaxation methods [7, 5, 16] to provide powerful alternatives to the distance preserving level set method. Nevertheless, distance functions are still essential in several applications, e.g. medical image segmentation, surface reconstruction or special effects in computer graphics. For this reason, it is important to develop a fast and accurate algorithm for distance preserving level set methods.

A major issue of distance preserving level set methods is the limited speed of the existing algorithms, which are based on Hamilton-Jacobi equations and upwind schemes [31, 30]. Two main issues handicap these iterative methods. First, the speed of the algorithms is limited by the CFL condition [9], which is a necessary condition to guarantee the stability of PDE-based iterative schemes. Secondly, the level set energy is hard to optimize because of the ℓ^1 - based total variation (TV) term, which is not differentiable and is usually regularized. However, the regularization significantly slows down the minimization process and does not provide an exact solution to the problem. In our work, we will overcome these speed limitations using recent efficient techniques in ℓ^1 optimization.

Another major issue of the LSM, pointed out by Gomes and Faugeras in [18], is a contradiction between the theory and the implementation when LSF are represented by SDF. Indeed, the LSM does not intrinsically preserve the SDF during the contour propagation because SDFs are not solutions of the Hamilton-Jacobi equations associated to the LSM. Additional techniques are then necessary to preserve the LSF as an SDF during contour evolution. Unfortunately, no fully satisfying methods have been proposed so far. The two most common approaches that have been suggested to fix this problem are either re-distancing regularly the LSF as an SDF (this procedure is called re-distancing or re-initialization), or constraining the LSF to remain an SDF during the contour evolution.

Re-distancing is the most common approach. It consists in stopping the evolution of the LSF periodically and re-initializing it as an SDF while preserving the zero level set. This approach introduces the questions of when and how to re-initialize the LSF. It is hard to say when the re-distancing must be applied as there is a trade-off between speed (each

re-distancing task takes time) and accuracy (the LSM will develop irregularities during the evolution without re-distancing). Similarly, the re-initialization can be performed with different LSM techniques, using PDEs [36] or the fast marching algorithm [1, 29]. The main issue with the re-distancing approach is the difficulty to preserve exactly the location of the zero level set during the re-distancing process, which might shift the moving interface to undesired positions.

The second approach aims at constraining the LSF to stay an SDF during the contour evolution, avoiding the previous re-distancing procedure altogether. Gomes and Faugeras introduced in [18] a new level set formulation to restrict the LSF to an SDF. The new formulation consists of three coupled PDEs, which makes the analysis of the existence of a solution and the numerical implementation more difficult than the standard LSM. More recently, Li *et al.* [24, 25] proposed to add a penalty term in the level set energy to constrain the LSF to be close to an SDF. They also developed a new algorithm, simpler and more efficient than the conventional LSM method. However, the time step of the algorithm is still restricted by the CFL condition and the SDF property is only encouraged but not enforced. We will see that our method overcomes all of these limitations and provides an efficient way to constrain exactly the LSF to be an SDF.

The level set method was first introduced in computer vision to carry out image segmentation [6, 21, 8] and then extended to other tasks as stereo reconstruction [11], object tracking [32], and object recognition [23]. In this paper, we focus on image segmentation and surface reconstruction, but the proposed method can be easily extended to other problems. In image segmentation, the level set method re-formulates the parametric active contour into a non-parametric energy minimization problem (i.e. independent of the contour discretization). The active contour is embedded as the zero level set of the LSF, which moves according to the active contour evolution equation and drives its zero level set to the edges of the desired object. The level set formulation easily includes edge-, region- and shape-based terms in the image segmentation criteria, proving the high flexibility of the LSM to adapt to different tasks and justifying its extensive use in the field. Similar segmentation models have been applied to the problem of reconstructing a surface form unorganized data points [41]. In that case, the LSM allows us to obtain an implicit representation of the surface by an SDF, avoiding complex 3D parametrization methods which require a prior knowledge of the topology of the surface, and providing an easy and reliable way to directly estimate surface normals and curvature from the level set function.

Here, we propose a fast and accurate algorithm for distance preserving level set methods. We constrain the LSF to remain an SDF via a constrained minimization problem. The constrained minimization problem is hard to solve directly, so we propose to split the original hard problem into sub-optimization problems which are easier to solve, and combine them together using an augmented Lagrangian approach. This idea is borrowed from the split-Bregman method [17] (and more generally from the alternating direction methods of multipliers [12, 13, 26, 2, 35]), which is an efficient ℓ^1 minimization method originally developed to solve the compressed sensing problem. The proposed algorithm holds several important properties. Firstly, it is fast because it is not limited by the CFL condition, leading to a method 5 to 6 times faster than most distance preserving LSM algorithms for the image segmentation problem. This improvement is due to the splitting strategy and the reformulation of a constraint minimization, and allows us to

deal with the non-differentiability of the TV norm and go beyond the CFL time step restriction. Secondly, our algorithm preserves the LSM as an SDF, avoiding the classical re-distancing problem and providing desirable properties for some applications. For example, this makes an important difference in surface reconstruction, where surface normals can be fast and reliably estimated during the surface evolution instead of being required as input data s.a. [34, 22], and in medical image segmentation, where the distance information can be exploited to include topology restrictions into the problem. Finally, our algorithm is easy to implement because the iterative scheme is based on standard minimization problems.

The rest of the paper is organized as follows. In Section 2 we review very briefly the level set method applied to image segmentation and surface reconstruction. We introduce our algorithm to solve efficiently the level set method in Sections 3 and 4. Section 5 presents the results and Section 6 draws the conclusions.

2 Level Set Method for Image Segmentation and Surface Reconstruction

Level set method can be applied to perform image segmentation using the active contour method. In this context, the segmentation problem is defined as the following energy minimization problem w.r.t. a contour C:

$$\min_{C\subset\Omega} \int_C w_b(s)ds + \int_{C_{in}} w_r^{in}(x)dx + \int_{C_{out}} w_r^{out}(x)dx, \tag{1}$$

where Ω is the image domain, s is the arc-length parametrization and the first term weights the length of C by an edge detector function w_b, such as in [6, 21]. In the other terms, C_{in}, C_{out} designate the regions inside and outside the contour C and w_r^{in}, w_r^{out} are region-based terms, such as the ones proposed in [8]. We adopt this minimization model for image segmentation, as it represents a large class of active contour models published in the literature.

Actually, similar models have also been used to reconstruct smooth surfaces from unorganized data points. Given a set of (noisy) points $\{x_i\}_{1\leq i\leq N}$ lying close to the unknown surface, a smooth estimate of the surface S can be reconstructed by minimizing the area energy weighted by the distance to the set of points $\{x_i\}_{1\leq i\leq N}$, i.e. $w_b(x) = d_{\{x_i\}}(x)$ [20, 41]. Inspired by [22, 39], we propose a model which also includes a region-based term w_r^{in} to improve the performance with sparse data sets. We adapt the model proposed by Lempitsky and Boykov [22], which was designed to align the normals of the reconstructed surface to the pre-computed orientations $\{n_i\}_{1\leq i\leq N}$ of the data points. Based on this surface information, a semi-dense vector field of the surface normal $\hat{n}$ can be estimated everywhere in space and the unknown surface can thus be reconstructed by iteratively maximizing the alignment of the reconstructed surface normal $\mathcal{N}$ to $\hat{n}$. Making use of the divergence theorem, the function w_r^{in} can be identified as follows:

$$\arg\max_{C\subset\Omega} \int_C \hat{n}\cdot\mathcal{N}ds = \arg\min_{C\subset\Omega} \int_{C_{in}} -\operatorname{div}\hat{n}dx, \tag{2}$$

which corresponds to defining $w_r^{in} = -\operatorname{div}\hat{n}$. The flux of the semi-dense normal field at the point x is estimated with the formula:

$$\operatorname{div}\hat{n}\,(x) = \sum_{1\leq i\leq N} \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{|x-x_i|^2}{2\sigma^2}} \langle x - x_i, n_i\rangle, \tag{3}$$

where $x - x_i$ denotes the vector centred at x_i and pointing at x and n_i is the surface normal estimated at x_i. The proposed distance preserving LSM can estimate the surface normal during the reconstruction process: $n_i = \nabla\phi\,(\boldsymbol{x}_i)$, where ϕ is the SDF. To sum up, the proposed functions in (1) for the surface reconstruction problem are: $w_b(x) = d_{\{x_i\}}(x)$, $w_r^{in} = -\operatorname{div}\hat{n}$, and $w_r^{out} = 0$. Unlike the method of Lempitsky and Boykov, the proposed surface reconstruction algorithm does not need to know *a priori* the normal of the surface at the data points.

The level set method can be applied to solve (1) in general (and specifically the image segmentation problem and the surface reconstruction problem), by re-writing the minimization in terms of a level set function $\phi : \Omega \to \mathbb{R}$ as follows:

$$\min_{\phi} \int_{\Omega} w_b(x)|\nabla H(\phi)| + w_r(x)H(\phi) \quad \text{s.t.} \quad |\nabla\phi| = 1, \tag{4}$$

where the curve in $\mathbb{R}^2$ (or surface in $\mathbb{R}^3$) C is represented by the zero level set of ϕ, H is the Heaviside function, and the constraint $|\nabla\phi| = 1$ guarantees the level set function to be a signed distance function [36].

3 An Efficient Algorithm for Level Set Method preserving Signed Distance Function

In this section, we introduce an efficient algorithm to solve the level set minimization problem (4). The main idea is to split the original hard problem (4) into sub-optimization problems which are well-known and easy to solve, and combine them together using an augmented Lagrangian. This idea is borrowed from the split-Bregman method [17], which is an efficient ℓ^1 optimization method recently introduced in image processing to solve the compressed sensing problem.

Let us consider the following constrained minimization problem, which is equivalent to the original LSM problem (4):

$$\min_{\phi,\varphi,u,\boldsymbol{q},\boldsymbol{p}} \int_{\Omega} w_b(x)|\boldsymbol{q}| + w_r(x)u \quad \text{s.t.} \quad \begin{cases} u{=}H(\varphi) \\ \varphi{=}\phi \\ \boldsymbol{q}{=}\nabla u \\ \boldsymbol{p}{=}\nabla\phi \\ |\boldsymbol{p}|{=}1 \end{cases} \tag{5}$$

where we introduce functions $\varphi(x), u(x) \in \mathbb{R}$, $\boldsymbol{q}(x), \boldsymbol{p}(x) \in \mathbb{R}^n$ (boldface letters are used to denote vector functions and $n = 2$ for 2D images, and $n = 3$ for 3D images). The proposed splitting approach makes the original problem (4) easier to solve because (5) can better handle the non-differentiability and non-linearity of (4). Indeed, it is known from [17, 38] that the minimization of $|\nabla\phi|$ can be carried out efficiently by decoupling the ℓ^1-norm $|.|$ and the gradient operator ∇. The term $|\nabla\phi|$ is thus replaced by $|\boldsymbol{p}|$ and

$\boldsymbol{p} = \nabla\phi$, and the term $|\nabla H(\phi)|$ can be changed into $|\boldsymbol{q}|, \boldsymbol{q} = \nabla u, u = H(\varphi)$ and $\varphi = \phi$, where φ is introduced to handle the non-linear term $H(\phi)$.

Next, we want to reformulate this constrained minimization problem as an unconstrained optimization task. This can be done with an augmented Lagrangian approach [13], which translates the constraints into pairs of Lagrangian multiplier and penalty terms. Let us define the augmented Lagrangian energy associated to (5):

$$\begin{aligned}\mathcal{L}(\phi, \varphi, u, \boldsymbol{q}, \boldsymbol{p}, \Lambda) = \int_\Omega & w_b|\boldsymbol{q}| + w_r u \\ & + \lambda_1(\varphi - \phi) + \frac{r_1}{2}(\varphi - \phi)^2 \\ & + \lambda_2(u - H(\varphi)) + \frac{r_2}{2}(u - H(\varphi))^2 \\ & + \boldsymbol{\lambda_3} \cdot (\boldsymbol{q} - \nabla u) + \frac{r_3}{2}|\boldsymbol{q} - \nabla u|^2 \\ & + \boldsymbol{\lambda_4} \cdot (\boldsymbol{p} - \nabla\phi) + \frac{r_4}{2}|\boldsymbol{p} - \nabla\phi|^2 \quad \text{s.t.} \quad |\boldsymbol{p}| = 1 \end{aligned} \tag{6}$$

where $\Lambda = (\lambda_1, \lambda_2, \boldsymbol{\lambda_3}, \boldsymbol{\lambda_4})$ are the Lagrangian functions, i.e. $\lambda_1(x), \lambda_2(x) \in \mathbb{R}$, $\boldsymbol{\lambda_3}(x)$, $\boldsymbol{\lambda_4}(x) \in \mathbb{R}^n$, and $r_1, \ldots, r_4$ are positive constants. The constraint minimization problem (5) reduces to finding the saddle-point of the augmented Lagrangian energy $\mathcal{L}$ [13]. The solution to the saddle point problem (6) can be approximated by the iterative Algorithm 1.

Algorithm 1 Augmented Lagrangian method for distance preserving level set methods

1: Initialize $\phi, \varphi, u, \boldsymbol{q}, \boldsymbol{p}, \Lambda$

2: Find a minimizer of $\mathcal{L}$ with respect to variables $(\phi, \varphi, u, \boldsymbol{q}, \boldsymbol{p})$ with fixed Lagrange multipliers Λ^k:

$$(\phi^k, \varphi^k, u^k, \boldsymbol{q}^k, \boldsymbol{p}^k) = \underset{\phi,\varphi,u,\boldsymbol{q},\boldsymbol{p}}{\arg\min}\, \mathcal{L}\left(\phi, \varphi, u, \boldsymbol{q}, \boldsymbol{p}, \Lambda^{k-1}\right) \quad \text{s.t.} \quad |\boldsymbol{p}| = 1 \tag{7}$$

3: Update Lagrange multipliers

$$\lambda_1^k = \lambda_1^{k-1} + r_1(\varphi^k - \phi^k) \tag{8}$$
$$\lambda_2^k = \lambda_2^{k-1} + r_2(u^k - H\left(\varphi^k\right)) \tag{9}$$
$$\boldsymbol{\lambda_3}^k = \boldsymbol{\lambda_3}^{k-1} + r_3(\boldsymbol{q}^k - \nabla u^k) \tag{10}$$
$$\boldsymbol{\lambda_4}^k = \boldsymbol{\lambda_4}^{k-1} + r_4(\boldsymbol{p}^k - \nabla\phi^k) \tag{11}$$

4: Stop the iterative process when $\|\phi^k - \phi^{k-1}\|_2 < \epsilon$.

We initialize ϕ^0, φ^0 with the signed distance function of the initial contour, $u^0 = H(\phi_0)$, $\boldsymbol{q}^0 = \nabla u^0$, $\boldsymbol{p}^0 = \nabla\phi^0$ and the Lagrange multipliers $\Lambda^0 = 0$. At each iteration, an alternating minimization method is used to find an approximate minimizer of $\mathcal{L}(\phi, \varphi, u, \boldsymbol{q}, \boldsymbol{p}, \Lambda^{k-1})$ considering the Lagrange multipliers Λ^{k-1} fixed. Then the Lagrange multipliers are updated with the residual associated to each constraint and the process is repeated until the change of the level set function ϕ falls below a certain threshold ϵ, which happens at convergence.

In general, it is difficult to find the exact minimizer of the minimization problem (7) because the energy (6) is not convex (there is no guarantee of convergence). However, experiments show that a good approximation can be found by the alternating direction method of multipliers [13]. An approximate solution is thus computed by iteratively alternating the minimization of $\mathcal{L}(\phi, \varphi, u, \boldsymbol{q}, \boldsymbol{p}, \Lambda^{k-1})$ w.r.t. each variable while considering the others fixed. This leads to Algorithm 2.

The next step is to determine the solutions of the five sub-minimization problems (12),(13),(14),(15) and (16), which can actually be computed efficiently.

Algorithm 2 Alternate minimizations for an approximate solution of (7)

1: Initialize $\tilde{\phi}^0 = \phi^{k-1}$, $\tilde{\varphi}^0 = \varphi^{k-1}$, $\tilde{u}^0 = u^{k-1}$, $\tilde{\boldsymbol{q}}^0 = \boldsymbol{q}^{k-1}$, $\tilde{\boldsymbol{p}}^0 = \boldsymbol{p}^{k-1}$.

2: For $l = 1, ..., L$ and and fixed Lagrange multipliers Λ^k, solve the following sub-problems alternatively:

$$\tilde{\phi}^l = \arg\min_{\phi} \mathcal{L}(\phi, \tilde{\varphi}^{l-1}, \tilde{u}^{l-1}, \tilde{\boldsymbol{q}}^{l-1}, \tilde{\boldsymbol{p}}^{l-1}, \Lambda^{l-1}) \tag{12}$$

$$\tilde{\varphi}^l = \arg\min_{\varphi} \mathcal{L}(\tilde{\phi}^l, \varphi, \tilde{u}^{l-1}, \tilde{\boldsymbol{q}}^{l-1}, \tilde{\boldsymbol{p}}^{l-1}, \Lambda^{l-1}) \tag{13}$$

$$\tilde{u}^l = \arg\min_{u} \mathcal{L}(\tilde{\phi}^l, \tilde{\varphi}^l, u, \tilde{\boldsymbol{q}}^{l-1}, \tilde{\boldsymbol{p}}^{l-1}, \Lambda^{l-1}) \tag{14}$$

$$\tilde{\boldsymbol{q}}^l = \arg\min_{\boldsymbol{q}} \mathcal{L}(\tilde{\phi}^l, \tilde{\varphi}^l, \tilde{u}^l, \boldsymbol{q}, \tilde{\boldsymbol{p}}^{l-1}, \Lambda^{l-1}) \tag{15}$$

$$\tilde{\boldsymbol{p}}^l = \arg\min_{\boldsymbol{p}} \mathcal{L}(\tilde{\phi}^l, \tilde{\varphi}^l, \tilde{u}^l, \tilde{\boldsymbol{q}}^l, \boldsymbol{p}, \Lambda^{l-1}) \quad \text{s.t.} \quad |\boldsymbol{p}| = 1 \tag{16}$$

3: Set $(\phi^k, \varphi^k, u^k, \boldsymbol{q}^k, \boldsymbol{p}^k) = (\tilde{\phi}^L, \tilde{\varphi}^L, \tilde{u}^L, \tilde{\boldsymbol{q}}^L, \tilde{\boldsymbol{p}}^L)$.

4 Sub-Minimization Problems

In this Section, we simplify notation by omitting the super-index and the tilde symbol in the sub-minimization problems (12)-(16).

4.1 Sub-Minimization Problems w.r.t. ϕ and u

The sub-minimization problems (12) and (14) can be written as follows:

$$\min_{\phi} \int_{\Omega} \frac{r_1}{2}\Big(\phi - \Big(\varphi + \frac{\lambda_1}{r_1}\Big)\Big)^2 + \frac{r_4}{2}\Big|\nabla\phi - \Big(\boldsymbol{p} + \frac{\boldsymbol{\lambda_4}}{r_4}\Big)\Big|^2 \tag{17}$$

$$\min_{u} \int_{\Omega} w_r u + \frac{r_2}{2}\Big(u - \Big(H(\varphi) - \frac{\lambda_2}{r_2}\Big)\Big)^2 + \frac{r_3}{2}\Big|\nabla u - \Big(\boldsymbol{q} + \frac{\boldsymbol{\lambda_3}}{r_3}\Big)\Big|^2 \tag{18}$$

The Euler-Lagrange equation of (17) is:

$$(-r_4\Delta + r_1)\phi = -r_4 \operatorname{div} \boldsymbol{p} + \frac{\boldsymbol{\lambda_4}}{r_4} + r_1\Big(\varphi + \frac{\lambda_1}{r_1}\Big) \tag{19}$$

which can be solved efficiently by the fast Fourier transform (FFT) as proposed in [38, 37]. Let us denote by $f(i,j)$ a scalar 2D function discretized at the pixel location (i,j) in the discrete image domain $\Omega = [1, N_x] \times [1, N_y]$. We also define the identity operator $\mathcal{I}f(i,j) = f(i,j)$ and shifting operators:

$$\mathcal{S}_x^{\pm} f\,(i,j) = f\,(i \pm 1, j) \qquad\qquad \mathcal{S}_y^{\pm} f\,(i,j) = f\,(i, j \pm 1)\,, \tag{20}$$

and write the discretization of (19) as:

$$\left[r_4\left(\mathcal{S}_x^- - 2\mathcal{I} + \mathcal{S}_x^+ + \mathcal{S}_y^- - 2\mathcal{I} + \mathcal{S}_y^+\right) - r_1\right]\phi(i,j) = h(i,j), \tag{21}$$

whose right hand side is discretized as

$$\begin{aligned} h(i,j) = &-r_4(\mathcal{I} - \mathcal{S}_x^-)p_x(i,j) - r_4(\mathcal{I} - \mathcal{S}_y^-)p_y(i,j) + \\ &- (\mathcal{I} - \mathcal{S}_x^-)\lambda_{4x}(i,j) - (\mathcal{I} - \mathcal{S}_y^-)\lambda_{4y}(i,j) + r_1\varphi(i,j) + \lambda_1(i,j). \end{aligned} \tag{22}$$

Then we apply the discrete Fourier transform $\mathcal{F}$, taking into account that the shifting operators in frequency domain correspond to:

$$\mathcal{F}\mathcal{S}_1^{\pm} f\,(y_i, y_j) = e^{\pm \imath z_i}\mathcal{F}f\,(y_i, y_j) \qquad z_i = \frac{2\pi}{N} y_i \tag{23}$$

$$\mathcal{F}\mathcal{S}_2^{\pm} f\,(y_i, y_j) = e^{\pm \imath z_j}\mathcal{F}f\,(y_i, y_j) \qquad z_j = \frac{2\pi}{N} y_j. \tag{24}$$

where (y_i, y_j) are the discrete coordinates in the frequency domain. Assuming periodic boundary conditions, expression (21) in Fourier domain now equals:

$$\underbrace{\left[2r_4\left(\cos\left(z_i\right) + \cos\left(z_j\right) - 2\right) - r_1\right]}_{\mathcal{G}}\mathcal{F}\phi\left(y_i, y_j\right) = \mathcal{F}h\left(y_i, y_j\right), \tag{25}$$

and provides us with a closed-form solution $\phi^\star$ of (17):

$$\phi^\star = \mathcal{F}^{-1}(\ \mathcal{F}h/\mathcal{G}\ ) \tag{26}$$

which we can efficiently compute by FFT. It is now straightforward to apply the same procedure to compute the minimizer $u^\star$ of (18), whose Euler-Lagrange equation is:

$$(-r_3\Delta + r_2)u = -w_r - r_3 \operatorname{div} \boldsymbol{q} + \frac{\boldsymbol{\lambda_3}}{r_3} + r_2\Big(H(\varphi) - \frac{\lambda_2}{r_2}\Big). \tag{27}$$

4.2 Sub-Minimization Problem w.r.t. φ

The sub-minimization problem (13) can be written as follows:

$$\min_{\varphi} \int_\Omega \frac{r_1}{2}\Big(\varphi - \Big(\phi - \frac{\lambda_1}{r_1}\Big)\Big)^2 + \frac{r_2}{2}\Big(H(\varphi) - \Big(u + \frac{\lambda_2}{r_2}\Big)\Big)^2 \tag{28}$$

Let us call $z = \phi - \frac{\lambda_1}{r_1}$ and $v = u + \frac{\lambda_2}{r_2}$ and observe that the minimization is decoupled for each pixel. Let us define the function we want to minimize as

$$F(\varphi) = \frac{r_1}{2}(\varphi - z)^2 + \frac{r_2}{2}(H_\varepsilon(\varphi) - v)^2. \tag{29}$$

Observe that for practical implementations, the minimization problem (29) involves a smooth approximation H_ε of the Heaviside function. We propose two steps to find quickly a minimizer of (29).
Step 1: Find a solution φ^0 of (29) for $\epsilon = 0$ (i.e. for the distributional/non-smooth Heaviside function). A closed-form solution exists for this problem and can be computed as follows. The first term of (29) is minimized for $\varphi^0 = z$. As the distributional Heaviside function can take only values 0 or 1, the second term of (29) is minimized for $\varphi^0 < 0$ when $v < \frac{1}{2}$ and $\varphi^0 \geq 0$ when $v \geq \frac{1}{2}$. This means that both terms are minimized for $\varphi^0 = z$ if $v < \frac{1}{2}$ and $z < 0$ or $v \geq \frac{1}{2}$ and $z \geq 0$. Otherwise we must choose to minimize the greater of these terms and set $\varphi^0 = 0$ if $E(0) < E(z)$ and $\varphi^0 = z$ otherwise.
Step 2: Find a solution φ^0 of (29) for $\epsilon > 0$ using the standard Newton's method with φ^0 as initialization. The iterative Newton's method for finding the minimizer of (29) is as follows:

$$\varphi^{m+1} = \varphi^m - \frac{F'(\varphi^m)}{F''(\varphi^m)} = \varphi^m - \frac{r_1(\varphi^m - z) + r_2(H_\varepsilon(\varphi^m) - v)\,\delta_\varepsilon(\varphi^m)}{r_1 + r_2(H_\varepsilon(\varphi^m) - v)\,\delta'_\varepsilon(\varphi^m) + r_2\delta^2_\varepsilon(\varphi^m)}, \quad \text{with } \varphi^{m=0} = \varphi_0, \tag{30}$$

where δ_ε and δ'_ε are smooth approximations of the Dirac function and its derivative. Finally, we have observed that two iterations are enough to find a good approximation of the minimizer of (28), leading to a technique almost as fast as a closed-form solution.

4.3 Sub-Minimization Problem w.r.t. $\boldsymbol{q}$

The sub-minimization problem (15) can be written as follows:

$$\min_{\boldsymbol{q}} \int_\Omega w_b|\boldsymbol{q}| + \frac{r_3}{2}\left|\boldsymbol{q} - \left(\nabla u - \frac{\boldsymbol{\lambda_3}}{r_3}\right)\right|^2 \tag{31}$$

Let us call $\boldsymbol{z} = \nabla u - \frac{\boldsymbol{\lambda_3}}{r_3}$ and observe that the minimization (31) is decoupled for each pixel. The solution $\boldsymbol{q}^\star$ of (31) is given by soft-thresholding with the shrinkage operator [10]:

$$\boldsymbol{q}^\star = \max\left\{|\boldsymbol{z}| - \frac{w_b}{r_3}, 0\right\}\frac{\boldsymbol{z}}{|\boldsymbol{z}|} \tag{32}$$

4.4 Sub-Minimization Problem w.r.t. $\boldsymbol{p}$

The sub-minimization problem (16) can be written as follows:

$$\min_{\boldsymbol{p}} \int_\Omega \frac{r_4}{2}\left|\boldsymbol{p} - \left(\nabla\phi - \frac{\boldsymbol{\lambda_4}}{r_4}\right)\right|^2 \quad \text{s.t.} \quad |\boldsymbol{p}| = 1 \tag{33}$$

Let us call $\boldsymbol{z} = \nabla\phi - \frac{\boldsymbol{\lambda_4}}{r_4}$ and observe again that the minimization (33) is decoupled for each pixel and is equivalent to the minimization of the following function on the plane $F(\boldsymbol{p}) = \frac{r_4}{2}|\boldsymbol{p} - \boldsymbol{z}|^2 = \frac{r_4}{2}(|\boldsymbol{p}|^2 - |\boldsymbol{z}|^2 - 2\boldsymbol{p}\cdot\boldsymbol{z})$.

We can introduce the constraint in $|\boldsymbol{p}| = 1$ in F by writing it $F(\boldsymbol{p}) = \frac{r_4}{2}(1 - |\boldsymbol{z}|^2 - 2\boldsymbol{p}\cdot\boldsymbol{z})$. It is obvious that the minimizer of F is of the form $\boldsymbol{p}^\star = \frac{\boldsymbol{z}}{|\boldsymbol{z}|}$, i.e. when vectors $\boldsymbol{z}$ and $\boldsymbol{p}$

have the same orientation, the scalar product $\boldsymbol{p}\cdot\boldsymbol{z}$ reaches a maximum and the constraint $|\boldsymbol{p}| = 1$ is verified. The minimizer $\boldsymbol{p}^\star$ of (33) is then given by:

$$\boldsymbol{p}^\star = \frac{r_4}{|r_4\nabla\phi - \lambda_4|}\nabla\phi - \frac{1}{|r_4\nabla\phi - \lambda_4|}\lambda_4 \tag{34}$$

5 Experiments and discussions

Sections 5.1 and 5.2 present the results in image segmentation and surface reconstruction obtained with the proposed method. Section 5.3 compares our algorithm with existing distance preserving LSMs.

5.1 Image segmentation and surface reconstruction

We apply Algorithm 2 for image segmentation by defining the edge and region terms proposed in [6, 8], which have been extended to handle both gray level and color images. Figure 1 shows the results obtained for different images from the Berkeley[1], Weizmann[2] and GrabCut[3] databases. The method behaves as expected, providing the same results as redistancing or penalty methods in terms of final segmentation, but with a considerable speed-up in time.

We also use the proposed model to successfully reconstruct several surfaces from the Stanford dataset[4]. We initialize the method with a sphere containing all the data points and recompute the region term $w_r^{in} = -\operatorname{div}\hat{\boldsymbol{n}}$ every 5 iterations as described in Section 2. Unlike [22, 39], we do not need to estimate *a priori* the surface normal at the data points, as the surface normal at the points is directly estimated from the current value of the LSF. Finally, the process is sped up with a standard multi-resolution approach, see Figure 2. The final reconstructed surfaces are shown in figures 3 and 4.

5.2 Cortex segmentation with coupled surfaces

Cortex segmentation [19] is a problem that requires the distance information to be solved successfully. Based on [18], we can develop a segmentation algorithm for the cortical grey matter with two active contours coupled by their relative distance, which constrains the thickness of the cerebral cortex. Graph cut methods and convex relaxation methods cannot be directly applied to solve this problem because they use the binary function for representing the contour and binary functions do not hold the distance information.

The cerebral cortex is the layer of the brain bounded by the outer and inner cortical surfaces, that is, the outer interface between cerebral spinal fluid (CSF) and grey matter and the inner interface between grey and white matter. Locating this cortical surface is a first step in many brain imaging processes and measuring its thickness is a common procedure in the diagnosis of many neurological diseases. We will see that the use of SDF in the segmentation of cortical surface allows us to include information about the cortical

[1] http://www.eecs.berkeley.edu/vision

[2] http://www.wisdom.weizmann.ac.il/ vision

[3] http://research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm

[4] http://graphics.stanford.edu/data/3Dscanrep/

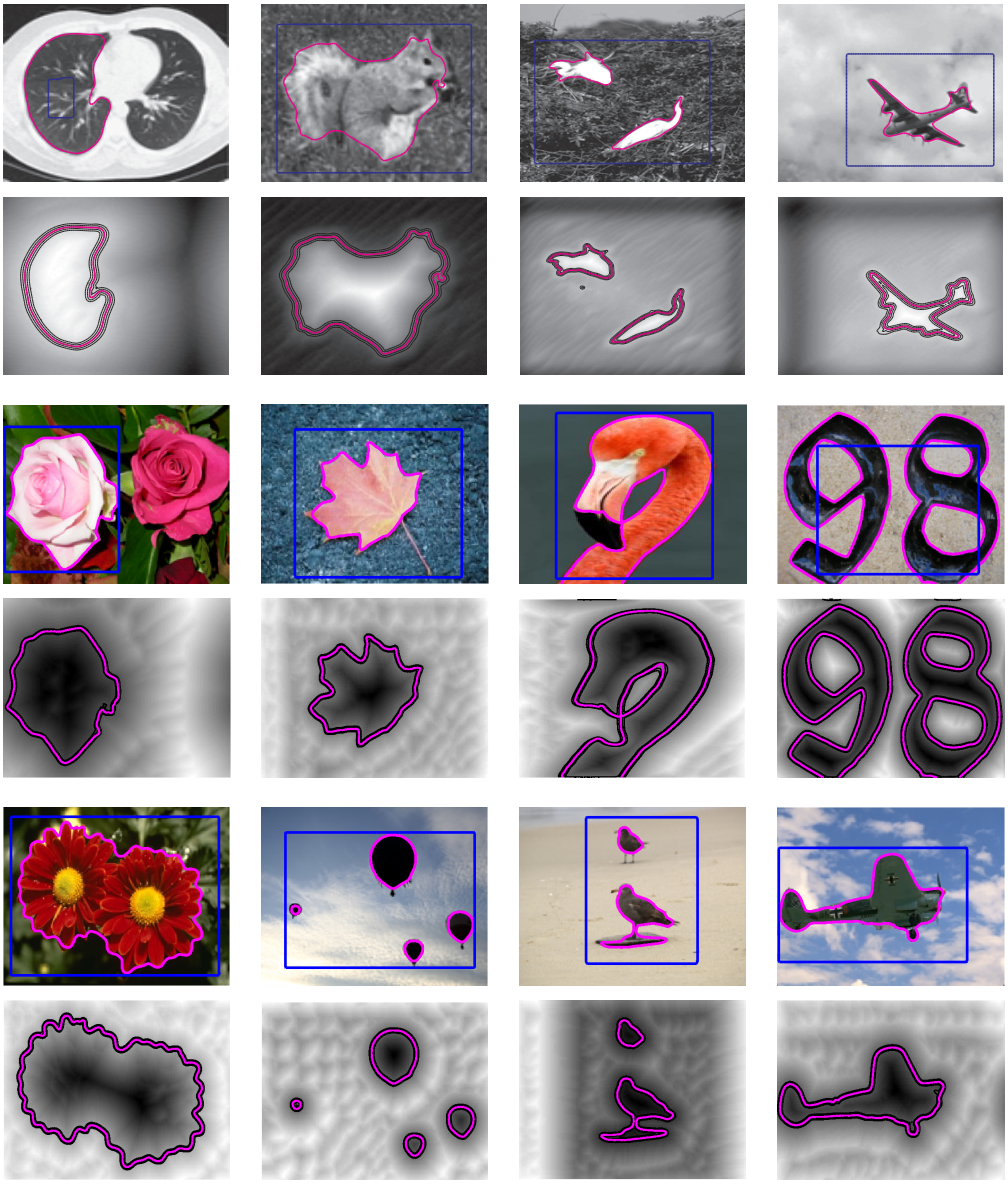
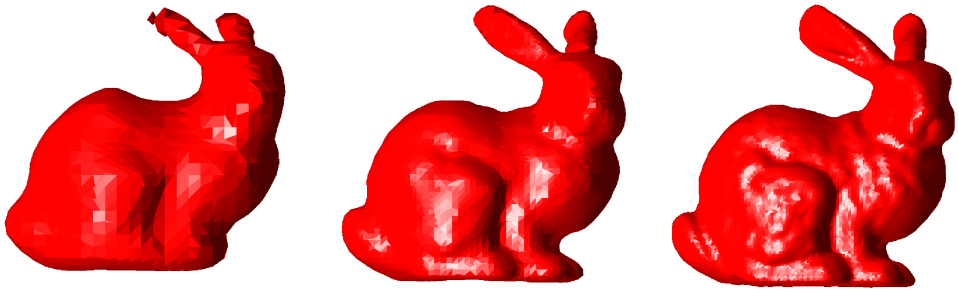


Figure 1: Proposed level set-based segmentation method applied to natural images. For each result, we show the segmentation result and the level set function. We plot the initial zero level set in blue, the final contour in pink and the $\pm 1, \pm 2$ level sets of the final function in black.



(a) grid 30x30x30, 2.4s (b) grid 60x60x60, 10.6s (c) grid 120x120x120, 50.5s

Figure 2: Reconstructed bunny at in the multiresolution approach. Linear interpolation of the SDF obtained at lower resolutions was used as initial LSF for higher resolutions.

structure into the segmentation problem and, at the same time, provides an estimate of the cortical thickness.

In order to extract the cortical layer we need to extract its two bounding surfaces C^1 and C^2. In theory, therefore, we could simply detect each of its bounding surfaces or segment the regions defined by the white matter and the exterior parts of the brain by independent minimization of the following functionals associated to C^1 and C^2

$$\min_{C^1} \mathcal{E}\left(w_b, w_{r1}, C^1\right) = \int_{C^1} w_b(s)ds + \int_{C^1_{in}} w_{r1}(x)dx, \tag{35}$$

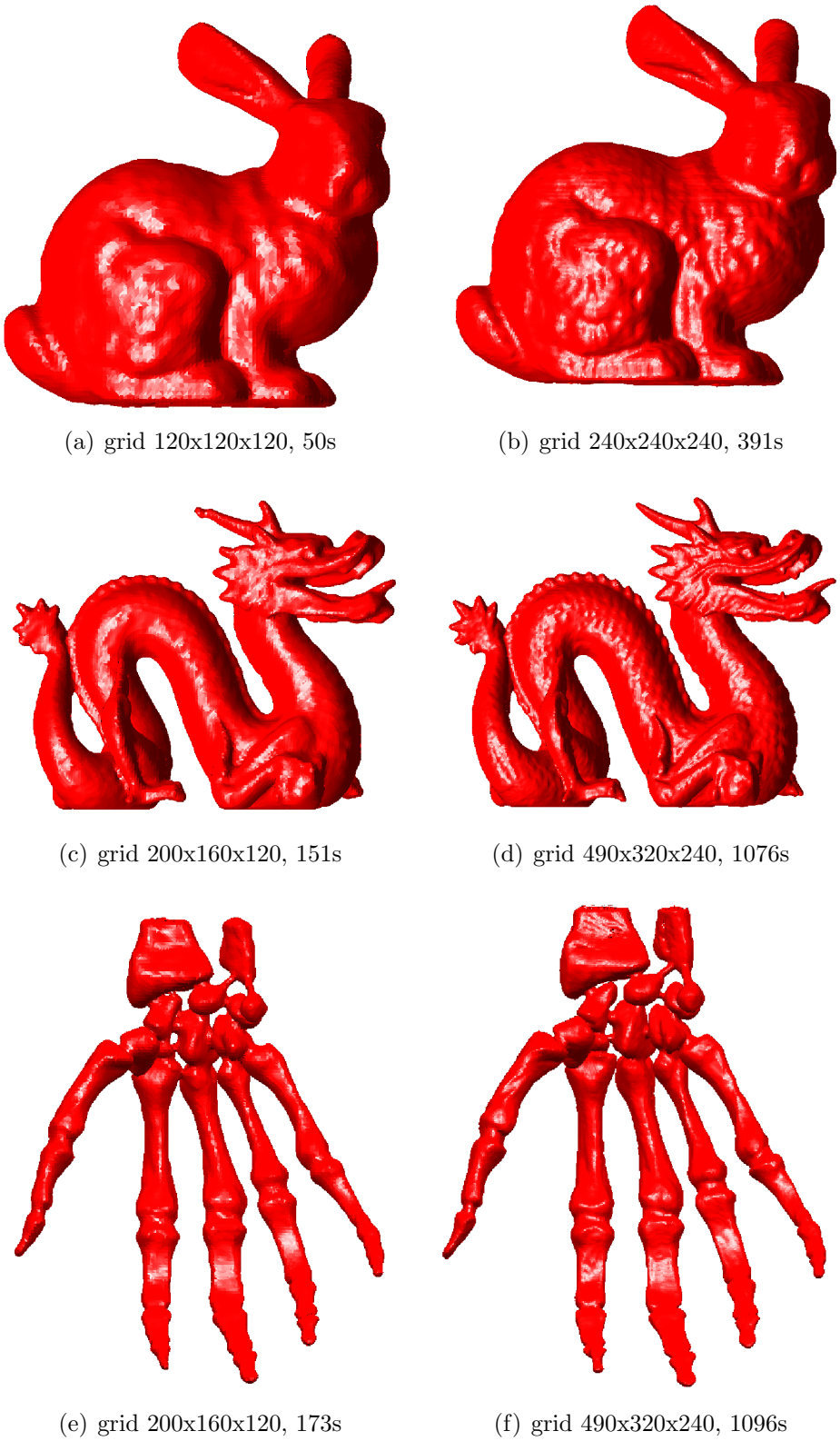
$$\min_{C^2} \mathcal{E}\left(w_b, w_{r2}, C^2\right) = \int_{C^1} w_b(s)ds + \int_{C^2_{in}} w_{r2}(x)dx, \tag{36}$$

where w_b is a boundary indicator and w_{r1}, w_{r2} are region descriptors for the white matter and the exterior areas of the brain. In practice, however, the boundaries between grey and white matter are not clear, MRI images suffer from intensity nonuniformity and the segmentations obtained with local region descriptors and boundary detectors do not correctly locate the cortical layer.

To overcome these limitations, it has been proposed to incorporate a constraint on the cortical structure to obtain a better segmentation. For simplicity, we modify the model proposed in [40] to suit our variational formulation, but similar models for cortex segmentation s.a. [28, 14] could also be adopted. We use a coupled surface model, where a functional is minimized when C^1 captures the CSF-grey matter interface, C^2 the grey-white matter boundary and the distance between them is close to the expected cortical thickness d (about 3mm). The problem is written in terms of LSM, making use of the SDF ϕ_1 and ϕ_2 to define the bounding surfaces C^1 and C^1. The functional to minimize is then given by

$$\min_{\phi_1,\phi_2} \mathcal{E}\left(w_b, w_{r1}, \phi_1\right) + \mathcal{E}\left(w_b, w_{r2}, \phi_2\right) + \frac{c}{2}\int_\Omega \left(\phi_1 - \phi_2 - d\right)^2 \quad \text{s.t.} \quad \begin{cases} |\nabla\phi_1|=1 \\ |\nabla\phi_2|=1. \end{cases} \tag{37}$$

The term $(\phi_1 - \phi_2 - d)^2$ penalizes segmentations where the distance between the bounding surfaces differs from the expected cortical thickness d. Indeed, when ϕ_1 and ϕ_2 are the SDF defined by C^1 and C^2, the distance between the surfaces can be measured on the



(a) grid 120x120x120, 50s

(b) grid 240x240x240, 391s

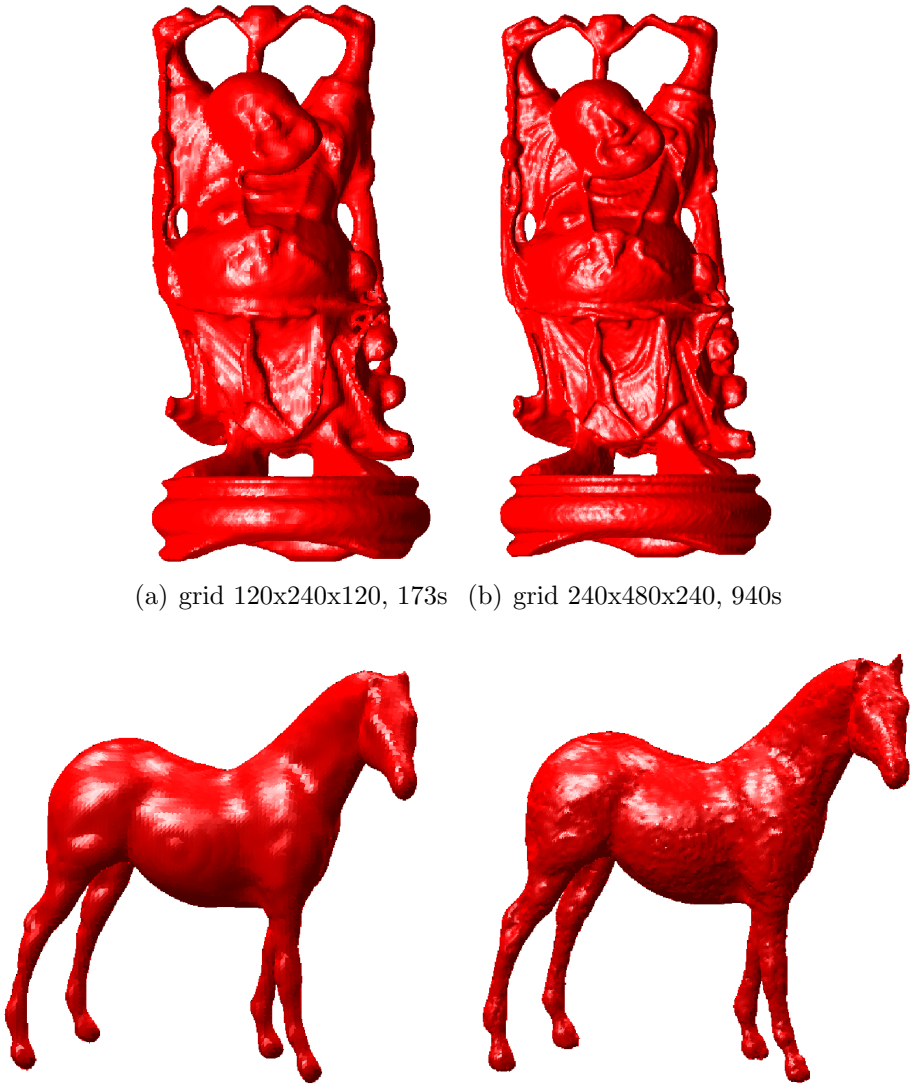
(c) grid 200x160x120, 151s

(d) grid 490x320x240, 1076s

(e) grid 200x160x120, 173s

(f) grid 490x320x240, 1096s

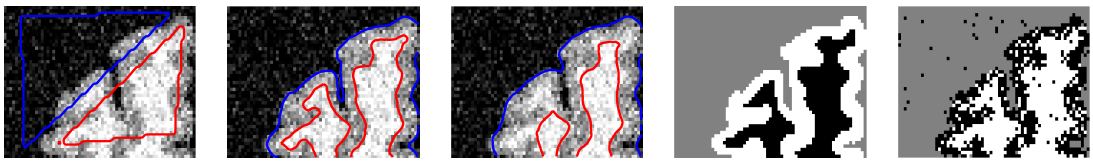
Figure 3: Reconstructed surfaces from scattered data points at different resolutions.



(a) grid 120x240x120, 173s (b) grid 240x480x240, 940s

(c) grid 60x60x120, 46s (d) grid 120x120x240, 315s

Figure 4: Reconstructed surfaces from scattered data points at different resolutions.



(a) Initialization. (b) Coupled level set functions. (c) Non coupled level set functions. (d) Ground truth. (e) k-mean algorithm.

Figure 5: Segmentation of grey-white matter interface on MRI images of human cortex. The segmentation obtained with the proposed method 5(b) is clearly closer to the ground truth 5(d) than the results obtained when no coupling terms is considered and the segmentation is performed independently for the inner and outer cortical surfaces 5(c). The segmentation 5(e) obtained with 3 phases of the k-mean algorithm fails because region descriptors alone cannot segment grey-matter.

whole domain by $\phi_1 - \phi_2$ and, consequently, the term $(\phi_1 - \phi_2 - d)^2$ drives the segmentation to solutions where the distance between the surfaces is consistent with the cortical structure.

The minimization technique presented in Section 4 can be directly applied to this problem. The same splitting variables and Lagrange multipliers are now defined and solved for each level set function and only the alternate minimization w.r.t ϕ_1 and ϕ_2 are modified. The minimization problems w.r.t ϕ_1 (the analogous applies to ϕ_2) is now

$$\min_{\phi_1} \int_\Omega \frac{r_1}{2}\Big(\phi_1 - \Big(\varphi_1 + \frac{\lambda_{1,1}}{r_1}\Big)\Big)^2 + \frac{c}{2}\left(\phi_1 - \phi_2 - d\right)^2 + \frac{r_4}{2}\Big|\nabla\phi_1 - \Big(\boldsymbol{p}_1 + \frac{\boldsymbol{\lambda_{1,4}}}{r_4}\Big)\Big|^2, \quad (38)$$

and its associated Euler-Lagrange equation

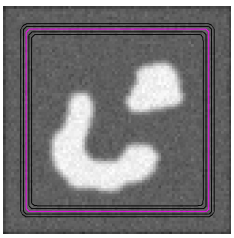
$$(-r_4\Delta + r_1 + c)\phi = -r_4 \operatorname{div} \boldsymbol{p} + \frac{\boldsymbol{\lambda_4}}{r_4} + r_1\Big(\varphi + \frac{\lambda_1}{r_1}\Big) + c\left(\phi_2 + d\right) \quad (39)$$

can be solved efficiently by the fast Fourier transform as we have already explained in Section 4.

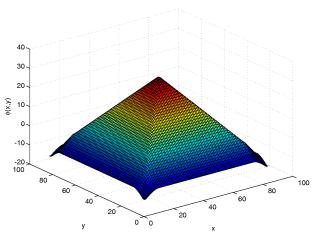
We have applied this technique in an illustrative experiment to segment the cortical layer in different slices of MRI images. Results are shown in Figure 5. We observe that coupling the level set functions and constraining the expected cortical thickness in the segmentation procedure produces a segmentation result (Fig. 5(b)) close to the ground truth (Fig. 5(d)). The segmentation obtained without coupling (Fig. 5(c)) is not able to find the fine structures.

5.3 Comparison to other distance preserving level set methods

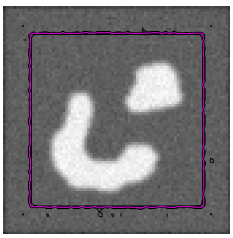
In this section, we compare our proposed algorithm to the other techniques designed to preserve the SDF in the LSM. To that purpose, we first remind the necessity of preserving the LSF as an SDF in the proposed segmentation task with a simple example. We initialize the LSF as an SDF and evolve it with the PDE techniques associated to the LSM for the image segmentation problem (4). The level set function becomes too steep around its zero level set after a few iterations, and stops its evolution because the



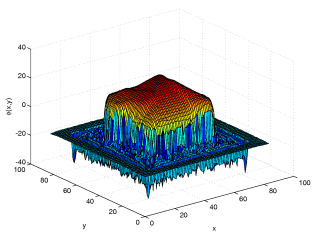
(a) Initial contour defined by level set function.



(b) Initial level set function.



(c) Final contour defined by level set function.



(d) Final level set function.

Figure 6: The level set function ($\phi = 0$ in pink and $\phi = \pm 1, \pm 2$ in dark) develops irregularities during the propagation and stops after a few iterations if the LSF is not constrained to be (or at least close to) an SDF.

geometric features (i.e. curvature and normal) are not correctly estimated, see Figure 6. Two common approaches have been introduced to overcome this problem, either by re-distancing the LSF or by maintaining the SDF during the contour evolution (as proposed in our method).

We will compare three different approaches in the case of image segmentation: our method, the standard re-distancing approach and the method of Li *et al.* [24]. The re-distancing process is carried out with the fast marching method [1], while the method of Li *et al.* is defined by introducing a penalty term in the energy to constrain the LSF to be close to an SDF as follows:

$$\int_{\Omega} w_b |\nabla H(\phi)| + w_r H(\phi) + \frac{\mu}{2}(|\nabla \phi| - 1)^2. \tag{40}$$

We refer the reader to [24, 25] for more details.

Our algorithm is presented on Figures 7(c)-7(g). Figures 7(h)-7(l) show the re-distancing method [1]. Although the final segmentation and the final LSF provide the desired results, the periodic re-initialization process produces a non-smooth minimization of the level set energy (observe the jumps in the energy plots). Besides, we remind that we do not know in general when to re-initialize the LSF as an SDF. In our experiments, we applied the re-initialization every 5 iterations. We also remind that the re-distancing process is not guaranteed to exactly preserve the location of the zero level set representing the moving interface.

Next, we will consider the method of Li *et al.* [24], which is more related to our method. Li *et al.* introduced the nice idea to constrain the LSF to be close to an SDF via a penalty term. However, the penalty term does not constrain exactly the LSF to be an SDF. Besides, the value of the penalty constant μ is a trade-off between speed and accuracy. A small penalty value μ in (40) provides a fast algorithm but does not preserve faithfully an SDF, leading to an LSF with small instabilities, while a large penalty value μ provides a better SDF but slows down significantly the minimization process as the number of iterations to reach the convergence state increases considerably. Figures 7(m)-7(q) present the results with a small value μ and Figures 7(r)-7(v) show the results with a large value μ.

Our proposed method overcomes the limitations of Li *et al.*, as our formulation constrains the LSF to be an SDF, and the proposed algorithm is fast because there is no need to assign a large penalty constant to the penalty term. Indeed, Figures 7(q), 7(v) and 7(g) show that our method keeps more faithfully the LSF as an SDF because the penalty energy is lower, by at least one order of magnitude (the minimum of the penalty energy is around 1 with our method, around 5 with Li *et al.*'s method with a large μ, and around 100 with Li *et al.*'s method with a small μ). Our method is almost as accurate as redistancing, when it comes to preserving the SDF, and overall faster than redistancing and Li's method, as shown in the next experiment with more detail. These advantages are provided by the augmented Lagrangian approach, which can preserve accurately the constraint while keeping a good minimization speed.

To quantify the improvement obtained with our method with respect to the other distance preserving LSM techniques, we have used the previous algorithms to segment 72 images from the Berkeley, Weizmann and GrabCut databases. We compare on Figure 8 the different algorithms in terms of the quality of preserving the distance function and speed. Figure 8(a) presents the values of the penalty terms at convergence for the 72 segmented images, showing that our method preserves the SDF almost as well as the redistancing method and clearly better than Li *et al.*'s method. Indeed, the penalty values for our method and the redistancing method are similar, while being an order of magnitude smaller than Li *et al.*'s method. We also compare the time for each method to converge, which is assumed when $\frac{||\phi-\phi^*||_2}{||\phi^*||_2} < \epsilon$. We observe in Figure 8(b) and Figure 8(c) that our algorithm is on average 5 to 6 times faster than the redistancing's or Li's methods for all images. Finally, we present on Figure 8(d) a scatter plot of the penalty obtained at convergence against the time required to converge, which shows that our method presents a good trade-off between accuracy and speed.

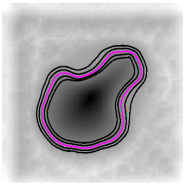
6 Conclusion

We have introduced an efficient algorithm for distance preserving level set methods which overcomes the main numerical limitations of the original level set method, i.e. the speed and the preservation of the distance function. Although other fast optimization techniques have been developed for the level set method [15, 7, 5, 16, 3], they cannot preserve the level set function as a distance function, which is essential in some applications like surface reconstruction, medical image segmentation or segmentation with higher order geometric features [27, 33]. Finally, observe that the proposed algorithm can be sped up significantly using a narrow-band approach and a parallelized version of the algorithm on graphics processing units (GPUs).

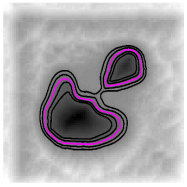
Acknowledgment

We would like to thank Dr. Meritxell Bach Cuadra for kindly providing the MRI images, and Dr. Li and his co-authors for making their code available. Virginia Estellers is supported by the Swiss SNF Grant 200021_130152. Dominique Zosso is supported by the National Competence Center in Biomedical Imaging (NCCBI). Rongjie Lai's work is supported by Zumberge Individual Award from USC's James H. Zumberge Faculty Research

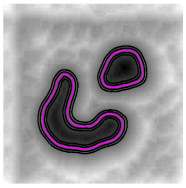
(a) Image. (b) Initial ϕ.



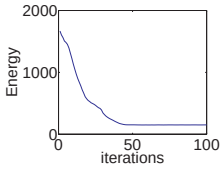
(c) *Our method*: ϕ after 15 iterations, 0.12s.



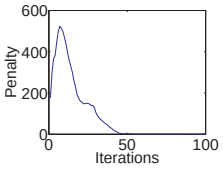
(d) *Our method*: ϕ after 19 iterations, 0.14s.



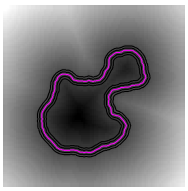
(e) *Our method*: Final ϕ after 30 iterations, 0.21s.



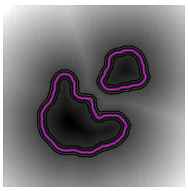
(f) *Our method.* : Energy evolution w.r.t. iterations.



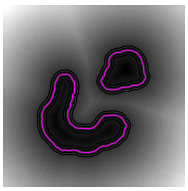
(g) *Our method*: Penalty energy w.r.t. iterations.



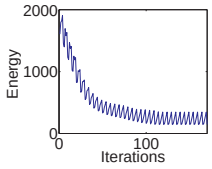
(h) *Redistancing [1]*: ϕ after 65 iterations, 1.04s.



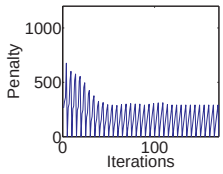
(i) *Redistancing [1]*: ϕ after 85 iterations, 1.35s.



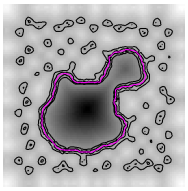
(j) *Redistancing [1]*: Final ϕ after 120 iterations, 1.84s.



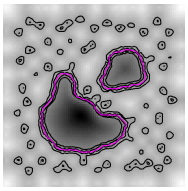
(k) *Redistancing [1]*: Energy evolution w.r.t. iterations.



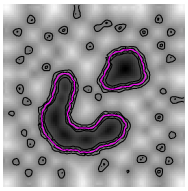
(l) *Redistancing [1]*: Penalty energy w.r.t. iterations.



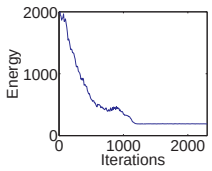
(m) *Li* et al.*'s algorithm [24]* (**small** *penalty)*: ϕ after 750 iterations, 0.85s.



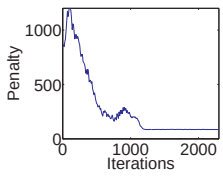
(n) *Li* et al.*'s algorithm [24]* (**small** *penalty)*: ϕ after 950 iterations, 1.35s.



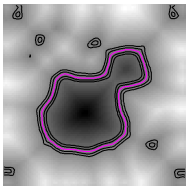
(o) *Li* et al.*'s algorithm [24]* (**small** *penalty)*: Final ϕ after 1500 iterations, 1.84s.



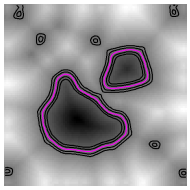
(p) *Li* et al.*'s algorithm [24]* (**small** *penalty)*: Energy evolution w.r.t. iterations.



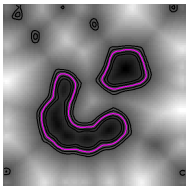
(q) *Li* et al.*'s algorithm [24]* (**small** *penalty)*: Penalty energy w.r.t. iterations.



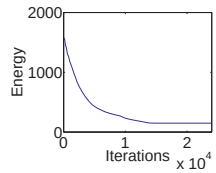
(r) *Li* et al.*'s algorithm [24]* (**large** *penalty)*: ϕ after 8,000 iterations, 8.98s.



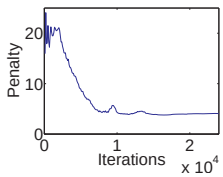
(s) *Li* et al.*'s algorithm [24]* (**large** *penalty)*: ϕ after 10,200 iterations, 11.37s.



(t) *Li* et al.*'s algorithm [24]* (**large** *penalty)*: Final ϕ after 20,000 iterations, 22.02s.

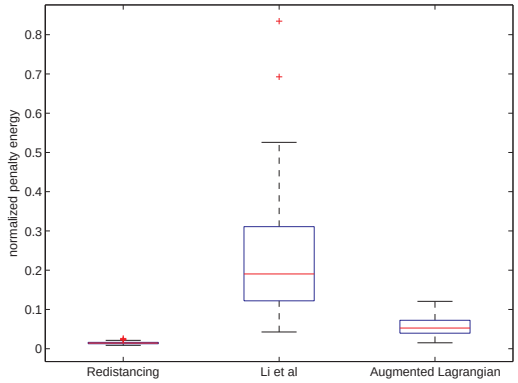


(u) *Li* et al.*'s algorithm [24]* (**large** *penalty)*: Energy evolution w.r.t. iterations.

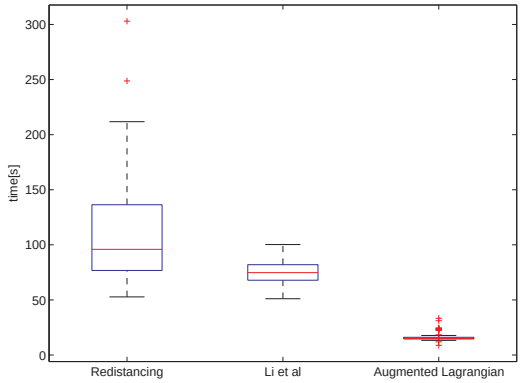


(v) *Li* et al.*'s algorithm [24]* (**large** *penalty)*: Penalty energy w.r.t. iterations.

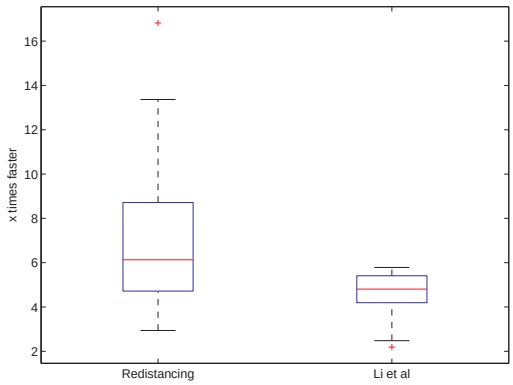
Figure 7: Level set method with our algorithm, with the re-distancing procedure [1], and with Li *et al.*'s algorithm [24]. The first three columns represent the evolution of the LSF at different times (the zero level set $\phi = 0$ is in magenta and the iso level sets $\phi = \pm 1, \pm 2$ are in dark). The fourth column plots the energy $\int_\Omega w_b |\nabla H(\phi)| + w_r H(\phi) + \frac{1}{2}(|\nabla \phi| - 1)^2$ and the last column shows the evolution of the penalty energy $\int (|\nabla \phi| - 1)^2$ (closeness measure between LSF and SDF) w.r.t. iterations. Our method provides the best trade-off between speed and preservation of the distance function.



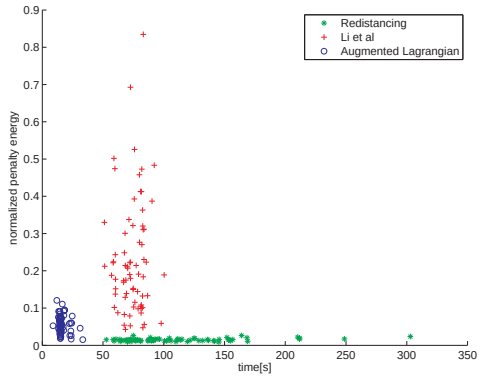
(a) LSF quality for the different methods



(b) Segmentation time for the different methods



(c) Speed improvement with our method



(d) Quality against speed for the three LSM

Figure 8: Comparison of quality of segmentation and speed for the different methods on a dataset of 72 images. Quality of the LSF is measured in terms of the penalty term at convergence $\frac{1}{|\Omega|}\int(|\nabla\phi|-1)^2$ when the obtained contours are equivalent. Our method preserves the SDF almost as well as redistancing, clearly better than Li *et al.*'s method and is faster than any of them.

and Innovation Fund. Stanley Osher is supported by USC/U.S. Army Grant 071413, ONR Grant N00014-11-1-0719, and ONR Grant N00014-08-1-1119. Xavier Bresson is supported by the Hong Kong RGC under Grant GRF110311.

References

[1] D. Adalsteinsson and J. Sethian. The Fast Construction of Extension Velocities in Level Set Methods. *Journal of Computational Physics*, 148(1):2–22, 1999. 3, 16, 18

[2] J.M. Bioucas-Dias and M.A. Figueiredo. A New TwIST: Two-Step Iterative Shrinkage/Thresholding Algorithms for Image Restoration. *IEEE Transactions on Image Processing*, 16(12):2992–3004, 2007. 3

[3] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004. 1, 2, 17

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001. 1, 2

[5] X. Bresson, S. Esedoglu, P. Vandergheynst, J. Thiran, and S. Osher. Fast Global Minimization of the Active Contour/Snake Models. *Journal of Mathematical Imaging and Vision*, 28(2):151–167, 2007. 1, 2, 17

[6] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International journal of computer vision*, 22(1):61–79, 1997. 3, 4, 10

[7] T.F. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66:1632, 2006. 1, 2, 17

[8] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001. 3, 4, 10

[9] R. Courant, K. Friedrichs, and H. Lewy. On the Partial Difference Equations of Mathematical Physics. *IBM Journal*, pages 215–234, 1967. 2

[10] D.L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 2002. 9

[11] O. Faugeras and R. Keriven. Level set methods and the stereo problem. *Scale-Space Theory in Computer Vision*, pages 272–283, 1997. 3

[12] R. Glowinski and A. Marrocco. *Sur l'Approximation, par Eléments Finis d'Ordre Un, et la Résolution, par Pénalisation-Dualité, d'une Classe de Problèmes de Dirichlet Non Linéaires*. Laboria, 1975. 3

[13] R. Glowinski and P. Le Tallec. *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*. SIAM, 1989. 3, 6, 7

[14] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Cortex segmentation: A fast variational geometric approach. *IEEE Transactions on Medical Imaging*, 21(12):1544–1551, 2002. 12

[15] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Fast geodesic active contours. *IEEE Transactions on Image Processing*, 10(10):1467–1475, 2002. 17

[16] T. Goldstein, X. Bresson, and S. Osher. Geometric Applications of the Split Bregman Method: Segmentation and Surface Reconstruction. *Journal of Scientific Computing*, 45(1-3):272–293, 2009. 1, 2, 17

[17] T. Goldstein and S. Osher. The Split Bregman Method for L1-Regularized Problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009. 3, 5

[18] J. Gomes and O. Faugeras. Reconciling Distance Functions and Level Sets. In *Scale-Space Theories in Computer Vision*, pages 70–81, 1999. 2, 3, 10

[19] X. Han, C. Xu, and J.L. Prince. A topology preserving level set method for geometric deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 755–768, 2003. 10

[20] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 26:71–71, 1992. 4

[21] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient Flows and Geometric Active Contour Models. In *International Conference on Computer Vision*, pages 810–815, 1995. 3, 4

[22] V. Lempitsky and Y. Boykov. Global optimization for shape fitting. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 4, 10

[23] M. Leventon, W.E. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 316–323, 2000. 3

[24] C. Li, C. Xu, C. Gui, and M.D. Fox. Level Set Evolution without Re-Initialization: A New Variational Formulation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 430–436, 2005. 3, 16, 18

[25] C. Li, C. Xu, C. Gui, and M.D. Fox. Distance regularized level set evolution and its application to image segmentation. *IEEE Transactions on Image Processing*, 19(12):3243–3254, 2010. 3, 16

[26] P.-L. Lions and B. Mercier. Splitting Algorithms For The Sum of Two Nonlinear Operators. *SIAM J. Numer. Anal.*, 16(6):964–979, 1979. 3

[27] L.M. Lorigo, W.E.L. Grimson, O. Faugeras, R. Keriven, R. Kikinis, A. Nabavi, and C.F. Westin. Codimension-two geodesic active contours for the segmentation of tubular structures. In *IEEE Conference on Computer Vision and Pattern Recognition*, page 1444, 2000. 17

[28] D. MacDonald, N. Kabani, D. Avis, and A.C. Evans. Automated 3-d extraction of inner and outer surfaces of cerebral cortex from mri. *NeuroImage*, 12(3):340–356, 2000. 12

[29] S. Osher and R.P. Fedkiw. *Level set methods and dynamic implicit surfaces*. Springer Verlag, 2003. 2, 3

[30] S. Osher and N. Paragios. *Geometric Level Set Methods in Imaging, Vision, and Graphics*. Springer-Verlag New York Inc, 2003. 2

[31] S. Osher and J.A. Sethian. Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *Journal of Computational Physics*, 79(1)(12-49), 1988. 1, 2

[32] N. Paraagios and R. Deriche. Geodesic active contours for supervised texture segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2002. 3

[33] M. Rochery, I. Jermyn, and J. Zerubia. Higher Order Active Contours. *International Journal of Computer Vision*, 69(1):27–42, 2006. 17

[34] P. Savadjiev, F.P. Ferrie, and K. Siddiqi. Surface Recovery from 3D Point Data Using a Combined Parametric and Geometric Flow Approach. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 325–340, 2003. 4

[35] S. Setzer. Operator Splittings, Bregman Methods and Frame Shrinkage in Image Processing. *International Journal of Computer Vision*, 92(3):265–280, 2011. 3

[36] M. Sussman, P. Smereka, and S. Osher. A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow. *Journal of Computational Physics*, 114(1):146–159, 1994. 3, 5

[37] X.-C. Tai, J. Hahn, and G.J. Chung. A Fast Algorithm for Euler's Elastica Model Using Augmented Lagrangian Method. *UCLA CAM Report 10-47*, 2010. 8

[38] Y. Wang, J. Yang, W. Yin, and Y. Zhang. A New Alternating Minimization Algorithm for Total Variation Image Reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008. 5, 8

[39] J. Ye, X. Bresson, T. Goldstein, and S. Osher. Fast Variational Methods for 3D Surface Reconstruction From a Set of Unorganized Points. *In Preparation*, 2009. 4, 10

[40] X. Zeng, L.H. Staib, R.T. Schultz, and J.S. Duncan. Segmentation and measurement of the cortex from 3-d mr images using coupled-surfaces propagation. *IEEE Transactions on Medical Imaging*, 18(10):927–937, 1999. 12

[41] H.K. Zhao, S. Osher, B. Merriman, and M. Kang. Implicit and Nonparametric Shape Reconstruction from Unorganized Data using a Variational Level Set Method. *Computer Vision and Image Understanding*, 80(3):295–314, 2000. 3, 4