

Point cloud surface segmentation based on volumetric eigenfunctions of the Laplace-Beltrami operator [☆]



Xinge Li ^a, Yongjie Jessica Zhang ^b, Xuyang Yang ^c, Haibo Xu ^a, Guoliang Xu ^{d,*}

^a Institute of Applied Physics and Computational Mathematics, Beijing, China

^b Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, USA

^c Kunming Shipborne Equipment Research and Test Center, 750 Proving Ground, Kunming, China

^d Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Article history:

Received 21 March 2018

Received in revised form 31 March 2019

Accepted 31 March 2019

Available online 4 April 2019

Keywords:

Point cloud surface

Segmentation

Level set form of Laplace-Beltrami operator

Volumetric eigenfunction

Split Bregman iteration

ABSTRACT

In the process of surface modeling from scanned point data, a segmentation that partitions a point cloud into meaningful regions or extracts important features from the 3D point data plays an important role in compressing the scanned data and fitting surface patches. In this paper, a new spectral point cloud surface segmentation method is proposed based on volumetric eigenfunctions of the Laplace-Beltrami operator. The proposed algorithm consists of two main steps. Firstly, the point cloud surface is modeled as the union of a bunch of level set surfaces, on which the eigenfunctions are computed from the level set form of the Laplace-Beltrami operator using the finite element method. Secondly, a new vectorial volumetric eigenfunction segmentation model is developed based on the classical Mumford-Shah model, in which we approximate volumetric eigenfunctions by piecewise-constant functions, and the point cloud surface is segmented via segmenting the volumetric eigenfunctions. Instead of solving the Euler-Lagrange equation by evolution implementation, the split Bregman iteration, which is shown to be a fast algorithm, is utilized. Experimental results demonstrate that our volumetric eigenfunction based technique yields superior segmentation results in terms of accuracy and robustness, compared with the surface eigenfunction based method.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Reverse engineering is a technology that enables us to generate a computerized representation of an existing object based on point data acquired from the surface. The existence of a computer model provides enormous gains in improving the quality and efficiency of design, manufacture and analysis (Várady et al., 1997). The typical process of reverse engineering begins with collecting point data from the surfaces of a physical object. The initial point data acquired by a measuring device generally require pre-processing such as noise filtering, smoothing, merging and data ordering in order to be useable in subsequent operations. Then the preprocessed point data should be divided into several smooth regions for surface modeling purposes, which is called the segmentation process. After that, a surface model can be generated either by a curve-net-based method or by a polygon-based modeling method. The curve-net-based method is more commonly used



This paper has been recommended for acceptance by Rida Farouki.

The Trial Version shows

E-mail address: xugl@ccit.ac.cn (G. Xu).

<https://doi.org/10.1016/j.cagd.2019.03.004>

0167-8396/© 2019 Elsevier B.V. All rights reserved.

when generating a surface model from the scanned data. The polygon-based modeling method is fast and efficient in fitting surfaces; however, it generates less accurate surface models than the curve-net-based method and it is also difficult to modify the final model (Várady and Benkő, 2000). Each method has its advantages and disadvantages and either method can be used based on the geometric shape of an object.

To facilitate surface modeling tasks, the point cloud should be arranged well and segmented. However, it is not easy for a computer to perform segmentation even for a simple object with quadric surfaces. Segmenting an object with free-form surface is even more difficult and usually leads to ambiguous solutions. In reverse engineering, segmentation has the greatest effects on product development duration and quality of the final surface model. The segmentation task is not only difficult but also time-consuming, and, in practice, has usually been performed with a lot of user interactions.

Many researchers have tried to develop segmentation methods to extract geometric information, such as edges and smooth regions, from the scanned data. These point data segmentation methods can be classified into three categories: edge-detection methods (Fan et al., 1987; Chen and Liu, 1997; Milroy et al., 1997; Yang and Lee, 1999), region-growing methods (Hoffman and Jain, 1987; Besl and Jain, 1988) and hybrid methods (Yokoya and Levine, 1989; Checchin et al., 1997; Zhao and Zhang, 1997). The edge-detection methods attempt to detect discontinuities in the surfaces that form the closed boundaries of components in the point data. The region-growing methods, on the other hand, proceed with segmentation by detecting continuous surfaces that have homogeneity or similar geometrical properties. Most segmentation methods based on discontinuity can run short when edges have small variation or when regions are homogeneous. The region-based segmentation methods allow for reliable homogeneity in a region but these methods often fail to localize regional outlines accurately. To resolve limitations of the above two approaches, hybrid segmentation approaches have been developed, where the edge- and region-based methods are combined. In addition, Rabbani et al. (2006) used local surface normals and point connectivity which can be enforced using either k -nearest or fixed distance neighbors to divide a given unstructured 3D point cloud into a set of smooth surface patches. Bizjak (2016) developed a segmentation method based on locally fitted surfaces. The topological structure of the input point cloud was realised by a nearest-neighbor graph, while locally fitted surfaces were used to quantify the object shapes. Noise was filtered by a graph-cut and segmentation was performed by linking nodes to faces which they fit the best.

Most of the above methods employ curve or surface fitting to find edge points or curves. However, the surface- or curve-fitting tasks are time-consuming and it is also difficult to extract the exact edge points because the scanned data consist of discrete points and the edge points are not always included in these data. The method proposed by Woo et al. (2002) can extract edge-neighborhood points using 3D grids and segment the point cloud of an object with quadric surfaces without fitting any curve or surface. Vo et al. (2015) presented an octree-based region growing algorithm for fast and accurate segmentation of terrestrial and aerial LiDAR point clouds. This approach employs a voxel model to simplify the initial data, index the data, and define neighborhood groups for computing local surface properties.

In addition to shape segmentation methods focusing on geometric features, many techniques were proposed based on the spectrum of the Laplace-Beltrami operator (LBO), machine learning and convolutional networks. Reuter (2009) employed eigenfunctions of the LBO to construct consistent shape segmentations. With topological persistence it is possible to construct hierarchical segmentations that remain stable across near-isometric populations of shapes and with respect to noise or mesh quality. Kalogerakis et al. (2010) introduced a data-driven approach to simultaneous segmentation and labeling of components in 3D meshes. Labeling of mesh components is expressed as an optimization of a Conditional Random Field (CRF), with terms assessing the consistency of faces with labels. The objective function is learned from a collection of labeled training meshes. Recently, a deep architecture combining image-based Fully Convolutional Networks and surface-based CRFs was proposed to yield coherent segmentations of 3D shapes (Kalogerakis et al., 2017). The entire architecture is trained end-to-end.

Note that the inputs of the above-mentioned spectral methods are 3D polygon meshes, while we start from point cloud. In this paper, we develop a novel spectral surface segmentation method based on volumetric eigenfunctions of the LBO, partitioning the point cloud of an object with free-form surfaces into meaningful regions. The main contribution of this paper is to introduce an efficiently computable set of volumetric eigenfunctions for segmentation. Different from Rustamov (2011), where Rustamov interpolated the LBO eigenfunctions of the boundary surface into the interior volume via barycentric coordinates, we compute volumetric eigenfunctions from the level set form of the LBO using the finite element method. To begin with, 3D Voronoi tessellation of the point cloud data is computed to approximate the radius of the point, thereby the 3D shape can be generated from the point cloud data using the Gaussian density map.

Furthermore, we adapt the classical Mumford-Shah model and the split Bregman method to the volumetric setting. A vectorial piecewise-constant variational model for 3D shape segmentation is developed, and the split Bregman iteration for minimizing the energy functional is established. We explore a new way to solve the partial differential equation (PDE) with boundary condition, that is, making use of the boundary condition to extend the domain of the unknowns onto the one with boundary condition, and then updating the unknowns in the original domain. For the domain extension which is formulated as a system of linear equations, we utilize the L^2 -gradient flow method to resolve its corresponding least square problem. The Trial Version of this paper is organized as follows. Section 2 briefly reviews the differential geometry on implicit surfaces. Section 3 is devoted to the generation of the 3D shape from the point cloud data and the computation of LB volumetric eigenfunctions. The proposed vectorial piecewise-constant based variational model for 3D shape segmentation and the corresponding split Bregman iteration are described in Section 4. Section 5 presents several experimental results. Finally, we conclude the paper with a summary and point out future directions in Section 6.

2. Differential geometry of implicit surfaces

In this section, we introduce some background of differential geometry and interested readers are referred to Xu and Zhang (2012) for details. For simplicity, we assume that $\|\nabla\phi\| \neq 0$ in an open neighborhood Ω_c of the level-set surface $S_c = \{\mathbf{x} : \phi(\mathbf{x}) = c\}$. Let ∇ , div and Δ denote the classical gradient, divergence and Laplace operators, respectively. From the implicit function theorem, we know that S_c is a smooth surface and for each point $\mathbf{x}$ on S_c , we have

$$\mathbf{n} = \frac{\nabla\phi}{\|\nabla\phi\|}.$$

The gradient of a vector-valued function $\mathbf{v} = [v_1, v_2, v_3]^T \in \mathbb{R}^3$ is written as $\nabla\mathbf{v} = [\nabla v_1, \nabla v_2, \nabla v_3] \in \mathbb{R}^{3 \times 3}$. Then we can define the following differential operators.

Tangential gradient operator. Let $f \in C^1(\Omega_c)$, then the tangential gradient of f on the level-set surface S_c is defined by

$$\nabla_\phi f = \mathbf{P} \nabla f, \quad (1)$$

where $\mathbf{P} = \mathbf{I} - \mathbf{n}\mathbf{n}^T$ is a projection operator onto the tangent plane of surface S_c , and $\mathbf{I}$ is the identity projection. Hence, $\mathbf{P} = \mathbf{P}^T = \mathbf{P}^2$ holds.

Tangential divergence operator. Assume $\mathbf{v}$ is a smooth vector field defined on Ω_c , then the action of the tangential divergence operator div_ϕ onto $\mathbf{v}$ is defined by

$$\text{div}_\phi(\mathbf{v}) = \langle 2H\mathbf{n}, \mathbf{v} \rangle + \text{div}(\mathbf{v}) - \mathbf{n}^T(\nabla\mathbf{v})\mathbf{n}. \quad (2)$$

In particular, if $\mathbf{v}$ is a tangential vector field on the level-set surface S_c , we have

$$\text{div}_\phi(\mathbf{v}) = \text{div}(\mathbf{v}) - \mathbf{n}^T(\nabla\mathbf{v})\mathbf{n}. \quad (3)$$

Laplace-Beltrami operator. Let $f \in C^2(\Omega_c)$, then the LBO Δ_ϕ applying to f is defined as

$$\Delta_\phi f = \text{div}_\phi(\nabla_\phi f). \quad (4)$$

Theorem 1. Given an $\epsilon > 0$, suppose $S_c = \{\mathbf{x} \in \mathbb{R}^3 : \phi(\mathbf{x}) = c\}$ is an orientable closed surface for each $c \in [-\epsilon, \epsilon]$, and $\|\nabla\phi(\mathbf{x})\| > 0$ for all $\mathbf{x} \in \Omega_\epsilon := \bigcup_{c \in [-\epsilon, \epsilon]} S_c$. We further assume that $\mathbf{v}$ is a smooth vector field on Ω_ϵ satisfying $\langle \mathbf{v}, \nabla\phi \rangle = 0$. Let $f \in C^1(\Omega_\epsilon)$, then we have

$$\int_{\Omega_\epsilon} \langle \mathbf{v}, \nabla_\phi f \rangle \|\nabla\phi\| d\mathbf{x} = - \int_{\Omega_\epsilon} f \text{div}_\phi(\mathbf{v}) \|\nabla\phi\| d\mathbf{x}. \quad (5)$$

Note that the Green's formula (5) for the LBO Δ_ϕ is a corollary of the Riemannian divergence Theorem (Xu and Zhang, 2012).

3. Shape generation and volumetric eigenfunction computation

In this section, we first generate the 3D shape from the point cloud data, and then compute volumetric eigenfunctions of the LBO using the finite element method.

3.1. 3D shape generation from the point cloud surface

Gaussian density map has been widely used in biomolecular geometry calculation, visualization, and biophysics. Gaussian density maps consist of the center location and radius information of each atom. Here we use the Gaussian density map to construct 3D shapes.

The Trial Version point cloud surface $\{\mathbf{x}_i \in \mathbb{R}^3\}_{i=1}^N$, we move its barycenter $\mathbf{c} = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$ to the origin of coordinates, that is $\mathbf{x}_i := \mathbf{x}_i - \mathbf{c}$, and then compute its 3D Voronoi tessellation using the software library Voro++ (2008). The custom output routines contain complete information about the structure of Voronoi cells, and we extract the list of neighbors for each particle and average the distances from this particle to all its neighbors. Then we obtain the data structure $\{(\mathbf{x}_i, r_i)\}_{i=1}^N$, where N is the number of the particles on the point cloud surface, and $(\mathbf{x}_i, r_i)$ are the position and radius of the i^{th} particle.



We compute its Gaussian density map ϕ in a rectilinear grid with

$$\phi(\mathbf{x}) = \sum_{i=1}^N e^{-b(\|\mathbf{x}-\mathbf{x}_i\|^2 - r_i^2)}, \quad (6)$$

where $b > 0$ is called the *blobbyness parameter*, which controls the rate of decay in each particle's Gaussian kernel. In Liu et al. (2015), the suggested $b_0 = 0.3875$ provides quite a good geometric description of the biomolecule. However, the particles on the point cloud surface are denser than the atoms in the biomolecule, and the shape boundary becomes smoother and more inflated if we choose $b = b_0$. To generate ideal shape with b_0 from the point cloud surface, we expand the point cloud data such that the distances between particles ($d_{\text{particles}}$) are similar to the distances between atoms (d_{atoms}). In fact, we do not need to move particles, we can just integrate the scaling factor $\kappa = \frac{d_{\text{atoms}}}{d_{\text{particles}}}$ into b , that is, $b = \kappa^2 b_0$. Here, d_{atoms} is computed as the average of the mean distance between the atoms in 1BPB, 1BPD, 1BPE, 2BPC, 2BPF and 2BPG, which are all selected from the Protein Data Bank (<http://www.rcsb.org/>). Therefore, we choose a large b when the particles are dense and a small b for the sparse case.

It is well known that $\phi \in C^\infty(\mathbb{R}^3)$. The 3D shape, denoted as Ω , is represented by the set of points where the density is greater than 1.0, that is $\Omega = \{\mathbf{x} \in \mathbb{R}^3 : \phi(\mathbf{x}) \geq 1\}$. It is a collection of a series of implicit surfaces $\mathcal{S}_c = \{\mathbf{x} : \phi(\mathbf{x}) = c\}$ with $c \geq 1$. Here, ϕ is called a *level-set function* and c is its corresponding *level-set value* or *isovalue*. Eigenfunctions are computed from the LBO defined on implicit surfaces, and the shape can be segmented using the distribution of its eigenfunctions.

3.2. Computation of LBO volumetric eigenfunctions

Suppose $\mathcal{S}_c$ is an orientable closed surface, and $\|\nabla\phi(\mathbf{x})\| \geq 0$ for all $\mathbf{x} \in \Omega$. Letting $f \in C^2(\Omega)$, the eigenfunction of LBO should satisfy

$$\Delta_\phi f = -\lambda f, \quad (7)$$

where λ is the corresponding eigenvalue. According to Theorem 1, $\forall f \in C^2(\Omega)$ and $\forall h \in C^1(\Omega)$, we have

$$\int_{\Omega} h \Delta_\phi f \|\nabla\phi\| d\mathbf{x} = - \int_{\Omega} \langle \nabla_\phi f, \nabla_\phi h \rangle \|\nabla\phi\| d\mathbf{x}. \quad (8)$$

Therefore, we plug Eq. (7) into Eq. (8) and obtain

$$\int_{\Omega} \langle \nabla_\phi f, \nabla_\phi h \rangle \|\nabla\phi\| d\mathbf{x} = \lambda \int_{\Omega} f h \|\nabla\phi\| d\mathbf{x}. \quad (9)$$

Letting $\{\varphi_\alpha\}_{\alpha=1}^K$ be a set of basis functions defined on Ω , K be the node number and $\varphi_\alpha \in C^2(\Omega)$, f can be approximately represented as $f = \sum_{\alpha=1}^K f_\alpha \varphi_\alpha$. Taking $h = \varphi_\beta$ ($\beta = 1, 2, \dots, K$), we have

$$\sum_{\alpha=1}^K f_\alpha \int_{\Omega} \langle \nabla_\phi \varphi_\alpha, \nabla_\phi \varphi_\beta \rangle \|\nabla\phi\| d\mathbf{x} = \lambda \sum_{\alpha=1}^K f_\alpha \int_{\Omega} \varphi_\alpha \varphi_\beta \|\nabla\phi\| d\mathbf{x}. \quad (10)$$

Letting $m_{\alpha\beta} = \int_{\Omega} \langle \nabla_\phi \varphi_\alpha, \nabla_\phi \varphi_\beta \rangle \|\nabla\phi\| d\mathbf{x}$ and $c_{\alpha\beta} = \int_{\Omega} \varphi_\alpha \varphi_\beta \|\nabla\phi\| d\mathbf{x}$, the eigenfunctions of LBO can be obtained by solving the eigenproblem

$$\mathbf{M}\mathbf{f} = \lambda \mathbf{C}\mathbf{f}, \quad (11)$$

where $\mathbf{M} = [m_{\alpha\beta}]$, $\mathbf{C} = [c_{\alpha\beta}]$ and $\mathbf{f} = [f_1, f_2, \dots, f_K]^T$. Note that $\mathbf{M}$ and $\mathbf{C}$ are large sparse symmetric matrices, $\mathbf{M}$ is positive semi-definite, and $\mathbf{C}$ is positive definite.

For the above generalized eigenvalue problem (11), the non-negative eigenvalues λ_α and the corresponding eigenvectors $\mathbf{f}_\alpha$ should satisfy

$$\mathbf{M}\mathbf{f}_\alpha = \lambda_\alpha \mathbf{C}\mathbf{f}_\alpha \text{ and } \mathbf{C}\mathbf{f}_\alpha = \mathbf{I}, \quad (12)$$

where $\mathbf{f}_\alpha = [f_{\alpha 1}, f_{\alpha 2}, \dots, f_{\alpha K}]^T$, $\mathbf{f}_\alpha = [f_{\alpha 1}, f_{\alpha 2}, \dots, f_{\alpha K}]^T$, $\alpha = 1, 2, \dots, K$, $\mathbf{I}$ is the identity matrix (Parlett, 1998). The α^{th} eigenfunction of the LBO on Ω can thereby be represented as

$$f_\alpha(\mathbf{x}) = \sum_{\beta=1}^K f_{\alpha\beta} \varphi_\beta(\mathbf{x}). \quad (13)$$

We will then give a detailed description of the basis functions $\{\varphi_\alpha\}_{\alpha=1}^K$. Let $\mathbf{x}_{ijk} = [x_i, y_j, z_k]^T$ ($i = 0, 1, \dots, W, j = 0, 1, \dots, H, k = 0, 1, \dots, L$) be the rectilinear grid points of the minimal cube which contains Ω , where $(W+1) \times (H+1) \times (L+1)$ is the mesh size. We can construct the basis function φ_{ijk} for each point in Ω as the tensor product of three cubic B-spline basis functions, that is

$$\varphi_{ijk}(\mathbf{x}) = N_i(x)N_j(y)N_k(z), \quad \forall \mathbf{x}_{ijk} \in \Omega, \quad (14)$$

where $N_i(x)$, $N_j(y)$ and $N_k(z)$ are cubic B-spline basis functions defined on $[x_{i-2}, x_{i+2}]$, $[y_{j-2}, y_{j+2}]$ and $[z_{k-2}, z_{k+2}]$, respectively.

In this study, only a few eigenvalues and eigenfunctions are needed. As we deal with a large but sparse problem, an iterative Krylov methods (such as the Lanczos method) can be used efficiently. We use the shift-invert method as implemented in ARPACK (2009) and SuperLU (Li et al., 2011, 2005) to obtain the lowest several eigenvalues and their corresponding eigenfunctions.

4. Shape segmentation

In this section, we first propose a vectorial piecewise-constant based variational model for the 3D shape segmentation, and then minimize the proposed energy functional with split Bregman iteration.

4.1. The vectorial piecewise-constant based variational model

Let

$$\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_\Pi(\mathbf{x})]^T,$$

where $f_\alpha(\mathbf{x})$ ($\alpha = 1, 2, \dots, \Pi$) are the first Π eigenfunctions of the LBO on Ω . To segment the shape Ω , we propose the vectorial piecewise-constant Mumford-Shah (VPCMS) model. Let $C = \{\mathbf{x} \in \Omega : \psi(\mathbf{x}) = 0\}$, where $\psi : \Omega \rightarrow \mathbb{R}$ is a Lipschitz function, the closed contour C separates Ω into two disjoint regions $\Omega_1 = \{\mathbf{x} \in \Omega : \psi(\mathbf{x}) > 0\}$ and $\Omega_2 = \{\mathbf{x} \in \Omega : \psi(\mathbf{x}) < 0\}$.

Let

$$\mathbf{G}(\mathbf{x}) = \begin{cases} \mathbf{c}_1, & \mathbf{x} \in \Omega_1, \\ \mathbf{c}_2, & \mathbf{x} \in \Omega_2, \\ \mathbf{0}, & \mathbf{x} \in \Omega \setminus (\bigcup_{l=1}^2 \Omega_l), \end{cases}$$

where $\mathbf{c}_1 = [c_{11}, c_{12}, \dots, c_{1\Pi}]^T \in \mathbb{R}^\Pi$, $\mathbf{c}_2 = [c_{21}, c_{22}, \dots, c_{2\Pi}]^T \in \mathbb{R}^\Pi$, then the error between the original intensities $\mathbf{F}$ and the approximate intensities $\mathbf{G}$ is

$$\int_{\Omega} \|\mathbf{F}(\mathbf{x}) - \mathbf{G}(\mathbf{x})\|^2 d\mathbf{x} = \int_{\Omega_1} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_1\|^2 d\mathbf{x} + \int_{\Omega_2} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_2\|^2 d\mathbf{x}.$$

We define the Heaviside function

$$H(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0, \end{cases}$$

and let $H_1(x) = H(x)$, $H_2(x) = 1 - H(x)$. Now we consider the following energy functional

$$\begin{aligned} \mathcal{E}_{\text{VPCMS}}(\mathbf{c}_1, \mathbf{c}_2, C) &:= \mathcal{E}_{\text{VPCMS}}(\mathbf{c}_1, \mathbf{c}_2, \psi) \\ &= \lambda_1 \int_{\Omega_1} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_1\|^2 d\mathbf{x} + \lambda_2 \int_{\Omega_2} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_2\|^2 d\mathbf{x} + \int_C dA \\ &= \lambda_1 \int_{\Omega} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_1\|^2 H_1(\psi) d\mathbf{x} + \lambda_2 \int_{\Omega} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_2\|^2 H_2(\psi) d\mathbf{x} \\ &\quad + \int_{\Omega} \|\nabla H(\psi)\| d\mathbf{x}, \end{aligned} \quad (15)$$

where dA denotes the surface element. In our vectorial piecewise-constant based variational model, we aim to find $(\mathbf{c}_1^*, \mathbf{c}_2^*, \psi^*)$ such that

$$(\mathbf{c}_1^*, \mathbf{c}_2^*, \psi^*) = \arg \min_{\mathbf{c}_1, \mathbf{c}_2, \psi} \mathcal{E}_{\text{VPCMS}}(\mathbf{c}_1, \mathbf{c}_2, \psi). \quad (16)$$

The first two terms of the energy functional (15) are referred as the fidelity term

$$\mathcal{E}_0(\mathbf{c}_1, \mathbf{c}_2, \psi) = \lambda_1 \int_{\Omega} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_1\|^2 H_1(\psi) d\mathbf{x} + \lambda_2 \int_{\Omega} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_2\|^2 H_2(\psi) d\mathbf{x}, \quad (17)$$

which is used to minimize the error between $\mathbf{F}$ and $\mathbf{G}$. In order to get a smooth segmentation contour, we impose the area regularization term $\int_{\Omega} \|\nabla H(\psi)\| d\mathbf{x}$, namely, the area of the segmentation contour, onto the above fidelity term. Positive parameters λ_1 and λ_2 are the weights of the errors on the regions Ω_1 and Ω_2 . The greater the weight coefficient λ_i is, the greater the error penalty of the region Ω_i we can obtain.

Assume that the to-be-segmented intensities $\mathbf{F}$ is distributed by two regions of vectorial piecewise-constant intensities $\mathbf{c}_1^0$ and $\mathbf{c}_2^0$, $C_0 = \{\mathbf{x} \in \Omega : \psi_0(\mathbf{x}) = 0\}$ is the intersecting contour of these two regions, $\mathbf{F} = \mathbf{c}_1^0$ inside C_0 , and $\mathbf{F} = \mathbf{c}_2^0$ outside C_0 . It is easy to find that if $C = C_0$, then

$$\mathcal{E}_0(\mathbf{c}_1^0, \mathbf{c}_2^0, \psi_0) = \min \mathcal{E}_0(\mathbf{c}_1, \mathbf{c}_2, \psi) = 0.$$

Otherwise $C \neq C_0$, we immediately have $\mathcal{E}_0(\mathbf{c}_1, \mathbf{c}_2, \psi) > 0$. Therefore, as the contour C is right on the intersecting contour of the two regions, the fidelity term is minimized to the desired result.

4.2. Split Bregman iteration for the proposed model

To achieve the target of segmentation, we need to solve the energy model (15). Here,

$$\begin{aligned} \mathcal{E}_{\text{VPCMS}}(\mathbf{c}_1, \mathbf{c}_2, \psi) = & \lambda_1 \sum_{\alpha=1}^{\Pi} \int_{\Omega} |f_{\alpha}(\mathbf{x}) - \mathbf{c}_{1\alpha}|^2 H_1(\psi) d\mathbf{x} \\ & + \lambda_2 \sum_{\alpha=1}^{\Pi} \int_{\Omega} |f_{\alpha}(\mathbf{x}) - \mathbf{c}_{2\alpha}|^2 H_2(\psi) d\mathbf{x} + \int_{\Omega} \|\nabla H(\psi)\| d\mathbf{x}. \end{aligned} \quad (18)$$

Alternating iterative method is applied to compute the minimum solution of the energy functional (18). In each iteration step, we minimize $\mathcal{E}_{\text{VPCMS}}(\mathbf{c}_1, \mathbf{c}_2, \psi)$ with respect to $\mathbf{c}_1$, $\mathbf{c}_2$ and ψ , respectively. So we first fix ψ , and the minimal solution of $\mathcal{E}_{\text{VPCMS}}(\mathbf{c}_1, \mathbf{c}_2, \psi)$ with respect to $\mathbf{c}_1$ and $\mathbf{c}_2$ are obtained:

$$c_{l\alpha} = \frac{\int_{\Omega} f_{\alpha}(\mathbf{x}) H_l(\psi) d\mathbf{x}}{\int_{\Omega} H_l(\psi) d\mathbf{x}} = \frac{\int_{\Omega} f_{\alpha}(\mathbf{x}) H_l(\psi) d\mathbf{x}}{|\Omega_l|}, \quad l = 1, 2, \quad \alpha = 1, \dots, \Pi. \quad (19)$$

Then, we minimize (15) in terms of ψ by fixing $\mathbf{c}_1$ and $\mathbf{c}_2$. Let

$$\psi(\mathbf{x}, \varepsilon) = \psi(\mathbf{x}) + \varepsilon \Psi(\mathbf{x}), \quad \Psi \in C^1(\Omega; \mathbb{R}),$$

we have the first-order variation of the energy functional (15)

$$\begin{aligned} & \left. \frac{d}{d\varepsilon} \mathcal{E}_{\text{VPCMS}}(\mathbf{c}_1, \mathbf{c}_2, \psi(\cdot, \varepsilon)) \right|_{\varepsilon=0} \\ = & \lambda_1 \int_{\Omega} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_1\|^2 \delta(\psi) \Psi d\mathbf{x} - \lambda_2 \int_{\Omega} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_2\|^2 \delta(\psi) \Psi d\mathbf{x} \\ & - \int_{\Omega} \operatorname{div} \left(\frac{\nabla \psi}{\|\nabla \psi\|} \right) \delta(\psi) \Psi d\mathbf{x} + \int_{\partial\Omega} \left\langle \frac{\nabla \psi}{\|\nabla \psi\|}, \frac{\nabla \phi}{\|\nabla \phi\|} \right\rangle \delta(\psi) \Psi dA. \end{aligned}$$

Introducing an artificial time variable $t \geq 0$, we derive the L^2 -gradient flow:

$$\begin{cases} \frac{\partial \psi}{\partial t} = \delta(\psi) \left[-\lambda_1 \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_1\|^2 + \lambda_2 \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_2\|^2 + \operatorname{div} \left(\frac{\nabla \psi}{\|\nabla \psi\|} \right) \right], & \text{in } \Omega, \\ \left\langle \frac{\nabla \psi}{\|\nabla \psi\|}, \frac{\nabla \phi}{\|\nabla \phi\|} \right\rangle = 0, & \text{on } \partial\Omega, \end{cases} \quad (20)$$

is the impulse function. In the numerical calculation, we usually approximate the Heaviside function $H(x)$ and impulse function $\delta(x)$ by continuous functions (Chan and Vese, 2001; Zhao et al., 1996).

It is obvious that updating ψ by solving the L^2 -gradient flow (20) is time consuming, and the obtained solution is just a local minimizer. Since for any given function ψ , the function $H(\psi)$ is binary. Then the minimization of $\mathcal{E}_{\text{VPCMS}}(\mathbf{c}_1, \mathbf{c}_2, \cdot)$ is to find a minimizer over all binary functions. This minimization is non-convex, and it is easy to get into the undesirable local minimizer for an inappropriately given initialization. To improve the efficiency of our segmentation algorithm and overcome the deficiency of non-convexity, we adopt the split Bregman iteration (Goldstein and Osher, 2009; Goldstein et al., 2010; Jia et al., 2009; Yang et al., 2010) to update ψ . We first give the following theorem, which is well-known and was used already in Chan and Nikolova (2006) in the 2D form.

Theorem 2. For any given fixed $\mathbf{c}_1, \mathbf{c}_2 \in \mathbb{R}^\Pi$, a global minimizer for $\mathcal{E}_{\text{VPCMS}}(\mathbf{c}_1, \mathbf{c}_2, \cdot)$ can be obtained by solving the following convex optimization problem

$$\min_{0 \leq \psi \leq 1} \left\{ \underbrace{\int_{\Omega} (\lambda_1 \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_1\|^2 - \lambda_2 \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_2\|^2) \psi(\mathbf{x}) d\mathbf{x}}_{\mathcal{F}_{\text{VPCMS}}(\mathbf{c}_1, \mathbf{c}_2, \psi)} + \int_{\Omega} \|\nabla \psi(\mathbf{x})\| d\mathbf{x} \right\}, \quad (21)$$

as long as $C(\gamma) = \{\mathbf{x} \in \Omega : \psi(\mathbf{x}) = \gamma\}$, $\Omega_1(\gamma) = \{\mathbf{x} \in \Omega : \psi(\mathbf{x}) > \gamma\}$ and $\Omega_2(\gamma) = \{\mathbf{x} \in \Omega : \psi(\mathbf{x}) < \gamma\}$, where $\gamma \in (0, 1)$.

Proof. It is obvious that

$$\begin{aligned} \int_{\Omega} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_1\|^2 \psi(\mathbf{x}) d\mathbf{x} &= \int_{\Omega} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_1\|^2 \int_0^1 \chi_{[0, \psi(\mathbf{x})]}(\gamma) d\gamma d\mathbf{x} \\ &= \int_0^1 \int_{\Omega} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_1\|^2 \chi_{[0, \psi(\mathbf{x})]}(\gamma) d\mathbf{x} d\gamma \\ &= \int_0^1 \int_{\{\mathbf{x} \in \Omega : \psi(\mathbf{x}) > \gamma\}} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_1\|^2 d\mathbf{x} d\gamma, \end{aligned}$$

where $\chi_{[0, \psi(\mathbf{x})]}(\gamma)$ is the characteristic function. Similarly,

$$\begin{aligned} \int_{\Omega} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_2\|^2 \psi(\mathbf{x}) d\mathbf{x} &= \int_{\Omega} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_2\|^2 \int_0^1 \chi_{[0, \psi(\mathbf{x})]}(\gamma) d\gamma d\mathbf{x} \\ &= \int_0^1 \int_{\Omega} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_2\|^2 \chi_{[0, \psi(\mathbf{x})]}(\gamma) d\mathbf{x} d\gamma \\ &= I - \int_0^1 \int_{\Omega \setminus \{\mathbf{x} \in \Omega : \psi(\mathbf{x}) > \gamma\}} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_2\|^2 d\mathbf{x} d\gamma, \end{aligned}$$

where $I = \int_{\Omega} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_2\|^2 d\mathbf{x}$ is independent of ψ . Using the coarea formula and the constraint $0 \leq \psi \leq 1$, we have

$$\int_{\Omega} \|\nabla \psi(\mathbf{x})\| d\mathbf{x} = \int_0^1 \text{Area}(\{\mathbf{x} \in \Omega : \psi(\mathbf{x}) > \gamma\}) d\gamma,$$

where $\text{Area}(\{\mathbf{x} \in \Omega : \psi(\mathbf{x}) > \gamma\})$ denotes the surface area of $\{\mathbf{x} \in \Omega : \psi(\mathbf{x}) > \gamma\}$.

Let $\Omega_1(\gamma) = \{\mathbf{x} \in \Omega : \psi(\mathbf{x}) > \gamma\}$, $\Omega_2(\gamma) = \{\mathbf{x} \in \Omega : \psi(\mathbf{x}) < \gamma\}$ and $C(\gamma) = \{\mathbf{x} \in \Omega : \psi(\mathbf{x}) = \gamma\}$, the energy functional can be rewritten as

$$\int_0^1 \left\{ \lambda_1 \int_{\Omega_1(\gamma)} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_1\|^2 d\mathbf{x} + \lambda_2 \int_{\Omega_2(\gamma)} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_2\|^2 d\mathbf{x} + \int_{C(\gamma)} dA \right\} d\gamma,$$

namely,

$$\min_{0 \leq \psi \leq 1} \mathcal{F}_{\text{VPCMS}}(\mathbf{c}_1, \mathbf{c}_2, \psi) = \min_{0 \leq \psi \leq 1} \int_0^1 \mathcal{E}_{\text{VPCMS}}(\mathbf{c}_1, \mathbf{c}_2, C(\gamma)) d\gamma.$$

Hence, if ψ is a minimizer of the convex optimization problem (21), then for $\gamma \in (0, 1)$, $C(\gamma)$ is a minimizer of $\mathcal{E}_{\text{VPCMS}}(\mathbf{c}_1, \mathbf{c}_2, \cdot)$. $\square$ $\square$

It is clear that the gradient descent equation of the energy functional $\mathcal{F}_{\text{VPCMS}}(\mathbf{c}_1, \mathbf{c}_2, \psi)$ is

$$\begin{cases} \frac{\partial \psi}{\partial t} = -\lambda_1 \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_1\|^2 + \lambda_2 \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_2\|^2 + \operatorname{div} \left(\frac{\nabla \psi}{\|\nabla \psi\|} \right), & \text{in } \Omega, \\ \langle \nabla \psi, \nabla \phi \rangle = 0, & \text{on } \partial\Omega, \end{cases}$$

and it has the same stationary solution with the gradient flow (20).

According to Theorem 2, we can replace the minimization of $\mathcal{E}_{\text{VPCMS}}(\mathbf{c}_1, \mathbf{c}_2, \cdot)$ by the ℓ_1 -based convex optimization problem (21), where the value of γ can be selected as 0.5 in the real computation. The split Bregman iteration algorithm can be thus applied. For completeness, we introduce this method as follows.

For ease of description, let

$$g = \lambda_1 \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_1\|^2 - \lambda_2 \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_2\|^2.$$

Then the minimization problem (21) can be rewritten as

$$\min_{0 \leq \psi \leq 1} \left\{ \int_{\Omega} g \psi d\mathbf{x} + \int_{\Omega} \|\nabla \psi\| d\mathbf{x} \right\}. \quad (22)$$

Rather than considering the minimization problem (22), we shall treat another optimization problem as follows:

$$\min_{0 \leq \psi \leq 1, \mathbf{d}} \left\{ \int_{\Omega} g \psi d\mathbf{x} + \|\mathbf{d}\|_1 \right\}, \quad \text{s.t. } \mathbf{d} = \nabla \psi, \quad (23)$$

where $\mathbf{d} = [d_x, d_y, d_z]^T$ is an introduced auxiliary variable, $\|\mathbf{d}\|_1 = \int_{\Omega} \|\mathbf{d}\| d\mathbf{x}$. The following Bregman iteration method can be efficiently used to solve the above optimization problem:

$$(\psi^{n+1}, \mathbf{d}^{n+1}) = \arg \min_{0 \leq \psi \leq 1, \mathbf{d}} \left\{ \int_{\Omega} g \psi d\mathbf{x} + \|\mathbf{d}\|_1 + \frac{\mu}{2} \int_{\Omega} \|\mathbf{d} - \nabla \psi - \mathbf{b}^n\|^2 d\mathbf{x} \right\}, \quad (24)$$

$$\mathbf{b}^{n+1} = \mathbf{b}^n + \nabla \psi^{n+1} - \mathbf{d}^{n+1}, \quad (25)$$

where the initial conditions $\mathbf{d}^0 = \mathbf{0}$, $\mathbf{b}^0 = \mathbf{0}$. In order to resolve (24), we adopt the alternating iteration technique to solve the minimization problems with respect to ψ and $\mathbf{d}$ separately, and obtain the split Bregman iteration scheme:

$$\psi^{n+1} = \arg \min_{0 \leq \psi \leq 1} \left\{ \int_{\Omega} g \psi d\mathbf{x} + \frac{\mu}{2} \int_{\Omega} \|\mathbf{d}^n - \nabla \psi - \mathbf{b}^n\|^2 d\mathbf{x} \right\}, \quad (26)$$

$$\mathbf{d}^{n+1} = \arg \min_{\mathbf{d}} \left\{ \|\mathbf{d}\|_1 + \frac{\mu}{2} \int_{\Omega} \|\mathbf{d} - \nabla \psi^{n+1} - \mathbf{b}^n\|^2 d\mathbf{x} \right\}, \quad (27)$$

$$\mathbf{b}^{n+1} = \mathbf{b}^n + \nabla \psi^{n+1} - \mathbf{d}^{n+1}. \quad (28)$$

For the optimization problem (26), the first-order optimality condition with respect to ψ is

$$\begin{cases} -\Delta \psi = \mathbf{d}^n - \mathbf{b}^n, & \text{in } \Omega, \\ \langle \mathbf{d}^n - \nabla \psi - \mathbf{b}^n, \nabla \phi \rangle = 0, & \text{on } \partial\Omega, \end{cases} \quad (29)$$

We use central difference to discretize the Laplace operator Δ and the gradient operator ∇ respectively, and use forward or backward difference to discretize the divergence operator div . In order to solve the above PDE with boundary condition (29), we first make use of the boundary condition to extend the domain of ψ^n onto the one ring neighbor R of Ω , where

R denotes the union of the one ring neighbor of the points on $\partial\Omega$. We discretize the boundary condition and obtain the following linear system of equations

$$\mathbf{W}\mathbf{p} = \mathbf{q},$$

where $\mathbf{W} \in \mathbb{R}^{m_1 \times m_2}$, $\mathbf{p} = [\psi_{i,j,k}^n] \in \mathbb{R}^{m_2}$, $\mathbf{x}_{ijk} \in R$, and $\mathbf{q} \in \mathbb{R}^{m_1}$, $m_1, m_2 \in \mathbb{Z}^+$. In general case, $m_1 < m_2$. Hence, we turn to resolve the following least square problem

$$\min_{\mathbf{p}} \frac{1}{2} \|\mathbf{W}\mathbf{p} - \mathbf{q}\|^2,$$

whose L^2 -gradient flow is as follows:

$$\frac{d\mathbf{p}(t)}{dt} = -\mathbf{W}^T(\mathbf{W}\mathbf{p} - \mathbf{q}).$$

We discretize the time variable by forward difference and obtain

$$\mathbf{p}^{(s+1)} = \mathbf{p}^{(s)} - \tau_s \mathbf{W}^T(\mathbf{W}\mathbf{p}^{(s)} - \mathbf{q}), \quad s = 0, 1, \dots, \quad (30)$$

with the initial value $\mathbf{p}^{(0)} = [\psi_{i,j,k}^{n,0}]$, $\mathbf{x}_{ijk} \in R$, $\psi_{i,j,k}^{n,0}$ is the mean value of the nearest neighbor on Ω of $\psi_{i,j,k}^n$, s is the iteration number, and τ_s is the time step. To achieve high computational efficiency, we seek the best temporal step size τ_s . We define τ_s such that

$$E(\tau) = \frac{1}{2} \left\| \mathbf{W} \left(\mathbf{p}^{(s)} - \tau \mathbf{W}^T(\mathbf{W}\mathbf{p}^{(s)} - \mathbf{q}) \right) - \mathbf{q} \right\|^2$$

is minimized. From $E'(\tau) = E'(0) + \tau E''(0) + o(\tau^2) = 0$, we obtain an approximation of τ_s as

$$\tau_s = -\frac{E'(0)}{E''(0)} = \frac{\langle \mathbf{e}_1, \mathbf{e}_2 \rangle}{\langle \mathbf{e}_2, \mathbf{e}_2 \rangle}, \quad \mathbf{e}_1 = \mathbf{W}\mathbf{p}^{(s)} - \mathbf{q}, \quad \mathbf{e}_2 = \mathbf{W}\mathbf{W}^T \mathbf{e}_1.$$

Then we update ψ^{n+1} in component form with the following scheme:

$$\psi_{i,j,k}^{n+1} = \min\{\max\{0, \omega_{i,j,k}^{n+1}\}, 1\}, \quad \mathbf{x}_{ijk} \in \Omega, \quad (31)$$

where

$$\begin{aligned} \omega_{i,j,k}^{n+1} &= \frac{1}{6} \left(\psi_{i-1,j,k}^n + \psi_{i+1,j,k}^n + \psi_{i,j-1,k}^n + \psi_{i,j+1,k}^n + \psi_{i,j,k-1}^n + \psi_{i,j,k+1}^n - \frac{g_{i,j,k}^n}{\mu} - \rho_{i,j,k}^n \right), \\ g_{i,j,k}^n &= \left(\lambda_1 \|\mathbf{F} - \mathbf{c}_1^n\|^2 - \lambda_2 \|\mathbf{F} - \mathbf{c}_2^n\|^2 \right)_{i,j,k}, \\ \rho_{i,j,k}^n &= (\text{div}(\mathbf{d}^n - \mathbf{b}^n))_{i,j,k}. \end{aligned}$$

In order to solve the optimization problem (27), we first investigate the following minimization problem

$$\min_{\mathbf{d} \in \mathbb{R}^M} \left\{ \|\mathbf{d}\| + \frac{\mu}{2} \|\mathbf{d} - \mathbf{b}\|^2 \right\}, \quad (32)$$

where $\mathbf{d} = [d_1, \dots, d_M]^T$ and $\mathbf{b} = [b_1, \dots, b_M]^T$, that is,

$$\min_{d_1, \dots, d_M \in \mathbb{R}} \left\{ \left(\sum_{m=1}^M d_m^2 \right)^{\frac{1}{2}} + \frac{\mu}{2} \sum_{m=1}^M |d_m - b_m|^2 \right\}.$$

Let $G(\mathbf{d}) := \left(\sum_{m=1}^M d_m^2 \right)^{\frac{1}{2}}$, $F(\mathbf{d}) := \frac{\mu}{2} \sum_{m=1}^M |d_m - b_m|^2$, and $E(\mathbf{d}) := G(\mathbf{d}) + F(\mathbf{d})$. It is obvious that the subdifferentials of F and G at $\mathbf{0} \in \mathbb{R}^M$ are

pdfelement

The Trial Version

$$\partial G(\mathbf{0}) = \{\mathbf{g} \in \mathbb{R}^M \mid \|\mathbf{g}\| \leq 1\}.$$

Consequently, $\mathbf{0} \in \partial E(\mathbf{0})$ if and only if $\|\mathbf{b}\| \leq \frac{1}{\mu}$. Suppose that $E(\mathbf{d})$ achieves the minimum. If $\|\mathbf{b}\| \leq \frac{1}{\mu}$, then $\mathbf{d} = \mathbf{0}$. Otherwise, we have $\|\mathbf{d}\| > 0$. Moreover,

$$\frac{d_m}{\|\mathbf{d}\|} + \mu(d_m - b_m) = 0, \quad m = 1, \dots, M.$$

It follows that

$$\begin{cases} d_2 b_1 = d_1 b_2, & d_3 b_1 = d_1 b_3, & \dots, & d_M b_1 = d_1 b_M, \\ & d_3 b_2 = d_2 b_3, & \dots, & d_M b_2 = d_2 b_M, \\ & & \ddots & \vdots \\ & & & d_M b_{M-1} = d_{M-1} b_M. \end{cases}$$

There exists a real number r such that $d_m = r b_m$ ($m = 1, \dots, M$). Consequently,

$$E(\mathbf{d}) := E(r) = |r| \|\mathbf{b}\| + \frac{\mu}{2} |r - 1|^2 \|\mathbf{b}\|^2.$$

It is obvious that $r = \text{shrink}(1, \frac{1}{\mu \|\mathbf{b}\|})$ is the unique point such that $E(r)$ achieves its minimum. Here,

$$\text{shrink}(x, \lambda) = \frac{x}{|x|} \max\{|x| - \lambda, 0\}, \quad x \in \mathbb{R}.$$

Let $r' := \|\mathbf{b}\| r = \max\{\|\mathbf{b}\| - \frac{1}{\mu}, 0\}$, we conclude that

$$d_m = \frac{r' b_m}{\|\mathbf{b}\|} = \frac{b_m}{\|\mathbf{b}\|} \max\{\|\mathbf{b}\| - \frac{1}{\mu}, 0\}, \quad m = 1, \dots, M.$$

The above formula is also valid when $\|\mathbf{b}\| \leq \frac{1}{\mu}$, provided we interpret d_m , $m = 1, \dots, M$, as 0 when $r' = 0$. Therefore, $E(\mathbf{d})$ achieves the minimum if and only if

$$\mathbf{d} = \frac{\mathbf{b}}{\|\mathbf{b}\|} \max\{\|\mathbf{b}\| - \frac{1}{\mu}, 0\} := \text{shrink}(\mathbf{b}, \frac{1}{\mu}). \quad (33)$$

Let $M = 3$ and $\mathbf{b} = \nabla \psi^{n+1} + \mathbf{b}^n$, we update $\mathbf{d}^{n+1}$ in (27) by the formula as follows:

$$\mathbf{d}^{n+1} = \text{shrink}(\nabla \psi^{n+1} + \mathbf{b}^n, \frac{1}{\mu}). \quad (34)$$

In summary, we show the split Bregman iteration algorithm for the proposed model (15) as follows.

Algorithm 1 (Split Bregman Iteration for Minimizing the Energy Model (15)).

(1) Given the initial segmentation Ω_1 and Ω_2 , the threshold $0 < \epsilon_0 \ll 1$, the iteration number $N_0 > 0$. Let

$$\psi^0(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \Omega_1, \\ 0, & \mathbf{x} \in \Omega_2, \\ 0.5, & \text{others,} \end{cases}$$

$\mathbf{d}^0 = \mathbf{0}$, $\mathbf{b}^0 = \mathbf{0}$. Set $n := 0$.

(2) Update $\mathbf{c}_1^n$ and $\mathbf{c}_2^n$:

$$c_{\alpha}^n = \frac{\int_{\Omega} f_{\alpha}(\mathbf{x}) H_l(\psi^n - 0.5) d\mathbf{x}}{\int_{\Omega} H_l(\psi^n - 0.5) d\mathbf{x}}, \quad l = 1, 2, \quad \alpha = 1, \dots, \Pi.$$

(3) Extend the domain of ψ^n onto the one ring neighbor R of Ω by (30), then update ψ^{n+1} by (31), and compute $\varepsilon_n = \|\psi^{n+1} - \psi^n\|_2$. If $\varepsilon_n \leq \epsilon_0$ or $n + 1 \geq N_0$, stop the iteration, set $\Omega_1 = \{\mathbf{x} \in \Omega : \psi(\mathbf{x}) > 0.5\}$, $\Omega_2 = \{\mathbf{x} \in \Omega : \psi(\mathbf{x}) < 0.5\}$; otherwise, go to the next step.

(4) Update $\mathbf{d}^{n+1}$ and $\mathbf{b}^{n+1}$ by (34) and (28) respectively. Set $n := n + 1$, return to (2).

4.3. Multi-phase segmentation algorithm

When the original shape is segmented into more than two regions, two or more level set functions are needed to represent the segmentation regions, and the corresponding energy functional can be minimized with the similar method as the two-phase segmentation. In the following, we introduce the two-phase segmentation algorithm briefly.

Assume that $\psi_1(\mathbf{x})$ and $\psi_2(\mathbf{x})$ are two level set functions defined on Ω , let

$$\begin{cases} \Omega_1 = \{\mathbf{x} \in \Omega : \psi_1(\mathbf{x}) > 0, \psi_2(\mathbf{x}) > 0\}, \\ \Omega_2 = \{\mathbf{x} \in \Omega : \psi_1(\mathbf{x}) > 0, \psi_2(\mathbf{x}) < 0\}, \\ \Omega_3 = \{\mathbf{x} \in \Omega : \psi_1(\mathbf{x}) < 0, \psi_2(\mathbf{x}) > 0\}, \\ \Omega_4 = \{\mathbf{x} \in \Omega : \psi_1(\mathbf{x}) < 0, \psi_2(\mathbf{x}) < 0\}, \end{cases} \quad (35)$$

then ψ_1 and ψ_2 segment Ω into four regions $\{\Omega_l\}_{l=1}^4$, and the segmentation contour is $C = \{\mathbf{x} \in \Omega : \psi_1(\mathbf{x}) = 0 \text{ or } \psi_2(\mathbf{x}) = 0\}$. The corresponding variational model is as follows:

$$\begin{aligned} \tilde{\mathcal{E}}_{\text{VPCMS}}(\mathbf{C}, \boldsymbol{\psi}) = & \lambda_1 \int_{\Omega} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_1\|^2 M_1(\boldsymbol{\psi}) d\mathbf{x} + \lambda_2 \int_{\Omega} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_2\|^2 M_2(\boldsymbol{\psi}) d\mathbf{x} \\ & + \lambda_3 \int_{\Omega} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_3\|^2 M_3(\boldsymbol{\psi}) d\mathbf{x} + \lambda_4 \int_{\Omega} \|\mathbf{F}(\mathbf{x}) - \mathbf{c}_4\|^2 M_4(\boldsymbol{\psi}) d\mathbf{x} \\ & + \int_{\Omega} \|\nabla H(\psi_1)\| d\mathbf{x} + \int_{\Omega} \|\nabla H(\psi_2)\| d\mathbf{x} \\ = & \lambda_1 \sum_{\alpha=1}^{\Pi} \int_{\Omega} |f_{\alpha}(\mathbf{x}) - c_{1\alpha}|^2 M_1(\boldsymbol{\psi}) d\mathbf{x} + \lambda_2 \sum_{\alpha=1}^{\Pi} \int_{\Omega} |f_{\alpha}(\mathbf{x}) - c_{2\alpha}|^2 M_2(\boldsymbol{\psi}) d\mathbf{x} \\ & + \lambda_3 \sum_{\alpha=1}^{\Pi} \int_{\Omega} |f_{\alpha}(\mathbf{x}) - c_{3\alpha}|^2 M_3(\boldsymbol{\psi}) d\mathbf{x} + \lambda_4 \sum_{\alpha=1}^{\Pi} \int_{\Omega} |f_{\alpha}(\mathbf{x}) - c_{4\alpha}|^2 M_4(\boldsymbol{\psi}) d\mathbf{x} \\ & + \int_{\Omega} \|\nabla H(\psi_1)\| d\mathbf{x} + \int_{\Omega} \|\nabla H(\psi_2)\| d\mathbf{x}, \end{aligned} \quad (36)$$

where $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \mathbf{c}_4]$, $\boldsymbol{\psi} = [\psi_1, \psi_2]$, and

$$\begin{aligned} M_1(\boldsymbol{\psi}) &= H(\psi_1)H(\psi_2), & M_2(\boldsymbol{\psi}) &= H(\psi_1)(1 - H(\psi_2)), \\ M_3(\boldsymbol{\psi}) &= (1 - H(\psi_1))H(\psi_2), & M_4(\boldsymbol{\psi}) &= (1 - H(\psi_1))(1 - H(\psi_2)). \end{aligned}$$

We adopt the alternating iterative method to compute the minimum solution of the energy functional (36). In each iteration step, we minimize $\tilde{\mathcal{E}}_{\text{VPCMS}}(\mathbf{C}, \boldsymbol{\psi})$ with respect to $\mathbf{C}$ and $\boldsymbol{\psi}$, respectively. We first fix $\boldsymbol{\psi}$, minimize $\tilde{\mathcal{E}}_{\text{VPCMS}}(\cdot, \boldsymbol{\psi})$ and obtain

$$c_{l\alpha} = \frac{\int_{\Omega} f_{\alpha}(\mathbf{x}) M_l(\boldsymbol{\psi}) d\mathbf{x}}{\int_{\Omega} M_l(\boldsymbol{\psi}) d\mathbf{x}} = \frac{\int_{\Omega} f_{\alpha}(\mathbf{x}) M_l(\boldsymbol{\psi}) d\mathbf{x}}{|\Omega_l|}, \quad l = 1, 2, 3, 4, \quad \alpha = 1, \dots, \Pi. \quad (37)$$

Then we minimize (36) in terms of $\boldsymbol{\psi}$ by fixed $\mathbf{C}$. According to Theorem 2, a global minimizer for $\tilde{\mathcal{E}}_{\text{VPCMS}}(\mathbf{C}, \cdot)$ can be obtained by solving the following convex optimization problems

$$\min_{0 \leq \psi_1 \leq 1} \left\{ \int_{\Omega} g_1 \psi_1 d\mathbf{x} + \int_{\Omega} \|\nabla \psi_1\| d\mathbf{x} \right\}, \quad (38)$$

$$\min_{0 \leq \psi_2 \leq 1} \left\{ \int_{\Omega} g_2 \psi_2 d\mathbf{x} + \int_{\Omega} \|\nabla \psi_2\| d\mathbf{x} \right\}, \quad (39)$$

where

$$g_1 = \left(\lambda_1 \|\mathbf{F} - \mathbf{c}_1\|^2 - \lambda_3 \|\mathbf{F} - \mathbf{c}_3\|^2 \right) \psi_2 + \left(\lambda_2 \|\mathbf{F} - \mathbf{c}_2\|^2 - \lambda_4 \|\mathbf{F} - \mathbf{c}_4\|^2 \right) (1 - \psi_2), \quad (40)$$

$$g_2 = \left(\lambda_1 \|\mathbf{F} - \mathbf{c}_1\|^2 - \lambda_2 \|\mathbf{F} - \mathbf{c}_2\|^2 \right) \psi_1 + \left(\lambda_3 \|\mathbf{F} - \mathbf{c}_3\|^2 - \lambda_4 \|\mathbf{F} - \mathbf{c}_4\|^2 \right) (1 - \psi_1). \quad (41)$$

In order to resolve (38) and (39) with the split Bregman Iteration, we assume that g_1 and g_2 are known, and they can be calculated with ψ_1^n and ψ_2^n which have been computed in the previous step. Then we consider the following optimization problem

$$\min_{0 \leq \psi_p \leq 1, \mathbf{d}_p} \left\{ \int_{\Omega} g_p \psi_p d\mathbf{x} + \int_{\Omega} \|\mathbf{d}_p\| d\mathbf{x} \right\}, \quad \text{s.t.} \quad \mathbf{d}_p = \nabla \psi_p, \quad p = 1, 2, \quad (42)$$

where $\mathbf{d}_p$, $p = 1, 2$, are introduced auxiliary variables. For $p = 1, 2$, the following split Bregman iteration method can be used to solve the above minimization problem:

$$\psi_p^{n+1} = \arg \min_{0 \leq \psi_p \leq 1} \left\{ \int_{\Omega} g_p \psi_p d\mathbf{x} + \frac{\mu}{2} \int_{\Omega} \|\mathbf{d}_p^n - \nabla \psi_p - \mathbf{b}_p^n\|^2 d\mathbf{x} \right\}, \quad (43)$$

$$\mathbf{d}_p^{n+1} = \arg \min_{\mathbf{d}_p} \left\{ \int_{\Omega} \|\mathbf{d}_p\| d\mathbf{x} + \frac{\mu}{2} \int_{\Omega} \|\mathbf{d}_p - \nabla \psi_p^{n+1} - \mathbf{b}_p^n\|^2 d\mathbf{x} \right\}, \quad (44)$$

$$\mathbf{b}_p^{n+1} = \mathbf{b}_p^n + \nabla \psi_p^{n+1} - \mathbf{d}_p^{n+1}, \quad (45)$$

with the initial values $\mathbf{d}_p^0 = \mathbf{0}$, $\mathbf{b}_p^0 = \mathbf{0}$, $p = 1, 2$. We can solve (43) and (44) in the similar way as we solve (26) and (27). In the following, we summarise the split Bregman iteration method for minimizing the energy functional (36).

Algorithm 2 (Split Bregman Iteration for Minimizing the Energy Functional (36)).

(1) Given the initial segmentation $\{\Omega_l\}_{l=1}^4$, the iteration number $N_0 > 0$. Let

$$\psi_1^0(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \Omega_1 \cup \Omega_2, \\ 0, & \mathbf{x} \in \Omega_3 \cup \Omega_4, \\ 0.5, & \text{others,} \end{cases} \quad \psi_2^0(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \Omega_1 \cup \Omega_3, \\ 0, & \mathbf{x} \in \Omega_2 \cup \Omega_4, \\ 0.5, & \text{others,} \end{cases}$$

$\mathbf{d}_p^0 = \mathbf{0}$, $\mathbf{b}_p^0 = \mathbf{0}$, $p = 1, 2$. Set $n := 0$.

(2) Update $\mathbf{C}^n$:

$$c_{\alpha}^n = \frac{\int_{\Omega} f_{\alpha}(\mathbf{x}) M_l(\psi_1^n - 0.5, \psi_2^n - 0.5) d\mathbf{x}}{\int_{\Omega} M_l(\psi_1^n - 0.5, \psi_2^n - 0.5) d\mathbf{x}}, \quad l = 1, 2, 3, 4, \quad \alpha = 1, \dots, \Pi.$$

(3) Compute g_1, g_2 by (40) and (41) with ψ_1^n, ψ_2^n . For $p = 1, 2$, update $\psi_p^{n+1}, \mathbf{d}_p^{n+1}$ and $\mathbf{b}_p^{n+1}$ by solving (43)–(45).

(4) Set $n := n + 1$. If $n < N_0$, return to (2); otherwise, stop the iteration, set

$$\Omega_1 = \{\mathbf{x} \in \Omega : \psi_1(\mathbf{x}) > 0.5, \psi_2(\mathbf{x}) > 0.5\}, \Omega_2 = \{\mathbf{x} \in \Omega : \psi_1(\mathbf{x}) > 0.5, \psi_2(\mathbf{x}) < 0.5\},$$

$$\Omega_3 = \{\mathbf{x} \in \Omega : \psi_1(\mathbf{x}) < 0.5, \psi_2(\mathbf{x}) > 0.5\}, \Omega_4 = \{\mathbf{x} \in \Omega : \psi_1(\mathbf{x}) < 0.5, \psi_2(\mathbf{x}) < 0.5\}.$$

5. Results and discussion

In this section, several numerical experiments are tested using our point cloud surface segmentation method. We aim to segment the shape into components that are intuitively meaningful and free of noise. All the results were generated using a computer with Intel core 2.83 GHz CPU, 4 threads, and 8 GB of memory. Table 1 shows a summary of the computation of the first ten eigenfunctions for LBO in both volumetric and surface forms. In the shape segmentation stage, we choose the uniform parameters $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1.0$. The iteration terminates when the segmentation result becomes stable, and the runtime are shown in Table 2. Note that our volumetric eigenfunctions and VPCMS segmentation results are originally restored in volume data format. In order to show the difference of segmentation performance between the volumetric and surface LBO, we map the VPCMS segmentation result onto surface via voxel-to-vertex correspondence.

Firstly, we segment the point cloud surface Bunny, Horse and Octopus using only one level set function. We adapt Algorithm 1 to segment each eigenfunction illustrated in Figs. 1(a), 2(a) and 3(a), that is to say, $\Pi = 1$ in the VPCMS model. The segmentation results are shown in Figs. 1(c), 2(c) and 3(c), and the two subregions are shown in different colors.

Table 1

A summary of the computation time of the first ten eigenfunctions for LBO in both volumetric and surface forms. (Time unit: Second).

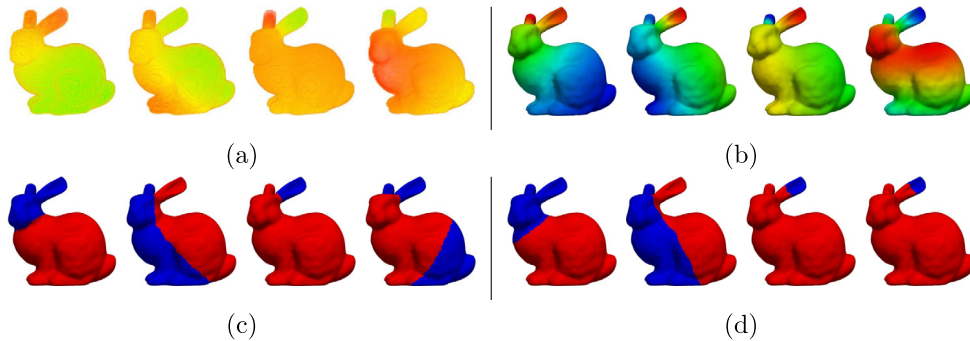
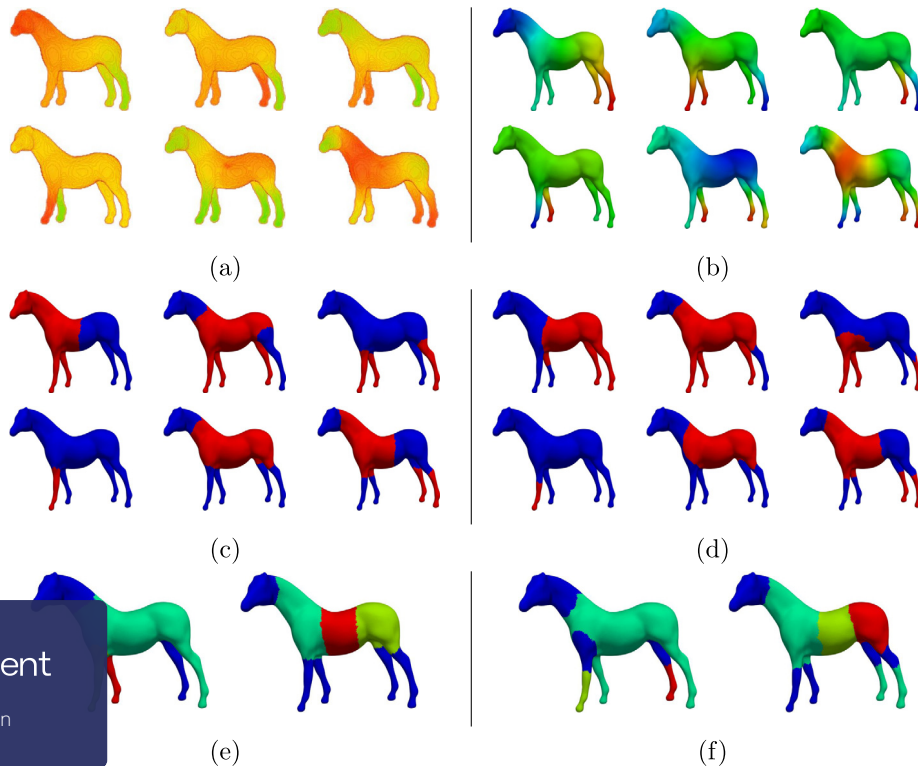
Model	Point cloud surface (N_P, N_G)	Triangle mesh surface (N_V, N_T)	T_E (Volumetric / Surface)
Bunny	(14,076, 38,888)	(14,076, 28,148)	1,919.56 / 131.97
Horse	(15,698, 37,383)	(15,698, 31,392)	2,432.90 / 416.81
Octopus	(14,870, 24,464)	(14,870, 29,736)	3,694.72 / 284.68
Ant	(6,372, 17,099)	(6,372, 12,740)	2,035.33 / 26.49
Armadillo	(25,220, 67,106)	(25,220, 50,436)	8,480.80 / 1,222.88
Human	(10,041, 26,255)	(10,041, 20,078)	3,327.55 / 40.83
Bust	(25,227, 67,185)	(25,227, 50,450)	8,296.82 / 199.51
Bird	(6,470, 16,623)	(6,470, 12,936)	2,661.73 / 21.89
Teddy	(13,822, 37,579)	(13,822, 27,640)	4,237.49 / 74.08

Note: N_P – particle number; N_G – grid number; N_V – vertex number; N_T – triangular element number; T_E – time for eigenfunction computation.

Table 2

Statistics of the computation of segmentation results in Figs. 1–3. (Time unit: Second).

Segmentation Type		Two-region									Four-region							
Modes		1	2	3	4	5	6	7	8	2-3	2-3	3-4	5-6	6-7	7-8	4-6	5-7	6-8
Bunny	VPCMS	2.39	2.39	2.41	2.39	-	-	-	-	-	-	-	-	-	-	-	-	-
	HBECVT	0.26	0.28	0.27	0.27	-	-	-	-	-	-	-	-	-	-	-	-	-
Horse	VPCMS	3.15	3.08	3.08	3.12	3.08	3.05	-	-	-	-	6.35	6.42	-	-	-	-	-
	HBECVT	0.29	0.30	0.30	0.30	0.30	0.29	-	-	-	-	0.29	0.30	-	-	-	-	-
Octopus	VPCMS	2.28	2.28	2.39	2.25	2.31	2.38	2.28	2.25	2.23	4.72	-	4.64	4.62	4.63	4.74	4.71	4.72
	HBECVT	0.29	0.28	0.30	0.28	0.28	0.28	0.28	0.29	0.25	0.25	-	0.28	0.28	0.29	0.28	0.29	0.28

**Fig. 1.** Bunny model. (a, b) The first four non-constant volumetric and surface eigenfunctions of LBO; (c, d) two-region segmentation results from Modes 1–4 in (a) and (b), respectively.**Fig. 2.** Horse model. (a, b) The first six non-constant volumetric and surface eigenfunctions of LBO; (c, d) two-region segmentation results from Modes 1–6 in (a) and (b), respectively; (e, f) four-region segmentation results from Modes 3–4 and 5–6 in (a) and (b), respectively.

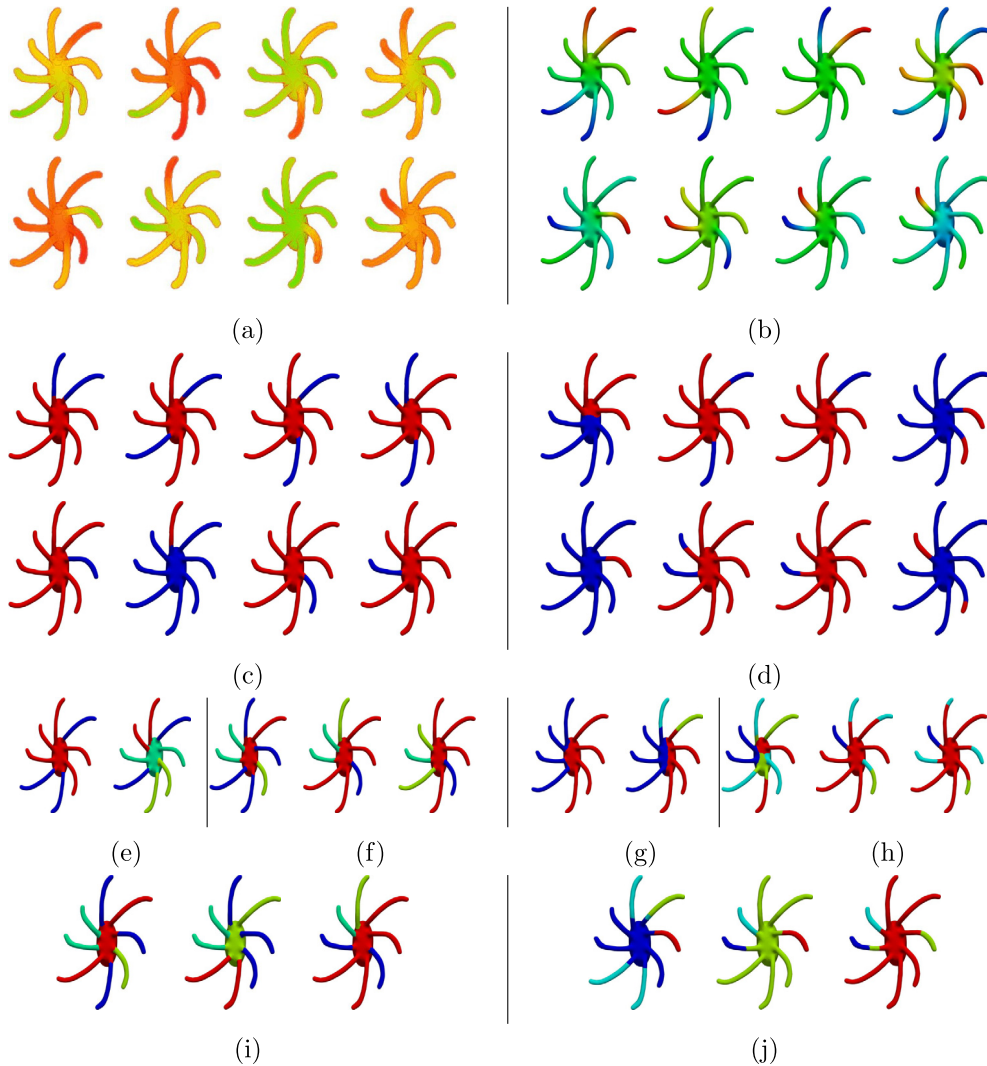


Fig. 3. Octopus model. (a, b) The first eight non-constant volumetric and surface eigenfunctions of LBO; (c, d) two-region segmentation results from Modes 1–8 in (a) and (b), respectively; (e, g) two-region and four-region segmentation results from Modes 2–3 in (a) and (b), respectively; (f, h) four-region segmentation results from Modes 5–6, 6–7 and 7–8 in (a) and (b), respectively; and (i, j) four-region segmentation results from Modes 4–6, 5–7 and 6–8 in (a) and (b), respectively.

It is easy to see that the segmentation results are consistent with the distribution of the corresponding eigenfunctions. Therefore, according to different segmentation purposes, we can choose different eigenfunctions, so as to achieve the goal of segmenting point cloud surface.

Then, we consider segmenting the point cloud surface Horse and Octopus with two level set functions. The point cloud surface can be divided into four subregions by the zero level set of these two level set functions. Since the nodal lines (zero sets of the function) of the α^{th} eigenfunction subdivide the domain into α subdomains at the most Courant and Hilbert (1953), we need to combine multiple eigenfunctions for lower modes to get the ideal segmentation results. As illustrated in Fig. 2(e), we use Modes 3–4 and 5–6 to separate the Horse model into four regions. For the Octopus model, we choose to combine two or three eigenfunctions to divide, and the segmentation results show that our VPCMS model can get closed areas. The segmented subregions using different combinations of eigenfunctions are not the same, but all of them can split the octopus model into four meaningful regions, as shown in Fig. 3(f) and (i).

In Fig. 3(e), we combine two different eigenfunctions, namely, let $\mathbf{F} = [f_2, f_3]^T$ in the VPCMS model, and segment the octopus model with one and two level set functions respectively. The segmentation result of the former (see Fig. 3(e) left) is different from the segmentation results with any one eigenfunction used individually, and the segmentation effect in the joint of tentacles and body is better; the latter derives four segmented regions displayed by different color (see Fig. 3(e) right). Both methods can separate some antennas completely from the whole model, which can be used for subsequent processing.

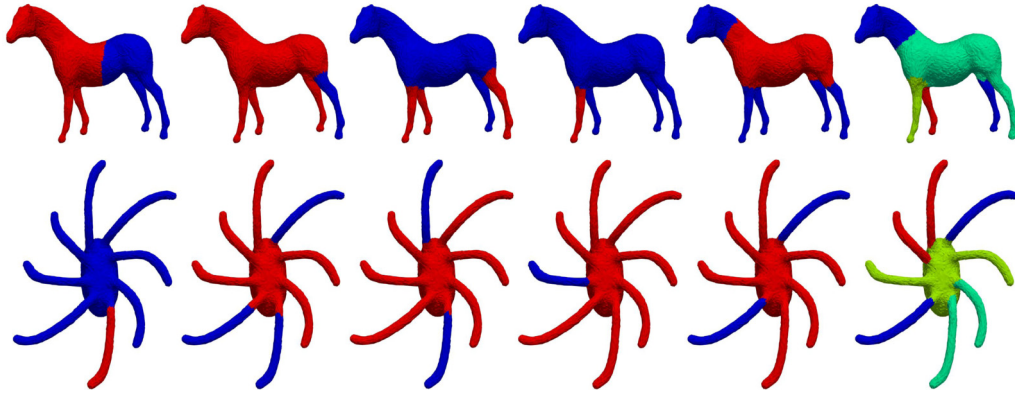


Fig. 4. The segmentation results from applying our algorithm to noisy Horse and Octopus models. The first five are for two-region segmentation, and the last for four-region segmentation.

Various eigenfunction-based surface segmentation schemes have been developed. To reveal the differences of the LBO between the volumetric form and the surface form, we adopt the centroidal Voronoi tessellation based surface segmentation method (HBECVT) Hu et al. (2016) to compute the segmentation results from the eigenfunctions of LBO in the surface form. Triangle meshes are generated for the surface eigenfunction computation which utilizes Loop subdivision basis functions. Figs. 1–3 (right column) show surface eigenfunctions of the LBO and the corresponding segmentation results. It is easy to see that our volumetric eigenfunction based technique yields more accurate segmentation results than the surface eigenfunction based method. This is because in addition to surface information of the shape, volumetric eigenfunctions also reveal the interior information, which can be useful for shape segmentation. In Fig. 1(d), the first two segmentation results are similar to those in Fig. 1(c), while the last two miss the connection regions of ears and body which are detected in Fig. 1(c). In Fig. 2(f), the legs of the horse are subdivided into two smaller patches, which are inconsistent with the divisions in Fig. 2(e). For the Octopus model, concave creases at the junctions of antennas and body are completely ignored by the surface eigenfunctions of LBO, and the antennas surrounded by them can not be segmented as a patch, see Fig. 3 (d), (g), (h) and (j).

In addition, we also tested our method on some noisy input, for example, Horse and Octopus. The distributions of the volumetric eigenfunctions are similar to those of clean data, which is not shown to make this section concise. As shown in Fig. 4, the segmentation results match the previous results in Fig. 2 and Fig. 3 well.

It is worth mentioning that both of the above methods use eigenfunctions of the LBO as basis for segmentation. Differently, our algorithm applies the Mumford-Shah model to the Π -dimensional vector valued function. In other words, our segmentation part is equivalent to the vectorial 3D image segmentation, while HBECVT directly clusters elements in the Π -dimensional eigenspace. In addition, HBECVT needs the cluster number and seed elements as input, while our method does not.

To show more general results, we tested our method on models sampled from triangular surfaces in the Princeton Segmentation Benchmark (Chen et al., 2009). The segmentation results can be evaluated quantitatively using the Princeton benchmark and software (Chen et al., 2009), by comparing them with the ground truth and results from the baseline methods respectively. These seven automatic segmentation algorithms include the Core Extraction, Fitting Primitives, K-Means, Normalized Cuts, Randomized Cuts, Random Walks and Shape Diameter Function, please refer to the cited papers in Chen et al. (2009) for details. There are four metrics (Chen et al., 2009) to measure the quality of segmentation results:

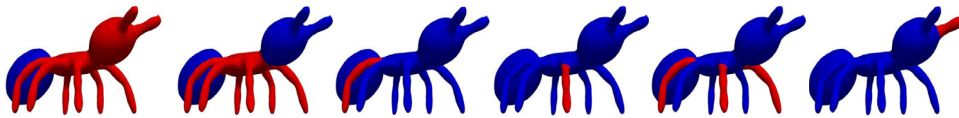
1. The *Cut Discrepancy* is the sum of distance between points along the cuts in the computed segmentation to the closest cuts in the ground truth segmentation, and vice-versa;
2. The *Hamming Distance* measures the overall difference between the patches in different segmentation results;
3. The *Rand Index* measures the likelihood that a random pair of elements are in the same patch or not in different segmentation results; and
4. The *Consistency Error* measures the hierarchical similarities and differences of segmentation results.

Table 3 shows evaluations of our segmentation results compared to the ground truth and seven other algorithms in the Princeton Segmentation Benchmark. Table 3 shows a different evaluation metric computed with respect to all eight baseline results and averaged across the randomly picked 26 shapes from the benchmark. In all cases, smaller metrics represent closer performance to the ground truth baseline. From Table 3, we can observe that all these four evaluation metrics are remarkably consistent with one another.

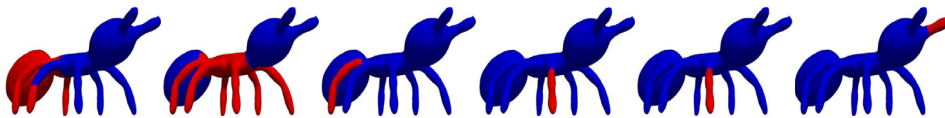
To make it brief, LBO eigenfunctions of these models are not displayed here. Instead, we show the two-region and four-region segmentation results against the HBECVT results for the first six models in Figs. 5, 6 respectively. For completeness, the segmentation results of the remaining 20 models are presented in Fig. 7. From these results, we can observe that if

Table 3
Analysis of segmentation quality according to the baseline results in the Princeton Segmentation Benchmark (Chen et al., 2009).

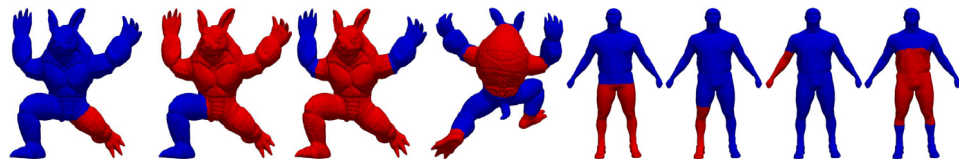
Metric \ Baseline	HumanGen	CoreExtra	FitPrim	Kmeans	NormCuts	RandCuts	RandWalks	ShapeDiam
Cut Discrepancy	0.330865584	0.421842289	0.438121632	0.308286684	0.345570105	0.418750294	0.465868474	0.471960053
Hamming Distance	0.144822068	0.208973684	0.261503616	0.208584242	0.179833526	0.164623905	0.248335737	0.208679058
Rand Index	0.194581089	0.305990632	0.276338337	0.251625811	0.228947658	0.228567489	0.319435295	0.261956337
Consistency Error	0.095641805	0.142385821	0.198597905	0.180318232	0.162910102	0.122551984	0.198289799	0.134400669



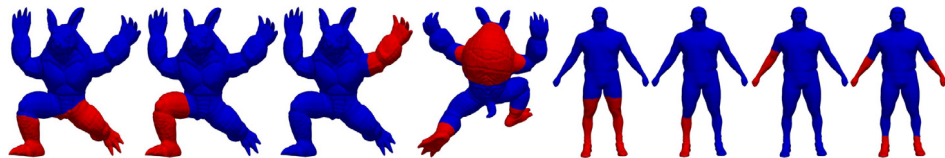
(a) Results from volumetric eigenfunctions



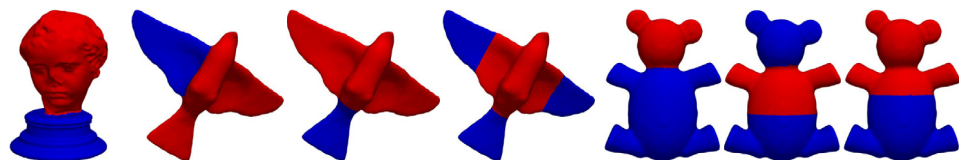
(b) Results from surface eigenfunctions



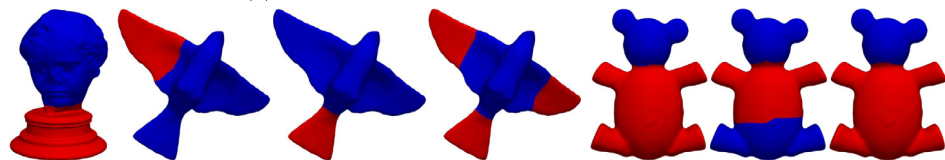
(c) Results from volumetric eigenfunctions



(d) Results from surface eigenfunctions



(e) Results from volumetric eigenfunctions



(f) Results from surface eigenfunctions

the eigenfunctions are well distributed or if the eigenfunctions can capture geometric features of the shape, then the shape will be segmented into intuitively meaningful components. As shown in Figs. 5, 6, our method always yields satisfactory segmentation results and is comparable with HBECVT. Note that the difference between results of volumetric and surface

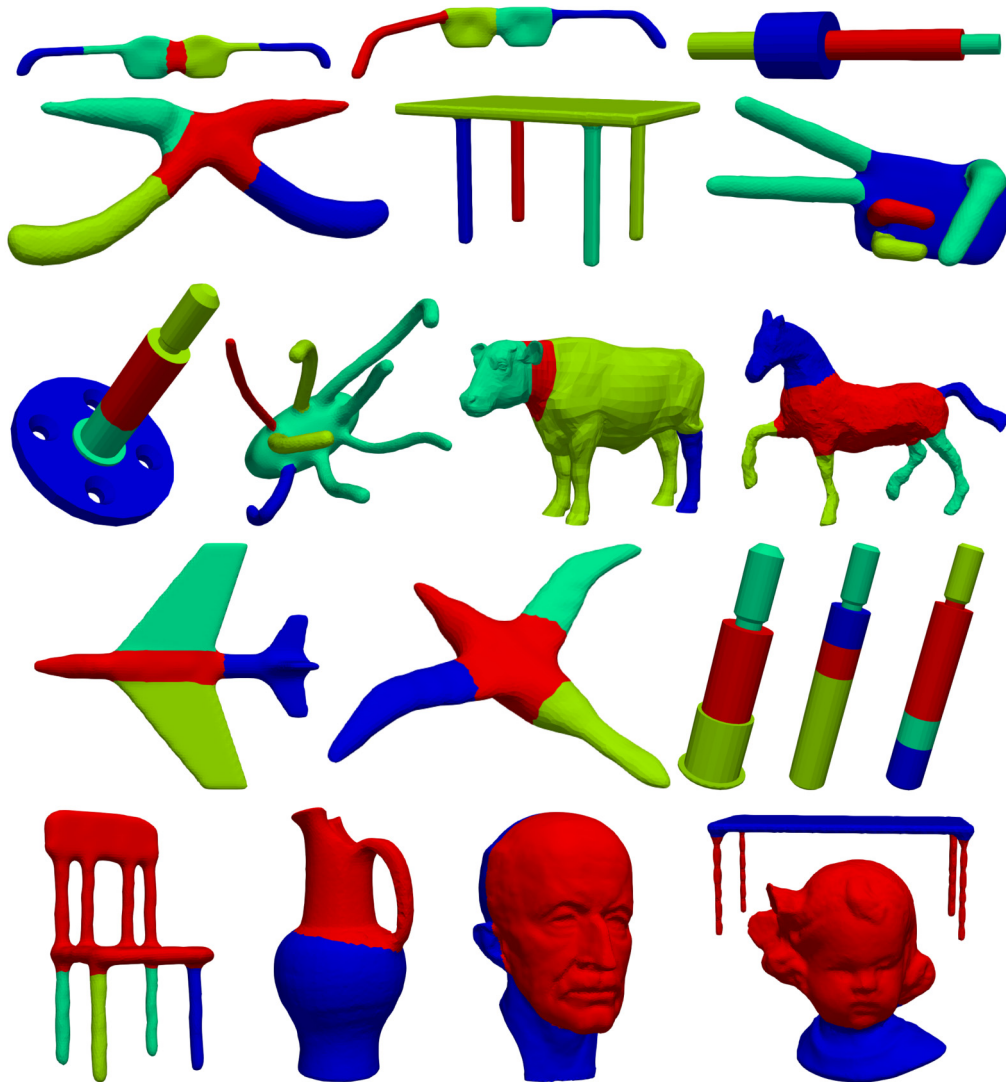


Fig. 7. Segmentation results of models in the Princeton Segmentation Benchmark (Chen et al., 2009).

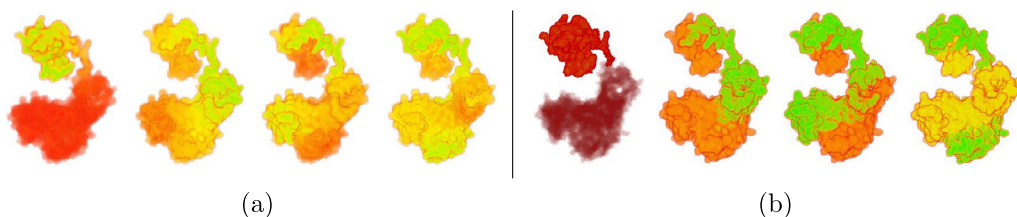


Fig. 8. 2BPF model. (a) The first four non-constant volumetric eigenfunctions of LBO; and (b) two-region segmentation result from Modes 1–4, respectively.

References

- ARPACK, 2009. Arnoldi package. <http://www.caam.rice.edu/software/ARPACK/>.
- Besl, P.J., Jain, R.C., 1988. Segmentation through variable-order surface fitting. *IEEE Trans. Pattern Anal. Mach. Intell.* 10 (2), 167–192.
- Bizjak, M., 2016. The segmentation of a point cloud using locally fitted surfaces. In: 18th Mediterranean Electrotechnical Conference (MELECON), pp. 1–6.
- Chan, T.F., Nikolova, M., 2006. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.* 66 (5), 1632–1648.
- Chan, T.F., Vese, L.A., 2001. Active contours without edges. *IEEE Trans. Image Process.* 10 (2), 266–277.
- Cecchin, P., Trassoudaine, L., Alizon, J., 1997. Segmentation of range images into planar regions. In: *Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling, NRC '97*. IEEE Computer Society, Washington, DC, USA, p. 156.
- Chen, X., Golovinskiy, A., Funkhouser, T., 2009. A benchmark for 3D mesh segmentation. *ACM Trans. Graph. (Proc. SIGGRAPH)* 28 (3), 1–12.
- Chen, Y.H., Liu, C.Y., 1997. Robust segmentation of CMM data based on NURBS. *Int. J. Adv. Manuf. Technol.* 13 (8), 530–534.
- Courant, R., Hilbert, D., 1953. *Methods of Mathematical Physics, Vol. I*. Interscience, New York.
- Fan, T.-J., Medioni, G., Nevatia, R., 1987. Segmented descriptions of 3-D surfaces. *IEEE J. Robot. Autom.* 3 (6), 527–538.
- Goldstein, T., Bresson, X., Osher, S., 2010. Geometric applications of the split Bregman method: segmentation and surface reconstruction. *J. Sci. Comput.* 45 (1–3), 272–293.
- Goldstein, T., Osher, S., 2009. The split Bregman method for L1-regularized problems. *SIAM J. Imaging Sci.* 2 (2), 323–343.
- Hoffman, R., Jain, A.K., 1987. Segmentation and classification of range images. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI-9* (5), 608–620.
- Hu, K., Zhang, Y.J., Li, X., Xu, G., 2016. Feature-aligned surface parameterization using secondary Laplace operator and loop subdivision. *Proc. Eng.* 163, 186–198.
- Jia, R.-Q., Zhao, H., Zhao, W., 2009. Convergence analysis of the Bregman method for the variational model of image denoising. *Appl. Comput. Harmon. Anal.* 27 (3), 367–379.
- Kalogerakis, E., Averkiou, M., Maji, S., Chaudhuri, S., 2017. 3D shape segmentation with projective convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6630–6639.
- Kalogerakis, E., Hertzmann, A., Singh, K., 2010. Learning 3D mesh segmentation and labeling. In: *ACM Siggraph*, pp. 1–12.
- Li, X.S., 2005. An overview of SuperLU: algorithms, implementation, and user interface. *ACM Trans. Math. Softw.* 31 (3), 302–325.
- Li, X.S., Demmel, J.W., Gilbert, J.R., Grigori, L., Sao, P., Shao, M., Yamazaki, I., 2011. SuperLU users' guide. Technical Report (LBNL-44289) <http://crd.lbl.gov/~xiaoye/SuperLU/>. Last update.
- Liu, T., Chen, M., Lu, B., 2015. Parameterization for molecular Gaussian surface and a comparison study of surface mesh generation. *J. Mol. Model.* 21 (5), 113–126.
- Milroy, M.J., Bradley, C., Vickers, G.W., 1997. Segmentation of a wrap-around model using an active contour. *Comput. Aided Des.* 29 (4), 299–320.
- Parlett, B.N., 1998. *The Symmetric Eigenvalue Problem*. Society for Industrial and Applied Mathematics.
- Rabbani, T., Heuvel, F.A.V.D., Vosselman, G., 2006. Segmentation of point clouds using smoothness constraint. In: *International Archives of Photogrammetry. Remote Sensing & Spatial Information Sciences*.
- Reuter, M., 2009. Hierarchical shape segmentation and registration via topological features of Laplace-Beltrami eigenfunctions. *Int. J. Comput. Vis.* 89 (2), 287–308.
- Rustamov, R.M., 2011. Interpolated eigenfunctions for volumetric shape processing. *Vis. Comput.* 27 (11), 951–961.
- Várady, T., Benkő, P., 2000. Reverse engineering B-rep models from multiple point clouds. In: *Proceedings Geometric Modeling and Processing 2000. Theory and Applications*, pp. 3–12.
- Várady, T., Martin, R.R., Cox, J., 1997. Reverse engineering of geometric models - an introduction. *Comput. Aided Des.* 29 (4), 255–268.
- Vo, A.-V., Truong-Hong, L., Laefer, D.F., Bertolotto, M., 2015. Octree-based region growing for point cloud segmentation. *ISPRS J. Photogramm. Remote Sens.* 104, 88–100.
- Voro++, 2008. Voronoi diagram package. <http://www.math.lbl.gov/voro++/>.
- Woo, H., Kang, E., Wang, S., Lee, K.H., 2002. A new segmentation method for point cloud data. *Int. J. Mach. Tools Manuf.* 42 (2), 167–178.
- Xu, G., Zhang, Q., 2012. *Geometric Partial Differential Equation Methods in Computational Geometry*. Science Press.
- Yang, M., Lee, E., 1999. Segmentation of measured point data using a parametric quadric surface approximation. *Comput. Aided Des.* 31 (7), 449–457.
- Yang, Y., Li, C., Kao, C.Y., Osher, S., 2010. Split Bregman method for minimization of region-scalable fitting energy for image segmentation. In: *Proceedings of the 6th International Conference on Advances in Visual Computing - Volume Part II*. Springer-Verlag, Berlin, Heidelberg, pp. 117–128.
- Yokoya, N., Levine, M.D., 1989. Range image segmentation based on differential geometry: a hybrid approach. *IEEE Trans. Pattern Anal. Mach. Intell.* 11 (6), 643–649.
- Zhao, H.-K., Chan, T., Merriman, B., Osher, S., 1996. A variational level set approach to multiphase motion. *J. Comput. Phys.* 127 (1), 179–195.
- Zhao, D., Zhang, X., 1997. Range-data-based object surface segmentation via edges and critical points. *IEEE Trans. Image Process.* 6 (6), 826–830.