ISSN 2616-5775 Vol. 7, Issue 5: 44-54, DOI: 10.25236/AJCIS.2024.070506

Total Generalized Variation Regularized Piecewise Smooth Mumford-Shah Point Cloud Surface Segmentation

Shanqiang Wang[a], Huayan Zhang[b,*]

Tiangong University, Tianjin, 300387, China
[a]wsq9972@163.com, [b]zhanghy307@163.com
[*]Corresponding author

Abstract: *The Mumford-Shah (MS) model is an important tool for data segmentation. The previous research on piecewise constant MS segmentation model with total variation regularization pursued the shortest length of boundaries. By contrast, in this article, we propose a novel piecewise smooth Mumford-Shah segmentation model by utilizing the total generalized variation (TGV) regularization, which assumes that the feature function of a data can be approximated by the sum of a piecewise constant function and a smooth function. The newly introduced TGV regularized piecewise smooth model is effective in segmenting point cloud surfaces with irregular structures and getting the optimal boundaries rather than the shortest boundaries. We solve the piecewise smooth MS model by alternating minimization and alternating direction method of multipliers (ADMM), where the subproblems are solved by either the closed-form solution or numerical packages. Our algorithm is discussed from several aspects, and comparisons with the piecewise constant MS model. Experimental results show that our TGV regularized segmentation method can yield competitive results when compared to other approaches.*

Keywords: *Point cloud surface segmentation, Total generalized variation, The Mumford-Shah model, ADMM*

1. Introduction

The point cloud data is usually acquired from the surfaces of a physical object, which is a fundamental understanding of the physical structure. With the improvement of scanning technology, the amount of point cloud data is increasing. Modeling of data will face many problems such as large storage space, low computational efficiency, poor computational effect, and so on. Therefore, for point cloud data with large size and complex structures, it needs to be preprocessed via segmentation and fragmentation algorithms in order to be used effectively in subsequent applications such as parameterization, simplification, shape retrieval, multiresolution modeling, skeleton extraction, and so on. As the point cloud lacks of topological connection, the point cloud segmentation algorithm has greater challenges. So far, a wide variety of point cloud segmentation algorithms have been developed. The interested reader can refer to several excellent surveys [1,2] as well as some recent developments [3,4,5,6,7,8,9].

For a given point cloud surface S with region Ω, point cloud segmentation aims to decompose the data region Ω into $\mathbf{K}$ disjoint connected piecewise smooth subsets Ω_k with a union of smooth boundaries Γ such that

$$\Omega = \Omega_1 \cup \Omega_2 \cup \ldots \cup \Omega_{\mathrm{K}} \cup \Gamma, \qquad \Omega_i \cap \Omega_j = \emptyset.$$

The pioneered Mumford-Shah (MS) segmentation model [10] has been successfully applied to image segmentation and mesh surface segmentation [11,12,13,14,15,16,17,18], which tries to find an optimal piecewise smooth approximation $\mathbf{u}$ of f by the following minimization problem:

$$\min_{\mathbf{u},\Gamma} |\Gamma| + \beta \int_{\Omega\setminus\Gamma} |\nabla \mathbf{u}|^2 dx + \alpha \int_{\Omega} |f - \mathbf{u}|^2 dx, \tag{1}$$

where f is the input feature function of S, $\mathbf{u}$ is a piecewise smooth approximation of f, such that $\mathbf{u}$ varies smoothly within each Ω_i and discontinuously across the boundaries of Ω_i, and $|\Gamma|$ stands for the total boundary length.

To better deal with images with intensity inhomogeneity and surfaces with local uneven regions, the

ISSN 2616-5775 Vol. 7, Issue 5: 44-54, DOI: 10.25236/AJCIS.2024.070506

following model is proposed by decomposing a piecewise smooth function into a piecewise constant function and a smooth function.

$$I(x) = b(x)\mu_i + n(x), x \in \Omega, \tag{2}$$

where $b(x)$ is a smooth function, and $n(x)$ is the noise. Based on (2), Li et al. [15] built up the following piecewise smooth MS model by utilizing a non-convex $L^p(0 \le p \le 1)$ regularity term and laplace smooth term.

$$\min_{\{b,\mathbf{u},\mu\}} \int_\Omega |\nabla \mathbf{u}|^p \, dx + \frac{\beta}{2}\int_\Omega |\Delta \mathrm{b}|^2 dx + \frac{\eta}{2}\int_\Omega |b|^2 dx + \alpha \sum_k \int_{\Omega_k} \langle f_k, \mathbf{u}\rangle dx \,, \tag{3}$$

where $f = (f_1, f_2, \dots, f_{\mathbf{K}})$ with $f_k = (I - b - \mu_k)^2$.

The above piecewise smooth MS models have been demonstrated to be effective in dealing with images with intensity inhomogeneity. To the best of our knowledge, there are few research discussing piecewise smooth MS model techniques for point cloud segmentation. Inspired by the research work of piecewise smooth MS models in image segmentation, in the paper, we focus on studying piecewise smooth MS model for point cloud segmentation. Moreover, the existing Mumford-Shah models try to obtain the shortest length of boundaries $|\Gamma|$ with the total variation regularization (TV). However, as the surface is irregular, the shortest boundaries sometimes are not necessarily the optimal segmentation boundaries. Recently, the total generalized variation regularization (TGV) on surfaces [19,20] has been proved to be able to alleviate the stair-case effects of TV regularization effectively. Based on the good properties of TGV penalizing the one order discontinuity of a function. We consider total generalized variation regularized piecewise smooth MS point cloud segmentation method.

The previous research on piecewise constant MS segmentation method with TV regularization pursue the shortest length of boundaries. Different from previous work, the goal of this paper is to investigate the piecewise smooth Mumford-Shah model for point cloud surface segmentation with the TGV regularization trying to obtain the optimal boundaries instead of the shortest boundaries. The TGV regularization containing first order TV and second order differential operator is first presented. With this regularization in hand, we devise the TGV regularized piecewise smooth MS point cloud segmentation method. The optimization problem is solved by alternating minimization and alternating direction method of multipliers. Our algorithm is discussed and compared to several state-of-the-art methods in various aspects. Experimental results show that our piecewise smooth MS method.

The remainder of the paper is organized as follows. Section 2 gives some notations, differential operators on point cloud surface and the definition of TGV regularization. In section 3, we introduce our TGV regularized piecewise smooth MS point cloud segmentation method. Section 4 presents the details of solving our segmentation method. In section 5, we present our experiments and comparisons. Section 6 concludes the paper.

2. Total Generalized Variation on Point Cloud Surface

In the section, we introduce some notations followed by differential operators and total generalized variation regularization on point cloud surfaces.

2.1. Notations

Without loss of generality, we denote a discrete point cloud surface sampled from a smooth manifold as $S = v_i \in \mathbb{R}^3, i = 0,1, \dots, \mathrm{N}_s - 1$, where N_s is the number of vertices. The local mesh structure of v_i is denoted as $\mathcal{G}(i) = (v_i, \mathcal{N}(i), \mathcal{T}(i))$, where $\mathcal{N}(i)$ is the neighbour point set of v_i obtained by the k-Nearest Neighbour (kNN) method, and $\mathcal{T}(i)$ is the triangles set containing v_i achieved by the technique in [21]. Let $\{\tau_i, i = 0,1, \dots, \mathrm{T}_s - 1\}$ be the set of triangles and T_s is the number of triangles. In addition, for each vertex v_i, a linear basis function is denoted as $\phi_i(v_j) = \delta_{ij}$, where δ_{ij} is the Kronecker delta with $\mathcal{T}(i)$ being the local support.

2.2. Differential operators on point cloud surface

For self-inclusion, we present the definitions of differential operators on point cloud surface (see [16,20] for details).

We first denote the space $\mathbf{U}_S = \mathbf{R}^{\mathrm{N}_S \times n}$. For $\mathbf{u} = (\mathbf{u}_0, \dots, \mathbf{u}_\tau, \dots, \mathbf{u}_{\mathrm{N}_S - 1}) \in \mathbf{U}_S, \mathbf{u}_i = (u_{i,1}, \dots, u_{i,n})$, is

ISSN 2616-5775 Vol. 7, Issue 5: 44-54, DOI: 10.25236/AJCIS.2024.070506

a n-dimensional vector. We then have $\mathbf{u}$ on S.

$$\mathrm{u} = \sum_{0\le i\le \mathrm{N}_S-1} \mathrm{u}_i\, \phi_i. \tag{4}$$

For a triangle $\tau = (v_i, v_j, v_k)$, the gradient operator ∇ restricted on τ has the following form

$$(\nabla \mathbf{u})_\tau = \mathbf{u}_i(\nabla\phi_i)_\tau + \mathbf{u}_j\big(\nabla\phi_j\big)_\tau + \mathbf{u}_k(\nabla\phi_k)_\tau, \tag{5}$$

where $(\nabla\phi_i)_\tau = (v_i - O)/\|v_i - O\|^2$ is a piecewise constant vector (O is an intersection point of straight lines v_iO and v_iv_j satisfying $v_iO \perp v_jv_k$).

Based on (4) and (5), the gradient operator $\nabla\mathbf{u}$ on S is given as follows

$$\nabla\mathbf{u} = \sum_{0\le i\le \mathrm{N}_S-1} \mathbf{u}_i\, \nabla\phi_i = \sum_i \mathbf{u}_i \sum_{\tau\in\mathcal{T}(i)}\big(\nabla\phi_j\big)_\tau = \sum_{0\le\tau\le \mathrm{T}_S-1}(\nabla\mathbf{u})_\tau\,. \tag{6}$$

We then denote the range of ∇ as $\mathbf{V}_S$. For $\forall\mathbf{p}\in\mathbf{V}_S$, as the adjoint operator of ∇ is $-\mathrm{div}$, we then have the following divergence operator for S.

$$\big(\mathrm{div}(\mathbf{p})\big)_i = -\tfrac{1}{A_i}\sum_{\tau\in\mathcal{T}(i)}\langle\mathbf{p}_\tau, (\nabla\phi_i)_\tau\rangle\, A_\tau, \tag{7}$$

where $A_i = \frac{1}{3}\sum_{\tau\in\mathcal{T}(i)} A_\tau$, and A_τ is the area of τ.

Generally, we can get the following weighted divergence operator $\big(\mathrm{div}(T\mathbf{p})\big)_i$

$$\big(\mathrm{div}(T\mathbf{p})\big)_i = -\tfrac{1}{A_i}\sum_{\tau\in T(i)}\langle T\mathbf{p}_\tau, (\nabla\phi_i)_\tau\rangle\, A_\tau, \tag{8}$$

where T is a 3×3 matrix. When T is identity matrix, $\big(\mathrm{div}(T\mathbf{p})\big)_i = \big(\mathrm{div}(\mathbf{p})\big)_i$.

Based on the div operator (7) and (8), for $\mathbf{p}\in\mathbf{V}_S$, we present the definition of $(\nabla\mathbf{p})_i$ restricted on each v_i

$$(\nabla\mathbf{p})_i = \begin{bmatrix} \big(\mathrm{div}(T_{11}\mathbf{p})\big)_i & \big(\mathrm{div}(T_{12}\mathbf{p})\big)_i & \big(\mathrm{div}(T_{13}\mathbf{p})\big)_i \\ \big(\mathrm{div}(T_{21}\mathbf{p})\big)_i & \big(\mathrm{div}(T_{22}\mathbf{p})\big)_i & \big(\mathrm{div}(T_{23}\mathbf{p})\big)_i \\ \big(\mathrm{div}(T_{31}\mathbf{p})\big)_i & \big(\mathrm{div}(T_{32}\mathbf{p})\big)_i & \big(\mathrm{div}(T_{33}\mathbf{p})\big)_i \end{bmatrix}, \tag{9}$$

where $T_{ij} = [t_{mn}]$ is a 3×3 matrix with $t_{mn} = f(x) = \begin{cases} 1, & m = i, n = j, \\ 0, & else, \end{cases}$ $i,j = 0,1,2.$

For $\forall\mathbf{p}\in\mathbf{V}_S$, we then introduce the symmetrical space $\mathbf{W}_S$, which is the range of $\frac{\nabla\mathbf{p}+(\nabla\mathbf{p})^T}{2}$.

2.3. Total Generalized Variation Regularization (TGV)

According to above differential operator, we present the following TGV regularization on point cloud surface.

For $\mathbf{u}\in\mathbf{U}_S, \mathbf{v}\in\mathbf{V}_S, \mathbf{w}\in\mathbf{W}_S$, we then have

$$\begin{aligned} \mathrm{TGV}(\mathbf{u}) &= \min_{\mathbf{v}} \|\nabla\mathbf{u} - \mathbf{v}\|_1 + \alpha_0\|\mathcal{E}(\mathbf{v})\|_1 \\ &= \textstyle\sum_\tau\sum_{k=1}^n |(\nabla\mathbf{u} - \mathbf{v})_{\tau k}| A_\tau + \alpha_0 \sum_i \sum_{k=1}^n |\mathcal{E}(\mathbf{v})_{ik}| A_i\,, \end{aligned} \tag{10}$$

where $\mathcal{E}(\mathbf{v}) = \frac{\nabla\mathbf{v}+\nabla\mathbf{v}^T}{2}$ is the symmetrical partial differential operator.

The above TGV regularization has been applied for mesh denoising [20]. As point cloud surface is irregular and lack of connection structure, the existing TGV model has rarely been generalized to point cloud surface. In addition, the piecewise smooth Mumford-Shah (MS) segmentation model with total variation (TV) regularization has been proved to be able to deal with images with intensity inhomogeneity effectively [15,22,23]. However, the piecewise smooth MS model is also rarely applied to point cloud segmentation. In the following, we focus on TGV regularized piecewise smooth MS mesh segmentation method to point cloud segmentation.

ISSN 2616-5775 Vol. 7, Issue 5: 44-54, DOI: 10.25236/AJCIS.2024.070506

3. TGV regularized piecewise smooth MS point cloud segmentation method

3.1. The feature space $\boldsymbol{f}$

For a point cloud surface S to be divided into $\mathbf{K}$ disjoint parts, the spectral space is an effective way to describe the structure of S, which is consisted of the eigenvectors of the Laplacian matrix [17,18,24,25]. In the paper, we consider using the spectral space to define the feature space $\boldsymbol{f}$. Specifically, the Laplacian matrix $L = [L_{ij}]$ of S is given as follows:

$$L_{ij} = \begin{cases} -\omega_{ij}, & i \neq j \text{ and } \tau_i, \tau_j \text{ share an edge,} \\ \sum_k \omega_{i,k}, & i = j \text{ and } k \in \mathcal{N}_i, \\ 0, & \text{otherwise,} \end{cases} \tag{11}$$

where $\omega_{ij} = l_e \exp(-\frac{d(n_i,n_j)}{\overline{d}})$, $\overline{d}$ being the average of $d(\tau_i, \tau_j)$ over all edges, and $d(n_i, n_j) = \sigma\|n_i - n_j\|$ with n_i being the normal of vertices v_i. the parameter $\sigma = 1$ for a concave edge and $\sigma = 0.1$ for a convex edge. Suppose $\{l_0, l_1, \dots, l_K\}$ are the eigenvectors of the $K+1$ smallest eigenvalues of L. We then define the feature space as $\boldsymbol{f} = \{l_1, l_2, \dots, l_{K-1}\}$.

3.2. The proposed point cloud segmentation method

By introducing a multi-continuous label $\mathbf{u} \in C_{\mathbf{u}} = \{\mathbf{u}_i | u_{i,k} \geq 0, \sum_{k=1}^{\mathbf{K}} u_{i,k} = 1, \forall i\} \subset \mathbf{U}_S$, we then obtain the following piecewise smooth MS point cloud segmentation optimization problem.

$$\min_{\mathbf{u} \in C_{\mathbf{u}}, \boldsymbol{b}, \mu} \mathrm{TGV}(\mathbf{u}) + \frac{\beta}{2}\|\Delta\mathbf{b}\|_{\mathbf{U}_S}^2 + \frac{\eta}{2}\|\mathbf{b}\|_{\mathbf{U}_S}^2 + \frac{\alpha}{2}\langle \mathbf{u}, s(\mathbf{f}, \mathbf{b}, \mu)\rangle_{\mathbf{U}_S}, \tag{12}$$

where $s(\mathbf{f}, \mathbf{b}, \mu) = \{s_i(\mathbf{f}, \mathbf{b}, \mu)\}$ with $s_i(\mathbf{f}, \mathbf{b}, \mu) = \{\|\mathbf{f}_i - \mathbf{b}_i - \mu_k\|^2\}_{k=1}^{\mathbf{K}}$, $\mathbf{f}$ can be normals or eigenvectors of a Laplace matrix, and β, η, α are positive parameters.

3.3. Alternating minimization method (AMM) for solving (12)

By introducing the following function:

$$\chi(\mathbf{u}) = \begin{cases} 0, & \mathbf{u} \in \boldsymbol{C}_{\mathbf{u}} \\ +\infty, & \mathbf{u} \notin \boldsymbol{C}_{\mathbf{u}} \end{cases}, \tag{13}$$

The minimization problem (12) can be solved alternately by the following two optimization problems.

- For fixed μ: $\mathbf{u}, \mathbf{b}$ can be obtained by

$$\min_{\mathbf{u}, \mathbf{b}, \mu} \mathrm{TGV}(\mathbf{u}) + \frac{\beta}{2}\|\Delta\mathbf{b}\|_{\mathbf{U}_S}^2 + \frac{\eta}{2}\|\mathbf{b}\|_{\mathbf{U}_S}^2 + \frac{\alpha}{2}\langle \mathbf{u}, s(\mathbf{f}, \mathbf{b}, \mu)\rangle_{\mathbf{U}_S} + \chi(\mathbf{u}), \tag{14}$$

which is non-differentiable, and can be efficiently solved by augmented Lagrangian method (ALM) or alternating direction multiplier method (ADMM) [26,27,28]; see section 4 for details.

- For fixed $\mathbf{u}, \mathbf{b}$: μ can be obtained by

$$\min_{\mu} \frac{\alpha}{2}\langle \mathbf{u}, s(\mathbf{f}, \mathbf{b}, \mu)\rangle_{\mathbf{U}_S},$$

and is exactly

$$\mu_k = \frac{\sum_i u_{k,i}(\mathbf{f}_i - \mathbf{b}_i)}{\sum_i u_{k,i} A_i}, k = 0,1, \dots, \mathbf{K}. \tag{15}$$

In our implementation, the alternating minimization method is listed in Algorithm 1.

Algorithm 1 AMM for solving (12)

1. Initialization:
1.1 μ^{-1}: computed according to the technique in [18];
1.2 $\mathbf{u}^{-1}$: obtained by μ^{-1};
1.3 $\mathbf{v}^{-1} = 0, \mathbf{b}^{-1} = 0, \mathbf{p}^{-1} = 0, \mathbf{q}^{-1} = 0, \mathbf{z}^{-1} = 0, \lambda_{\mathbf{p}}^{-1} = 0, \lambda_{\mathbf{q}}^{-1} = 0, \lambda_{\mathbf{z}}^{-1} = 0, l = -1$;
2. Repeat
2.1 For fixed μ^{l-1}, computing $(\mathbf{u}^l, \mathbf{v}^l, \mathbf{b}^l, \mathbf{p}^l, \mathbf{q}^l, \mathbf{z}^l, \lambda_{\mathbf{p}}^l, \lambda_{\mathbf{q}}^l, \lambda_{\mathbf{z}}^l)$ by solving (14) through Algorithm 2;
2.2 For fixed $(\mathbf{u}^l, \mathbf{b}^l)$, computing μ^l from (15);
Until$(\|u^{n+1} - u^n\|_{\mathbf{U}_S} < 10^{-5})$.
3. Classify $\mathbf{u}$ by the method in [29].

ISSN 2616-5775 Vol. 7, Issue 5: 44-54, DOI: 10.25236/AJCIS.2024.070506

4. Algorithm details for solving (14)

In the following, we present the details for solving (14). By introducing three auxiliary variables $\mathbf{p} \in \mathbf{V}_S$ and $\mathbf{q}, \mathbf{z} \in \mathbf{U}_S$. The problem (14) with $p = 2$ can be further written as

$$\min_{\substack{\mathbf{u},\mathbf{b}\in\mathbf{U}_S,\mathbf{v}\in\mathbf{V}_S,\\ \mathbf{p}\in\mathbf{V}_S,\mathbf{q},\mathbf{z}\in\mathbf{U}_S}} \|\mathbf{p}\|_1 + \alpha_0\|\mathbf{q}\|_1 + \frac{\beta}{2}\|\Delta\mathbf{b}\|_{\mathbf{U}_S}^2 + \frac{\eta}{2}\|\mathbf{b}\|_{\mathbf{U}_S}^2 + \frac{\alpha}{2}\langle \mathbf{z}, s(\mathbf{f},\mathbf{b},\mu)\rangle_{\mathbf{U}_S} + \chi(\mathbf{z}), \tag{16}$$

$$\text{s.t. } \mathbf{p} = \nabla\mathbf{u} - \mathbf{v}, \mathbf{q} = \mathcal{E}(\mathbf{v}), \mathbf{z} = \mathbf{u}.$$

To solve (16) effectively, we define the following augmented Lagrangian functional:

$$\begin{aligned}\mathcal{L}(\mathbf{u},\mathbf{v},\mathbf{b},\mathbf{p},\mathbf{q},\mathbf{z};\lambda_\mathbf{p},\lambda_\mathbf{q},\lambda_\mathbf{z}) = {} & \|\mathbf{p}\|_1 + \alpha_0\|\mathbf{q}\|_1 + \frac{\beta}{2}\|\Delta\mathbf{b}\|_{\mathbf{U}_S}^2 + \frac{\eta}{2}\|\mathbf{b}\|_{\mathbf{U}_S}^2 \\ & + \frac{\alpha}{2}\langle \mathbf{z}, s(\mathbf{f},\mathbf{b},\mu)\rangle_{\mathbf{U}_S} + \chi(\mathbf{z}) \\ & + \left(\lambda_\mathbf{p}, \mathbf{p} - (\nabla\mathbf{u} - \mathbf{v})\right)_{\mathbf{V}_S} + \frac{r_\mathbf{p}}{2}\|\mathbf{p} - (\nabla\mathbf{u} - \mathbf{v})\|_{\mathbf{V}_S}^2 \\ & + \left(\lambda_\mathbf{q}, \mathbf{q} - \mathcal{E}(\mathbf{v})\right)_{\mathbf{U}_S} + \frac{r_\mathbf{q}}{2}\|\mathbf{q} - \mathcal{E}(\mathbf{v})\|_{\mathbf{U}_S}^2 \\ & + (\lambda_\mathbf{z}, \mathbf{z} - \mathbf{u})_{\mathbf{U}_S} + \frac{r_\mathbf{z}}{2}\|\mathbf{z} - \mathbf{u}\|_{\mathbf{U}_S}^2,\end{aligned} \tag{17}$$

where $r_\mathbf{p}, r_\mathbf{q}$ and $r_\mathbf{z}$ are positive parameters.

The solution of (17) is equivalent to the following saddle-point problem

$$\max_{\lambda_\mathbf{p},\lambda_\mathbf{q},\lambda_\mathbf{z}} \min_{\substack{\mathbf{u},\mathbf{v},\mathbf{b},\\ \mathbf{p},\mathbf{q},\mathbf{z}}} \mathcal{L}(\mathbf{u},\mathbf{v},\mathbf{b},\mathbf{p},\mathbf{q},\mathbf{z};\lambda_\mathbf{p},\lambda_\mathbf{q},\lambda_\mathbf{z}), \tag{18}$$

which can be iteratively solved by splitting (18) into the following several subproblems.

4.1. Sub-minimizations with respect to u, v, b

The $\mathbf{u},\mathbf{v},\mathbf{b}$ sub-problems are reformulated as follows

$$\min_{\mathbf{u}\in\mathbf{U}_S} \left(\lambda_\mathbf{p}, -\nabla\mathbf{u}\right)_{\mathbf{V}_S} + (\lambda_\mathbf{z}, -\mathbf{u})_{\mathbf{U}_S} + \frac{r_\mathbf{p}}{2}\|\mathbf{p} - (\nabla\mathbf{u} - \mathbf{v})\|_{\mathbf{V}_S}^2 + \frac{r_\mathbf{z}}{2}\|\mathbf{z} - \mathbf{u}\|_{\mathbf{U}_S}^2 \tag{19}$$

$$\min_{\mathbf{v}\in\mathbf{V}_S} \left(\lambda_\mathbf{q}, -\mathcal{E}(\mathbf{v})\right)_{\mathbf{U}_S} + \left(\lambda_\mathbf{p}, \mathbf{v}\right)_{\mathbf{V}_S} + \frac{r_\mathbf{q}}{2}\|\mathbf{q} - \mathcal{E}(\mathbf{v})\|_{\mathbf{U}_S}^2 + \frac{r_\mathbf{p}}{2}\|\mathbf{p} - (\nabla\mathbf{u} - \mathbf{v})\|_{\mathbf{V}_S}^2 \tag{20}$$

$$\min_{\mathbf{b}\in\mathbf{U}_S} \frac{\alpha}{2}\langle \mathbf{z}, s(\mathbf{f},\mathbf{b},\mu)\rangle_{\mathbf{U}_S} + \frac{\beta}{2}\|\Delta\mathbf{b}\|_{\mathbf{U}_S}^2 + \frac{\eta}{2}\|\mathbf{b}\|_{\mathbf{U}_S}^2. \tag{21}$$

The above sub-problems are quadratic programming problems, and can be solved by various numerical packages, such as MKL, Taucs and Eigen.

4.2. Sub-minimizations with respect to p, q, z

Specifically, the $\mathbf{p},\mathbf{q},\mathbf{z}$ sub-problems can be reformulated as follows and solved with the closed form solutions.

- For $\mathbf{p}$ sub-problem, we have

$$\min_{\mathbf{p}\in\mathbf{V}_S} \|\mathbf{p}\|_1 + \left(\lambda_\mathbf{p}, \mathbf{p}\right)_{\mathbf{V}_S} + \frac{r_\mathbf{p}}{2}\|\mathbf{p} - (\nabla\mathbf{u} - \mathbf{v})\|_{\mathbf{V}_S}^2. \tag{22}$$

It has the following closed form solution

$$\forall e, \mathbf{p}_e = \begin{cases} \left(1 - \frac{1}{r_\mathbf{p}|\mathbf{w}_e|}\right)\mathbf{w}_e, & |\mathbf{w}_e| > \frac{1}{r_\mathbf{p}}, \\ 0, & |\mathbf{w}_e| \le \frac{1}{r_\mathbf{p}}, \end{cases} \tag{23}$$

where $\mathbf{w}_e = \left(\nabla\mathbf{u} - \mathbf{v} - \frac{\lambda_\mathbf{p}}{r_\mathbf{p}}\right)|_e$.

- For $\mathbf{q}$ sub-problem, we have

ISSN 2616-5775 Vol. 7, Issue 5: 44-54, DOI: 10.25236/AJCIS.2024.070506

$$\min_{\mathbf{u}\in\mathbf{U}_S} \alpha_0\|\mathbf{q}\|_1 + \left(\lambda_{\mathbf{q}}, \mathbf{q}\right)_{\mathbf{U}_S} + \frac{r_{\mathbf{q}}}{2}\|\mathbf{q} - \mathcal{E}(\mathbf{v})\|_{\mathbf{U}_S}^2, \tag{24}$$

which has the following closed form solution

$$\forall \tau, \mathbf{q}_i = \begin{cases} \left(1 - \frac{\alpha_0}{r_{\mathbf{q}}|\mathbf{c}_i|}\right)\mathbf{c}_i, & |\mathbf{c}_i| > \frac{\alpha_0}{r_{\mathbf{q}}}, \\ 0, & |\mathbf{c}_i| \le \frac{\alpha_0}{r_{\mathbf{q}}}, \end{cases} \tag{25}$$

where $\mathbf{c}_i = \mathcal{E}(\mathbf{v}) - \frac{\lambda_{\mathbf{q}}}{r_{\mathbf{q}}}$.

● For $\mathbf{z}$ sub-problem, we get

$$\min_{\mathbf{z}\in\mathbf{U}_S} \frac{\alpha}{2}\langle \mathbf{z}, s(\mathbf{f}, \mathbf{b}, \mu)\rangle_{\mathbf{U}_S} + \chi(\mathbf{z}) + (\lambda_{\mathbf{z}}, \mathbf{z})_{\mathbf{U}_S} + \frac{r_{\mathbf{z}}}{2}\|\mathbf{z} - \mathbf{u}\|_{\mathbf{U}_S}^2, \tag{26}$$

It can be solved through

$$\mathrm{z} = \mathrm{Proj}_{C_{\mathrm{u}}}\left(\mathrm{u} - \frac{\alpha s(\mathrm{f},\mathrm{b},\mu) + \lambda_{\mathrm{z}}}{r_{\mathrm{z}}}\right), \tag{27}$$

which can be calculated via Michelot's algorithm [30].

The whole algorithm for solving (18) is listed in Algorithm 2.

Algorithm 2 ADMM for solving (18)

1. Initialization:
1.1 $\mathbf{u}^{l,0} = \mathbf{u}^{l-1}, \mathbf{v}^{l,0} = \mathbf{v}^{l-1}, \mathbf{b}^{l,0} = \mathbf{b}^{l-1}, \mathbf{p}^{l,0} = \mathbf{p}^{l-1}, \mathbf{q}^{l,0} = \mathbf{q}^{l-1}, \mathbf{z}^{l,0} = \mathbf{z}^{l-1}$;
1.2 $\lambda_{\mathbf{p}}^{l+1,0} = \lambda_{\mathbf{p}}^{l}, \lambda_{\mathbf{q}}^{l+1,0} = \lambda_{\mathbf{q}}^{l}, \lambda_{\mathbf{z}}^{l+1,0} = \lambda_{\mathbf{z}}^{l}, k = 0$;
2. Repeat
2.1 **z**-subproblem: For fixed $(\lambda_{\mathbf{z}}^{l+1,k}, \mathbf{u}^{l,k}, \mathbf{b}^{l,k})$, compute $\mathbf{z}^{l,k+1}$ from (27);
2.2 **u**-subproblem: For fixed $\left(\lambda_{\mathbf{p}}^{l+1,k}, \lambda_{\mathbf{z}}^{l+1,k}, \mathbf{p}^{l,k}, \mathbf{v}^{l,k}, \mathbf{z}^{l,k+1}\right)$, compute $\mathbf{u}^{l,k+1}$ by solving (19);
2.3 **v**-subproblem: For fixed $\left(\lambda_{\mathbf{p}}^{l+1,k}, \lambda_{\mathbf{q}}^{l+1,k}, \mathbf{p}^{l,k}, \mathbf{q}^{l,k}, \mathbf{u}^{l,k+1}\right)$, compute $\mathbf{v}^{l,k+1}$ by solving (20);
2.4 **b**-subproblem: For fixed $(\mathbf{z}^{l,k+1})$, compute $\mathbf{b}^{l,k+1}$ by solving (21);
2.5 **p**-subproblem: For fixed $\left(\lambda_{\mathbf{p}}^{l+1,k}, \mathbf{v}^{l,k+1}, \mathbf{u}^{l,k+1}\right)$, compute $\mathbf{p}^{l,k+1}$ from (23);
2.6 **q**-subproblem: For fixed $\left(\lambda_{\mathbf{q}}^{l+1,k}, \mathbf{v}^{l,k+1}\right)$, compute $\mathbf{q}^{l,k+1}$ from (25);
2.7 Update Lagrange multipliers:
$\lambda_{\mathbf{p}}^{l+1,k+1} = \lambda_{\mathbf{p}}^{l+1,k} + r_{\mathbf{p}}(\mathbf{p}^{l,k+1} + \mathbf{v}^{l,k+1} - \nabla\mathbf{u}^{l,k+1})$;
$\lambda_{\mathbf{q}}^{l+1,k+1} = \lambda_{\mathbf{q}}^{l+1,k} + r_{\mathbf{q}}\left(\mathbf{q}^{l,k+1} - \mathcal{E}(\mathbf{v}^{l,k+1})\right)$;
$\lambda_{\mathbf{z}}^{l+1,k+1} = \lambda_{\mathbf{z}}^{l+1,k} + r_{\mathbf{z}}(\mathbf{u}^{l,k+1} - \mathbf{z}^{l,k+1})$;
Until $(\|\mathbf{u}^{l,k+1} - \mathbf{u}^{l,k}\|_{\mathbf{U}_S} < 10^{-3})$.
3. $\mathbf{u}^{l} = \mathbf{u}^{l,*}, \mathbf{v}^{l} = \mathbf{v}^{l,*}, \mathbf{b}^{l} = \mathbf{b}^{l,*}, \mathbf{p}^{l} = \mathbf{p}^{l,*}, \mathbf{q}^{l} = \mathbf{q}^{l,*}, \mathbf{z}^{l} = \mathbf{z}^{l,*}$,
$\lambda_{\mathbf{p}}^{l+1} = \lambda_{\mathbf{p}}^{l,*}, \lambda_{\mathbf{q}}^{l+1} = \lambda_{\mathbf{q}}^{l,*}, \lambda_{\mathbf{z}}^{l+1} = \lambda_{\mathbf{z}}^{l,*}$.

5. Experimental results and discussions

In this section, we present experiments of our piecewise smooth MS segmentation method with TGV regularization on a wide variety of point cloud surfaces. The experiments of our algorithms are conducted using Microsoft Visual Studio 2010 on a desktop with Intel(R) Core (TM) i7-8850 CPU @2.60GHz and 16GB memory and all point cloud surfaces were rendered using point shading.

This section will discuss our method from several aspects, such as choices of parameters, comparison with other algorithms, experimental results on various point cloud surfaces, computational cost and limitation analysis.

5.1. Parameters and the number of segments

There are seven parameters in our algorithm: $\alpha, \beta, \alpha_0, \eta, r_{\mathbf{p}}, r_{\mathbf{q}}$ and $r_{\mathbf{z}}$. Therein $\alpha, \beta, \alpha_0, \eta$ are variational model parameters; $r_{\mathbf{p}}, r_{\mathbf{q}}, r_{\mathbf{z}}$ are optimization algorithmic parameters. The parameters $\eta, r_{\mathbf{p}}, r_{\mathbf{q}}$ and $r_{\mathbf{z}}$ can be fixed by $\eta = 0.00001, r_{\mathbf{p}} = 1, r_{\mathbf{q}} = 1, r_{\mathbf{z}} = 100$. Moreover, α, β and α_0 affect the segmentation results. According to lots of experimental tests, the parameter settings of α, β and α_0

ISSN 2616-5775 Vol. 7, Issue 5: 44-54, DOI: 10.25236/AJCIS.2024.070506

are summarized as follows.

Firstly, the parameter α affects the segmentation boundary. The larger α is, the better the segmentation boundary is; see Fig.1 for the impact of the parameter α on the results. However, when the α is too large, the algorithm may over segment the point cloud surfaces (see the result with $\alpha = 5.0 \times 10^5$ in Fig.1).

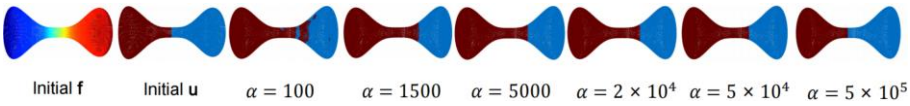

Figure 1: The segmentation results of different α by our TGV-PS method with other parameters fixed.

Secondly, the parameter β also plays an important impact on the segmentation boundary. Fig.2 shows the impact of parameter β on the experimental results when other parameters are fixed. It can be seen from the figure that the larger the β value, the better the segmentation boundary. When β reaches a certain threshold, the result reaches the optimal, and subsequent increases in β will have less impact on the result. According to a large number of experiments, β is in the range of $[10^3, 10^7]$.

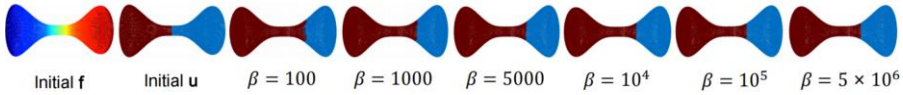

Figure 2: The segmentation results of different β by our TGV-PS method with other parameters fixed.

Finally, α_0 has an effect on the segmentation result. Fig.3 shows the impact of different parameters α_0 on the segmentation results when other parameters are fixed. As shown, similar to the parameter β, the larger the value of the parameter α_0, the better the segmentation boundary is. When α_0 reaches a certain threshold, the result reaches the optimal, and subsequent increasing of α_0 has less impact on the result. Due to the diversity of point clouds and the non-convexity of the Mumford-Shah model, it is difficult to estimate this parameter with a formula. However, according to a large number of experiments, it can be seen that α_0 is in the range of $(0, 100]$.

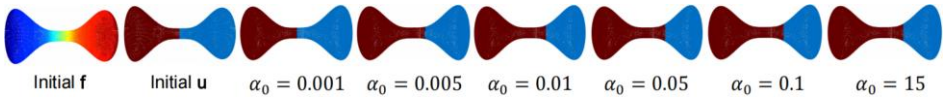

Figure 3: The segmentation results of different α_0 by our TGV-PS method with other parameters fixed.

The number of segments **K** are affected by the topological structure and semantic information by the point cloud. It is difficult to find an empirical formula suitable for all point cloud surfaces. In the paper, **K** is set manually.

5.2. Comparisons to other MS-based methods

As our method is based on TGV regularized piecewise smooth MS segmentation model, we consider compare our method with three MS-based segmentation methods. Specifically, these methods are our piecewise smooth MS model method based on TGV regularization (denoted as "TGV-PS"), the piecewise linear MS model based on TV regularization (denoted as "TV-PL"), the piecewise smooth MS model based on TV regularization (denoted as "TV-PS") and the piecewise linear MS model based on TGV regularization (denoted as "TGV-PL").

Fig.4 shows the comparison results between our method and other three MS-based segmentation approaches. From the results, it can be found that the TV-PL method and TV-PS method based on the TV regularization obtain the shortest segmentation boundaries, which is not the optimal segmentation boundaries. By comparison, the TGV-PL method and the TGV-PS method adopt the second-order TGV regularization, which is able to capture the first-order discontinuous information, and can achieve the better segmentation boundaries.

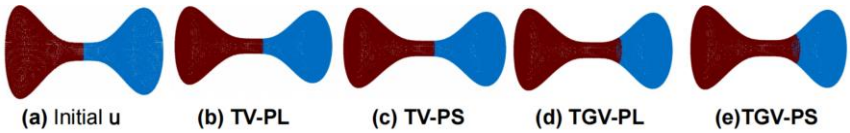

Figure 4: Comparisons of segmentation results among Initialization (a), TV-PL(b), TV-PS(c), TGV-PL(d) and TGV-PS(e).

ISSN 2616-5775 Vol. 7, Issue 5: 44-54, DOI: 10.25236/AJCIS.2024.070506

In addition, we further analyze the differences between TGV-PL and TGV-PS. In Fig.5, we present the segmentation results produced by the TGV-PL method and TGV-PS method with different parameters α_0 and α, respectively. As the TGV-PS method containing the smooth function $\mathbf{b}$, the results are semantic and stable with the changing of the parameter α (see the third row of Fig.5). However, the TGV-PL method is more sensitive to the changing of the parameter α_0 and α due to lack of smooth function $\mathbf{b}$. The segmentation results of the TGV-PL method are prone to falling to the local minima.

Therefore, the TGV regularized piecewise smooth MS method exhibits the greater advantage than other MS-based segmentation techniques.

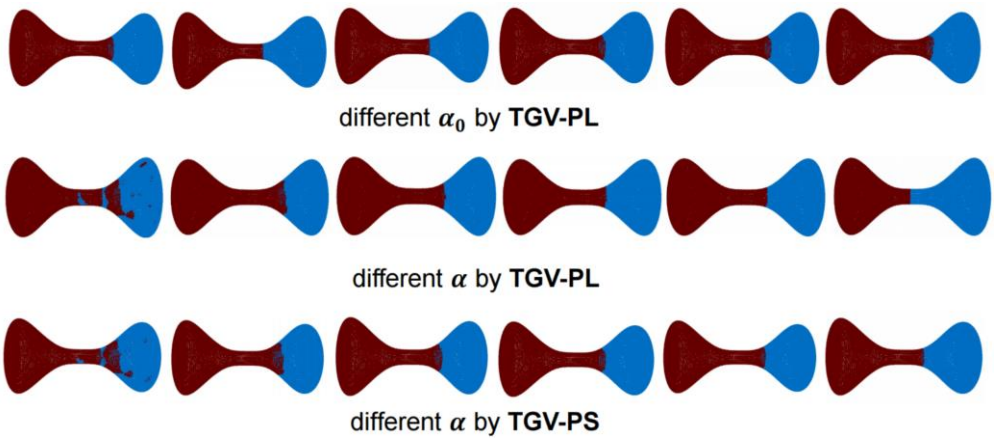

Figure 5: The segmentation results of different α_0 by TGV-PL method with other parameters fixed (The first row), the segmentation results of different α by TGV-PL method with other parameters fixed (The second row), and the segmentation results of different α by TGV-PS method with other parameters fixed (The last row).

5.3. More segmentation results

This section will show the segmentation results of the piecewise smooth MS point cloud segmentation method based on TGV regularization on various point cloud surfaces. It includes two aspects: two-region segmentation and multi-region segmentation.

The first is the display of the two-region segmentation results of point cloud surfaces. Fig.6 shows the two-region segmentation results of our TGV-PS method on some point cloud surfaces. The four rows results of Fig.6 are the original point cloud, the initialization segmentation results, the distribution map of the characteristic function $\mathbf{f}$, and the two-region segmentation result of the TGV-PS method, respectively. As observed, our method can capture the geometric structure of the point cloud very well, and obtain the better segmentation results consisting with the human semantic cognition.

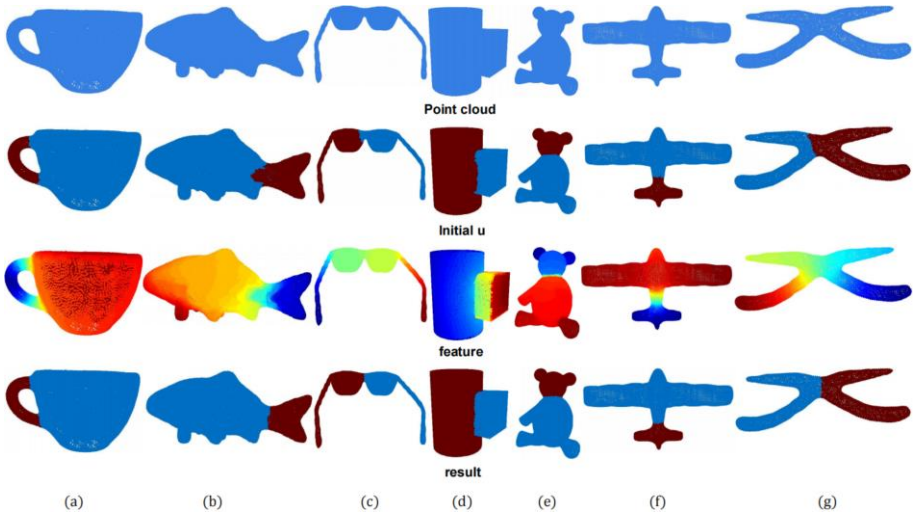

Figure 6: The two-region segmentation results of some point cloud surfaces by our TGV-PS method.

Then, Fig.7 shows the multi-region segmentation results of our TGV-PS method on various point

ISSN 2616-5775 Vol. 7, Issue 5: 44-54, DOI: 10.25236/AJCIS.2024.070506

cloud surfaces. It can be seen that the results are consistent with the semantics of human perception, indicating that our method can achieve satisfactory results.

Figure 7: The multi-region segmentation results of some point cloud surfaces by our TGV-PS method.

5.4. Computational cost

We now discuss the computational cost. The main computational cost is the iteratively solving the six sub problems in Algorithm 2. The main factors affecting the computational cost are the number of points in the point cloud, the number of segmentations and the parameter α. When the point cloud size is large or the number of segmentations is large, the computational time is longer. It was also found that if the parameter α is too small, the algorithm will be very slow. The new smooth function $\mathbf{b}$ and TGV regularization based on piecewise linear space will also make the running time of the method longer.

5.5. Limitations

Our method has been demonstrated very effective in getting the optimal segmentation boundaries on point cloud surfaces, but it still has some limitations. Firstly, we cannot give a formula to precisely compute the parameters and segmentation parts $\mathbf{K}$. Secondly, for point cloud surfaces with too sparse density and uneven distribution, it is difficult to construct an accurate topology. It causes our algorithm to be unable to give satisfactory segmentation results (see Fig.8 for an example), where the "guitar" point cloud density in Fig.8(a) is too sparse, and the "cup" handle in Fig.8(b) is unevenly distributed, both of which lead to poor segmentation results. Finally, due to the non-convexity of the algorithm, the convergence of the algorithm cannot be guaranteed and the computational efficiency is relatively low.

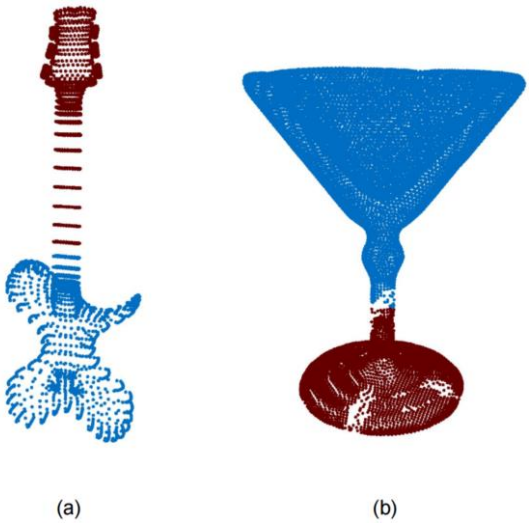

Figure 8: Failed results produced by our TGV-PS method.

ISSN 2616-5775 Vol. 7, Issue 5: 44-54, DOI: 10.25236/AJCIS.2024.070506

6. Conclusions

The Mumford-Shah (MS) model is a vital tool for data segmentation, which pursues the shortest length of boundaries by the total variation regularization. The previous efforts try to solve the MS model through approximating the solutions with the piecewise constant functions. Different from most previous methods, in the article, we presented a novel piecewise smooth Mumford-Shah point cloud surface segmentation technique using the total general variation regularization. The new model assumes to approximate the solutions by a sum of piecewise constant functions and a smooth function, which is effective in segmenting point cloud surfaces with irregular structures and getting the optimal boundaries rather than the shortest boundaries.

We solve our piecewise smooth MS method by the optimization iterative algorithm based on alternating minimization and ADMM, where the subproblems are solved by either the closed-form solution or numerical packages. Our algorithm is discussed from several aspects, including setting of parameters, computational costs and comparisons with other methods. Experimental results show that our piecewise smooth MS method can yield competitive results.

There are a few problems for further investigation. For instance, design segmentation algorithm based on the TGV diffusion model.

References

[1] A. Nguyen and B. Le. 3d point cloud segmentation: A survey. In 2013 6th IEEE conference on robotics, automation and mechatronics (RAM), pages 225230. IEEE, 2013.
[2] E. Grilli, F. Menna, and F. Remondino. A review of point clouds segmentation and classification algorithms. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 42:339344, 2017.
[3] O. Kaick, N. Fish, Y. Kleiman, S. Asa, and D. Cohen-Or. Shape segmentation by approximate convexity analysis. ACM Trans. Graph., 34(1), 2014.
[4] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classication and segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 652660, 2017.
[5] X. Li, Y. J. Zhang, X. Yang, H. Xu, and G. Xu. Point cloud surface segmentation based on volumetric eigenfunctions of the laplace-beltrami operator. Computer Aided Geometric Design, 71:157175, 2019.
[6] M. Xu, Z. Zhou, and Y. Qiao. Geometry sharing network for 3d point cloud classification and segmentation. In Proceedings of the AAAI Conference on Articial Intelligence, volume 34, pages 1250012507, 2020.
[7] H. Zhang and C. Wang. Total variation diffusion and its application in shape decom position. Computers & Graphics, 90:95107, 2020.
[8] S. Qiu, S. Anwar, and N. Barnes. Dense-resolution network for point cloud classification and segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 38133822, 2021.
[9] L. Tang, Y. Zhan, Z. Chen, B. Yu, and D. Tao. Contrastive boundary learning for point cloud segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 84898499, 2022.
[10] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. Commun. Pur. Appl. Math., 42(5):577685, 1989.
[11] A. Chambolle. Image segmentation by variational methods: Mumford and shah functional and the discrete approximations. SIAM J. Appl. Math., 55(3):827863, 1995.
[12] L. Vese and T. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. International Journal of Computer Vision, 50(3):271293, 2002.
[13] J. Lie, M. Lysaker, and X.-C. Tai. A binary level set model and some applications for mumford-shah image segmentation. IEEE Trans. Image Proc., 15(5):11711181, 2006.
[14] E. Brown, T. Chan, and X. Bresson. Completely convex formulation of the chan-vese image segmentation model. Int. J. Comput. Vis., 98(1):103121, 2012.
[15] Y. Li, C. Wu, and Y. Duan. The tv_p regularized mumford-shah model for image labeling and segmentation. IEEE Transactions on Image Processing, 29:70617075, 2020.
[16] C. Wu, J. Zhang, Y. Duan, and X. Tai. Augmented lagrangian method for total variation based image restoration and segmentation over triangulated surfaces. J. Sci. Comput., 50(1):145166, 2012.
[17] J. Zhang, J. Zheng, C. Wu, and J. Cai. Variational mesh decomposition. ACM Trans. Graph.,

ISSN 2616-5775 Vol. 7, Issue 5: 44-54, DOI: 10.25236/AJCIS.2024.070506

31(3):21, 2012.
[18] H. Zhang, C. Wu, J. Deng, Z. Liu, and Y. Yang. A new two-stage mesh surface seg mentation method. The Visual Computer, 34:15971615, 2018.
[19] H. Zhang, Z. He, and X. Wang. A novel mesh denoising method based on relaxed second-order total generalized variation. SIAM Journal on Imaging Sciences, 15(1):1 22, 2022.
[20] H. Zhang and P. Zhichao. Total generalized variation for triangulated surface data. Journal of Scientific Computing, 93(3):87, 2022.
[21] R. Lai, J. Liang, and H. Zhao. A local mesh method for solving pdes on point clouds. Inverse Problems & Imaging, 7(3), 2013.
[22] Y. Gu, W. Xiong, L.-L. Wang, and J. Cheng. Generalizing mumford-shah model for multiphase piecewise smooth image segmentation. IEEE Transactions on Image Processing, 26(2):942952, 2016.
[23] M. Jung. Piecewise-smooth image segmentation models with l1 data-fidelity terms. Journal of Scientific Computing, 70(3):12291261, 2017.
[24] J. Shi and J. Malik. Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell., 22(8):888905, 2000.
[25] W. Tong, X. Yang, M. Pan, and F. Chen. Spectral mesh segmentation via l0 gradient minimization. IEEE transactions on visualization and computer graphics, 26(4):1807 1820, 2020.
[26] C. Wu and X. Tai. Augmented lagrangian method, dual methods, and split bregman iteration for rof, vectorial tv, and high order models. SIAM J. Imaging Sci., 3(3):300 339, 2010.
[27] M. NG, P. Weiss, and X. Yuan. Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods. SIAM J. Sci. Comput., 32(5):27102736, 2010.
[28] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning, 3(1):1122, 2011.
[29] J. Lellmann and C. Schnoerr. Continuous multiclass labeling approaches and algorithms. SIAM J. Imaging Sci., 4(4):10491096, 2011.
[30] C. Michelot. A finite algorithm for finding the projection of a point onto the canonical simplex of r^n. J. Optim. Theory Appl., 50(1):195200, 1986.