Journal of Marine Science and Application (2026) 25: 1500-1517
https://doi.org/10.1007/s11804-025-00755-9

RESEARCH ARTICLE

Filtering Multibeam Bathymetric Point Cloud Based on Smoothed Particle Simulation

Tian Zhou[1,2,3], Zhenyu Yan[1,2,3], Weidong Du[1,2,3], Chao Xu[1,2,3], Weilu Liu[1,2,3] and Sen Xu[4]

Received: 09 April 2025 / Accepted: 03 August 2025
© Harbin Engineering University and Springer-Verlag GmbH Germany, part of Springer Nature 2026

Abstract

Recently, the underwater digital terrain model (UDTM) has been extensively used for seafloor terrain measurement. Identifying terrain and nonterrain points from an underwater bathymetric point cloud has been a fundamental problem in UDTM modeling. Hence, we propose a new ground filtering method derived from smoothed particle hydrodynamics to distinguish bathymetric points. The proposed method is called smoothed particle simulation filtering (SPSF), which embodies filtering as a matching of a smoothed particle grid to the point-set surface. First, four seafloor terrain features, namely curvature-based shape factor, slope, roughness, and concave–convex coefficient, are extracted from the moving least-square fitting surface of the original point-set surface, and the terrain complexity of each point is computed. Then, an enhanced particle motion model appropriate for ground filtering is introduced. To improve filtering performance, we divide the point-set surface into several subregions and set adaptive parameters for diverse subregions according to the terrain complexity. Finally, subregional point cloud filtering is performed, and each bathymetric point is marked as "terrain" or "nonterrain". To assess the filtering performance, SPSF is applied to process the multibeam bathymetric point cloud data collected around Chi Island in Qingdao, China. Results indicate that compared with the same type of filtering methods, SPSF improves the filtering accuracy by more than 50% and the filtering efficiency by more than 40%. Furthermore, compared with other types of filtering methods, SPSF displays great potential in alleviating over- and under-filtering in seafloor terrains.

Keywords Multibeam bathymetric point cloud; Underwater digital terrain model; Ground filtering; Smoothed particle simulation

1 Introduction

The measurement results of submarine surveying have crucial application value in many fields, such as marine scientific research, engineering construction, resource investigation and development, and underwater archaeology (Cutter, 2005; Pratt, 2007; Vicki, 2004; Zhou et al., 2022). In the previous decade, with the advancement of computer science and sonar technology, the multibeam bathymetric system (MBS) has become the foremost equipment for obtaining underwater bathymetric data (Xu et al., 2020; Zhang et al., 2022; Liang et al., 2024). Constructing an underwater digital terrain model (UDTM) (Yan et al., 2022) from underwater bathymetric data is a significant application of MBS.

Although the vast majority of multibeam bathymetric data are of high quality after a series of postprocessing and data quality evaluations, data acquisition is inevitably influenced by inhomogeneous water media, sensor errors, positioning error of the bathymetric point, and secondary echoes from the seafloor, resulting in a certain number of outliers in bathymetric data. Detection methods related to outliers can be classified into manual and automatic. Traditional manual detection methods typically entail a heavy workload for users and are therefore rarely employed. Automatic detection method is a long-term research hotspot for marine equipment manufacturers and researchers; examples include median filtering (Mann et al., 2001), sur-

Article Highlights

- Inspired by smoothed particle hydrodynamics (SPH) and cloth simulation filtering (CSF), a new filtering method for multibeam bathymetric data is proposed.
- The partition optimization method is introduced to produce several subregional terrains.
- The terrain complexity, composed of four terrain features, is calculated to set the filtering parameters for subregions.
- The seafloor terrain data measured in China's sea areas are used to verify the proposed method.

✉ Weidong Du
duweidong@hrbeu.edu.cn

[1] National Key Laboratory of Underwater Acoustic Technology, Harbin Engineering University, Harbin 150001, China

[2] Key Laboratory of Marine Information Acquisition and Security (Harbin Engineering University), Ministry of Industry and Information Technology, Harbin 150001, China

[3] College of Underwater Acoustic Engineering, Harbin Engineering University, Harbin 150001, China

[4] School of Information Engineering, Yancheng Institute of Technology, Yancheng 224051, China

face filtering (Mitchell, 1996), M-estimate filtering (Rezvani et al., 2015), and the combined uncertainty and bathymetric estimator (CUBE) method (Calder and Mayer, 2003). CUBE is extensively known for its high efficiency and robustness. CUBE uses Bayesian estimation to solve the seafloor trend surface recursively and leverages the vertical and horizontal uncertainties of bathymetric points. However, CUBE has numerous parameters and a tedious configuration. If the data quality is poor, the filtering effect becomes unsatisfactory.

In addition to points with anomalous bathymetric values, fish and plankton in the water, vegetation patches, and large artificial structures on the seafloor, such as pipelines (Guan et al., 2019) and shipwrecks (Palomer et al., 2015), are strong interference sources in UDTM modeling. Currently, filtering methods for these interferences are few. Ground filtering that distinguishes the seafloor terrain points (STP) from the seafloor nonterrain points (SNP) may be a good choice. Ground filtering is broadly used for processing airborne Light Detection and Ranging (LiDAR) point clouds. In recent years, developed methods can be grouped into three types, namely slope-based, morphology-based, and grid model-based.

Slope-based filters assume a substantial drop between manufactured buildings and the ground. Based on the assumption, Vosselmann (2000) determined whether a point is a ground point or a nonground point by comparing its slope with that of neighbors. Shan and Aparajithan (2005) improved the efficiency of slope calculation by specifying the scanning direction. Generally, setting an appropriate slope threshold for complex terrains is difficult. Susaki (2012) and Wang and Tseng (2010) developed adaptive and bidirectional slope-based filters to address this issue, respectively. However, many research results reveal that the slope-based filter cannot maintain good performance in complex terrains (Liu, 2008), and filtering accuracy decreases with increasing slope.

Mathematical morphology-based filters perform open and closed operations to remove nonground points. The filtering result depends on the filtering window size. Progressive morphological filter (PMF), proposed by Zhang et al. (2003), iteratively increases the filtering window size. Rodríguez-Caballero et al. (2016) designed a dynamic adjustment method for window size. Although adjusting the filtering window size can improve accuracy, it sacrifices efficiency. Compared with some slope-based filters, morphology-based filters are more robust and accessible. However, finding the optimal window size takes a large amount of time. Thus, providing additional terrain priors is necessary.

Grid model-based filters typically construct a particle grid to approximate the ground surface and to extract nonground points by comparing the grid model and the terrain surface. For example, the adaptive triangulated irregular network (TIN) starts from the selected seed and iteratively encrypts a sparse triangular network (Axelsson, 2000). The TIN-based filter is sensitive to two parameters. The first is the distance threshold from the candidate ground point to the TIN facet. The second is the angle between the TIN facet and the line connecting the candidate ground point and the nearest point on that TIN facet. Recently, methods based on physical models have been proposed, such as the cloth simulation filtering (CSF) proposed by Zhang et al. (2016), which constructs a particle grid with cloth properties. This method is suitable for airborne LiDAR point clouds in most terrestrial landscapes. However, when handling large-scale targets (low-rise buildings) with minor relative height differences from the ground, particles may adhere to the target surface, which leads to misclassification. To solve these problems, Cai and Yu (2023) improved CSF using a multidirectional narrow window that employs a moving least-square (MLS) plane to reconstruct the terrain and uses a progressively refined method to extract terrain points. This improved CSF's filtering accuracy is high in forest sites with different terrain and vegetation distributions. Moreover, Cai et al. (2023) presented an improved CSF, which applies the closing operation to initialize the cloth model and designs an adaptive height difference threshold. The initialized cloth grid can cover terrain slopes. With this approach, raised terrains can be effectively detected, while the cloth rigidness is kept to a minimum. In recent years, researchers have proposed several filtering methods based on physical models for bathymetric data. Yang et al. (2020) proposed bilateral cloth simulation filtering (BCSF), which is applied to process airborne LiDAR bathymetric point clouds. This method creates forward- and inverse-filtered surfaces by performing CSF twice. Lv et al. (2023) developed the improved cone model filtering (CMF). This method produces a virtual grid and introduces voxel downsampling. CMF can alleviate overfiltering in areas with complex terrain.

The above filtering methods have been proven effective. However, filtering performance is related to terrain features. Moreover, due to the differences in measurement principles, data acquisition, and postprocessing between acoustic bathymetric and optical LiDAR, the feasibility of applying the above methods to bathymetric data still needs to be verified.

In this paper, we develop a filtering method based on a statistical physical model, that is, smoothed particle simulation filtering (SPSF). Inspired by smoothed particle hydrodynamics (SPH; Gingold and Monaghan, 1977) and BCSF (Yang et al., 2020), SPSF creates a smoothed particle grid to match the point-set surface and establishes a collision detection and correction model between the particles and the point-set surface to move particles. In addition, we create an MLS fitting surface for the original

point cloud and extract four terrain features on the MLS surface. To improve filtering accuracy, we set adaptive parameters for subregions based on terrain complexity. The remaining parts of this paper are organized as follows. Section 2 presents the foundation of SPH and the procedure of SPSF. Section 3 provides the measured data, results, and effects. Section 4 discusses the selection of filtering parameters. Section 5 presents the conclusion.

2 Materials and methods

2.1 Foundation of SPH

SPH is a numerical simulation method based on integral interpolation theory, and is widely used for solving statistical physical problems in astrophysics, fluid mechanics, solid mechanics, and other fields (Le Touzé and Colagrossi, 2025). SPH uses the field quantities defined at discrete particle positions to assess the field quantities at any position in the physical space. Assuming that virtual particles are abstract entities in a continuous field, SPH naturally uses symmetric smoothing kernels to extend the physical properties of particles to an interaction domain, as illustrated in Figure 1.

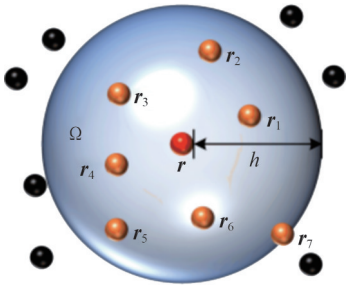


Figure 1 Schematic of the smoothing kernel and interaction domain Ω

As shown in Figure 1, a particle is located at the position $\boldsymbol{r}$ and is surrounded by K neighboring particles inside the interaction domain Ω (with $K = 7$ in this case), and $\boldsymbol{r}_k$ is the position vector of the k-th neighboring particle. In SPH, the physical quantity φ of a particle is provided by the weighted sum of contributions from all neighboring particles. The expression is given as follows:

$$\varphi(\boldsymbol{r}) = \sum_{k=1}^{K} m_k \frac{\varphi_k}{\rho_k} W(\boldsymbol{r} - \boldsymbol{r}_k, h) \tag{1}$$

where m_k is the mass of the k-th neighboring particle; ρ_k and φ_k are the density and physical quantity at $\boldsymbol{r}_k$, respectively; W is a smoothing kernel function, and h is the smoothing length. In our method, we refer to Müller et al. (2003) to set W.

2.1.1 Governing Equations

In the proposed method, we use the Euler equation to describe particle motion (Becker and Teschner, 2007). The Euler equation focuses on velocity, density, and pressure fields, depending on a continuity equation and a momentum equation. The continuity equation assures the conservation of mass and is expressed as follows:

$$\nabla \cdot \boldsymbol{v} = 0 \tag{2}$$

where $\boldsymbol{v}$ is the velocity field; ∇ and $\nabla\cdot$ are the gradient and divergence operators, respectively. In our method, the masses of all particles are identical, so the conservation of mass can be satisfied. The momentum equation is given as follows:

$$\rho \frac{\mathrm{D}\boldsymbol{v}}{\mathrm{D}t} = -\nabla P + \rho \boldsymbol{g} \tag{3}$$

where ρ is the density field, P is the pressure field, and $\boldsymbol{g}$ is the external force density field. Equation (3) is used for inviscid fluid and ensures the conservation of momentum. On the left side of Eq. (3), $\mathrm{D}\boldsymbol{v}/\mathrm{D}t = \partial \boldsymbol{v}/\partial t + \boldsymbol{v} \cdot \nabla \boldsymbol{v}$ is the substantial derivative of $\boldsymbol{v}$ that consists of the local derivative $\partial \boldsymbol{v}/\partial t$ and the convective derivative $\boldsymbol{v} \cdot \nabla \boldsymbol{v}$. In the proposed particle motion model, the velocity field is uniform, so $\boldsymbol{v} \cdot \nabla \boldsymbol{v} = 0$ and $\mathrm{D}\boldsymbol{v}/\mathrm{D}t = \partial \boldsymbol{v}/\partial t$. The right side of Eq. (3) has two force density fields modeling pressure, namely $-\nabla P$ and external force $\rho \boldsymbol{g}$, that determine the momentum variation of particles. The acceleration vector of the particle i is expressed as follows:

$$\boldsymbol{a}_i = \partial \boldsymbol{v}_i / \partial t = \boldsymbol{F}_i / \rho_i \tag{4}$$

where $\boldsymbol{v}_i$ is the velocity vector of particle i; ρ_i and $\boldsymbol{F}_i$ are the density field and the force density field evaluated at the position of particle i, respectively; $\boldsymbol{F}_i = \boldsymbol{F}_i^{\text{pressure}} + \boldsymbol{F}_i^{\text{external}}$ is composed of a pressure density field $\boldsymbol{F}_i^{\text{pressure}}$ and an external force density field $\boldsymbol{F}_i^{\text{external}}$. Next, we utilize a numerical modeling method (Müller et al., 2003) to obtain the density and pressure density terms.

2.1.2 SPH modeling

We use a familiar summation expression to estimate the density of particle i, as follows:

$$\rho_i(\boldsymbol{r}_i) = \sum_{j \in \Omega} m_j \frac{\rho_j}{\rho_j} W(\boldsymbol{r}_i - \boldsymbol{r}_j, h) = \sum_{j \in \Omega} m_j W_{\text{Poly6}}(\boldsymbol{r}_i - \boldsymbol{r}_j, h) \tag{5}$$

$$W_{\text{Poly6}}(\boldsymbol{r}_i - \boldsymbol{r}_j, h) = \begin{cases} \dfrac{315}{64\pi h^9}\left(h^2 - r_{ij}^2\right)^3, & 0 \leqslant r_{ij} \leqslant h \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

$r_{ij} = \| \boldsymbol{r}_i - \boldsymbol{r}_j \|$, where $\boldsymbol{r}_i$ is the position vector of particle i; m_j and $\boldsymbol{r}_j$ are the mass and position vector of particle j in the Ω, respectively; ρ_j is the density at $\boldsymbol{r}_j$. Smoothing ker-

nel function W_{Poly6} is appropriate for smoothing the density fields of forces other than pressure and viscous forces (Müller et al., 2003).

When only two particles interact, if the pressure at their positions is not equal, the pressure is usually asymmetric. Therefore, we adopt Eq. (7) to model the pressure density field. Eq. (7) computes the average pressure, which is a solution that pursues speed and stability, as follows:

$$\boldsymbol{F}_i^{\text{pressure}} = -\nabla P_i(\boldsymbol{r}_i) = -\sum_{j\in\Omega} m_j \frac{P_i + P_j}{2\rho_j} \nabla W_{\text{Spiky}}(\boldsymbol{r}_i - \boldsymbol{r}_j, h) \tag{7}$$

$$W_{\text{Spiky}}(\boldsymbol{r}_i - \boldsymbol{r}_j, h) = \begin{cases} \dfrac{15}{\pi h^6}(h - r_{ij})^3, & 0 \leqslant r_{ij} \leqslant h \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where P_i and P_j are the pressures at $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$, respectively. Desbrun and Gascuel (1996) stated that the smoothing kernel function W_{Spiky} has a nonvanishing gradient near the kernel center.

For pressure, the Poisson equation is a standard solution. However, solving this equation is very time-consuming. Thus, we employ a more efficient equation, namely the ideal gas equation (Desbrun and Gascuel, 1996), to estimate pressure.

$$P = k_P(\rho - \rho_0) \tag{9}$$

where ρ_0 is the reference density, and k_P is the pressure constant. Equation (9) is computationally simple and efficient, so it is a good choice for our method.

In summary, Eq. (4) is rewritten as follows:

$$\boldsymbol{a}_i = \sum_{j\in\Omega} m_j \frac{45}{\pi h^6} \left(\frac{P_i + P_j}{2\rho_i \rho_j} (h - r_{ij})^2 \frac{\boldsymbol{r}_i - \boldsymbol{r}_j}{r_{ij}} \right) + \boldsymbol{g}_i \tag{10}$$

where $\boldsymbol{g}_i$ is the acceleration vector of the external force. If the external force is gravity, $\boldsymbol{g}_i$ is the gravitational acceleration vector.

2.1.3 Artificial viscosity

In numerical simulation, artificial viscosity is frequently used to increase computational stability (Schechter and Bridson, 2012). Based on this, we introduce a velocity field smoothing term inspired by extended SPH (Monaghan, 2005), denoted as XSPH, which has a stabilizing effect similar to artificial viscosity. This term is expressed as follows:

$$\boldsymbol{v}_i' = \boldsymbol{v}_i + \varepsilon \sum_{j\in\Omega} m_j \frac{\boldsymbol{v}_j - \boldsymbol{v}_i}{\rho_j} W_{\text{Viscosity}}(\boldsymbol{r}_i - \boldsymbol{r}_j, h) \tag{11}$$

$$W_{\text{Viscosity}}(\boldsymbol{r}_i - \boldsymbol{r}_j, h) = \begin{cases} \dfrac{15}{2\pi h^3} \left(-\dfrac{r_{ij}^3}{2h^3} + \dfrac{r_{ij}^2}{h^2} + \dfrac{h}{2r_{ij}} - 1 \right), & 0 \leqslant r_{ij} \leqslant h \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are the velocity vectors of the particle i and j, respectively; $\boldsymbol{v}_i'$ is the new velocity vector of particle i; $0 \leqslant \varepsilon \leqslant 1$ is an adjustable parameter, as explained in Section 2.2.1. The smoothing kernel function $W_{\text{Viscosity}}$ has a smoothing effect on the velocity field (Müller et al., 2003). This term ensures the symmetry of viscosity by computing the velocity difference of interacting particles.

2.2 SPSF

In our method, we use a simple physical model to describe the motion of particles. During this process, the particles attach to the point-set surface subjected to external and internal forces. Assuming that all particles are inviscid, the particle grid falls from above the bathymetric point cloud, and its final shape is a digital surface model including seafloor targets and terrain. However, if the point cloud is inverted and a particle grid with viscosity is used to match the point-set surface, then the final shape of the particle grid can be directly considered a UDTM with targets removed, as illustrated in Figure 2.

In Figure 2, the point-set surface of the bathymetric point cloud is first inverted, and then all the points are projected onto a plane above the inverted surface along yellow arrows. In this projection plane, a particle grid is constructed according to the surface density of the point cloud. Moreover, each point has a corresponding particle, which is the particle closest to this point in the plane, as explained in Section 2.2.2. Next, particles move along the green arrows under an external force. Once this grid contacts the point-set surface, the position of particles changes under the action of an internal force until the simulation is terminated. Finally, the height difference H between the point and the corresponding particle is computed and compared with a height threshold δ_h. If $H > \delta_h$, this point is classified as "nonterrain". If $H \leqslant \delta_h$, this point is classified as "terrain".

2.2.1 Parameter Optimization

Before filtering, creating a particle grid G and initializing the particle properties are essential. In our method, G is formulated as follows:

$$G = \left\{ X_l + \left(n - \left\lfloor \frac{n}{N_x} \right\rfloor \times N_x - 1 \right) \times \lambda, Y_l + \left\lfloor \frac{n}{N_x} \right\rfloor \times \lambda, Z_h \right\}$$
$$N_G = N_x \times N_y, N_x = \left\lceil (X_h - X_l)/\lambda \right\rceil, N_y = \left\lceil (Y_h - Y_l)/\lambda \right\rceil \tag{13}$$

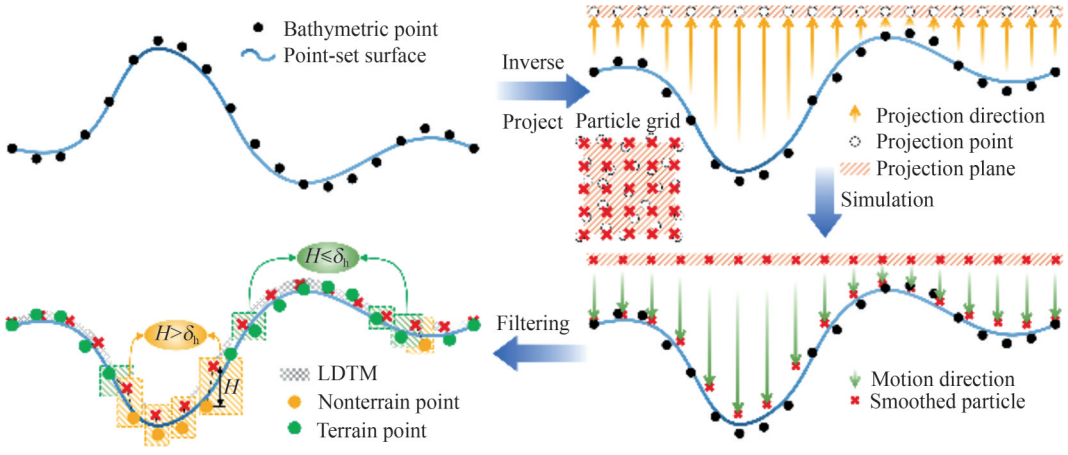


Figure 2 Diagram of the physical simulation

where $n \in \mathbb{Z}^+$ and $n \leqslant N_G$. X_{min} and X_{max} are the minimum and maximum of the point-set surface in the X direction, respectively. Y_{min} and Y_{max} are the minimum and maximum in the Y direction, respectively. Z_{max} is the minimum in the Z direction. Parameter λ is the grid spacing. $N_G = N_x \times N_y$ represents the total number of particles, with the number of particles in the X and Y directions, denoted as N_x and N_y, respectively. The down and up rounding operators are illustrated by $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$, respectively.

Two terrains with 100 × 100 m^2 are taken as examples, and their particle grids are presented in Figure 3.

As shown in Figure 3, the particle grid and the projection plane are above the point-set surface. $X_{min} = Y_{min} = 0$ m, $X_{max} = Y_{max} = 100$ m, $N_x = N_y = 21$, and $N_G = 441$. The Z_h of the two terrains are different. $Z_{max} = -45.24$ m in Figure 3(a), and $Z_{max} = -46.90$ m in Figure 3(b). To ensure that each particle has at least one corresponding point, we set the parameter λ according to the point cloud density, as discussed in Section 3.3. As shown in the windows located in the upper right corner of Figure 3, $\lambda = 5$ m is appropriate for both seafloor terrains.

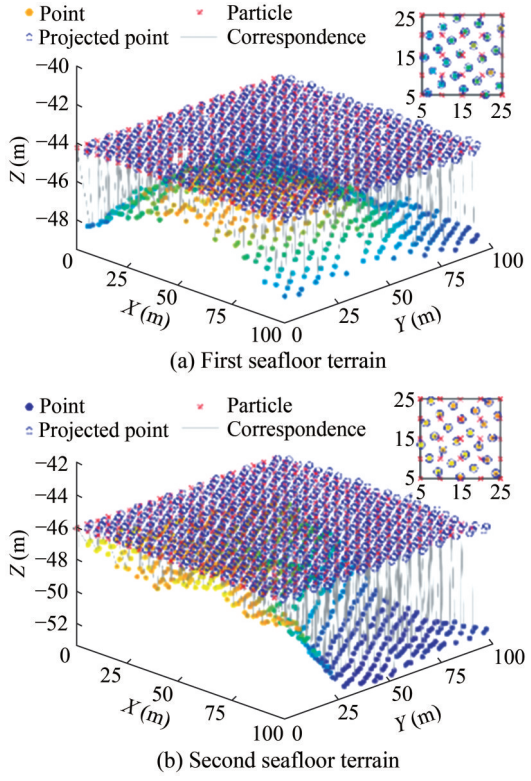


Figure 3 Schematics of particle grids

In our method, some custom parameters must be set, as listed in Table 1.

Table 1 Description of custom parameters

Parameters	Description
Grid spacing λ	Horizontal spacing between two adjacent particles
Viscosity coefficient ε	Control factor for artificial viscosity
Time step τ	Particle motion time in a single iteration
Height difference threshold δ_h	Criteria for determining seafloor and nonseafloor points

To improve accuracy, we formulate the selection criteria for parameter ε and provide an adaptive calculation method for parameter δ_h. In addition, we set model parameters before filtering. The model parameters and other custom parameters are set, as discussed in Section 3.3.

In numerical calculation, artificial viscosity adjusts the velocity of particles and provides the interaction force between particles. For complex terrains, the movement of the particles must be constrained to prevent the particle grid from adhering to the point-set surface and causing underfiltering; therefore, an interaction force is introduced. For flat terrains, most of the points are terrain points; the higher the matching accuracy between the particle grid and the point-set surface, the better the filtering effect. Hence, the interaction force can be ignored. Based on this, we formulate the selection criteria for the viscosity coeffi-

cient according to the terrain complexity. First, an MLS fitting surface is constructed, and then four terrain features are extracted to calculate the terrain complexity of each point. Finally, the viscosity coefficient ε is determined by subregional terrain complexity, which is the average terrain complexity within a subregion.

In the proposed method, the implicit function $g(\boldsymbol{p}')$ of the MLS fitting surface (Yang and Qian, 2007) is given as follows:

$$g(\boldsymbol{p}') = \boldsymbol{n}(\boldsymbol{p})^{\mathrm{T}}\left(\partial e\left(\boldsymbol{q}, \boldsymbol{n}(\boldsymbol{p})\right)\big/\partial \boldsymbol{q}\big|_{\boldsymbol{q}=\boldsymbol{p}'}\right) \tag{14}$$

where $\boldsymbol{p}'$ is the projection point of point $\boldsymbol{p}$ along the normal on the MLS surface, $\boldsymbol{q}$ is a candidate projection point, $\boldsymbol{n}(\boldsymbol{p})$ is the normal vector at $\boldsymbol{p}$, and e is an energy function. $\boldsymbol{p}'$ causes e to reach the local minimum. Goldman (2005) provided curvature calculation formulas. Gaussian curvature k_{Gauss} and average curvature k_{mean} are expressed as follows:

$$k_{\mathrm{Gauss}}(\boldsymbol{p}') = \det\begin{pmatrix} \mathbf{H}\left(g(\boldsymbol{p}')\right) & \nabla^{\mathrm{T}} g(\boldsymbol{p}') \\ \nabla g(\boldsymbol{p}') & 0 \end{pmatrix} \Big/ \left\| \nabla g(\boldsymbol{p}') \right\|^4 \tag{15}$$

$$k_{\mathrm{mean}}(\boldsymbol{p}') = \begin{Bmatrix} \left\| \nabla g(\boldsymbol{p}') \right\|^2 \cdot Tr\left(\boldsymbol{H}\left(g(\boldsymbol{p}')\right)\right) - \\ \nabla g(\boldsymbol{p}') \cdot \boldsymbol{H}\left(g(\boldsymbol{p}')\right) \cdot \nabla^{\mathrm{T}} g(\boldsymbol{p}') \end{Bmatrix} \Big/ \left\| \nabla g(\boldsymbol{p}') \right\|^3 \tag{16}$$

where det(**A**) and Tr(**A**) are the determinant and trace of the matrix **A**, respectively. $\nabla g(\boldsymbol{p}')$ and $\boldsymbol{H}(g(\boldsymbol{p}'))$ are the gradient and Hessian matrix of $g(\boldsymbol{p}')$, respectively, as follows:

$$\nabla g(\boldsymbol{p}') = \left(\partial g(\boldsymbol{p}')/\partial x' \quad \partial g(\boldsymbol{p}')/\partial y' \quad \partial g(\boldsymbol{p}')/\partial z'\right)^{\mathrm{T}} \tag{17}$$

$$\boldsymbol{H}\left(g(\boldsymbol{p}')\right) = \nabla\left(\nabla g(\boldsymbol{p}')\right) = \begin{pmatrix} \frac{\partial^2 g(\boldsymbol{p}')}{\partial x' \partial x'} & \frac{\partial^2 g(\boldsymbol{p}')}{\partial x' \partial y'} & \frac{\partial^2 g(\boldsymbol{p}')}{\partial x' \partial z'} \\ \frac{\partial^2 g(\boldsymbol{p}')}{\partial y' \partial x'} & \frac{\partial^2 g(\boldsymbol{p}')}{\partial y' \partial y'} & \frac{\partial^2 g(\boldsymbol{p}')}{\partial y' \partial z'} \\ \frac{\partial^2 g(\boldsymbol{p}')}{\partial z' \partial x'} & \frac{\partial^2 g(\boldsymbol{p}')}{\partial z' \partial y'} & \frac{\partial^2 g(\boldsymbol{p}')}{\partial z' \partial z'} \end{pmatrix} \tag{18}$$

where $\boldsymbol{p}' = (x', y', z')^{\mathrm{T}}$, (x', y') is the position coordinate, and z' is the bathymetric value.

The maximum and minimum curvatures, namely k_1 and k_2, respectively, are given as follows:

$$k_1(\boldsymbol{p}') = k_{\mathrm{mean}}(\boldsymbol{p}') + \sqrt{k_{\mathrm{mean}}^2(\boldsymbol{p}') - k_{\mathrm{Gauss}}(\boldsymbol{p}')} \tag{19}$$

$$k_2(\boldsymbol{p}') = k_{\mathrm{mean}}(\boldsymbol{p}') - \sqrt{k_{\mathrm{mean}}^2(\boldsymbol{p}') - k_{\mathrm{Gauss}}(\boldsymbol{p}')} \tag{20}$$

On the MLS fitting surface, we extract four local terrain features, namely curvature-based shape factor, slope, roughness, and concave–convex coefficient (Dorai and Jain, 1997; Hollaus and Höfle, 2010; Wu et al., 2017; Minár et al., 2020), as detailed in Table 2. These features are calculated within the neighborhood with radius r.

Table 2 Description of terrain features

Symbol	Description
η	Curvature-based shape operator at an arbitrary point
φ	Terrain slope at an arbitrary point
R	Terrain roughness at an arbitrary point
C	Concave–convex coefficient at an arbitrary point

Curvature-based shape factor is a measure of terrain deformation and is defined as follows:

$$\eta(\boldsymbol{p}') = \frac{1}{2} - \frac{1}{\pi}\arctan\frac{k_1(\boldsymbol{p}') + k_2(\boldsymbol{p}')}{k_1(\boldsymbol{p}') - k_2(\boldsymbol{p}')} \tag{21}$$

where $\eta \in [0, 1]$. If the MLS surface is convex at point $\boldsymbol{p}'$, η is close to 1. If it is concave, η is close to 0.

Terrain slope describes the terrain inclination and is expressed as follows:

$$\varphi(\boldsymbol{p}') = \arctan\sqrt{\left(\partial g(\boldsymbol{p}')/\partial x'\right)^2 + \left(\partial g(\boldsymbol{p}')/\partial y'\right)^2} \tag{22}$$

Terrain roughness reflects terrain relief, and is the ratio of the actual area S_{act} to its projected area S_{proj} and is expressed as follows:

$$R(\boldsymbol{p}') = S_{\mathrm{act}}/S_{\mathrm{proj}} \tag{23}$$

$$S_{\mathrm{act}} = \int_{x_{\min}}^{x_{\max}}\int_{y_{\min}}^{y_{\max}} \sqrt{1 + \left(\partial g(\boldsymbol{p}')/\partial x'\right)^2 + \left(\partial g(\boldsymbol{p}')/\partial y'\right)^2}\,\mathrm{d}x\mathrm{d}y$$

Assuming that the four vertices of the unit grid are given in order, S_{proj} is calculated as follows:

$$S_{\mathrm{proj}} = \frac{1}{2}\left|\begin{matrix}(x_1 y_2 - x_2 y_1) + (x_2 y_3 - x_3 y_2) \\ +(x_3 y_4 - x_4 y_3) + (x_4 y_1 - x_1 y_4)\end{matrix}\right| \tag{24}$$

where (x_i, y_i) is the position coordinate of the i-th vertex.

$$C(\boldsymbol{p}') = \frac{\left|\partial^2 g(\boldsymbol{p}')/(\partial x' \partial x') + \partial^2 g(\boldsymbol{p}')/(\partial y' \partial y')\right|}{\left[1 + \left(\partial g(\boldsymbol{p}')/\partial x'\right)^2 + \left(\partial g(\boldsymbol{p}')/\partial y'\right)^2\right]^{3/2}} \tag{25}$$

The above definition reveals a correlation between the four terrain features. Thus, we use principal component

analysis to create a comprehensive evaluation model for the terrain complexity of points:

$$T_{\text{complex}}(\boldsymbol{p}') = \omega_1\eta(\boldsymbol{p}') + \omega_2\varphi(\boldsymbol{p}') + \omega_3 R(\boldsymbol{p}') + \omega_4 C(\boldsymbol{p}') \tag{26}$$

where ω_i is the evaluation coefficient of terrain features. In Section 3.3, we explain how ω_i is obtained.

In our method, the point-set surface is divided into $M \times N$ subregions, as illustrated in Figure 4.

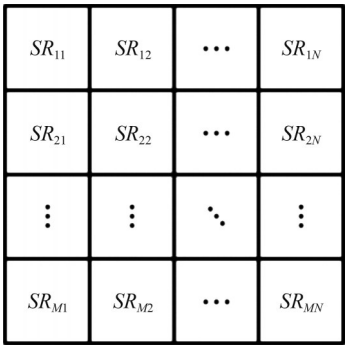


Figure 4 Schematic of subregional distribution

For flat terrains, all particles almost simultaneously contact the point-set surface. Thus, the constraint on particle displacement can be ignored. The particle grid with low viscosity can achieve ground filtering on flat terrains. For complex terrains, particles cannot reach the terrain surface at the same time. The height difference between the particle grid and the point-set surface is used to distinguish between the terrain and nonterrain points, so the constraint on particle displacement is applied. The particle grid with high viscosity is appropriate for ground filtering on complex terrains. We improve filtering accuracy by altering the artificial viscosity according to terrain complexity.

The viscosity coefficient influences the magnitude of artificial viscosity. If it is extremely large, the particle grid has high viscosity. An interaction force between particles exists. If it is overly small, the viscosity of the particle grid is close to 0, and the interaction force between particles can be neglected. Thus, we divide the artificial viscosity into three levels in ascending order of the average terrain complexity, corresponding to three types of terrain, denoted by Ⅰ, Ⅱ, and Ⅲ. Type Ⅰ terrain is flat, whereas type Ⅲ terrain is complex. The selection criteria for the viscosity coefficient are listed in Table 3. In Section 4.1, we provide more details on how these values are obtained.

Height difference threshold δ_h is the criterion for distinguishing between terrain and nonterrain points. If the height difference H between the bathymetric point and the corresponding particle is less than δ_h, the point is classified as an STP; otherwise, it is classified as an SNP.

In our method, a bathymetric quality estimating method is applied to determine the height difference threshold.

Table 3 Selection criteria of viscosity coefficient ε

Terrain type	Viscosity level	Value interval of parameter ε
Ⅰ	Low	0.0–0.3
Ⅱ	Medium	0.3–0.6
Ⅲ	High	0.6–0.9

Combining the underwater terrain features, this method can avoid misclassifying small undulating terrains or aquatic plants (Yang et al., 2020). The height difference threshold of a subregion is defined as follows:

$$\delta_h(m,n) = \sum_{\boldsymbol{p}\in \boldsymbol{SR}_{mn}} \sigma_p \Big/ \boldsymbol{K}_{mn} \tag{27}$$

where σ_p is the weighted standard deviation of point $\boldsymbol{p}$. The number of points within the subregion $\boldsymbol{SR}_{mn}$ is $\boldsymbol{K}_{mn}$. The weighted variance is expressed as follows:

$$\sigma_p^2 = \frac{\sum\limits_{q\in\Omega_p} z_q^2 W(\boldsymbol{p},\boldsymbol{q})}{\sum\limits_{q\in\Omega_p} W(\boldsymbol{p},\boldsymbol{q})} - \left(\frac{\sum\limits_{q\in\Omega_p} z_q W(\boldsymbol{p},\boldsymbol{q})}{\sum\limits_{q\in\Omega_p} W(\boldsymbol{p},\boldsymbol{q})}\right)^2 \tag{28}$$

where Ω_p is the neighborhood with radius r of point $\boldsymbol{p}$, $\boldsymbol{q}$ is the neighbor of $\boldsymbol{p}$, and z_q is the bathymetric value of $\boldsymbol{p}$. $W(\boldsymbol{p}, \boldsymbol{q})$ is a weight distribution function and is defined as follows:

$$W(\boldsymbol{p},\boldsymbol{q}) = \begin{cases} 1 - \|\boldsymbol{p}-\boldsymbol{q}\|/r, & \|\boldsymbol{p}-\boldsymbol{q}\| < r \\ 0, & \|\boldsymbol{p}-\boldsymbol{q}\| \geqslant r \end{cases} \tag{29}$$

2.2.2 Collision detection and correction model

In our method, the simulation time is discretized to enable numerical calculations. The difference between adjacent moments is called the time step and is denoted as τ. To make the particle simulation method suitable for point cloud filtering, we adopt a time integration scheme based on the semi-implicit Euler method to update the velocity vector and position vector of particles (Schechter and Bridson, 2012) as follows:

$$\begin{cases} \boldsymbol{v}_i''(t+\tau) = \boldsymbol{v}_i'(t) + \boldsymbol{a}_i(t)\tau \\ \boldsymbol{r}_i'(t+\tau) = \boldsymbol{r}_i'(t) + \boldsymbol{v}_i'(t+\tau)\tau \end{cases} \tag{30}$$

where the acceleration vector $\boldsymbol{a}_i(t)$ of particle i at time t is given by Eq. (10). If the viscosity effect is ignored, the step-advanced equations of velocity vector and position vector are $\boldsymbol{v}_i(t+\tau) = \boldsymbol{v}_i(t) + \boldsymbol{a}_i(t)\tau$ and $\boldsymbol{r}_i(t+\tau) = \boldsymbol{r}_i(t) + \boldsymbol{v}_i(t+\tau)\tau$, respectively. If artificial viscosity is introduced, $\boldsymbol{v}_i(t+\tau)$ and $\boldsymbol{r}_i(t+\tau)$ should be converted to $\boldsymbol{v}_i'(t+\tau)$ and $\boldsymbol{r}_i'(t+\tau)$, respectively. In Eq. (30), $\boldsymbol{v}_i''(t+\tau)$ is the uncorrected velocity vector at time $t+\tau$. $\boldsymbol{v}_i'(t+\tau)$ is the sum of $\boldsymbol{v}_i''(t+\tau)$ and a velocity adjustment term, as shown

in Eq. (11). $\boldsymbol{v}_i'(t)$ is the updated velocity vector at time t. $\boldsymbol{r}_i'(t+\tau)$ and $\boldsymbol{r}_i'(t)$ denote the updated position vectors at time $t+\tau$ and time t, respectively. The position vectors of particles at an arbitrary time can be computed.

Variable α represents the number of time steps. The displacement of a particle in the vertical direction within a time step is given as follows:

$$l_i(\alpha) = \left[\boldsymbol{r}_i'(t) - \boldsymbol{r}_i'(t-\tau)\right] \cdot \boldsymbol{e}, \alpha = 1, 2, \cdots \tag{31}$$

where $\boldsymbol{e} = (0, 0, 1)^{\mathrm{T}}$ is a unit vector, and $t = \alpha\tau$ is the total motion time of particles.

Before filtering, we determine a corresponding particle for each point, find the nearest projected point for each particle in the projection plane, and record the height of the projected point before projection. If the height of a particle at time t is smaller than the recorded height, this particle is fixed at the recorded height. Conversely, we calculate the displacement difference within adjacent time steps as follows:

$$\begin{aligned} L_i(\alpha, \alpha\boldsymbol{p} - \boldsymbol{q}1) &= l_i(\alpha)\,\boldsymbol{p} - \boldsymbol{q}l_i(\alpha\boldsymbol{p} - \boldsymbol{q}1) \\ &= \left[\boldsymbol{r}_i'(t+\tau) + \boldsymbol{r}_i'(t\boldsymbol{p} - \boldsymbol{q}\tau)\,\boldsymbol{p} - \boldsymbol{q}2\boldsymbol{r}_i'(t)\right] \cdot \boldsymbol{e} \end{aligned} \tag{32}$$

During filtering, if most of the neighboring particles of an unfixed particle are fixed (in contact with the point-set surface), the displacement difference of this unfixed particle is reduced. A displacement difference equivalent to 0 indicates that its neighboring particles no longer provide velocity components to this unfixed particle. In this situation, the artificial viscosity fails, and its movement is not constrained. Therefore, we set a threshold for displacement differences, which terminates the iterations before the maximum displacement difference reaches 0. After each time step, we record the maximum displacement difference L of unfixed particles, called $L_{\max}$. When $L_{\max} \leqslant \delta_l$, all particles are fixed at their current position. Ground filtering is terminated.

2.2.3 SPSF workflow

The detailed steps of the proposed method are illustrated in Figure 5.

The primary implementation steps of SPSF are illustrated as follows.

1) Establish the MLS surface of the point-set surface.

2) For points, extract four terrain features and calculate terrain complexity.

3) Divide the point-set surface into $M \times N$ subregions.

4) For subregions, calculate the average terrain complexity and set filtering parameters.

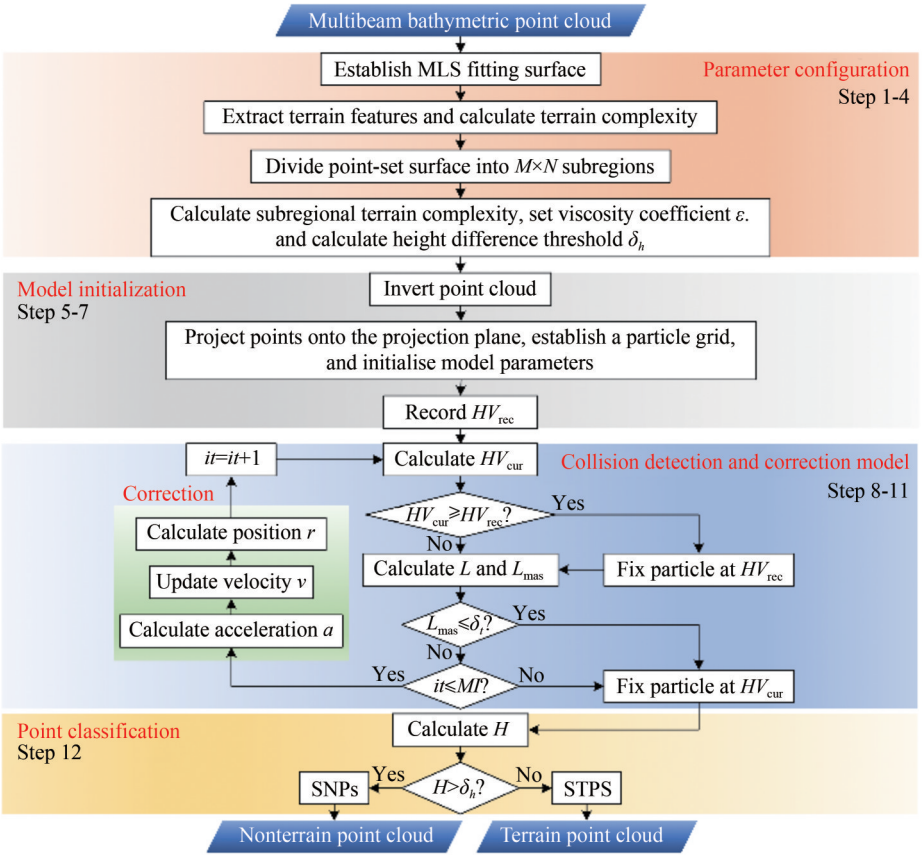


Figure 5 Flowchart of smoothed particle simulation filtering

5) Invert the point-set surface and project points onto the plane above this surface.

6) Create a particle grid and initialize the model parameters.

7) Search for the nearest point for each particle and record its height before projecting, denoted by HV_{rec}.

8) Move a particle to reckon the current height of the particle, denoted by $HV_{rec} = \boldsymbol{r} \cdot \boldsymbol{e}$.

9) Compare HV_{cur} with HV_{rec} after a time step. If $HV_{cur} < HV_{rec}$, fix the particle at HV_{rec}; otherwise, update HV_{cur}.

10) Calculate the L of all unfixed particles and record L_{max}.

11) Repeat (8)–(10). When $L_{max} \leqslant \delta_l$ or iterations reach the preset value, denoted by MI, fix all particles at HV_{cur} and terminate the filtering.

12) Calculate the height difference H between a point and its corresponding particle. If $H > \delta_h$, classify the point as an SNP. If $H \leqslant \delta_h$, classify the point as an STP.

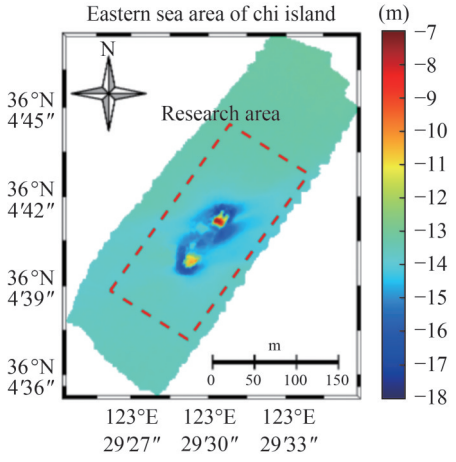


Figure 6 Point cloud of the experimental area: The planimetry and vertical data are WGS_1984_UTM_Zone_51N coordinates and the depth, respectively

3 Experiments

3.1 Data description

We conducted data collection experiments in a sea area near Qingdao Port, China, and used the MBS developed by Harbin Engineering University on an unmanned vehicle to collect bathymetric data. In the experiments, the velocity of this vehicle was about 6 kn ≈ 3.084 $m \cdot s^{-1}$. The parameter setting of the MBS is listed in Table 4.

Table 4 Parameter settings of MBS

Operating frequency	Coverage sector	Beam width	Number of beams	Ping rate
300 kHz	130°	1° × 1°	512	8 Hz

As shown in Figure 6, the research area includes a shipwreck (75 m in length and 12 m in width, approximately located at 120°29′29.44″E; 36°03′49.21″N) as the primary source of nonterrain points, enclosed by dashed red lines. According to statistics, the point density in this area is 25 points·m^{-2}. The reference data were generated through manual filtering, and each bathymetric point was classified as STP or SNP.

3.2 Evaluation metrics

In our experiments, we adopt four standard metrics in point cloud filtering, namely Type Ⅰ Error (I.E.), Type Ⅱ Error (II.E.), Total Error (T.E.), and Cohen's kappa coefficient κ (Hu et al., 2014) to evaluate the filtering accuracy. STP is set as a positive sample, and SNP is set as a negative sample. These metrics depend on a matrix based on the number of points, as presented in Table 5.

As shown in Table 5, TP (true positive) and TN (true negative) are the correctly classified terrain and nonterrain point sets; FP (false positive) and FN (false negative) are the misclassified terrain and nonterrain points.

Table 5 Point count-based cross matrix for the four metrics

Point Count	Referred STP	Referred SNP
Filtered STP	TP	FN
Filtered SNP	FP	TN

Type Ⅰ Error (I.E.) and Type Ⅱ Error (II.E.) are defined as follows:

$$\text{I.E.} = \text{FP}/(\text{TP} + \text{FP}),\ \text{II.E.} = \text{FN}/(\text{TN} + \text{FN}) \quad (33)$$

where I.E. is the proportion of STP misdiagnosed as SNP in the reference STP set, and II.E. is the proportion of SNP mistakenly detected as STP in the reference SNP set. T.E. is given as follows:

$$\text{T.E.} = (\text{FP} + \text{FN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (34)$$

where T.E. denotes the ratio of misclassifications to the total number of points. In addition, we calculate Cohen's kappa coefficient, which evaluates consistency between categories rather than accidental consistency, as follows:

$$\begin{aligned} \kappa &= [\Pr(a) - \Pr(e)]/[1 - \Pr(e)] \\ \Pr(a) &= (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN}) \\ \Pr(e) &= \left[(\text{TP} + \text{FP}) \times (\text{TP} + \text{FN}) + (\text{TN} + \text{FN}) \times (\text{TN} + \text{FP})\right] \Big/ \left(\text{TP} + \text{TN} + \text{FP} + \text{FN}\right)^2 \end{aligned} \quad (35)$$

where Pr(a) is the proportion of correctly classified points to the total number of points, and Pr(e) is utilized to penal-

ize the classification method for class bias.

In addition to the four metrics, we compute the running time (T_r) to evaluate the filtering efficiency.

3.3 Parameter settings

To ensure that the average terrain complexity of points within the subregion can represent the subregional terrain complexity, we divide the point-set surface into 21 subregions, as shown in Figure 7. The area of the research area is 90 × 210 m^2, and that of the subregion is 30 × 30 m^2.

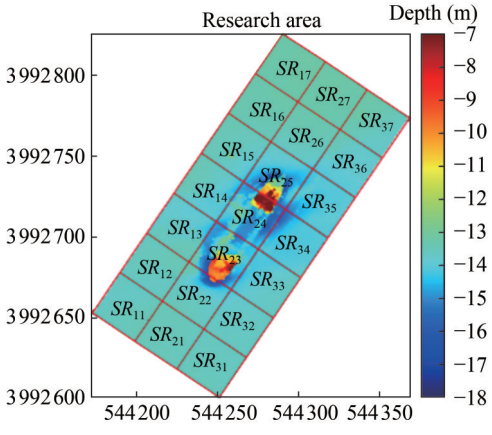


Figure 7 Subregions of the research area

During seafloor feature extractions, the fitting radius of the MLS fitting method is determined by calculating the root mean square error (RMSE) of the depth difference between points and projection points. Let z_i be the depth of an arbitrary point and z_i' be the depth of its projection point on the MLS surface. RMSE is given as follows:

$$\mathrm{RMSE} = \sqrt{\sum_{i=1}^{N}\left(z_i - z_i'\right)^2 \Big/ N} \tag{36}$$

where N is the number of points. The relationship between fitting accuracy and fitting radius is presented in Figure 8. According to the point density, we set the interval of fitting radius from 0.5 m to 1.5 m with a step of 0.1 m.

As shown in Figure 8, the fitting accuracy is highest when r = 1.0 m. Thus, we obtain the MLS fitting surface using the point neighborhood with a radius r = 1.0 m, and extract four terrain features, namely curvature-based shape factor η, terrain slope φ, roughness R, and concave–convex coefficient C, within this neighborhood. The comprehensive evaluation model for terrain complexity is stated as follows:

$$T_{\mathrm{complex}} = 0.252\,9\eta + 0.375\,6\varphi + 0.363\,1R - 0.092\,4C \tag{37}$$

The variation range of subregional terrain complexity in the entire research area is 1.0–10.0, as presented in Figure 9a.

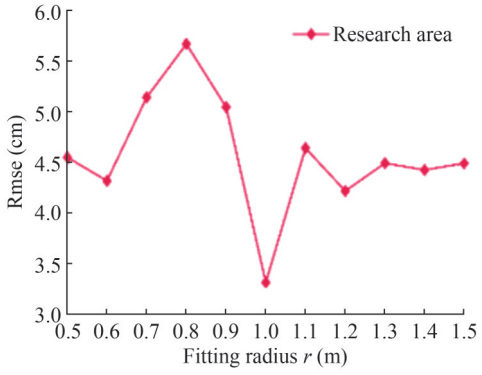


Figure 8 Fitting accuracy under various fitting radii

The subregions are divided into three categories. The value intervals of parameter ε corresponding to different types of subregional terrain are listed in Table 6.

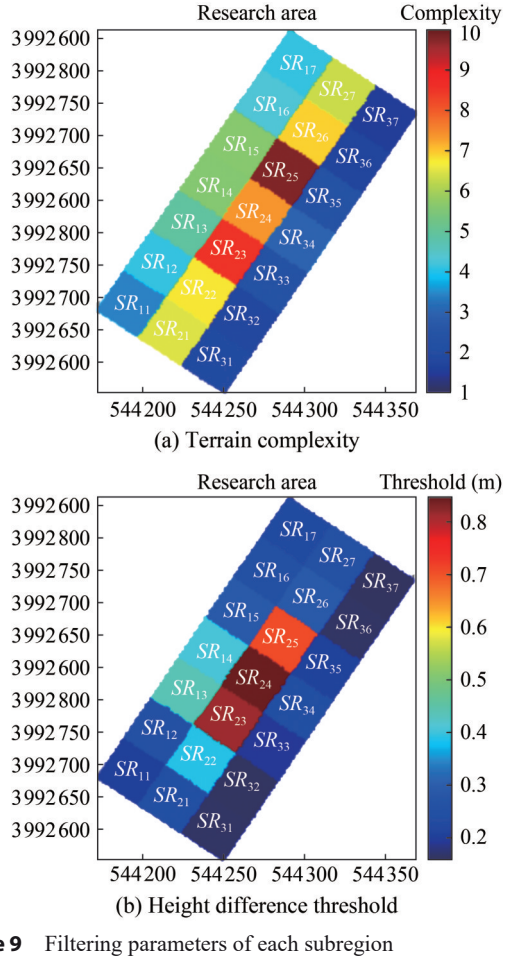


Figure 9 Filtering parameters of each subregion

Table 6 Selection results of viscosity coefficient ε

Terrain type	Subregional terrain complexity	Value interval	Number of subregions
Ⅰ	1.0–4.0	0.0–0.3	8
Ⅱ	4.0–7.0	0.3–0.6	10
Ⅲ	7.0–10.0	0.6–0.9	3

The calculation results of the adaptive height difference threshold are presented in Figure 9b. On flat terrain, the height difference between terrain points and nonterrain points is small. A smaller height difference threshold is more appropriate for flat terrains. On complex terrain, a difference in height is noted between terrain points and nonterrain points. A larger height difference threshold is a better choice.

In addition to these parameters, setting grid spacing λ, time step τ, and maximum number of iterations MI is necessary. In our method, λ is determined by the point density of the horizontal surface. However, during the data collection, the spacing (the distance between adjacent points within the same ping) in the vertical track direction and the spacing (the distance between points of the same beam on different pings) along the track direction are inconsistent, which results in an uneven distribution of points in measurement space. To ensure that a particle corresponds to at least one point, parameter λ should be greater than the minimum distance between adjacent points. As shown in Table 4, the minimum spacing in the vertical track direction is close to $\tan(130°/2) \times 2 \times 18/512 \approx 0.150\,8$ m, and the minimum spacing along the track direction is close to 6 kn/8 Hz ≈ 0.385 5 m. Thus, λ is set to 0.4 m. The analysis and discussion on τ and MI are available in Section 4.

The model parameters for particles are set according to their distribution characteristics, as listed in Table 7. m, ρ_0, and k_P are the mass, reference density, and pressure constant, respectively. In Section 2.1, the physical quantities of a particle are determined by its neighboring particles, which implies that at least one neighboring particle exists within its interaction domain Ω. Thus, ensuring that the smoothing length h is greater than or equal to λ is necessary. Multiple experimental results show that h has a small influence on the filtering accuracy. However, an extremely large h increases the computational amount and decreases the filtering efficiency. Therefore, h is equal to λ to balance accuracy and efficiency in our method.

Table 7 Setting of several model parameters for particles

m (kg)	ρ_0 (kg·m^{-3})	k_P	h (m)
1.0	$1.0/0.4^3 = 15.625$	1.0	0.4

Furthermore, we consider gravity as the external force, and $\boldsymbol{g}$ is the acceleration vector of gravity. Other scalar and vectors are set to 0 and $(0, 0, 0)^{\mathrm{T}}$, respectively.

3.4 Filtering performance analysis

To estimate the filtering performance, we compared it with filtering methods including two morphology-based methods, namely simple morphological filtering (Pingel et al., 2013) and progressive morphological filtering (Tan et al., 2018), as well as three grid model-based methods, namely CSF (Zhang et al., 2016), BCSF (Yang et al., 2020), and improved CSF (ICSF) (Cai et al., 2023).

The operating environment is an Intel Core i5 processor (2.5 GHz) with 16 GB of running memory. In our experiment, SMF is implemented using MATLAB 2024b, whereas other methods are developed using Visual Studio 2022. The performance of morphology-based methods mainly depends on the size of the filtering window. The experimental data show that SMF and PMF exhibit the best filtering performance when the window size is 0.5 m. BCSF and ICSF are derivative methods of CSF. BCSF and ICSF have similar main parameters but different settings, as listed in Table 8.

Table 8 Parameter settings of CSF, BCSF, and ICSF

Parameter	GR (m)	dT	RI	h_{cc} (m)	MI
CSF	0.40	0.65	3	0.30	500
BCSF	0.40	0.50	2	0.40	500
ICSF	0.40	0.65	3	0.30	500

As presented in Table 8, the grid resolution (GR) of BCSF, ICSF, and CSF is equal to the parameter λ of the proposed method. The other parameters, such as time step (dT), rigidness (RI), and distance threshold (h_{cc}), are the optimal settings verified by multiple experiments.

The filtering results and effects are shown in Table 9 and Figure 10, respectively.

Table 9 Filtering results of the entire research area

Method	I.E.	II.E.	T.E.	κ	T_r
SMF	2.28	9.20	2.67	77.93	1.56
PMF	1.91	9.75	2.33	79.35	273.10
CSF	2.13	10.64	2.58	77.42	7.09
BCSF	1.22	13.40	1.87	82.23	12.93
ICSF	0.67	48.73	2.84	60.62	10.65
SPSF	0.30	20.46	**1.21**	**85.00**	**4.07**

As indicated in Table 9, SMF has the shortest running time. However, SMF has many custom parameters, which may not be user-friendly for operators lacking experience. For PMF, selecting an optimal size of filtering window can consume a large amount of computing time. For CSF and its derived methods, the displacement of cloth particles within a time step is influenced by the rigidness of the cloth. Consequently, we set appropriate rigidness parameters for subregional terrains to ensure the simulation accuracy of the cloth simulation. BCSF incorporates two filtering processes: forward and reverse. Although the computing time of BCSF is almost double that of CSF, the T.E. is reduced by 27.52%. This result indicates that BCSF improves the accuracy by costing more computing time. On the one hand, cloth initialization utilizing morphologi-

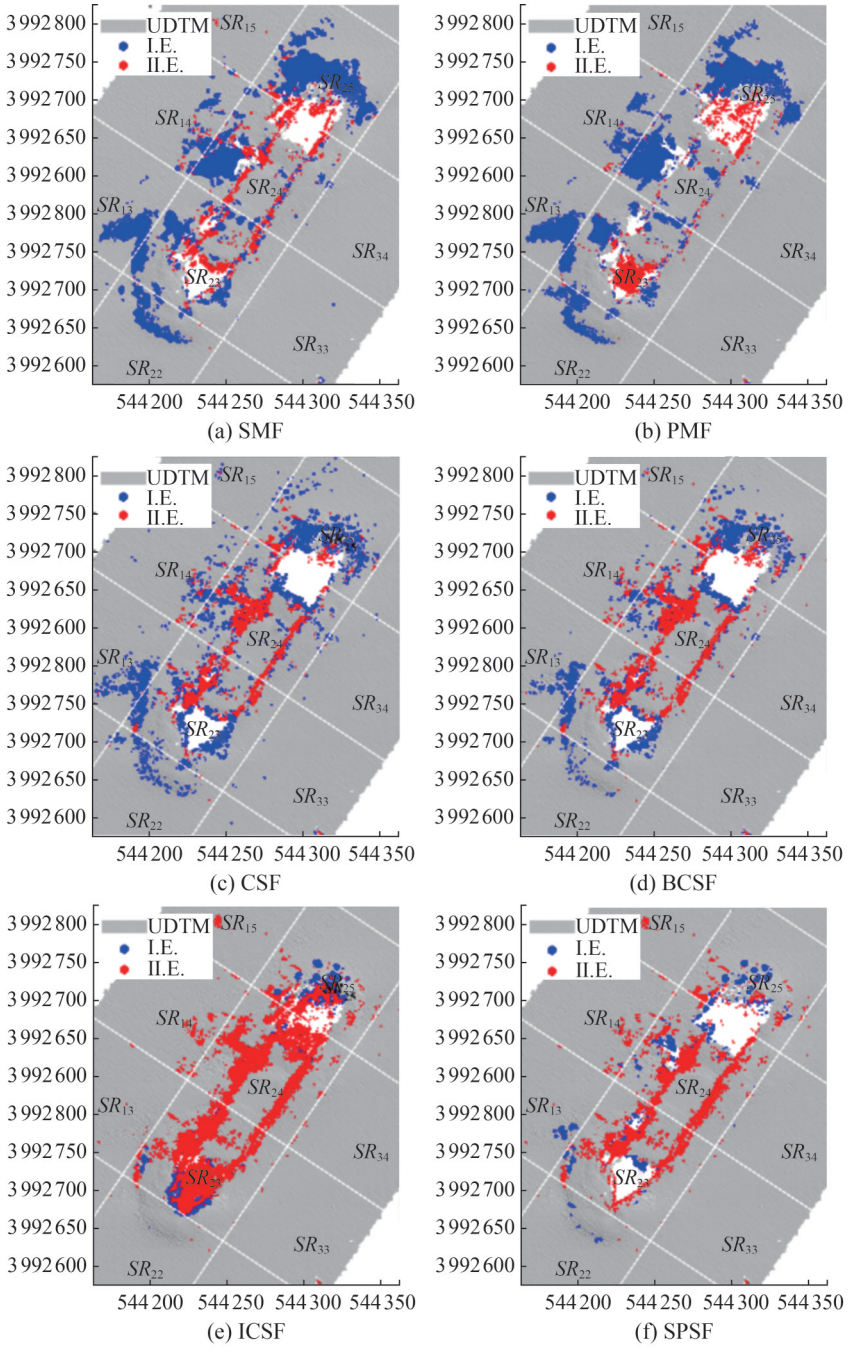


Figure 10 Filtering effects of the comparison methods and the proposed method

cal operations introduces computational burdens in ICSF. On the other hand, multibeam bathymetric data have many anomalous points below the point-set surface and make morphology-based processing methods ineffective. Thus, CSF is superior to ICSF in filtering accuracy and efficiency.

Compared with other methods, SPSF has two advantages. First, it determines the viscosity coefficient and the height difference threshold according to the subregional terrain complexity. Second, the corrected particle motion model ensures that the particle simulation method is suitable for filtering to facilitate matching the particle grid with the point-set surface more accurately and efficiently. As listed in Table 9, the T.E. of SPSF is smaller than other methods, and κ is the highest among them.

As shown in Figure 10, the filtering effects of SMF and PMF are poor compared with those of the other methods. Both methods identify a large number of STPs as SNPs, leading to an increase in I.E. In addition, PMF misclassifies some SNPs as STPs; consequently, its II.E. is greater than that of SMF. CSF and its derivative methods lose some

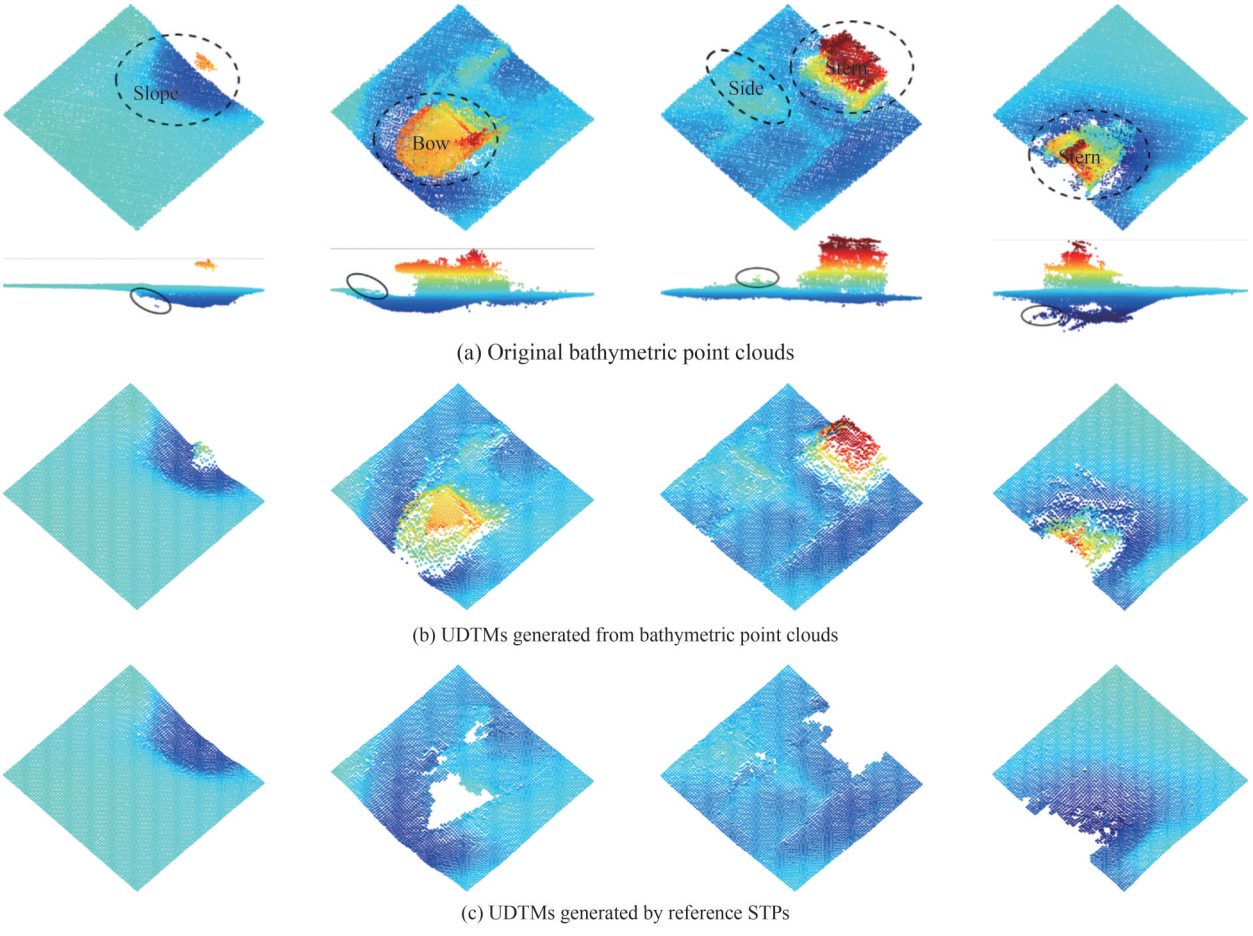
underwater microtopography and affect the terrain integrity. By contrast, SPSF exhibits stronger terrain continuity. The filtering results of these methods on different types of subregional terrain are exhibited in Table 10.

On the same type of terrain, the T.E. of SPSF is always the smallest among these methods. This outcome indicates that SPSF is more robust and stable than CSF and its derivative methods. We select four subregional terrains (SR_{22}, SR_{23}, SR_{24}, and SR_{25}) as references.

These four reference terrains contain different parts of the shipwreck. SR_{22} includes a terrain slope and anomalous bathymetric points generated by seafloor secondary echoes during data collection (solid black circle, Figure 11a). Such anomalies affect the accuracy of UDTM and must be removed. SR_{23} covers the bow and exhibits overlapping bathymetric points caused by splicing errors during data processing (solid black circle, Figure 11b). Overlapping points are stacked on the point-set surface. When they accumulate to a certain quantity or height, they may affect the modeling of UDTM. Therefore, overlapping points should be removed to improve modeling accuracy while preserving terrain details. SR_{24} represents the side and the

Table 10 T.E. (%) of the different types of subregional terrains

Subregion	SR_{33}	SR_{35}	SR_{14}	SR_{15}	SR_{23}	SR_{24}	SR_{25}
Type	Ⅰ		Ⅱ		Ⅲ		
Complexity	2.4188	2.4322	5.5849	5.6237	8.5330	7.3884	9.7914
CSF	0.1800	0.0600	2.8600	0.4800	10.2900	10.8100	10.1500
BCSF	0.0600	0.0400	1.2600	0.1400	8.5000	9.0400	7.5600
ICSF	0.0500	0.0400	0.6500	0.1600	23.2300	16.7400	7.3400
SPSF	**0.0500**	**0.0200**	**0.5100**	**0.0800**	**6.8800**	**8.8100**	**3.6700**



(a) Original bathymetric point clouds

(b) UDTMs generated from bathymetric point clouds

(c) UDTMs generated by reference STPs

Figure 11 Reference subregional terrains SR_{22}, SR_{23}, SR_{24}, SR_{25}

front section of the stern, where the side forms a slight convex ridge (dashed black circles, Figure 11c). SR_{25} contains the rear section of the stern and includes a terrain slope and many anomalous and overlapping points (dashed black circle, Figure 11d). As illustrated in the second row of Figure 11, underwater targets substantially influence terrain modeling, and the UDTM cannot reflect the actual terrain undulations. Thus, the nonterrain points of the original point cloud need to be filtered out. The filtering effects are presented in Figure 12.

For SR_{22}, SPSF continuously expresses seafloor terrain and correctly detects SNPs. CSF and BCSF misclassify some STPs and increase I.E. For SR_{23}, CSF, BCSF, and SPSF can detect the bow. However, CSF and BCSF incorrectly filter out some STPs, which leads to voids in the UDTM, as displayed by the solid yellow circles in the second column of Figure 12. By contrast, SPSF can preserve some terrain details. We conclude that in SR_{24} and SR_{25}, the I.E. of SPSF is small. Although SPSF has a higher II.E. than CSF and BCSF, it filters out most of the overlapping points that influence the modeling accuracy of UDTM, as revealed by the solid yellow circles in the third column of Figure 12. For SR_{25}, the UDTM generated by SPSF is closest to the reference UDTM. SPSF filters out more anomalous points than CSF and BCSF. ICSF has good filtering performance on SR_{22}. For other subregions, ICSF classifies the majority of SNPs as STPs; consequently, the generated UDTM is inconsistent with the reference UDTM.

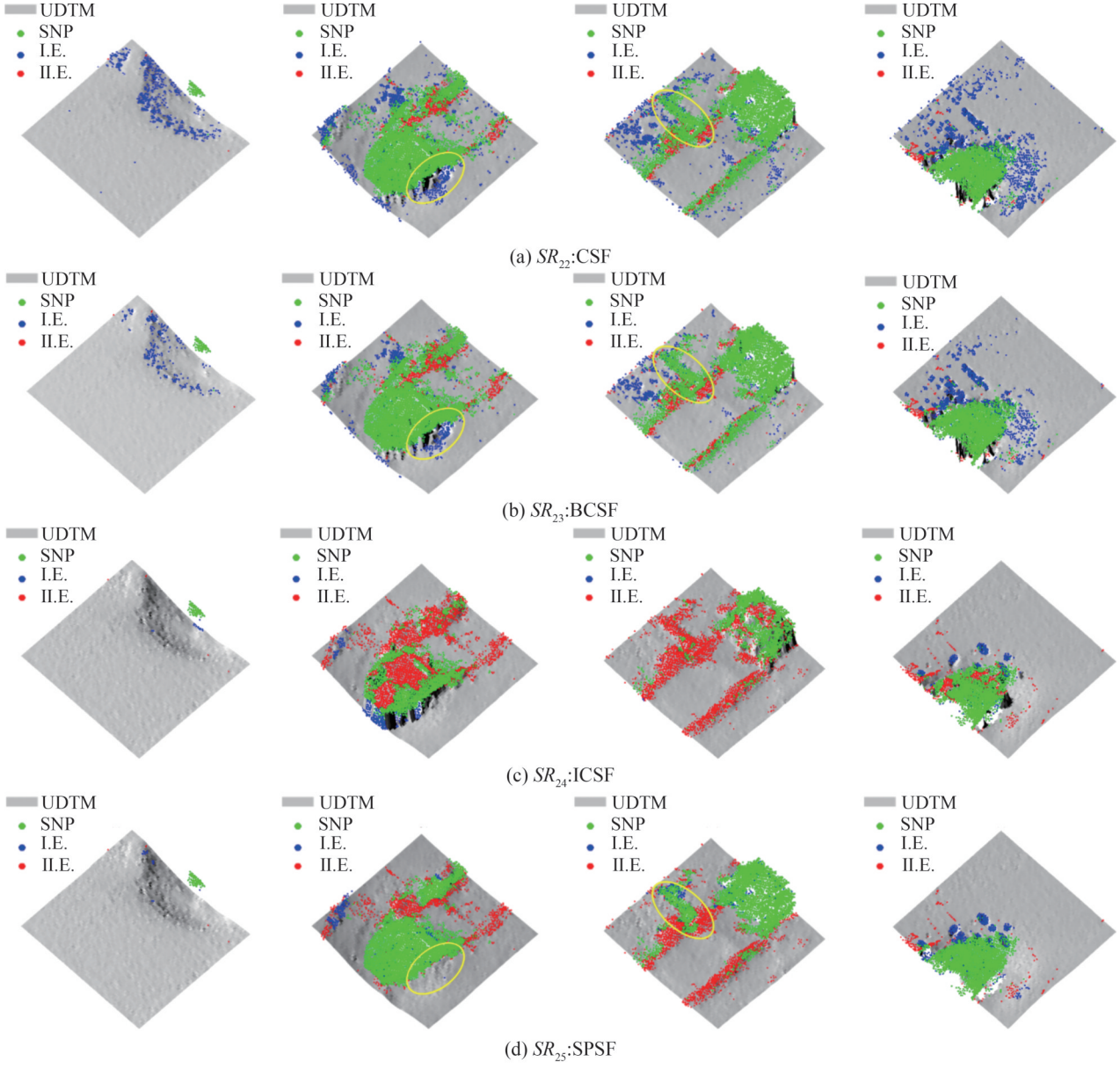


Figure 12 Filtering effects of the four reference subregional terrains

4 Discussion

In Section 2.2.1, we summarize the custom parameters of SPSF, as presented in Table 1. Grid spacing λ is related to point density, and height difference threshold δ_h can be obtained using an adaptive calculation method. Because the viscosity coefficient ε and the time step τ control the particle motion of the particle, they influence the filtering results. This section determines their optimal values to achieve the best filtering effect and analyzes the termination conditions for filtering. Section 4.4 discusses the limitations of SPSF and provides some feasible solutions for these issues.

4.1 Viscosity coefficient

The value interval of the viscosity coefficient ε is provided in Table 3. The T.E. of different types of subregional terrain under various ε are presented in Figure 13.

For Type Ⅰ terrains, the T.E. can achieve the minimum value when $\varepsilon \in [0.0, 0.3)$, as shown in Figure 13a. The small variation in T.E. indicates that the ε has a small influence on the filtering accuracy of flat terrains. For Type Ⅱ terrains, the T.E. can reach the minimum when $\varepsilon \in [0.3, 0.6)$, as presented in Figure 13b. If $\varepsilon < 0.3$, some SNPs are classified as STPs, which leads to overfiltering. If $\varepsilon > 0.6$, many STPs may be classified as SNPs, which results in underfiltering. Both situations increase the T.E. Type Ⅲ terrains need a high ε to increase filtering accuracy, as illustrated in Figure 13c. Thus, the value interval of various viscosities is [0.6, 0.9).

4.2 Time step

According to Eq. (30), time step τ is used to control the position vectors of particles. In theory, the smaller the τ, the less displacement of particles in a single iteration. If τ is extremely small, when the iteration terminates, the particle grid may not contact the point-set surface or may not match the point-set surface well. This situation results in underfiltering. If τ is extremely large, the particle grid may adhere to the point-set surface. This situation leads to the misclassification of some SNPs as STPs and causes overfiltering. Thus, extremely small τ or extremely large τ increases the T.E. The T.E. of different types of subregions under various τ and the average T.E. are shown in Figure 14. Because the T.E. approaches 99.99% when $\tau < 0.3$ s, the value interval of the time step is from 0.3 to 2.0 s with a step of 0.1 s.

The comparison results in Figure 14a indicate that for Type Ⅰ terrains, increasing τ enables rapid contact between the particle grid and the point-set surface. After all particles are fixed, the T.E. is a constant. Type Ⅱ and Type Ⅲ terrains have the same conclusions, as illustrated in Figures 14b and 14c, respectively. Due to the viscous effect, the motion of particles is restricted, so T.E. has slight fluctuations. As shown in Figure 14d, average T.E. reaches the minimum when $\tau = 0.8$ s. Hence, we set τ to 0.8 s to obtain good filtering results without adjusting τ.

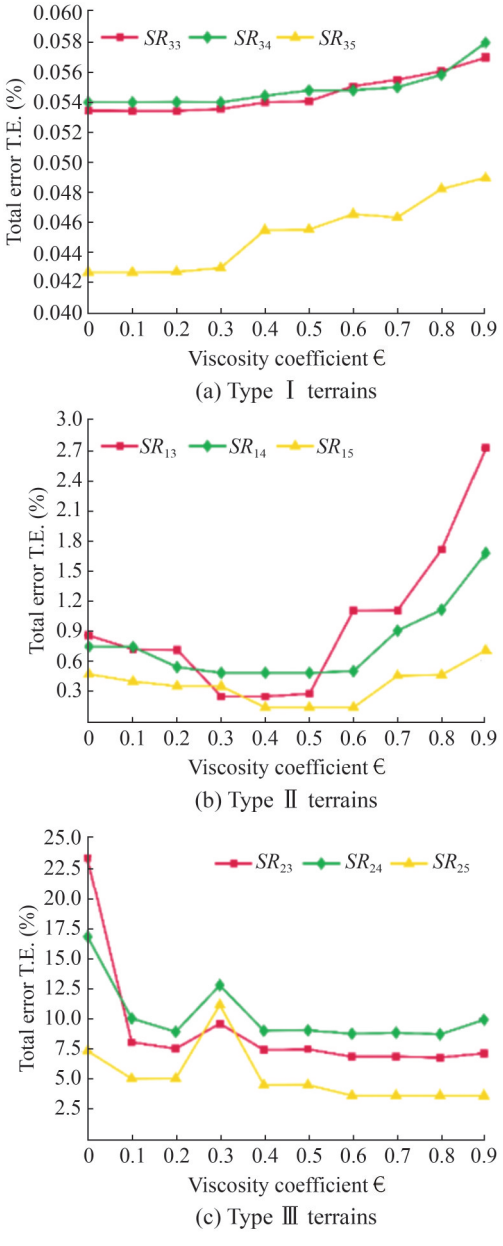


Figure 13 Total Error (T.E.) of subregional terrains under various viscosity coefficients

4.3 Termination conditions

The proposed method has two termination conditions. First, L_{max} is small enough. Second, the number of iterations exceeds the preset value *MI*. Generally, the *MI* is set sufficiently large to prevent filtering failures. Filtering is terminated by the first condition in most situations. Thus, *MI* is set to 200 in our experiment. The curves of the maximum and average displacement differences as the number

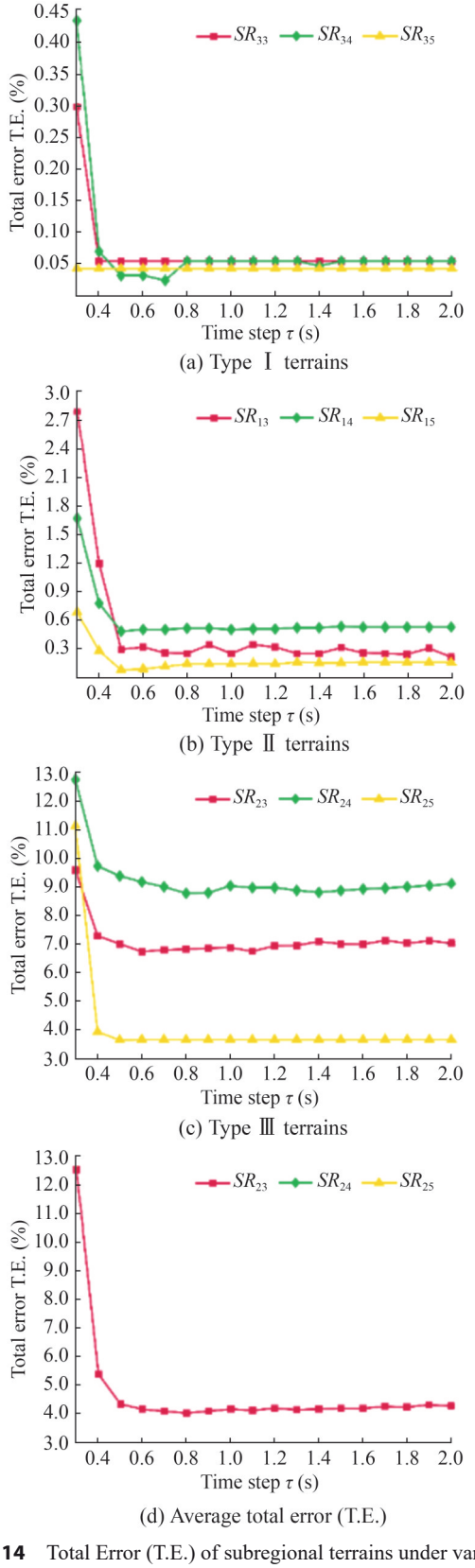


Figure 14 Total Error (T.E.) of subregional terrains under various τ

of iterations increases, L_{max} and L_{avg}, respectively, are exhibited in Figure 15.

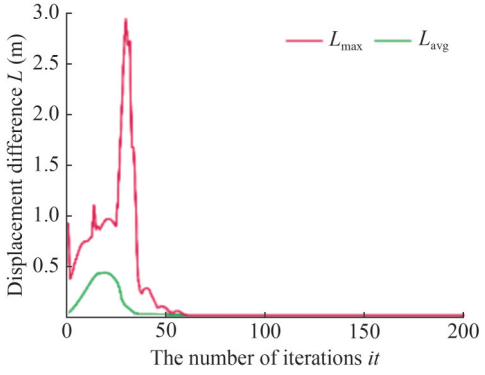


Figure 15 Maximum and average displacement differences

As exhibited in Figure 15, the L_{max} obtains the nonzero minimum value when the number of iterations is 60. After 60 iterations, L_{max} and L_{avg} converge to 0. An L_{max} reaching 0 indicates that artificial viscosity has failed, and all unfixed particles move at a constant velocity. Thus, the iteration must be terminated before L_{max} equals 0. After calculation, the nonzero minimum value of L_{max} is 0.003 5. This value is close to 0.01 times the average height difference threshold. Thus, we set a parameter δ_l equal to 0.01 times the average height difference threshold to prevent L_{max} from becoming 0.

4.4 Limitations

The above results indicate that the two primary factors influencing the filtering accuracy of SPSF are anomalous bathymetric points and overlapping bathymetric points. When anomalous or overlapping points form surfaces, the particle grid of SPSF cannot accurately reflect the seafloor terrain continuity. Thus, anomalous and overlapping points must be minimized during the acquisition and postprocessing of multibeam bathymetric data. In our future work, we will improve bathymetric accuracy by optimizing bathymetric point detection and reduce the splicing errors of multibeam strips by updating auxiliary information acquisition equipment.

5 Conclusion

In this paper, we propose a new ground filtering method for multibeam bathymetric point clouds, namely SPSF, which establishes a particle grid to match the point-set surface. SPSF modifies the particle motion and makes it more appropriate for point cloud filtering. In addition, we introduce the partition optimization method to improve the filtering performance. The results reveal that SPSF can detect anomalous and overlapping points and continuously express seafloor terrain. Compared with SMF and PMF, SPSF has a smaller T.E. and a higher κ. Compared with CSF, BCSF, and ICSF, SPSF has higher accuracy and efficiency.

In the future, we will optimize the particle motion model

and parameters of SPSF to make them applicable to different types of bathymetric data and to study new filtering methods further based on statistical physical models. Moreover, we will explore the influence of the stability of time steps on filtering accuracy.

Funding This study is supported by the National Natural Science Foundation of China (Grant number: U2441254, 42176192, 42176188); the Harbin Engineering University Doctoral Research and Innovation Fund (Grant number: XK2050021034); the National Key R&D Program of China (Grant number: 2022YFB3206900); the Stable Supporting Fund of Acoustic Science and Technology (Grant number: JCKYS2022604SSJS002, JCKYS2024604SSJS007); and the Fundamental Research Funds for the Central Universities (Grant number: 3072024CFJ0504).

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

References

Axelsson P (2000) DEM Generation from Laser Scanner Data Using Adaptive TIN Models. Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. 33(4): 110-117. https://www.scopus.com/pages/publications/85044575962

Becker M, Teschner M (2007) Weakly compressible SPH for free surface flows. the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA'07). Eurographics Association Goslar 209-217. https://dl.acm.org/doi/pdf/10.5555/1272690.1272719

Calder BR, Mayer LA (2003) Automatic processing of high-rate, high-density multibeam echosounder data. Geochem Geophys Geosyst 4(6): 1-22. https://doi.org/10.1029/2002GC000486

Cai SS, Yu SS (2023) Filtering Airborne LiDAR Data in Forested Environments Based on Multi-Directional Narrow Window and Cloth Simulation. Remote Sens 15(5): 1400. https://doi.org/10.3390/rs15051400

Cai SS, Yu SS, Hui ZY, Tang ZZ (2023) ICSF: An Improved Cloth Simulation Filtering Algorithm for Airborne LiDAR Data Based on Morphological Operations Forests 14(8): 1520. https://doi.org/10.3390/f14081520

Cutter GR (2005) Seafloor habitat characterization, classification, and maps for the lower Piscataqua River estuary. PhD thesis, University of New Hampshire

Desbrun M, Gascuel MP (1996) Smoothed particles: A new paradigm for animating highly deformable bodies. Boulic, R., Hégron, G. (eds) Computer Animation and Simulation'96. Eurographics. Springer, Vienna. https://doi.org/10.1007/978-3-7091-7486-9_5

Dorai C, Jain AK (1997) COSMOS-A representation scheme for 3D free-form objects. IEEE Trans. Pattern Anal. Mach. Intell. 19(10): 1115-1130. https://doi.org/10.1109/34.625113

Gingold RA, Monaghan JJ (1977) Smoothed particle hydrodynamics: theory and application to non-spherical stars. Mon. Not. R. Astron. Soc. 181(3): 375-389. https://doi.org/10.1093/mnras/181.3.375

Goldman R (2005) Curvature formulas for implicit curves and surfaces. Comput. Aided Geom. Des. 22(7): 632-658. https://doi.org/10.1016/j.cagd.2005.06.005

Guan ML, Cheng YX, Li QQ, Wang CS, Fang X, Yu JW (2019) An Effective Method for Submarine Buried Pipeline Detection via Multi-Sensor Data Fusion. IEEE Access 7: 125300-125309. https://doi.org/10.1109/access.2019.2938264

Hollaus M, Höfle B (2010) Terrain roughness parameters from full-waveform airborne LiDAR data. ISPRS TC VII Symposium: 100 Years ISPRS, Vienna, Austria, 38(7B): 287-292

Hu H, Ding YL, Zhu Q, Wu B, Lin H, Du ZQ, Zhang YT, Zhang YS (2014) An adaptive surface filter for airborne laser scanning point clouds by means of regularization and bending energy. ISPRS J. Photogramm. Remote Sens. 92: 98-111. https://doi.org/10.1016/j.isprsjprs.2014.02.014

Le Touzé D, Colagrossi A (2025) Smoothed particle hydrodynamics for free-surface and multiphase flows: a review. Rep Prog Phys. 88(3). https://doi.org/10.1016/10.1088/1361-6633/ada80f

Liang GL, Zhang BY, QI B, Hao Y, Du ZY, LI X (2024) Underwater multitarget fusion tracking method for passive sonar. Acta Acust. 49(3): 501-512. https://doi.org/10.12395/0371-0025.2022188.

Liu X (2008) Airborne LiDAR for DEM generation: Some critical issues. Prog. Phys. Geogr. Earth Environ. 32(1): 31-49. https://doi.org/10.1177/0309133308089496

Lyu XY, Wang L, Huang DX, Wang SL (2023) A Novel Cone Model Filtering Method for Outlier Rejection of Multibeam Bathymetric Point Cloud: Principles and Applications. Sensors 23(17): 7483. https://doi.org/10.3390/s23177483

Mann M, Agathoklis P, Antoniou A (2001) Automatic outlier detection in multibeam data using median filtering. 2001 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, Victoria, BC, Canada, August 2001. https://doi.org/10.1109/PACRIM.2001.953726

Minár J, Evans IS, Jenčo M (2020) A comprehensive system of definitions of land surface (topographic) curvatures, with implications for their application in geoscience modeling and prediction. Earth-Sci Rev 211(11). https://doi.org/10.1016/j.earscirev.2020.103414

Mitchell NC (1996) Processing and Analysis of Simrad Multibeam Sonar Data. Mar. Geophys Res 18: 729-739. https://doi.org/10.1007/BF00313883

Monaghan JJ (2005) Smoothed Particle Hydrodynamics. Rep Prog. Phys 68(8): 1703. https://doi.org/10.1088/0034-4885/68/8/R01

Müller M, Charypar D, Gross M (2003) Particle-Based Fluid Simulation for Interactive Applications. the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA'03). Eurographics Association Goslar DEU 154-159. https://dl.acm.org/doi/pdf/10.5555/846276.846298

Palomer A, Rida P, Ribas D, Vallicrosa G (2015) Multi-beam Terrain/Object Classification for Underwater Navigation Correction. OCEANS 2015-Genova Genova Italy 18-21 May. https://doi.org/10.1109/OCEANS-Genova.2015.7271587

Pingel TJ, Clarke KC, McBride WA (2013) "An improved simple morphological filter for the terrain classification of airborne LIDAR data. ISPRS J. Photogramm Remote Sens 77: 21-30. https://doi.org/10.1016/j.isprsjprs.2012.12.002

Pratt JC (2007) Cape Henlopen Spit Complex and Recent Evolution of Breakwater Harbor. University of Delaware. Lewes, Delaware

Rezvani MH, Sabbagh A, Ardalan AA (2015) Robust Automatic Reduction of Multibeam Bathymetric Data Based on M-Estimators. Mar Geod 38: 327-344. https://doi.org/10.1080/01490419.2015.1053639

Rodríguez-Caballero E, Afana A, Chamizo S, Sol´e-Benet A, Canton Y (2016) A new adaptive method to filter terrestrial laser scanner point clouds using morphological filters and spectral information to conserve surface micro-topography. ISPRS J. Photogramm

Remote Sens 117: 141-148. https://doi.org/10.1016/j.isprsjprs.2016.04.004

Schechter H, Bridson R (2012) Ghost SPH for animating water. ACM Trans Graph 31(4): 1-8. https://doi.org/10.1145/2185520.2185557

Shan J, Aparajithan S (2005) Urban DEM generation from raw lidar data. Photogramm. Eng. Remote Sens. 71(2): 217-226. https://doi.org/10.14358/PERS.71.2.217

Susaki J (2012) Adaptive slope filtering of airborne LiDAR data in urban areas for digital terrain model (DTM) generation. Remote Sens 4(6): 1804-1819. https://doi.org/10.3390/rs4061804

Tan YM, Wang S, Xu B, Zhang JB (2018) An improved progressive morphological filter for UAV-based photogrammetric point clouds in river bank monitoring. ISPRS J. Photogramm Remote Sens 146: 421-429. https://doi.org/10.1016/j.isprsjprs.2018.10.013

Vicki LF (2004) Dynamics of nearshore sedimentary environments revealed through the analysis of multibeam sonar data. State University of New York at Stony Brook

Vosselmann G (2000) Slope based filtering of laser altimetry data. Int. Arch. Photogramm. Remote Sens Spatial Inf Sci 33(Part B3/2): 935-942. https://doi.org/10.1016/S0924-2716(98)00009-4

Wang CK, Tseng YH (2010) Dem generation from airborne LiDAR data by an adaptive dual-directional slope filter. Int. Arch. Photogramm. Remote Sens. Spatial Inf Sci Vienna Austria 38: 628-632. https://api.semanticscholar.org/CorpusID:64620952

Wu Z, Li J, Li S, Shang J, Jin X (2017) A new method to identify the foot of continental slope based on an integrated profile analysis. Mar Geophys Res 38: 199-207. https://doi.org/10.1007/s11001-016-9273-4

Xu C, Wu MX, Zhou T Li JH, Du W.D, Zhang WY, White PR (2020) Optical flow-based detection of gas leaks from pipelines using multibeam water column images. Remote Sens 12(1): 119. https://doi.org/10.3390/rs12010119

Yan ZY, Zhou T, Guo QJ, Xu C, Wang TH, Peng DD, Yu XY (2022) Terrain matching positioning method for underwater vehicles based on curvature discrimination. Ocean Eng 260: 111965. https://doi.org/10.1016/j.oceaneng.2022.111965

Yang AX, Wu ZY, Yang FL, Su DP, Ma Y, Zhao DN, Qi C (2020) Filtering of airborne LiDAR bathymetry based on bidirectional cloth simulation. ISPRS J Photogramm Remote Sens 163: 49-61. https://doi.org/10.1016/j.isprsjprs.2020.03.004

Yang P, Qian X (2007) Direct computing of surface curvatures for point-set surfaces. Eurographics Symp. Point-Based Graph 1: 29-36. https://doi.org/10.2312/SPBG/SPBG07/029-036

Zhang K, Chen SC, Whitman D, Shyu ML, Yan J, Zhang C (2003) A progressive morphological filter for removing nonground measurements from airborne LIDAR data. IEEE Trans Geosci. Remote Sens 41(4): 872-882. https://doi.org/10.1109/TGRS.2003.810682

Zhang WM, Qi JB, Wan P, Wang HT, Xie DH, Wang XY, Yan GJ, (2016) An easy-to-use airborne LiDAR data filtering method based on cloth simulation. Remote Sens 8(6): 501. https://doi.org/10.3390/rs8060501

Zhang WY, Zhou T, Li JH, Xu C (2022) An efficient method for detection and quantitation of underwater gas leakage based on a 300-khz multibeam sonar. Remote Sens 14(17): 4301. https://doi.org/10.3390/rs14174301

Zhou T, Si JK, Wang LY, Xu C, Yu XY (2022) Automatic detection of underwater small targets using forward-looking sonar images. IEEE Trans Geosci Remote Sens 60: 1-12. https://10.1109/TGRS.2022.3181417