Real-Time Obstacle Detection Based on Monocular Vision for Unmanned Surface Vehicles

Zhang Rui, Liu Jingyi, Li Hengyu, and Cheng Qixing(✉)

Shanghai University, Baoshan, Shanghai 200444, China
xing_shu@shu.edu.cn

Abstract. The reliable obstacle detection is a challenging task in autonomous navigation of unmanned surface vehicles. In this paper, we present a novel real-time obstacles detection based on monocular vision which can effectively tell apart obstacles on the sea surface from complex background. The main innovation of this paper is to propose a water-boundary-line algorithm based on semantic segmentation and random sample consistency line fitting. And use a simple and effective saliency detection method based on background prior and foreground prior to detect obstacles under the water-boundary-line. Our method can efficiently and quickly obtain obstacle information from images captured by shipborne cameras, and it has the ability to process more than 33 frames/s.

Keywords: Obstacle detection · Unmanned surface vehicle · Computer vision

1 Introduction

The unmanned surface vehicle (USV) is a kind of unmanned surface platform which can sail autonomously in the marine environment [16]. It can carry different task modules according to the needs of tasks, and better deal with the challenges brought by different tasks. The USV can not only be used in environmental monitoring, sea chart mapping, water quality sampling and other aspects of migrant workers, but also can be used in maritime reconnaissance, armed patrol, anti submarine and other military tasks [17]. Compared with manned water vehicle, it has the characteristics of small size, intelligence, unmanned and low cost.

Obstacle detection technology, as a key part of the environmental perception of the unmanned surface vehicle, is a prerequisite for its autonomous obstacle avoidance and safe navigation. At present, the obstacle detection technology of the unmanned surface vehicle mainly includes the target detection technology based on radar, the target detection technology based on underwater sound and the target detection technology based on vision. Compared with the former two

Y. Chen et al. (Eds.): BICT 2020, LNICST 329, pp. 166–180, 2020.
https://doi.org/10.1007/978-3-030-57115-3_14

technologies, the obstacle detection technology based on vision can obtain more abundant target area information, easier detection and recognition of sea targets, and can obtain information about plane objects, such as fragments, floating rods and so on, which are difficult to obtain by other detection methods. In this paper, we proposed a suitable monocular vision-based approach that is able to automatically detect above-surface obstacles for USVs and runs at over 33 frames/s on GPU.

2 Related Work

With the increase in demand for collision avoidance, various approaches have been proposed to solve the problems of obstacle detection in unmanned surface vehicles. A common practice of these approaches is the use of sonar, radar, or LIDAR. Heidarsson and Sukhatme [8] proposed a method that estimates a map of static obstacles by combining data from an on-board sonar with the overhead images of the operating area from Google maps and Bing maps. However, it is difficult for this method to detect dynamic obstacle that do not appear in the satellite map. Gal and Zeitouni [6] conducted an initial research that utilizes a LIDAR to identify and track object for USVs, but this approach is sensitive to USV motion and cannot detect flat objects. Almeida et al. [2] used an on-board radar to detect potential obstacles in maritime environments. However, they encountered difficulties in detecting small obstacles at close distances in practical tests. In addition, the accuracy of this approach is susceptible to high waves and water reflectivity. Therefore, several approaches have focused on obstacle detection for USVs using cameras.

Gal [5] presented an Automatic Target Detection(ATD) algorithm for USVs motion planning and automatic decision models using visual sensors. In order to reduce the search space, he combined the Canny Edge Detector algorithm and Hough Line Transformation to locate the horizon line. After that, a co-occurrence matrix was applied to learn the sea pattern and detect the potential obstacles in the region below the horizon. However, the horizon detection described in [5] is quite sensitive to the strong edges of land, waves, clouds, etc. It may lead to a poor result in the next step of obstacle detection. In addition, the sea pattern learned by co-occurrence matrix is susceptible to sea clutter so that the probability of false detection will increase significantly in practice. Guo et al. [7] attempted to use a panoramic camera to detect obstacles in water. They first identified the coarse regions of moving objects based on the image difference between two consecutive images, and then used a colorimetric criterion to segment objects from these coarse regions. A major problem of this method is that it does not consider the effects of cloud and sea movement. This leads to spurious difference between consecutive images, resulting in incorrect foreground extraction. Li et al. [10] applied fusion of objectness and saliency to detect obstacles around the USV. They first used the objectness to get the object proposals, and then applied the salient density to remove the proposals with false alarms. However, when there are birds, airplanes or other possible flying objects in the sky, this approach will mistake them as obstacles.

Wang et al. [13–15] proposed an obstacle detection system based on monocular and stereo vision methods for USVs. Similar to the approach of [5], they also applied horizon line to reduce the search space of potential obstacles. They used Sobel operator to calculate vertical gradient and RANSAC method to fit horizon line according to the maximum or minimum vertical gradient points. Following the horizon line detection, they applied saliency detection in left camera to obtain a rough estimation of obstacles and motion evaluation to refine the result of saliency detection. To find the stereo correspondence in right camera, epipolar constraint of stereo camera system and template matching were implemented. However, their assumption of sharp boundary between sea and sky is often violated in practice. In addition, it is difficult to handle coastline detection. Wang and Wei [12] developed an obstacle detection system based on stereo vision. They applied a block-matching based dense stereo correspondence method to reconstruct the scene in 3D. By using RANSAC fitting method, they estimated the sea surface from the 3D point set of the reconstructed scene. After aligning the reconstructed 3D points with the sea surface, they performed obstacle detection based on occupancy grid and height grid. However, when the sea is calm, this approach may encounter poor 3D reconstruction of the sea due to the lack of texture, resulting in inaccurate sea surface estimation. Moreover, such approach requires objects to significantly stand out from water in order to be distinguished as obstacles. Thus, it is difficult to effectively detect small buoys and floating debris. Recently, Mou and Wang [11] proposed a wide-baseline stereo-based static obstacle mapping approach, in which only one camera was used, and the depth was restored via the motion of the USV. They reconstructed the world locations of the detected static obstacles by integrating monocular camera with GPS and compass information, and then used multiple pairs of frames to synthesize the final reconstruction results in a weighting model. Compared to the methods described in [12–15], this approach can not only eliminate the complicated calibration work and the bulky rig, but also improve the ranging ability. But such approach can only reconstruct static obstacles, and the USV should be in a state of travel. Furthermore, measurement noise from the GPS, the compass, horizon detection, and especially feature matching might influence the accuracy of this approach.

Kristan et al. [9] presented a probabilistic graphical model for monocular obstacle detection via semantic segmentation of the observed marine scene. This model assumes that an image captured from the USV can be roughly split into three approximately parallel semantic regions in the vertical direction: sky at the top, haze or land in the middle, and water at the bottom of the image. More specifically, the graphical model assumes a mixture model with three Gaussian components for the three dominant semantic regions and a uniform component for explaining the outliers. A Markov random field framework is adopted to enforce smooth segmentation. This state-of-the-art algorithm generates a water segmentation mask, and treats all outliers in the water region as obstacles. It significantly outperforms the related approaches on the task of marine obstacle detection. Nevertheless, such approach cannot effectively detect obstacles

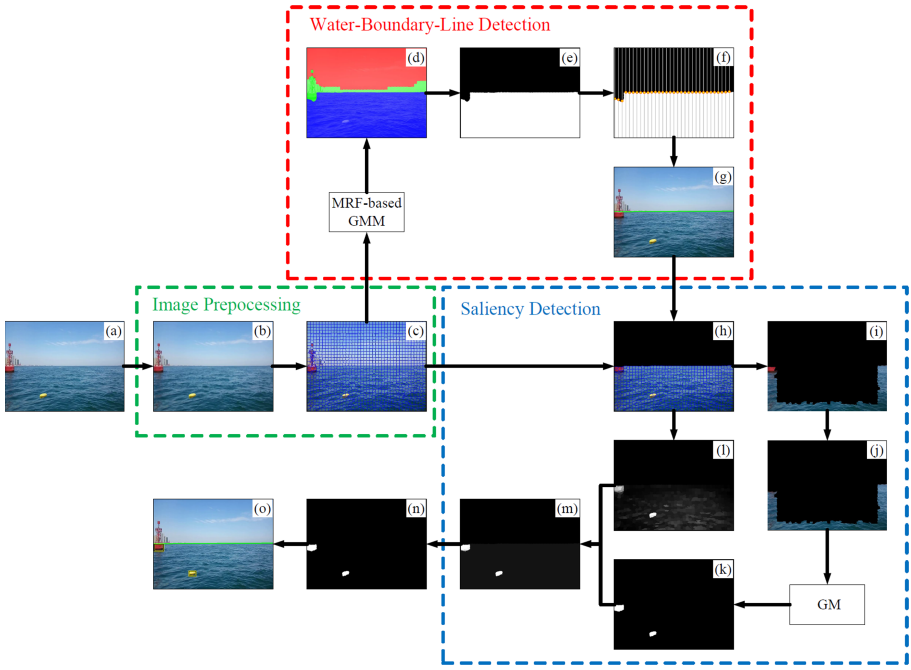

Fig. 1. Algorithm flow chart, mainly including Image Preprocessing, Water-Boundary-Line Detection and Saliency Detection. "MRF-based GMM" is "Gaussian mixture model based on Markov random field". And "GM" is "Gaussian distribution".

that are adjacent to the sea-sky-line/coastline or partially under the sea-sky-line/coastline, because these obstacle regions usually be classified as the middle semantic region by the graphical model. In addition, in order to reduce the running time, this approach performs detection on a reduced-size image of 50×50 pixels and then rescales the results to the original image size, which may result in inaccurate contours of the detected obstacles. Bovcon et al. [3] proposed an improving probabilistic graphical model for obstacle detection in USVs. This model extends the semantic segmentation model from [9] by incorporating the boat tilt measurements from the on-board IMU to improve the performance in the presence of visual ambiguities. Recently, Bovcon et al. [4] presented a new stereo-based obstacle detector for USVs by IMU-assisted semantic segmentation. They build upon [3], and improved the performance of the segmentation-based obstacle detection via stereo verification.

3 Our Approach

In this paper, we introduce a real-time method to detect obstacles for unmanned surface vehicles using a monocular camera. Firstly, the input image is blurred by a two dimensional median filter to eliminate isolated noise and then over

segmented into super pixels by simple linear iterative clustering (SLIC) algorithm as the elementary processing units instead of pixels in the follow-up process. Secondly, a novel water-boundary-line detection based on semantic segmentation and random sample consensus (RANSAC) line fitting algorithm is exploited to reduce the search space of potential obstacles. Thirdly, a simply and efficient saliency detection approach based on background prior and contrast prior is proposed to detect obstacles under the water boundary line. Our approach is outlined in Fig. 1.

Our main contribution in this work is to develop a water-boundary-line detection approach that can be used not only for sea-sky-line detection but also for coastline detection. Similar to the method of [9], this approach adopts a Gaussian mixture model based on Markov random field (MRF) to split the input image into three approximately parallel semantic regions: sky at the top, haze or land in the middle, and sea at the bottom of the image. By using RANSAC line fitting algorithm, the parameters of the water-boundary-line can be estimated from a set of boundary points between the middle semantic region and the bottom semantic region. As a secondary contribution, we propose a saliency detection approach based on boundary prior and contrast prior to detect obstacles for USVs. The proposed approach performs saliency detection under the water-boundary-line and chooses non-saliency boundary super pixels as boundary priors to construct the background probability model. The background-based saliency map can be obtained through calculating the probability that each superpixel under the water-boundary-line belongs to the background mod-el. By using the region contrast against its surroundings as a saliency cue, the foreground-based saliency map is computed as the summation of its appearance distance to all other superpixels under the water-boundary-line, weighted by their spatial distance. Finally, we can get a unified saliency map by combining the two saliency maps using a unified formula. Moreover, our saliency detection method can handle pure background images well.

3.1 Image Preprocessing

In the image preprocessing step, the main operation is to use the simple linear iterative clustering (SLIC) algorithm proposed by Achanta et al. [1] in 2012. First, we scale the input picture as shown in Fig. 1(a) to 640×480, and then filter to remove the isolated noise to obtain the picture as shown in Fig. 1(b). Then, the SLIC algorithm is used to perform superpixel segmentation on the picture obtained in the previous step. The resulting superpixels are used as the input and basis for the subsequent processing steps.

During the execution of the SLIC algorithm, the input image is also converted to the *Lab* color space, and the feature vector of each pixel is expressed as $[L,\ a,\ b,\ x,\ y]$ (L, a and b are the color components of Lab color space, and x and y represent the column and row coordinates of image respectively). In the process of image preprocessing, the selected filter may be a two-dimensional median filter or a bilateral filter when filtering the scaled image. Through many experiments, we find that when the number of superpixels is set to 500 or more,

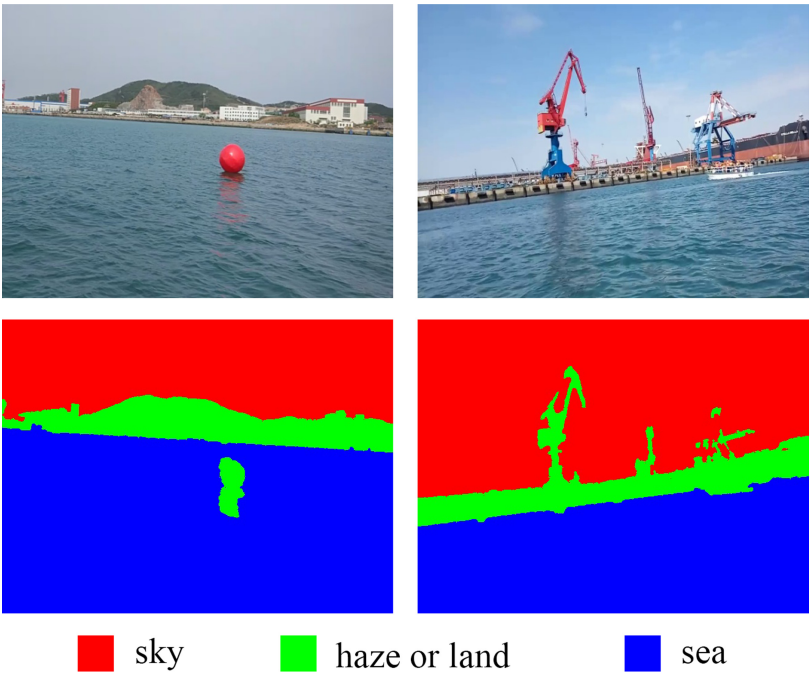

Fig. 2. Example of scaling to 640×480 images and their semantic segmentation masks generated by the program. Red, green, and blue correspond to the sky at the top, haze or land in the middle, and the sea at the bottom. (Color figure online)

SLIC algorithm can more accurately segment the small objects in the image captured by the USV. Finally, the number of superpixels is set to 1200 in our algorithm.

3.2 Water-Boundary-Line Detection

It can be seen from Fig. 1 that the processing object of the Water-Boundary-Line detection is the superpixel obtained by the image preprocessing step. Finally a water-boundary-line is obtained to distinguish the seaw and non-sea parts. This step can be divided into two parts: semantic segmentation from Fig. 1(c) to Fig. 1(d) and fitting water-boundary-line from Fig. 1(d) to Fig. 1(g).

Semantic Segmentation. Similar to the semantic segmentation results in [9], it is assumed that the input image can be divided into three approximately parallel regions: sky at the top, haze or land in the middle, and sea at the bottom, as shown in Fig. 2. The probability of the i-th superpixel feature vector

is modelled as a mixture model with three Gaussians component:

$$p\left(y_i|\Theta,\pi\right)=\sum_{k=1}^{3}N\left(y_i|m_k,C_k\right)\pi_{ik} \tag{1}$$

Where $\Theta=\{m_k,C_k\}_{k=1:3}$ are the means and covariances of the Gaussian kernels $N\left(*|m_k,C_k\right)$. Different values of k indicate different semantic segmentation categories. y_i is observable data which refers to the feature vector of the i-th superpixel. π is the class prior distribution of all superpixels, and $\pi=\{\pi_i\}_{i=1:M}$(for π_i, π_{ik} is the prior probability that the i-th superpixel belong to class k, and its expression is π_{ik} where x_i is the class of i-th superpixel. M is the number of superpixels).

Assuming the set $\pi=\{\pi_i\}_{i=1:M}$ is a Markov random field, the joint probability density function of π can be expressed as:

$$p\left(\pi\right)\approx\prod_{i=1}^{M}p\left(\pi_i|\pi_{N_i}\right) \tag{2}$$

$$p\left(\pi_i|\pi_{N_i}\right)\propto\exp\left[-\frac{1}{2}\left(D\left(\pi_i||\pi_{N_i}\right)+H\left(\pi_i\right)\right)\right] \tag{3}$$

N_i is the set of neighborhood superpixels of the i-th superpixel. $H\left(\pi_i\right)$ is information entropy (where $H\left(\pi_i\right)=-\sum_{k=1}^{4}\pi_{ik}\log\pi_{ik}$). $D\left(\pi_i||\pi_{N_i}\right)$ is relative entropy (where $D\left(\pi_i||\pi_{N_i}\right)=\sum_{k=1}^{4}\pi_{ik}\log\pi_{ik}-\sum_{k=1}^{4}\pi_{ik}\log\pi_{N_ik}$). π_{N_i} is the class prior distribution of the superpixel set N_i in the neighborhood of superpixel i, which can be expressed as follows:

$$\pi_{N_i}=\sum_{j\in N_i,j\neq i}\lambda_{ij}\pi_j \tag{4}$$

In formula (4), λ_{ij} is the influence coefficient of the adjacent superpixel j on the center superpixel i, and $\lambda_{ij}=1/n$ (where n is the number of adjacent superpixels).

Secondly, suppose that the class posterior distribution set $P=\{p_i\}_{i=1:M}$ of all superpixels is a Markov random field. The joint probability density function of P can be approximately as follows:

$$p\left(P|Y,\pi,\theta\right)\propto\prod_{i=1}^{M}\exp\left[-\frac{1}{2}\left(D\left(p_i||p_{N_i}\right)+H\left(p_i\right)\right)\right] \tag{5}$$

where the posterior probability p_{ik} in $P=\{p_i\}_{i=1:M}$ is calculated as follows:

$$p_{ik}=p\left(x_i=k|y_i,\pi_{ik},\theta\right)=\frac{p\left(y_i|x_i=k,\theta\right)\pi_{ik}}{\sum_{l=1}^{3}p\left(y_i|x_i=l,\theta\right)\pi_{il}} \tag{6}$$

Summarizing with the above formula, the joint probability density function of probability graph model can be expressed as:

$$\begin{aligned} p\left(P,Y,\pi|\theta\right) \propto \exp\sum_{i=1}^{M}\{&\log\sum\nolimits_{k} p\left(y_i|x_i=k,\theta\right)\pi_{ik} \\ &-\tfrac{1}{2}\left[D\left(\pi_i||\pi_{N_i}\right)+H\left(\pi_i\right)\right] \\ &-\tfrac{1}{2}\left[D\left(p_i||p_{N_i}\right)+H\left(p_i\right)\right]\} \end{aligned} \tag{7}$$

Due to the coupling relationship between $\pi_i||\pi_{N_i}$ and $p_i||p_{N_i}$, it is difficult to directly estimate the parameters of the model using formula (7). To solve this problem, the auxiliary probability distribution sets $s=\{s_i\}$ and $q=\{q_i\}$ should be introduced into the above formula. By taking the logarithms on both sides of the equation, the penalty log-likelihood function of the probability graph model can be obtained:

$$\begin{aligned} F=\sum_{i=1}^{M}\{&\log\sum\nolimits_{k} p\left(y_i|x_i=k,\theta\right)\pi_{ik} \\ &-\tfrac{1}{2}\left[D\left(s_i||\pi_i\right)+D\left(s_i||\pi_{N_i}\right)+H\left(s_i\right)\right] \\ &-\tfrac{1}{2}\left[D\left(q_i||p_i\right)+D\left(q_i||p_{N_i}\right)+H\left(q_i\right)\right]\} \end{aligned} \tag{8}$$

The auxiliary probability distribution sets $s=\{s_i\}_{i=1:M}$ and $q=\{q_i\}_{i=1:M}$ are calculated as follows:

$$s_i=\pi_i\circ\pi_{N_i} \tag{9}$$

$$q_i=p_i\circ p_{N_i} \tag{10}$$

Where $\circ$ is Hadamard product.

The expectation maximization algorithm is used in (8) to estimate the parameters of the model. Use the estimated parameters to calculate the semantic segmentation results of the image.

For the above penalized log likelihood function, the maximum expectation (EM) algorithm is used to estimate the model parameter θ, so as to achieve the semantic segmentation of the sea image. The formula for updating the relevant parameters in the mixed model is as follows:

$$m_k=\frac{\sum\limits_{i=1}^{M}\left(q_{ik}+q_{N_ik}\right)y_i}{\sum\limits_{i=1}^{M}\left(q_{ik}+q_{N_ik}\right)} \tag{11}$$

$$C_k=\frac{\sum\limits_{i=1}^{M}\left(q_{ik}+q_{N_ik}\right)y_iy_i^T}{\sum\limits_{i=1}^{M}\left(q_{ik}+q_{N_ik}\right)}-m_km_k^T \tag{12}$$

The formula for calculating the class prior distribution π_i is as follows:

$$\pi_i=\frac{1}{4}\left[\left(q_i+q_{N_i}\right)+\left(s_i+s_{N_i}\right)\right] \tag{13}$$

After obtaining the parameters of the mixed model, the probability that each superpixel belongs to three regions is calculated by formula (6). The class with the highest probability is taken as the class of the superpixel. After visualizing the segmentation results, the semantic segmentation mask map shown in Fig. 1(d) can be obtained.

Fitting Water-Boundary-Line. As shown in Fig. 1(d) to (e), firstly, the areas classified as sky and the areas classified as haze or land in the semantic segmentation mask are merged. Binarize the merged mask map. According to the gradient information in the column direction, a certain number of boundary points between the sea area and the upper area are obtained, as shown in Fig. 1(e) to (f). Using the Random Sampling Consensus (RANSAC) Algorithm to fit the previously obtained boundary points, the sea boundary line shown in Fig. 1(g) is obtained.

3.3 Saliency Detection

Zhu et al. [19] Proposed a significance optimization algorithm based on robust background detection, namely RBD algorithm in 2014. The algorithm first estimated the background probability of the superpixel through the boundary connectivity. And then calculates the foreground probability of the superpixel by using the global contrast method. Finally optimizes the background probability and the foreground probability of the superpixel significantly through an optimization framework. Finally, an optimization framework is used to fuse the background probability and foreground probability of superpixels to obtain a saliency map. Because the RBD algorithm is not suitable for parallel computing, we use a method that is generally similar to the RBD algorithm, but differs in specific details (mainly focusing on background estimation).

By processing the superpixels shown in Fig. 1(c) in combination with the water-boundary-line, a set of superpixels of the sea class shown in Fig. 1(h) can be obtained. Then the foreground and background probabilities are obtained through foreground estimation and background estimation, which are shown in Fig. 1(l) and Fig. 1(k) after visualization. Using the optimization framework, the foreground probability and the background probability are fused to obtain the saliency map shown in Fig. 1(m).

Background Estimation. For the RBD algorithm, first, the superpixels in the processed area that directly contact the image boundary need to be extracted. In our method, the boundary superpixel set used in the RBD algorithm is operated to remove possible obstacle superpixels.

First, all superpixels connected to the image boundary are defined as $Bnd = \{p_i | p_i \; is \; the \; boundary \; superpixel\}$, as shown in Fig. 1(i). Based on the difference between the obstacle target and the background area, the possible obstacle superpixels in the initial boundary set Bnd are eliminated. Suppose the initial

boundary set $Bnd = \{p_1, p_2, ..., p_n\}$, and define the feature difference value of superpixel p_i as:

$$d_i = \sum_{j=1,2,...,n, j \neq i} d_{app}\left(p_i, p_j\right) / \left(n-1\right) \tag{14}$$

Where $d_{app}\left(p_i, p_j\right)$ is the Euclidean distance of superpixel p_i and p_j in lab color space. The feature difference values of all superpixels in the initial boundary set Bnd can be calculated, expressed as $\{d_1, d_2, ..., d_n\}$. By calculating the mean μ and variance σ of the set $\{d_1, d_2, ..., d_n\}$, the superpixels satisfying the condition $|d_i - \mu| > 3\sigma$ are removed from the set Bnd. The resulting new superpixel set Bnd is visualized as shown in Fig. 1(j).

A new Gaussian distribution $p_i^{bg} = N\left(y_i|\mu, \sigma\right)$ is defined to distinguish all superpixels as obstacles or background. The initial mean μ and covariance σ of the model are calculated by the new boundary set Bnd. Using the model, the probability p_i^{bg} that each superpixel belongs to the background can be calculated. Visualize the background probability, as shown in Fig. 1(k).

Foreground Estimation. For foreground estimation, the method in this paper is to directly calculate the significance of the foreground (obstacle) through the method of regional comparison. The formula for calculating the contrast value of superpixel p_i is:

$$Ctr\left(p_i\right) = \sum_{j=1}^{N} d_{app}\left(p_i, p_j\right) \omega_{spa}\left(p_i, p_j\right) \tag{15}$$

$$\omega_{spa}\left(p_i, p_j\right) = \exp\left(-\frac{d_{spa}^2\left(p_i, p_j\right)}{2\sigma_{spa}^2}\right) \tag{16}$$

Where d_{app} is the spatial distance between the superpixels p_i and p_j. The value of parameter σ_{spa} is set to 0.25 in this paper.

In order to suppress the interference of background information, the background probability w_i^{bg} obtained in the background estimation step is introduced into formula (15) as a new weighting term, so that the background weighted contrast formula is as follows:

$$\omega_i^{fg} = \sum_{j=1}^{N} d_{app}\left(p_i, p_j\right) \omega_{spa}\left(p_i, p_j\right) \omega_j^{bg} \tag{17}$$

From (17), the probability that each superpixel belongs to the foreground can be calculated. The visualization of the foreground probability is shown in Fig. 1(l).

Optimization. In order to improve the accuracy and robustness of the algorithm, we use the optimization framework of RBD algorithm to integrate the foreground probability and background probability, so as to achieve the purpose of saliency optimization. The objective cost function of the optimization framework is defined as:

$$\sum_{i=1}^{N} \omega_i^{bg} s_i^2 + \sum_{i=1}^{N} \omega_i^{fg} \left(s_i - 1\right)^2 + \sum_{i,j} \omega_{i,j} \left(s_i - s_j\right)^2 \tag{18}$$

The optimization process is to minimize the objective function. In formula (18), N is the number of superpixels in the image. s_i and s_j represent the saliency values of superpixel i and j respectively and their value ranges are $[0, 1]$. $w_{i,j}$ represents the smooth weight between superpixel i and its adjacent superpixel j, and its calculation formula is:

$$w_{i,j} = \exp\left(-\frac{d_{app}^2\left(p_i, p_j\right)}{2\sigma_{clr}^2}\right) + \mu \tag{19}$$

where the value of parameter σ_{clr} is 10, and the value of parameter μ is 0.1

For the optimization framework, when the background probability ω_i^{bg} of the superpixel i is larger, the background optimization term of the cost function will cause its significant value s_i to become smaller. Similarly, when the foreground probability ω_i^{fg} of the superpixel i is larger, the foreground optimization term of the cost function will cause its significant value s to become larger. For smooth optimization, its role is to make the saliency values between superpixels relatively smooth.

After saliency optimization, the saliency value corresponding to each superpixel can be obtained, and its address range is $[0, 1]$. In order to perform threshold segmentation, it needs to be normalized to $[0, 255]$. Visualize the normalized saliency map as shown in Fig. 1(m). Afterwards, the saliency map is binary segmented by an appropriate fixed threshold. The choice of threshold directly affects the segmentation accuracy of the image. After many experiments, we set the threshold to 70. Figure 1(n) is the result of binary segmentation of Fig. 1(m) through the selected threshold.

4 Result

The test image set used in the experiment was collected by the shipboard camera of the unmanned surface boat. The test image set contains 8 groups of image sequences, each group has about 300 to 400 frames of images. The experimental algorithm is implemented in the C++ programming language and uses cuda acceleration technology. The experimental program runs on an ordinary desktop computer (CPU: Intel i9 9900k, GPU: NVIDIA 1080ti, 8 GB memory). For the algorithm described in this article, the experiment mainly evaluates it from two aspects: water-boundary-line detection and obstacle detection.

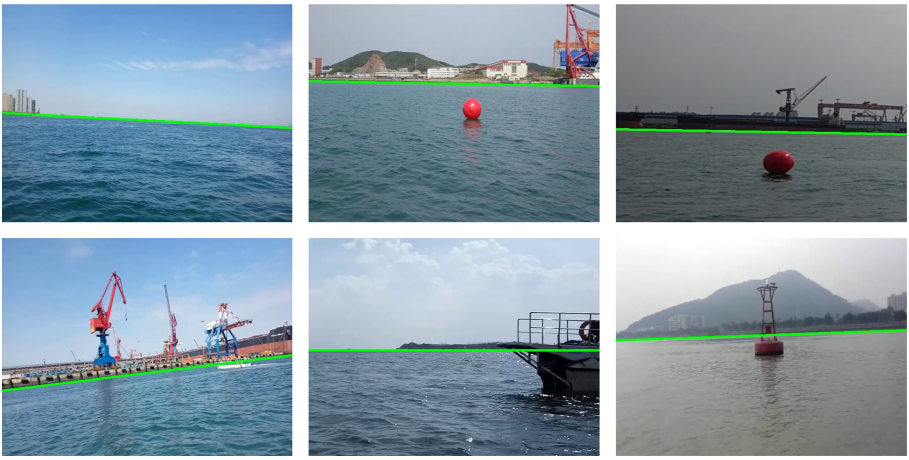

Fig. 3. Example of water-boundary-line detection.

4.1 Results of Water-Boundary-Line Detection

Figure 3 is the water-boundary-line detection result obtained by the algorithm of this paper. It can be seen from the above that when the image has disadvantages such as low visibility of the water-boundary-line, buoys and part of the land near the water-boundary-line, the algorithm can still effectively detect the position of the water-boundary-line.

In order to quantitatively evaluate the accuracy of the water-boundary-line detection algorithm, the method used in this paper first manually marks the water-boundary-line on the test image. The corresponding reference straight line function is established through manual marking. Then the obtained straight line function and reference straight line function calculate the corresponding detection error by the following formula:

$$\delta = \frac{1}{C}\sum_{x=1}^{C} |y(x) - y'(x)|, \ x = 1, 2, ..., C \tag{20}$$

Where x is the column coordinate of the image, and C is the number of image columns. $y(\cdot)$ is the linear equation of the water-boundary-line determined by the detection algorithm, and $y'(\cdot)$ is the linear function of the water-boundary-line obtained by manual labeling.

This paper also defines the criterion for the accurate detection of the water-boundary-line line: when the detection error calculated by (20) is less than 8 pixels, the detection result is considered valid, that is, the algorithm accurately detects the water-boundary-line. Through this criterion, the detection results of the test image set can be counted, and the corresponding accuracy rate can be calculated according to the ratio of the number of correct detections to the number of test images.

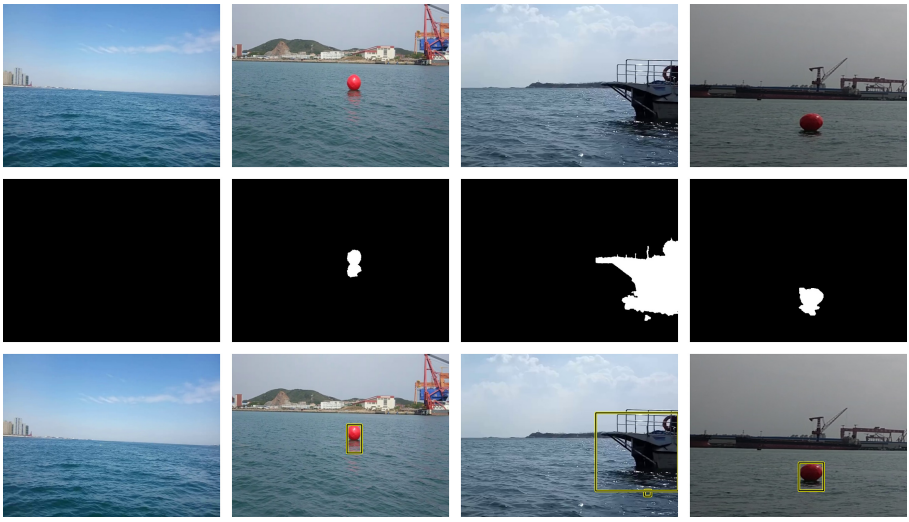

Fig. 4. Examples of obstacle detection. The image in the first row is an image scaled to 640×480. The image in the second row is the corresponding saliency map. The image in the third row is the result of obstacle detection.

In this paper, the RANSAC [14,15] line fitting algorithm, the Canny [18] detection algorithm and our algorithm are compared and tested on the test image set. The results are shown in the Table. 1.

Table 1. Comparison experiment results of water-boundary-line detection

	RANSAC	Canny	Our method
Average error/(pixel)	9.92	14 7.22	4.31
Accuracy/(%)	67.2	77.8	93.1
Processing time/(ms/frame)	5	186	13

4.2 Results of Obstacle Detection

Figure 4 shows the intuitive effect of our algorithm in obstacle detection. It can be seen that our algorithm can achieve good results no matter where there are obstacles, and there are almost no misjudgments. In the picture of obstacles, we can find that the size of obstacles, changes in lighting, etc., our algorithm has demonstrated robustness. But it can also be found from the pictures in the second column that our algorithm is difficult to deal with the problem of reflection of obstacles at sea.

In order to objectively evaluate the detection performance of the algorithm, this paper uses the precision rate, recall rate and F value to quantify and compare the algorithms in this paper. This article also defines the criterion for the accurate detection of obstacles: when the overlap rate between the target window generated by the algorithm and the artificial marker window is greater than or equal to 0.5, the target window is considered to be valid, that is, the algorithm has accurately detected Obstacle target.

The statistical results of the algorithm in this paper on the test image set are: the precision rate is 0.817, the recall rate is 0.885, and the F value is 0.825.

5 Conclusion

In this paper, we propose an image-based obstacle detection method for automatic obstacle avoidance for unmanned boats. Our method uses superpixels to reduce the amount of computation. At the same time, the semantic information in the captured image is used to segment the seawater area to reduce the search space for obstacles. Finally, our method combines the foreground and background information of the seawater area, making it possible to better detect obstacles from the image.

Acknowledgement. This work was supported in part by the National Science Foundation of China under Grant 61703181 and Grant 61525305, and in part by the Natural Science Foundation of Shanghai under Grant 17ZR1409700 and Grant 18ZR1415300.

References

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S.: Slic superpixels compared to state-of-the-art superpixel methods. IEEE Trans. Pattern Anal. Mach. Intell. **34**(11), 2274–2282 (2012)
2. Almeida, C., et al.: Radar based collision detection developments on USV ROAZ II. In: Oceans 2009-Europe, pp. 1–6. IEEE (2009)
3. Bovcon, B., Perš, J., Kristan, M., et al.: Improving vision-based obstacle detection on USV using inertial sensor. In: Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis, pp. 1–6. IEEE (2017)
4. Bovcon, B., Perš, J., Kristan, M., et al.: Stereo obstacle detection for unmanned surface vehicles by IMU-assisted semantic segmentation. Robot. Auton. Syst. **104**, 1–13 (2018)
5. Gal, O.: Automatic obstacle detection for USV's navigation using vision sensors. In: Schlaefer, A., Blaurock, O. (eds.) Robotic Sailing, pp. 127–140 . Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22836-0_9
6. Gal, O., Zeitouni, E.: Tracking objects using PHD filter for USV autonomous capabilities. In: Sauze, C., Finnis, J. (eds.) Robotic Sailing 2012, pp. 3–12. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-33084-1_1
7. Guo, Y., Romero, M., Ieng, S.H., Plumet, F., Benosman, R., Gas, B.: Reactive path planning for autonomous sailboat using an omni-directional camera for obstacle detection. In: 2011 IEEE International Conference on Mechatronics, pp. 445–450. IEEE (2011)

8. Heidarsson, H.K., Sukhatme, G.S.: Obstacle detection from overhead imagery using self-supervised learning for autonomous surface vehicles. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3160–3165. IEEE (2011)
9. Kristan, M., Kenk, V.S., Kovačič, S., Perš, J.: Fast image-based obstacle detection from unmanned surface vehicles. IEEE Trans. Cybern. **46**(3), 641–654 (2015)
10. Li, C., Cao, Z., Xiao, Y., Fang, Z.: Fast object detection from unmanned surface vehicles via objectness and saliency. In: 2015 Chinese Automation Congress (CAC), pp. 500–505. IEEE (2015)
11. Mou, X., Wang, H.: Wide-baseline stereo-based obstacle mapping for unmanned surface vehicles. Sensor **18**(4), 1085 (2018)
12. Wang, H., Wei, Z.: Stereovision based obstacle detection system for unmanned surface vehicle. In: 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 917–921. IEEE (2013)
13. Wang, H., Wei, Z., Ow, C.S., Ho, K.T., Feng, B., Huang, J.: Improvement in real-time obstacle detection system for USV. In: 2012 12th International Conference on Control Automation Robotics & Vision (ICARCV), pp. 1317–1322. IEEE (2012)
14. Wang, H., Wei, Z., Wang, S., Ow, C.S., Ho, K.T., Feng, B.: A vision-based obstacle detection system for unmanned surface vehicle. In: 2011 IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM), pp. 364–369. IEEE (2011)
15. Wang, H., et al.: Real-time obstacle detection for unmanned surface vehicle. In: 2011 Defense Science Research Conference and Expo (DSR), pp. 1–4. IEEE (2011)
16. Wang, J., Gu, W., Zhu, J., Zhang, J.: An unmanned surface vehicle for multi-mission applications. In: 2009 International Conference on Electronic Computer Technology, pp. 358–361. IEEE (2009)
17. Yan, R.J., Pang, S., Sun, H.B., Pang, Y.j.: Development and missions of unmanned surface vehicle. J. Marine Sci. Appl. **9**(4), 451–457 (2010)
18. Yuxing, D., Weining, L., Shuang, W.: Study of sea-sky-line detection algorithm based on Canny theory. Comput. Meas. Control. **18**(3), 697–698 (2010)
19. Zhu, W., Liang, S., Wei, Y., Sun, J.: Saliency optimization from robust background detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2814–2821. IEEE (2014)