Multiscale and Multisubgraph-Based Segmentation Method for Ocean Remote Sensing Images

Qianna Cui, Haiwei Pan, *Member, IEEE*, Kejia Zhang, Xiaokun Li, and Haoyu Sun

***Abstract*— Interpreting ocean remote sensing images is still a challenge that is worth studying because they can carry valuable information for various important applications. Due to the absence of labeled datasets, unsupervised object-based image analysis (OBIA) methods provide an effective solution to understand remote sensing images with the advantage of grouping local similar pixels into a homogeneous area. However, ocean remote sensing images usually have the characteristics of large size, large background, and coexisting of large and small objects, which results in previous OBIA methods easily falling into the difficulty of accurately segmenting the large and small objects at the same time and the dilemma of time-consuming computation. To solve this problem, a novel multiscale and multisubgraph (MSMSG)-based image segmentation method is presented in this article. First, a coarse-to-fine superpixel generation method is designed to generate optimal superpixels, which can not only solve the problem of coexisting large objects and small objects but also the problem of manually setting the initial segmentation number. Second, the proposed background removal strategy helps to eliminate the trouble of large background areas in ocean remote sensing images. Third, a multisubgraph is constructed with the help of background removal. Finally, the MSMSG merging strategy is addressed to group all similar superpixels into the same cluster, which not only reduces the useless computation of nonadjacent superpixels but also avoids segmentation errors with the same scale. Experiments conducted on three different datasets show that the proposed segmentation method is high-performance and high-efficiency.**

***Index Terms*— Background removal, multiscale, multisubgraph (MSG), ocean remote sensing image, superpixel.**

I. Introduction

RECENTLY, ocean remote sensing images are becoming more available with the growing techniques of various satellites, and segmenting these images has been widely used in sea monitoring missions, such as ship detection [1], [2], coastline detection [3], and oil spill monitoring [4]. Ocean remote sensing images, large size and complex backgrounds, are high spatial remote sensing images captured with the sustainable improvement of Earth observation satellite technology. Due to the two image characteristics, traditional pixel-based segmentation methods may fall into time-consuming and inaccurate dilemmas [5]. Object-based image analysis (OBIA) methods, instead of pixel-based image segmentation methods, not only accelerate large-size image segmentation [6], [7] but also suppress the sensitivity of speckle noise on segmentation [7]. The OBIA method is commonly composed of two steps: 1) objects generated with image presegmentation methods and 2) object-based feature extraction and classification [8], [9], [10].

Deep learning has made brilliant achievements in remote sensing images, which mainly plays a key role in feature extraction and classification for the OBIA method. By using the open water features extracted with a convolutional neural network (CNN), the OBIA method is doing well on the optical satellite imagery [11]. Benefiting from the attention mechanism and the fusion representation of the first- and second-order features, an improved CNN model can help to improve the classification accuracy of land use and land cover classification [12]. The difficulty of segmenting the small objects has been solved [13] by combining the semantic features generated with the DeepLabv3+ network [14] and region information generated with OBIA. The CNN can be employed to learn the perception ability of visual elements in each object, and the multilayer-integration graph attention network model is to mine the spatiotopological relationships of the scene [15]. Thus, the superior performance of the multilabel remote sensing image scene classification is obtained [15]. An improved bilinear CNN is proposed to obtain bilinear ensemble features with lower dimensionality and, thus, to detect the polygon change on remote sensing images automatically [16]. To overcome the detrimental influence of heterogeneous content generated with the common deep neural networks, a simplified object-based deep neural network is addressed to improve the remote sensing image classification [17].

However, a large number of tagged datasets are generally required to adequately train the deep learning model, which is difficult to meet in the scarce datasets. It is a hugely high cost to catch and tag sea surface remote sensing datasets, so unsupervised OBIA methods are worth studying. In the unsupervised OBIA method, spectral, intensity, and texture features are selectively extracted from all objects, by which the similarity (or dissimilarity) can be built between adjacent

Manuscript received 15 October 2022; revised 29 January 2023; accepted 17 February 2023. Date of publication 22 February 2023; date of current version 7 March 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62072135, in part by the Innovative Research Foundation of Ship General Performance under Grant 26622211, and in part by the Ningxia Natural Science Foundation Project under Grant 2022AAC03346. *(Corresponding author: Haiwei Pan.)*

Qianna Cui, Haiwei Pan, Kejia Zhang, and Haoyu Sun are with the School of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China (e-mail: cuiqianna@hrbeu.edu.cn; panhaiwei@hrbeu.edu.cn; kejiazhang@hrbeu.edu.cn; sunhy@hrbeu.edu.cn).

Xiaokun Li is with the School of Computer Science and Technology, Heilongjiang University, Harbin 150080, China (e-mail: li.xiaokun@163.com).

Digital Object Identifier 10.1109/TGRS.2023.3247697

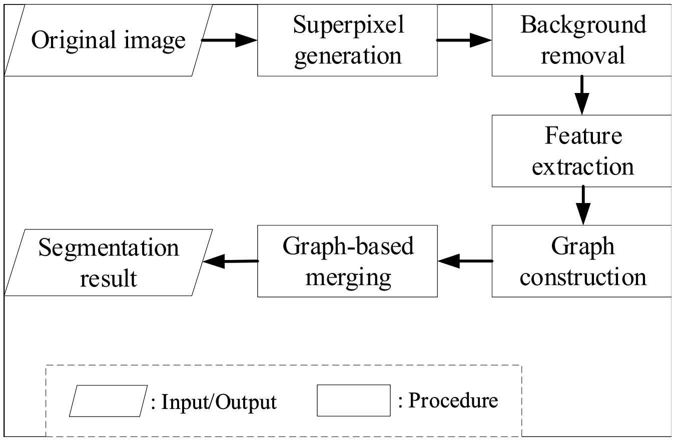


Fig. 1. Flow diagram of the modified unsupervised OBIA method.

objects [5], [18], [19]. With the given merging condition, similar adjacent objects can be merged into a new object [20]. Good objects, high homogeneous superpixels actually, are grouped with adjacent similar pixels [21]. Thus, in the first procedure of unsupervised OBIA, a prominent superpixel generation method is a must to produce good objects. The second step is to merge all adjacent similar superpixels into a larger group with the abovementioned traditional features. Due to region adjacent graph (RAG) representing adjacent structures with high efficiency mathematically, graph-based merging is a research popular in unsupervised OBIA [5], [22], [23]. A large-scale area in an ocean remote sensing image is obviously to be computationally intractable for the RAG-based merging process, which means that background removal is imperative in the unsupervised OBIA. In this way, the flow diagram of the modified unsupervised OBIA method is shown in Fig. 1.

A. Superpixel Generation

Superpixel generation methods mainly include graph-based strategies [24], [25], [26] and clustering techniques [27], [28]. Graph-based strategies consider each pixel in an image as a vertex and the relationship between two adjacent vertexes as an edge, where the weight of each edge is calculated with the dissimilarity of a pair of neighbor vertexes [29]. Superpixels are generated by finding a minimal spanning tree proposed by Felzenszwalb and Huttenlocher (FH) [29]. The normalized cut (NC) method produces superpixels by considering the dissimilarity between different parties and the similarity of each intragroup [30]. The entropy rate superpixel (ERS) method constructs high homogeneity groups with maximum entropy functions [31]. Superpixels, retaining well marginal information, are generated by these graph-based methods. However, they are all short of controlling compactness factors and determining graph parameters with high computational time [29]. Clustering techniques are widely applied to image segmentation due to the advantage of low computational complexity, by which high-compact superpixels outperform superpixels by the other superpixel generation methods [32], [33]. Simple linear iterative clustering (SLIC) [21], a k-means clustering-based algorithm, is widely used in superpixel generation tasks [34], [35]. Subsequently, certain clustering methods are designed to improve the performance of SLIC. The linear spectral clustering (LSC) [36] method in remote sensing image segmentation [37] uses a kernel function as the similarity metric instead of the traditional Euclidean distance in SLIC, which has stronger boundary adherence performance. Simple noniterative clustering (SNIC) [38] requires less memory due to its noniterative method but with generating irregular superpixels. Fuzzy SLIC (FSLIC) adopts the local spatial fuzzy C-means clustering technique to assign multiple cluster centers to each pixel with the help of multiple fuzzy partition degrees, which is the key to selecting the more accurate clustering center for pixels [39]. The competitive mechanism-based image segmentation (CMSuG) method [18] leverages a competitive mechanism to weaken the dependence of initial clusters in FSLIC. In a fuzzy manner, the FSLIC and CMSuG superpixel generation methods generate more accurate superpixels by improving the classification of indistinct pixels but with much computation time.

Although these superpixel generation methods have so far been widely applied to remote sensing image segmentation [5], [19], [40], [41], there are still some limitations, including: 1) the difficulty in capturing small objects and large objects in large-size ocean remote sensing images at the same time and 2) the huge time-consuming and nonautomatic of the optimal superpixel generation scale to be found. To overcome these drawbacks, we put forward a coarse-to-fine superpixel generation method by the joint of SLIC [21] and the classical Otsu algorithm [42], in which Otsu is to resegment the worst superpixels generated by SLIC. In addition, an automatic optimal scale selection strategy is designed with the help of the scale evaluation criterion [43], [44] in multiscale superpixels.

B. Background Removal

Due to the fact of the large sea background in the ocean remote sensing image, the distant superpixels are separated. It is impossible to group these far superpixels into the same cluster, and the redundancy of background superpixels results in large invalid and time-consuming calculations in the superpixel-based merging. Therefore, before merging the superpixels, background removal is a must in ocean remote sensing images to tackle this problem. To split the foreground from the background for an image, Grahcuts do the task by interactively specifying the part of the foreground and background areas [45]. Grabcut labels a few bounding boxes of the background to model the whole image splitting iteratively [46]. The two methods are based on manually labeling background pixels or areas, the labor cost of which is too high in the large remote sensing images. Alpha matting extracts all foreground pixels by the estimation of color and opacity (α value) [47], in which two problems need to be solved. One is to solve the image pixel color distribution model, and the other is to solve the relationship between pixel color and the α value. Given the local pixel color distribution model and pixel color α value model in advance, Bayesian matting [48] fit the model through sampling and close-form matting [49] fit it by propagation to solve the α value. Regarding the matting problem as a semi-supervised learning process, Levin et al. [49] used the learning method to fit the model and solve the α value [50]. The absence of automation is a difficult problem for these alpha matting methods.

As mentioned before, it is difficult to remove the background pixels automatically due to the similar color feature of which foreground pixels for large-size ocean remote sensing images by using the above matting algorithms. The Gaussian mixture model (GMM) performs well in classifying pixels into foreground and background [51], [52]. However, such a fashion of taking the color features of each pixel into classification could lead to two kinds of incorrect classification. One is that the speckle noise produced by the sea clutter response is likely to partition into the foreground. The other is that some foreground pixels are apt to belong to the background by the weak signal response of small objects. For the wrong classification of the GMM, we design a two-step background filtering method with mathematical morphology (MM) operators. The first step is the opening operator, and the second is the closing operator [53], [54]. The important parameter of MM is a preset kernel size rather than being automatic. We, thus, propose a multiscale MM technique to improve the problem inspired by the extended morphological profile [26], [55]. After that, we use the principal component analysis (PCA) [56] to obtain the final background removal image by reducing the redundancy feature image from the stack of all MM images in the two steps.

C. Feature Extraction

The use of spectral features helps to construct the similarity between two adjacent superpixels and, thus, to complete the remote sensing image segmentation [5], [18]. Other color features, transformed from RGB colorspace, are also designed to improve the segmentation accuracy of remote sensing images, such as YUV [19] and CIELab features [10]. The intensity feature, computed with the median intensity value of a superpixel, can be added to the feature vector to take part in the similarity computation [19], [22]. Homogeneous superpixels usually have the same intensity changes in the remote sensing images. Texture features are computed to quantitively analyze these changes between two adjacent superpixels [10]. Local binary pattern (LBP) has attracted increasing attention due to its low computational cost [10], [19], [40]. Nevertheless, LBP is strongly sensitive to image rotation and noise and blurring [57]. Texture feature refers to the local contrast and standard deviation of brightness in an image, which is measured using the gray level co-occurrence matrix (GLCM) method efficiently [58], [59]. Fourteen texture features need to be extracted for the GLCM feature, which is time-consuming. The distance parameter of GLCM is another difficult task to select.

The hybrid features by combining color feature, intensity feature, and LBP feature have been experienced enabling high-efficient techniques for road extraction in remote sensing images [19]. We use the hybrid features to carry out the ocean remote sensing image segmentation. Instead of LBP, the rotation invariant LBP (RLBP) is introduced to improve the robustness of the texture feature.

D. Graph Construction and Graph-Based Merging

In the process of merging adjacent similar superpixels by the RAG-based merging method, each vertex performs a superpixel, and each edge connects a pair of adjacent vertexes. In addition, the weight of each edge is computed with the dissimilarity of each pair of vertexes, which is the whole process to create the weighted RAG (WRAG) [22], [60]. RAG is mathematically denoted with $G = (V, E)$ to represent adjacent relationships between superpixels of an image. Each vertex $v_i \in V$ represents the ith superpixel, and each edge $e(v_i, v_j) \in E$ connects a pair of adjacent vertices v_i, v_j. The WRAG $\widetilde{G} = (V, E, W)$ is built by weighting edge $e(v_i, v_j)$ with weight $w(v_i, v_j)$, where $w(v_i, v_j)$ is to compute the distance of the two superpixels corresponding to the two vertices. However, it is unsuitable to merge all similar superpixels with the abovementioned graph construction method in a large-size ocean remote sensing image due to its large background area. To tackle this problem, we address a multisubgraph (MSG) construction and the MSG merging method guided by background removal. This means that each vertex can only be limited to merge with certain vertices that belong to the same subgraph.

With the help of the WRAG, some adjacent superpixels with the most similar features are grouped into the same cluster [19], [23]. WRAG-based superpixel merging can be completed to obtain the final segmentation result with the help of a must-given merging threshold [19], [23], [24]. To alleviate the nonautomation of the merging process, the optimal merging parameter can be selected with multiscale merging thresholds [5], [20]. By comparing the local variance (LV) from the same image at various segmentation levels, the optimal scale is selected [61]. Based on the LV, the ROC-LV calculated with two adjacent scale levels is used to evaluate the multiscale merging results [5]. Considering both the characteristics of intrasegment homogeneity and intersegment heterogeneity, a newly energy function is developed to select the optimal scale [35].

However, doing merging operations with just one preset merging threshold is inapplicable in multiple subgraph-based merging. That is to say, how to apply the previous scale selection methods to MSG merging is the key to successful ocean remote sensing image segmentation. For that, we address each subgraph merging with multiple thresholds and then select the optimal merging result by the variance.

The main contributions of this work are summarized as follows.

1) We propose a multiscale and MSG (MSMSG)-based image segmentation model for ocean remote sensing images that have the characteristics of being label-free, large size, large background, and coexisting of large and small objects.
2) To address the difficulty of traditional superpixel generation methods in large-size ocean remote sensing images with coexisting large and small objects, we explore a

coarse-to-fine superpixel generation method guided by the SLIC and Otsu algorithms.

3) We design a background removal method to overcome the large background areas in ocean remote sensing images, which contains GMM and a two-step MM.
4) Guided with the background removal, we construct MSG structures and then address an MSMSG merging algorithm, which can not only split unrelated objects successfully but also greatly reduce the computation.

The rest of the article is organized as follows. In Section II, we review and introduce the related works, including SLIC [21], the global score (GS) computation strategy [43], GMM [62], MM filtering operations [53], [54], PCA [56], and the superpixel-based single graph merging (SSGM) method [18]. Section III proposes our new MSMSG-based segmentation method in detail and theoretically analyzes the time complexity of our method. In Section IV, some experiments present the high performance of our segmentation method. Section V further analyzes the influence of certain steps in the whole method. A conclusion of the work and future research is provided in Section VI.

II. Related Work

In this section, we briefly introduce the classic methods related to our MSMSG-based segmentation method, which includes SLIC, GS computation strategy, GMM, MM, PCA, and SSGM methods.

A. SLIC

SLIC [21], the k-means clustering method indeed, is adopted to produce regular and compact superpixels. In the SLIC, k cluster centers are initialized by sampling pixels at regular grid steps $\xi = (N/k)^{1/2}$, where N is the pixel number of the input ocean remote sensing image. Then, each pixel is assigned to the nearest cluster center with a $2\xi \times 2\xi$ searching scope. Each pixel, in the clustering process, is represented as a 5-D vector $\boldsymbol{p}_i = [l_i, a_i, b_i, x_i, y_i]^T$. The distance measure between each cluster $\boldsymbol{c}_j$ center and the pixels around it is denoted as follows:

$$\begin{cases} d_c = \sqrt{\left(l_i - l_j\right)^2 + \left(a_i - a_j\right)^2 + \left(b_i - b_j\right)^2} \\ d_s = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \\ d\left(\boldsymbol{p}_i, \boldsymbol{c}_j\right) = \sqrt{\left(\dfrac{d_c}{\xi'}\right)^2 + \left(\dfrac{d_s}{\xi}\right)^2} \end{cases} \tag{1}$$

where ξ' is the normalized factor to balance the variation significantly from cluster to cluster and image to image.

B. GS Computation Strategy

GS, proposed by Johnson and Xie [43], contains area-weighted variance (wVar) and global Moran's I (MI) [44], which performs well in the task of global optimal scale parameter selection [44]. The two components are calculated as follows:

$$\begin{cases} \text{wVar} = \dfrac{\sum_{i=1}^{n} a_i \times \delta_i}{\sum_{i=1}^{n} a_i} \\ \text{MI} = \dfrac{\sum_{i=1}^{n}\sum_{j=1}^{n} \delta_{ij} \times \left(y_i - \bar{y}\right)\left(y_j - \bar{y}\right)}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2 \left(\sum_i \sum_j \delta_{ij}\right)} \end{cases} \tag{2}$$

where n is the total number of superpixels, and a_i and δ_i are the area and the variance of superpixel i, respectively. y_i and y_j are the mean spectral values of superpixels i and j, respectively. The value of $\bar{y}$ is the mean spectral value of the whole ocean remote sensing image. Weight $\omega_{ij} = 1$ represents the adjacent segments i and j, whereas $\omega_{ij} = 0$ represents the opposite. Then, GS is calculated by

$$\text{GS} = \|\text{wVar}\| + \|\text{MI}\| \tag{3}$$

where $\|\text{wVar}\|$ and $\|\text{MI}\|$ are the normalized wVar and the normalized MI, respectively. Two normalized scores are computed with $(X - X_{\min})/(X_{\max} - X_{\min})$, in which $X_{\min}$ and $X_{\max}$ are the minimum and maximum values of wVar or MI. The variance δ_i of segment i is denoted as follows:

$$\delta_i = \frac{\sum_{j=0}^{n_i}\left(\boldsymbol{x}_j - \bar{\boldsymbol{x}}\right)}{3n_i} \tag{4}$$

where n_i is the sum of pixels in superpixel i. $\boldsymbol{x}_j = (r_j, g_i, b_j)$ is the jth pixel in superpixel i, which is a color vector. $\bar{\boldsymbol{x}} = \sum_{j=0}^{n_i} \boldsymbol{x}_j / n_i$ represents the mean value of pixels in the superpixel j.

C. GMM

The GMM [62] is a probability distribution model with the definition as follows:

$$\mathscr{G}(\boldsymbol{x}|\theta) = \sum_{k=1}^{K} \alpha_k g\left(\boldsymbol{x}|\mu_k, \sigma_k^2\right) \tag{5}$$

where α_k is the positive weight that pixel $\boldsymbol{x}$ belongs to foreground or background and satisfies $(\sum_{k=1}^{K} \alpha_k) = 1$, and K is the total number of categories set manually. The Gaussian density function $g(\boldsymbol{x}|\theta_k)$ is closely related to the mean μ_k and covariance σ_k of the kth Gaussian component, which is

$$g\left(\boldsymbol{x}|\mu_k, \sigma_k^2\right) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{(\boldsymbol{x} - \mu_k)^2}{2\sigma_k^2}\right) \tag{6}$$

where $\boldsymbol{x}$ is a color vector representing each pixel of the ocean remote sensing image.

D. MM Filtering Operations

For an image $\mathbf{I}$, opening and closing operations [53], [54] are defined as follows:

$$\phi(\mathbf{I}) = \mathscr{D}_\epsilon(\mathscr{E}_\epsilon(\mathbf{I})) \tag{7}$$

$$\psi(\mathbf{I}) = \mathscr{E}_\epsilon(\mathscr{D}_\epsilon(\mathbf{I})). \tag{8}$$

Given a structuring element ϵ to cover the pixel areas to be filtered, the erosion operation $\mathscr{E}(\cdot)$ is to replace these pixels with the minimum value of the pixels inside ϵ. Likewise, the dilation operation $\mathscr{D}(\cdot)$ is to replace these pixels with the maximum value of the pixels inside ϵ. Generally, ϵ is a square structure with a manual kernel size.

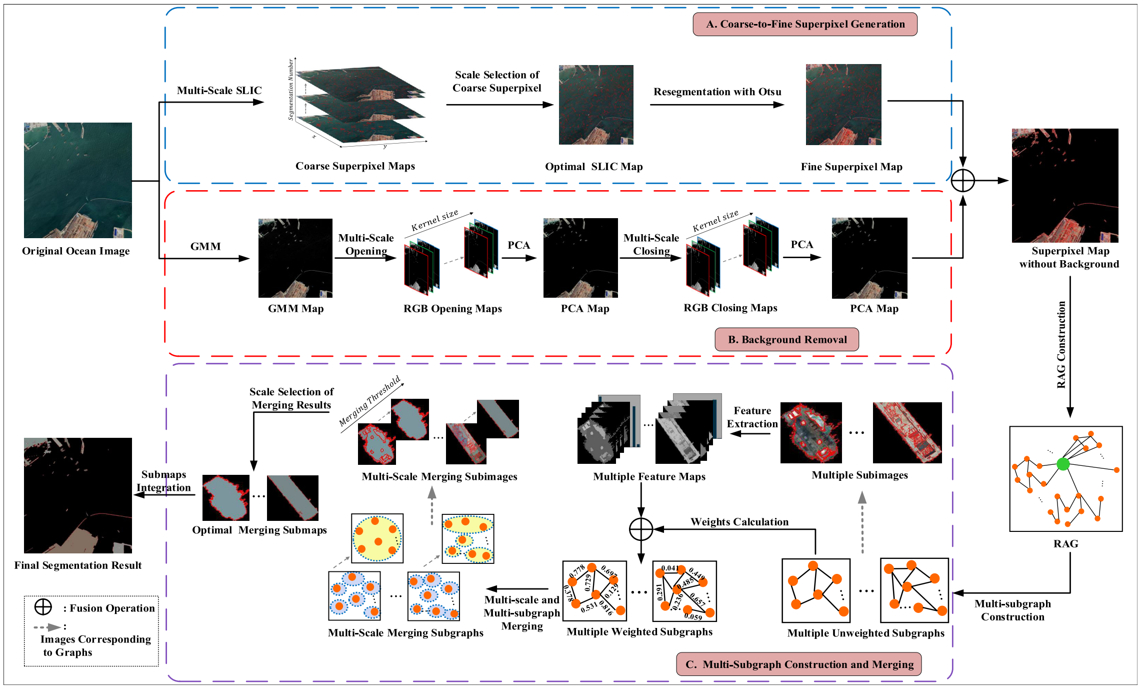


Fig. 2. Framework of the proposed image segmentation method.

E. PCA

The purpose of PCA [56] is to find a transformation matrix to transfer the high-dimensional data $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{N\times D}$ to the low d-dimensional space $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N] \in \mathbb{R}^{N\times d}$. The main process of PCA is to compute the projection matrix $\mathbf{W}^*$ as follows:

$$\mathbf{W}^* = \mathrm{argmax}_{\mathbf{W}} \mathrm{Tr}\left(\mathbf{W}^T\mathbf{X}\mathbf{X}^T\mathbf{W}\right) \tag{9}$$

where $\mathbf{W}^* = \{\mathbf{w}_1, \ldots, \mathbf{w}_d\}$ is the d-dimensional eigenvectors of covariance matrix $\mathbf{X}\mathbf{X}^T$.

F. SSGM

The WRAG $\widetilde{G} = (V, E, W)$ is constructed by all superpixels of the ocean remote sensing image, in which each vertex $v_i \in E$ represents the ith superpixel, edge $e_{ij} \in E$ connects adjacent superpixel i and superpixel j, and weight $w_{ij} \in W$ is calculated with the Euclidean distance of the two adjacent superpixels. SSGM is to merge adjacent similar superpixels into the same cluster with the built $\widetilde{G} = (V, E, W)$ [18].

The merging is first performed with an initial region set $(\mathcal{C}_1, \ldots, \mathcal{C}_n)$, where $\mathcal{C}_i = \{v_i\}$ is the ith vertex in the vertex set $V = \{v_1, \ldots, v_n\}$ and n is the number of vertices in V. Then, all edges in E are visited with the nondecreasing order of weight in W. If weight $w(v_i, v_j)$ connecting vertex v_i and vertex v_j satisfies the merging condition in formula (10), then the two components $\mathcal{C}_i$ and $\mathcal{C}_j$ are merged into the same component, and the region set is updated. In addition, the edge set is also updated by removing the edge $e(v_i, v_j)$. The above merging operation is repeated until the set E is an empty set. The merging condition is defined as follows:

$$f\left(\mathcal{C}_i, \mathcal{C}_j\right) = \begin{cases} 1, & \mathrm{MInt}\left(\mathcal{C}_i, \mathcal{C}_j\right) \le \mathrm{Dif}\left(\mathcal{C}_i, \mathcal{C}_j\right) \\ 0, & \text{Otherwise} \end{cases} \tag{10}$$

where $\mathrm{Dif}(\mathcal{C}_i, \mathcal{C}_j) = \min_{v_i \in \mathcal{C}_i, v_j \in \mathcal{C}_j} w(v_i, v_j)$ describes the minimum interdifference between two components $\mathcal{C}_i$ and $\mathcal{C}_j$ corresponding to the edge $e(v_i, v_j)$ connecting vertex $v_i \in \mathcal{C}_i$ and $v_j \in \mathcal{C}_j$. $\mathrm{MInt}(\mathcal{C}_i, \mathcal{C}_j) = \min(\mathrm{Int}(\mathcal{C}_i) + \tau(\mathcal{C}_i))$ is the minimum intradifference in a single component, where $\mathrm{Int}(\mathcal{C}_i) = \max_{v_{i1}, v_{i2} \in \mathcal{C}_i} w(v_{i1}, v_{i2})$ and $\tau(\mathcal{C}_i) = \kappa/|\mathcal{C}_i|$ is the threshold function to avoid generating too small regions and suppress producing too large connected region. The number of vertices in component $\mathcal{C}_i$ is denoted as $|\mathcal{C}_i|$, and the constant parameter κ is a user-set value [18].

III. Proposed Method

To overcome the problem that small objects and large objects coexist, and they are difficult to segment in large-size ocean remote sensing images with large background areas and sparse information, in this article, an MSMSG-based method is proposed. Fig. 2 illustrates the model of the proposed segmentation method for ocean remote sensing images. First, a coarse-to-fine superpixel generation manner is proposed to obtain superpixels by jointly using SLIC [21] and Otsu [42] segmentation algorithms, as well as an optimal scale selection strategy [43]. Then, two-step multiscale MM filtering [53] and PCA [56] are adopted to improve the background removal with GMM [19]. Furthermore, an MSG is obtained by a designed MSG construction strategy and multiple features of each superpixel. Finally, MSMSG-based merging accompanied by variance as optimal scale selection is proposed to generate the final segmentation result.

A. Coarse-to-Fine Superpixel Generation

Just with the traditional superpixel generation method, the coexisting of large and small objects cannot be segmented completely. For that, a coarse-to-fine superpixel generation method is proposed in this section. To overcome the problem of nonautomatic scale parameters in segmentation, we present a scale selection method for the coarse superpixel method, which is to select the optimal coarse superpixels with the GS strategy.

Superpixels, groups of adjacent similar pixels, are well generated with the SLIC algorithm quickly. However, affected by the preset segmentation number k, SLIC cannot obtain the optimal superpixels automatically. To generate the optimal segmentation scale k automatically, we propose an optimal segmentation selection method with multiscale superpixels. For an ocean image I with the total number of pixels N, we perform the SLIC algorithm with T_c iteration times by using different superpixel numbers $k_1, k_2, \ldots, k_{T_C}$. According to the large size character of ocean remote sensing images, let k_i increase from 0 to $(N)^{1/2}$ with a fixed step, which is the multiples of η. Then, the optimal superpixel segmentation can be selected with the scale evaluation GS, by which the optimal superpixel scale is selected by the minimum GS value.

Because small objects are difficult to detect on large-scale ocean images, undersegmentation error easily occurs only by the simple SLIC algorithm, even with the optimal superpixel number. These undersegmentation superpixels, called worst segments, are finely segmented by Otsu [42] in our coarse-to-fine superpixel generation method. The worst segments accompanied by high variances need to be subdivided. The variance of superpixel i is computed with δ_i. After sorting all variances in the nonincreasing order, we use Otsu [42] to resegment the top 5% segments. Otsu [42] is mainly up to the probability of each gray value in the range [0, 255]. For each superpixel in the range of requiring resegmented, Otsu resegments it into two segments.

The whole coarse-to-fine superpixel generation method is illustrated in Algorithm 1.

Algorithm 1 Coarse-to-Fine Superpixel Generation Algorithm

```
Input: Initial image $I$, total coarse segmentation number of times $T_C$, the factor of multiple $\eta$.
Output: Fine superpixel set $\mathcal{S}$.
Stage 1: Coarse Superpixel Generation
Transfer $I$ into CIELab space $I_{Lab}$, Compute the updated step-length of segmentation number $s = \lfloor \frac{\sqrt{N/T_C}}{\eta} \rfloor * \eta$, set $t = 1$.
while $t \leq T_C$ do
    Run SLIC algorithm in image $I_{Lab}$ to generate superpixels set $\mathcal{S}_t$ with a total segmentation number of $s$.
    $s+ = s$, $t+ = 1$.
end while
for $t = 1$ to $T_C$ do
    For segments $\mathcal{S}_t$, Compute $\mathrm{wVar}_t$ and $\mathrm{MI}_t$ values by using equation (2).
    Compute $\mathrm{GS}_t$ value by the formular (3).
end for
Output optimal coarse superpixels $\mathcal{S}_c$, which is selected with the minimum $\mathrm{GS}_c$ value from $\{\mathrm{GS}_1, \ldots, \mathrm{GS}_{T_C}\}$.
Stage 2: Fine Superpixel Generation
Calculate each superpixel's variance in $\mathcal{S}_c$ by equation (4).
Sort all variance in the non-increasing order.
Select the superpixels, which belong to the variance of the top 5%.
For each selected superpixel, take all gray value in the superpixel as threshold, run Otsu algorithm iteratively to obtain fine superpixel set $\mathcal{S}_f$.
Output the fine superpixels $\mathcal{S}$ by setting $\mathcal{S} = \mathcal{S}_f$.
```

B. Background Removal

Due to the large background area being in an ocean image, such a character makes the superpixel-based merging difficult and time-consuming. We propose a background removal method that combines GMM and two-step MM operations. Thus, guided by the background removal, the superpixels generated with the coarse-to-fine superpixel generation method can be updated.

1) Background Removal: The goal of background removal is to separate all foreground pixels from background pixels. GMM is a parametric probability density function that assumes that a mixture Gaussian distribution models all pixels in the whole remote sensing image. To classify the ocean image into two kinds, foreground and background, the total number of the mixture model is $K = 2$. The GMM is to classify all pixels, in which each pixel $\boldsymbol{x}$ is a 3-D color vector.

Some speckle pixels are wrongly classified into the foreground because certain background scenes are strongly responded to by remote sensing sensors. Coincidentally, some weak signal responses of small objects result in the object pixels being wrongly labeled as background. To improve the two serious misclassifications by GMM [62], we propose a two-step MM filtering method. The first step is the combination of multiscale opening [53], [54] and PCA [56] operations, which is designed to filter the speckle noise pixels. The second step combines the multiscale closing and PCA operations to solve the mistakes resulting from weak foreground pixels.

In the opening and closing filtering operations, it is difficult to obtain the optimal filtering images by using nonautomatic kernel ϵ. Applying MM to a multichannel input image $\mathbf{I}$ with an increasing size of ϵ, the multiscale MM maps can be obtained to extract the optimal filtering image [63]. A series of kernel sizes can be mathematically given by

$$\left\{\lambda_{\min}, \lambda_{\min} + \lambda', \lambda_{\min} + 2\lambda', \ldots, \lambda_{\max}\right\} \tag{11}$$

where $\lambda_{\min}$, $\lambda_{\max}$, and λ' are the minimal kernel size, the maximum kernel size, and the increasing size of one step, respectively. To eliminate speckle noise and keep all objects as complete as much as possible, $\lambda_{\min}$ is larger than 1, and $\lambda_{\max}$ must be smaller than the minimum of the short edge of all superpixel bounding boxes. Then, multiscale opening and closing maps are obtained as $\{(\phi(\mathbf{I}))_{\lambda_{\min}}, \ldots, (\phi(\mathbf{I}))_{\lambda_{\max}}\}$ and $\{(\psi(\mathbf{I}))_{\lambda_{\min}}, \ldots, (\psi(\mathbf{I}))_{\lambda_{\max}}\}$, respectively. We denote a high-dimensional space $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_P] \in \mathbb{R}^{N \times B \times P}$ to integrate multiscale opening or closing maps, in which N is the pixel number, B is the bands of an input image $\mathbf{I}$,

Algorithm 2 Background Removal Algorithm

```
Input: Initial image $I$, fine superpixel $\mathcal{S}$, minimal kernel size $\lambda_{\min} = 1 + \lambda'$, increasing size $\lambda'$.
Output: Background pixel set $\mathcal{B}$ and foreground pixel set $\mathcal{F}$.
Compute the maximum kernel size $\lambda_{\max}$ with all bounding box of $\mathcal{S}$.
Run GMM algorithm in $I$ to generate background separation image $\mathbf{I}_0$, set $\lambda = \lambda_{\min}$.
while $\lambda \leq \lambda_{\max}$ do
    Run opening operation (7) into $\mathbf{I}_0$ to generate filtering image $(\phi(\mathbf{I}_0))_\lambda$.
    $\lambda + = \lambda'$.
end while
Construct high-dimension space $\mathbf{X}_0 = [(\phi(\mathbf{I}_0))_{\lambda_{\min}}, \cdots, (\phi(\mathbf{I}_0))_{\lambda_{\max}}]$.
Use PCA into $\mathbf{X}_0$ to obtain the first filtered image $\mathbf{I}_1$, reset $\lambda = \lambda_{\min}$.
while $\lambda \leq \lambda_{\max}$ do
    Run closing operation (8) into $\mathbf{I}_1$ to generate filtering image $(\psi(\mathbf{I}_1))_\lambda$.
    $\lambda + = \lambda'$.
end while
Construct high-dimension space $\mathbf{X}_1 = [(\psi(\mathbf{I}_1))_{\lambda_{\min}}, \cdots, (\psi(\mathbf{I}_1))_{\lambda_{\max}}]$.
Use PCA into $\mathbf{X}_1$ to obtain the second filtered image $\mathbf{I}_2$.
for all pixel $\boldsymbol{x}_j \in \mathbf{I}_2$ do
    if the pixel values of $\boldsymbol{x}_j$ are zeros then
        Put $\boldsymbol{x}_j$ into background set $\mathcal{B}$.
    else
        Put $\boldsymbol{x}_j$ into foreground set $\mathcal{F}$.
    end if
end for
Output background set $\mathcal{B}$ and foreground set $\mathcal{F}$.
```

$P = \lfloor(\lambda_{\max} - \lambda_{\min})/\lambda'\rfloor + 1$ is the number of kernel size sets, and $\mathbf{x}_\lambda$ is the λth opening map $(\phi(\mathbf{I}))_\lambda$ or closing map $(\psi(\mathbf{I}))_\lambda$. Thus, extracting the optimal filtered image problem is skillfully transferred into an accessing principle component problem from high-dimensional space $\mathbf{X}$. By applying the PCA method [56], we acquire the principle component $\mathbf{X}' \in \mathbb{R}^{N \times B \times 1}$. With the help of $\mathbf{X}'$, the filtered opening and closing images are easy to be obtained. It results in classifying foreground and background well by using all pixel $\boldsymbol{x}_j$ in the filtered image, where $\boldsymbol{x}_j$ is the 3-D pixel values of the jth pixel. The whole background removal process is shown in Algorithm 2.

2) Superpixel Updating: After removing the background, the superpixel set $\mathcal{S}$ obtained in Algorithm 1 can be redefined. For each superpixel $S_i \in \mathcal{S}$, three cases happen possibly.

1) *Case 1:* All pixels in S_i belong to the background set $\mathcal{B}$.
2) *Case 2:* All pixels in S_i belong to the foreground set $\mathcal{F}$.
3) *Case 3:* In the superpixel S_i, some pixels belong to the background set $\mathcal{B}$, and some others belong to the foreground set $\mathcal{F}$.

We remove superpixel S_i from $\mathcal{S}$ when superpixel S_i satisfies case 1. When case 2 happens, keep the superpixel S_i for the superpixel set $\mathcal{S}$. We segment superpixel S_i into two segment sets $\{S_i^1, \ldots, S_i^{n_i}\}$ and S_i^0 if case 3 occurs. Segment S_i^0 is the group of background pixels in S_i, and segment sets $\{S_i^1, \ldots, S_i^{n_i}\}$ are the foreground group, which is because the background S_i^0 is likely to split the foreground of superpixel S_i into n_i segments. With the help of removing background set S_i^0, we can obtain the segment sets $\{S_i^1, \ldots, S_i^{n_i}\}$ by finding the location of circles generated with the boundaries of S_i. The two segment sets satisfy $S_i^0 \cup S_i^1 \cup \ldots \cup S_i^{n_i} = S_i$, and $S_i^i \cap S_i^j = \varnothing$, where $i, j \in [0, 1, \ldots, n_i]$. It is noticed that the common situation is $n_i = 1$. For case 3, we replace S_i stored in $\mathcal{S}$ with the foreground sets $\{S_i^1, \ldots, S_i^{n_i}\}$.

Based on the operations, the new superpixel set is updated with the combination of the current foreground superpixel sets $\mathcal{S} = \{\ldots, S_i^1, \ldots, S_i^{n_i}, \ldots\}$ and the background set $\mathcal{B}$. For convenience, we reset the foreground superpixel set as $\mathcal{R} = \{R_1, \ldots, R_m\}$, where $|\mathcal{S}| = |\mathcal{R}|$, and $|\cdot|$ is the number of set $\mathcal{S}$ or $\mathcal{R}$. Such notes construct the superpixel set $\{\mathcal{R}, \mathcal{B}\}$ of the whole ocean remote sensing image.

C. Multisubgraph Construction

Some adjacent superpixels, generated with the above segmentation process, can be grouped into the same cluster by similar features. RAG is an important key to merging these adjacent superpixels into the same group [18], which can also divide heterogeneous regions well. Each vertex represents a superpixel, and each edge connects a pair of adjacent superpixels in RAG. It is obvious that background $\mathcal{B}$ in an ocean remote sensing image is a separate cluster, which means that any superpixel in $\mathcal{R}$ does not do merging with it. The other case that occurs in the ocean image is that some superpixels are impossibly merged, which are separated away by the background. For these two reasons, an amount of pointless computation happens in the merging process of the RAG G generated with all superpixels. To solve this problem, we propose an MSG merging method by generating multiple subgraphs.

For doing that, we first construct an RAG $G = (V, E)$, where V is the vertex set corresponding to the superpixel set and the edge set is E. In the graph G, vertex set V and edge set E are built with $V = \{u, v_1, v_2, \ldots, v_m\}$ and $E = E^0 \cup E^1$. The definitions of E^0, E^1 are given as follows:

$$\begin{cases} E^0 = \{(u, v_i) | 1 \leq i \leq m\} \\ E^1 = \{(v_i, v_j) | 1 \leq i, j \leq m\} \end{cases} \tag{12}$$

where vertex $v_i \in V$ $(i \in [1, \ldots, m])$ corresponding to superpixel R_i and vertex u corresponding to the background superpixel $\mathcal{B}$ form the vertex set. Similarly, we denote each edge $e(v_i, v_j) \in E^1$ connecting two adjacent superpixels in $\mathcal{R}$ and an edge $e(v_i, u) \in E^0$ connecting the two vertexes located in u and v_i, where $i, j \in [1, \ldots, m]$. The MSG construction idea that we proposed is just done by removing the edge set E^0. In fact, each subgraph is a connected component [64] in graph G without edge set E^0. The definitions of the connected component and MSG are given as follows. By using the

Algorithm 3 MSG Construction Algorithm

```
Input: Superpixel set $\{R_1, \cdots, R_m, \mathcal{B}\}$ of image $I$.
Output: Multi-subgraph $\{G_1^1, \cdots, G_z^1\}$, where $G_1^1 = (V_1, E_1^1), \cdots, G_z^1 = (V_z, E_z^1)$.
Construct an RAG graph $G = (V, E)$ with input superpixel set.
Find all edge connecting vertex $u$ and other vertexes, and put all these edges into $E^0$.
Remove all edges in $E^0$ from $E$ and output graph $G^1 = (V, E^1)$, where $E^1 = E \setminus E^0$.
set $x = 1$.
while $V \neq \varnothing$ do
    Set vertex set $V_x = \varnothing$ and edge set $E_x^1 = \varnothing$, neighbor vertex set $\mathcal{N}_x = \varnothing$.
    Randomly select a vertex $v_x$ from $V$ and put it into $V_x$.
    Find all neighbors of $v_x$ in $V$ and put them into $\mathcal{N}_x$.
    Remove vertex $v_x$ from $V$.
    while $\mathcal{N}_x \neq \varnothing$ do
        Randomly select a vertex $v_{x,i}$ from $\mathcal{N}_x$.
        if $v_{x,i} \notin V_x$ then
            Find all neighbors of $v_{x,i}$ in $V$ and put them into $\mathcal{N}_x$.
            Put the vertex $v_{x,i}$ into $V_x$ and remove it from $V$.
        else
            Remove the vertex $v_{x,i}$ from $\mathcal{N}_x$.
        end if
    end while
    Put the union of all the edges that connect the vertex $v_{x,j}$ and other neighbors into $E_x^1$, where $\forall v_{x,j} \in \mathcal{V}_x$.
    Remove all edges in $E_x^1$ from $E^1$.
    set $x += 1$.
end while
Output $\{G_1^1 = (V_1, E_1^1), \cdots, G_z^1 = (V_z, E_z^1)\}$.
```

definition of the MSG, a theorem about that is given to guide the MSG creation.

Definition 1: A connected component of graph $G = (V, E^1)$ is an isolated point or a graph $G_x^1 = (V_x, E_x^1)$ such that the following holds.

1) $V_x \subseteq V$, $E_x^1 \subseteq E^1$, and there is at least one path $\langle v_{x,i}, \ldots, v_{x,j} \rangle$ between any pair of vertices $v_{x,i}$ and $v_{x,j}$, where $v_{x,i}$ and $v_{x,j} \in V_x$.
2) There is no path between vertex $v_{x,i}$ and vertex $v_{y,j}$, where $v_{y,j} \in V_y$ from the graph $G_y^1 = (V_y, E_y^1)$ that satisfies the condition (1).

Definition 2: The MSG of graph $G^1 = (V, E^1)$ is a connected component set $\{G_1^1 = (V_1, E_1^1), \ldots, G_z^1 = (V_z, E_z^1)\}$ such that the following holds.

1) $V_1 \cup \cdots \cup V_z = V$ and $E_1^1 \cup \cdots \cup E_z^1 = E^1$.
2) $V_x \cap V_y = \varnothing$ and $E_x^1 \cap E_y^1 = \varnothing$, where $x, y \in [1, \ldots, z]$.

Theorem 1: A connected component set $\{G_1^1, \ldots, G_z^1)\}$ is the MSG of graph G^1 if and only if all neighbor vertexes $v_j \in V$ of any vertex $v_{x,i} \in V_x$ and the edge $e(v_j, v_{x,i}) \in E^1$ are included in the connected component G_x^1, where $x \in [1, \ldots, z]$ and $e(v_j, v_{x,i})$ connects vertex $v_{x,i}$ and v_j.

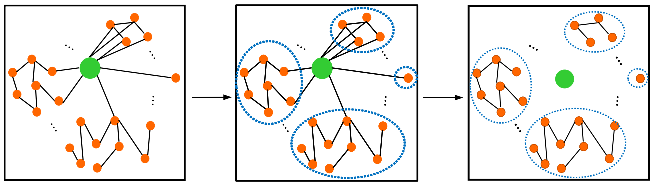


Fig. 3. Example of the MSG creation method.

The proof of Theorem 1 is given in Appendix A. Guided by Theorem 1, we design an MSG construction method, an example of which is lively shown in Fig. 3, and the process is illustrated in Algorithm 3.

D. Multisubgraph Merging

When we generate the MSG set $\{G_1^1, \ldots, G_z^1\}$ by Algorithm 3, the similar adjacent vertexes can be grouped into the same class occurring in each subgraph, where $G_1^1 = (V_1, E_1^1), \ldots, G_z^1 = (V_z, E_z^1)$. It is likely to fall into the undersegmentation of small objects and oversegmentation of large objects by using only one merging parameter in the whole graph [18]. Therefore, we propose that each subgraph does the merging operation with adaptive multiscale merging parameters, respectively. Our proposed method can not only avoid the above dilemma but also improve the problem of a nonautomatic parameter set.

1) Feature Extraction: To measure the distance of the two adjacent vertexes, we compute the features of each superpixel to construct the weight of each edge. Generally, color, intensity, and texture features are commonly used in the OBIS method [10], [19]. Color features are calculated with a 3-D CIELab vector (l_i, a_i, b_i) of each superpixel i, as in our superpixel generation method. Intensity is naturally extracted by the gray value of each superpixel. The median values of the whole superpixel are extracted in the computation of the two features, which also express context well instead of extracting from traditional single pixels. The LBP is an important cue to represent the texture of each superpixel. In this work, we use the RLBP operator for texture description superpixels [65], and the texture histogram of each superpixel is a bin size of 10. The texture feature, extracted by RLBP, is more robust than the traditional LBP in image rotation. A multifeature extraction example is plotted in Fig. 4.

2) Weights Computing: Each edge can be weighted with the dissimilarity of two adjacent superpixels, and then, the MSG set is updated as weighted graphs $\{\widetilde{G}_1, \ldots, \widetilde{G}_z\}$, where $\widetilde{G}_x = (V_x, E_x^1, W_x^1)$, W_x^1 is the weight set added in E_x^1, and $x \in [1, \ldots, z]$. In subgraph $\widetilde{G}_x$, the weight $w(v_{x,i}, v_{x,j})$ of edge $e(v_{x,i}, v_{x,j})$ is defined by the combination of the above three features, which is

$$\begin{aligned} w\big(v_{x,i}, v_{x,j}\big) = {} & \gamma_1 \|\text{color}\big(v_{x,i}\big) - \text{color}\big(v_{x,j}\big)\|_2 \\ & + \gamma_2 \|\text{intensity}\big(v_{x,i}\big) - \text{intensity}\big(v_{x,j}\big)\|_2 \\ & + \gamma_3 \|\text{texture}\big(v_{x,i}\big) - \text{texture}\big(v_{x,j}\big)\|_2 \qquad (13) \end{aligned}$$

where color$(v_{x,i})$, intensity$(v_{x,i})$, and texture$(v_{x,i})$ are the color feature, the intensity feature, and the texture histogram of the superpixel corresponding to vertex $v_{x,i}$, respectively. The

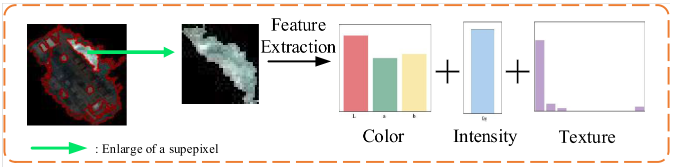


Fig. 4. Example of the feature extraction process.

dissimilarity in each feature is calculated with 2-norm $\|\cdot\|_2$. γ_1, γ_2, and γ_3 are coefficients to control the equilibrium of all distances, which satisfies $\gamma_1+\gamma_2+\gamma_3=1$ [19]. It is worth noticing that no merging process is done in each isolated point, which means that the features of these isolated points do not require computing.

3) Multisubgraph Merging: After the MSG doing weighted completely, we do the merging operation for each subgraph with the SSGM method. With a user-set parameter κ in the SSGM method, the optimal merging result is difficult to obtain by a nonmanual method. Thus, we design an optimal merging method by iteratively doing the SSGM operation with a series of $\{\kappa_1,\ldots,\kappa_{t_{\max}}\}$, where $t_{\max}$ represents the total number of the set of κ values. The same merging results might occur by using different κ values; we set the maximum number of valid results to be T_M. In addition, once all vertices merge into a cluster with a κ_i value, the merging result does not change even if κ_i increases, where $i\in[1,\ldots,t_{\max}]$. This is because the value of MInt in (10) rises with the increase in κ_i, but the Dif value is unchanged. Thus, we design the stopping condition of merging operations in each subgraph with the combination of T_M and the number of the current cluster.

After that, we can select the optimal merging result by using the variance (14), which means that the minimum difference of each good region is a must. Each segmentation result contains n superpixels, and the summary of the variance of that is computed with

$$\Delta=\sum_{i=1}^{n}\delta_i \tag{14}$$

where δ_i is the variance value of the ith superpixel. We can obtain the final segmentation as soon as we get the optimal region for each subgraph, which is done by integrating all regions into the whole image. The MSG merging method is described in Algorithm 4 in detail.

E. Complexity Analysis

In the coarse-to-fine superpixel generation method, SLIC is an algorithm with $O(N)$ time complexity, where N is the total pixel number of input image I. The SLIC algorithm runs with T_C times repeatedly, and the time complexity is still $O(N)$ by the reason of constant T_C. The time complexity of Otsu is related to the constant, which is $O(1)$. The computation of each GS value includes two steps: wVar is computed in $O(N)$ time, and MI is calculated with $O(s)$, where s is the number of superpixels. As shown in Algorithm 1, the maximum superpixel number is smaller than $\lfloor(N)^{1/2}\rfloor$. Thus, the whole superpixel generation method is done in the time complexity of $O(N)$ even with constant T_C iterations.

Algorithm 4 MSG Merging Method

Input: Superpixel set $\{R_1,\cdots,R_m,\mathcal{B}\}$ of image I, Multi-subgraph $\{G_1^1,\cdots,G_z^1\}$, maximum number of merging set is T_M.

Output: Merging result $\{\Omega_1,\cdots,\Omega_z\}$.

```
for $x=1$ to $z$ do
    Set region set $\Omega_x=\{G_x^1\}$.
    if $G_x^1$ is an isolated point then
        Pass.
    else
        Extract color, intensity and texture features for the superpixels corresponding to all vertexes in $V_x$.
        Compute the weight of each edge by formula (13), then weighted subgraph $G_x=(V_x,E_x^1,W_x^1)$ is constructed.
        Compute the sum of variance of $\Omega_x$ and set it to $\Delta_x$.
        Set $\kappa=0.02$, $t=1$, $\mathcal{M}_x=\varnothing$.
        repeat
            Run SSGM algorithm with parameter $\kappa$ and obtain the merging result $\mathcal{M}_{x_t}$, compute $|\mathcal{M}_{x_t}|$, which is the total number of all regions in $\mathcal{M}_{x_t}$.
            if $\mathcal{M}_{x_t}\notin\mathcal{M}_x$ then
                Set $\mathcal{M}_x=\mathcal{M}_x\cup\mathcal{M}_{x_t}$.
                Compute the variance $\Delta_x^t$ of $\mathcal{M}_{x_t}$.
                if $\Delta_{x_t}\le\Delta_x$ then
                    Set $\Omega_x=\mathcal{M}_{x_t}$.
                end if
                $t+=1$.
            end if
            $\kappa+=0.02$.
        until $t>T_M$ or $|\mathcal{M}_{x_t}|=1$.
    end if
end for
Output Merging result $\{\Omega_1,\cdots,\Omega_z\}$.
```

For all bands B, erosion and dilation operations are done in each pixel, so the time complexity of opening and closing operations is $O(2BN)$. Because opening and closing operations run P times iteratively, the two operation's time complexity is $O(2BPN)$. The most time-consuming process in the PCA method is the eigenvalue decomposition, the time complexity of which is $O(P^2N)$ [66]. The value of $\lambda_{\max}$ is the minimum edge in all bounding boxes, the maximum of which approximately ranges from $N^{(1/4)}$ to $(T_C)^{(1/2)}N^{(1/4)}$ by the reason of superpixel size approximately ranging from $\lfloor(N)^{1/2}/T_C\rfloor$ to $(N)^{1/2}$. Thus, the whole background removal method can run with the complexity $O(T_CN^{(3/2)})$. This is because the band B is a constant and $P=\lfloor(\lambda_{\max}-\lambda_{\min})/\lambda'\rfloor+1$, where $\lambda_{\min}$ and λ' are constants. By the constant T_C, the time complexity reduces to $O(N^{(3/2)})$.

With the fact that at most m edges connect with the background vertex, and all edges that are removed from the edge set E need $O(m)$ time. In the next step, MSG creation visits all vertexes in V, the time complexity of which is $O(m)$ by $|V|=(m+1)$. Then, the whole complexity of the proposed MSG creation method is $O(m)$.

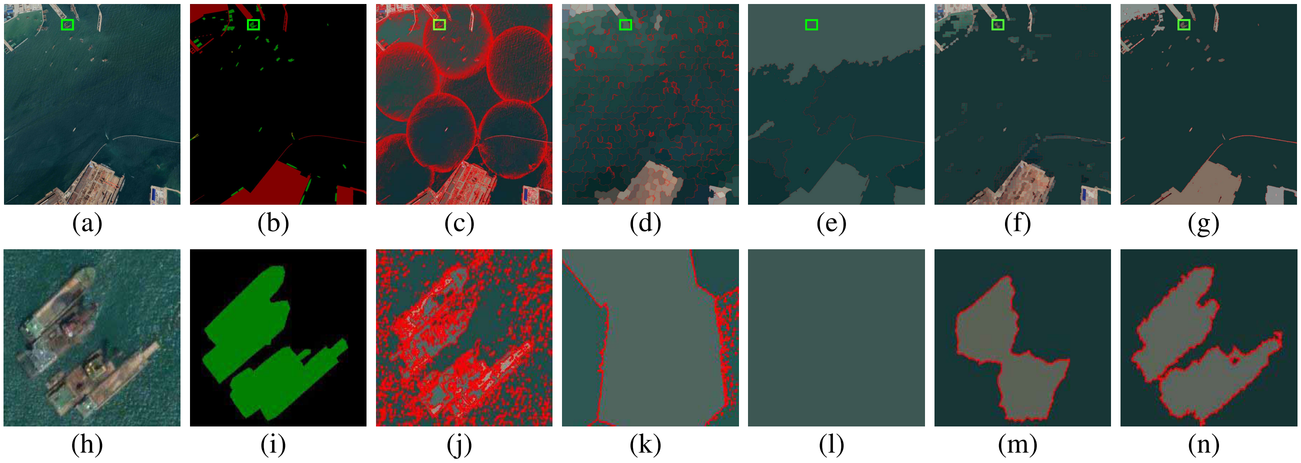


Fig. 5. (a) Original Google Earth dataset and (b) Ground Truth. Comparison of segmentation results in (c) MS, (d) SMSI, (e) IHIH, (f) CMIS, and (g) MSMSG methods. (h)–(n) Local magnification images of (a)–(g).

Feature extraction for each superpixel needs to visit each pixel except the isolated superpixels, and the running time is smaller than $O(N)$. The SSGM algorithm, $O(|V_x|\log(|V_x|))$ [18] in subgraph $\widetilde{G}_x$, is the main operation in the merging process. Suppose that each subgraph repeats t_x times to reach the stopping condition, and SSGM performs with the computation time of $O(t_x|V_x|\log(|V_x|))$. Thus, the sum time of the proposed multiscale SSGM method is smaller than $O(t_{\max}(m^2))$, which is seen in Theorem 2. The proof of Theorem 2 is given in Appendix B. Because the variance computation occurs in the case of nonrepeated merging results, the maximum number of which is a constant value T_M. The optimal merging selection visits all pixels except the isolated points in the whole image at most T_M times, by which the sum time of variance computation is $O(N)$. To sum up, the time complexity of the whole MSG merging method is limited at $O(t_{\max}m^2)$ or $O(N)$.

By the above analysis and the fact of $t_{\max} \ll (N)^{1/2}$, the whole complexity time of our proposed segmentation method is $O(N^{(3/2)})$ or $O(N^{(1/2)}m^2)$. That is to say, the time complexity of the whole MSMSG-based segmentation method is $O(N^{(3/2)})$ when $m < (N)^{1/2}$, which is $O(N^{(1/2)}m^2)$ when $m > (N)^{1/2}$.

Theorem 2: The sum time of the multiscale SSGM method is smaller than $O(t_{\max}m^2)$, where $t_{\max}$ is the maximum number of repetitions of the SSGM method and m is the total number of superpixels in the raw image.

IV. Experiment

A. Datasets and Settings

To fully analyze the computational efficiency and segmentation performance of the proposed method, we use three different sources of ocean remote sensing images, which are collected from the Bohai Sea [67], [68]. The size of the first one is 1567 × 3177, which is collected with ArcGIS and located from (121°43′52″ E, 39°00′16″ N) to (121°45′20″ E, 38°57′39″ N). The WDMI dataset is the second ocean dataset with a size of 2272 × 2376, and it ranges from (121°00′36″ E, 40°44′17″ N) to (121°01′18″ E, 40°43′42″ N). We capture an ocean image with a large size from Google Earth to verify all performances of our proposed method in the former two images, the longitude and the latitude of which range from 121°00′36″ E to 121°39′14″ E and 38°57′47″ E to 38°00′36″ E, respectively. The Google Earth dataset, as displayed in Fig. 5(a), has the large size of 4800 × 5344.

In the whole method, we set the normalized factor $\xi' = 10$ in (1) and the multiple $\eta = 50$. The total iteration number of the coarse superpixel generation method is $T_C = 20$. In addition, we set the increasing step $\lambda' = 2$, and let the coefficients of all features be $\gamma_1 = \gamma_2 = \gamma_3 = (1/3)$ [19]. In the MSG merging process, each subgraph can sufficiently obtain the optimal result when the number of merging sets is no more than $T_M = 15$.

When the segmentation results are obtained with our proposed MSMSG method, we use the boundary recall (BR) and the boundary precision (BP) to evaluate the boundary performance of all segments [18]. To evaluate the precision of all segments, we combine the rule of oversegmentation error (OE) with undersegmentation error (UE) of segmentation results [69]. We compare our proposed segmentation method with the mean-shift (MS) [70], the segmentation with minimum spanning tree (SMST) method [5], the segmentation method combining with intrasegment homogeneity and intersegment heterogeneity (IHIH) [35], and the CMSuG-based image segmentation (CMIS) [18].

In addition, we use the running time to measure the efficiency of our method. All experiments are done with Python 3.6 in the AMD Ryzen 9 5900X 12-Core CPU at 3488 MHz.

B. Results

In this section, we compare the segmentation results with our MSMSG method and the other four methods. The performances are shown in Table I clearly, in which each bold value is the best performance of every metric. The best BP values of the three datasets declare that the rates of correct boundary pixels are the highest. BP is the percentage of the positively true boundaries in all positively predicted boundaries. We can see that all BP values of our method are higher than other

TABLE I

PERFORMANCE METRICS OF DIFFERENT SEGMENTATION METHODS AT ALL DATASETS

Dataset	Method	BR	BP	UE	OE	TIME / s
ArcGIS	MS	**0.8179**	0.0477	0.0190	0.8689	2.3×10^4
	SMSI	0.1311	0.0081	0.0144	0.9638	1.2×10^4
	IHIH	0.1997	0.0574	0.0195	0.5874	8.5×10^3
	CMIS	0.1391	0.0169	0.0107	0.9847	3.1×10^3
	MSMSG	0.5115	**0.2376**	**0.0061**	**0.0145**	$\mathbf{2.5 \times 10^2}$
WDMI	MS	**0.5684**	0.0326	0.0241	0.8702	9.9×10^3
	SMSI	0.1267	0.0108	0.0312	0.9780	9.3×10^3
	IHIH	0.1230	0.0562	0.0312	0.2822	1.2×10^4
	CMIS	0.1192	0.0410	0.0334	0.1677	2.6×10^4
	MSMSG	0.5041	**0.0852**	**0.0187**	**0.0390**	$\mathbf{3.0 \times 10^2}$
Google Earth	MS	**0.8605**	0.0980	0.0186	0.8604	1.1×10^5
	SMSI	0.1639	0.0173	0.0259	0.9846	2.3×10^4
	IHIH	0.1961	0.1371	0.0998	0.6055	2.9×10^4
	CMIS	0.2845	0.0857	**0.0113**	0.1708	1.7×10^5
	MSMSG	0.5967	**0.2614**	0.0131	**0.0473**	$\mathbf{6.9 \times 10^3}$

methods. It indicates that the positively predicted boundaries with our method are quite high. Each BR value is the highest by using the MS method on every dataset, which occurs at the expense of segmentation accuracy. As the example segmentation vision of the Google Earth dataset shown in Fig. 5(c), the apparent redundant boundaries are displayed. This value obtained with the proposed MSMSG-based segmentation method is in second place on the WDMI dataset and is in third on ArcGIS and Google Earth datasets.

With our segmentation method, the results are good in ArcGIS and WDMI datasets by the lowest UE and OE values. Although the UE of our method on the Google Earth dataset is not the lowest, the remarkable OE demonstrates that the number of segments hitting the ground truth is the highest with our method. The undersegmentation results of SMSI and IHIH are vividly shown in Fig. 5(k) and (l) by the fact that the whole object is segmented into a few groups. However, the good result is plotted in Fig. Fig. 5(n) with the proposed MSMSG segmentation method. In addition, the CMIS method performs well both in small-size ocean remote sensing images and nature images [18]. However, the terrible performance of the CMIS method in Table I tells us that the method is not applied to large-size remote sensing image segmentation.

The running time of our whole method is a summary of the four steps, which are superpixel generation, background removal, MSG creation, and MSG merging methods. In the last column of Table I, the running time of all segmentation methods is compared. For each dataset, our method is the most efficient because of the lowest running time. Our method has the best BP and OE performances, accompanied by the shortest running time on the three datasets, the total running time of which is far less than others. Although the BR of our method is worse than MS, and UE is worse than the CMIS method for the Google Earth dataset, the total time is far less than MS and CMIS methods.

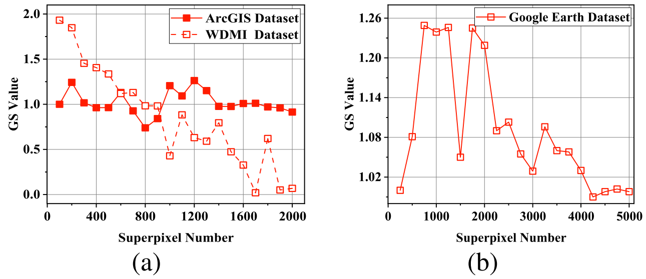


Fig. 6. GS values of multiscale coarse superpixels in (a) ArcGIS and WDMI datasets, and (b) Google Earth dataset.

Totally, our proposed method is not only high-performance but also high-efficient with good stability in all datasets. By the reason of the two excellent measures of our method, the segmentation tasks of large-size ocean remote sensing images can be done by using them efficiently.

V. DISCUSSION

Our segmentation method contains a coarse-to-fine superpixel generation technique, a background removal strategy, an MSG construction, and an MSG merging method. Each step has benefits in the segmentation task of ocean remote sensing datasets; we explore what affects the segmentation results in this section. There are some parameters in the segmentation algorithm; we analyze the segmentation results on the side of parameters T_C and T_M. Moreover, we analyze the determination of parameters of λ' and η. Because of the parameter analyses, the resegmentation percentage is selected with 5%, and the increasing step length of κ is 0.02. To verify the strong adaptability of our proposed MSMSG method, we compare the performances on another quantitative dataset with the other four techniques.

A. Analysis of Coarse-to-Fine Superpixel Generation Algorithm

The superpixels generated with the proposed coarse-to-fine superpixel generation method are compared with the superpixels generated by only using the coarse superpixel generation method on three datasets. Before that, we analyze how the optimal scales are selected with the proposed method.

Optimal coarse superpixels depend on the multiple GS values, which are computed with the multiscale coarse superpixels. Fig. 6 illustrates the GS values of different scales, in which Fig. 6(a) shows the results describing the ArcGIS and WDMI datasets by the same increasing step 100 of superpixel number. The increasing step of the Google Earth dataset is 250, computed with the total pixel number of the input image, the GS results of which are shown in Fig. 6(b). It is obviously seen that the optimal superpixel number of the ArcGIS dataset is 800 and that of the WDMI dataset is 1700. When the

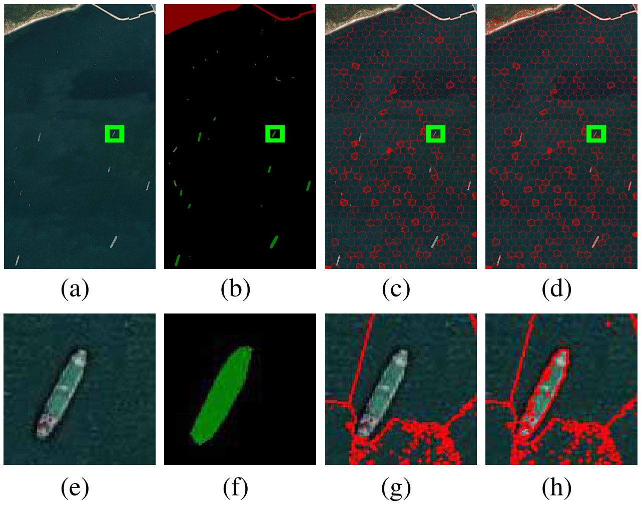


Fig. 7. Comparison of coarse and fine superpixel results. (a) Original ArcGIS dataset. (b) Ground truth of the ArcGIS dataset. (c) Optimal coarse superpixels with multiscale SLIC. (d) Fine superpixels with the multiscale Otsu method. (e)–(h) Local magnification images of (a)–(d).

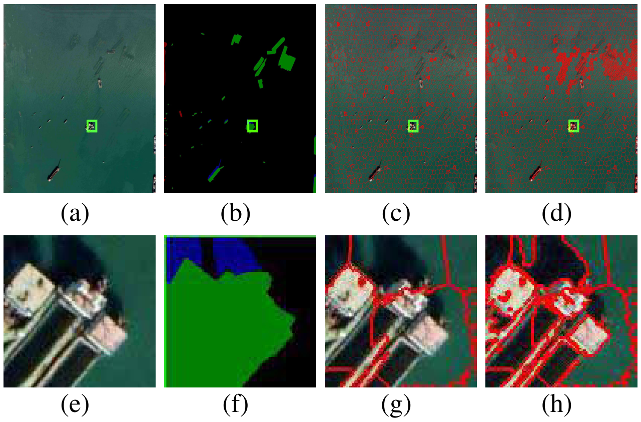


Fig. 8. Comparison of coarse and fine superpixel results. (a) Original WDMI dataset. (b) Ground truth of the WDMI dataset. (c) Optimal coarse superpixels with multiscale SLIC. (d) Fine superpixels with the multiscale Otsu method. (e)–(h) Local magnification images of (a)–(d).

superpixel number is 4250, the GS value is the minimum applied on the Google Earth dataset. In the fine step, the worse superpixels that count in the top 5% of all superpixels need to be resegmented on the three datasets. Figs. 7 and 8 show the coarse-to-fine superpixel results on the ArcGIS and WDMI datasets lively. It is easy to see that the fine superpixels [see Figs. 7(d) and 8(d)] are superior to coarse superpixels [see Figs. 7(c) and 8(c)] due to the apparent UEs that occur in the coarse superpixels.

To quantitatively analyze the performance of our coarse-to-fine superpixel method, we compare the performance of coarse superpixels with that of the coarse-to-fine superpixels and display the results in Table II. For each segment generated by our superpixel generation method, we use the BR, BP, UE, and achievable segmentation accuracy (ASA) [7] to evaluate the performance. For convenience, we use "Fine" to represent our coarse-to-fine method. Although the running time of the coarse-to-fine method is slightly higher than that of the coarse superpixel generation method, all performances are improved from coarse superpixels to fine superpixels, which is shown in Table II. Especially, the BR and BP values generated after fine operation increase to twice those of the coarse superpixels. Hence, the superpixels generated by the coarse-to-fine method on the three datasets all have good boundary adherence. The UE and ASA improve slightly. However, the improvement is tremendous in the dataset with a large size, even with increasing 0.01. By comparing the methods from the results shown in Table II, it can be observed that all performances of the coarse-to-fine superpixel generation method outperform those of coarse superpixel generation method except for the running time. This indicates that our method performs well not only on large objects but also on small objects in large-size ocean remote sensing datasets.

TABLE II
PERFORMANCE METRICS OF BACKGROUND REMOVAL METHODS AT ALL DATASETS

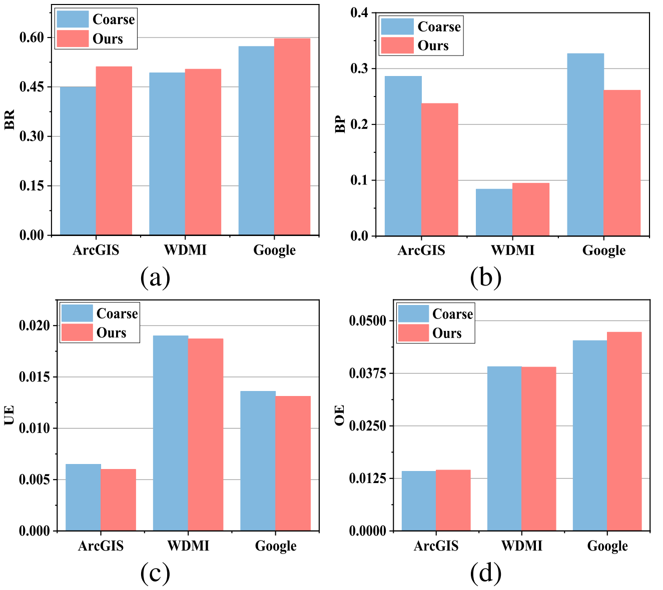
Dataset	Method	BR	BP	UE	ASA	TIME / s
ArcGIS	Coarse	0.2085	0.0119	0.0107	0.9892	**129.80**
	Fine	**0.4903**	**0.0292**	**0.0063**	**0.9936**	136.24
WDMI	Coarse	0.2576	0.0165	0.0207	0.9792	**152.29**
	Fine	**0.5073**	**0.0394**	**0.0182**	**0.9817**	166.04
Google Earth	Coarse	0.2994	0.0332	0.0131	0.9868	**997.88**
	Fine	**0.5462**	**0.0602**	**0.0094**	**0.9905**	1071.92

TABLE III
RUNNING TIME OF THE WHOLE SEGMENTATION METHOD WITH DIFFERENT SUPERPIXEL GENERATION METHODS AT ALL DATASETS

Method	Time / s (ArcGIS)	Time / s (WDMI)	Time / s (Google)
Coarse	**2.4** $\mathbf{\times 10^{2}}$	**2.8** $\mathbf{\times 10^{2}}$	**5.5** $\mathbf{\times 10^{3}}$
Ours	2.5 $\times 10^{2}$	3.0$\times 10^{2}$	6.9 $\times 10^{3}$

Affected by the superpixels, the final segmentation results may differ from the different superpixels. With the coarse superpixels and fine superpixels, the final segmentation results are compared in Fig. 9. The high BR values indicate that few ground-truth boundaries are missed on the three datasets. Higher UE values based on the coarse superpixel generation method state that the errors in superpixels are still in the final results on all datasets. Fewer undersegmentation errors in our final segmentation results lead to lower BP values because of the redundant boundaries. We can see that our method brings about slightly higher OE values than the coarse method on the Google Earth dataset. This case is because the undersegmentation segments hit the ground truth with the number of pixels.

The total running time of our whole segmentation framework based on the two superpixel generation methods is illustrated in Table III. We can see that the time of our method is slightly higher than that of the segmentation framework



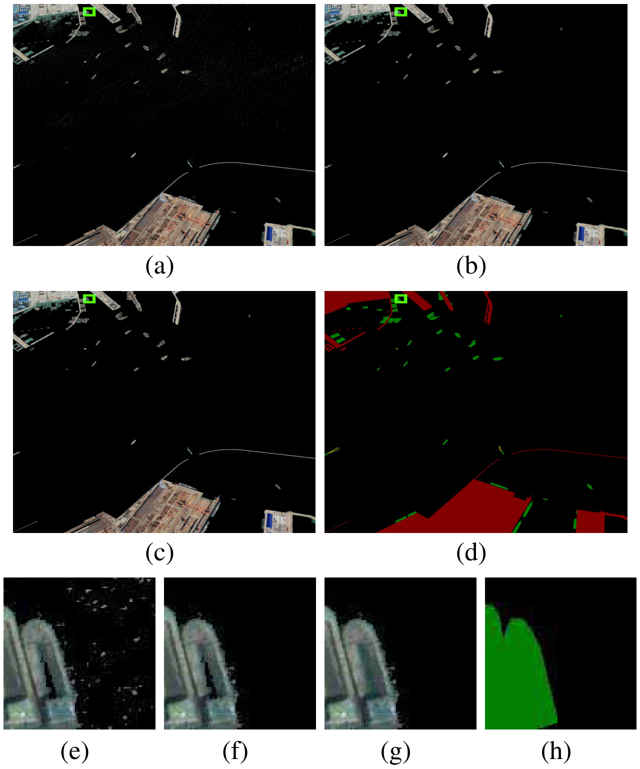
(a) (b)

(c) (d)

Fig. 9. Performance comparison of final segmentation based on coarse superpixels and coarse-to-fine superpixels on three datasets. (a) BR. (b) BP. (c) UE. (d) OE.

without using the fine resegmentation method. This is because the resegmentation method subdivides the worse superpixels into a number of disconnected segments, and the total running time is related to the number of segments.

Thus, our coarse-to-fine method can segment both small and large objects well at the expense of running time.



(a) (b)

(c) (d)

(e) (f) (g) (h)

Fig. 10. Comparison of background removal results in different steps for the Google Earth dataset. (a) Background removal results with GMM. (b) Background removal results with the joint of GMM and the 1st-MM filtering method. (c) Background removal results with our whole background removal method. (d) Ground Truth of the Google Earth dataset. (e)–(h) Local magnification images of (a)–(d).

B. Analysis of Background Removal Strategy

The proposed background removal strategy contains GMM, first MM, and second MM steps; we discuss how the different background removal steps of the proposed background removal method influence the superpixels and final segmentation results.

Fig. 10 vividly shows the background removal results with the three steps, respectively. As the local magnification image illustrated in Fig. 10(e), there are lots of background pixels wrongly classified into the foreground. By comparing the background removal vision in Fig. 10(e) and (f), the clutter pixels are eliminated with the first MM step due to the advantage of opening operation. As seen in Fig. 10(e) and (g), an obvious classification error, the hole in the ship, is filled with the closing operation of MM. These visual results demonstrate that our background removal method is more suitable for ocean remote sensing datasets with indistinguishable foreground and background areas than the just GMM method.

To quantitatively evaluate the performance of the background removal results with different removal steps, we compute the precision (P), recall (R), and F1-score (F1) values [71]. The performance of correct predicted background pixels constantly increases from GMM to the second MM step, which is exhibited in Table IV. For convenience, let “1st-MM” be the background method with the joint of GMM and first MM operation steps and “2nd-MM” be the joint of GMM, first MM, and the second MM steps. The increasing values of R state that the total number of predicted background pixels hitting the real background pixels keeps growing. High F1 scores computed after running the second MM method declare that the background classification with our method is not only better than the result generated with the just GMM method but also superior to that generated with the first MM filtering step. It is worth noting that the running time of our background removal method is higher than the only GMM and 1st-MM methods, which is because the second MM filtering operation is based on the two steps. The total pixel number of the ArcGIS dataset is nearly equal to that of the WDMI dataset, but the running time is also related to the maximum size of the kernel in MM filtering. As evident from Table IV, the proposed background removal strategy provides a slightly better F1 score than GMM and the first MM methods of the Google Earth dataset. For large-size ocean remote sensing datasets, these tiny improvements have great benefits.

The distinct superpixels generated with different background removal methods result in various segmentation results, and the quantitative performances are compared in Fig. 11. In the picture, the BR values increase with the addition of the background removal steps. The OE values decrease step by step on all datasets though the UE of the Google Earth dataset rises slightly. It reveals that the clutters and holes generated with GMM are successfully removed by the two-step MM filtering operation. Applying the three background removal methods in our segmentation framework, the total running time of which is illustrated in Table V. For the ArcGIS and Google

TABLE IV

PERFORMANCE METRICS OF BACKGROUND REMOVAL METHODS AT ALL DATASETS

Dataset	Method	P	R	F1	TIME / s
ArcGIS	GMM	0.999322	0.988223	0.993742	**12.15**
	1st-MM	0.999370	0.993395	0.996374	23.41
	2nd-MM	**0.999401**	**0.993409**	**0.996396**	34.89
WDMI	GMM	0.986389	0.968635	0.977432	**24.92**
	1st-MM	**0.986572**	0.977623	0.982077	50.19
	2nd-MM	0.986517	**0.978412**	**0.982448**	83.52
Google Earth	GMM	**0.998533**	0.982443	0.990422	**50.93**
	1st-MM	0.998354	0.991986	0.995159	113.88
	2nd-MM	0.998513	**0.992489**	**0.995491**	181.18

TABLE V

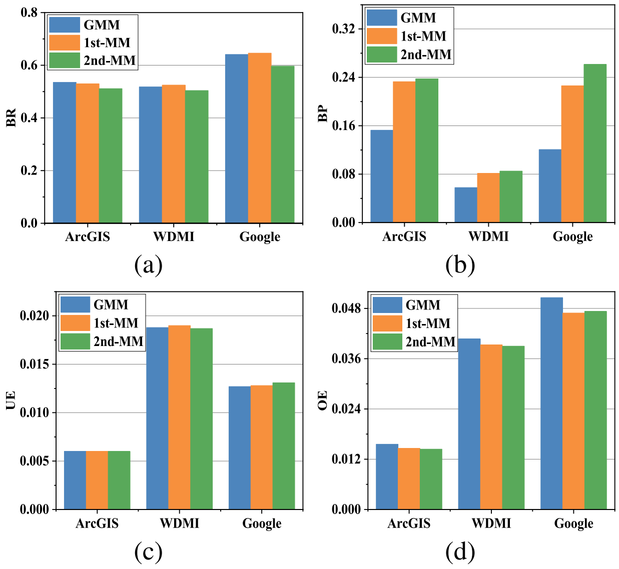
RUNNING TIME OF THE WHOLE SEGMENTATION METHOD WITH DIFFERENT BACKGROUND REMOVAL STEPS AT ALL DATASETS

Background	Time / s	Time / s	Time / s
Removal	(ArcGIS)	(WDMI)	(Google)
GMM	$2.7\ \times 10^2$	$\mathbf{2.6\ \times 10^2}$	$7.2\ \times 10^3$
1st-MM	$2.6\ \times 10^2$	$2.2\ \times 10^3$	$7.1\ \times 10^3$
2nd-MM	$\mathbf{2.5\ \times 10^2}$	$3.0\ \times 10^2$	$\mathbf{6.9\ \times 10^3}$

TABLE VI

PERFORMANCE METRICS OF BACKGROUND REMOVAL METHODS AT ALL DATASETS

Dataset	Method	BR	BP	UE	OE	TIME / s
ArcGIS	SG	**0.6139**	0.2488	**0.0036**	0.0152	$1.4\ \times 10^3$
	MSG	0.5115	**0.2376**	0.0061	**0.0145**	$\mathbf{2.5\ \times 10^2}$
WDMI	SG	0.4138	0.0803	0.0224	**0.0310**	$6.3\ \times 10^3$
	MSG	**0.4898**	**0.0947**	**0.0188**	0.0390	$\mathbf{3.0\ \times 10^2}$
Google Earth	SG	0.4432	**0.2840**	0.0913	**0.0129**	$4.2\ \times 10^4$
	MSG	**0.5967**	0.2614	**0.0131**	0.0473	$\mathbf{6.9\ \times 10^3}$



(a) (b) (c) (d)

Fig. 11. Performance comparison of segmentation results with different background removal methods. (a) BR. (b) BP. (c) UE. (d) OE.

Earth datasets, the running time of our proposed method is the lowest. The whole segmentation framework with only GMM as the background removal method has the highest efficiency on the WDMI but with worse performances except BR values than ours.

In summary, our background removal strategy can not only help to classify the foreground and background areas but also well resegment the superpixels that contain both background and foreground pixels in ocean datasets, which results in good segmentation results with higher efficiency. As a result, our background removal method has a nonnegligible advantage in the segmentation of ocean remote sensing images with large sizes.

C. Analysis of Multisubgraph Construction Method

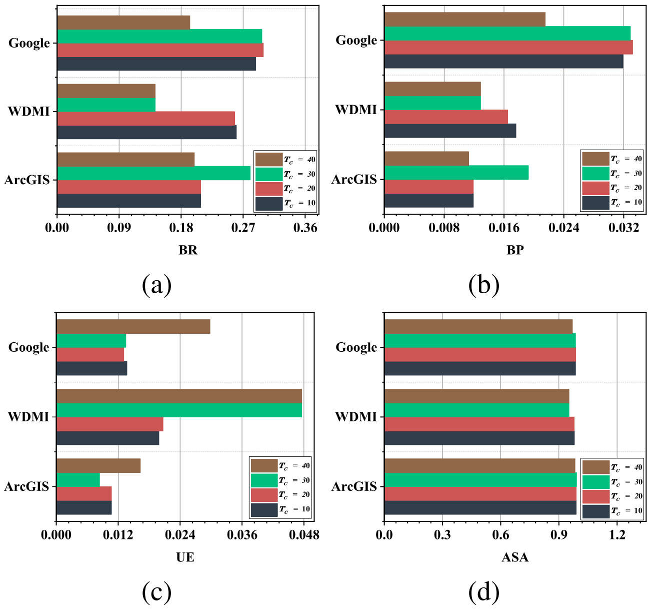
MSG construction is the key to executing MSG merging; the section discusses the segmentation results with our proposed segmentation model based on MSG and just a single graph (SG). Table VI lists the performance comparison of the two methods MSG performs better than SG on the WDMI dataset by all properties. Although the BP and OE values generated with MSG are better than SG on the ArcGIS dataset, the BR and UE reverse. In the contrast, the four opposite performances completely are displayed in the table on the Google Earth dataset.

What cannot be ignored is that the total running time of the MSG-based segmentation method is far less than the SG-based segmentation method. Thus, our proposed MSG method can not only obtain better segmentation results but also greatly reduce the time consumption of the segmentation on ocean remote sensing images.

D. Analysis of Parameters

1) Analysis of Parameter T_C: The optimal coarse superpixel generation method is up to iteration number T_C. The final segmentation results perform a difference with the different coarse segmentation numbers as the various iteration numbers. In this section, we analyze how the parameter T_C influences the final segmentation results.

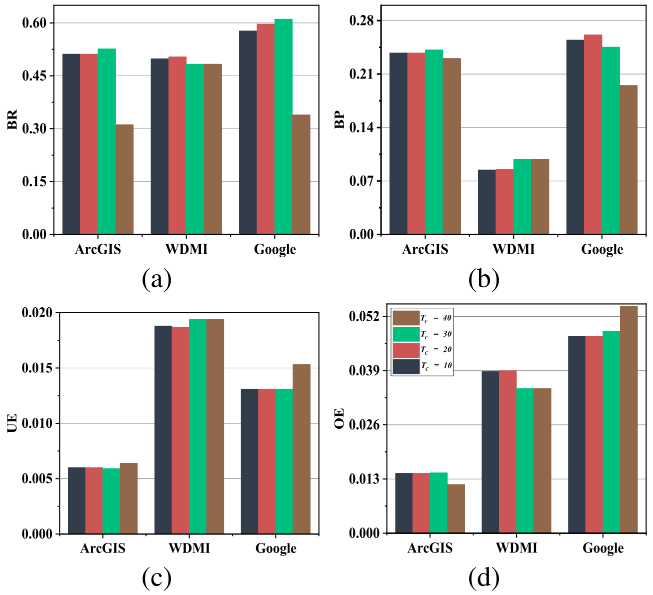
To study the coarse superpixel generation method using different iteration numbers, four performances are carried out, which are plotted in Fig. 12. On the ArcGIS dataset, all performances are the best when $T_C = 30$, which is because the optimal segmentation number is selected with 1450. The four performances indicated that selecting $T_C = 10$ is the most suitable for the WDMI dataset, whereas $T_C = 20$ is the best for the Google Earth dataset. The coarse superpixels



(a) (b)

(c) (d)

Fig. 12. Performance of optimal coarse superpixels with different parameter T_C's on the three datasets. (a) BR. (b) BP. (c) UE. (d) ASA.



(a) (b)

(c) (d)

Fig. 13. Performance of the final segmentation results with different superpixels generated by increasing parameter T_C on the three datasets. (a) BR. (b) BP. (c) UE. (d) OE.

TABLE VII

Running Time of the Whole Segmentation Method With Different Iteration Numbers T_C at All Datasets

	Time / s (ArcGIS)	Time / s (WDMI)	Time / s (Google)
$T_C = 10$	$\mathbf{2.1\ \times 10^2}$	$4.1\ \times 10^2$	$\mathbf{6.6\ \times 10^3}$
$\boldsymbol{T_C = 20}$	$2.5\ \times 10^2$	$\mathbf{3.0\ \times 10^2}$	$6.9\ \times 10^3$
$T_C = 30$	$6.0\ \times 10^2$	$2.2\ \times 10^3$	$7.1\ \times 10^3$
$T_C = 40$	$1.3\ \times 10^3$	$2.5\ \times 10^3$	$3.0\ \times 10^5$

TABLE VIII

Running Time of the Whole Segmentation Method With Different Merging Parameter T_M at All Datasets

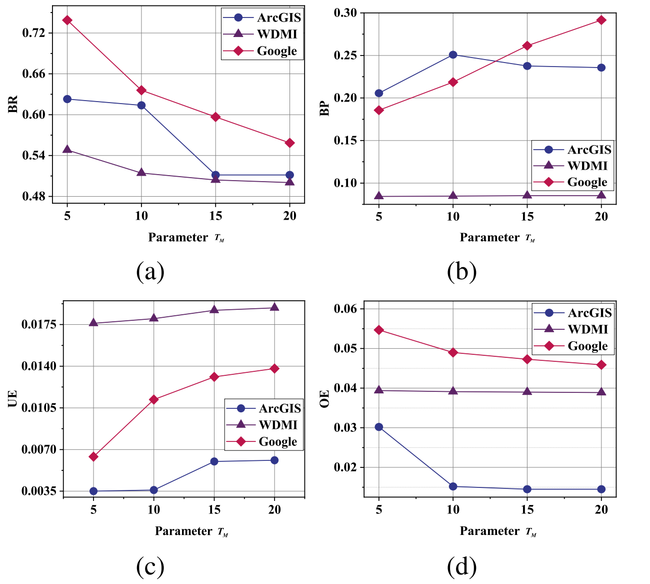
	Time / s (ArcGIS)	Time / s (WDMI)	Time / s (Google)
$T_M = 5$	$\mathbf{2.0\ \times 10^2}$	$\mathbf{2.6\ \times 10^2}$	$\mathbf{2.8\ \times 10^3}$
$T_M = 10$	$2.3\ \times 10^2$	$3.2\ \times 10^2$	$4.8\ \times 10^3$
$\boldsymbol{T_M = 15}$	$2.5\ \times 10^2$	$3.0\ \times 10^2$	$6.9\ \times 10^3$
$T_M = 20$	$4.1\ \times 10^2$	$3.6\ \times 10^2$	$1.1\ \times 10^4$

are the worst when the number of iterations is 40, which is clearly shown in the highest UR values on all datasets. Only comparing the coarse superpixels makes it difficult to select the perfect superpixel segmentation number. Then, we compare the performances that the final segmentation results achieve by using these different superpixels.

Fig. 13 shows the comparisons of all segmentation results with different T_C's, and the color of each bar is corresponding to the settings in Fig. 12. We can observe that the four worst measure values are gained in the case of $T_C = 40$ on the ArcGIS and Google Earth datasets, which accompany the highest running time listed in Table VII. Regarding the running time with T_C set as 10, 20, and 30, respectively, the speed of the proposed method has an apparent disadvantage on the WDMI dataset when $T_C = 30$. It is worth noting that there is a slight difference in the running time between $T_C = 10$ and $T_C = 20$. However, the BR and BP values as $T_C = 20$ are higher than that with $T_C = 10$ on the Google Earth dataset obviously. The side fact indicates that the optimal coarse superpixel selection from 20 is the most suitable for all large-size ocean remote sensing images.

2) Analysis of Parameter T_M: In the proposed segmentation framework, parameter T_M controls the final merging results. We analyze the merging results with the growth of the maximum number of the merging set by the increasing step of 5. The variation tendency of the four performances is plotted in Fig. 14. In the figure, the BR and OE values decrease with the growth of parameter T_M on three datasets. Compared with the distribution of the UE curves in Fig. 14(c), they have similar increasing trends on the three datasets. BP values have no evident changes by using the four different parameters on the WDMI dataset. That value curve raises gradually with the increase in the T_M value on the Google Earth dataset. Our segmentation framework has a slight reduction in BP performance from $T_M = 15$ to $T_M = 20$ on the ArcGIS dataset, and the highest of which is achieved in $T_M = 10$. It is interesting to note that all curves flatten out from $T_M = 15$. Globally, the changes in all performances are the minimum from $T_M = 15$ to $T_M = 20$ on all datasets.

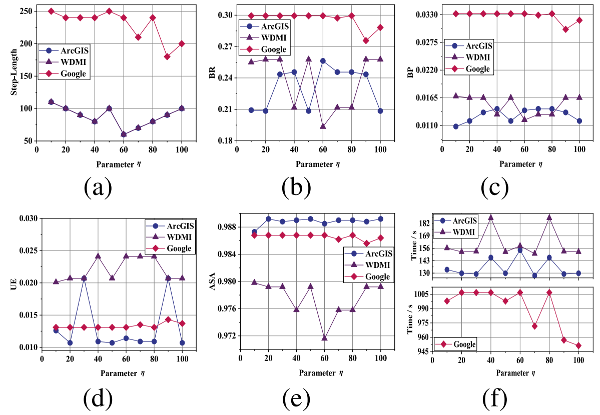
The computation time of the proposed segmentation framework with all merging parameters is summarized in Table VIII. With the increasing of T_M, the running time of the whole



(a) (b)

(c) (d)

Fig. 14. Performance of the final segmentation results with different merging parameters T_M. (a) BR. (b) BP. (c) UE. (d) OE.



(a) (b) (c)

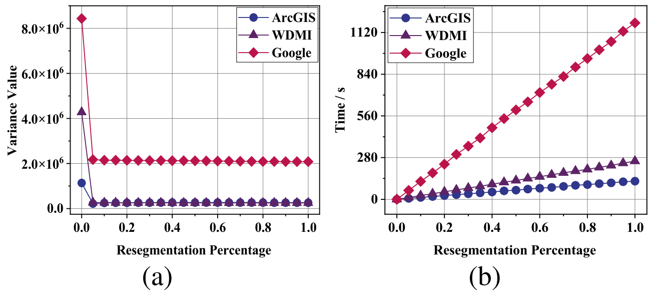
(d) (e) (f)

Fig. 15. (a) Step length of segmentation number in the 20 superpixels. Performance of optimal coarse superpixel on the three datasets with different multiple η's. (b) BR. (c) BP. (d) UE. (e) ASA. (f) Total running time.

segmentation method grows gradually. By the above analysis, we select $T_M = 15$ as the maximum number of each merging set to select the optimal merging results.

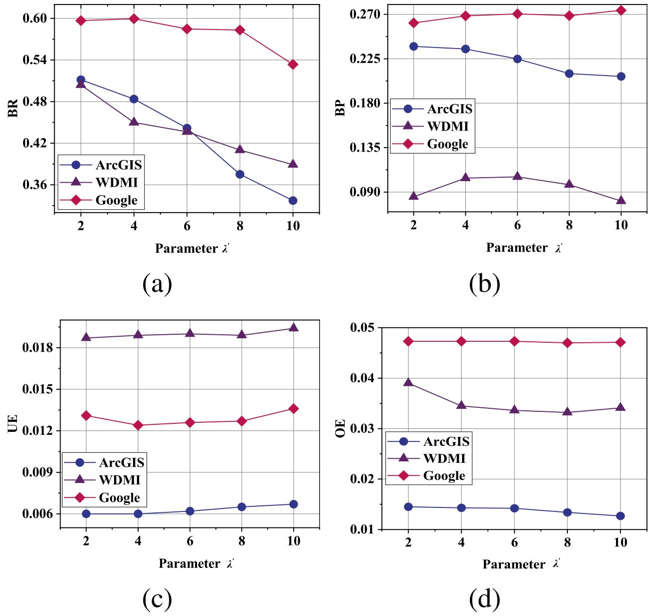
3) Analysis of Parameter η: In order to ensure that the total number of T_C superpixels are integers, the parameter η is to control the original segmentation number and the increasing step length. As shown in Fig. 15(a), when parameter η is selected, the step length is determined. On the whole of the running time on the three datasets, we can see that the time is smaller when η is 50, 70, 90, and 100, respectively, which is shown in Fig. 15(f). When η is 90 or 100, the four performances are obviously lower than the other two on the Google Earth dataset. Besides, the BP and UE values generated with $\eta = 50$ are better than that with 70 on the WDMI dataset clearly. In view of the above analysis, we select the stable and good performance of $\eta = 50$ to control the step length of the number of superpixel segmentations.

4) Analysis of Resegmentation Percentage: Compared with the variance values of coarse superpixels on three datasets,



(a) (b)

Fig. 16. (a) Variance values and (b) total running time of fine superpixels with different resegmentation percentages.



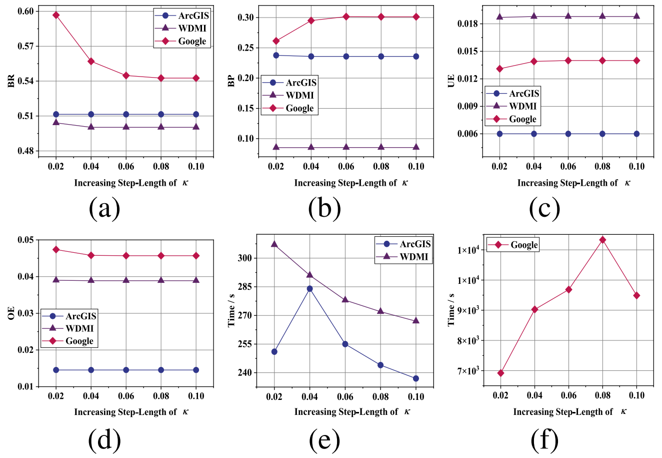
(a) (b)

(c) (d)

Fig. 17. Performance of the final segmentation results by a different increasing parameter λ' on the three datasets. (a) BR. (b) BP. (c) UE. (d) OE.

we can see that the variance values greatly decrease when percentage values are 5%. As shown in Fig. 16, the variances have no evident changes from percentage 5% to 100%. When the percentage value is 100%, all superpixels need to be resegmented. However, the purpose of the proposed coarse-to-fine superpixel generation method is to resegment the worse superpixels but not all superpixels, which is to solve the difficulties of segmenting small objects in large ocean remote sensing images. With the increasing resegmentation percentage value, the resegmentation time raises on the three datasets. For the balance of variance and running time, we adopt the 5% as the resegmentation percentage.

5) Analysis of Parameter λ': In the background removal algorithm, λ' can control the initial size and the increasing step length of the filtering kernel. Fig. 17 presents the variation of segmentation performance with different values of λ', which is set from 2 to 10. With the increase in λ', BR values of all datasets decrease distinctly. On the ArcGIS dataset, the BP value is the highest at $\lambda' = 2$. A very small incremental value of the BP curve happens on the Google Earth dataset. For ArcGIS and WDMI datasets, the UE increases and OE decreases with the rising of λ'. That is because small objects with small sizes are more likely to be removed from the



(a) (b) (c)

(d) (e) (f)

Fig. 18. Performance of segmentation results by using different increasing step-length values. (a) BR. (b) BP. (c) UE. (d) OE. (e) Total running time on the ArcGIS and WDMI datasets. (f) Total running time on the Google Earth dataset.

foreground by larger λ' on the ArcGIS and WDMI datasets. To protect the small objects, λ' is set as 2 tactfully.

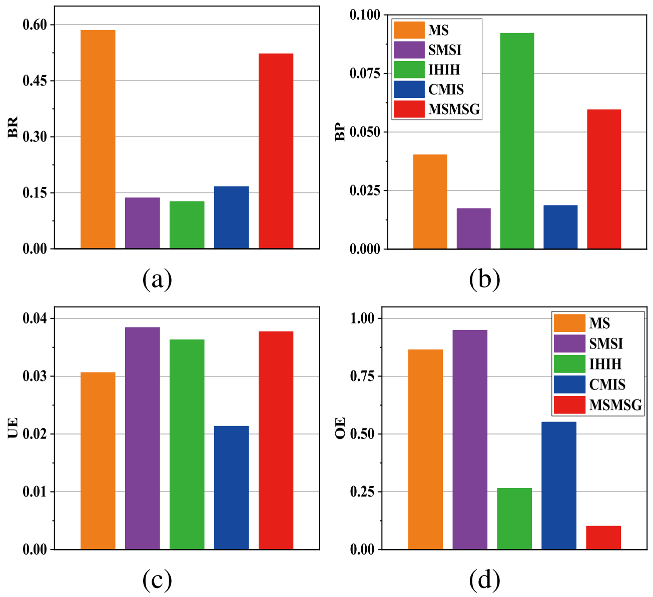
In addition, the total running times are 2.5×10^2, 3.0×10^2, and 6.9×10^3 with each λ' on the ArcGIS dataset, the WDMI dataset, and the Google Earth dataset, respectively. It means that the running time is independent of the parameter λ'.

6) Analysis of Increasing Step Length of κ: The merging parameter κ is the key to merging two similar objects into a cluster. The higher the κ, the more likely is to merge. If κ is too large, the UE is easy to occur. Like Fig. 18(c), the UE value with step length 0.02 is smaller than the other step-length values on the Google Earth dataset. Although the performance of the OE value is higher when the step length is 0.02, and the total running time by which is far lower than the others. On the ArcGIS and WDMI datasets, the segmentation results have great similarity by using different step lengths. There is a small difference in total running time on the ArcGIS and WDMI datasets and the magnitude difference on the Google Earth dataset. Thus, good segmentation results are obtained when the increasing step length is $\kappa = 0.02$ by considering the balance of performance and running time.

E. Quantitative Analysis

To examine the robustness of the MSMSG segmentation method proposed by this article, the effectiveness of the method is verified by a quantitative dataset collected from the Google Earth. The dataset consists of ten ocean remote sensing images due to the heavy cost to collect, all of which have different sizes from 1880×1168 to 2996×2666. The average performance values of the dataset are applied to reflect the segmentation efficiency, and the corresponding quantitative results are reported in Fig. 19. It is obvious that the OE is the lowest with the proposed MSMSG-based segmentation method. Good boundary adherence reveals that higher segmentation accuracy can be obtained with the MSMSG method, which is because the BR and BP values are all in the second position among all methods. Slight differences occur in the UE with different methods.

In Fig. 20, we summarize the average running time of different segmentation methods that perform on the quantitative dataset. The quite low running time carrying out the MSMSG method indicates that the speed of the proposed segmentation method has an apparent advantage on the ocean remote sensing dataset. This demonstrates the superiority of the idea of MSMSG newly investigated in the segmentation task of ocean remote sensing images.



(a) (b)

(c) (d)

Fig. 19. Performance of the segmentation results by using different methods on the quantitative dataset. (a) BR. (b) BP. (c) UE. (d) OE.

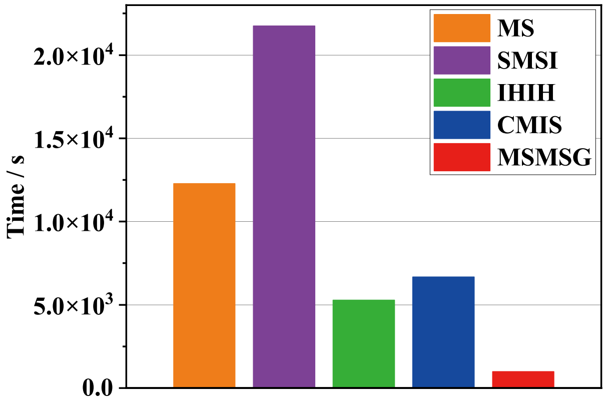


Fig. 20. Running time of different methods on the quantitative dataset.

VI. Conclusion

In this article, a novel MSMSG-based image segmentation method is proposed for large-size ocean remote sensing image segmentation, which contains coarse-to-fine superpixel generation, background removal, MSG construction, and MSG merging methods. The proposed coarse-to-fine superpixel generation method is the combination of a multiscale SLIC method and the Otsu method, by which both large objects and small objects are segmented adaptively. With the joint of GMM and a two-step MM filtering operation, ocean remote sensing images are efficiently parted into the foreground and background clusters not only by eliminating noise but also by retaining objects adequately. To the best of our knowledge, the MSG construction strategy is the first designed for ocean remote sensing image segmentation, which is guided by the background removal method. Moreover, an MSMSG-based superpixel merging manner is presented to complete the large-size ocean image segmentation automatically. Experimental results have shown the outstanding performance of the proposed segmentation method on three ocean remote sensing datasets.

However, the whole method is still time-consuming for large-size ocean remote sensing images, especially in MSG merging steps. Parallel methods will help to speed up the merging operation, which will be the focus of our future research.

Appendix

A. Proof of Theorem1

Proof: One Case:

An isolated point is a connected component and the neighbor set of which is empty, so the theorem is obviously established.

Other Case:

Sufficiency:

The nature of each edge in the graph $G(V, E^1)$ is to connect a pair of adjacent neighbors. If an connected component $G_x^1 = (V_x, E_x^1)$ includes all edges connects all adjacent vertexes, which is the nature of E^1. It means that the edge set E^1 is just the union set of all edge subsets E_x^1 if any connected component $G_x^1 = (V_x, E_x^1)$ includes all neighbors of each vertex $v_{x,i} \in V_x$, which is the sum of vertex set V.

Because all subgraphs are connected components, the intersection of the vertex set and the edge set between two different connected components is empty. The connected component set $\{G_1^1 = (V_1, E_1^1), \ldots, G_z^1(V_z, E_z^1)\}$ is an MSG.

Necessity: Suppose that there is a neighbor vertex $v_{y,j} \in V$ of vertex $v_{x,i}$ belonging to the vertex set of connected component $G_y^1 = (V_y, E_y^1)$, which is different from $G_x^1 = (V_x, E_x^1)$. It means that there is not one path between the pair of vertices $v_{x,i}$ and $v_{y,j}$. Thus, the edge $e(v_{x,i}, v_{y,j})$ that connects the two vertices does not belong to E_x^1 nor does it exist in E_y^1. Hence, the connected component set $\{G_1^1, \ldots, G_x^1, \ldots, G_y^1, \ldots, G_z^1\}$ is not an MSG by the reason of $E_1^1 \cup \ldots \cup E_x^1 \cup \ldots \cup E_y^1 \cup \ldots \cup E_z^1 \neq E^1$. That contradicts the condition of necessity, so necessity holds. ■

B. Proof of Theorem2

Proof: Each subgraph $\widetilde{G}_x = (V_x, E_x^1, W_x^1)$ runs the SSGM method by repeating t_x times, the time complexity of which is $O(t_x|V_x|\log(|V_x|))$, where $x \in [1, z]$. The summary time complexity of performing the multiscale SSGM method in all subgraphs is given as follows:

$$\begin{aligned} &O(t_1|V_1|\log(|V_1|) + \cdots + t_z|V_z|\log(|V_z|)) \\ &\quad < O\left(t_1|V_1|^2 + \cdots + t_z|V_z|^2\right) \\ &\quad < O\left(t_{\max}\left(|V_1|^2 + \cdots + |V_z|^2\right)\right) \\ &\quad < O\left(t_{\max}(|V_1| + \cdots + |V_z|)^2\right) \\ &\quad \leq O\left(t_{\max}(m+1)^2\right) \end{aligned} \tag{15}$$

where m is the total number of the vertex set V, i.e., $|V_1| + \cdots + |V_z| = m + 1$. $t_{\max}$ is the maximum time of all running times $\{t_1, \ldots, t_z\}$. There is at least one isolated point in all subgraphs, so the sum time is smaller than $O(t_{\max}m^2)$. ■

Acknowledgment

The authors would like to thank the editors and anonymous reviewers for their valuable comments and suggestions.

References

[1] N. Wang, B. Li, X. Wei, Y. Wang, and H. Yan, "Ship detection in spaceborne infrared image based on lightweight CNN and multisource feature cascade decision," *IEEE Trans. Geosci. Remote Sens.*, vol. 99, pp. 1–16, 2020.

[2] Y. Zhang, L. Guo, Z. Wang, Y. Yu, X. Liu, and F. Xu, "Intelligent ship detection in remote sensing images based on multi-layer convolutional feature fusion," *Remote Sens.*, vol. 12, no. 20, p. 3316, Oct. 2020.

[3] M. Modava, G. Akbarizadeh, and M. Soroosh, "Integration of spectral histogram and level set for coastline detection in SAR images," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 2, pp. 810–819, Apr. 2019.

[4] L. Gemme and S. G. Dellepiane, "An automatic data-driven method for SAR image segmentation in sea surface analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 1–14, May 2018.

[5] M. Wang, Z. Dong, Y. Cheng, and D. Li, "Optimal segmentation of high-resolution remote sensing image by combining superpixels with the minimum spanning tree," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 1, pp. 1–11, Sep. 2017.

[6] V. Mezaris, I. Kompatsiaris, and M. G. Strintzis, "Still image segmentation tools for object-based multimedia applications," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 18, no. 4, pp. 701–725, Jun. 2004.

[7] X. Wang, G. Li, A. Plaza, and Y. He, "Revisiting SLIC: Fast superpixel segmentation of marine SAR images using density features," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5221818.

[8] M. D. Hossain and D. Chen, "Segmentation for object-based image analysis (OBIA): A review of algorithms and challenges from remote sensing perspective," *ISPRS J. Photogram. Remote Sens.*, vol. 150, pp. 115–134, Apr. 2019.

[9] G. Chen, Q. Weng, G. J. Hay, and Y. He, "Geographic object-based image analysis (GEOBIA): Emerging trends and future opportunities," *GISci. Remote Sens.*, vol. 55, no. 2, pp. 159–182, Jan. 2018.

[10] Z. Fu, Y. Sun, L. Fan, and Y. Han, "Multiscale and multifeature segmentation of high-spatial resolution remote sensing images using superpixels with mutual optimal strategy," *Remote Sens.*, vol. 10, no. 8, p. 1289, Aug. 2018.

[11] M. A. Merchant, "Classification of open water features using OBIA and deep learning," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2021, pp. 104–107.

[12] Z. Li, E. Li, A. Samat, T. Xu, W. Liu, and Y. Zhu, "An object-oriented CNN model based on improved superpixel segmentation for high-resolution remote sensing image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 4782–4796, 2022.

[13] G. Liu and Z. Chai, "Image semantic segmentation based on improved DeepLabv3+ network and superpixel edge optimization," *J. Electron. Imag.*, vol. 31, no. 1, Jan. 2022, Art. no. 013011.

[14] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with atrous separable convolution for semantic image segmentation," in *Computer Vision—ECCV*, vol. 11211, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham, Switzerland: Springer, 2018, pp. 833–851.

[15] Y. Li, R. Chen, Y. Zhang, M. Zhang, and L. Chen, "Multi-label remote sensing image scene classification by combining a convolutional neural network and a graph neural network," *Remote Sens.*, vol. 12, no. 23, p. 4003, Dec. 2020.

[16] "Analysis of PCA algorithms in distributed environments," pp. 1–18, Apr. 2015, doi: 10.48550/arXiv.1503.05214.

[17] X. Pan, C. Zhang, J. Xu, and J. Zhao, "Simplified object-based deep neural network for very high resolution remote sensing image classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 181, pp. 218–237, Nov. 2021.

[18] Q. Cui, H. Pan, X. Li, K. Zhang, and W. Chen, “CMSuG: Competitive mechanism-based superpixel generation method for image segmentation,” *J. Intell. Fuzzy Syst.*, vol. 43, no. 4, pp. 4409–4430, Aug. 2022.

[19] J. Li, Q. Hu, and M. Ai, “Unsupervised road extraction via a Gaussian mixture model with object-based features,” *Int. J. Remote Sens.*, vol. 39, no. 8, pp. 2421–2440, Apr. 2018.

[20] X. Zhang, P. Xiao, X. Song, and J. She, “Boundary-constrained multiscale segmentation method for remote sensing images,” *ISPRS J. Photogramm. Remote Sens.*, vol. 78, pp. 15–25, Apr. 2013.

[21] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[22] R. Alshehhi and P. R. Marpu, “Hierarchical graph-based segmentation for extracting road networks from high-resolution satellite images,” *ISPRS J. Photogramm. Remote Sens.*, vol. 126, pp. 245–260, Apr. 2017.

[23] Z. Zhao, B. Li, X. Kang, L. Chen, X. Wei, and M. Xin, “Hybrid image segmentation method based on anisotropic Gaussian kernels and adjacent graph region merging,” *Rev. Sci. Instrum.*, vol. 91, no. 1, Jan. 2020, Art. no. 015104.

[24] H. Gu et al., “An efficient parallel multi-scale segmentation method for remote sensing imagery,” *Remote Sens.*, vol. 10, no. 4, p. 590, Apr. 2018.

[25] Y. Zhong, R. Gao, and L. Zhang, “Multiscale and multifeature normalized cut segmentation for high spatial resolution remote sensing imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6061–6075, Oct. 2016.

[26] X. Zhang, X. Jiang, J. Jiang, Y. Zhang, X. Liu, and Z. Cai, “Spectral–spatial and superpixelwise PCA for unsupervised feature extraction of hyperspectral imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022.

[27] O. Csillik, “Fast segmentation and classification of very high resolution remote sensing data using SLIC superpixels,” *Remote Sens.*, vol. 9, no. 3, p. 243, Mar. 2017.

[28] T. Kavzoglu and H. Tonbul, “An experimental comparison of multiresolution segmentation, SLIC and K-means clustering for object-based classification of VHR imagery,” *Int. J. Remote Sens.*, vol. 39, no. 18, pp. 1–17, 2018.

[29] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient graph-based image segmentation,” *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

[30] J. Shi and J. M. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[31] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, “Entropy rate superpixel segmentation,” in *Proc. CVPR*, Jun. 2011, pp. 20–25.

[32] C. Y. Hsu and J. J. Ding, “Efficient image segmentation algorithm using slic superpixels and boundary-focused region merging,” in *Proc. 9th Int. Conf. Inf., Commun. Signal Process.*, 2014, pp. 1–5.

[33] S. Di, M. Liao, Y. Zhao, Y. Li, and Y. Zeng, “Image superpixel segmentation based on hierarchical multi-level LI-SLIC,” *Opt. Laser Technol.*, vol. 135, Mar. 2021, Art. no. 106703.

[34] G. Zhang, X. Jia, and J. Hu, “Superpixel-based graphical model for remote sensing image mapping,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 11, pp. 5861–5871, Nov. 2015.

[35] Y. Wang, Q. Qi, L. Jiang, and Y. Liu, “Hybrid remote sensing image segmentation considering intrasegment homogeneity and intersegment heterogeneity,” *IEEE Geosci. Remote Sens. Lett.*, vol. PP, no. 99, pp. 1–5, May 2019.

[36] Z. Li and J. Chen, “Superpixel segmentation using linear spectral clustering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1356–1363.

[37] L. Zhu, J. Zhang, and Y. Sun, “Remote sensing image change detection using superpixel cosegmentation,” *Information*, vol. 12, no. 2, p. 94, Feb. 2021.

[38] R. Achanta and S. Susstrunk, “Superpixels and polygons using simple non-iterative clustering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4600–4651.

[39] C. Wu et al., “Fuzzy SLIC: Fuzzy simple linear iterative clustering,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 6, pp. 2114–2124, Jun. 2021.

[40] W. Cui et al., “Multi-scale semantic segmentation and spatial relationship recognition of remote sensing images based on an attention model,” *Remote Sens.*, vol. 11, no. 9, p. 1044, May 2019.

[41] A. Farooq, X. Jia, J. Hu, and J. Zhou, “Multi-resolution weed classification via convolutional neural network and superpixel based local binary pattern using remote sensing images,” *Remote Sens.*, vol. 11, no. 14, p. 1692, Jul. 2019.

[42] N. Otsu, “Threshold selection method from gray-level histograms, ieee transactions on systems man and cybernetics,” *IEEE Trans. Syst. Man Cybern.-Syst.*, vol. SMCS-9, no. 1, pp. 62–66, Jan. 1979.

[43] B. Johnson and Z. Xie, “Unsupervised image segmentation evaluation and refinement using a multi-scale approach,” *ISPRS J. Photogram. Remote Sens.*, vol. 66, no. 4, pp. 473–483, Jul. 2011.

[44] T. Su, “Scale-variable region-merging for high resolution remote sensing image segmentation,” *ISPRS J. Photogramm. Remote Sens.*, vol. 147, pp. 319–334, Jan. 2019.

[45] Y. Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images,” in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, 2001.

[46] C. Rother, V. Kolmogorov, and A. Blake, “‘Grabcut’—Interactive foreground extraction using iterated graph cuts,” *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[47] E. S. L. Gastal and M. M. Oliveira, “Shared sampling for real-time alpha matting,” *Comput. Graph. Forum*, vol. 29, no. 2, pp. 575–584, May 2010.

[48] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski, “A Bayesian approach to digital matting,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2003, pp. 1–8.

[49] A. Levin, D. Lischinski, and Y. Weiss, “A closed-form solution to natural image matting,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 228–242, Feb. 2008.

[50] Y. Zheng and C. Kambhamettu, “Learning based digital matting,” in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 889–896.

[51] C. Hou, L. Can, and D. Chen, “Research on sound source recognition algorithm of pickup array based on adaptive background noise removal,” in *Proc. Int. Conf. Artif. Intell. Commun. Netw.*, Oct. 2021, pp. 346–361.

[52] A. Xia, Y. Gui, L. Yao, L. Ma, and X. Lin, “Exemplar-based object removal in video using GMM,” in *Proc. Int. Conf. Multimedia Signal Process.*, May 2011, pp. 366–370.

[53] P. Soille, *Morphological Image Analysis-Principles and Applications*. New York, NY, USA: Springer-Verlag, 2003.

[54] P. Soille, “Morphological image compositing,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 673–683, May 2006.

[55] M. Dalla Mura, J. A. Benediktsson, and L. Bruzzone, “Classification of hyperspectral images with extended attribute profiles and feature extraction techniques,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2010, pp. 76–79.

[56] I. T. Jolliffe, “Principal component analysis,” *J. Marketing Res.*, vol. 87, p. 513, Aug. 2002.

[57] A. Humeau-Heurtier, “Texture feature extraction methods: A survey,” *IEEE Access*, vol. 7, pp. 8975–9000, 2019.

[58] G. Chen, G. J. Hay, G. Castilla, B. St-Onge, and R. Powers, “A multiscale geographic object-based image analysis to estimate LiDAR-measured forest canopy height using quickbird imagery,” *Int. J. Geographical Inf. Sci.*, vol. 25, no. 6, pp. 877–893, Jun. 2011.

[59] M. K. Garajeh, B. Feizizadeh, T. Blaschke, and T. Lakes, “Detecting and mapping Karst landforms using object-based image analysis: Case study: Takht–Soleiman and Parava mountains, Iran,” *Egyptian J. Remote Sens. Space Sci.*, vol. 25, no. 2, pp. 473–489, Aug. 2022.

[60] X. Zhang, P. Xiao, X. Feng, and G. He, “Another look on region merging procedure from seed region shift for high-resolution remote sensing image segmentation,” *ISPRS J. Photogramm. Remote Sens.*, vol. 148, pp. 197–207, Feb. 2019.

[61] L. Drăguţ, O. Csillik, C. Eisank, and D. Tiede, “Automated parameterisation for multi-scale image segmentation on multiple layers,” *ISPRS J. Photogramm. Remote Sens.*, vol. 88, pp. 119–127, Feb. 2014.

[62] M. S. Farid, M. Lucenteforte, and M. Grangetto, “DOST: A distributed object segmentation tool,” *Multimedia Tools Appl.*, vol. 77, no. 16, pp. 20839–20862, Aug. 2018.

[63] P. Ghamisi, M. Dalla Mura, and J. A. Benediktsson, “A survey on spectral–spatial classification techniques based on attribute profiles,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2335–2353, May 2015.

[64] D. B. West, *Introduction To Graph Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.

[65] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[66] T. Elgamal and M. Hefeeda, "Analysis of PCA algorithms in distributed environments," *Comput. Sci.*, 2015.

[67] *Dataset*. Accessed: Oct. 27, 2022. [Online]. Available: http://www.tuxingis.com

[68] *Google Earth Dataset*. Accessed: Aug. 25, 2021. [Online]. Available: https://earth.google.com

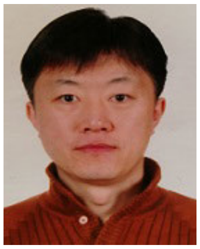
[69] Y. Wang, Q. Qi, Y. Liu, L. Jiang, and J. Wang, "Unsupervised segmentation parameter selection using the local spatial statistics for remote sensing image segmentation," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 81, pp. 98–109, Sep. 2019.

[70] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[71] B. Sathiyaprasad and B. S. Kumar, "Multi spectral image retrieval in remote sensing big data using fast recurrent convolutional neural network," in *Proc. Int. Conf. Advancement Technol. (ICONAT)*, Jan. 2022, pp. 1–7.

Qianna Cui received the B.S. degree in mathematics and statistics from Anyang Normal University, Anyang, China, in 2015, and the M.S. degree in mathematics and statistics from Yunnan University, Kunming, China, in 2018. She is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Harbin Engineering University, Harbin, China.

Her research interests include remote sensing image processing, route recommendation systems, and machine learning.



Haiwei Pan (Member, IEEE) received the Ph.D. degree from the College of Computer and Science, Harbin Institute of Technology, Harbin, China, in 2006.

From 2007 to 2009, he was a Post-Doctoral Research Fellow with the School of Computer Science and Technology, Harbin Engineering University, Harbin, where he is currently a Professor. His research interests include remote sensing image processing, medical image processing, data mining, machine learning, and deep learning.

Kejia Zhang received the B.S. degree in specialty of information management and information systems from Beijing Normal University, Beijing, China, in 2004, and the M.E. and Ph.D. degrees from the College of Computer and Science, Harbin Institute of Technology, Harbin, China, in 2007 and 2012, respectively.

He is currently an Associate Professor with the School of Computer Science and Technology, Harbin Engineering University, Harbin. His research interests include remote sensing image processing, medical image processing, recommendation system research, machine learning, and deep learning.

Xiaokun Li received the M.S. degree from School of Computer Science and Technology, Northeastern University, Shenyang, China, in 2011.

He is currently a Professor with the School of Computer Science and Technology, Heilongjiang University. His research interests include image processing, artificial intelligence, machine learning, social networks, and deep learning.

Haoyu Sun received the B.S. degree from the School of Electrical and Information, Northeast Agricultural University, Harbin, China, in 2021. He is currently pursuing the M.E degree with the School of Computer Science and Technology, Harbin Engineering University, Harbin.

His research interests include remote sensing image processing, medical image processing, recommendation system research, machine learning, and deep learning.