Received: 21 October 2019 | Revised: 21 June 2020 | Accepted: 10 August 2020

DOI: 10.1002/rob.21983

REGULAR ARTICLE

WILEY

Efficient obstacle detection based on prior estimation network and spatially constrained mixture model for unmanned surface vehicles

Jingyi Liu[1] | Hengyu Li[1] | Jun Luo[1] | Shaorong Xie[1] | Yu Sun[2]

[1]School of Mechatronic Engineering and Automation, Shanghai University, 99 Shangda Road, Shanghai, China

[2]Department of Mechanical and Industrial Engineering, University of Toronto, 5 King's College Road, Toronto, Ontario, M5S 3G8, Canada

Correspondence

Hengyu Li, School of Mechatronic Engineering and Automation Shanghai University, 99 Shangda Road, Shanghai 200444, China.
Email: lihengyu@shu.edu.cn

Funding information

National Natural Science Foundation of China, Grant/Award Numbers: 61933008, 61525305, 61827812; Key Research and Development Project of Jiangxi Province of China, Grant/Award Number: 20192BBEL50004

Abstract

Recently, spatially constrained mixture model has become the mainstream method for the task of vision-based obstacle detection in unmanned surface vehicles (USVs), and has shown its potential of modeling the semantic structure of the marine environment. However, the expectation maximization (EM) optimization of this model is quite sensitive to initial values and easily falls into a local optimal solution in the presence of significant rolling and pitching in rough seas. In addition, existing methods based on spatially constrained mixture model are susceptible to false positives in the presence of sun glitter. In this paper, a prior estimation network (PEN) is proposed to improve the mixture model, which together enable reliable monocular obstacle detection for USVs. We develop a weakly supervised E-step to train the PEN for learning the semantic structure of marine images and estimating initial class priors in obstacle detection. To mitigate the influence of poor initial parameters on the convergence of EM optimization, we use the priors estimated by the PEN to calculate the initial parameters of the mixture model and automatically adjust the hyper priors on the semantic components in the mixture model. The output of the PEN is also applied to set the probability values of the outlier component in the mixture model, aiming to reduce false positives caused by sun glitter. Experimental results show that our approach outperforms the current state-of-the-art monocular method by 15% improvement in sea edge estimation and a 3.3% increase in *F*-score on the marine obstacle detection data set, as well as 69.5% improvement in sea edge estimation and a 39.2% increase in *F*-score on our data set, while running over 40 fps.

KEYWORDS

computer vision, environmental monitoring, marine robotics, obstacleavoidance, perception

1 | INTRODUCTION

In recent decades, there has been considerable focus on the development of unmanned surface vehicles (USVs) due to the advantages of small size, high mobility, good concealment, and low cost (Benjamin, Leonard, Curcio, & Newman, 2006; Liu, Zhang, Yu, & Yuan, 2016). At present, USVs are widely used in oceanographic research, coordinated rescue, electronic warfare, reconnaissance, and patrolling missions (He, Wang, Dai, & Luo, 2019; R. Zhang et al., 2014; Zhao, Li, & Shi, 2016; Zheng & Feroskhan, 2017). All of the missions listed above require USVs to operate autonomously in highly diverse environments. However, various obstacles may appear on a preplanned path during

navigation. Thus, it is necessary for a practical USV to have an onboard logic used for detecting potentially dangerous obstacles and helping control systems adjust and modify the planned route. To enhance the safety of USVs, an emphasis on obstacle detection is of paramount importance.

There are a variety of sensing modalities available for USVs to observe the surroundings, including radar, sonar, LIDAR, and visual sensors (Liu et al., 2016). In particular, vision-based obstacle detection has become a more popular choice in recent years. One reason for this is that visual sensors can provide richer details of targets than other range sensors, which is helpful for improving the environmental perception ability of USVs. Another reason is that cameras are small, inexpensive, and low in power consumption. These characteristics make them suitable for small USVs that have limitations of payload and power consumption. In addition, it is difficult for the commonly used radar or LIDAR to detect flat or small objects in the marine environment, such as debris, floating wood, plastic bottles, and emerging scuba divers (Bovcon, Perš, & Kristan, 2017; Kristan, Kenk, Kovačič, & Perš, 2016). Moreover, the advances in graphical processing units (GPUs) and multi-core processing enable real-time performance of computer vision approaches for obstacle detection (Sivaraman & Trivedi, 2013). However, there exist challenges associated with computer vision in marine environments of USVs, such as visual blur due to unfavorable weather conditions (haze and fog), reflections of the surrounding environment in water, glitter, or large highlighted water areas caused by sunlight, dynamic water surface, and varying backgrounds. In this paper, we focus on systems where cameras are available and seek to provide real-time obstacle detection for USVs.

1.1 | Related work

Obstacle detection for USVs using cameras is still a relatively young research area, especially compared with the field of unmanned ground vehicles (UGVs). In recent years, several vision-based methods (Bloisi, Previtali, Pennisi, Nardi, & Fiorini, 2016; Bovcon & Kristan, 2018; Bovcon et al., 2017, 2018; Kristan et al., 2016; Shi, Jin, & Zhang, 2018; Wang, Wei, Wang, Ow, Ho, and Feng, 2011; Wang, Wei, Wang, Ow, Ho, Feng, & Lubing, 2011; Wang et al., 2012; Y. Zhang, Li, & Zang, 2017) have been proposed to solve the problem of obstacle detection in USVs. Most of them (Bloisi et al., 2016; Shi et al., 2018; Wang, Wei, Wang, Ow, Ho, and Feng, 2011; Wang, Wei, Wang, Ow, Ho, Feng, & Lubing, 2011; Wang et al., 2012; Y. Zhang et al., 2017) first detect the horizon line and then search for potential obstacles under the guidance of the estimated horizon line using background modeling (Y. Zhang et al., 2017), object classification (Bloisi et al., 2016), and saliency (Shi et al., 2018; Wang, Wei, Wang, Ow, Ho, and Feng, 2011; Wang, Wei, Wang, Ow, Ho, Feng, & Lubing, 2011; Wang et al., 2012) based methods. However, their assumption of approximating the sea edge to a straight horizon line is often violated in coastal waters or in marinas because the sea edge at that location is no longer a straight line. In addition, the methods (Bloisi et al., 2016; Shi et al., 2018; Wang, Wei, Wang, Ow, Ho, and Feng, 2011; Wang, Wei, Wang, Ow, Ho, Feng, & Lubing, 2011; Wang et al., 2012) use edge detection approaches to extract the horizon, which are quite sensitive to widely distributed noise and strong interference edges from sea waves or land, eventually resulting in poor obstacle detection.

To prevent the above problems, some works (Bovcon & Kristan, 2018; Bovcon et al., 2017, 2018; Kristan et al., 2016) explored semantic segmentation methods based on probabilistic graphical model to detect obstacles for USVs without the guidance of the horizon line. In Kristan et al. (2016), a novel graphical model was proposed to account for the three semantic regions of marine images captured from USVs: sky at the top, haze or land in the middle, and water at the bottom of the image. More specifically, the method assumed a mixture model with three Gaussian components for the three dominant semantic regions and a uniform component for the outliers and adopted a Markov random field (MRF) framework to enforce spatial consistency. An expectation-maximization (EM) algorithm was used for estimating model parameters and generating a water segmentation mask. All blobs of non-water pixels inside the water region were treated as obstacles. This approach did not assume a straight water edge and achieved state-of-the-art results on a marine obstacle detection data set (MODD). In Bovcon et al. (2017), an improved probabilistic graphical model for monocular obstacle detection was presented. This model was an extension of Kristan et al. (2016) by incorporating the boat tilt measurements from the onboard inertial measurement unit (IMU) to improve the performance in the presence of visual ambiguities. In the works (Bovcon & Kristan, 2018; Bovcon et al., 2018), the graphical model (Bovcon et al., 2017) was extended to stereo obstacle detection. However, the approaches (Bovcon & Kristan, 2018; Bovcon et al., 2018, 2017) all require extra IMU sensors for support, which increases cost and power consumption.

Recently, a number of approaches based on deep convolutional neural networks (DCNNs) have been proposed for scene perception of UGVs, including object detection (X. Chen et al., 2016; X. Chen, Ma, Wan, Li, & Xia, 2017; Lee, Kim, Park, Cui, & Kim, 2017) and semantic segmentation (Alvarez, LeCun, Gevers, & Lopez, 2012; L.-C. Chen, Papandreou, Kokkinos, Murphy, & Yuille, 2017; Levi, Garnett, Fetaya, & Herzlyia, 2015; Oliveira, Burgard, & Brox, 2016; X. Zhang et al., 2019). Due to the strong feature extraction and integration capabilities of DCNNs, these methods are able to achieve excellent performance in practice. However, because of many differences of the environment between UGVs and USVs, DCNNs cannot be directly applied to USVs. The most noticeable difference is that the navigable surface of USVs is dynamic and non-flat, and its appearance varies more significantly due to the influence of weather conditions. In addition, DCNNs are notoriously hungry for data; however, there is a lack of large-scale publicly available annotated data sets captured from USVs for training object detection networks or semantic segmentation networks, making DCNNs less frequently used in the field of USVs.

1.2 | Our approach

Up to now, the semantic segmentation model (SSM) proposed in Kristan et al. (2016) stays to be a state-of-the-art monocular obstacle

detection method for USVs. It significantly outperforms other related methods. However, the EM optimization of this kind of method based on mixture model is quite sensitive to initial values (Yang, Lai, & Lin, 2012). The initial parameters of the three Gaussian components of SSM are simply computed from three vertical regions {0, 0.2}, {0.2, 0.4} and {0.6, 1}, written in proportions of the image height. Such assumption on region splitting for initialization is often violated because of significant rolling and pitching in rough seas, resulting in over- or under-segmentation of the water region. In addition to the drawback in the initialization of EM algorithm, SSM also has difficulty in suppressing severe background noise (e.g., sun glitter) while highlighting foreground obstacles similar to the background. The main reason is that it sets the values of the uniform component for the outliers at all pixels to the same probability Figure 1.

In this paper, we build upon the spatially constrained mixture model of SSM (Kristan et al., 2016) and propose a prior estimation network (PEN) to provide prior information for improving the mixture model, which together enable reliable obstacle detection in USVs. We develop a weakly supervised E-step to train our PEN for learning the semantic structure of marine images taken from USVs and estimating initial class priors in practical application. The proposed learning method significantly reduces the time and cost of annotation work. Different from the state-of-the-art SSM, which defines three regions for constructing the initial Gaussian components, we apply the priors estimated by PEN to calculate the initial Gaussian parameters of the mixture model and automatically adjust the hyper priors over the mean values of the Gaussians. It greatly improves the robustness of semantic segmentation in the case of image jitter because of significant rolling and pitching in rough seas. Through training from a weakly supervised data set, PEN is able to learn the structure of the three dominant semantic regions and model the outliers. Thus, the probability values of the outlier component in the class priors estimated by PEN vary from pixel to pixel, and the outlier regions have higher probability values than other regions. To improve the problem of SSM in suppressing severe background noise (e.g., sun glitter) and detecting obstacles similar to the background, we take advantage of the preliminary discriminating power of our network (PEN) and apply the outlier component of the network output to set the probability values of the uniform component at all pixels. The experimental results show that our algorithm achieves better performance than the state-of-the-art SSM, both in the sea edge estimation and obstacle detection.

In summary, the main contributions of this paper include:

- We present a novel semantic segmentation method for monocular obstacle detection in the marine environment based on spatially constrained mixture model which is initialized by a PEN.
- We propose an efficient PEN and develop a weakly supervised E-step to train the network for learning the semantic structure of marine images from weakly annotated data.
- We offer extensive quantitative and qualitative performance evaluations of the proposed method on a publicly available MODD and our data set.

2 | PEN

In this section, we introduce an efficient deep neural network architecture named the PEN, which is used to predict initial class priors for the spatially constrained mixture model. The task of PEN belongs to dense prediction, which predicts every pixel of an image. In the field of deep learning, many networks for dense prediction improve their performance by increasing the depth of the network. However, it will greatly increases running time and memory as a result of redundant parameters. To meet the real-time requirement of obstacle detection, we use the same dilated convolution as ENet (Paszke, Chaurasia, Kim, & Culurciello, 2016) and FSSNet

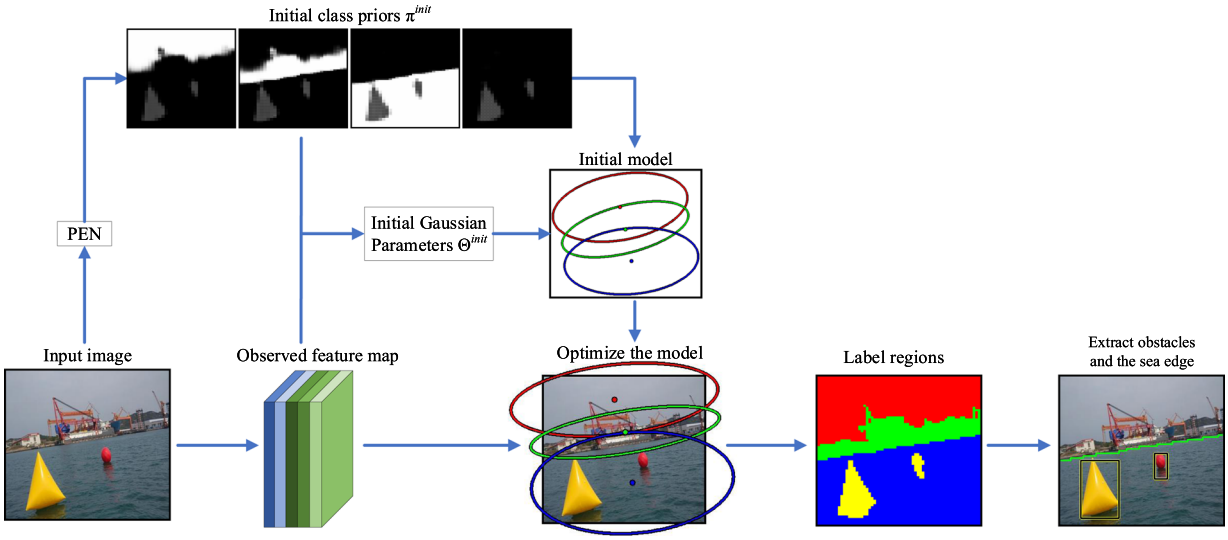


FIGURE 1 The outline of our obstacle detection approach. PEN, prior estimation network [Color figure can be viewed at wileyonlinelibrary.com]

(X. Zhang et al., 2019) to reduce the number of parameters while keeping as much accuracy as possible. Due to the lack of a large, fully supervised data set from USVs to train convolutional neural networks (CNNs) and inspired by the EM algorithm, we develop a novel weakly supervised learning method for training our network from weakly annotated data for reducing the time and cost of annotation.

2.1 | Network architecture

The architecture of PEN is visualized in Figure 2 and detailed in Table 1. It adopts an asymmetric encoder-decoder structure. The encoder stage contains an initial block, which is shown in Figure 3a. In addition, we use the same continuous factorized block and continuous dilated block as FSSNet to extract features in the encoder stage. The role of the decoder is used for upsampling the features from the encoder to the same size as the input image and outputting class prior distribution for each pixel through the softmax function.

In our network, we refer to the basic building block of ResNet (He, Zhang, Ren, & Sun, 2016) as the bottleneck block (see Figure 3b). There are two branches in the bottleneck block that are separated from the input. One branch called identity shortcuts directly connects the input and output, which can speed up the convergence of deep networks and help training. Another branch consists of three convolutional layers with kernel sizes of 1×1, 3×3 and 1×1. The dimension of the feature map is reduced by the first 1×1 convolutional layer and then increased by the second layer. Thus, the 3×3 convolutional layer has smaller input and output dimensions, which significantly reduces the number of parameters. In addition, each convolution layer is followed by a batch normalization (Ioffe & Szegedy, 2015) and PReLU (He, Zhang, Ren, & Sun, 2015) layer.

TABLE 1 Network architecture of prior estimation network

Stage	Name	Block type	Output size[a]
Encoder	Initial block		64 × 64 × 16
	Block1	Downsampling	32 × 32 × 64
		Factorized	32 × 32 × 64
		Factorized	32 × 32 × 64
		Factorized	32 × 32 × 64
		Factorized	32 × 32 × 64
	Block2	Downsampling	16 × 16 × 128
		Dilated $r = 2$	16 × 16 × 128
		Dilated $r = 3$	16 × 16 × 128
		Dilated $r = 5$	16 × 16 × 128
		Dilated $r = 2$	16 × 16 × 128
		Dilated $r = 3$	16 × 16 × 128
		Dilated $r = 5$	16 × 16 × 128
Decoder	Block3	Upsampling	32 × 32 × 64
		Bottleneck	32 × 32 × 64
		Bottleneck	32 × 32 × 64
		Dual Attention	32 × 32 × 32
	Block4	DUpsampling	64 × 64 × 4
		Softmax	64 × 64 × 4

[a]Output sizes are given for an example input of 64 × 64 × 3.

The factorized block and the dilated block in the PEN are both modified from the bottleneck block. The factorized block replaces the 3×3 convolution in the bottleneck block with two stacked asymmetric convolutions (1×3, 3×1), which is equivalent to sliding a two-layer network with the same receptive field as in the 3×3 convolution. It greatly reduces the number of parameters and improves training speed. In our network, we apply four consecutive

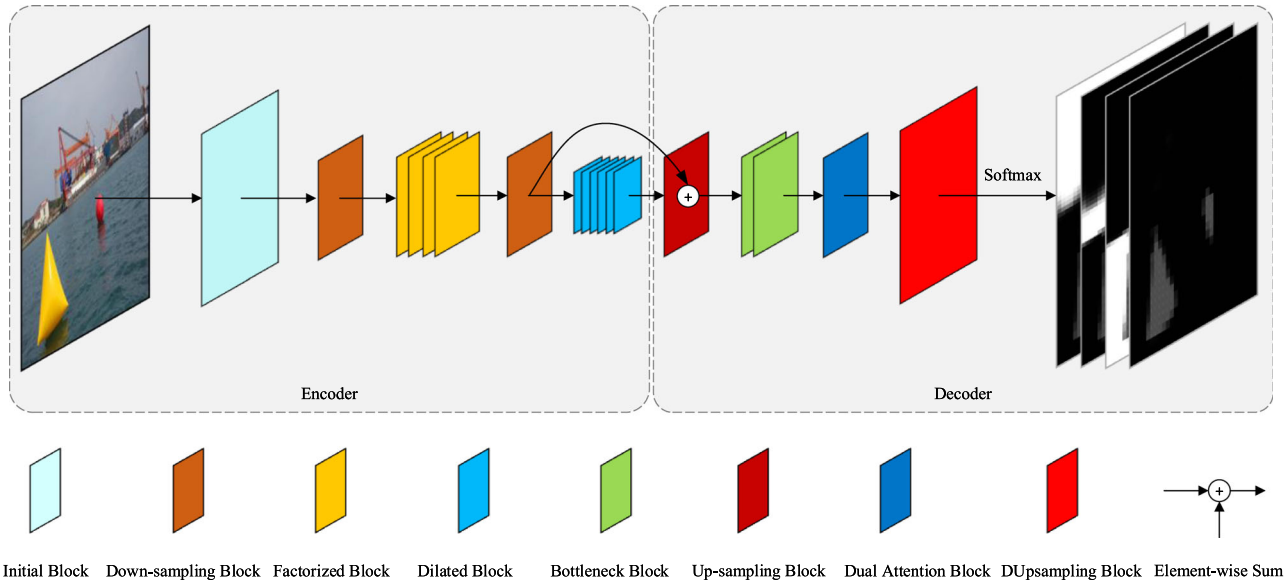


FIGURE 2 Overall architecture of the prior estimation network [Color figure can be viewed at wileyonlinelibrary.com]

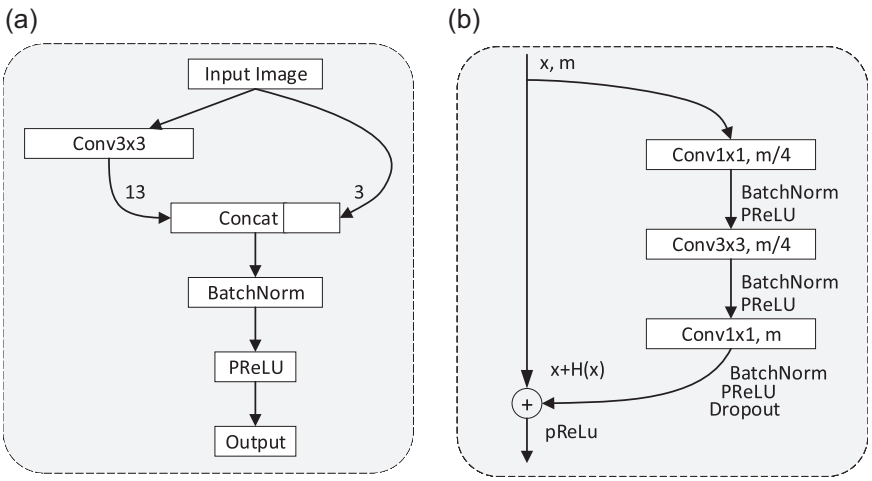


FIGURE 3 Initial block and bottleneck block. (a) Initial block. (b) Bottleneck block. A rectangle represents a convolution with kernel size 1×1 or 3×3. *m* and *m*/4 denote the number of convolution filters, and *Concat* presents concatenate operator

factorized blocks on Block1 of the encoder to extract low-level features. The dilated block uses a 3×3 dilated convolution with a dilation rate to replace the 3×3 convolutional filter in the bottleneck block. The dilated convolution can help to expand the receptive field while keeping the network small-scale and shallow. To ease the gridding problem of dilated convolutions and look wider, different dilation rates are adopted in each dilated block. We choose three stacked dilated blocks as a group and set their dilation rates to 2, 3, and 5. We apply six consecutive dilated blocks on Block2 of the encoder, as shown in Table 1.

We insert a maxpooling-convolutional layer in the identity shortcut branch and apply a 2×2 convolution with stride 2 to replace the first 1×1 convolution in the bottleneck block. This modified bottleneck is used as a downsampling block, which is shown in Figure 4a. Similarly, we replace the 3×3 convolution of the bottleneck block with a deconvolution layer and insert a convolution with a kernel size of 1×1 and a DUpsampling (Tian, He, Shen, & Yan, 2019) layer in the shortcut branch. This changed bottleneck block is used as the upsampling block (see Figure 4b). DUpsampling is a data-dependent upsampling, which allows almost arbitrary feature aggregation, and thus, a better feature aggregation can be leveraged to improve the upsampling performance. In Block3 of the decoder, we adopt a dual attention module (Fu et al., 2019) to adaptively integrate local features with their global dependencies and provide a better feature representation. At the end of the decoder, a softmax layer is assigned to obtain the initial prior distribution for each pixel.

2.2 | Weakly supervised learning

Because of the lack of publicly available, fully supervised data sets to train our network, we constructed a weakly annotated data set. The data set comprises 2,185 marine images. Most of the images were captured from our Jinghai VIII USV (see Section 4 for details) in the coastal waters of Qingdao and Shanghai, China. To prevent the proposed network from overfitting in modeling, we also selected a few of images from MODD (Kristan et al., 2016), SMD (Prasad, Rajan, Rachmawati, Rajabally, & Quek, 2017) and the Internet for increasing the complexity of the data set. For each image in the data set, we manually extracted lines from the three dominant semantic regions, and recorded all pixels in these lines by category (see Figure 5). Furthermore, three Gaussian distributions were computed from the recorded pixels, forming observed components

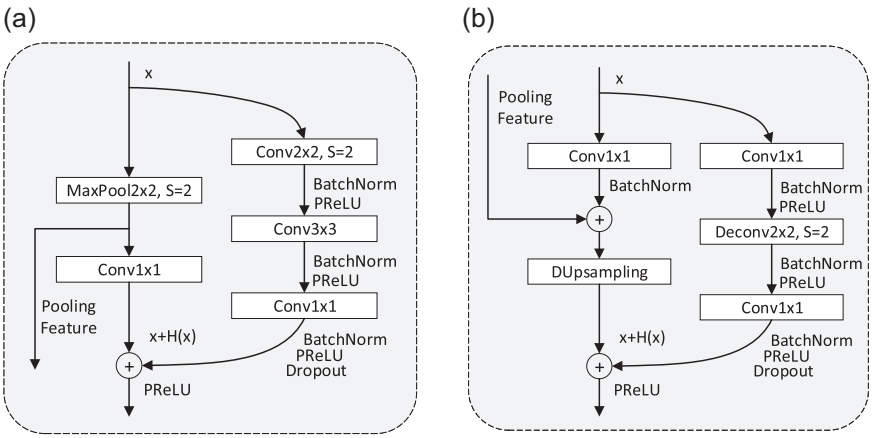


FIGURE 4 Downsampling block and upsampling block. (a) Downsampling block. (b) Upsampling block. *S* denotes stride in a convolutional layer, which is a metric for regulating the movement of a convolutional filter

$\Theta^{obs} = \left\{\mu_k^{obs}, \Sigma_k^{obs}\right\}_{k=1:3}$ where μ and Σ denote the mean and covariance of a Gaussian, respectively. Compared with fully supervised semantic segmentation approaches, which require precise polylines or polygons to mark target boundaries in data sets, our labeling method only samples a few straight lines per image instead of polylines or polygons, which lessens the burden of annotation.

Following the notation from Kristan et al. (2016), we regard each pixel i in an image as an observed feature vector $\boldsymbol{y}_i = [x_i, y_i, c_i^1, c_i^2, c_i^3]^T$, where (x_i, y_i) denotes the pixel's position and (c_i^1, c_i^2, c_i^3) denotes the corresponding color values. Thus, an image is represented as an array of values $\boldsymbol{Y} = \{\boldsymbol{y}_i\}_{i=1:M}$, where M is the number of pixels in the image. We describe each pixel with a four-component mixture model consisting of three Gaussians and a uniform component. Inspired by the EM algorithm, we only employ the E-step of the EM algorithm to obtain the ground truth of each image and propose a weakly supervised E-step for training our network, in which the per-pixel class priors treated as ground truth are computed by

$$\pi_{ik}^{GT} = \begin{cases} \dfrac{\phi\left(\boldsymbol{y}_i \mid \mu_k^{obs}, \Sigma_k^{obs}\right)\alpha_{ik}}{\sum_{l=1}^{3}\phi\left(\boldsymbol{y}_i \mid \mu_l^{obs}, \Sigma_l^{obs}\right)\alpha_{il} + \mathcal{U}(\boldsymbol{y}_i)\alpha_{i4}} & k = 1{:}\,3 \\ \dfrac{\mathcal{U}(\boldsymbol{y}_i)\alpha_{ik}}{\sum_{l=1}^{3}\phi\left(\boldsymbol{y}_i \mid \mu_l^{obs}, \Sigma_l^{obs}\right)\alpha_{il} + \mathcal{U}(\boldsymbol{y}_i)\alpha_{i4}} & k = 4, \end{cases} \tag{1}$$

where $\phi(\cdot|\mu, \Sigma)$ are Gaussian distributions that model the three dominant semantic regions, and $\mathcal{U}(\cdot)$ is a uniform distribution corresponding to the outliers. $\left\{\mu_k^{obs}, \Sigma_k^{obs}\right\}_{k=1:3}$ are observed Gaussian parameters from weak annotations. In our settings, the uniform distribution at each pixel is set to a low positive constant value $\varepsilon = 1 \times 10^{-15}$. α_{ik} denotes the prior probability when the class of the i-th pixel is k, which is set as

$$\alpha_{ik} = \begin{cases} \frac{1}{3} \times (1 - \varepsilon) & k = 1{:}\,3 \\ \varepsilon & k = 4. \end{cases} \tag{2}$$

In network training, we estimate the per-pixel class priors π_i^{GT} from the weak annotations using (1), then treat these estimated priors as ground truth and employ the training procedure, as illustrated in Figure 6. To train PEN, a new loss function for an image is defined as

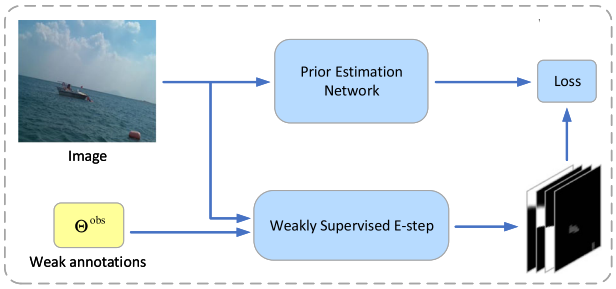


FIGURE 6 Prior estimation network training using weak annotations [Color figure can be viewed at wileyonlinelibrary.com]

$$L(\boldsymbol{\pi}^*, \boldsymbol{\pi}^{GT}) = \frac{1}{4 \times M}\sum_{i=1}^{M}\sum_{k=1}^{4}\left[-\log\left(1 - \left|\pi_{ik}^* - \pi_{ik}^{GT}\right|\right)\right], \tag{3}$$

where $\boldsymbol{\pi}^* = \{\pi_i^*\}_{i=1:M}$ is the prediction result of PEN. In Section 3, $\boldsymbol{\pi}^*$ is used for initializing the mixture model; thus, it is also written as $\boldsymbol{\pi}^{init}$ in our paper.

2.3 | Implementation details

Our network is implemented based on PyTorch and trained end-to-end, pixel-to-pixel. Kristan et al. (2016) noted that good obstacle detection is achievable on an image larger than 50 × 50 pixels, so we rescale original images to 64 × 64 pixels and use these reduced-size images as input to our network. An Adam optimization algorithm (Kingma & Ba, 2014) is adopted to train our network. We randomly initialize all convolutional layers by drawing weights from a Gaussian distribution $\mathcal{N}(0, \frac{2}{k^2 \times c})$, which is described in He et al. (2015). Here, k denotes the filter size of the layer, and c is the number of output channels of the layer. The probability of dropout is set to 0.3. We use a learning rate of 0.001 and a momentum of 0.9. The learning rate is decayed by a factor of 0.1 every 50 epochs. The size of a mini-batch

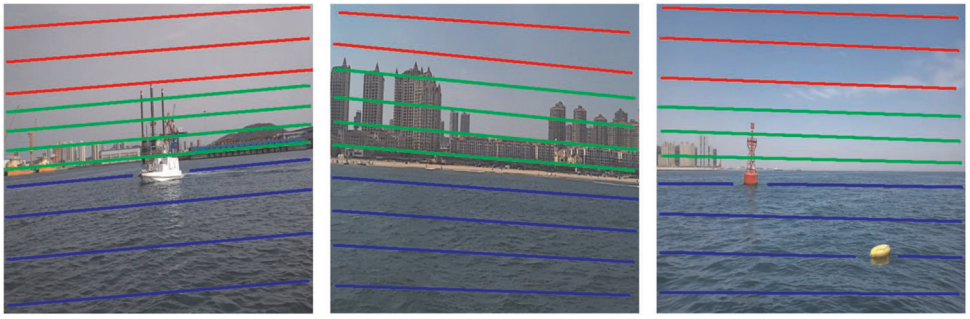


FIGURE 5 Some results of manual annotation. The red lines in the images are extracted from the sky region, the green lines from the middle region, and the blue lines from the water region. The pixels in these lines are used for calculating Θ^{obs} [Color figure can be viewed at wileyonlinelibrary.com]

is set to 32. The network is trained from scratch on a NVIDIA GTX1080Ti GPU in 120 epochs.

According to the data splitting strategy reported in Borovicka, Jirina, Kordik, and Jirina (2012), we randomly divide our weakly annotated data set into 1,585 images for training, 200 images for validation, and 400 images for testing. First, we downsample the images from 512×512 to 64×64 and then train our network on the training data set and validate on the validation data set. Finally, the performance of our network is evaluated on the test data set.

3 | AUGMENTED MIXTURE MODEL

We assume a mixture model consisting of three Gaussian distributions for modeling the three dominant semantic regions and a uniform distribution for the outliers. Following the notation in Section 2, the probability of the i-th pixel feature vector $\boldsymbol{y}_i$ can be written as

$$p(\boldsymbol{y}_i|\Theta,\boldsymbol{\pi}) = \sum_{k=1}^{3}\phi(\boldsymbol{y}_i|\mu_k,\Sigma_k)\pi_{ik} + \mathcal{U}(\boldsymbol{y}_i)\pi_{i4}, \tag{4}$$

where $\phi(\cdot|\mu_k,\Sigma_k)$ is a Gaussian kernel, and $\mathcal{U}(\cdot)$ is a uniform distribution. $\Theta = \{\mu_k,\Sigma_k\}_{k=1:3}$ denotes the means and covariances of Gaussian components. $\boldsymbol{\pi} = \{\pi_i\}_{i=1:M}$ denotes class priors for all pixels. The ith pixel label x_i is an unobserved random variable governed by the class prior $\pi_i = [\pi_{i1}, ..., \pi_{ik}, ..., \pi_{i4}]$ with $\pi_{ik} = p(x_i = k)$.

To encourage segmentation into three vertically stacked semantic regions, SSM defines a set of static hyper priors over the Gaussian means in (4) for all images. However, the spatial centers of Gaussian distributions are not constant in different images, it is not appropriate to define static hyper priors for all images. Therefore, we apply the output of PEN to calculate the initial Gaussian parameters $\Theta^{init} = \left\{\mu_k^{init}, \Sigma_k^{init}\right\}_{k=1:3}$ and treat them as hyper priors φ_0 over the Gaussian means in the mixture model, that is, $p(\Theta|\varphi_0) = \prod_{k=1}^{3}\phi\left(\mu_k|\mu_k^{init},\Sigma_k^{init}\right)$. Here, the initial Gaussian parameters Θ^{init} are calculated from the initial class priors $\boldsymbol{\pi}^{init}$ (which are estimated by PEN) and the observed features of all pixels $\{\boldsymbol{y}_i\}_{i=1:M}$, which can be written as

$$\mu_k^{init} = \frac{\sum_{i=1}^{M}\pi_{ik}^{init}\boldsymbol{y}_i}{\sum_{i=1}^{M}\pi_{ik}^{init}}, \tag{5}$$

$$\Sigma_k^{init} = \frac{\sum_{i=1}^{M}\pi_{ik}^{init}\boldsymbol{y}_i\boldsymbol{y}_i^T}{\sum_{i=1}^{M}\pi_{ik}^{init}} - \mu_k^{init}\left(\mu_k^{init}\right)^T. \tag{6}$$

To make segmentation smooth, the priors $\boldsymbol{\pi} = \{\pi_i\}_{i=1:M}$ and posteriors $\boldsymbol{P} = \{\boldsymbol{p}_i\}_{i=1:M}$ over the pixel class labels are both treated as random variables that form a MRF to model spatial dependencies between image pixels, where the components of $\boldsymbol{p}_i$ are defined as

$$p_{ik} = p(x_i = k|\boldsymbol{y}_i,\Theta,\boldsymbol{\pi},\varphi_0) = \begin{cases} \dfrac{p(\boldsymbol{y}_i|x_i=k,\Theta)\pi_{ik}}{\sum_{l=1}^{3}p(\boldsymbol{y}_i|x_i=l,\Theta)\pi_{il} + \mathcal{U}(\boldsymbol{y}_i)\pi_{i4}} & k = 1{:}\,3 \\ \dfrac{\mathcal{U}(\boldsymbol{y}_i)\pi_{ik}}{\sum_{l=1}^{3}p(\boldsymbol{y}_i|x_i=l,\Theta)\pi_{il} + \mathcal{U}(\boldsymbol{y}_i)\pi_{i4}} & k = 4. \end{cases} \tag{7}$$

MRF is a set of random variables having a Markov property described by an undirected graph, which provides a powerful and formal way to account for spatial dependencies between image pixels (Li, 2009). Like Diplaros, Vlassis, and Gevers (2007), the MRF of the priors in our model is approximated by $p(\boldsymbol{\pi}) \approx \prod_i^M p(\pi_i|\pi_{N_i})$, and π_{N_i} is a mixture distribution over the priors of the neighbors of a pixel i, which is determined by $\pi_{N_i} = \sum_{j\in N_i, j\neq i}\lambda_{ij}\pi_j$, where λ_{ij} is a fixed positive weight, and for each pixel i, $\sum_{j\in N_i, j\neq i}\lambda_{ij} = 1$ holds. The conditional density $p(\pi_i|\pi_{N_i})$ represents the potentials in the MRF, which are defined as $p(\pi_i|\pi_{N_i}) \propto \exp(-\frac{1}{2}E(\pi_i,\pi_{N_i}))$ with the exponent defined as $E(\pi_i,\pi_{N_i}) = D(\pi_i\|\pi_{N_i}) + H(\pi_i)$, where the term $D(\pi_i\|\pi_{N_i})$ is the Kullback-Leibler divergence and $H(\pi_i)$ is the entropy. The divergence term $D(\pi_i\|\pi_{N_i})$ is a similarity measure of the priors between the ith pixel and its neighbors, which penalizes the difference between π_i and π_{N_i} and constrains neighboring pixels to have the same class labels. The entropy term $H(\pi_i)$ constrains the class prior π_i to be as informative as possible. Similarly, the posterior over $\boldsymbol{P}$ is approximated as $p(\boldsymbol{P}|\boldsymbol{Y},\Theta,\boldsymbol{\pi},\varphi_0) \propto \prod_{i=1}^{M}\exp(-\frac{1}{2}E(\boldsymbol{p}_i,\boldsymbol{p}_{N_i}))$, where $\boldsymbol{p}_{N_i}$ is defined similar to π_{N_i}. Thus, the following joint probability density function is derived:

$$\begin{aligned} p(\boldsymbol{P},\boldsymbol{Y},\Theta,\boldsymbol{\pi}|\varphi_0) &= p(\boldsymbol{P}|\boldsymbol{Y},\Theta,\boldsymbol{\pi},\varphi_0)p(\boldsymbol{Y}|\Theta,\boldsymbol{\pi})p(\boldsymbol{\pi})p(\Theta|\varphi_0) \\ &\propto \exp\left[\sum_{i=1}^{M}\log p(\boldsymbol{y}_i,\Theta|\boldsymbol{\pi},\varphi_0)\right. \\ &\quad \left. -\frac{1}{2}(E(\pi_i,\pi_{N_i}) + E(\boldsymbol{p}_i,\boldsymbol{p}_{N_i}))\right]. \end{aligned} \tag{8}$$

Due to the coupling between π_i/π_{N_i} and $\boldsymbol{p}_i/\boldsymbol{p}_{N_i}$, the optimization of (8) is not straightforward. Following the derivations reported in Kristan et al. (2016) and Diplaros et al. (2007), we introduce auxiliary priors $\{\boldsymbol{s}_i\}_{i=1:M}$ and posteriors $\{\boldsymbol{q}_i\}_{i=1:M}$ to decouple them, leading to an EM-like algorithm for maximizing (8). In the E-step, we maximize (8) over $\boldsymbol{s}_i$ and $\boldsymbol{q}_i$, assuming Θ and $\boldsymbol{\pi}$ are fixed. This gives

$$\begin{aligned} \boldsymbol{s}_i^{(t+1)} &= \xi_{s_i}\pi_i^{(t)}\circ\pi_{N_i}^{(t)}, \\ \boldsymbol{q}_i^{(t+1)} &= \xi_{q_i}\boldsymbol{p}_i^{(t)}\circ\boldsymbol{p}_{N_i}^{(t)}, \\ \hat{\boldsymbol{s}}_i^{(t+1)} &= \boldsymbol{s}_i^{(t+1)} + \boldsymbol{s}_{N_i}^{(t+1)}, \\ \hat{\boldsymbol{q}}_i^{(t+1)} &= \boldsymbol{q}_i^{(t+1)} + \boldsymbol{q}_{N_i}^{(t+1)}, \end{aligned} \tag{9}$$

where t indicates the iteration step and $\circ$ denotes the Hadamard product. ξ_{s_i} and ξ_{q_i} are normalization constants. Here, $\boldsymbol{p}_{N_i}$, $\boldsymbol{s}_{N_i}$, and $\boldsymbol{q}_{N_i}$ are the distributions defined in the same spirit as π_{N_i}. In practice, the neighborhood priors are determined by the following convolution $\pi_{N,k} = \pi_{.k}{}^*\lambda$, where λ is a small discrete Gaussian kernel (e.g., 3×3) with its central element set to zero and its elements summing to one.

In the M-step, we fix $\boldsymbol{s}_i$ and $\boldsymbol{q}_i$ and maximize (8) over Θ and $\boldsymbol{\pi}$:

$$\begin{aligned}
\pi_i^{(t+1)} &= \left(\hat{\boldsymbol{s}}_i^{(t+1)} + \hat{\boldsymbol{q}}_i^{(t+1)}\right)/4, \\
\tilde{\mu}_k^{(t+1)} &= \beta_k^{-1} \sum_{i=1}^{M} \hat{q}_{ik}^{(t+1)} \boldsymbol{y}_i, \\
\mu_k^{(t+1)} &= \Lambda_k \left[\left(\Sigma_k^{(t+1)}\right)^{-1} \tilde{\mu}_k^{(t+1)} + \left(\Sigma_k^{init}\right)^{-1} \mu_k^{init} \right], \\
\Sigma_k^{(t+1)} &= \beta_k^{-1} \sum_{i=1}^{M} \hat{q}_{ik}^{(t+1)} \left(\boldsymbol{y}_i - \tilde{\mu}_k^{(t+1)}\right)\left(\boldsymbol{y}_i - \tilde{\mu}_k^{(t+1)}\right)^T,
\end{aligned} \tag{10}$$

where we define $\beta_k = \sum_{i=1}^{M} \hat{q}_{ik}^{(t+1)}$ and $\Lambda_k = \left[\left(\Sigma_k^{(t+1)}\right)^{-1} + \left(\Sigma_k^{init}\right)^{-1} \right]^{-1}$.

The expectation-maximization (EM) algorithm (Diplaros et al., 2007; Moon, 1996) aims to find maximum a posteriori (MAP) estimation of parameters in latent variable models. The EM iteration alternates between performing an expectation (E) step and a maximization (M) step. The E-step evaluates the expectation of the log-likelihood using the current parameter values and the observation. The M-step then provides a new estimate of the parameters.

3.1 | Initialization

The above EM-like algorithm requires initial parameters Θ^{init} and $\boldsymbol{\pi}^{init}$ to perform iterative optimization. In fact, the performance of the EM algorithm is easily affected by poor initial values and falls into the local optimal solution. In SSM (Kristan et al., 2016), the initial parameters of Gaussian components at the first frame are simply computed from three regions split by the ratios $\{0, 0.2\}, \{0.2, 0.4\}$, and $\{0.6, 1\}$ in the image height direction. However, the assumption on region splitting is often violated because of boat inclination at turning maneuvers or boat tilting forward and backward. Although the continuity of sequential images is applied to improve the problem for the intermediate frames, SSM still cannot handle the case where the area of obstacles under the sea edge is too large. Because the pixels in the large obstacles will cause the Gaussian parameters of water component calculated by the above assumption to be inaccurate. To solve this problem, we propose a PEN to predict initial class priors for all pixels $\boldsymbol{\pi}^{init}$ and then calculate the initial Gaussian parameters $\Theta^{init} = \left\{\mu_k^{init}, \Sigma_k^{init}\right\}_{k=1:3}$ using (5) and (6). From Figure 7, we can see that the initial Gaussian parameters estimated by our method are better than that of SSM in the presence of significant rolling. In addition, unlike SSM, which initializes the priors on the three Gaussian components to equal values and the prior on the uniform component to a low constant value, we directly apply the output of PEN as initial class priors $\boldsymbol{\pi}^{init}$ to speed up the convergence of the EM algorithm. The EM procedure for fitting the mixture model to the input image is summarized in Algorithm 1. In our settings, EM is considered to be convergent when either of the following conditions is met: i) the maximum number of iterations is reached, which is empirically set to 10; and ii) $\sum| \boldsymbol{\pi}^{(t+1)} - \boldsymbol{\pi}^{(t)} |/N < 0.01$, where N is the number of elements of $\boldsymbol{\pi}$.

Algorithm 1 EM for semantic segmentation

Initialization:
1. Initialize the priors $\boldsymbol{\pi}^{init}(=\boldsymbol{\pi}^{(0)})$ using the output of PEN.
2. Initialize the Gaussian parameters $\Theta^{init}(=\Theta^{(0)})$ according to (5) and (6).
3. Assign Θ^{init} to the weak priors φ_0.

E-step:
4. Calculate the pixel posteriors $\boldsymbol{p}_k^{(t)}$ using (7) and the current estimates of $\boldsymbol{\pi}^{(t)}$ and $\Theta^{(t)}$.
5. Calculate new auxiliary priors $\hat{\boldsymbol{s}}_k^{(t+1)}$ and posteriors $\hat{\boldsymbol{q}}_k^{(t+1)}$ according to (9).

M-step:
6. Calculate the new priors $\boldsymbol{\pi}^{(t+1)}$ and Gaussian parameters $\Theta^{(t+1)}$ using (10).
7. Repeat steps 4 to 6 until convergence.

3.2 | Implementation details

The above EM algorithm is performed on a reduced-size image of 64×64, and then we rescale the posterior map to the original image size. The reduced size is set to meet the detection of small obstacles that present danger to the USV. The uniform distribution $\mathcal{U}(\boldsymbol{y}_i)$ in (4) is defined over all image pixels. The value of the uniform distribution has a great impact on outlier detection. If the value is too small, it is not conducive to detecting obstacles that are less different from the surrounding water region. Conversely, if the value is too large, it is easy to detect fragmented glitter regions as obstacles when the boat is directly

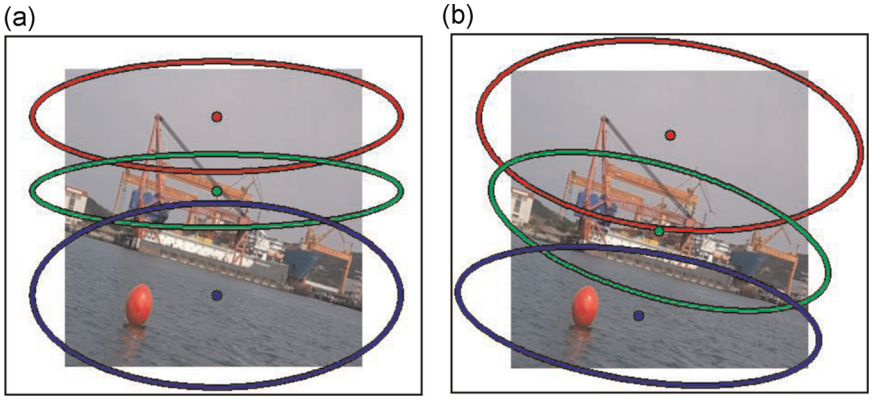


FIGURE 7 Visualization of the spatial part of the initial Gaussian components. (a) Semantic segmentation model and (b) our approach [Color figure can be viewed at wileyonlinelibrary.com]

facing the sun. Specifically, it is not a good idea to detect different images using the same uniform distribution. To solve the problem, we use the outlier component of $\boldsymbol{\pi}^{init}$ to set the values of the uniform distribution for all pixels. It is more flexible than SSM, which sets the value of the uniform distribution at each pixel to the same probability.

The mixture model with PEN is fitted to each image through the EM algorithm, and the smoothed posterior distribution $\hat{\boldsymbol{q}}_{ik}$ across the four semantic components for each pixel is eventually estimated. As seen from (9), $\hat{\boldsymbol{q}}_{ik}$ takes into account the posteriors of the neighborhood pixels, which tends to make neighboring pixels have similar posteriors. Therefore, the semantic segmentation result obtained by $\hat{\boldsymbol{q}}_{ik}$ is smoother than that obtained by $\boldsymbol{p}_{ik}$. According to the result of $\text{argmax}_k \hat{\boldsymbol{q}}_{ik}$, we classify each pixel into one of the four classes and obtain a labeled image. Moreover, all pixels that do not correspond to the sea component are regarded as part of an obstacle. In our experiments, all obstacles within the connected water region are located by bounding boxes, and the sea edge is used to specify the range of water.

4 | EXPERIMENTS

In this section, we denote our SSM with a PEN (Algorithm 1) as PEN-SSM. The experiments were divided into two parts. In the first part, we evaluated the performance of PEN on our weakly annotated test set to demonstrate accurate and real-time prior estimation. In the second part, we analyzed how various improvement elements of PEN-SSM affect its performance and compared it to SSM. All experiments were performed on a desktop PC with an Intel Core i7-6700K 4 GHz CPU with 32 GB RAM, with an externally connected NVIDIA GTX1080Ti GPU with 11 GB of memory.

4.1 | Evaluation of PEN

To evaluate our proposed network, we added a softmax layer to both ENet (Paszke et al., 2016) and FSSNet (X. Zhang et al., 2019) as the output layer and compared them with our network. For a fair comparison, we trained all networks on the 64×64 images using the same learning scheduler. In addition, we reported our results using the following metrics, where N is the sample number, and $\hat{v}_i$ and v_i denote the predicted value and the ground truth value of the ith sample, respectively:

i. Mean absolute error (MAE)

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N} | v_i - \hat{v}_i |, \tag{11}$$

ii. Mean squared error (MSE)

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N} (v_i - \hat{v}_i)^2, \tag{12}$$

iii. Root mean squared error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (v_i - \hat{v}_i)^2}, \tag{13}$$

where the metrics (MAE, MSE, and RMSE) are error metrics, meaning that the lower the value, the better the performance.

4.1.1 | Results

Table 2 shows the performances of different methods on our weakly annotated test set, which contained 400 images (see Sections 2.2 and 2.3 for details). As we can see from Table 2, our network achieved 120 fps on the test set, which was slightly slower than the best counterpart (FSSNet). Aside from the running speed and the parameters, our network achieved the lowest error.

4.2 | Evaluation of obstacle detection

To demonstrate the performance of PEN-SSM, we experimented on a public marine scene data set (MODD; Kristan et al., 2016) and our own data set. We set SSM (Kristan et al., 2016) as a baseline because it is one of the most accurate and fastest obstacle detection algorithms in USV. During the operation of USVs, SSM needs to use the optimized parameters in the previous frame to guide the processing of the current frame. Specifically, it utilizes the parameters of the converged model from the previous time-step for the initialization of the EM algorithm in the current time-step. To compare and evaluate the detection performance of a single-frame image, we implemented a variant of SSM, which we denoted by SSM(S). In contrast to SSM, the SSM(S) does not use the continuity of sequential images in the videostream for initialization of Gaussian components. In addition, we implemented two variants of our approach to analyze how much each improvement contributes to the final performance. We denoted the two variants by PEN-SSM(P) and PEN-SSM(U). PEN-SSM(P) and PEN-SSM(U) both use the output of the PEN network to obtain the

TABLE 2 Comparison of different networks on the test set

Network	Parameters	fps	MAE	MSE	RMSE
ENet	0.37*M*	102	0.1630	0.0132	0.1150
FSSNet	**0.20*M***	**125**	0.1658	0.0130	0.1142
PEN	0.23*M*	120	**0.1453**	**0.0099**	**0.0994**

Note: The bold value represents the best score under each evaluation metric.

Abbreviations: MAE, mean absolute error; MSE, mean squared error; RMSE, root mean squared error.

initial Gaussian parameters for implementing EM iterations. Different from PEN-SSM, PEN-SSM(P) does not employ the output of PEN as the initial class priors $\boldsymbol{\pi}^{init}$ in the EM procedure. Specifically, the class priors over all pixels in PEN-SSM(P) is initialized to equal probabilities for the three Gaussian components, while the prior on the uniform component is set to a low constant value (e.g., 7.9×10^{-15}). In contrast to PEN-SSM, which uses the outlier component of $\boldsymbol{\pi}^{init}$ to set the values of the uniform distribution for all pixels, PEN-SSM(U) adopts the same method as SSM to set the value of the uniform distribution at each pixel to a constant, for instance, $\mathcal{U}(\mathbf{y}_i) = 1/(W \times H)$ where W and H denote the width and height of an image. PEN-SSM(U) and PEN-SSM are the same otherwise.

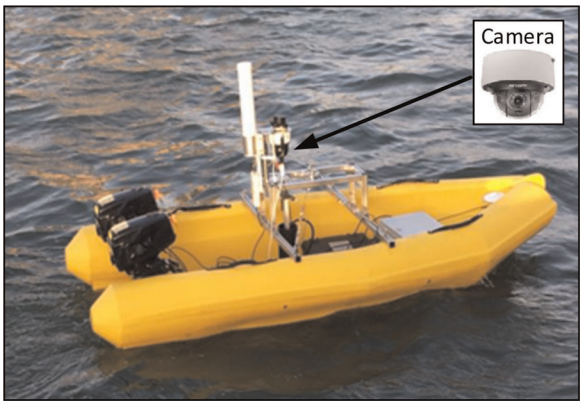


FIGURE 8 Jinghai VIII USV [Color figure can be viewed at wileyonlinelibrary.com]

4.2.1 | Data sets

We evaluated PEN-SSM on two maritime obstacle detection data sets: MODD and our own marine image data set (MID).

MODD: This data set consists of twelve video sequences for marine scene obstacle detection that were captured from a small 2.2 m USV. There are 4,454 fully annotated images with a resolution of 640×480 pixels. The first ten videos in this data set were acquired under normal conditions with different backgrounds and obstacles. The last two videos were captured when the USV was directly facing the sun, causing large amount of fragmented glitter or large highlighted areas on the water. In addition, facing the sun also caused significant changes in the contrast and color of the images, posing challenges to the visual algorithm. Images in the data set have a distinctive visual structure: sea in the lower part, sky in the upper part, and haze or land in the middle. The sea may contain obstacles, such as boats, plastic bottles, buoys, and even a diver. In the data set, obstacles are separated into large and small obstacles, which are annotated by bounding boxes. Large obstacles are those that straddle the horizon and span both the visual area of the sea and the area of the sky or shore. The small obstacles are those that are visually fully enclosed with the sea from all sides. This data set also provides the landmarks of the horizon for each image.

MID: The MID[1] contains eight video sequences for marine obstacle detection. There are 2,655 labeled images with a resolution of 640×480 pixels in the data set, which were captured from our Jinghai VIII USV (see Figure 8). The USV is 2.7 m long and adopts two-propeller electric propulsion that has the advantages of high thrust and flexible steering. It can reach a maximum speed of seven knots. There are two industrial computers carried on the USV. One is mainly responsible for automatic navigation and obstacle avoidance, and the other is used for image acquisition and processing. A high-definition camera with a 50° field of view is mounted on the vehicle approximately 1 m above the water surface. The data set was acquired in the coastal waters of Qingdao and Shanghai, China, over several months at different times of the day. As we can see from Figure 9, the first frames of Sequence 1~5 violate SSM's assumption on region splitting due to the boat's attitude changes while SSM assumes three fixed vertical regions split by the ratios {0, 0.2}, {0.2, 0.4}, and {0.6, 1} to initialize the Gaussians of SSM. Sequences 6 and 7 were acquired in yellow waters. The images in the last sequence were captured under severe weather, resulting in a blurred transition from the sea to sky. The data set contains various objects (see Figure 10) and covers a range of different realistic conditions that coastal USVs encounter, including water reflections of the surrounding environment, visual blur caused by unfavorable weather conditions, salient waves, low illumination, sun glitter, and horizon tilt due to waves and USV motion. Like MODD, obstacles in this data set are also separated into large obstacles (those that straddle the water edge) and small obstacles (those that are fully surrounded by water). The distribution of the annotated size of small and large obstacles is shown in Figure 11. In addition, this data set also provides ground truth data for the horizon.

[1]https://github.com/aluckyi/MID.

4.2.2 | Evaluation metrics

In our experiments, we adopt the same evaluation metrics as Kristan et al. (2016). There are two distinct challenges that USVs face: (1) sea edge detection and (2) obstacle detection. Therefore, the evaluation protocol is designed to reflect the above challenges. The former is measured by the root mean square error (RMSE) of the sea edge position and its standard deviation over all sequences (μ_{edg}, σ_{edg}). The latter is measured via the efficiency of small obstacle detection, including precision (*Prec*), recall (*Rec*), F-score (*F*), and the number of false positives per frame (αFP).

4.2.3 | Results

The results of quantitative comparison on the MODD and MID are summarized in Tables 3 and 4. In terms of the sea edge detection performance, our model (PEN-SSM) and its two variants are superior to SSM and SSM(S). Among them, PEN-SSM achieves the best results, while SSM(S) is the worst. As we can see from the second

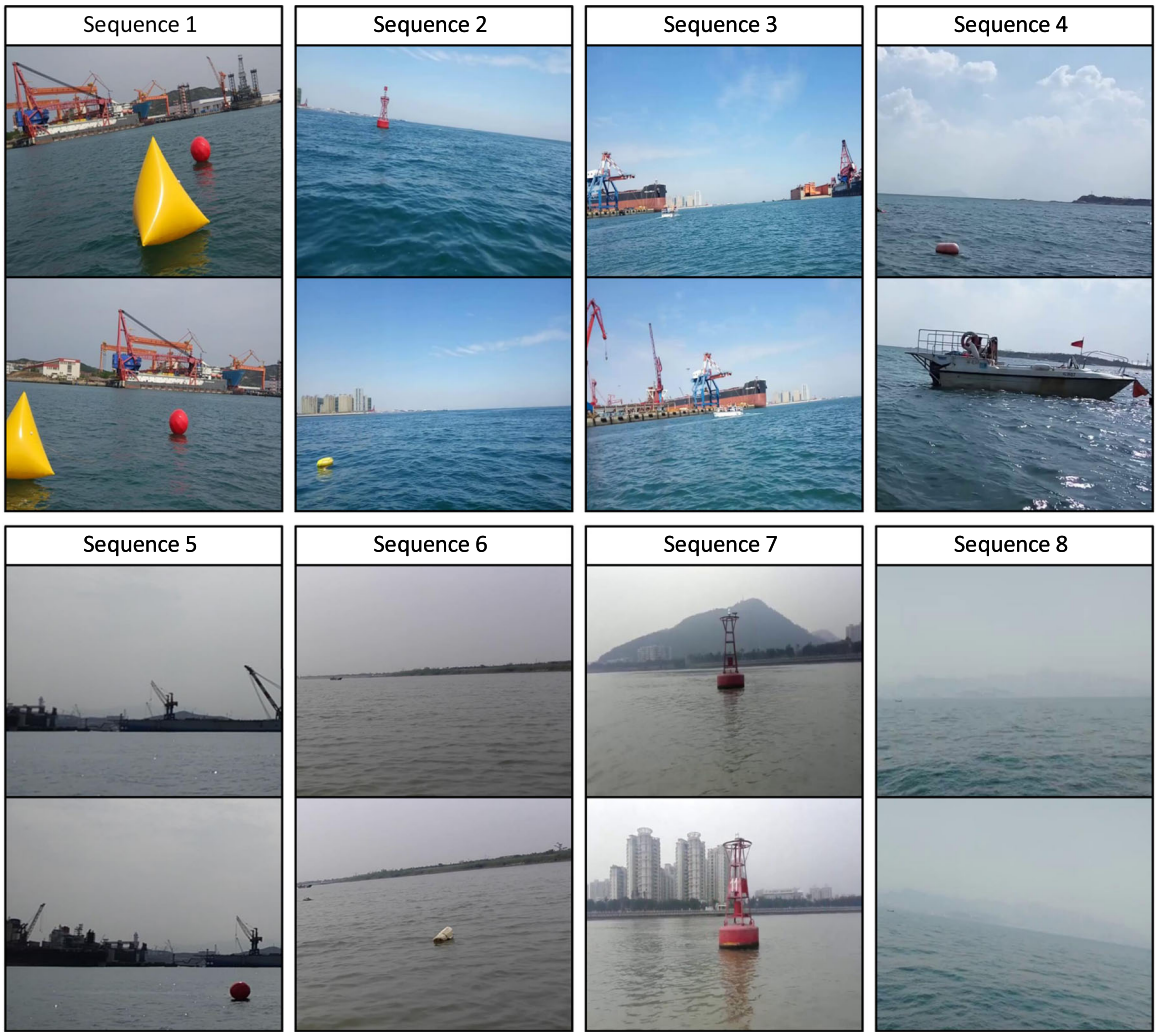


FIGURE 9 Images selected from the marine image data set (MID). The first frame in each video sequence is shown in the odd row [Color figure can be viewed at wileyonlinelibrary.com]

column of Tables 3 and 4, the proposed PEN-SSM outperforms the original SSM by 15% on the MODD and 69.5% on the MID. Compared with SSM(S), the sea edge detection performance of our approach on the MODD and MID increased by 39.3% and 71.4%, respectively. The comparison results show that our method is more reliable than the original SSM and its variant SSM(S) in the sea edge detection. The reason for this is that our proposed PEN can effectively learn the semantic structure of marine images from weakly annotated data and provide more accurate initial values for EM optimization, thus avoiding over- or under-segmentation of the water region when the boat is rolling and pitching significantly. From Figure 12d and Figure 13a–c, we can see that SSM and SSM(S) perform poorly in the sea edge detection when images violates their assumption of region splitting or there are oversized obstacles in the water. The results in Figure 13f show that our method performs better than SSM and SSM(S) under unfavorable weather conditions.

On the task of obstacle detection, our PEN-SSM achieves the highest precision and the lowest number of false positives per frame, along with the highest *F*-score. The excellent performance of our PEN-SSM is due to the fact that the initial Gaussian parameters obtained from PEN are more accurate in the presence of significant rolling and pitching than those calculated from three fixed regions predefined by SSM, and in suppressing sun glitter, the uniform component determined by PEN for modeling the outliers is more flexible than that of SSM. In our experiments, the results of SSM(S) are the worst on both the MODD and the MID, which indicates that SSM has poor detection performance for single-frame images. As seen from Table 3, the precision and recall of PEN-SSM on the MODD are 1% and 4.8% higher, respectively, than those of SSM, resulting in a 3.3% increase in the F-score. The gap between SSM and our PEN-SSM is greater on the MID. As shown in Table 4, compared with SSM, the precision of PEN-SSM increases by 52.8%, the recall rate increases by 24.9%, and the F-score increases by 39.2%. One reason for this is that the poor initial Gaussian parameters cause the EM procedure of SSM to fall into local minimum when detecting an image that violates its assumption of region splitting (see Figure 13a). Another reason is that the oversized obstacles under the sea edge make the initial Gaussian parameters of the water component inaccurate (see Figure 13c). As seen from

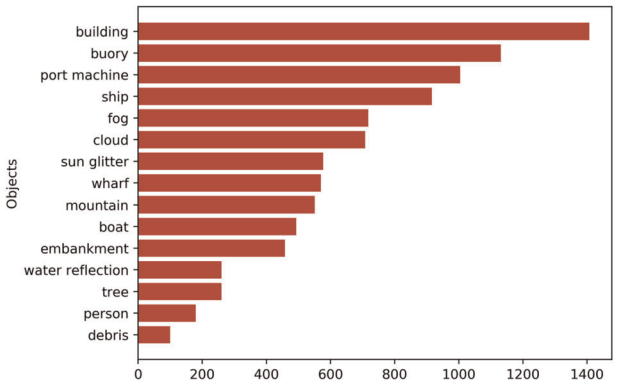


FIGURE 10 Number of images for different objects in the marine image data set [Color figure can be viewed at wileyonlinelibrary.com]

Tables 3 and 4, PEN-SSM(U) achieves the highest recall, but the false positive rate increases, resulting in a lower F-score than PEN-SSM. Moreover, we find that the F-score of PEN-SSM(U) on the MODD is close to that of PEN-SSM, but it is significantly lower than PEN-SSM on the MID. This proves that it is more robust to use the outlier component of the output of PEN for setting the values of the uniform distribution than to use a constant. Like SSM and SSM(S), PEN-SSM(U) is more prone to detecting sun glitter as obstacles than PEN-SSM (see Figures 12c and 13d), thus increasing the false positive. In terms of the performance of the sea edge and obstacle detection, PEN-SSM(P) and PEN-SSM have little difference, but the latter requires significantly less time than the former. This is because PEN-SSM directly applies the output of PEN as initial class priors to accelerate the convergence of the EM algorithm.

Figure 14 shows results of SSM and our PEN-SSM under different extreme conditions, including large highlighted areas, water reflections, heavy cloudy background and low illumination. From Figure 14a,b, it can be seen that SSM is susceptible to large highlight areas and water reflections, making the extent of the sea region significantly underestimated and causing false obstacle detection. In comparison, our method outperforms SSM under such conditions although it slightly underestimates the water region. From the results shown in Figure 14c,d, it can also be seen that SSM in heavy cloudy and low illumination conditions fails by grossly overestimating the water region because water region and land in those conditions

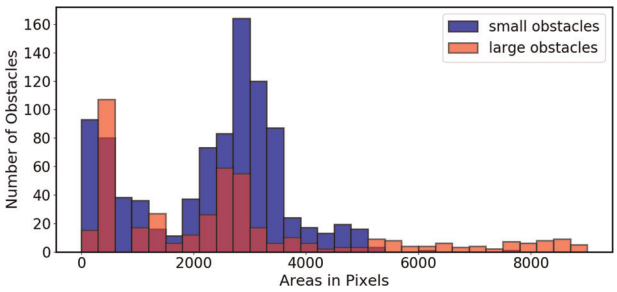


FIGURE 11 Distributions of bounding box areas of large and small obstacles [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 3 Results of various methods on the marine obstacle detection data set (MODD)

Method	μ_{edg} (σ_{edg})	*Prec*	*Rec*	*F*	αFP	t^*
PEN-SSM	**0.017 (0.008)**	**0.909**	0.809	**0.853**	**0.032**	19
PEN-SSM(P)	0.018 (0.009)	0.908	0.804	0.850	0.033	22
PEN-SSM(U)	0.017 (0.009)	0.891	**0.818**	0.851	0.059	19
SSM	0.020 (0.011)	0.900	0.772	0.826	0.050	10
SSM(S)	0.028 (0.015)	0.876	0.748	0.802	0.067	**9**

Note: The bold value represents the best score under each evaluation metric.

*t denotes the average processing time in ms.

are visually similar. Due to the accurate priors provided by PEN, our method is superior to SSM. To further assess whether there is a significant difference between PEN-SSM and SSM in sea edge estimation, we recorded the average RMSE values of the sea edge estimation of PEN-SSM and SSM on each video in the MODD and MID data sets, respectively, and obtained two sets of the sea edge errors. We then performed a Friedman test on the two sets of data and obtained a p value of 0.0017. Similarly, we also performed a Friedman test to the average F-scores of obstacle detection by PEN-SSM and SSM on each video in the MODD and MID data sets, and obtained a p value of 0.0253. Since both p values are less than 0.05, the results indicate that there were significant differences between PEN-SSM and SSM in sea edge estimation and obstacle detection.

The average processing time of our PEN-SSM per frame is 19 ms on the MODD and 21 ms on the MID. The computation and application of PEN decrease the speed of the algorithm by approximately 40%. Although the speed of PEN-SSM is slower than SSM, it does not slow the overall onboard control system in practice because the frame rate of the camera mounted on the USV does not exceed 15 fps.

4.2.4 | Failure cases

To gain further insight into the limitations of the proposed method, we show two addition failure cases in Figure 15.

TABLE 4 Results of various methods on the marine image data set (MID)

Method	μ_{edg} (σ_{edg})	*Prec*	*Rec*	*F*	αFP	t^*
PEN-SSM	**0.018 (0.017)**	**0.978**	0.798	**0.867**	**0.008**	21
PEN-SSM(P)	0.020 (0.018)	0.969	0.797	0.863	0.009	25
PEN-SSM(U)	0.018 (0.017)	0.759	**0.817**	0.771	0.082	21
SSM	0.059 (0.028)	0.739	0.639	0.623	0.105	11
SSM(S)	0.063 (0.027)	0.640	0.623	0.609	0.153	**10**

Note: The bold value represents the best score under each evaluation metric.

*t denotes the average processing time in ms.

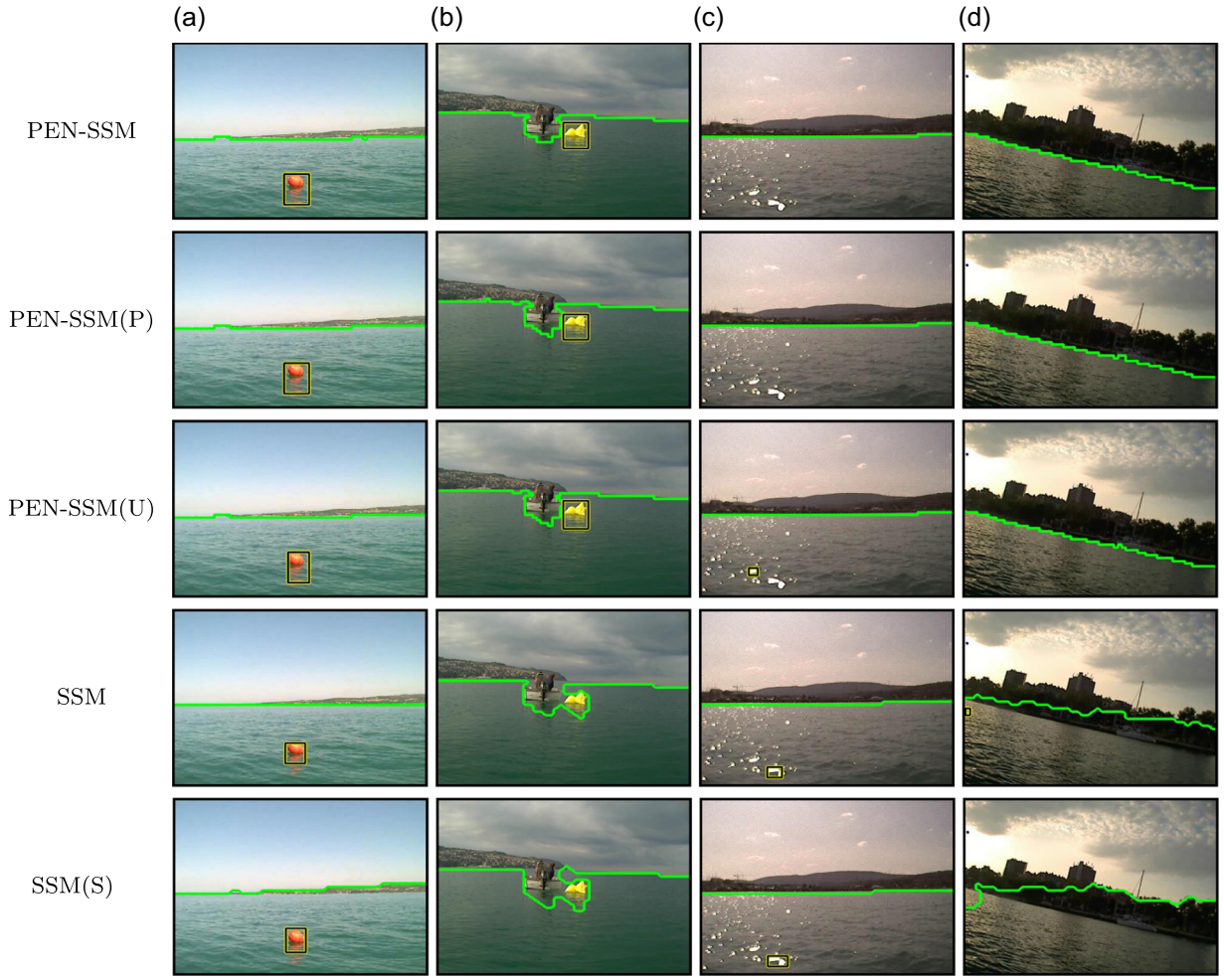


FIGURE 12 Example results of different methods on the MODD. The estimated sea edge is depicted by a green curve, while the obstacles in the water region are depicted by yellow rectangles. It can be seen that PEN-SSM is superior to SSM and SSM(S) in the presence of (a) small obstacle (b) cluster of objects, (c) sun glitter, and (d) significant rolling. MODD, marine obstacle detection data set; PEN, prior estimation network; SSM, semantic segmentation model [Color figure can be viewed at wileyonlinelibrary.com]

The left image of Figure 15 shows failures in obstacle detection due to the strong reflection of the red buoy. In this case, the buoy is over-segmentation, meaning that portions of water are misclassified as obstacle, but the true obstacles that threaten USVs are not mislabeled as water. A more dangerous case is shown in Figure 15b, where the left part of the boat is incorrectly detected as water because of visual similarity. Comparing the results with Figure 14, we can find that our method is better in distinguishing water region and lands than in distinguishing obstacles from water region. This is mainly because our PEN is trained by the pixels from the three dominant semantic regions, without using the pixels from obstacles.

The method proposed in this paper only uses a few of simple visual features to perform segmentation. In fact, there exists significant difference in texture between obstacles and their surroundings. In our future work, we will explore texture features or deep features extracted by CNNs to further improve the segmentation in the presence of strong reflection and obstacles visually similar to water.

5 | LESSONS LEARNED

The obstacle detection method presented in this paper can be readily applied to other surface ships to improve their autonomy. The rapidity and efficacy of the method is helpful for USVs to realize real-time collision avoidance and safely execute their tasks. For practical deployment of our obstacle detection system in a small USV, a few lessons were learned as follows.

The marine scenes captured by USVs are subject to sunlight conditions. Especially when the USV is directly facing the sun, it leads to extreme changes in the contrast and color of images and even causes a significant amount of fragmented glitter or bright areas on the water surface. To improve the generalization of our PEN for the enhancement of obstacle detection, it would be advantageous to collect a more diverse training data set that covers a range of realistic conditions encountered in a coastal surveillance task.

As described in Section 1, camera-based obstacle detection methods can detect not only objects that stand out from the water but also flat objects. However, such methods still have difficulties in

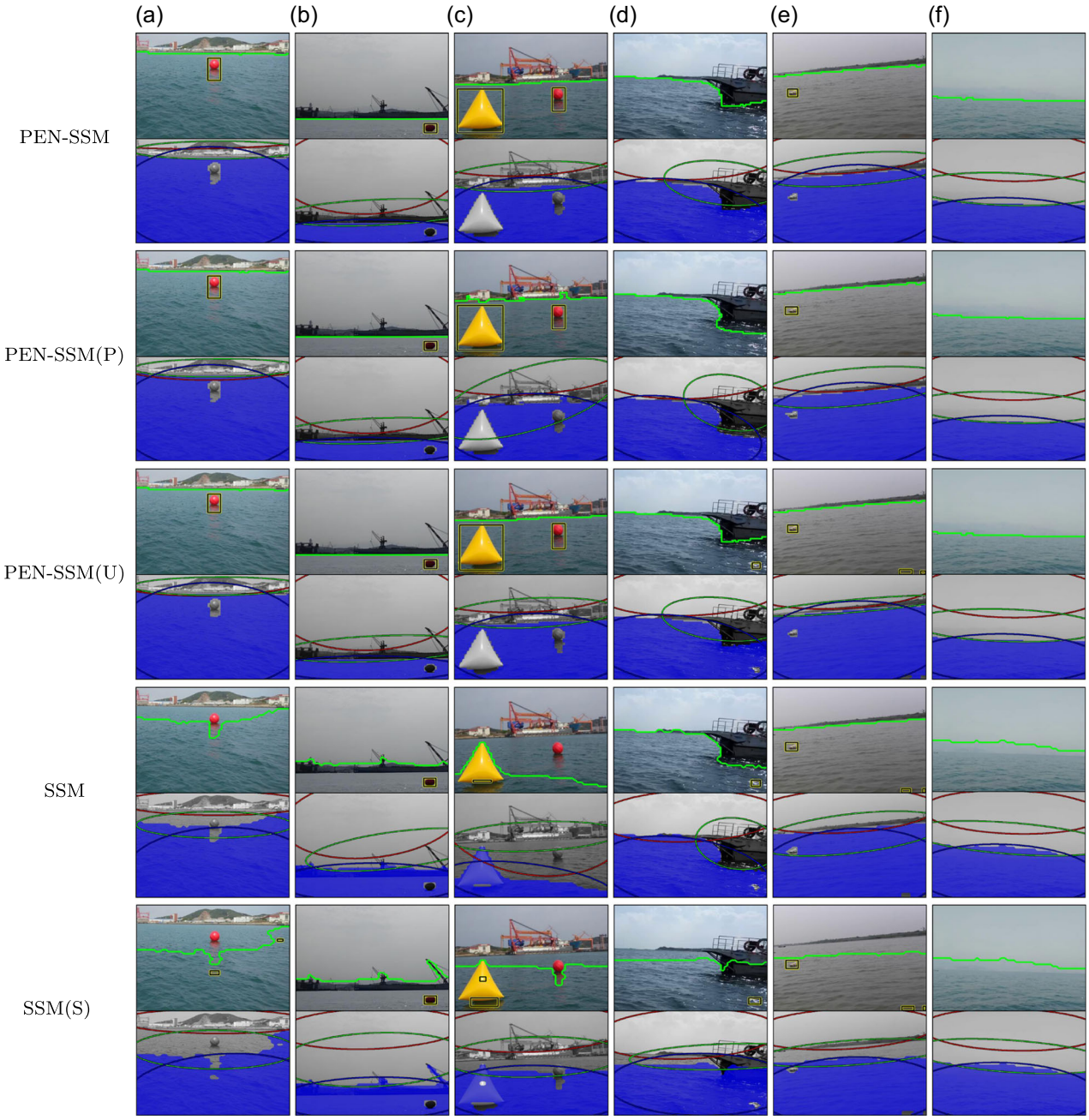


FIGURE 13 Example results of different methods on the MID data set. For each image, we show the estimated sea edge by a green curve and the detected obstacle in the water region by a yellow rectangle. Below each image, the spatial part of the three semantic components is depicted by three Gaussian ellipses, and the segmented water region is shown in blue. In terms of obstacle detection and sea edge estimation, it can be seen that PEN-SSM outperforms SSM and SSM(S) in the presence of (a)–(b) significant pitching, (c) oversized obstacles, (d) sun glitter, (e) salient waves, and (f) visual blur caused by fog. MID, marine image data set; PEN, prior estimation network; SSM, semantic segmentation model [Color figure can be viewed at wileyonlinelibrary.com]

detecting potential underwater obstacles for USVs, such as submerged reefs and sunken ships. In addition, our current method only works in the daytime. Further work is required to explore multisensor fusion and expand the application to USVs at night.

The mixture model in the proposed framework only applies color and position features to fit the semantic structure of marine environment and perform semantic segmentation. The experimental results show that the model is sufficiently general to be directly applied in most realistic marine scenes. However, it has limitations in distinguishing obstacles and their strong reflections. Deep features extracted by CNNs need to be explored to improve the segmentation performance of the mixture model.

The application of the obstacle detection system in small USVs is constrained by sea state. In fact, the camera mounted on a small coastal USV may only capture the sky or water area under severe sea conditions, resulting in the failure of obstacle detection. Installing the

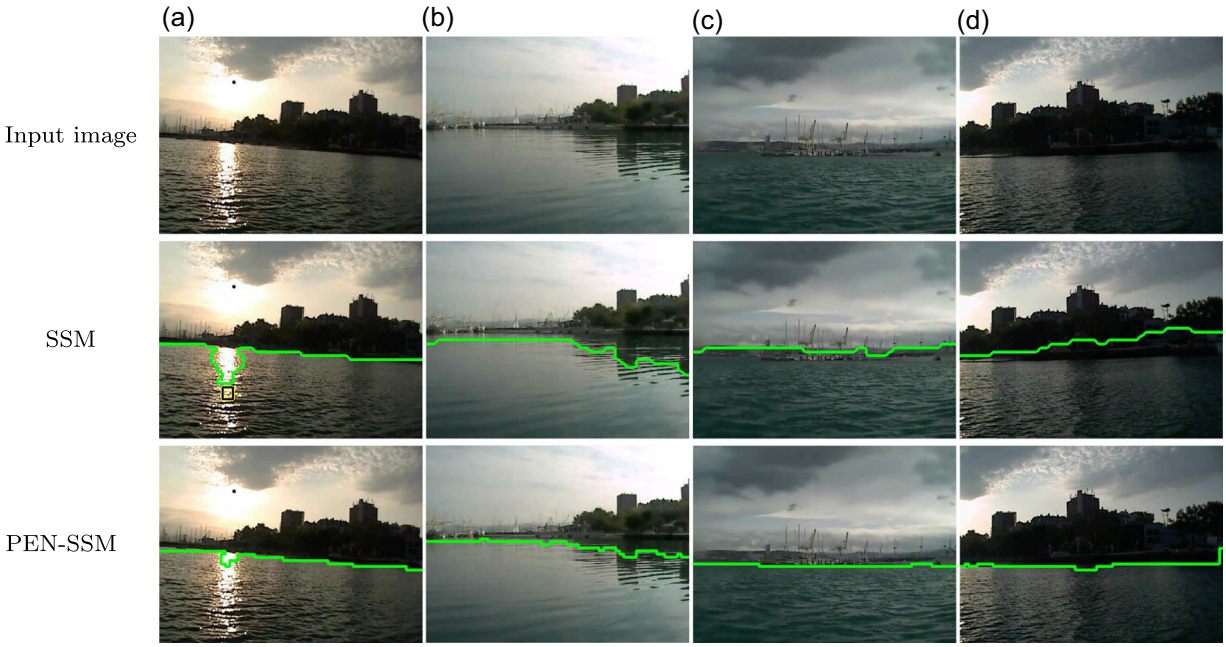
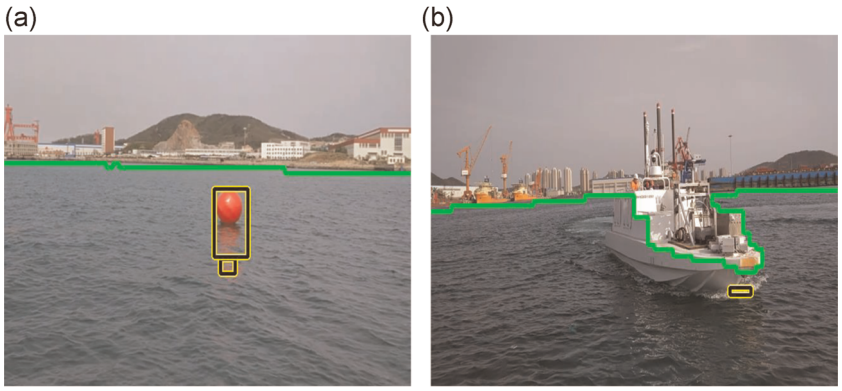


FIGURE 14 Example results of PEN-SSM and SSM under different extreme conditions. The estimated sea edge is depicted by a green curve, while the obstacles in the water region are depicted by yellow rectangles. It can be seen that PEN-SSM is superior to SSM in the presence of (a) large highlighted area in the water, (b) water reflections, (c) heavy cloudy, and (d) low illumination. PEN, prior estimation network; SSM, semantic segmentation model [Color figure can be viewed at wileyonlinelibrary.com]

FIGURE 15 Examples of poor segmentation of our method due to (a) strong and salient reflection of the obstacle and (b) obstacles visually similar to water [Color figure can be viewed at wileyonlinelibrary.com]



camera on a stable gimbal could help mitigate this problem and ensure that proper images are captured.

6 | CONCLUSION

In this paper, we introduced a novel semantic segmentation algorithm based on a spatially constrained mixture model for real-time obstacle detection in marine environments. To avoid the EM procedure falling into a local minimum due to poor initial parameters, we proposed a PEN for the initialization of model parameters. The network is trained by a new weakly supervised E-step. A series of detailed experiments were conducted on a public MODD and our data set (MID) to evaluate our method. Compared to the recent state-of-the-art SSM (Kristan et al., 2016), our algorithm achieved 15% improvement in sea edge estimation and a 3.3% increase in *F*-score on the MODD, and achieved 69.5% improvement in sea edge estimation and a 39.2% increase in *F*-score on our data set.

Our future work will focus on exploring texture features or deep features extracted by CNNs to further improve segmentation in the presence of strong reflection and confounding obstacles. Our future work will also involve obstacle detection based on multisensor fusion to expand the application of USVs at night, and apply the detection results as feedback for collision avoidance.

ACKNOWLEDGMENTS

The authors would like to thank Matej Kristan for sharing the MODD data set and Xuetao Zhang for providing the source code of FSSNet. The work described in this paper was supported by the National Natural Science Foundation of China(Nos 61933008, 61525305, and

61827812) and Key Research and Development Project of Jiangxi Province of China(No. 20192BBEL50004).

ORCID

Jingyi Liu https://orcid.org/0000-0001-6251-6051
Hengyu Li http://orcid.org/0000-0002-2243-5908

REFERENCES

Alvarez, J. M., LeCun, Y., Gevers, T., & Lopez, A. M. (2012). Semantic road segmentation via multi-scale ensembles of learned features. In Proceedings of the 12th European Conference on Computer Vision (ECCV) (pp. 586-595). Florence, Italy.

Benjamin, M. R., Leonard, J. J., Curcio, J. A., & Newman, P. M. (2006). A method for protocol-based collision avoidance between autonomous marine surface craft. *Journal of Field Robotics*, *23*(5), 333-346.

Bloisi, D. D., Previtali, F., Pennisi, A., Nardi, D., & Fiorini, M. (2016). Enhancing automatic maritime surveillance systems with visual information. *IEEE Transactions on Intelligent Transportation Systems*, *18*(4), 824-833.

Borovicka, T., Jirina, M., Jr., Kordik, P., & Jirina, M. (2012). Selecting representative data sets. In A. Karahoca (ed) *Advances in Data Mining Knowledge Discovery and Applications* (pp. 43-70). London: IntechOpen.

Bovcon, B., & Kristan, M. (2018). Obstacle detection for usvs by joint stereo-view semantic segmentation. *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5807-5812). Madrid, Spain

Bovcon, B., Mandeljc, R., Perš, J., & Kristan, M. (2018). Stereo obstacle detection for unmanned surface vehicles by imu-assisted semantic segmentation. *Robotics and Autonomous Systems*, *104*, 1-13.

Bovcon, B., Perš, J., Kristan, M. (2017). Improving vision-based obstacle detection on usv using inertial sensor. *In Proceedings of the 10th International Symposium on Image and Signal Processing and Analysis (ISPA)* (pp.1–6). Ljubljana, Slovenia

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *40*(4), 834-848.

Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., & Urtasun, R. (2016). Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 2147-2156). Las Vegas, NV.

Chen, X., Ma, H., Wan, J., Li, B., & Xia, T. (2017) Multi-view 3d object detection network for autonomous driving. In *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1907-1915). Honolulu, HI.

Diplaros, A., Vlassis, N., & Gevers, T. (2007). A spatially constrained generative model and an em algorithm for image segmentation. *IEEE Transactions on Neural Networks*, *18*(3), 798-808.

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., & Lu, H. (2019). Dual attention network for scene segmentation. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3146-3154). Long Beach, CA

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 1026-1034). Santiago, Chile.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition* (CVPR) (pp. 770-778). Vegas, NV.

He, S., Wang, M., Dai, S.-L., & Luo, F. (2019). Leader-follower formation control of usvs with prescribed performance and collision avoidance. *IEEE Transactions on Industrial Informatics*. *15*(1), 572–581.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *In Proceedings of the 32nd International Conference on Machine Learning (ICML)* (pp. 448-456). Lile, France.

Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint, arXiv:1412.6980*.

Kristan, M., Kenk, V. S., Kovačič, S., & Perš, J. (2016). Fast image-based obstacle detection from unmanned surface vehicles. *IEEE Transactions on Cybernetics*, *46*(3), 641-654.

Lee, Y., Kim, H., Park, E., Cui, X., & Kim, H. (2017) Wide-residual-inception networks for real-time object detection. *In Proceedings of the 28th IEEE Intelligent Vehicles Symposium (IV)* (pp. 758-764). Redondo Beach, CA.

Levi, D., Garnett, N., Fetaya, E., & Herzlyia, I. (2015). Stixelnet: A deep convolutional network for obstacle detection and road segmentation. In Proceedings of the 26th British Machine Vision Conference (BMVC) (pp. 109.1-109.12). Swansea, UK.

Li, S. Z. (2009). *Markov random field modeling in image analysis*, London: Springer Science & Business Media.

Liu, Z., Zhang, Y., Yu, X., & Yuan, C. (2016). Unmanned surface vehicles: An overview of developments and challenges. *Annual Reviews in Control*, *41*, 71-93.

Moon, T. K. (1996). The expectation-maximization algorithm. *IEEE Signal processing magazine*, *13*(6), 47-60.

Oliveira, G. L., Burgard, W., & Brox, T. (2016). Efficient deep models for monocular road segmentation. *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4885–4891). Daejeon, Korea.

Paszke, A., Chaurasia, A., Kim, S., & Culurciello, E. (2016). Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147.

Prasad, D. K., Rajan, D., Rachmawati, L., Rajabally, E., & Quek, C. (2017). Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey. *IEEE Transactions on Intelligent Transportation Systems*, *18*(8), 1993-2016.

Shi, J., Jin, J., & Zhang, J. (2018). Object detection based on saliency and sea-sky line for usv vision. *In Proceedings of the 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)* (pp.1581-1586). Chongqing, China.

Sivaraman, S., & Trivedi, M. M. (2013). Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE Transactions on Intelligent Transportation Systems*, *14*(4), 1773-1795.

Tian, Z., He, T., Shen, C., & Yan, Y. (2019). Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3126-3135). Long Beach, CA.

Wang, H., Wei, Z., Ow, C. S., Ho, K. T., Feng, B., & Huang, J. (2012). Improvement in real-time obstacle detection system for USV. In *2012 12th International Conference on* Control Automation Robotics & Vision (ICARCV) (pp.1317-1322). IEEE.

Wang, H., Wei, Z., Wang, S., Ow, C. S., Ho, K. T., & Feng, B. (2011). A vision-based obstacle detection system for unmanned surface vehicle. *In Proceedings of the 2011 IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM)* (pp. 364-369). Qingdao, China.

Wang, H., Wei, Z., Wang, S., Ow, C. S., Ho, K. T., Feng, B., & Lubing, Z. (2011). Real-time obstacle detection for unmanned surface vehicle. *In Proceedings of the 2011 Defense Science Research Conference and Expo (DSR)* (pp.1-4). Singapore, Singapore.

Yang, M. S., Lai, C. Y., & Lin, C. Y. (2012). A robust em clustering algorithm for gaussian mixture models. *Pattern Recognition*, *45*(11), 3950-3961.

Zhang, R., Tang, P., Su, Y., Li, X., Yang, G., & Shi, C. (2014). An adaptive obstacle avoidance algorithm for unmanned surface vehicle in

complicated marine environments. *IEEE/CAA Journal of Automatica Sinica*, *1*(4), 385-396.
Zhang, X., Chen, Z., Wu, Q. J., Cai, L., Lu, D., & Li, X. (2019). Fast semantic segmentation for scene perception. *IEEE Transactions on Industrial Informatics*, *15*(2), 1183-1192.
Zhang, Y., Li, Q.-Z., & Zang, F.-N. (2017). Ship detection for visual maritime surveillance from non-stationary platforms. *Ocean Engineering*, *141*, 53-63.
Zhao, Y., Li, W., & Shi, P. (2016). A real-time collision avoidance learning system for unmanned surface vessels. *Neurocomputing*, *182*, 255-266.
Zheng, Z., & Feroskhan, M. (2017). Path following of a surface vessel with prescribed performance in the presence of input saturation and external disturbances. *IEEE/ASME Transactions on Mechatronics*, *22*(6), 2564-2575.

SUPPORTING INFORMATION

Additional Supporting Information may be found online in the supporting information tab for this article.

How to cite this article: Liu J, Li H, Luo J, Xie S, Sun Y. Efficient obstacle detection based on prior estimation network and spatially constrained mixture model for unmanned surface vehicles. *J Field Robotics*. 2021;38: 212–228. https://doi.org/10.1002/rob.21983

APPENDIX A

TABLE A1 Index to multimedia extensions

Extension	Media type	Description
1	Video	It shows the experimental results in the presence of significant rolling and pitching
2	Video	It shows the experimental results in the presence of fragmented sun glitter
3	Video	It shows the experimental results in the presence of visual ambiguities caused by the unfavorable weather condition