

A Splitting Algorithm for Image Segmentation on Manifolds Represented by the Grid based Particle Method

Jun Liu · Shingyu Leung

Abstract We propose a numerical approach to solve variational problems on manifolds represented by the Grid Based Particle Method (GBPM) recently developed in [20,19,21,18]. In particular, we propose a splitting algorithm for image segmentation on manifolds represented by unconnected sampling particles. To develop a fast minimization algorithm, we propose a new splitting method by generalizing the augmented Lagrangian method (ALM). To efficiently implement the resulting method, we incorporate with the local polynomial approximations of the manifold in the GBPM. The resulting method is flexible for segmentation on various manifolds including closed or open or even surfaces which are not orientable.

Keywords Image Segmentation · Manifolds · Convex Relaxation · Operator Splitting · Local Reconstruction · Eulerian Mesh · Lagrangian Sampling

1 Introduction

Image segmentation aims to partition a given image into several non-overlapping domains based on statistical similarities such as means, distributions and structure tensors. In the past several decades, many methods have been proposed to address this issue including the level set methods, expectation maximization (EM) methods, and graph cut methods *etc.* The key difficulty of the image segmentation task is to provide a stable and fast algorithm for working well with low quality images. For example, the images may contain heavy noise and weak edges. To get a robust segmentation result under noise, some smoothness constraints of the partitioned domains have to be imposed in the segmentation cost functional.

For 2D plane and 3D volume images, segmentation can be achieved by minimizing the total variation (TV) of the characteristic function for the segmented domains since the TV of a characteristic function equals to the boundary length of an area according to the coarea formula. The Mumford-Shah segmentation model [28] and its variants [6,24,1,8] are examples of high efficient TV regularization models in image segmentation. However, since the space of the characteristic functions is not convex, segmentation results obtained by TV-based level set methods such as the popular Chan-Vese model [6] depend heavily on the choice of the initial guess. To obtain a global minimization in the image segmentation when the means of the regional intensity are known, [3] has proposed to *convexify* the model by relaxing the characteristic function from $\{0,1\}$ to the interval $[0,1]$. Combining a recently developed TV minimization methods [13,38,32], a stable and fast segmentation algorithm for 2D plane and 3D volume images has been introduced in [12,1,25].

However, the above algorithms have been originally for 2D plane or 3D volume images where given data are defined over a closed subset of Euclidean space. It could be tricky to generalize these techniques to images on manifolds depending on the way how the surface is represented.

Jun Liu
School of Mathematical Sciences, Laboratory of Mathematics and Complex Systems, Beijing Normal University, Beijing 100875, P.R. China
E-mail: jliu@bnu.edu.cn

Shingyu Leung
Department of Mathematics, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong.
Tel.: +852-23587414
Fax.: +852-23581643
E-mail: masyleung@ust.hk

One choice to represent the manifold is an explicit formulation where the surface is represented by triangulation. A recent work [9] extended the convex segmentation model [3] to manifolds using Delaunay triangulation. It directly solves the degraded nonlinear PDE arises from the segmentation cost functional on a triangulated mesh. Although it can handle open manifolds, the implementation is slow since the TV defined on the triangulated manifolds is non-smooth and its associated PDE has a singularity (See for example [31, 5, 27, 9, 39]).

It worths also to point out a recent paper [17] which works directly on a given point cloud data representation. The work proposes a local triangulation method for solving PDE's and computing integrals. It is therefore possible to combine with our proposed segmentation algorithm. However, the solution may depend heavily on the quality of the given data and so we do not further explore this method.

Another class of formulations based on Eulerian representation of the manifold where the surface is represented implicitly by embedding the manifold into a higher dimensional space using the level set method [29]. For example, a geodesic curvature flow model and an active contour model have been proposed in [15] and [16], respectively. Based on the intrinsic gradient and the work in [7], the Chan-Vese model [6] has been extended to manifolds in [34] by the closest point method [26]. There are advantages of using such implicit approach. For example, it can easily handle topology changes if the interface evolves as in image registration. Also, numerical implementation on the underlying uniform mesh is usually easier than working on the explicit triangulated surface mesh. However, these implicit representations require to solve PDE not only on the manifold itself, but also have to be in a neighborhood of it. This introduces extra computational requirements on the algorithm. A more challenging situation is to process images on open manifolds. For implicit Eulerian methods, there is no natural way to represent open curves and open surfaces since there is no distinction of interior and exterior regions. Recently, a few approaches have been proposed for dealing with open curves and surfaces based on the level set method. One approach was the work of [33] for modeling spiral crystal growth. The author used the intersection of two level set functions to represent the codimension-two boundary of the open curve or surface. The curve or surface of interest was implicitly defined as the zero level set of one signed distance function at which another one was positive, i.e. $S = \{x : \phi(x) = 0 \text{ and } \psi > 0\}$. However such approach works only with orientable manifolds. For instance, such approach will not work for surface like Möbius strip. Some other methods combine the level set method with triangulated mesh technique are proposed in [39, 40]. In [36], an implicit surface representation and domain decomposition methods (DDM) has been proposed to construct open surfaces and non-orientable surfaces using the graph cut methods. It is possible to combine the convex relaxation proposed here and DDM to segment images on open and non-orientable surfaces using the idea in [36]. However, the method still requires a triangulation of the manifold and so we do not pursue the investigation here.

In this paper, we propose to apply and extend a recently developed interface representation called the Grid Based Particle Method (GBPM) for modeling dynamic interface [20, 19, 21, 18]. In this approach, the interface is represented by meshless, i.e., no triangulation or parametrization, Lagrangian particles which are associated to an underlying uniform or an adaptive Eulerian mesh. We assume that the given Lagrangian sampling particles are the L^2 projection from grid points next to the manifold. This results in a quasi-uniform sampling of the interface. In this method, the model and the PDE solving are both directly defined on the sampling points. Since the manifold has been directly represented, thus unlike some implicit function methods, we do not need any extra steps to determinate the accurate position of the surfaces. To describe the local topology of the manifolds and to solve PDEs on these sampling locations, a local fitting method is employed to approximate the manifolds and the derivatives defined on them.

In order to obtain a stable and fast algorithm, we propose a splitting scheme for a new segmentation model. The idea is to move the singularity of the TV term in the convex model to a L^1 - L^2 minimization problem on manifolds, which can then be efficiently solved by a \mathbf{g} -shrinkage operator. The Euler-Lagrange equation from the optimality condition is the Laplace-Beltrami equation on the sampling points. It can then be solved by the GBPM formulation [18]. A similar approach has been recently proposed to solve PDEs on point cloud data [23, 22]. Our sampling particles could also be interpreted as a special case of a point cloud representation. However, our local interface approximation is different from the work of [23].

There are several contributions of the paper. First, we propose a new algorithm for image segmentation on manifolds. Based on the splitting technique, the variational functional can be easily optimized. Because of the flexible interface representation in the GBPM, our method can be applied to both open and closed manifolds. We also introduce a method to compute integrals on a surface represented by the GBPM. The idea is to first convert the GBPM to an implicit distance function representation and then the surface integral can be converted into a volume integral as in the level set method.

The rest of the paper is organized as follow. Section 2 summarizes existing methods on performing image segmentation on manifolds and introduces the GBPM representation of manifolds. Section 3 we propose a new algorithm to minimize the resulting variational functional defined on a given manifold represented by unconnected sampling points in the GBPM formulation. Section 4 contains some experimental results. Finally, concluding remarks and discussion can be found in section 5.

2 Background

2.1 Convex Segmentation Models on Manifolds

Total variation is a popular constraint in image segmentation because of its ability for removing small variations while smoothing boundaries between large segments. Many image segmentation methods have been proposed using TV regularization. For example, the well known Chan-Vese model [6] partitions an image using the piecewise constants approximation and TV regularization. However, the result from this model depends heavily on the choice of the initial guess. In order to overcome this difficulty, some interesting convex models and algorithms for K -phase image segmentation have been recently proposed in [3] (for $K = 2$) and [1, 4]. The idea is to replace the typical indicator function $\mathbf{u} \in \{0, 1\}$ by a continuum version

$$E(\mathbf{u}) = \int_{\Omega} \langle (f - \mathbf{c})^2, \mathbf{u} \rangle dx + \lambda \int_{\Omega} |\nabla \mathbf{u}| dx, \quad (1)$$

where $f : \Omega \rightarrow \mathbb{R}$ is the given image intensity function, λ is a regularization parameter, and $\mathbf{c} = (c_1, c_2, \dots, c_K)^T$ contains the mean intensity in each phase. The relaxed classification function $\mathbf{u} : \Omega \rightarrow [0, 1]^K$ is restricted on a convex set $\Delta_+ = \{\mathbf{u}(x) : \sum_{k=1}^K u_k(x) = 1, 0 \leq u_k(x) \leq 1\}$, where $u_k(x)$ is the k -th component of \mathbf{u} . Since this cost functional is strictly convex, the model has a unique global minimizer and so the segmentation result does not depend on the choice of the initial parameters.

This convex model has also been extended to manifolds recently in [9]

$$E(\mathbf{u}) = \int_{\mathcal{M}} \langle (f - \mathbf{c})^2, \mathbf{u} \rangle dM + \lambda \int_{\mathcal{M}} |\nabla_{\mathcal{M}} \mathbf{u}| dM. \quad (2)$$

Here, the functions f and \mathbf{u} are defined on a manifold \mathcal{M} instead of a 2D Euclidean domain Ω , $|\cdot|$ is the Riemannian norm, $\nabla_{\mathcal{M}}$ is the intrinsic gradient defined on \mathcal{M} and dM is the manifolds element measure.

Numerically, such approach may be difficult since the regularization term in the energy has a singularity at 0, and the Euler-Lagrangian equation from the first order optimality is nonlinear and degraded. It is, therefore, numerically challenging to directly solve the partial differential equation on the manifold.

2.2 Grid Based Particle Method for Dynamic Interfaces

For the convenience of readers, we give a brief summary of the Grid Based Particle Method (GBPM). For a complete and detailed description of the algorithm, we refer the readers to [20, 19, 21, 18].

In the GBPM, we represent the interface by meshless particles which are associated to an underlying Eulerian mesh. Each sampling particle on the interface is chosen to be the closest point from each underlying grid point in a small neighborhood of the interface. This one to one correspondence gives each particle an Eulerian reference during the evolution.

At the first step, we define an initial computational tube for active grid points and use their corresponding closest points as the sampling particles for the interface. A grid point \mathbf{y} is called **active** if its distance to the interface is smaller than a given **tube radius**, ϵ , and we label the set containing all active grids Γ . To each of these active grid points, we associate the corresponding closest point on the interface, and denote this point by \mathbf{x} . This particle is called the **foot-point** associated to this active grid point. Furthermore, we can also compute and store certain Lagrangian information of the interface at the foot-points, including normal, curvature and parametrization, which will be useful in various applications.

This representation is illustrated in figure 1 (a) using a circular manifold as an example. We plot the underlying mesh in solid line, all active grids using small circles and their associated foot-points on the manifold using squares. To each grid point near the interface (blue circles), we associate a foot-point on the interface (red squares). The relationship of each of these pairs is shown by a solid line link.

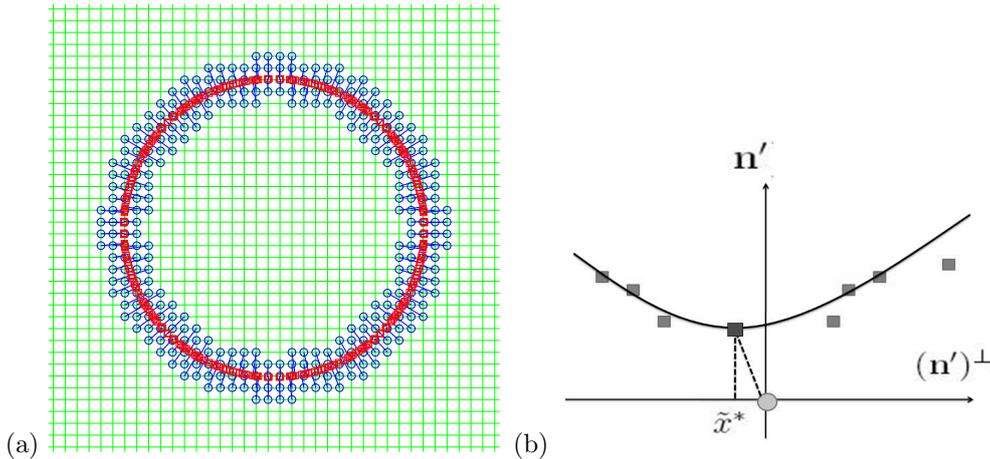


Fig. 1 (a) Grid Based Particle Method (GBPM) representation. (b) Determining the foot-point using a local least-square reconstruction of the interface.

More importantly, associated to each of these footpoints and activated grid point, we also have a least square local approximation of the interface using polynomials in a local coordinate system, $\{(\mathbf{n})^\perp, \mathbf{n}\}$ with \mathbf{x} as the origin, figure 1 (b). This will be very useful in solving partial differential equations (PDEs) on the interface [21, 18].

3 The Proposed Method

3.1 The New Splitting Methods on Manifolds

To simplify the following expressions, we only consider the 2-phase segmentation, i.e. $K = 2$. Together with the convex relaxation range Δ_+ , using the fact that $u_2 = 1 - u_1$ and rewriting u_1 with u , then the cost functional (2) becomes

$$E(u, c_1, c_2) = \int_{\mathcal{M}} (f - c_1)^2 u \, dM + \int_{\mathcal{M}} (f - c_2)^2 (1 - u) \, dM + \lambda \int_{\mathcal{M}} |\nabla_{\mathcal{M}} u| \, dM, \quad (3)$$

where $0 \leq u \leq 1$.

Instead of dealing with the energy directly, we propose to extend some recently developed operator splitting methods proposed in [37, 38, 12, 13, 32] which have been shown to be very efficient for solving L^1 or total variation (TV) minimization problems in the 2-D plane image processing. In these algorithms, the singularity of L^1 norm at 0 is transferred into a shrinkage problem together with a smooth minimization problem so that one can apply various stable and fast algorithms. Here, we employ the Augmented Lagrangian Method (ALM) (see e.g. [14]) to minimize the cost functional (3) on manifolds, which is an extension of ALM for TV [38] on 2D plane images.

In order to overcome the non-smoothness of the L^1 Riemannian norm, we have to separate the norm and the intrinsic gradient operator $\nabla_{\mathcal{M}}$. The intrinsic gradient $\nabla_{\mathcal{M}} u$ is used to describe the variation of u on a manifold \mathcal{M} . Suppose a surface (or a manifold) \mathcal{M} is embedded into the Euclidean space \mathbb{R}^3 , then $\nabla_{\mathcal{M}} u$ is an orthogonal projection of the gradient vector ∇u in the embedding space onto \mathcal{M} if u is well defined on entire \mathbb{R}^3 . Mathematically, the intrinsic gradient has been developed in the level set method framework [7]

$$\nabla_{\mathcal{M}} u = \nabla u - \frac{\langle \nabla u, \mathbf{N} \rangle}{|\mathbf{N}|^2} \mathbf{N} = \left(\mathbf{I} - \frac{\mathbf{N} \otimes \mathbf{N}}{|\mathbf{N}|^2} \right) \nabla u,$$

where \mathbf{N} is the normal vector on \mathcal{M} , \mathbf{I} stands for the 3×3 identity matrix, and the symbol \otimes represents the Kronecker tensor product. In this paper, we do not intend to directly use this formulation of intrinsic gradient since it requires the gradient of u in the entire embedding space \mathbb{R}^3 which is not always available. On the other hand, if the manifold or surface \mathcal{M} in 3D space has a parameterized expression $\mathbf{x}(s_1, s_2)$, then the intrinsic gradient also can be written as [10]

$$\nabla_{\mathcal{M}} u = \begin{pmatrix} \frac{\partial \mathbf{x}}{\partial s_1} & \frac{\partial \mathbf{x}}{\partial s_2} \end{pmatrix} \begin{pmatrix} E & F \\ F & G \end{pmatrix}^{-1} \begin{pmatrix} \frac{\partial u}{\partial s_1} \\ \frac{\partial u}{\partial s_2} \end{pmatrix}.$$

Here E, F, G are the coefficients of the first fundamental form of surface \mathcal{M} given by $E = \langle \frac{\partial \mathbf{x}}{\partial s_1}, \frac{\partial \mathbf{x}}{\partial s_1} \rangle$, $F = \langle \frac{\partial \mathbf{x}}{\partial s_1}, \frac{\partial \mathbf{x}}{\partial s_2} \rangle$, $G = \langle \frac{\partial \mathbf{x}}{\partial s_2}, \frac{\partial \mathbf{x}}{\partial s_2} \rangle$. As a result, one could obtain

$$|\nabla_{\mathcal{M}} u| = \sqrt{(\nabla_{\mathcal{M}} u)^T (\nabla_{\mathcal{M}} u)} = \sqrt{\begin{pmatrix} \frac{\partial u}{\partial s_1} & \frac{\partial u}{\partial s_2} \end{pmatrix} \begin{pmatrix} E & F \\ F & G \end{pmatrix}^{-1} \begin{pmatrix} \frac{\partial u}{\partial s_1} \\ \frac{\partial u}{\partial s_2} \end{pmatrix}} = \sqrt{(\nabla_{\mathbf{s}} u)^T \begin{pmatrix} E & F \\ F & G \end{pmatrix}^{-1} \nabla_{\mathbf{s}} u}. \quad (4)$$

Now we let

$$\mathbf{g} := \begin{pmatrix} E & F \\ F & G \end{pmatrix}^{-1}.$$

Introducing an auxiliary variable \mathbf{v} , the variational problem (3) is equivalent to the following constrained optimization problem for $\nabla_{\mathbf{s}} u$

$$\min_{u \in [0,1], \mathbf{v}, c_1, c_2} \left\{ \int_{\mathcal{M}} (f - c_1)^2 u \, dM + \int_{\mathcal{M}} (f - c_2)^2 (1 - u) \, dM + \lambda \int_{\mathcal{M}} \sqrt{\mathbf{v}^T \mathbf{g} \mathbf{v}} \, dM \right\},$$

such that $\mathbf{v} = \nabla_{\mathbf{s}} u$. (5)

Since \mathbf{g} is positive definite for regular surfaces, the condition $\mathbf{v} = \nabla_{\mathbf{s}} u$ in (5) can be replaced by $\mathbf{g}(\mathbf{v} - \nabla_{\mathbf{s}} u) = 0$. By applying a similar idea as in the ALM (see e.g. [14, 38]), the constraint problem (5) can be further rewritten as the following unconstrained Lagrangian functional for saddle point problem

$$\begin{aligned} L(u, \mathbf{v}, \mathbf{p}, c_1, c_2) &= \int_{\mathcal{M}} (f - c_1)^2 u \, dM + \int_{\mathcal{M}} (f - c_2)^2 (1 - u) \, dM + \lambda \int_{\mathcal{M}} \sqrt{\mathbf{v}^T \mathbf{g} \mathbf{v}} \, dM \\ &\quad + \int_{\mathcal{M}} \langle \mathbf{p}, \mathbf{g}(\mathbf{v} - \nabla_{\mathbf{s}} u) \rangle \, dM + \frac{\eta}{2} \int_{\mathcal{M}} (\mathbf{v} - \nabla_{\mathbf{s}} u)^T \mathbf{g} (\mathbf{v} - \nabla_{\mathbf{s}} u) \, dM, \end{aligned} \quad (6)$$

where \mathbf{p} is the Lagrangian multiplier, and η is a positive penalty parameter.

Note however that such modification in (6) is not the classical ALM which should be given by

$$\frac{\eta}{2} \int_{\mathcal{M}} (\mathbf{v} - \nabla_{\mathbf{s}} u)^T \mathbf{g} (\mathbf{v} - \nabla_{\mathbf{s}} u) \, dM$$

(such as [14, 38]). If one follows this original ALM directly, the sub-minimization problem for \mathbf{v} would have no closed-form solution and it could slow down the convergence of the overall algorithm. On the other hand, our proposed modification in (6) would lead to a nice simple sub-minimization problem for \mathbf{v} which can be solved easily using shrinkage.

To numerically get a saddle point of L , we use the following alternating minimization algorithm

$$\begin{cases} (u^{n+1}, \mathbf{v}^{n+1}, c_1^{n+1}, c_2^{n+1}) = \arg \min_{u \in [0,1], \mathbf{v}, c_1, c_2} L(u, \mathbf{v}, \mathbf{p}^n, c_1, c_2), \\ \mathbf{p}^{n+1} = \arg \max_{\mathbf{p}} L(u^{n+1}, \mathbf{v}^{n+1}, \mathbf{p}, c_1^{n+1}, c_2^{n+1}), \end{cases}$$

which can be further reduced into 4 subproblems:

$$u^{n+1} = \arg \min_{u \in [0,1]} \tilde{L}(u, \mathbf{v}^n, \mathbf{p}^n, c_1^n, c_2^n), \quad (7)$$

$$\mathbf{v}^{n+1} = \arg \min_{\mathbf{v}} \tilde{L}(u^{n+1}, \mathbf{v}, \mathbf{p}^n, c_1^n, c_2^n), \quad (8)$$

$$(c_1^{n+1}, c_2^{n+1}) = \arg \min_{c_1, c_2} \tilde{L}(u^{n+1}, \mathbf{v}^{n+1}, \mathbf{p}^n, c_1, c_2), \quad (9)$$

$$\mathbf{p}^{n+1} = \mathbf{p}^n + \frac{\eta}{\Lambda_{\max}} \mathbf{g}(\mathbf{v}^{n+1} - \nabla_{\mathbf{s}} u^{n+1}), \quad (10)$$

where Λ_{\max} is the maximum eigenvalue of matrix \mathbf{g} and \tilde{L} is given by

$$\begin{aligned} \tilde{L}(u, \mathbf{v}, \mathbf{p}, c_1, c_2) &= \int_{\mathcal{M}} (f - c_1)^2 u \, dM + \int_{\mathcal{M}} (f - c_2)^2 (1 - u) \, dM + \lambda \int_{\mathcal{M}} \sqrt{\mathbf{v}^T \mathbf{g} \mathbf{v}} \, dM \\ &\quad + \frac{\eta}{2} \int_{\mathcal{M}} (\mathbf{v} - \nabla_{\mathbf{s}} u + \frac{1}{\eta} \mathbf{p})^T \mathbf{g} (\mathbf{v} - \nabla_{\mathbf{s}} u + \frac{1}{\eta} \mathbf{p}) \, dM. \end{aligned}$$

We can prove the following convergence theorem of the above iteration scheme:

Theorem 1 For any fixed u and \mathbf{c} , the sequence $(\mathbf{v}^n, \mathbf{p}^n)$ produced by iteration (8) and (10) converges and $\mathbf{v}^n \rightarrow \nabla_s u$.

Proof 1 See Appendix A for details.

Now, we only need to solve the above three minimization problems to update variables u , v , c_1 and c_2 .

As to the subproblem (7), the associated Euler-Lagrange equation is the surface Laplace equation

$$-\frac{\eta}{\sqrt{EG-F^2}} \sum_{i=1}^2 \frac{\partial}{\partial s_i} \left(\sqrt{EG-F^2} \sum_{j=1}^2 g_{ij} \frac{\partial u}{\partial s_j} \right) + \frac{\eta}{\sqrt{EG-F^2}} \sum_{i=1}^2 \frac{\partial}{\partial s_i} \left(\sqrt{EG-F^2} \sum_{j=1}^2 g_{ij} (v_j^n + \frac{1}{\eta} p_j^n) \right) + (f - c_1)^2 - (f - c_2)^2 = 0, \quad \text{when } u \in [0, 1] \quad (11)$$

where g_{ij} is the ij -th element of the matrix \mathbf{g} , and the similar notations for v_j and p_j . The first term in the equation is the Laplace-Beltrami operator. We leave in Appendix B a detailed calculation in obtaining (11) from minimizing the subproblem (7).

The subproblem (8) is a $L^1 - L^2$ minimization problem on manifolds which can be efficiently solved by the following \mathbf{g} -shrinkage operator

$$\mathbf{v}^{n+1} = \text{shrink}_{\mathbf{g}} \left(\nabla_s u^{n+1} - \frac{\mathbf{p}^n}{\eta}, \frac{\lambda}{\eta} \right) = \frac{\nabla_s u^{n+1} - \frac{\mathbf{p}^n}{\eta}}{\|\nabla_s u^{n+1} - \frac{\mathbf{p}^n}{\eta}\|_{\mathbf{g}}} \max \left(\|\nabla_s u^{n+1} - \frac{\mathbf{p}^n}{\eta}\|_{\mathbf{g}} - \frac{\lambda}{\eta}, 0 \right), \quad (12)$$

where $\|\cdot\|_{\mathbf{g}}$ is a weighted norm with the expression $\|\mathbf{z}\|_{\mathbf{g}} = \sqrt{\mathbf{z}^T \mathbf{g} \mathbf{z}}$ for any vector \mathbf{z} . A detailed derivation of this \mathbf{g} -shrinkage operator can be found in Appendix C. Indeed, there is another splitting scheme for this subproblem by decomposing \mathbf{g} into $\tilde{\mathbf{g}}^T \tilde{\mathbf{g}}$ using, for example, the Cholesky decomposition, and introducing the condition $\mathbf{v} = \tilde{\mathbf{g}} \nabla_s u$. However, such formulation requires to differentiate $\tilde{\mathbf{g}}$ in the subproblem for u which is complicated.

Finally, the means c_1^{n+1} and c_2^{n+1} can be easily updated with

$$c_1^{n+1} = \frac{\int_{\mathcal{M}} f u^{n+1} dM}{\int_{\mathcal{M}} u^{n+1} dM}, \quad c_2^{n+1} = \frac{\int_{\mathcal{M}} f(1 - u^{n+1}) dM}{\int_{\mathcal{M}} (1 - u^{n+1}) dM}. \quad (13)$$

To summarize, we get the following ALM for image segmentation on manifolds:

Algorithm 1 Given initial guess values c_1^0, c_2^0 . We set $\mathbf{v}^0 = \mathbf{p}^0 = 0$ and iterate the following steps until a convergence criterion is reached:

Step 1: calculate u^{n+1} by solving the PDE (11);

Step 2: update \mathbf{v}^{n+1} with the \mathbf{g} -shrinkage operator (12);

Step 3: determine the mean vector \mathbf{c}^{n+1} according to (13);

Step 4: find \mathbf{p}^{n+1} using (10).

There are several complications in the above algorithms. The first is that Steps 2 and 4 require computations of the surface gradient $\nabla_s u$ on the manifold or surface which might be tricky depending on the manifold representation. The second is to solve the Laplace-Beltrami equation (11) in Step 1. The third is to compute integrals (13) in Step 3 on a general manifold. Indeed, these problems might be solved if we are given a surface triangulation. In this paper, however, we propose to apply and extend a newly developed GBPM proposed in [20, 19, 21, 18] which provides another natural way to solve these subproblems. Details will be given in the next two subsections.

To end this section, we further note that the image segmentation model is convex if the averages of the regional intensity are known. Otherwise, it is still possible to obtain a convex relaxation by following the idea in [2].

3.2 Computing Derivatives and Solving PDE on Manifolds in the GBPM Representation

In this section, we explain how to incorporate the GBPM to compute derivatives of function and to solve a PDE on manifolds. These ideas will be similar to those proposed in [21, 18], but are slightly modified to better fit the current application. We will explain the difference in details in later paragraphs.

The GBPM originally proposed in [20] is to model the interface motions. The key idea of GBPM is to sample the surface according to an underlying mesh such that each sampling particle is a L^2 projection of the grid point in a neighborhood of the interface. Then the function and its derivative defined on surface can be approximated in a local coordinate system, and thus we can solve various PDEs defined on an evolving interface represented by particles [21, 18].

Assume that we are given a uniform Cartesian mesh in \mathbb{R}^3 which can well sample the manifold \mathcal{M} . For each grid point \mathbf{y}^i near the manifold, we assume also that we are given its L^2 projection onto the surface. We call this closest point the foot point and we denote it by $\mathbf{x}^i = (x_1^i, x_2^i, x_3^i)^T$. Further, we assume the image f is defined on these foot point locations, and we denote it by $f^i = f(\mathbf{x}^i)$. These assumptions are mild comparing to other representations. For instance, we do not require the connectivity of these foot points as in the surface triangulation. Also, we do not require to specify the normal vector at these foot points since it might not be available for non-orientable surface. Equivalent to this condition, we do not require a global implicit representation as in the level set function.

Now, if the manifold \mathcal{M} is smooth, it can be well-approximated by functions such as polynomials in a local coordinate system in a small local neighborhood of each \mathbf{x}^i . A detailed description of the construction is given here:

1. Determine the local neighborhood. For each neighboring grid point \mathbf{y}^i , we search m nearest activated grid point \mathbf{y}^j and collect their associated footpoints. We denote this set $\{\mathbf{x}^j\}_{j=1}^m$.
2. Estimate the normal vector of the local coordinate system. We choose a unit direction $\mathbf{n} = (n_1, n_2, n_3)^T$ in \mathbb{R}^3 which minimizes the sum of the squared differences of all the variation between \mathbf{x}^j and \mathbf{n} . Mathematically, we minimize

$$\mathbf{n} = \arg \min_{|\tilde{\mathbf{n}}|=1} \sum_{j=1}^m (\langle \mathbf{x}^j, \tilde{\mathbf{n}} \rangle - \langle \mathbf{x}^i, \tilde{\mathbf{n}} \rangle)^2. \quad (14)$$

The minimizer of this problem can be determined by finding the eigenvector corresponding to the smallest eigenvalue of the symmetry matrix $\mathbf{H} = \sum_{j=1}^m (\mathbf{x}^j - \mathbf{x}^i) (\mathbf{x}^j - \mathbf{x}^i)^T$. A simple proof is given in Appendix D for completeness. Further, if \mathbf{x}^i is the mean of all the \mathbf{x}^j (which is not the case in the current application), (14) is equivalent to the well known principal component analysis (PCA).

3. Construct a coordinate system with the three eigenvectors $\mathbf{T}_1, \mathbf{T}_2, \mathbf{n}$ of \mathbf{H} as the x -axis, y -axis and z -axis respectively. We denote this coordinate system as $\{\mathbf{x}^i; \mathbf{T}_1, \mathbf{T}_2, \mathbf{n}\}$.

Once we obtain the local coordinate system, we can easily approximate the metric tensor of manifold and the derivatives of any function defined on this manifold. In particular, let us assume that the manifold or the surface in a local neighborhood is approximated by a degree k polynomial. For each \mathbf{x}^j , $j = 1, \dots, m$, let $\bar{\mathbf{x}}^j$ be the new coordinates of \mathbf{x}^j in the local system $\{\mathbf{x}^i; \mathbf{T}_1, \mathbf{T}_2, \mathbf{n}\}$, i.e.

$$\bar{\mathbf{x}}^j = \begin{pmatrix} \frac{n_1^2}{1+n_3} - 1 & \frac{n_1 n_2}{1+n_3} & n_1 \\ \frac{n_1 n_2}{1+n_3} & \frac{n_2^2}{1+n_3} - 1 & n_2 \\ n_1 & n_2 & n_3 \end{pmatrix} (\mathbf{x}^j - \mathbf{x}^i).$$

Now, assume the underlying manifold is locally quadratic, i.e. $\bar{x}_3(\bar{x}_1, \bar{x}_2) \approx \sum_{\tau_1=0}^2 \sum_{0 \leq \tau_1 + \tau_2 \leq 2} \alpha_{\tau_1 \tau_2} \bar{x}_1^{\tau_1} \bar{x}_2^{\tau_2}$,

in which $\alpha_{\tau_1 \tau_2}$ are some unknown coefficients. For every activated grid point \mathbf{y}^i near the manifold, these local coefficients $\alpha_{\tau_1 \tau_2}^i$ can be determined by minimizing the following least squares sum:

$$\sum_{j=1}^m \left(\bar{x}_3^j - \sum_{\tau_1=0}^2 \sum_{0 \leq \tau_1 + \tau_2 \leq 2} \alpha_{\tau_1 \tau_2}^i (\bar{x}_1^j)^{\tau_1} (\bar{x}_2^j)^{\tau_2} \right)^2.$$

Denote

$$\begin{aligned} \boldsymbol{\alpha}^i &= (\alpha_{00}^i \ \alpha_{01}^i \ \alpha_{02}^i \ \alpha_{10}^i \ \alpha_{11}^i \ \alpha_{20}^i)^T, \\ \mathbf{A} &= \begin{pmatrix} 1 & \bar{x}_2^1 & (\bar{x}_2^1)^2 & \bar{x}_1^1 & \bar{x}_1^1 \bar{x}_2^1 & (\bar{x}_1^1)^2 \\ 1 & \bar{x}_2^2 & (\bar{x}_2^2)^2 & \bar{x}_1^2 & \bar{x}_1^2 \bar{x}_2^2 & (\bar{x}_1^2)^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \bar{x}_2^m & (\bar{x}_2^m)^2 & \bar{x}_1^m & \bar{x}_1^m \bar{x}_2^m & (\bar{x}_1^m)^2 \end{pmatrix}_{m \times 6}, \\ \mathbf{b} &= (\bar{x}_3^1 \ \bar{x}_3^2 \ \dots \ \bar{x}_3^m)^T, \end{aligned}$$

then this least squares polynomial can be obtained by solving the over-determined linear system

$$\mathbf{A}\boldsymbol{\alpha}^i = \mathbf{b}$$

by the SVD or the QR decomposition.

As a result, the elements of metric tensor E^i, G^i, F^i at \mathbf{x}^i occur in equation (11) can be approximated using

$$\begin{aligned} E^i &= \left\langle \frac{\partial \mathbf{x}}{\partial s_1} \Big|_{\mathbf{x}=\mathbf{x}^i}, \frac{\partial \mathbf{x}}{\partial s_1} \Big|_{\mathbf{x}=\mathbf{x}^i} \right\rangle = \left\langle \frac{\partial \mathbf{x}}{\partial \bar{x}_1} \Big|_{\bar{\mathbf{x}}=\bar{\mathbf{x}}^i}, \frac{\partial \mathbf{x}}{\partial \bar{x}_1} \Big|_{\bar{\mathbf{x}}=\bar{\mathbf{x}}^i} \right\rangle \approx 1 + (\alpha_{10}^i + \alpha_{11}^i \bar{x}_2^i + 2\alpha_{20}^i \bar{x}_1^i)^2, \\ G^i &= \left\langle \frac{\partial \mathbf{x}}{\partial s_2} \Big|_{\mathbf{x}=\mathbf{x}^i}, \frac{\partial \mathbf{x}}{\partial s_2} \Big|_{\mathbf{x}=\mathbf{x}^i} \right\rangle = \left\langle \frac{\partial \mathbf{x}}{\partial \bar{x}_2} \Big|_{\bar{\mathbf{x}}=\bar{\mathbf{x}}^i}, \frac{\partial \mathbf{x}}{\partial \bar{x}_2} \Big|_{\bar{\mathbf{x}}=\bar{\mathbf{x}}^i} \right\rangle \approx 1 + (\alpha_{01}^i + \alpha_{11}^i \bar{x}_1^i + 2\alpha_{02}^i \bar{x}_2^i)^2, \\ F^i &= \left\langle \frac{\partial \mathbf{x}}{\partial s_1} \Big|_{\mathbf{x}=\mathbf{x}^i}, \frac{\partial \mathbf{x}}{\partial s_2} \Big|_{\mathbf{x}=\mathbf{x}^i} \right\rangle = \left\langle \frac{\partial \mathbf{x}}{\partial \bar{x}_1} \Big|_{\bar{\mathbf{x}}=\bar{\mathbf{x}}^i}, \frac{\partial \mathbf{x}}{\partial \bar{x}_2} \Big|_{\bar{\mathbf{x}}=\bar{\mathbf{x}}^i} \right\rangle \approx (\alpha_{10}^i + \alpha_{11}^i \bar{x}_2^i + 2\alpha_{20}^i \bar{x}_1^i)(\alpha_{01}^i + \alpha_{11}^i \bar{x}_1^i + 2\alpha_{02}^i \bar{x}_2^i). \end{aligned}$$

Similarly, the relaxed classification function u defined on the manifold can be approximated using least squares approximation [21, 18]

$$u(\bar{\mathbf{x}}) \approx \sum_{\tau_1=0}^2 \sum_{0 \leq \tau_1 + \tau_2 \leq 2} \beta_{\tau_1 \tau_2} \bar{x}_1^{\tau_1} \bar{x}_2^{\tau_2}.$$

Mathematically, we need to solve the least square fitting problem with energy

$$\sum_{j=1, j \neq i}^m \left(u(\bar{\mathbf{x}}^j) - \sum_{\tau_1=0}^2 \sum_{0 < \tau_1 + \tau_2 \leq 2} \beta_{\tau_1 \tau_2}^i (\bar{x}_1^j)^{\tau_1} (\bar{x}_2^j)^{\tau_2} \right)^2.$$

The corresponding linear system is given by

$$\bar{\mathbf{A}}\boldsymbol{\beta}^i = \bar{\mathbf{b}},$$

where

$$\begin{aligned} \boldsymbol{\beta}^i &= (\beta_{01}^i \ \beta_{02}^i \ \beta_{10}^i \ \beta_{11}^i \ \beta_{20}^i)^\top, \\ \bar{\mathbf{A}} &= \begin{pmatrix} \bar{x}_2^1 & (\bar{x}_2^1)^2 & \bar{x}_1^1 & \bar{x}_1^1 \bar{x}_2^1 & (\bar{x}_1^1)^2 \\ \bar{x}_2^2 & (\bar{x}_2^2)^2 & \bar{x}_1^2 & \bar{x}_1^2 \bar{x}_2^2 & (\bar{x}_1^2)^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{x}_2^m & (\bar{x}_2^m)^2 & \bar{x}_1^m & \bar{x}_1^m \bar{x}_2^m & (\bar{x}_1^m)^2 \end{pmatrix}_{(m-1) \times 5}, \\ \bar{\mathbf{b}} &= (u(\bar{\mathbf{x}}^1) \ u(\bar{\mathbf{x}}^2) \ \dots \ u(\bar{\mathbf{x}}^m))^\top. \end{aligned}$$

In the above approximation, we let the fitting quadratic surface through the point $(0, 0, u(\bar{\mathbf{x}}^i))$ and this leads to $\beta_{00}^i = u(\bar{\mathbf{x}}^i)$.

Once these coefficients $\boldsymbol{\beta}^i$ are determined, the derivatives of u at \mathbf{x}^i can be approximated by

$$\begin{aligned} \frac{\partial u}{\partial s_1} \Big|_{\mathbf{x}=\mathbf{x}^i} &= \frac{\partial u}{\partial \bar{x}_1} \Big|_{\bar{\mathbf{x}}=\bar{\mathbf{x}}^i} \approx \beta_{10}^i + \beta_{11}^i \bar{x}_2^i + 2\beta_{20}^i \bar{x}_1^i, \\ \frac{\partial u}{\partial s_2} \Big|_{\mathbf{x}=\mathbf{x}^i} &= \frac{\partial u}{\partial \bar{x}_2} \Big|_{\bar{\mathbf{x}}=\bar{\mathbf{x}}^i} \approx \beta_{01}^i + \beta_{11}^i \bar{x}_1^i + 2\beta_{02}^i \bar{x}_2^i, \\ \frac{\partial^2 u}{\partial s_1 \partial s_2} \Big|_{\mathbf{x}=\mathbf{x}^i} &= \frac{\partial^2 u}{\partial \bar{x}_1 \partial \bar{x}_2} \Big|_{\bar{\mathbf{x}}=\bar{\mathbf{x}}^i} \approx \beta_{11}^i, \\ \frac{\partial^2 u}{\partial s_1^2} \Big|_{\mathbf{x}=\mathbf{x}^i} &= \frac{\partial^2 u}{\partial \bar{x}_1^2} \Big|_{\bar{\mathbf{x}}=\bar{\mathbf{x}}^i} \approx 2\beta_{20}^i, \\ \frac{\partial^2 u}{\partial s_2^2} \Big|_{\mathbf{x}=\mathbf{x}^i} &= \frac{\partial^2 u}{\partial \bar{x}_2^2} \Big|_{\bar{\mathbf{x}}=\bar{\mathbf{x}}^i} \approx 2\beta_{02}^i. \end{aligned}$$

Similarly, we can approximate the first order derivative of $v_j^n + \frac{1}{\eta} p_j^n$ in equation (11) using exactly the same strategy. With all these ingredients, we solve the PDE (11) using the gradient descent, i.e. we introduce a pseudo-time t and update u using the following iteration

$$(u^i)^{n_1+1} = (u^i)^{n_1} + \Delta t (\Delta u^i)^{n_1} \quad (15)$$

for some inner iteration index n_1 . Using the GBPM, all derivatives in the gradient descent update Δu^i can be represented using the coefficients of the second-degree polynomial in the local least-squares approximation. The explicit formulation is given in Appendix E for completeness.

In practice, we do not obtain the steady state solution of the gradient descent equation to get u^{n+1} in Step 1 of **Algorithm 1**, but only perform several inner iterations in (15). Similar treatment can be found in the split Bregman iterations [12, 13]. One motivation is that the outer iteration only provides an intermediate solution to the final minimizer to (3). It is unnecessary to obtain the exact minimizers

in these inner iterations. Moreover, we impose the convex relaxation condition $u \in [0, 1]$ in this inner iteration step, which leads to the following projection gradient descent method

$$(u^i)^{n_1+1} = \text{Proj}_{[0,1]}((u^i)^{n_1} + \Delta t(\Delta u^i)^{n_1}) = \min\{\max\{(u^i)^{n_1} + \Delta t(\Delta u^i)^{n_1}, 0\}, 1\}. \quad (16)$$

In this paper, the inner iteration number is fixed as 1, which already provides good results for image segmentation according to our experiments.

There are some similarities and differences of the current approach comparing to the original GBPM in [20, 19, 21, 18] and also a recent approach for point cloud data in [23]. Similar to the original GBPM, we have a local polynomial least squares approximation to the interface in a local coordinate system for each grid point on a neighborhood of the interface. However, in the GBPM, we computed the L^2 projection of the active grid point onto the local polynomial. The current approach assumes, on the other hand, that we are given the closest point \mathbf{x} on the interface from any active grid point \mathbf{y} . And then we use these sampling points to construct the corresponding least square polynomial by constraining that the local polynomial contains the closest point \mathbf{x} . [23] has proposed to solve PDEs on manifolds by obtaining an approximation to the local metric using a similar idea as in the original GBPM. Therefore, the local reconstruction might not pass through its associated point cloud data location at all.

3.3 Computing Integrals on Manifolds in the GBPM Representation

In this subsection, we introduce a numerical approach to perform numerical integration on surfaces represented in the GBPM without constructing a triangulation of the manifold. The idea is based on the level set formulation where a surface integration is approximated by a volume integral using a smoothed delta-function

$$\int_{\mathcal{M}} f(s) ds = \int_{\Omega} f(\mathbf{y}) \delta(\phi) d\mathbf{y}, \quad (17)$$

where the curve \mathcal{M} contained in a domain Ω is represented by the zero level set of a signed distance function $\phi : \Omega \rightarrow \mathbb{R}$ (i.e. $|\nabla\phi| = 1$), the function $f(\mathbf{y})$ is an extension of $f(s)$ from the interface to the whole domain Ω , and the function $\delta(\cdot)$ is the Dirac delta function. Numerically, this delta function has to be smoothed over several $\Delta\mathbf{y}$ [30]. For example, one might use

$$\delta_{\epsilon}(\phi) = \begin{cases} 0, & |\phi| > \epsilon \\ \frac{1}{2\epsilon} + \frac{1}{2\epsilon} \cos\left(\frac{\pi\phi}{\epsilon}\right), & |\phi| \leq \epsilon \end{cases}$$

for $\epsilon = 1.5\Delta\mathbf{y}$. Note that in this integral formulation, the sign of the level set function plays no role at all. For a more detailed analysis of the approximation, we refer interested readers to [35, 11].

To approximate (17) in the current application, we first convert the GBPM representation to the distance function representation within ϵ -distance from the interface, i.e. we will activate any grid point which has distance from its foot point less than $\epsilon = 1.5\Delta\mathbf{y}$. This is simple in the GBPM since the foot point \mathbf{x}^i is defined to be the L^2 projection of the activated grid point \mathbf{y}^i onto the interface \mathcal{M} , which means the distance function $|\phi^i| = \|\mathbf{x}^i - \mathbf{y}^i\|_2$. Because the sign of the level set function is not needed in the evaluation, there is no need to determine the normal vector at the foot-points in practice.

Next, the level set formulation (17) requires an extension of f from \mathcal{M} to Ω , at least in the ϵ -neighborhood of \mathcal{M} . A natural way to do this is the orthogonal extension. Once again, since \mathbf{x}^i is the L^2 projection of \mathbf{y}^i onto \mathcal{M} , we simply use $f(\mathbf{y}^i) = f(\mathbf{x}^i)$. To conclusion, we approximate the surface integral (17) using

$$\int_{\mathcal{M}} f(s) ds \simeq \sum_i f(\mathbf{y}^i) \delta_{\epsilon}(\|\mathbf{x}^i - \mathbf{y}^i\|_2) \Delta\mathbf{y}. \quad (18)$$

If the manifold is open, we further require that the summation is done only on the grid point \mathbf{y}^i if its associated foot-point \mathbf{x}^i is not at the boundary of the surface.

4 The Experimental Results

In this section, we show the image segmentation results on various 2-D manifolds with the proposed algorithm. All these manifolds are represented by GPBM and the images defined on these manifolds contain noise. In all the experiments, the image intensity f are normalized $[0, 1]$, and the initial means c_1^0, c_2^0 are set as 0.33, 0.66 respectively. In addition, the classification functions u are often segmented into a binary image using a simple threshold at the value 0.5.

4.1 Segmentation on Closed Geometric Surfaces

First, we test our method on some closed geometric surfaces. Fig.2 shows some segmentation results with algorithm 1 on a sphere. In this figure, several simple geometry objects are mapped onto a sphere surface. The foot points \mathbf{x}^i are given by $\|\mathbf{x}^i - 0.5\|_2^2 = 0.4^2$ (a sphere centered at $(0.5, 0.5, 0.5)$, with radius 0.4) on domain $[0, 1] \times [0, 1] \times [0, 1]$, while the underlying computation mesh grid size is $200 \times 200 \times 200$. The intensities on the sphere are piecewise constants and corrupted by noise with a standard deviation $\frac{100}{255}$. In the experiments, the regularization parameter λ is set to 2.0×10^{-5} and the penalty parameter is fixed as 10λ . In order to better show the visual effect of the segmentation, three images from different views are given in Fig.2 (a)-(c), respectively. In the second row of this figure, the corresponding segmentation results are displayed. In these figures, we use different colors to represent different classes. It can be seen that the proposed algorithm can provide good results in this situation: the line on the sphere are straight along with the sphere and the curves are smooth though there are heavy noise in the original images. To compare the segmentation without regularization, the results with k -mean (i.e. set regularization parameter λ as 0) are illustrated in the third row of Fig.2. Fig.3 shows some classification results for the noisy world map on the earth. It can be seen that the proposed method can well segment this map.

Segmentation result on a smooth torus is displayed in Fig.4. In this experiment, a brain MR image is mapped onto a cloud points based torus, then the proposed algorithm is applied to segment the torus into 2 parts. Fig.4 (b) shows the final classification result. In this figure, we still use different color to represent the different classes.

4.2 Segmentation on Open Geometric Surfaces

Since the PDE is directly solved on particles (foot points) in the proposed method, it works well on the open surfaces, which may be difficult for implicit methods such as the level set method. Fig.5 shows that the proposed method nicely segments the cameraman image on a cylindrical surface into 2 parts.

To test more complicated surfaces, we applied the proposed algorithm on a Mobius strip which is open and is not orientable, and thus it has no level set representation. We have tested two different images on the strip and the results are shown in Fig.6 (b) and (d), respectively. In order to show the details of the GBPM surface, we zoom-in to these solutions and plot the results in Fig.6 (e) and (f). Note that the colors in these figures do not correspond to the true intensity value of the binary label function u , but is only used to represent the two different classes.

4.3 Segmentation on Some Scanned Surfaces in Applications

We test the proposed algorithm on the scanned Stanford bunny data (<http://graphics.stanford.edu/data/3Dscanrep/>). We define a piecewise image f on the bunny with noise (see the figures in the first row of Fig.7). The level of the noise is high, with a standard derivation $\frac{100}{255}$. We have shown our segmentation results in Fig.7 (d)-(f). To better show our results, we zoom-in to these solutions and plot our solutions in Fig.7 (g) and (h), respectively.

5 Discussions and Conclusions

We propose a method to segment images defined on manifolds represented by the GBPM. This method extends the convex segmentation methods on 2-D plane images to some general manifolds. By integrating the recently developed operator splitting techniques, we use some local approximation methods to solve the PDE which comes from the energy minimization on the foot points. Compared to the level set method, our proposed method provides a flexibly representation to handle open manifolds which are difficult to be defined using a signed distance function.

In this paper, even though we have only discussed the 2 phases segmentation, one can easily extend the method to address the multi-phases clustering. The proposed method can be further improved by employing the adaptive mesh techniques and some other local fitting techniques. It can also be extended to some other image processing areas such as image registration and restoration.

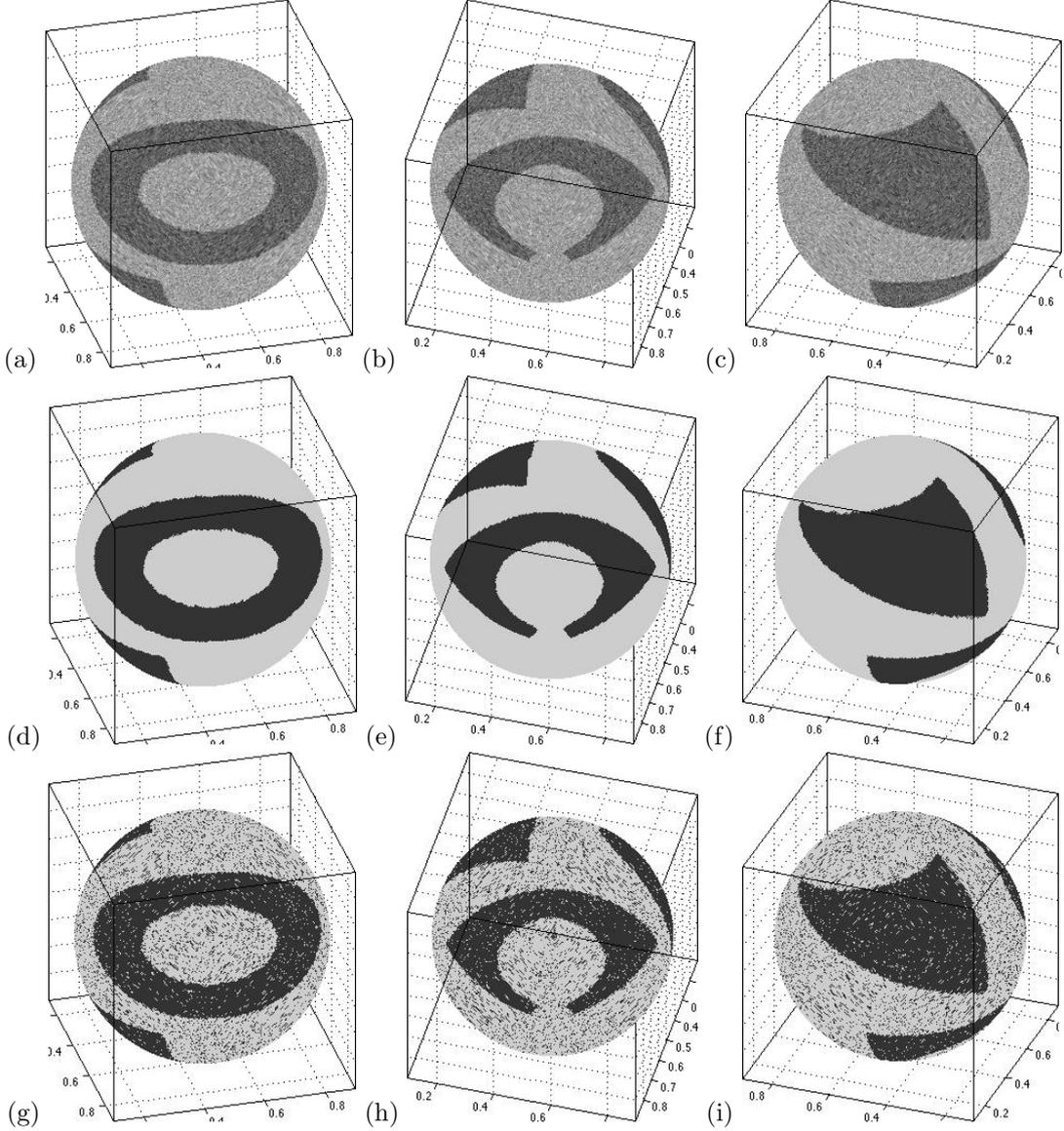


Fig. 2 (a-c) The noisy image on a sphere with three different views (using MATLAB function `view(75,30)`, `view(-15,-45)` and `view(-70,30)`, respectively). (d-f) Segmentation results using the proposed algorithm. (g-h) Segmentation results without the regularization term.

Acknowledgment

Leung would like to thank Prof. Ronald LM Lui for providing a conformal map of the Stanford bunny data to a sphere. The work of Leung was supported in part by the Hong Kong RGC under Grant GRF602210 and the HKUST grant RPC11SC06. The work of Liu was supported by National Natural Science Foundation of China (No. 11201032).

Appendix A. Proof of Theorem 1

We first prove the following lemma

Lemma 1 *Suppose $\mathcal{J}_1(\mathbf{v})$ is continuous and convex on a Hilbert space \mathbb{V} , and*

$$\mathcal{J} = \mathcal{J}_1(\mathbf{v}) + \langle \mathbf{p}, \mathbf{g}(\mathbf{v} - \mathbf{b}) \rangle + \frac{\eta}{2} \langle \mathbf{v} - \mathbf{b}, \mathbf{g}(\mathbf{v} - \mathbf{b}) \rangle$$

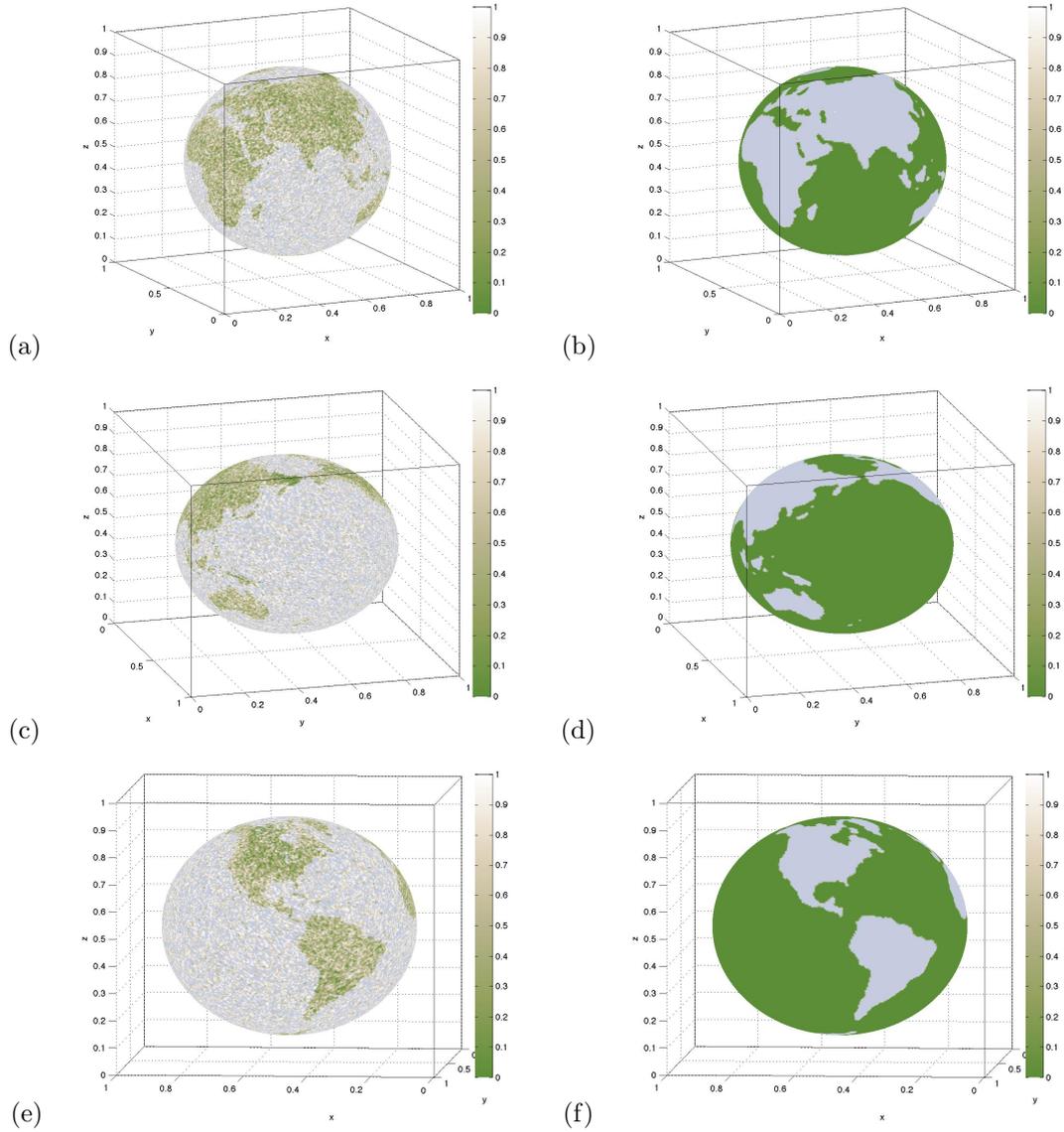


Fig. 3 (a,c,e) The world map on a sphere with three different views. (b,d,f) Our segmentation results.

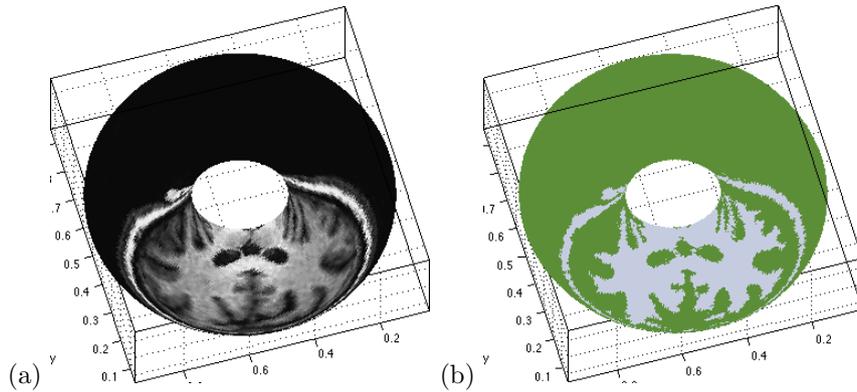


Fig. 4 (a) Brain image on torus. (b) Our segmentation result.

where $\eta > 0$ and \mathbf{g} is a positive definite and linear symmetric operator with bounded inverse, then the sequence $\{\mathbf{v}^n\}$ produced by the iteration scheme

$$\mathbf{v}^{n+1} = \arg \min_{\mathbf{v}} \mathcal{J}(\mathbf{v}, \mathbf{p}^n), \quad (19)$$

$$\mathbf{p}^{n+1} = \mathbf{p}^n + \tau \mathbf{g}(\mathbf{v}^{n+1} - \mathbf{b}), \quad (20)$$

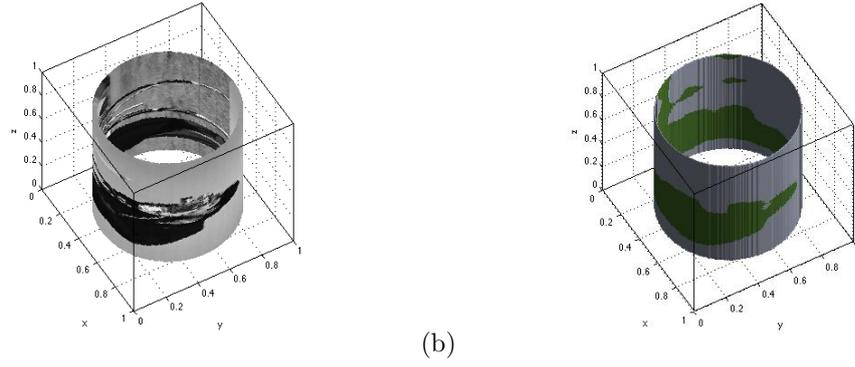


Fig. 5 (a) The mapped cameraman image on the cylinder. (b) Our segmentation result.

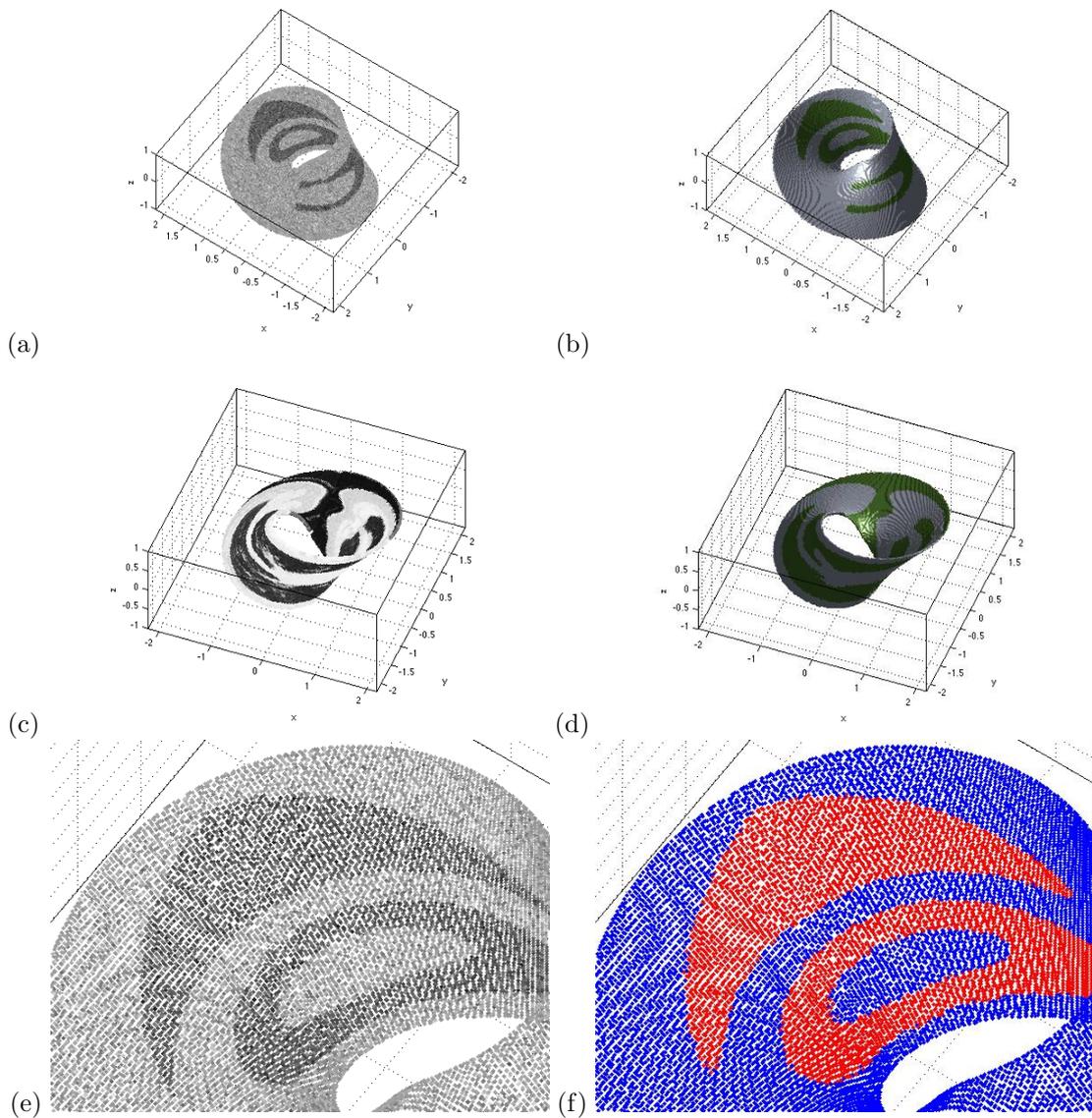


Fig. 6 (a,c) Noisy images mapped on the Mobius strip. (b,d) Our segmentation results. (e,f) Zoom-in of (a) and (b), respectively.

converges, i.e. $\mathbf{v}^n \rightarrow \mathbf{v}^*$ when $0 < \tau < \frac{2\eta}{\Lambda_{\max}}$, where \mathbf{v}^* is the saddle point $(\mathbf{v}^*, \mathbf{p}^*)$ of $\mathcal{J}(\mathbf{v}, \mathbf{p})$. Here Λ_{\max} is the largest eigenvalue of \mathbf{g} .

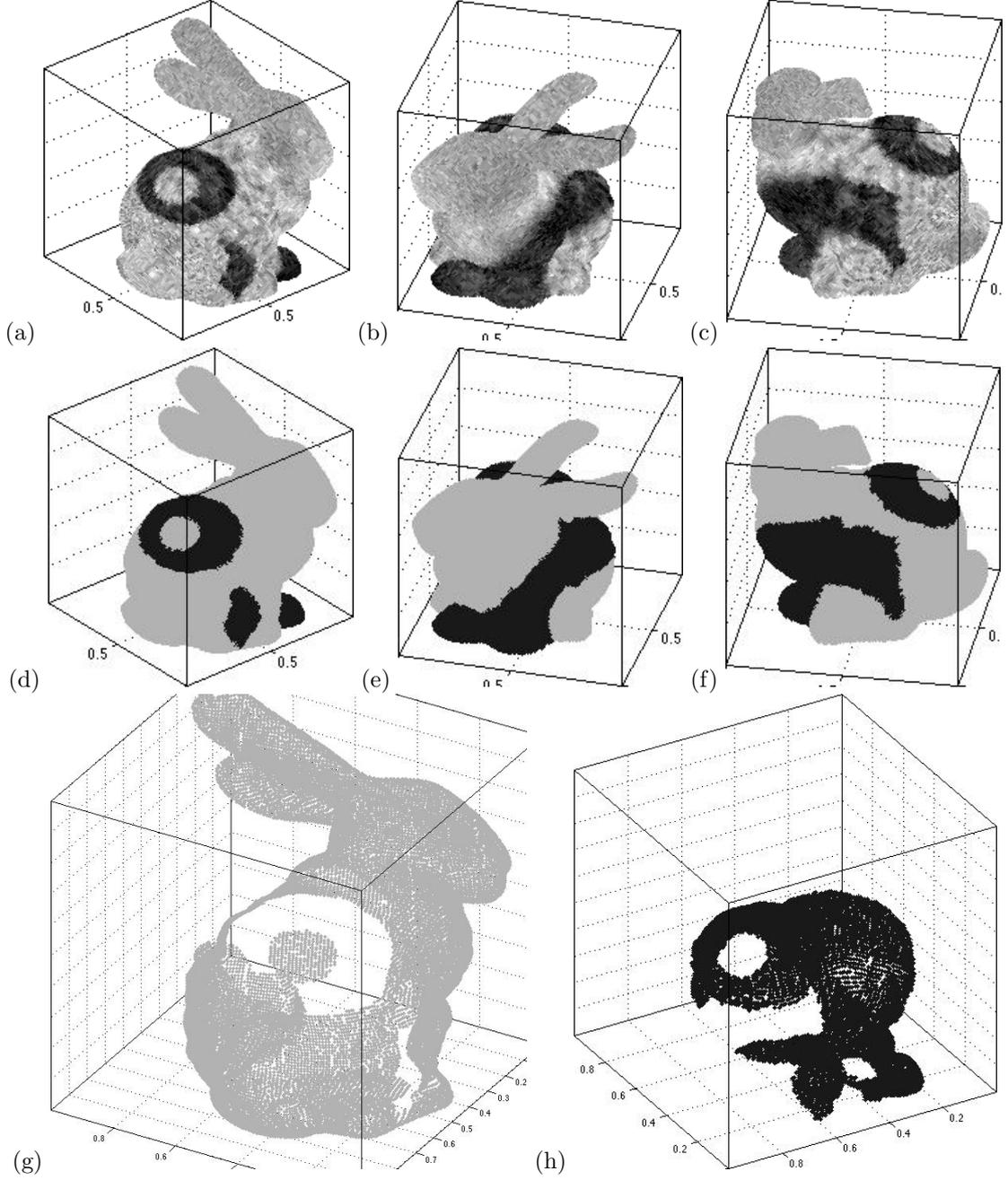


Fig. 7 Segmentation on the Stanford bunny data by our method. (a-c) The noisy images on the bunny. (d-f) Our segmentation results. (g-h) Zoom-in of the results in (d-f).

Proof Since $(\mathbf{v}^*, \mathbf{p}^*)$ is a saddle of \mathcal{J} , we have $\mathbf{v}^* = \mathbf{b}$ by $\frac{\partial \mathcal{J}}{\partial \mathbf{p}}|_{(\mathbf{p}^*, \mathbf{v}^*)} = 0$. Let $\partial \mathcal{J}_1(\mathbf{v})$ be the subgradient of \mathcal{J}_1 at \mathbf{v} , i.e. $\partial \mathcal{J}_1(\mathbf{v}) = \{\bar{\mathbf{v}} \in \bar{\mathbb{V}} : \mathcal{J}_1(\mathbf{q}) - \mathcal{J}_1(\mathbf{v}) \geq \langle \bar{\mathbf{v}}, \mathbf{q} - \mathbf{v} \rangle, \forall \mathbf{q} \in \mathbb{V}\}$, where $\bar{\mathbb{V}}$ is the conjugate space of \mathbb{V} . According to the first order optimization conditions of (19), we have

$$\begin{aligned} \mathbf{d}^{n+1} &:= -\mathbf{g}\mathbf{p}^n - \eta\mathbf{g}(\mathbf{v}^{n+1} - \mathbf{b}) \in \partial \mathcal{J}_1(\mathbf{v}^{n+1}), \\ \mathbf{d}^* &:= -\mathbf{g}\mathbf{p}^* - \eta\mathbf{g}(\mathbf{v}^* - \mathbf{b}) \in \partial \mathcal{J}_1(\mathbf{v}^*), \end{aligned}$$

thus

$$\mathbf{d}^{n+1} - \mathbf{d}^* = -\mathbf{g}(\mathbf{p}^n - \mathbf{p}^*) - \eta\mathbf{g}(\mathbf{v}^{n+1} - \mathbf{v}^*).$$

Taking the inner product with $\mathbf{v}^{n+1} - \mathbf{v}^*$ for both sides of the above equation, it becomes

$$\langle \mathbf{p}^n - \mathbf{p}^*, \mathbf{g}(\mathbf{v}^{n+1} - \mathbf{v}^*) \rangle = -\langle \mathbf{d}^{n+1} - \mathbf{d}^*, \mathbf{v}^{n+1} - \mathbf{v}^* \rangle - \eta \langle \mathbf{v}^{n+1} - \mathbf{v}^*, \mathbf{g}(\mathbf{v}^{n+1} - \mathbf{v}^*) \rangle. \quad (21)$$

By the iteration equation (20) and the fact that $\mathbf{v}^* - \mathbf{b} = 0$, we have

$$\mathbf{p}^{n+1} - \mathbf{p}^* = \mathbf{p}^n - \mathbf{p}^* + \tau\mathbf{g}(\mathbf{v}^{n+1} - \mathbf{v}^*).$$

Taking the norm for the both sides of the above equation, we get

$$\|\mathbf{p}^{n+1} - \mathbf{p}^*\|^2 = \|\mathbf{p}^n - \mathbf{p}^*\|^2 + \tau^2 \|\mathbf{g}(\mathbf{v}^{n+1} - \mathbf{v}^*)\|^2 + 2\tau \langle \mathbf{p}^n - \mathbf{p}^*, \mathbf{g}(\mathbf{v}^{n+1} - \mathbf{v}^*) \rangle. \quad (22)$$

Substituting (21) into (22), we have

$$\|\mathbf{p}^{n+1} - \mathbf{p}^*\|^2 - \|\mathbf{p}^n - \mathbf{p}^*\|^2 = -2\tau \langle \mathbf{d}^{n+1} - \mathbf{d}^*, \mathbf{v}^{n+1} - \mathbf{v}^* \rangle - \langle \mathbf{v}^{n+1} - \mathbf{v}^*, (-\tau^2 \mathbf{g}^2 + 2\tau \eta \mathbf{g})(\mathbf{v}^{n+1} - \mathbf{v}^*) \rangle. \quad (23)$$

Since \mathcal{J}_1 is convex, $\mathbf{d}^{n+1} \in \partial \mathcal{J}_1(\mathbf{v}^{n+1})$ and $\mathbf{d}^* \in \partial \mathcal{J}_1(\mathbf{v}^*)$, thus

$$\langle \mathbf{d}^{n+1} - \mathbf{d}^*, \mathbf{v}^{n+1} - \mathbf{v}^* \rangle \geq 0. \quad (24)$$

With the condition $0 < \tau < \frac{2\eta}{\Lambda_{\max}}$, we conclude that the operator $-\tau^2 \mathbf{g}^2 + 2\tau \eta \mathbf{g}$ is positive definite.

Now, using both (23) and (24), we have $\|\mathbf{p}^{n+1} - \mathbf{p}^*\|^2 - \|\mathbf{p}^n - \mathbf{p}^*\|^2 < 0$, which implies that the sequence $\|\mathbf{p}^n - \mathbf{p}^*\|^2$ is monotonic decreasing with a lower bound 0 and so it must be convergent.

Finally from (23), we conclude that $\mathbf{v}^n \rightarrow \mathbf{v}^*$.

Now, we can use this Lemma to prove Theorem 1. It is easy to check that $\lambda \int_{\mathcal{M}} \sqrt{\mathbf{v}^T \mathbf{g} \mathbf{v}} dM$ is convex when \mathbf{g} is positive definite. For any fixed u and \mathbf{c} , let $\mathbf{b} = \nabla_s u$ and follow the lemma, one can show that $\{\mathbf{v}^n\}$ produced by equations (8) and (10) converges to the saddle point $(\cdot, \mathbf{v}^*, \mathbf{p}^*, \cdot)$ of $L(\cdot, \mathbf{v}, \mathbf{p}, \cdot)$, i.e. $\mathbf{v}^n \rightarrow \mathbf{v}^*$. Since $(\cdot, \mathbf{v}^*, \mathbf{p}^*, \cdot)$ is the saddle point, we have $\mathbf{v}^* = \nabla_s u$, which completes the proof.

Appendix B. Derivation of (11)

The directional derivative

$$\begin{aligned} \frac{d\tilde{L}(u + \tau w)}{d\tau} \Big|_{\tau=0} &= \frac{d}{d\tau} \left\{ \int_{\mathcal{M}} (f - c_1)^2 (u + \tau w) dM + \int_{\mathcal{M}} (f - c_2)^2 [1 - (u + \tau w)] dM \right. \\ &\quad \left. + \frac{\eta}{2} \int_{\mathcal{M}} (\mathbf{v} - \nabla_s (u + \tau w) + \frac{1}{\eta} \mathbf{p})^T \mathbf{g} (\mathbf{v} - \nabla_s (u + \tau w) + \frac{1}{\eta} \mathbf{p}) dM \right\} \Big|_{\tau=0} \\ &= \frac{\eta}{2} \frac{d}{d\tau} \left\{ \int_f \sqrt{EG - F^2} (\mathbf{v} - \nabla_s (u + \tau w) + \frac{1}{\eta} \mathbf{p})^T \mathbf{g} (\mathbf{v} - \nabla_s (u + \tau w) + \frac{1}{\eta} \mathbf{p}) ds \right\} \Big|_{\tau=0} \\ &\quad + \int_{\mathcal{M}} [(f - c_1)^2 - (f - c_2)^2] w dM \\ &= -\eta \int_f \sqrt{EG - F^2} \langle \mathbf{g} \mathbf{v} - \mathbf{g} \nabla_s u + \frac{1}{\eta} \mathbf{g} \mathbf{p}, \nabla_s w \rangle ds + \int_{\mathcal{M}} [(f - c_1)^2 - (f - c_2)^2] w dM \\ &= \eta \int_f \nabla_s \cdot \left(\sqrt{EG - F^2} (\mathbf{g} \mathbf{v} - \mathbf{g} \nabla_s u + \frac{1}{\eta} \mathbf{g} \mathbf{p}) \right) w ds + \int_{\mathcal{M}} [(f - c_1)^2 - (f - c_2)^2] w dM \\ &= \eta \int_{\mathcal{M}} \frac{1}{\sqrt{EG - F^2}} \nabla_s \cdot \left(\sqrt{EG - F^2} (\mathbf{g} \mathbf{v} - \mathbf{g} \nabla_s u + \frac{1}{\eta} \mathbf{g} \mathbf{p}) \right) w dM \\ &\quad + \int_{\mathcal{M}} [(f - c_1)^2 - (f - c_2)^2] w dM. \end{aligned}$$

By the variational formulation $\frac{d\tilde{L}(u + \tau w)}{d\tau} \Big|_{\tau=0} = \langle \frac{\delta \tilde{L}}{\delta u}, w \rangle$, one gets

$$\begin{aligned} \frac{\delta \tilde{L}}{\delta u} &= -\frac{\eta}{\sqrt{EG - F^2}} \nabla_s \cdot \left(\sqrt{EG - F^2} \mathbf{g} \nabla_s u \right) + \frac{\eta}{\sqrt{EG - F^2}} \nabla_s \cdot \left(\sqrt{EG - F^2} \mathbf{g} (\mathbf{v} + \frac{1}{\eta} \mathbf{p}) \right) \\ &\quad + (f - c_1)^2 - (f - c_2)^2, \end{aligned}$$

which leads to equation (11).

Appendix C. Derivation of (12)

Denote $\mathbf{q}^n = \nabla_s u^{n+1} - \frac{\mathbf{p}^n}{\eta}$, we have

$$\frac{\delta \tilde{L}}{\delta \mathbf{v}} = \frac{\lambda}{\sqrt{\mathbf{v}^T \mathbf{g} \mathbf{v}}} \mathbf{g} \mathbf{v} + \eta \mathbf{g} (\mathbf{v} - \mathbf{q}^n).$$

Solving $\frac{\delta \tilde{L}}{\delta \mathbf{v}} = 0$ for \mathbf{v} , we get

$$\left(\frac{\lambda}{\sqrt{(\mathbf{v}^{n+1})^T \mathbf{g} \mathbf{v}^{n+1}}} + \eta \right) \mathbf{v}^{n+1} = \eta \mathbf{q}^n. \quad (25)$$

Taking modulus $\|\cdot\|_{\mathbf{g}}$ for the two sides of (25), it becomes

$$\|\mathbf{v}^{n+1}\|_{\mathbf{g}} = \|\mathbf{q}^n\|_{\mathbf{g}} - \frac{\lambda}{\eta}$$

if $\|\mathbf{q}^n\|_{\mathbf{g}} \geq \frac{\lambda}{\eta}$. Plugging it back to the above (25), we get

$$\mathbf{v}^{n+1} = \frac{\mathbf{q}^n}{\|\mathbf{q}^n\|_{\mathbf{g}}} \left(\|\mathbf{q}^n\|_{\mathbf{g}} - \frac{\lambda}{\eta} \right). \quad (26)$$

On the other hand, if $\|\mathbf{q}^n\|_{\mathbf{g}} < \frac{\lambda}{\eta}$, then

$$\begin{aligned} \tilde{L}(\mathbf{v}, \cdot) &= \lambda \int_{\mathcal{M}} \sqrt{\mathbf{v}^T \mathbf{g} \mathbf{v}} dM + \frac{\eta}{2} \int_{\mathcal{M}} (\mathbf{v} - \mathbf{q})^T \mathbf{g} (\mathbf{v} - \mathbf{q}^n) dM \\ &= \lambda \int_{\mathcal{M}} \sqrt{\mathbf{v}^T \mathbf{g} \mathbf{v}} dM + \frac{\eta}{2} \int_{\mathcal{M}} (\|\mathbf{v}\|_{\mathbf{g}}^2 + \|\mathbf{q}^n\|_{\mathbf{g}}^2) dM - \eta \int_{\mathcal{M}} \mathbf{v}^T \mathbf{g} \mathbf{q}^n dM \\ &\geq \lambda \int_{\mathcal{M}} \sqrt{\mathbf{v}^T \mathbf{g} \mathbf{v}} dM + \frac{\eta}{2} \int_{\mathcal{M}} (\|\mathbf{v}\|_{\mathbf{g}}^2 + \|\mathbf{q}^n\|_{\mathbf{g}}^2) dM - \eta \int_{\mathcal{M}} \|\mathbf{v}\|_{\mathbf{g}} \cdot \|\mathbf{q}^n\|_{\mathbf{g}} dM \\ &> \frac{\eta}{2} \int_{\mathcal{M}} (\|\mathbf{v}\|_{\mathbf{g}}^2 + \|\mathbf{q}^n\|_{\mathbf{g}}^2) dM. \end{aligned}$$

This means that $\mathbf{v}^{n+1} = 0$ must be the minimizer of \tilde{L} with respect to \mathbf{v} . Summarizing these two results, we get the \mathbf{g} -shrinkage operator (12).

Appendix D. Minimizer of (14)

Let $\mathbf{H} = \sum_{j=1}^m (\mathbf{x}^j - \mathbf{x}^i) (\mathbf{x}^j - \mathbf{x}^i)^T = \mathbf{U} \begin{pmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{pmatrix} \mathbf{U}^T$, where $\lambda_1 \geq \lambda_2 \geq \lambda_3$ and $\mathbf{U} = (U_1 \ U_2 \ U_3)$ is an orthogonal matrix. Then

$$E = \sum_{j=1}^m (\langle \mathbf{x}^j, \tilde{\mathbf{n}} \rangle - \langle \mathbf{x}^i, \tilde{\mathbf{n}} \rangle)^2 = \tilde{\mathbf{n}}^T \mathbf{H} \tilde{\mathbf{n}} = (\mathbf{U}^T \tilde{\mathbf{n}})^T \begin{pmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{pmatrix} (\mathbf{U}^T \tilde{\mathbf{n}}).$$

Since $(\mathbf{U}^T \tilde{\mathbf{n}})^T (\mathbf{U}^T \tilde{\mathbf{n}}) = 1$, thus we have $E \geq \lambda_3$. If $\tilde{\mathbf{n}} = U_3$, $E = U_3^T \mathbf{H} U_3 = \lambda_3$. Thus we conclude that $\mathbf{n} = U_3$ is a minimizer of E such that $|\mathbf{n}| = 1$, which completes the proof.

Appendix E. Explicit formula for the gradient descent update $\Delta \mathbf{u}^i$

In this Appendix, we state explicitly the formula to compute the gradient descent update of u^i as a function of the coefficients of the second order polynomial approximation in the GBPM representation. For convenience, let us denote $h_j = v_j^n + \frac{1}{\eta} p_j^n$ and write the coefficients of the approximated second-degree polynomial at \mathbf{x}^i for functions h_1 and h_2 as $\gamma_{\tau_1 \tau_2}^i$ and $\delta_{\tau_1 \tau_2}^i$ ($0 \leq \tau_1 + \tau_2 \leq 2$, $\tau_1, \tau_2 \in \mathbb{N}$), respectively.

The PDE (11) in the explicit form is given by

$$-\frac{\eta}{EG - F^2} (S_1 + S_2 S_3 + S_4 S_5) + \frac{\eta}{EG - F^2} \left[S_6 + S_7 + S_8 + \frac{1}{EG - F^2} S_9 \right] + (f - c_1)^2 - (f - c_2)^2 = 0,$$

where

$$\begin{aligned}
S_1 &= G \frac{\partial^2 u}{\partial s_1^2} - 2F \frac{\partial^2 u}{\partial s_1 \partial s_2} + E \frac{\partial^2 u}{\partial s_2^2}, \\
S_2 &= \left\langle \frac{\partial^2 \mathbf{x}}{\partial s_1 \partial s_2}, \frac{\partial \mathbf{x}}{\partial s_2} \right\rangle - \left\langle \frac{\partial^2 \mathbf{x}}{\partial s_2^2}, \frac{\partial \mathbf{x}}{\partial s_1} \right\rangle - \frac{1}{EG - F^2} (T_1 G - T_2 F), \\
S_3 &= \frac{\partial u}{\partial s_1}, \\
S_4 &= \left\langle \frac{\partial^2 \mathbf{x}}{\partial s_1 \partial s_2}, \frac{\partial \mathbf{x}}{\partial s_1} \right\rangle - \left\langle \frac{\partial^2 \mathbf{x}}{\partial s_1^2}, \frac{\partial \mathbf{x}}{\partial s_2} \right\rangle - \frac{1}{EG - F^2} (T_2 E - T_1 F), \\
S_5 &= \frac{\partial u}{\partial s_2}, \\
S_6 &= h_1 \left(\left\langle \frac{\partial^2 \mathbf{x}}{\partial s_1 \partial s_2}, \frac{\partial \mathbf{x}}{\partial s_2} \right\rangle - \left\langle \frac{\partial^2 \mathbf{x}}{\partial s_2^2}, \frac{\partial \mathbf{x}}{\partial s_1} \right\rangle \right), \\
S_7 &= h_2 \left(\left\langle \frac{\partial^2 \mathbf{x}}{\partial s_1 \partial s_2}, \frac{\partial \mathbf{x}}{\partial s_1} \right\rangle - \left\langle \frac{\partial^2 \mathbf{x}}{\partial s_1^2}, \frac{\partial \mathbf{x}}{\partial s_2} \right\rangle \right), \\
S_8 &= G \frac{\partial h_1}{\partial s_1} - F \left(\frac{\partial h_2}{\partial s_1} + \frac{\partial h_1}{\partial s_2} \right) + E \frac{\partial h_2}{\partial s_2}, \\
S_9 &= (G h_1 - F h_2) T_1 + (-F h_1 + E h_2) T_2
\end{aligned}$$

and

$$\begin{aligned}
T_1 &= \left\langle \frac{\partial^2 \mathbf{x}}{\partial s_1^2}, \frac{\partial \mathbf{x}}{\partial s_1} \right\rangle G + \left\langle \frac{\partial^2 \mathbf{x}}{\partial s_1 \partial s_2}, \frac{\partial \mathbf{x}}{\partial s_2} \right\rangle E - \left(\left\langle \frac{\partial^2 \mathbf{x}}{\partial s_1^2}, \frac{\partial \mathbf{x}}{\partial s_2} \right\rangle + \left\langle \frac{\partial^2 \mathbf{x}}{\partial s_1 \partial s_2}, \frac{\partial \mathbf{x}}{\partial s_1} \right\rangle \right) F, \\
T_2 &= \left\langle \frac{\partial^2 \mathbf{x}}{\partial s_1 \partial s_2}, \frac{\partial \mathbf{x}}{\partial s_1} \right\rangle G + \left\langle \frac{\partial^2 \mathbf{x}}{\partial s_2^2}, \frac{\partial \mathbf{x}}{\partial s_2} \right\rangle E - \left(\left\langle \frac{\partial^2 \mathbf{x}}{\partial s_1 \partial s_2}, \frac{\partial \mathbf{x}}{\partial s_2} \right\rangle + \left\langle \frac{\partial^2 \mathbf{x}}{\partial s_2^2}, \frac{\partial \mathbf{x}}{\partial s_1} \right\rangle \right) F.
\end{aligned}$$

Now, replacing all local geometry by the local polynomial least square approximation, we have

$$\begin{aligned}
\Delta u^i &= \frac{\eta}{E^i G^i - (F^i)^2} (S_1^i + S_2^i S_3^i + S_4^i S_5^i) - \frac{\eta}{E^i G^i - (F^i)^2} \left[S_6^i + S_7^i + S_8^i + \frac{1}{E^i G^i - (F^i)^2} S_9^i \right] \\
&\quad - (f - c_1)^2 + (f - c_2)^2,
\end{aligned}$$

where

$$\begin{aligned}
S_1^i &= 2\beta_{20}^i G^i - 2\beta_{11}^i F^i + 2\beta_{02}^i E^i, \\
S_2^i &= \alpha_{11}^i (\alpha_{01}^i + \alpha_{11}^i \bar{x}_1^i + 2\alpha_{02}^i \bar{x}_2^i) - 2\alpha_{02}^i (\alpha_{10}^i + \alpha_{11}^i \bar{x}_2^i + 2\alpha_{20}^i \bar{x}_1^i) - \frac{1}{E^i G^i - (F^i)^2} (T_1^i G^i - T_2^i F^i), \\
S_3^i &= \beta_{10}^i + \beta_{11}^i \bar{x}_2^i + 2\beta_{20}^i \bar{x}_1^i, \\
S_4^i &= \alpha_{11}^i (\alpha_{10}^i + \alpha_{11}^i \bar{x}_2^i + 2\alpha_{20}^i \bar{x}_1^i) - 2\alpha_{20}^i (\alpha_{01}^i + \alpha_{11}^i \bar{x}_1^i + 2\alpha_{02}^i \bar{x}_2^i) - \frac{1}{E^i G^i - (F^i)^2} (T_2^i E^i - T_1^i F^i), \\
S_5^i &= \beta_{01}^i + \beta_{11}^i \bar{x}_1^i + 2\beta_{02}^i \bar{x}_2^i, \\
S_6^i &= h_1^i [\alpha_{11}^i (\alpha_{01}^i + \alpha_{11}^i \bar{x}_1^i + 2\alpha_{02}^i \bar{x}_2^i) - 2\alpha_{02}^i (\alpha_{10}^i + \alpha_{11}^i \bar{x}_2^i + 2\alpha_{20}^i \bar{x}_1^i)], \\
S_7^i &= h_2^i [\alpha_{11}^i (\alpha_{10}^i + \alpha_{11}^i \bar{x}_2^i + 2\alpha_{20}^i \bar{x}_1^i) - 2\alpha_{20}^i (\alpha_{01}^i + \alpha_{11}^i \bar{x}_1^i + 2\alpha_{02}^i \bar{x}_2^i)], \\
S_8^i &= G^i (\gamma_{10}^i + \gamma_{11}^i \bar{x}_2^i + 2\gamma_{20}^i \bar{x}_1^i) + E^i (\delta_{01}^i + \delta_{11}^i \bar{x}_1^i + 2\delta_{02}^i \bar{x}_2^i) \\
&\quad - F^i (\gamma_{01}^i + \gamma_{11}^i \bar{x}_1^i + 2\gamma_{02}^i \bar{x}_2^i + \delta_{10}^i + \delta_{11}^i \bar{x}_2^i + 2\delta_{20}^i \bar{x}_1^i), \\
S_9^i &= (G^i h_1^i - F^i h_2^i) T_1^i + (-F^i h_1^i + E^i h_2^i) T_2^i
\end{aligned}$$

and

$$\begin{aligned}
T_1^i &= 2\alpha_{20}^i (\alpha_{10}^i + \alpha_{11}^i \bar{x}_2^i + 2\alpha_{20}^i \bar{x}_1^i) G^i + \alpha_{11}^i (\alpha_{01}^i + \alpha_{11}^i \bar{x}_1^i + 2\alpha_{02}^i \bar{x}_2^i) E^i \\
&\quad - \left[2\alpha_{20}^i (\alpha_{01}^i + \alpha_{11}^i \bar{x}_1^i + 2\alpha_{02}^i \bar{x}_2^i) + \alpha_{11}^i (\alpha_{10}^i + \alpha_{11}^i \bar{x}_2^i + 2\alpha_{20}^i \bar{x}_1^i) \right] F^i, \\
T_2^i &= \alpha_{11}^i (\alpha_{10}^i + \alpha_{11}^i \bar{x}_2^i + 2\alpha_{20}^i \bar{x}_1^i) G^i + 2\alpha_{02}^i (\alpha_{01}^i + \alpha_{11}^i \bar{x}_1^i + 2\alpha_{02}^i \bar{x}_2^i) E^i \\
&\quad - \left[2\alpha_{02}^i (\alpha_{10}^i + \alpha_{11}^i \bar{x}_2^i + 2\alpha_{20}^i \bar{x}_1^i) + \alpha_{11}^i (\alpha_{01}^i + \alpha_{11}^i \bar{x}_1^i + 2\alpha_{02}^i \bar{x}_2^i) \right] F^i, \\
h_1^i &= \sum_{\tau_1=0}^2 \sum_{0 \leq \tau_1 + \tau_2 \leq 2} \gamma_{\tau_1 \tau_2}^i (\bar{x}_1^i)^{\tau_1} (\bar{x}_2^i)^{\tau_2}, \\
h_2^i &= \sum_{\tau_1=0}^2 \sum_{0 \leq \tau_1 + \tau_2 \leq 2} \delta_{\tau_1 \tau_2}^i (\bar{x}_1^i)^{\tau_1} (\bar{x}_2^i)^{\tau_2}.
\end{aligned}$$

References

1. Bae, E., Yuan, J., Tai, X.: Global minimization for continuous multiphase partitioning problems using a dual approach. *International Journal of Computer Vision* **92**(1), 112–129 (2011)
2. Bae, E., Yuan, J., Tai, X.: Simultaneous convex optimization of regions and region parameters in image segmentation models. Tech. rep., UCLA CAM report 11-83 (2011)
3. Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J., Osher, S.: Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and Vision* **28**(2), 151–167 (2007)
4. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision* **40**, 120–145 (2011)
5. Chan, T., Golub, G., Mulet, P.: A nonlinear primal-dual method for total variation-based image restoration. *SIAM Journal on Scientific Computing* **20**(6), 1964–1977 (1999)
6. Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Transactions on Image Processing* **10**(2), 266–277 (2001). DOI 10.1109/83.902291
7. Cheng, L., Karagozian, A., Chan, T.: The level set method applied to geometrically based motion, materials science, and image processing. Ph.D. thesis, University of California Los Angeles (2000)
8. Cremers, D., Pock, T., Kolev, K., Chambolle, A.: Convex relaxation techniques for segmentation, stereo and multiview reconstruction. Tech. rep., in *Advances in Markov Random Fields for Vision and Image Processing*, MIT Press (2011)
9. Delaunoy, A., Fundana, K., Prados, E., Heyden, A.: Convex multi-region segmentation on manifolds. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 662–669 (2009)
10. Do Carmo, M.P.: *Differential Geometry of Curves and Surfaces*. Prentice Hall (1976)
11. Engquist, B., Tornberg, A., Tsai, R.: Discretization of dirac delta functions in level set methods. *Journal of Computational Physics* **207**, 28–51 (2005)
12. Goldstein, T., Bresson, X., Osher, S.: Geometric applications of the split bregman method: Segmentation and surface reconstruction. *Journal of Scientific Computing* **45**, 272–293 (2009)
13. Goldstein, T., Osher, S.: The split bregman method for l1 regularized problems. *SIAM Journal on Imaging Sciences* **2**, 323–343 (2009)
14. Kazufumi, I., Karl, K.: Augmented lagrangian methods for nonsmooth, convex optimization in hilbert spaces. *Nonlinear Analysis* **41**(5-6), 591–616 (2000). DOI DOI:10.1016/S0362-546X(98)00299-5
15. Kimmel, R.: Intrinsic scale space for images on surfaces: the geodesic curvature flow. *Graphical Models and Image Processing* **59**(5), 365–372 (1997)
16. Krueger, M., Delmas, P., GimelFarb, G.: Active contour based segmentation of 3d surfaces. In: *In European Conference on Computer Vision*, pp. 350–363 (2008)
17. Lai, R., Liang, J., Zhao, H.: A local mesh method for solving pdes on point clouds. Tech. rep., UCLA Report 12-60 (2012)
18. Leung, S., Lowengrub, J., Zhao, H.: A grid based particle method for solving partial differential equations on evolving surfaces and modeling high order geometrical motion. *Journal of Computational Physics* **230**(7), 2540–2561 (2011). DOI 10.1016/j.jcp.2010.12.029. URL <http://www.sciencedirect.com/science/article/pii/S0021999110007035>
19. Leung, S., Zhao, H.: A grid based particle method for evolution of open curves and surfaces. *Journal of Computational Physics* **228**, 7706–7728 (2009)
20. Leung, S., Zhao, H.: A grid based particle method for moving interface problems. *Journal of Computational Physics* **228**, 2993–3024 (2009)
21. Leung, S., Zhao, H.: Gaussian beam summation for diffraction in inhomogeneous media based on the grid based particle method. *Communications in Computational Physics* **8**, 758–796 (2010)
22. Liang, J., Lai, R., Wong, T., Zhao, H.: Geometric understanding of point clouds using laplace-beltrami operator. In: *IEEE Conference on Computer Vision and Pattern Recognition* (2012)
23. Liang, J., Zhao, H.: Solving partial differential equations on point clouds. Tech. rep., UCLA Report 12-25 (2012)
24. Lie, J., Lysaker, M., X.C., T.: A binary level set model and some applications to mumford-shah image segmentation. *IEEE Transactions on Image Processing* **15**(5), 1171–1181 (2006)
25. Liu, J., Ku, Y., Leung, S.: Expectation-maximization algorithm with total variation regularization for vector-valued image segmentation. *Journal of Visual Communication and Image Representation* **23**, 1234–1244 (2012)

-
26. Macdonald, C., Ruuth, S.: The implicit closest point method for the numerical solution of partial differential equations on surfaces. *SIAM Journal on Scientific Computing* **31**(6), 4330–4350 (2009). DOI 10.1137/080740003. URL <http://people.maths.ox.ac.uk/~macdonald/icpm.pdf>
 27. Meyer, M., Desbrun, M., Schroder, P., Barr, A.: *Discrete Differential-Geometry Operator for Triangulated 2-Manifolds*. Visualization and Mathematics III, H.-C. Hege and K. Polthier, eds, Springer Verlag (2002)
 28. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics* **42**, 577–685 (1989)
 29. Osher, S., Sethian, J.: Fronts propagating with curvature dependent speed: algorithms based on hamiltonjacobi formulations. *Journal of Computational Physics* **79**, 12–49 (1988)
 30. Peskin, C.: Numerical analysis of blood flow in the heart. *Journal of Computational Physics* **25**, 220–252 (1977)
 31. Rockafellar, R.: A dual approach to solving nonlinear programming problems by unconstrained optimization,. *Mathematical Programming* **5**, 354–373 (1973)
 32. Setzer, S.: Operator splittings, bregman methods and frame shrinkage in image processing. *International Journal of Computer Vision* **92**(3), 265–280 (2010)
 33. Smereka, P.: Spiral crystal growth. *Physica D* **128**, 282–301 (2000)
 34. Tian, L., Macdonald, C., Ruuth, S.: Segmentation on surfaces with the closest point method. In: Proc. ICIP09, 16th IEEE International Conference on Image Processing, pp. 3009–3012. Cairo, Egypt (2009). DOI 10.1109/ICIP.2009.5414447. URL <http://people.maths.ox.ac.uk/~macdonald/TianMacdonaldRuuth.pdf>
 35. Tornberg, A., Engquist, B.: Numerical approximations of singular source terms in differential equations. *Journal of Computational Physics* **200**, 462–488 (2004)
 36. Wan, M., Wang, Y., Bae, E., Tai, X., Wang, D.: Reconstructing open surfaces via graph-cuts. *IEEE Transactions on Visualization and Computer Graphics* DOI:10.1109/TVCG.2012.119 (2012). DOI 10.1109/TVCG.2012.119
 37. Wang, Y., Yang, J., Yin, W., Zhang, Y.: A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences* **1**(3), 248–272 (2008). DOI 10.1137/080724265
 38. Wu, C., Tai, X.: Augmented lagrangian method, dual methods, and split bregman iteration for rof, vectorial tv, and high order models. *SIAM Journal on Imaging Sciences* **3**(3), 300–339 (2010)
 39. Wu, C., Tai, X.: A level set formulation of geodesic curvature flow on simplicial surfaces. *IEEE Transactions on Visualization and Computer Graphics* **16**(4), 647–662 (2010)
 40. Wu, C., Zhang, J., Duan, Y., Tai, X.: Augmented lagrangian method for total variation based image restoration and segmentation over triangulated surfaces. *Journal of Scientific Computing* **50**(1), 145–166 (2012)