NANYANG
TECHNOLOGICAL
UNIVERSITY

VISION BASED OBSTACLE DETECTION AND MAPPING FOR UNMANNED SURFACE VEHICLES

XIAOZHENG MOU

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

2018

VISION BASED OBSTACLE DETECTION AND MAPPING FOR UNMANNED SURFACE VEHICLES

XIAOZHENG MOU

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirements for the degree of
Doctor of Philosophy

2018

Abstract

An unmanned surface vehicle (USV) is a speed boat that comes with a suit of sensors to understand maritime environment and navigational intelligence to know how to navigate itself autonomously. In this thesis, we focus on exploring and enhancing the perception ability of USV based on vision.

Detecting and mapping obstacles in maritime environments with high accuracy and robustness are two important issues for USV in real-life applications, such as surveillance and navigation. Solutions based on vision (cameras) are more cost-effective than its counter-part, radar. What's more, vision can compensate the blind area of radar at short distances.

This thesis addresses the tasks of improving the accuracy and robustness of vision based obstacle detection and mapping in open sea.

For obstacle detection, first, we present a novel image-based algorithm, in that the obstacle patches are separated from sea patches using the proposed patch distinctiveness measure, named as global sparsity potentials. Secondly, we develop a real-time long range obstacle detection and tracking system for USV based on binocular stereo vision, which improves the image-based methods in reducing false positives caused by noise (white wake, waves, sun reflection, etc.) and obtaining additional information of the detected obstacles' distances. In this system, we propose to implement the obstacle detection algorithm in an image-pyramid manner to speed up the processing, and we also propose a new solution for multiple-obstacle tracking with scale adapting and occlusion handling. Then, an approach by fusing 2D and 3D clues for further enhancing the performance of obstacle detection is proposed in the same binocular vision system. In this approach, the 2D and 3D information are combined in a weighting model, which gives more weights to the 2D detecting results when obstacles are distant, while gives more weights to the 3D detecting results when obstacles are nearby.

After detecting the obstacles, it is necessary to build the obstacle map for the navigation of USV. To this end, we develop a system for obstacle mapping based on motion stereo vision, which raises the ranging ability from 500 meters in the previous binocular stereo system to 1,000 meters with a even larger baseline obtained from

the travelling of USV. Moreover, the proposed motion stereo system eliminates the complicated calibration work and the bulky rig in binocular stereo. Integrating monocular camera with GPS and compass information in this proposed system, the world locations of the detected static obstacles are reconstructed when the USV is travelling, and finally an obstacle map is built. To achieve more accurate and robust performance, multiple pairs of frames are leveraged to synthesize the final reconstruction results in a weighting model. To the best of our knowledge, we are the first to address the task of monocular vision based obstacle mapping in maritime environment.

Since there are few available public datasets for this research, we evaluate the efficiencies of the proposed algorithms on our own datasets. Experimental results verify that our methods are highly accurate and robust as compared to other methods.

The thesis concludes with discussions to the presented research, and with suggestions to further studies in this field.

Acknowledgements

I have got a lot of help, supports and encouragements from a number of people for the completion of this thesis.

First of all, I would like to thank my supervisor Associate Professor Wang Han, who made it possible for me to come to NTU to pursue my PhD degree in his research group. Moreover, Prof. Wang gave me many advices and instructions on how to do research and what research topic to choose. I am deeply influenced by his passion for research and his patience for teaching, and some ideas of my research were inspired by the discussion with him. I cannot make my research go smoothly without the support of Prof. Wang. Therefore, I give a lot of thanks to Prof. Wang for his selfless help and support.

I feel very fortunate to work with many warm-hearted and friendly colleagues. I had many valuable discussions with Shin Bok-Suk, Yang Shuai, Yuan Shenghai, Mou Wei, Lim Kart-Leong, Zhang Handuo and Jiang Rui for my research. Yuan Shenghai helped to build up the binocular stereo rig, in which the synchronization of cameras were made by him. Mou Wei, Yuan Shenghai and Soner Ulun did many help in the stereo camera calibration. Shin Bok-Suk and Yang Shuai helped a lot in data collection on the sea. So, I thank all of them for their kind help and enlightening discussions.

Finally, I thank my parents for their endless love, support and encouragement, where I get courage and confidence to persist in my ideal.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

An unmanned surface vehicle (USV) is a speed boat that comes with a suit of sensors to understand maritime environment and navigational intelligence to know how to navigate itself autonomously. Detecting and mapping obstacles in maritime environments with high accuracy, speed and robustness is very important for USV in real-life applications, such as sea surveillance and USV self-navigation. According to different sensors used, researches in this topic can be classified as two groups: active detection and passive detection. Radar is the mainly used sensor in active detection [5, 6, 7], and it is efficient for long range obstacle detection, but has a blind area in short range. For instance, the FURUNO marine radar with models of DRS4W, FAR-1523BB, MODEL1715, etc. have a minimum detection range of 0.125 nautical miles (231.5 meters) [8]. Passive detection mostly uses camera sensors, which are able to perceive visible obstacles and are more cost-effective, so this vision-based detection is expected to make up the deficiency of radar-based methods in detecting short range obstacles. Many research in this line have arisen in the last two decades [9], nevertheless, it is still an ongoing challenge for vision-based obstacle detection to be commercially applied in USV due to the limitations in detecting speed, accuracy and robustness. For instance, obstacle detection in a single 2D image always suffers from the false positives caused by noise (white wake, waves, sun reflections, etc.), and detection in stereo images has the problem of processing extensive data, which makes it hard for real-time application. After detecting the obstacles, it is necessary to build the obstacle map for the navigation of USV. However, there are very few

literatures studying obstacle mapping in maritime environment based on monocular vision. Here we first attempt to explore this issue. Thus, in this thesis, we focus on addressing the vision-based obstacle detection and mapping in the scene of open sea for USV.

We study three categories of approaches to improve the obstacle detection and ranging ability for the USV, step by step. The first two categories are image-based method and binocular stereo, and three approaches for obstacle detection are proposed based on them. The third category is motion stereo, based on which one obstacle mapping approach is proposed.

1.1 Image-Based Method

Image-based method for obstacle detection is to process the image or video data from a monocular camera sensor, which is both cheap and easy to be setup.

As a first attempt, we introduced an image-based obstacle detection method via computing the proposed global sparsity potentials (GSP) of sampled image patches, which captures the sparseness or similarity rate of an image patch throughout the sea area. In this approach, image patches with smaller GSP value are considered as the main cluster (sea water), while their outliers, which have larger GSP value and larger Mahalanobis distance to the mean feature of the sea water, are taken as obstacles. Experimental results prove that this proposed approach improves the detecting accuracy compared to the traditional feature space reclustering method [2] and a state-of-the-art saliency detection method [3].

Although image-based method is the most convenient and economic way, and also higher processing speed can be achieved due to less data to be processed, it is always sensitive to the noise or outliers, such at white wake, waves, sun reflection, etc., and thus results in many false positives. Another deficiency is that the depth of detected obstacles can not be obtained.

Figure 1.1: Stereo rig mounted on the USV.

1.2 Binocular Stereo Based System

Challenges of visual obstacle detection in image-based methods basically come from two kinds of sources. One is the foreground, the obstacles, which have a lot of appearances varying from cargo ships, vessels, yachts, to small buoys. In addition, the appearance and size of one certain obstacle would render great variability when the USV was rotating or moving; thereby it is hard to train a model with machine learning methods to detect the obstacles due to this large number of appearance uncertainties. The other source caused challenge is the background where the noise reside, for instance, noise like white wake, waves, water speckles, and clouds have high probabilities to be wrongly detected as obstacles, because their appearances are much distinct compared with the major background (sea surface and sky).

Considering the above mentioned issues, and for better understanding the scene by USV, in this thesis, we proposed a binocular stereo vision system with large baseline, which is shown in Figure 1.1. The two cameras mounted on two sides of the structure form a stereo vision, through which the 3D point cloud can be reconstructed by the left and right 2D images. Therefore, we have not only the appearance information from 2D image, but also the 3D information from 3D point cloud. Based on this binocular stereo vision system, we proposed two approaches to detect obstacles in the open sea for USV.

First, we developed a real time long range obstacle detection and tracking approach based on both the reconstructed point cloud and 2D images from our stereo vision architecture. At the same time, we estimate accurately distances as an es-

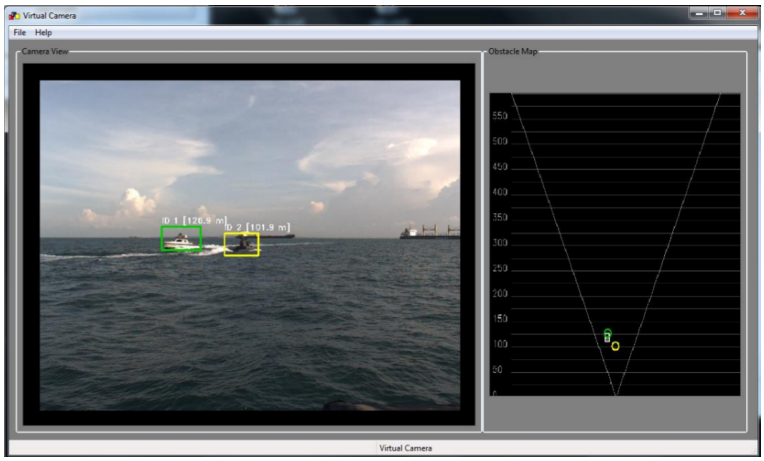


Figure 1.2: Developed GUI for obstacle detection and tracking in our USV.

sential property of detected objects of interest. In this approach, we estimate the sea surface plane using random sample consensus (RANSAC) on point cloud reconstructed from lower resolution images, which contains less noise from obstacles; The obstacles are detected on point cloud reconstructed from middle resolution images via the Bayesian plan-view map; Finally, obstacle tracking and distance computing is preformed in the original image (high resolution). Moreover, we improved the obstacle tracking to adapt the scenarios of multiple obstacles, scale changing, and occlusion happening. Figure 1.2 shows the GUI developed for this system. One can see that the identified obstacles in image are marked by rectangle bounding boxes, on top of which their labels and distances are shown. Moreover, the detected obstacles are also plotted in a plan-view map on the right side for better visualization.

Then, to further improve the detecting accuracy, we presented a new method that fuses 2D and 3D information. With 2D clue, we can detect all salient candidate obstacles in images no matter how far they are, however, noise might be included in the results, like white wake and water speckles. With 3D clue, we can detect obstacles in short range by clustering points protrude from the sea surface, though this results in less noise, obstacles far away are hard to be detected, because their depth may not be valid. Since 2D and 3D can compensate the drawbacks of each other, we propose to combine the 2D and 3D clues in a weighting model, which

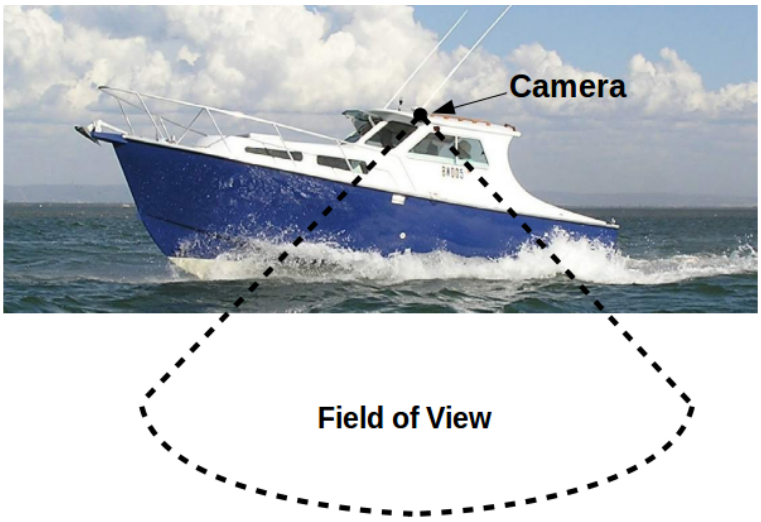


Figure 1.3: Proposed monocular camera mounted on an USV.

gives more weights to the 2D detecting results when obstacles are far away, while gives more weights to the 3D detecting results when obstacles are nearby. Therefore, by leveraging both 2D and 3D clues, more accurate and robust obstacle detecting results can be achieved compared to the conventional methods.

1.3 Motion Stereo Based System

Based on the above binocular stereo with wide baseline, we have achieved obstacle detection, tracking and ranging ability within 500 meters. However, wide baseline means a large structure on the boat, and this has big problem to calibrate. Moreover, binocular HD cameras means extensive data to be processed. To further enlarge the ranging ability within 1000 meters of our USV, it is really hard to continue developing the binocular stereo vision system. Therefore, we proposed a monocular vision based system, in which we use only one camera, and restore the depth via the USV's own motion. Figure 1.3 illustrates the position of our monocular camera mounted on an USV. The camera is facing the side of the USV.

In this system, the USV is travelling at fast speed. The sensors include a single digital camera, a GPS device and a compass. According to the theory of motion parallax, the depth of a feature point from a same stationary obstacle in real world

appear in two frames of the moving camera can be computed via the knowledge of camera translation and rotation. In our method, the translation is calculated from recorded GPS of each frame, and in rotation, the roll and pitch angles are derived from the detected horizon line, while the yaw angle is rendered by a compass or IMU. We can simultaneously build an obstacle map with range of 1000 meters when the USV is travelling.

To the best of our knowledge, we are the first to tackle vision based obstacle mapping in maritime environment. Besides enlarges the ranging ability, our proposed system is very convenient to be setup and just needs the calibration for intrinsic parameters of a single camera using a conventional checker board method. Nevertheless, it also has some limitations, including 1) Only stationary obstacles can be reliably mapped; 2) The USV must be in travelling; 3) The mapping accuracy is sensitive to various measurement noise, such as GPS, compass/IMU, horizon detection and feature matching.

1.4 Our Contributions

Table 1.1 summarised the pros and cons of each approach category presented in this thesis. In Table 1.1, one can see that our proposed vision based obstacle detection and mapping system for USV is improved step by step. Our main contributions in this thesis include

1) To improve the accuracy of obstacle detection in 2D maritime images, a new measurement GSP for evaluating the similarities of image patches is proposed, and it is then utilised in our proposed new image-based method for obstacle detection in maritime scenes.

2) To reduce the false positives caused by noise in image-based methods and know the depth of detected obstacles, a real-time processing framework for long range (within 500m) obstacle detection and tracking based on a binocular stereo vision system with large baseline is proposed. In this framework, we propose an image-pyramid approach to accelerate processing speed, and also by fusing 3D information, an improved solution for multiple-obstacle tracking with scale adapting and occlusion handling is proposed.

3) To further improve the detecting accuracy based on the binocular stereo system, a novel obstacle detection approach that combines the detected candidate obstacles from 2D and 3D modalities is proposed. This approach fuses the advantages of the above two presented methods (image-based and binocular stereo) to achieve a better performance. Furthermore, in this approach, we propose a new method for horizon detection that uses the prior knowledge of vanishing line estimated from sparse stereo matching to obtain a more accurate horizon line.

4) To further enlarge the ability of obstacle ranging within 1000 meters, we get rid of the bulky binocular stereo rig together with its difficult calibration, and we propose and develop a monocular vision based system for obstacle mapping in maritime environment using motion stereo. Moreover, an approach that leverage multiple frame pairs to compute the final reconstructed feature points is proposed to achieve more accurate and robust performance.

1.5 Outline of the Thesis

The rest of the thesis is structured as follows. Chapter 2 reviews the related work; Chapter 3 proposes an image patch similarity measure to improve image-based obstacle detection in maritime scenes. Chapter 4 presents our binocular stereo vision based system in USV for obstacle detection and tracking in real time. Chapter 5 presents our proposed method to enhance the performance of visual obstacle detection in open sea by fusing 2D and 3D clues. Chapter 6 describes our proposed approach for building obstacle map in the sea environment based on motion stereo. Experimental results with our own datasets are presented, and evaluations and comparisons with other related work are made in each of their corresponding chapters. Finally, conclusions are drawn, and future work are discussed in Chapter 7.

Table 1.1: Comparison of three categories of approaches presented in this thesis.

Categories	**Pros:**	**Cons:**
Image-based methods	• Higher processing speed • Convenient and economic	• More false positives caused by noise • Depth of obstacles can not be obtained
Binocular stereo	• Less false positives caused by noise • Depth of obstacles in near-field can be obtained with high accuracy	• Bulky stereo rig and hard to calibrate • Obstacle detection is limited in range $[50m, 500m]$ • Extensive data to be processed
Motion stereo	• Convenient and simple calibration for a single camera • Obstacle ranging ability is enlarged to 1,000 meters • Less data to be processed	• Only stationary obstacles can be mapped, but not for moving ones • The USV must be in travelling • Sensitive to various measurement noise

Chapter 2

Literature Review

Vision based obstacle detection in maritime environments has been extensively studied, and a variety of solutions to this task have been published in recent years. We reviewed the image-based methods in a monocular vision system and the 3D point cloud based methods in a binocular stereo vision system for this task. Since there are very few literatures studying vision based obstacle mapping for USV, we only reviewed the well-known obstacle mapping approaches for unmanned ground vehicles (UGV) and unmanned aerial vehicles (UAV) with the motion stereo system.

2.1 Image-Based Methods

Image-based methods refers to processing the images or videos captured from a monocular camera. Many researches on obstacle detection in maritime environment have been done with the image-based methods, which include basic image processing, background subtraction, saliency detection, Gaussian mixture model, feature space clustering and machine learning. In the following part, we have done a detailed review for each category of these methods.

2.1.1 Basic Image Processing

Basic image processing (e.g. binarization [10], edge detection [11], contour detection [12], etc.) is the most fast and direct way to segment the foreground (obstacles) from

the background in an image of maritime environment, since the sea surface basically appears uniform to itself but different from the obstacles.

For instance, one can binarize an image with a threshold value to partition it into one or more obstacle regions and one background region. As in [13], the obstacles close to the horizon line detected with Hough transform [14] are segmented using color-gradient filter [15] followed by Otsu thresholding [10]. Moreover, [16] uses adaptive threshold on saturation image to highlight the salient regions which are thereby segmented as obstacles. Another image processing technique to detect the obstacles is based on edge detection, which has been applied in [17, 18]. Their common processing steps generally consist of extracting Canny edges [11], post-processing, and filtering out obstacles.

The basic image processing methods are quite sensitive to the illumination changing and outliers in image. For example, the sun reflection and the white foam generated by a fast moving boat in an image of maritime scene are very easy to be wrongly detected as objects, because these outliers usually have strong contrast to the background.

2.1.2 Background Subtraction

Background subtraction is a method extensively used for detecting moving objects in videos from static cameras. It can segment the moving foreground from the static background with two main steps: initialization and updating. In initialization step, the first background model is generated from several frames. In updating step, the background model is updated by incorporating the information of changes of observed scenes.

In [19], the background subtraction methods of mixture of Gaussians (MOG) [20, 21] and visual background extractor (ViBe) [22] were applied and compared in their proposed hybrid approach for boat detection in maritime surveillance. Another hybrid method that integrates background subtraction [23] and color segmentation [24] was proposed in [25] to detect objects for marine surveillance. Furthermore, authors of [26] applied the background model of [27] as a preprocessing to segment the foreground, and then exploited level set with shape priors [28] to detect and

track moving objects in a maritime environment.

In general, the methods of background subtraction are restricted to the applications that use static camera. Moreover, it can only detect the moving objects in the static background. Such natures of background subtraction limit its application.

2.1.3 Saliency Detection

Humans are able to detect visually distinctive, so called salient, scene regions effortlessly and rapidly (i.e., pre-attentive stage). These filtered regions are then perceived and processed in finer details for the extraction of richer high-level information (i.e., attentive stage) [29]. Based on such fact, saliency detection methods as a preprocessing or final-processing for object detection has been widely studied in recent years. Generally, saliency detection consists two stages of process: 1) Salient region detection; 2) Accurate object region segmentation.

Assuming that the obstacles in an image are more salient or catchier than the background, saliency detection is a widely used method in maritime obstacle detection. Recently, in the work of [30], obstacles are detected by adaptive hysteresis thresholding of a saliency map generated with a modified boolean map saliency (BMS) [31] approach. The work of [32] obtains constrains of object shape and confidence maps from the saliency detector BMS, and then forwards the two constrains to a robust principal component analysis (RPCA) [33] approach to enhance the detecting accuracy. Authors of [34] integrated objectness [35] and saliency detection [36] to get a fast object detector for USV. Furthermore, saliency detection [37, 38] is applied in a static camera case of [19] as a compensation for background subtraction, because background subtraction can only detect moving obstacles, thus the static obstacles can be detected as well by combining saliency detection.

One drawback of saliency detection in maritime environment is that the noise like white wake and sun reflection are salient as well, and hard to be differentiated from obstacles in a resulted saliency map. [39] exploits optical flow [40] to track the features in salient regions and then estimate their motion, thus knowing this cue, some noise can be excluded, since meaningful obstacles are supposed to have continuous trajectory while noise merely appear in the field of view for a short period

of time.

2.1.4 Gaussian Mixture Model

Gaussian mixture model (GMM) [41] is an efficient approach for image segmentation. In [42] and [43], the background of a pixel is represented by a GMM, and if a pixel does not fall within a certain range from any mean vector of the K Gaussians, this pixel is considered a foreground pixel candidate; otherwise, it is classified as background. [44] formulates the pixel-level segmentation for maritime image as a weakly supervised GMM based on semantic structure to identify the four components of sky, complex scenes above horizon or seashore, sea water, and obstacles in pixel level at the same time.

In [44], they consider the image as an array of feature vectors $Y = \{y_i\}_{i=1:M}$, in which the ith feature y_i is composed of pixel's color and image coordinates. Then, the semantic generative model is formulated with four components, three Gaussians and a single uniform component:

$$p(y_i|\Theta) = \sum_{k=1}^{3} \phi(y_i|\mu_k, \Sigma_k)\pi_{ik} + \mathcal{U}(y_i)\pi_{i4}, \tag{2.1}$$

where $\Theta = \{\mu_k, \Sigma_k\}_{k=1:3}$ denote the means and covariances of the Gaussian kernels $\phi(\cdot|\mu, \Sigma)$, and $\mathcal{U}(\cdot)$ denotes an uniform distribution. The ith pixel label x_i is an unobserved random variable governed by the class prior distribution $\pi_i = [\pi_{i1}, ..., \pi_{i4}]$ with $\pi_{i1} = p(x_i = 1)$. The three Gaussian components represent the three dominant semantic regions (sky, complex scenes above horizon or seashore, sea water) in the image, while the uniform component represents the foreground (objects), pixels from which do not belong to any of the three regions. The parameters of the GMM model and the segmentation mask are finally computed by a Markov random field (MRF) [45] framework and an expectation-maximization (EM) algorithm [46].

GMM classifies image pixels to different clusters according to their intensities and locations, and it can not avoid the affecting of outliers (white wake, sun reflection, etc.) either.

2.1.5 Feature Space Clustering

Clustering methods in feature space are also commonly used for maritime obstacle detection. The steps are usually first extracting features (e.g. intensity, texture, etc.) from the image, and then clustering the extracted features using Mean Shift [47], K-Means [48], or etc..

As in [49], image pixels are classified as sea or not using co-occurence matrix [50]. Since it processes in pixel level, the approach of [49] is very sensitive to image noise. [2] proposed the use of variable-size image windows and feature space reclustering for detecting obstacles in maritime images. Nevertheless, the clustering process in this method is sensitive to the outliers in the computation of the mean or the median of the feature set, thus leading to poor performance when there are a larger number of obstacles or more white wake outliers in the image. To solve this problem, [51] omitted the reclustering process and used global sparsity potentials (GSPs) to estimate the mean feature of the sea.

2.1.6 Machine Learning

Machine learning is an intelligent method for pattern recognition. It basically consists of two stages: training and testing. In training, a model is trained with a dataset. In testing, the best label is assigned to a given instance using the trained model. According to the type of learning procedure used to generate the trained model, the machine learning methods can be categorized into supervised learning, unsupervised learning and semi-supervised learning. Supervised learning requires the training dataset to be properly labelled with correct outputs. A learning procedure then generates a model that should have a high recognition accuracy on the training dataset and have a good generalization to the new data. Unsupervised learning does not need the training dataset to be labelled, while it trains the model by finding the inherent patterns in the training data, and then determines correct outputs for the new data with the model. Semi-supervised learning is a combination of supervised and unsupervised learning, so that the model is trained with both labelled and unlabelled data.

Most literatures addressing the object detection task in maritime scenes exploited

the supervised learning. For instance, [52] train a Haar classifier with Haar-like features [53] for vessel recognition. [54] utilize a classical linear regression model [55] to compute a predictive map for the candidate obstacles, and then it is combined with the saliency map [37] of the original image to render an enhanced performance. [56] first use weaker detectors DPM [57] trained with low-cost hand-engineered features (e.g. HOG [58], LBP [59], etc.) to get some candidate regions, where the deep CNN features [60] are then extracted and forwarded to a trained support vector machine (SVM) classifier [61, 62], finally the outputs of SVM are the resulted objects.

Deep learning is one potential machine learning method to detect obstacles in maritime images more efficiently, since it has demonstrated good performance for object detection in many areas [63, 64]. However, due to lack of sufficient number of data samples, we have not yet studied the deep learning methods in this thesis, but it would be an interesting future work.

2.2 Stereo Vision Based Methods

Stereo vision is the reconstruction of 3D information from 2D images. By comparing information about a scene from two vantage points, 3D information can be reconstructed by examination of the relative positions of corresponding points in the two images.

Principle of stereo vision. As illustrated in Figure 2.1, a 3D world point $A = (X, Y, Z)$ is projected onto two camera images C_l and C_r with corresponding 2D image points $a_l = (u_l, v_l)$ and $a_r = (u_r, v_r)$, respectively. O_l and O_r are the focal points of the respective camera image. Thus, the lines $O_l a_l$ and $O_r a_r$ intersect at A. If a_l and a_r are given and the geometry of the two camera images are known, the two projection lines ($O_l a_l$ and $O_r a_r$) can be determined and it must be the case that they intersect at point A. Using basic linear algebra, the 3D point A can be triangulated in a straightforward way.

Generally, there are two forms of stereo vision system: binocular stereo and motion stereo. Binocular stereo refers to using two cameras to reconstruct the 3D information of the scene. If Figure 2.1 illustrates the case of binocular stereo, the camera images C_l and C_r should be captured from two cameras at the same time.

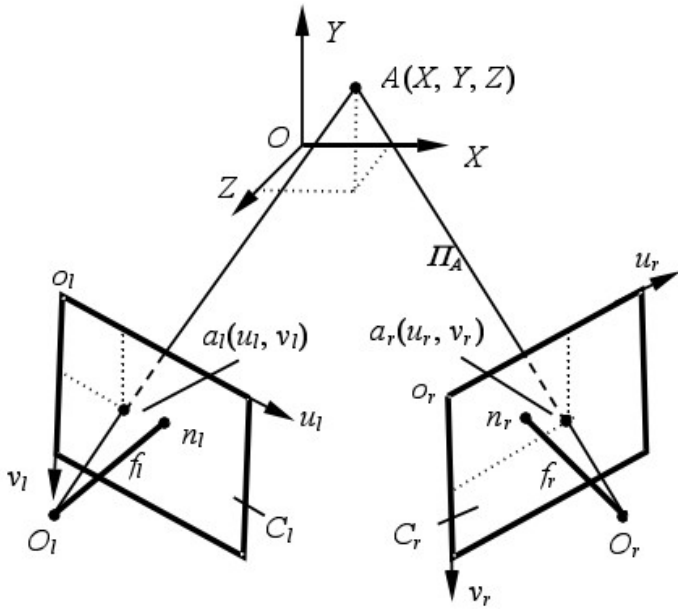


Figure 2.1: Illustration of the principle of stereo vision. (Extracted from [1])

Motion stereo refers to using one moving camera to reconstruct the 3D information of the static scene. If Figure 2.1 illustrates the case of motion stereo, the camera images C_l and C_r should be captured from one moving camera at different time and different locations.

Binocular stereo can reconstruct the 3D information of the scene no matter it is dynamic or static, and many object detection algorithms for unmanned ground vehicles (UGV) or unmanned aerial vehicles (UAV) are performed directly on the 3D point cloud rendered by it. However, motion stereo can only reconstruct static scenes, so it is more commonly used in building the map of the environment, which is the necessary input for robot navigation.

2.2.1 Binocular Stereo

The literature reviewed in Section 2.1 are all based on images or videos in a monocular vision, which has attracted a lot of studies in the past few years. However, when looking into the obstacle detection for USV using binocular stereo vision based system, the research is very scarce, perhaps because its low computing speed or inconvenient experimental setup.

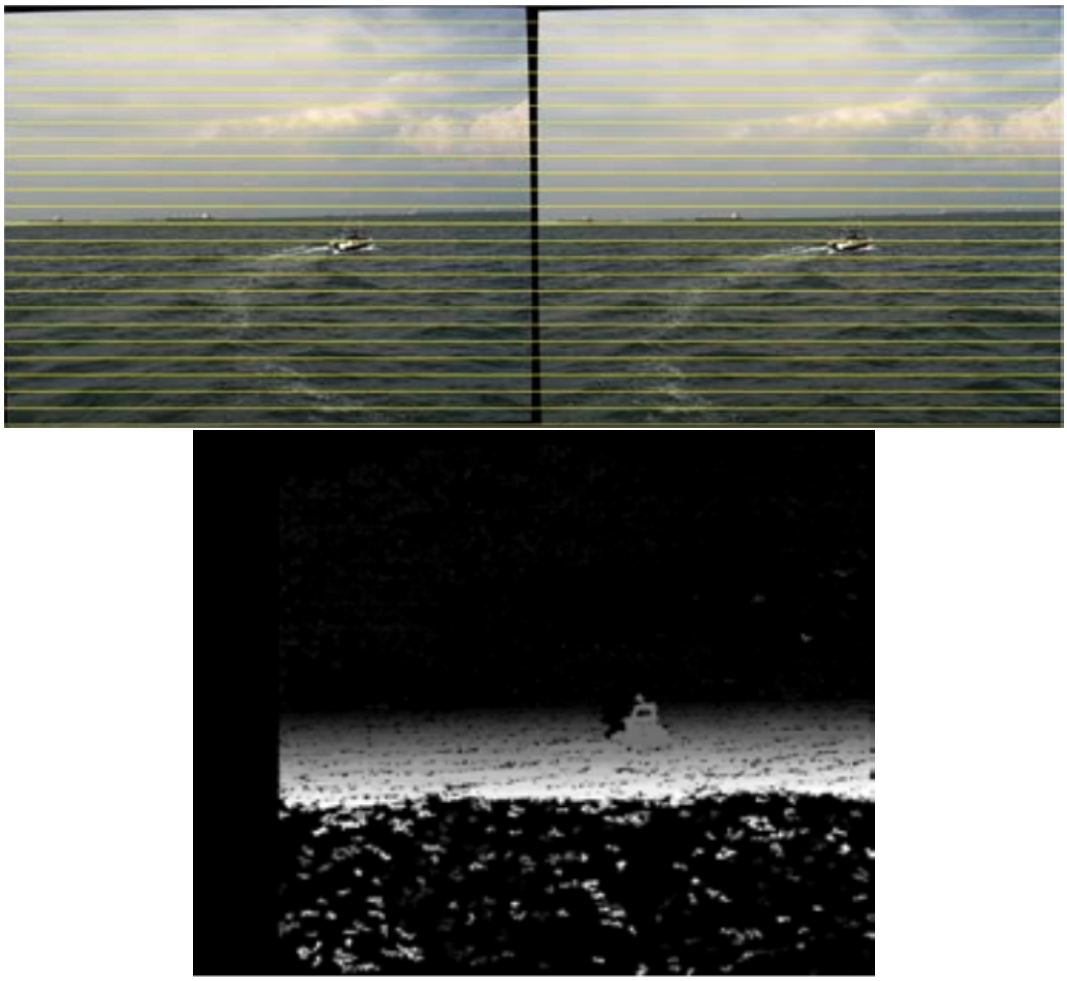


Figure 2.2: A rectified image pair (top) and its corresponding disparity image (bottom) obtained from stereo matching. The horizontal lines placed on the image pair are for visual evaluating the epipolar lines after rectification.

Larson et. al [65] in 2006 proposed that the binocular stereo vision system for UGV developed by NASA Jet Propulsion Laboratory can be extended to the USV domain with very promising initial results, and it provides high resolution 3D data about the near-field environment. Nevertheless, there was not any experimental results reported in their work. [66] is the first published literature for binocular stereo vision-based system on USV with solid experiments. In that work, obstacles are detected by filtering and then classifying 3D points above the fitted water surface plane. Inspired by [67] and [68], which leverage a Bayesian plan-view map based approach to detect and track persons, [69] and [70] transferred this approach to the

task of obstacle detection and tracking for USV by integrating the occupancy and height information of 3D points projected on the plan-view maps.

To further enlarge the detecting and ranging ability of the USV, [71] extended the binocular stereo system of [69] and [70] to larger baseline (2.9 meters), larger focal length of cameras (2820 pixels) and higher definition images (2736×2192). However, it is hard to calibrate the binocular stereo system with such a large baseline using conventional checker board methods. A new calibration method designed for the binocular stereo in maritime environment was proposed in [72] using the clues of horizon line and an infinite point. The top image in Figure 2.2 shows the rectified result of a pair of stereo images in the binocular stereo system of [71] using the calibration method of [72]. The corresponding disparity map obtained from the stereo matching method of SGBM [73] is shown in the bottom image of Figure 2.2. The 3D point cloud can then be calculated from this disparity map.

The advantage of using 3D point cloud for obstacle detection is that it is not affected by the outliers (white wake and sun reflection), which can be filtered out with their height information (their distances to the sea surface plane are relatively small). The disadvantages include the larger computing burden and the difficulty in reconstructing the 3D information of distant scenes.

2.2.2 Motion Stereo

We did not find any literature that develop motion stereo system for the maritime environment. [74] proposed an approach for estimating the distances of marine vehicles using a monocular video camera. The approach detects the horizon and uses its distance as a reference. It detects the contact point of the vehicle with the sea surface by finding a maximally stable extremal region (MSER) [75]. Then, it relies on geometries of the earth and on optical properties of the camera to compute the distance. However, this method can only roughly estimate the distance of an obstacle, but can not reconstruct the full 3D information of an image point.

Motion stereo is well studied and widely used in simultaneous localization and mapping (SLAM) for UGV and UAV, so we reviewed some of these methods for SLAM in the following and discussed the possibilities of applying them to the mar-

itime environment.

Given a series of sensor observations o_t over discrete time steps t, the SLAM problem for a robot moving in an unknown environment is to build a map of the environment m_t, and at the same time, use the map to estimate the robot's location x_t. Thus, the objective in a probabilistic form is to compute: $P(m_t, x_t|o_{1:t})$. In the process of SLAM, m_t and x_t are updated sequentially by applying Bayes' rule.

Given a map and a transition function $P(x_t|x_{t-1})$, the location of robot can be updated by

$$P(x_t|o_{1:t}, m_t) = \sum_{m_{t-1}} P(o_t|x_t, m_t) \sum_{x_{t-1}} P(x_t|x_{t-1}) P(x_{t-1}|m_t, o_{1:t-1})/Z, \tag{2.2}$$

where Z is a constant.

Given the location posteriors, the map can be updated by

$$P(m_t|x_t, o_{1:t}) = \sum_{x_t} \sum_{m_t} P(m_t|x_t, m_{t-1}, o_t) P(m_{t-1}, x_t|o_{1:t-1}, m_{t-1}). \tag{2.3}$$

In this inference problem for map and location, a local optimum solution can be obtained by alternating updates of the two beliefs in a form of EM algorithm.

2.2.2.1 ORB-SLAM

ORB-SLAM [76] is a monocular SLAM system that based on ORB features, which is applied for all SLAM tasks: tracking, mapping, relocalization, and loop closing. By selecting the points and key frames of the reconstruction with a survival of the fittest strategy, it can robustly generate a compact and trackable map that only grows when changes happen in the scene content. Further, ORB-SLAM has advantages in robustly handling severe motion clutter, allowing wide baseline loop closing and relocalization, and full automatic initialization. It can achieve real-time performance in various (small and large, indoor and outdoor) environments.

The performance of ORB-SLAM highly depends on the quality and the number of features detected in the image, because the wrong feature matching or the scarce matched features would definitely lead to a bad visual odometry (VO) estimation, and thus render unreliable results for localisation and mapping. However, there are always few features can be detected in an image of open sea environment, because

the ORB feature [77] can only find keypoints in the corner areas, while the sea surface, which takes most part of the image, has uniform appearance and is hard to detect keypoints on it. Even some corner features can be detected when the waves are shown in the image, they are unreliable because the waves are always moving, and it is impossible to match the features at two different time. Therefore, the ORB-SLAM is only suitable to the environment with plenty strong features that can be detected, such as indoor and urban scenes, but does not work in the maritime situations.

2.2.2.2 LSD-SLAM

LSD-SLAM [78] is a monocular SLAM technique that uses directly pixel intensities rather than keypoints for both tracking and mapping. By tracking the camera through direct image alignment, the geometry is estimated in the form of semi-dense depth maps, which is obtained via filtering over many pixelwise stereo comparisons. After building a Sim(3) pose-graph of keyframes, a scale-drift corrected, large-scale map including loop-closures can be built. LSD-SLAM is able to achieve real-time performance both on a CPU and on a modern smartphone.

One advantage of LSD-SLAM over the keypoint-based approaches is that it uses all information (edges, corners, etc.) in the image to achieve more accurate and more robust performance in sparsely textured indoor environment, and result in a much denser 3D reconstruction. Moreover, it also produce less outliers due to incorporating many small-baseline stereo comparisons instead of only few large-baseline frames.

Although different from ORB-SLAM which depends on feature detection and matching, LSD-SLAM looks for matches directly on the image by comparing the differences of pixels' intensity, it is not suitable to the maritime environment either, because the sea water is always moving and thus many wrong pixel matches would be generated in the region of sea surface, and this could lead to bad performance in LSD-SLAM.

Chapter 3

Image-Based Maritime Obstacle Detection Using Global Sparsity Potentials

3.1 Motivation and Objective

Image-based methods for obstacle detection in maritime scenes have been extensively studied in the last two decades (refer to Section 2.1). However, there are still some limitations in these methods. For example, in the work of [2], the authors proposed an iterative feature reclustering method to determine the centroid of the main cluster (sea), whose outliers are considered as obstacles. Nevertheless, the clustering process in this method is sensitive to the outliers in the computation of the mean or the median of the feature set, thus leading to poor performance when there are a larger number of obstacles or more white wake outliers in the image. Taking saliency detection methods as another example, it seems that this kind of methods are suitable to the this topic, because obstacles in maritime scenes in most cases are distinct from the background. However, experimental results with a state-of-the-art saliency detection method [3] show many false positives caused by wave, white wake, and water speckles.

It can be seen that the image-based methods suffer greatly from the large number of false positives caused by noise, therefore, to further improve the detection

accuracy, this chapter presents a novel algorithm for image-based maritime obstacle detection using global sparsity potentials(GSPs), in which "global" refers to the entire sea area. The horizon line is detected first to segment the sea area as the region of interest (ROI). Considering the geometric relationship between the camera and the sea surface, variable-size image windows are adopted to sample patches in the ROI. Then, each patch is represented by its texture feature, and its average distance to all the other patches is taken as the value of its GSP. Thereafter, patches with a smaller GSP are clustered as the sea surface, and patches with a higher GSP are taken as the obstacle candidates. Finally, the candidates far from the mean feature of the sea surface are selected and aggregated as the obstacles. Figure 3.1 illustrates this process. The proposed algorithm improves the detecting accuracy compared to the traditional feature space reclustering method [2] and a state-of-the-art saliency detection method [3].

To ensure that only the sea surface area is processed and the sea surface is the dominating cluster, the proposed algorithm for maritime obstacle detection is based on two assumptions:

1) The horizon can be detected in the images.

2) Obstacles form a small part ($< 50\%$) of the sea area in the images.

Therefore, only the obstacles below the horizon line in the images are considered the detection targets. In general, the proposed algorithm can be divided into three procedures: horizon detection, sampling and representation of image patches, and obstacle detection using GSP.

3.2 Horizon Detection

In [79], four different horizon detection methods were compared and analyzed, and it was concluded that the Random Sample Consensus (RANSAC) method provides the best results with high accuracy. Therefore, in this work, we apply the RANSAC method for horizon detection. To reduce the computational expense and the noise effect, it is usually better to set a region of interest (ROI) that contains the horizon. However, rather than predefine a fixed ROI for every frame as in [79], we propose a more general method to adaptively estimate the ROI.

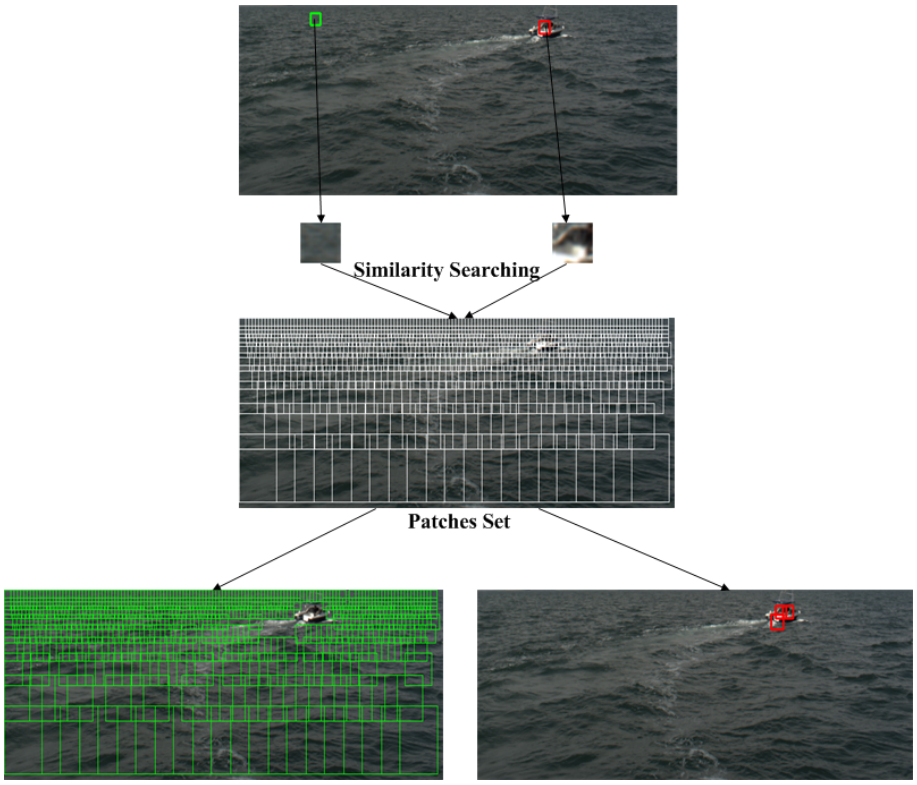


Figure 3.1: Illustration of image-based obstacle detection using GSPs. Each image patch exhibits a different GSP. The top image shows two image patches from the sea surface (green) and the obstacle (red), After they go through the entire patch set (middle image) for similarity searching, patches similar to them can be retrieved as shown in the two bottom images.

We first resize the original image to a smaller size, because downsampling eliminates a considerable amount of noise, and the horizon can still be roughly estimated without significantly biasing the ground truth. In the downsized image, shown in Figure 3.2, first, the gradient map is computed using a Sobel operator, and then, locations with the maximum gradient values along each sampled column are selected as the candidate points; RANSAC is used by randomly selecting two candidate points at each iteration to fit the horizon line. Finally, after reprojecting the estimated horizon in the small image onto the original image, we can define the ROI in the original image by moving the horizon vertically up and down for the same distance to form the upper and lower boundaries, respectively. Thereafter, RANSAC is used

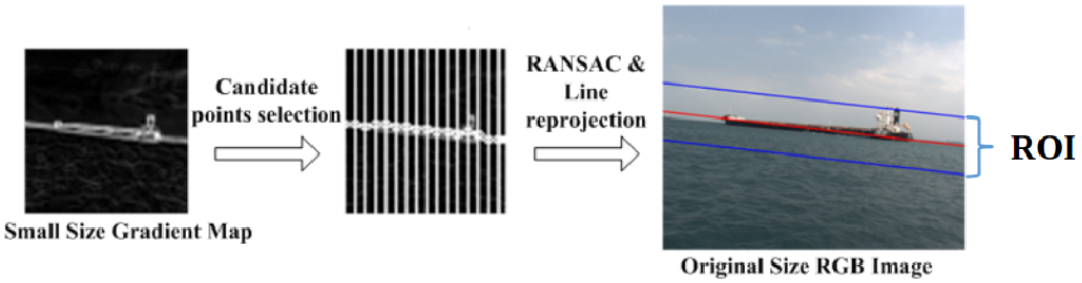


Figure 3.2: ROI (area between the two blue lines) estimation for horizon detection. For visual purposes, the size ratio between the small size gradient map and the original image is enlarged.

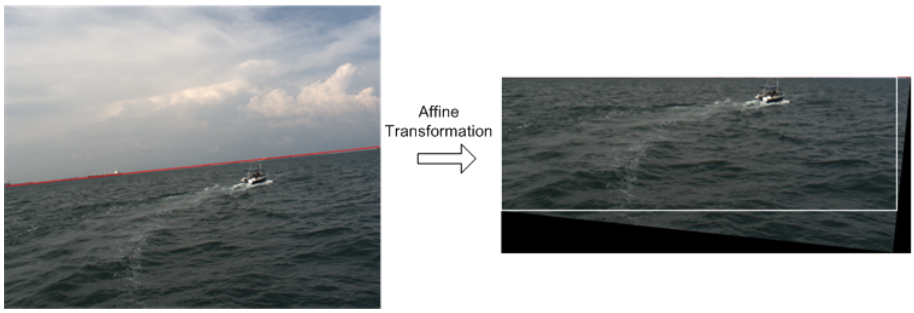


Figure 3.3: ROI for obstacle detection. Affine transformation is applied on the left image to horizontalize the horizon line (red), and then, a rectangle area without artificial pixels is cropped as the ROI (white rectangular area in the right image).

again as in [79] in the ROI for estimating a more accurate horizon in the original image.

As shown in Figure 3.3, after the detection of the horizon line, the ROI for obstacle detection can be obtained via affine transformation and cropping. Then, further processing is performed on the ROI.

3.3 Patch Sampling and Representation

In 2D images, considering the geometric relation between the camera and the sea surface, the resolution of the observation that is close to the horizon is smaller than that of the observation close to the image bottom. Similar to [2], here, square image

patches are sampled from the ROI by using variable-size image windows with an overlap rate of α and an expansion rate of β; the minimum window size is $w \times w$ pixels. An example of this image patch sampling method can be seen in the middle image of Figure 3.1, in which the white rectangles denote the sample patches from the whole image.

To represent the abovementioned sample patches, we adopt a gray-level co-occurrence matrix-based texture analysis [50] as in [2]. In this method, all patches are first resized to the same size ($w \times w$) and then, each image patch f is represented by a four-dimensional vector: $f = [Energy, Entropy, Contrast, Homogeneity]$. Here,

$$Energy = \sum_i \sum_j I^2(i,j), \tag{3.1}$$

$$Entropy = \sum_i \sum_j I(i,j) \log(I(i,j)+1), \tag{3.2}$$

$$Contrast = \sum_i \sum_j (i-j)^2 I(i,j), \tag{3.3}$$

$$Homogeneity = \sum_i \sum_j \frac{I(i,j)}{1+|i-j|}, \tag{3.4}$$

where i and j denote the row and column indices of the image patch, respectively, and $I(i,j)$ represents the intensity value at pixel location (i,j).

3.4 Global Sparsity Potentials

Sparsity potential (SP), originally proposed in [80] for object detection, is a measure that captures the sparseness or similarity of an image patch with respect to its neighborhood. In [80], image patches with a high value of SP are considered to be more discriminative and chosen for training and testing with Hough Forests. However, there are some limitations of applying such a local SP to maritime images because of the unique properties of a sea surface. For example, the neighboring patches of an image patch of the top of a sea wave may contain the bottom of the sea wave; thus, the image patch in the center would be highly distinct from its neighbors and have a low self-similarity or SP value, which would lead to a high probability of being classified as a patch of obstacles. In contrast, if the image

patch is sampled from a big cargo ship, it may have a very similar appearance to its neighboring patches; this may lead to a high SP value, and the image patch may be wrongly classified as the background.

To reduce the abovementioned detection error in maritime images, we propose global sparsity potential (GSP), which computes the self-similarity of an image patch within the entire sea surface area. By taking into account the entire sea surface area, we find that the image patch on the sea has more similar patches in the entire set of patches, while the image patch on the obstacles has the opposite. Thereafter, the discriminative power of image patches increases, and it becomes easier to separate the foreground (obstacles) patches from the background (sea) ones. As illustrated in Figure 3.1, the image patch sampled from the obstacle (boat) region only found three similar patches (red rectangles shown in the bottom right image) in the patch set, while the image patch sampled from the sea water region retrieved majority of patches (green rectangles shown in the bottom left image) in the patch set. That means a patch from the sea water region is less sparse (more similar to most of the patches) in the whole patch set, on the contrary, a patch from an obstacle is more sparse or distinct from the whole patch set.

3.4.1 Measure of GSP

In this work, the GSP of an image patch is measured by its global self-similarity, which computes the similarity of a query patch to the entire patch set. Different from [81], which extracts the global self-similarity descriptors by performing a cross correlation of the patches in the entire image for object classification and detection, we measure the texture similarity of a query patch to the entire patch set by respectively computing their Mahalanobis distances. The smaller the distance between two patches, the higher is the similarity between them. Then, all the computed distances to the patch set are summed and their average is taken as the global self-similarity measure of this query patch.

Here, we denote an image patch with a feature vector expression as f_k, and the entire image patch set as $F = \{f_k, k = 1, 2, ..., N\}$, where N represents the total number of patches sampled in the ROI of an image. Equation (3.5) formulates the

global self-similarity G_k of patch f_k in an image.

$$G_k = \frac{1}{N-1} \sum_{h=1, h \neq k}^{N} \sqrt{(f_k - f_h)^T C^{-1} (f_k - f_h)}, \tag{3.5}$$

where C denotes the covariance matrix of the features set. Then G_k is normalized to be in $[0, 1]$ with 0 representing the most similar and 1 the most dissimilar.

3.5 Obstacle Detection Using GSP

In [2], the researchers proposed the use of variable-size image windows and feature space reclustering for detecting obstacles in maritime images. In their work, an iterative reclustering method is proposed to determine the centroid of the main cluster (sea), whose outliers are considered obstacles. Nevertheless, the clustering process in this method is sensitive to the outliers, because at each iteration, it treats all image patches equally in order to compute the mean or median feature. Therefore, including these outliers in the estimation of the centroid of the sea may decrease the accuracy when there are a larger number of obstacles or more white wake outliers in the image.

To overcome this drawback, in this work, we omit the reclustering process and use GSP to estimate the mean feature of the sea. We propose to select image patches with a high probability to be a sea surface, i.e., having a relatively low GSP value, to estimate the centroid (mean feature) of the sea. Then, as in [2], the outliers are considered as obstacles.

Equation (3.6) and (3.7) formulate the centroid μ of the features of the sea in the proposed algorithm:

$$\mu = \frac{\sum_{k=1}^{N} w_k f_k}{\sum_{k=1}^{N} w_k}, \tag{3.6}$$

$$w_k = \begin{cases} 1, \text{ if } G_k \leq \tau_1 \\ 0, \text{ otherwise,} \end{cases} \tag{3.7}$$

where τ_1 denotes the threshold that selects patches with small global sparsity potentials (high probability for sea surface).

After finding the centroid of the sea, the distances d_r between the remaining patches with GSP $G_k > \tau_1$ and the centroid μ is computed using Equation (3.8).

$$d_r = \sqrt{(f_r - \mu)^T C_p^{-1} (f_r - \mu)}, \tag{3.8}$$

where C_p is the pooled covariance matrix as formulated in Equation (3.9).

$$C_p = \frac{m}{m+1} C_s + \frac{1}{m+1} C_{f_r}, \tag{3.9}$$

where C_s is the covariance matrix of the selected features, C_{f_r} is the covariance of a single candidate feature for obstacle, and m is the total number of selected features. Since the covariance (variance) of a single feature is 0, Equation (3.9) can be rewritten as

$$C_p = \frac{m}{m+1} C_s. \tag{3.10}$$

For each candidate feature of remaining patches, a distance threshold τ_2 is set to pick out the obstacle patches. If $d_r > \tau_2$, the corresponding patch of feature f_r is considered as a patch of obstacle, otherwise, discarded.

Finally, all the detected obstacle patches are grouped and merged to form the obstacle boundaries, and then, an inverse affine transform with respect to the horizon line is applied to map the location of the detected obstacles in the original image.

3.6 Experimental Results

Since there are few available public datasets for maritime obstacle detection, we built our own dataset, the details of which are described in Section 3.6.1. Using this new dataset, we evaluated the accuracy of the proposed algorithm and compared its performance with that of the traditional method [2] and that of a state-of-the-art saliency detection approach [3] in Section 3.6.2. All the experiments in this work were performed on a PC equipped with i7, 3.40GHz CPU.

3.6.1 Dataset

Our maritime obstacle detection dataset consists of four sequences (S_#1, S_#2, S_#3, and S_#4), which were captured by a Point Grey grasshopper CCD camera

mounted on a moving USV on the sea. Each sequence contains 600 RGB frames (size: 684×548 pixels). The obstacle in this dataset is a moving target boat, which varies its distance (approximately from $50m$ to $500m$) to the USV (camera boat). The different sequences present different challenges:

- S_#1 characterizes the detection ability for a short distance, in which the target boat moves close to the USV (within $100m$);
- S_#2 contains many white wake outliers generated by the fast moving target boat, and the distance is around $100m$ to $200m$;
- S_#3 has a majority of the frames without the obstacle shown, and the target boat quickly moves $200m$ away from the USV, and from the left to the right of the image in a few frames at the middle of the sequence; this also provides some white wake outliers;
- S_#4 renders the challenge of distant obstacle detection, and the target boat moves a distance of $200m$ to $500m$ away.

3.6.2 Performance Evaluation

Since the proposed algorithm is based on the prior knowledge of the horizon line, only images whose horizons are detected can be processed for the obstacle detection. Thus, images without a detected horizon are discarded, and not considered in the accuracy or the false rate calculation of the obstacle detection.

The size of small image for ROI estimation in horizon detection is set to 64×64. The parameters for variable-size windows in the image patch sampling part are set as follows: overlap rate $\alpha = 33\%$, expansion rate $\beta = 6\%$, and the minimum window size is 16×16. In Section 3.5, the thresholds τ_1 and τ_2 are set to 0.1 and 0.9, respectively.

Similar to [82], the accuracy evaluation is performed visually as follows:

1). For frames with an obstacle presented, the detection result for each frame is classified as the detected bounding box b being either *correct* (most part of b contains the obstacle), *half-correct* (about half part of b contains the obstacle), or

false (very small or no part of b contains the obstacle). Accordingly, the assigned numeric value $\varrho(b)$ is 1, 0.5, or -1, respectively.

2). For frames without the obstacle shown, the detected bounding box b is classified as either *correct* (b is not detected in the result, i.e., void detection) or *false* (b is detected in the result, i.e., false detection). Accordingly, the assigned numeric value $\varrho(b)$ is 1 or -1, respectively.

Integrating the two above-discussed cases, we can express the score assigned to b as follows:

$$\varrho(b) = \begin{cases} 1 & \text{if } b \text{ is identified as } \textit{correct}, \\ 0.5 & \text{if } b \text{ is identified as } \textit{about half-correct}, \\ -1 & \text{if } b \text{ is identified as } \textit{false}. \end{cases} \tag{3.11}$$

Finally, as formulated in Equation (3.12), the detection accuracy ξ and the false detection rate η can be calculated using all assigned values $\varrho(b)$ in each sequence.

$$\begin{cases} \xi = \frac{1}{n_+} \sum_{k=1}^{n_+} \varrho(b_k), & \text{if } \varrho(b_k) > 0, \\ \eta = \frac{1}{n_-} \sum_{k=1}^{n_-} |\varrho(b_k)|, & \text{if } \varrho(b_k) < 0, \end{cases} \tag{3.12}$$

where n_+ denotes the sum of the total number of ground-truth obstacles and the total number of frames without an obstacle in each sequence, and n_- represents the total number of frames in each sequence.

Table 3.1 summarizes the performance of obstacle detection using the proposed algorithm and two comparative algorithms.

3.6.3 Comparisons and Analysis

The main difference between the proposed algorithm and the feature space reclustering method [2] is the computation of the centroid of the sea features. In [2], the authors proposed the use of all the features of the sampled image patches to estimate the centroid iteratively, while our method involves the selection of image patches with small values of GSP and the calculation of their mean to estimate the centroid. Theoretically, the proposed algorithm is more insensitive to the outliers of the sea features, because rather than taking all the features, which may contain

Table 3.1: Comparison of accuracy (Acc) and false rate (FR) for maritime obstacle detection using different methods

	Saliency VOCUS2 [3]		Feature reclustering [2]		Proposed method	
Sequence	Acc	FR	Acc	FR	Acc	FR
S_#1	0.662	0.214	0.782	0.216	0.893	0.132
S_#2	0.056	0.933	0.652	0.381	0.842	0.230
S_#3	0.827	0.149	0.814	0.195	0.917	0.124
S_#4	0.053	0.925	0.927	0.151	0.965	0.058
Average	0.400	0.555	0.794	0.236	**0.904**	**0.136**

many outliers, to compute the mean or median feature, we just use features with high probabilities to be the sea to compute the mean feature. We reimplemented the method of [2] with the same parameter settings for variable-size window sampling and feature extraction on our maritime obstacle detection dataset. Experimental results show that the proposed algorithm is more accurate than that proposed in [2].

As shown in Table 3.1, the accuracy of the proposed algorithm is more than 10% compared to that proposed in [2] in the first three sequences, which contain many outliers caused by the white wake. Nevertheless, in sequence S_#4, our method performs only slightly better than that proposed in [2]. This could be attributed to the fact that the obstacles in most frames of S_#4 are far away from the camera, so there are very few outliers, such as white wake, in the images. Fewer outliers lead to more accurate estimation of the centroid of the sea; thus, only a small accuracy gap exists between our method and that proposed in [2]. In addition, our method exhibits a smaller false detecting rate than that proposed in [2].

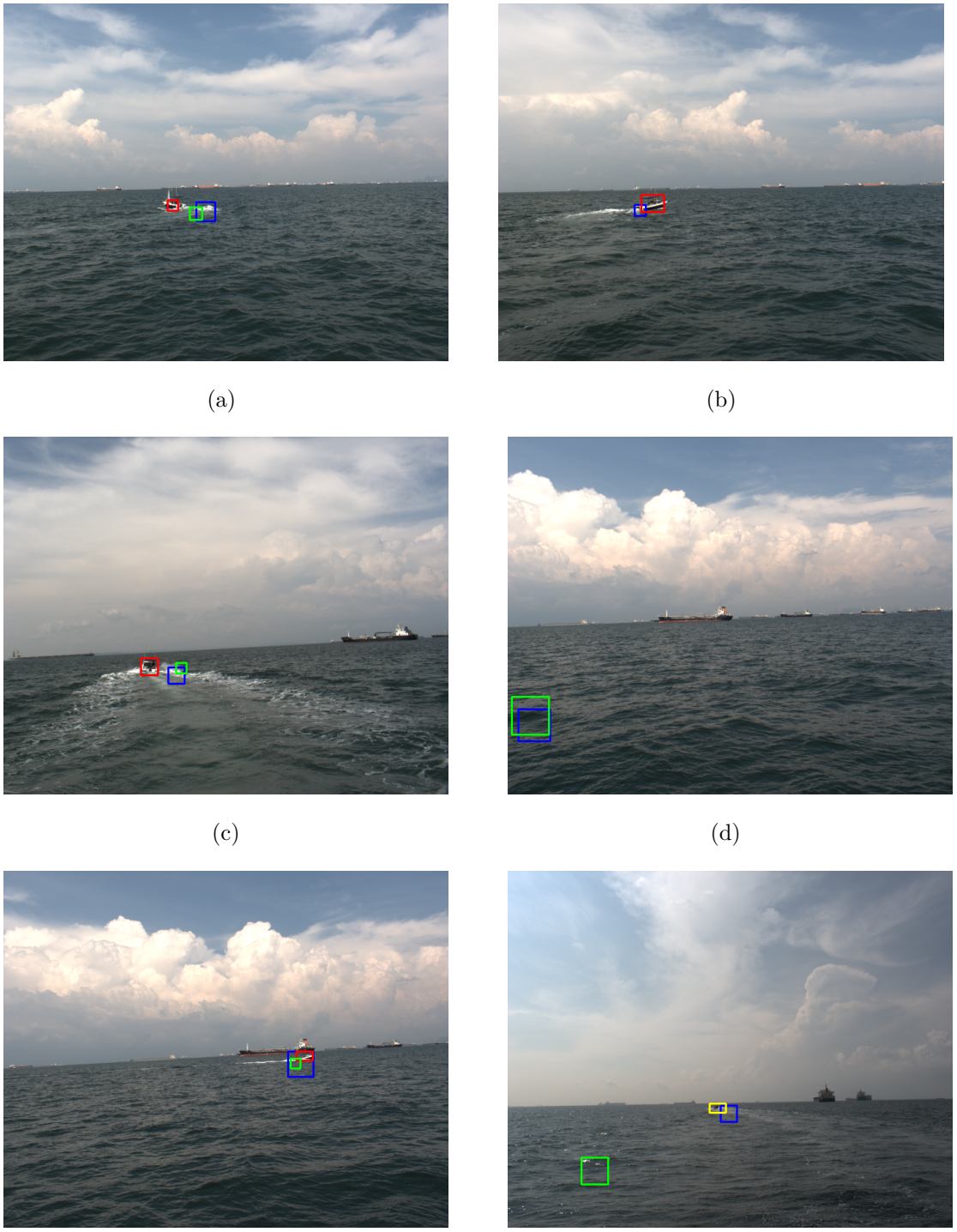
Some advantages of the proposed algorithm over the method proposed in [2] can be seen in Figure 3.4. The yellow bounding box in 3.4(f) means that the red and the green bounding boxes are overlaid. 3.4(a) and 3.4(b) are from S_#1; 3.4(c) is from S_#2; 3.4(d) and 3.4(e) are from S_#3; and 3.4(f) is from S_#4. One can see that the performances of these three methods can be easily evaluated by human eyes. In

Figure 3.4(a) and 3.4(c), false detection caused by white wake happens in the case of the method proposed in [2]. In Figure 3.4(f), the method proposed in [2] has two detections, in which one is correct and the other is false and is caused by a sea wave; the same false detection is also observed in the scenario of Figure 3.4(d). Only a small portion of the boat is detected by the method proposed in [2] in Figure 3.4(e), and this situation is classified as false in Equation (3.11). Figure 3.4(b) shows the missed detection for the method proposed in [2], which wrongly detects nothing for this frame.

To test the saliency detection method for our task, we implemented the work of [3] with our own dataset. As shown in Table 3.1, however, this state-of-the-art method for saliency detection does not perform well for our dataset. However, intuitively, it seems that the obstacles on the sea are more distinct and salient than the sea water. In fact, the saliency-based methods, which usually use the image local contrast information to detect distinct regions, are sensitive to the sea wave. For example, it can be seen in Figure 3.4(a), 3.4(b), 3.4(c), and 3.4(f) that the white wake generated by the boat causes a big problem for this saliency detection method, which results in many false detections. Similarly, in Figure 3.4(d), the dark region between the wave top and the wave bottom is detected as the saliency, but this region is not an obstacle. Although the detected saliency in Figure 3.4(e) contains the obstacle, the bounding box is very big and is not well-fitted to the obstacle. Therefore, we can conclude that it may not be a wise choice to only apply saliency detection to solve the maritime obstacle detection task.

3.7 Concluding Remarks

In this chapter, we introduced a new measure, global sparsity potential (GSP), to capture the sparseness of an image patch throughout the sea area. Using GSP, we developed an accurate and robust approach for moving camera-based obstacle detection in maritime images. In this approach, image patches with a relatively small GSP value are considered the main cluster (i.e., sea surface), while their outliers, which have a relatively large GSP value and a relatively large Mahalanobis distance with respect to the mean feature of the sea surface), are considered the obstacles.



(a) (b) (c) (d) (e) (f)

Figure 3.4: Superior performance for maritime obstacle detection of the proposed algorithm (red) compared to that of the method of feature space reclustering [2] (green) and that of saliency detection VOCUS2 [3] (blue).

Good performance is exhibited in experimental results.

Although this proposed image-based method is convenient, economic, and fast-processing, it always generates many false positives from white wake, waves, sun reflection, etc.. Furthermore, the distances of the detected obstacles can not be known.

Since only the intensity image and the texture feature are explored in the proposed approach, further improvements can be expected by combining the color information and other discriminative features in a future work.

Chapter 4

Binocular Vision Based Obstacle Detection and Tracking for Unmanned Surface Vehicles

4.1 Motivation and Objective

As presented in Chapter 3, image-based methods always suffer from two kinds of challenges for obstacle detection in maritime scenes. One is the foreground (obstacles), which has a lot of appearances varying from tanker, cargo ships, vessels, yachts, to small buoys. It needs a large number of data samples to train a good model with machine learning methods to detect the obstacles, however, in many cases we do not have sufficient data available for training. The other challenge is the background where the noise reside, for instance, white wake, waves, water speckles, and clouds have high probabilities to be wrongly detected as obstacles, because their appearances are much distinct compared with the major background (sea surface and sky).

One efficient solution for the above mentioned issues is to use 3D point cloud, which can be obtained from a binocular stereo vision system. With point cloud, first the sea surface plane can be estimated, and then points close to the sea surface plane is removed, so that the noise in 2D image can be avoided effectively. Finally, by clustering the retained points, we can locate the obstacles in 2D image and restore

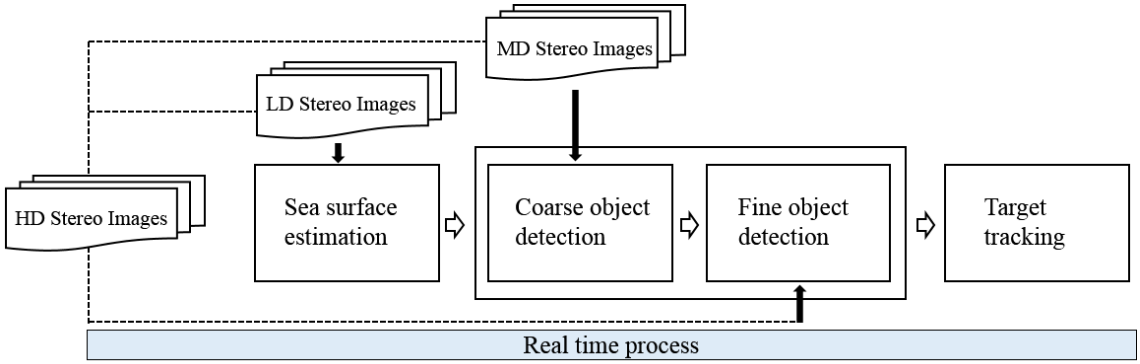


Figure 4.1: Pipeline of the proposed approach for real time obstacle detection and tracking using stereo vision.

their depth or 3D information.

Therefore, in this chapter, a real time binocular stereo vision based long range ($[50m - 500m]$) object detection and tracking algorithm for USV is proposed. Besides large baseline (2.9 meters) and focal length (2800 pixels), high definition (HD) image (2736×2192) is utilised in this work to obtain high accuracy for the obstacle distance estimation. To handle such high resolution images for real time performance, we propose an image-pyramid approach, which firstly estimates the sea surface plane from lower resolution images, and then coarsely detects obstacles from middle resolution images. Thereafter, the detected coarse locations or regions of interest (ROI) are projected to the original HD image. Finally, stereo matching is preformed on these extracted ROI of the original image. In obstacle tracking, we propose a scheme that leverage the depth clue to improve the tracking performance in handling multiple obstacles, scale changing and occlusion. The pipeline of this proposed approach is shown in Figure 4.1. It renders real time processing speed and high accuracy in obstacle detection and distance estimation.

4.2 Obstacle Detection

For the part of obstacle detection, we proposed an image-pyramid approach to reduce the heavy computing burden of the original HD images, and in this way, the system can be run in real time while still keeps high accuracy for object detection and

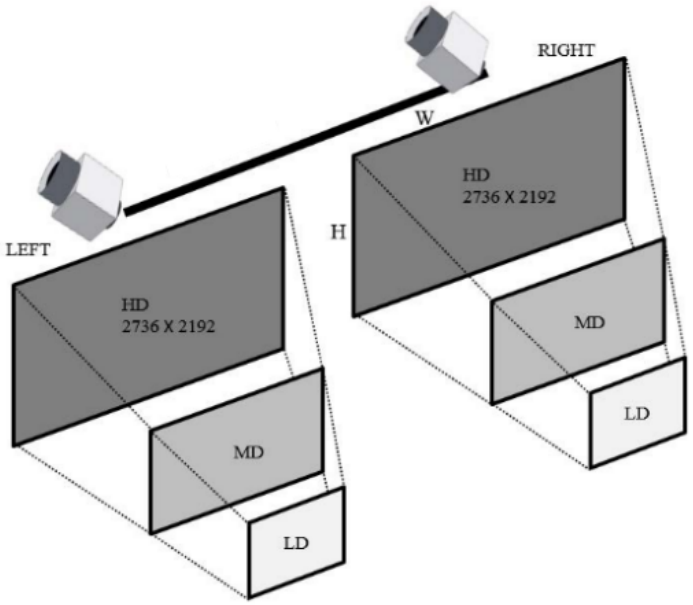


Figure 4.2: Illustration of manipulated images resolution for the proposed image-pyramid approach.

tracking. As shown in Figure 4.1, first, the original HD stereo images are down sampled to low definition (LD) for the sea surface plane estimation. Then the HD images are down sampled to medium definition (MD), on which the coarse obstacle detection is processed by projecting the obtained sea surface plane from LD to MD. With the knowledge of the ROI of detected obstacles in the coarse phase as well as the sea surface plane, fine detection is finally performed on the corresponding ROIs of HD images. Figure 4.2 illustrates the defined image resolution for this image-pyramid approach.

This proposed approach for obstacle detection can be elaborated in the following three parts: 1) sea surface plane estimation; 2) coarse obstacle detection; 3) fine obstacle detection.

4.2.1 Sea Surface Plane Estimation

Suppose the stereo camera is calibrated, and the original HD stereo images with size $W \times H$ are rectified. Given the left and right rectified stereo images, we first down sample them to a relative small size or LD images, from which the disparity map

is computed using the block matching method of [83]. This algorithm is efficient in both speed and accuracy, because it searches for only strong correspondences and rejects correspondences with low texture confidence and uniqueness ration. Then, the 3D point cloud can be reconstructed from the obtained disparity map based on the calibrated intrinsic and extrinsic parameters of the stereo cameras.

As is known, in 3D space, three points that are not in the same line can determine a plane. Therefore, in the obtained point cloud, a 3D plane can be fitted by randomly selecting three points. However, the sea surface plane may not be robustly or correctly estimated if only one or few planes are fitted. To derive an accurate and robust sea surface plane, the random sample consensus (RANSAC) fitting method [84] is applied in this part. To further improve the accuracy and accelerate the computing speed for plane estimation, we roughly estimate the horizon line in the image and only the points below this line are considered valid for RANSAC. Moreover, down sampling of the point cloud is also used to reduce the number of these valid points. The procedure of sea surface plane estimation is summarized below:

- Down sample the rectified HD images I_l and I_r to small size I_{sl} and I_{sr} (e.g. $W/8 \times H/8$);
- Compute the disparity map by implementing dense stereo matching on I_{sl} and I_{sr}, then convert the disparity map to 3D point cloud C_s;
- Estimate the sea surface plane S_s based on the obtained point cloud using RANSAC method.

4.2.2 Coarse Obstacle Detection

As [70], the stereo vision based approach for maritime obstacle detection needs the dense point cloud reconstructed from the stereo images using block matching. However, the larger the stereo images are, the more computational time cost. It is very hard to run in real time on standard CPU, if the dense stereo matching is implemented directly on such huge HD images utilised in this work. In this part, we propose to reduce the original image size to detect the objects coarsely first, so that the ROI can be obtained for further processing and the computing speed can

be increased significantly. However, the size of the down sampled image has to be defined carefully, because small obstacles may not be detected if the image size is too small, but would be taken as noise when doing stereo matching.

The stereo matching method of SGBM [73] is implemented on the rectified and down-sampled left and right images to get the dense disparity map, from which the corresponding 3D points cloud can be computed with the intrinsic and extrinsic parameters of our stereo camera.

Taking the assumption that visual obstacles in open sea are protruding from the sea surface, we can remove the 3D points close to the sea surface plane which has been obtained in section 4.2.1, and also the noise points, if the following criterion is satisfied:

$$Dist(p_i, S) < \delta_1, \quad \text{or} \quad Dist(p_i, S) > \delta_2, \quad i = [1, 2, ..., N], \tag{4.1}$$

where $Dist(\cdot)$ is the function of signed distance between a 3D point and a plane; p_i is a point from the retrieved 3D points cloud whose total number of points is N; S denotes the sea surface plane; δ_1 and δ_2 are predefined positive values, so that points below or close to S are filtered out with lower bound δ_1, and points above S too much are filtered out with upper bound δ_2. After the filtering process with (4.1), the remaining 3D points are considered as candidates for obstacles.

Suppose the normal vector of the sea surface plane is $\mathbf{n} = (n_1, n_2, n_3)^T$, then the roll α and pitch β angles of the USV can be respectively computed as

$$\alpha = \arctan(\frac{n_1}{n_2}), \tag{4.2}$$

$$\beta = \arctan(\frac{n_3}{\sqrt{n_1^2 + n_2^2}}). \tag{4.3}$$

Thereafter, the above selected candidate points with total number of N_c can be transformed to a new camera coordinate by doing inverse rotation with α and β, which makes the USV be placed in a flat sea surface. This transformation can be formulated as

$$p'_j = R_\alpha^{-1} R_\beta^{-1} p_i, \quad i \in [1, N], j = [1, 2, ..., N_c], \tag{4.4}$$

where p'_j denotes as the transformed candidate points; R_α^{-1} and R_β^{-1} are respectively the inverse rotation matrix of α and β.

Similar to [68] and [70], the transformed candidate points p'_j are then projected to two plan-view maps (occupancy map and height map), from which the locations of obstacles are determined by integrating the occupancy and height information. A plan-view map divides a region of the sea surface plane into a set of $n \times m$ cells of fixed size $\upsilon \times \nu$. The set of points $p'_j = (x'_j, y'_j, z'_j)$ projected in cell (u, v) is calculated as

$$P(u,v) = \{j | \frac{x'_j}{\upsilon} = u; \frac{z'_j}{\nu} = v; Dist(p'_j, S') \in [h_{min}, h_{max}]; j \in [1, N_c]\}, \tag{4.5}$$

where S' is the transformed sea surface plane by inverse rotating with roll angle α and pitch angle β; h_{min} and h_{max} are respectively the lower and higher bounds for the height of obstacles.

The occupancy map $O(u, v)$ is defined by weighted summing up the 3D points projected in the cell (u, v) as

$$O(u,v) = \sum_{j \in P(u,v)} (\frac{z'_j}{f})^2, \tag{4.6}$$

where f is the focal length of the camera. In Equation (4.6), one can observe that the weight of a 3D point is proportional to its distance z'_j to the camera, because as we know a same obstacle occupies different region size in an image when the distance between the obstacle and the camera is varying, which results in different amount of 3D points. That is, when the obstacle is close to the camera, it occupies larger area in the image, and thus more 3D points can be retrieved by stereo matching; while when the obstacle is far from the camera, it only takes small area in the image, so less 3D points can be retrieved. Therefore, with Equation (4.6), the scarce 3D points for far away obstacles can be compensated, otherwise they may be considered as noise points.

The height map $H(u, v)$ is defined by the largest height value of the 3D points projected in the cell (u, v) as

$$H(u,v) = \begin{cases} \max(Dist(p'_j, S') | j \in P(u,v)), & \text{if } P(u,v) \neq \emptyset \\ h_{min}. & \text{if } P(u,v) = \emptyset \end{cases} \tag{4.7}$$

Then, the obtained occupancy and height maps are combined to estimate the probability of a cell to be the center of an obstacle using a mixture Gaussian distri-

bution, which searches for regions where the occupancy and height values fall into certain normally distributed ranges. The probability measure $\varrho(u, v)$ at cell (u, v) is defined as

$$\varrho(u,v) = \frac{\exp(-(\frac{(O(u,v)-\mu_o)^2}{2\sigma_o^2} + \frac{(H(u,v)-\mu_h)^2}{2\sigma_h^2}))}{2\pi\sigma_o\sigma_h}, \tag{4.8}$$

where μ_o and μ_h are the expected mean of $O(u, v)$ and $H(u, v)$ respectively, and σ_o and σ_h are the corresponding standard derivation.

With Equation (4.8), we can get a probability map, on which the cells with low probability values are set to 0. After that, cells with probability values are clustered and contours are detected to be the boundaries of obstacles in probability map. Finally, the corresponding bounding boxes for obstacles shown in 2D image can be determined by projecting the points in probability map back to the left image.

4.2.3 Fine Obstacle Detection

In the coarse detection part, the accuracy of detection and distance estimation is usually not satisfying. Therefore, we propose to further improve the result by implementing stereo matching on the ROI in the original HD stereo images. The ROI is obtained by projecting the bounding boxes computed in the part of coarse obstacle detection. The same Bayesian plan-view map method as used in coarse obstacle detection is then applied to retrieve the obstacles' locations and distances. Since the ROI are quite small regions compared to the HD image, and there are less noise caused by other regions of the image, the processing speed in this part is quite fast, and the detection results are more accurate. The process of fine obstacle detection can be summarized as below:

- Project the regions of the candidate obstacles obtained in the coarse detection phase to the corresponding regions (ROI) in the left HD image I_l;
- Project the estimated sea surface plane S_s to the HD image size as S_o;
- Implement stereo matching and compute the 3D points on ROI, then cluster the points with certain heights above S_o as the final obstacle detection results. Meanwhile, the distance of each detected obstacle is computed.

4.3 Obstacle Tracking

In the phase of obstacle tracking, we propose a tracking-by-detection approach, which means that the obstacle detection and tracking are performed at the same time. Obstacle detection is performed in each frame to initialize newly detected obstacles for tracking.

Scale adapting and occlusion handling are two important issues in visual tracking, but usually tricky for many algorithms to address them efficiently. In this part, we propose a framework for multiple-obstacle tracking with the aid of their depth information, which renders the vital clue for scale adapting and occlusion handling. In the proposed framework, each detected obstacle is tracked by an independent tracker respectively, and the depth of an obstacle is obtained by computing the disparity between the matched obstacles in left and right rectified images. Thereafter, the bounding box scale of a tracked obstacle is adapted according to its linear relationship to the depth. When occlusion occurs, the occluded obstacle is terminated to track, and when it reappears, the same identity number before occluding is assigned to it to keep the consistency of tracking.

4.3.1 Spatio-Temporal Context Learning

Obstacle tracking in this work is based on 2D image using the method of spatio-temporal context (STC) learning [4], which exploits the spatio-temporal context for visual tracking. This approach formulates the spatio-temporal relationships between the object and its local context based on a Bayesian framework, which models the statistical correlation between the low-level features (i.e., image intensity and position) from the target and its surrounding regions. Its formulation is shown as below:

$$\begin{aligned} P(l) &= \sum_{c(n)\in F^c} p(l, c(n)|o) \\ &= \sum_{c(n)\in F^c} p(l|c(n), o)p(c(n)|o), \end{aligned} \tag{4.9}$$

where o denotes the object present in an image, and l is its location; $P(l)$ is the confidence map of the object location likelihood; $c(n)$ is a context feature in the context

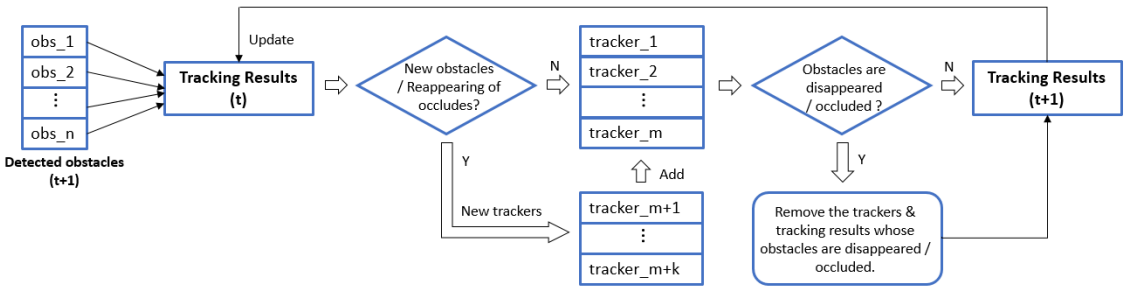


Figure 4.3: Pipeline of the proposed scheme for multiple-obstalce tracking.

feature set F^c; $p(l|c(n), o)$ models the relationship between the object location and its spatial context; $p(c(n)|o)$ models the context prior probability.

In this method, the spatio-temporal model is learned fast adopting the fast Fourier transform (FFT), which makes the tracking performance super fast.

4.3.2 Multiple-Object Tracking

The method of STC [4] is designed for single-object tracking. Multiple-obstacle tracking in this work is implemented by assigning each detected obstacle an independent STC tracker, and these trackers are managed in our proposed scheme during the run time. Figure 4.3 illustrates this proposed scheme for multiple-obstacle tracking.

As shown in Figure 4.3, obstacles are detected in the frame at time $t+1$, then they are examined with the tracking results at time t to determine whether there are new obstacles coming into the frame or obstacles that were previously occluded reappearing in the current frame. If yes, new trackers are initialized for these new obstacles, and added to the set of trackers. If no, the old trackers keep tracking, while the happening of tracking lost and occlusion are checked. If yes, the trackers and tracking results of these obstacles are removed, and the final tracking results are updated. If no, just updating the tracking results at time $t+1$. The scheme for managing of the trackers can be summarised as follows:

- An obstacle is confirmed for tracking only if it is detected in N consecutive frames.

- Each confirmed obstacle initializes a tracker independently, and the trackers run in parallel.
- When new obstacles are confirmed to appear in image, new trackers are initialized for each of them.
- When the tracked obstacle disappears (tracking lost) in image, its corresponding tracker is removed.

The tracked obstacles are considered disappearing (tracking lost) if one of the following criteria is satisfied:

- $R_o \cap R_I \neq R_o$;

  The bounding box R_o of the tracked obstacle exceeds the image boundary R_I.

- $d_t \notin [D_1, D_2]$;

  The depth d_t of the tracked obstacle exceeds the predefined distance range $[D_1, D_2]$.

- $|d_{t+1} - d_t| > D_3$;

  The depth change of the same tracked obstacle between two consecutive frames is larger than the predefined threshold D_3.

4.3.3 Occlusion Handling

In real life application, it is important to keep the identity number of the tracked obstacle consistent when it is occluded in a short while and reappears, i.e. the same obstacle has only one identity number. Nevertheless, most tracking methods fail to track the same obstacle when it reappears after being occluded, because the tracked appearance or features are changed when occlusion happens. [85] proposed a multiple-object detection and tracking method to handle dynamic occlusions for ground vehicles. In that work, a classifier for fully-visible vehicles (occluders) and two classifiers for partially-visible vehicles (occludees) are trained for object detection and state updating in tracking. However, their method can only be used for specific object detection and tracking, like cars. Since our goal is to detect and track

all possible obstacles on the sea surface, the method of [85] is not suitable to our case, because it is hard to train some classifiers that can handle all kinds of obstacles with diverse appearance features.

The tracking method [4] used in this work also suffers from the problem that it can not differentiate whether the tracked obstacle is occluded or not, i.e. it can not handle the occlusion well. Here, we propose a simple solution for the occlusion issue with the aid of depth clue:

- **Occlusion** is considered to happen when there is a sudden change of the depth of the tracked obstacle between two consecutive frames:

  $d_{t-1} - d_t > D_o, \quad (D_o > 0).$

  The change is assumed to be from large depth to small depth and larger than a threshold D_o, because only the far away obstacles can be occluded.

  Then, this occluded tracker is removed while its previous depth $d_o = d_{t-1}$ and identity number $ID_{occludee}$ as well as the identity number $ID_{occluder}$ of the occluder, which occludes the obstacle, are saved.

- **Reappearing** happens when the obstacle Loc_{new} is detected around the front obstacle $Loc_{occluder}$ that covered the back obstacle previously, and the depth change comparing with the saved depth when occlusion below a threshold D_{diff}:

  $dist(Loc_{new}, Loc_{occluder}) < T_{loc} \quad \& \quad |d_{new} - d_o| < D_{diff}.$

  Then, this newly detected obstacle initialize a new tracker, and the saved identity number is assigned to it, which indicates that it is the same obstacle as before occluding.

4.3.4 Scale Adapting

Scale adapting means that the bounding box size of the tracked obstacle should vary adaptively with the distance of the obstacle to camera. When the tracked obstacle is getting close to the camera, the bounding box should be getting larger than the initial size. Accordingly, the larger distance, the smaller bounding box compared to

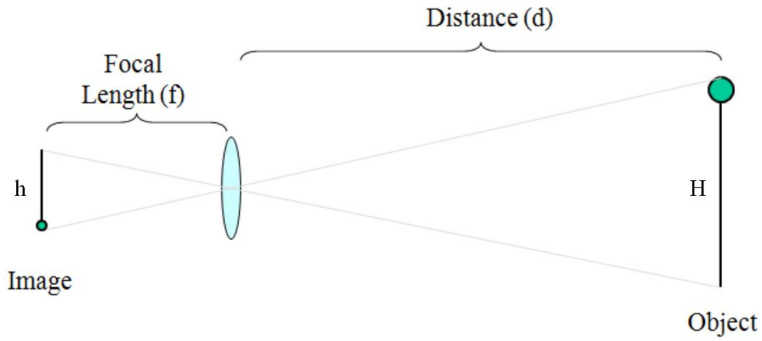


Figure 4.4: Pinhole camera geometry.

the initial size. However, it is hard to estimate the scale changing based only on 2D images. Though [4] proposed an approximation for the scale updating based on 2D images, it usually accumulates the error of scale updating in a longer time.

Based on the perspective view geometry of the camera as shown in Figure 4.4, the relationship between the scale of 2D image view and the 3D depth is linear, which can be formulated as the following equation:

$$h = \frac{Hf}{d}. \tag{4.10}$$

Therefore, for each frame, we have

$$\frac{h_i}{h_0} = \frac{d_0}{d_i}, \quad (i = 1, 2, ..., n) \tag{4.11}$$

where h_0, d_0 are respectively the size and distance of the obstacle when it is initialized to be tracked; h_i, d_i are the size and distance of this tracked obstacle at the ith frame counting from its tracker is initialized. With Equation (4.11), the scale adapting problem can be solved easily by taking the initial scale and depth of the tracked obstacle as reference, then the scale in the following frames adapts according to its depth.

The context prior model $p(c(n)|o)$ in [4] is formulated as

$$p(c(n)|o) = I(n) * ae^{-\frac{|n-x^*|^2}{\sigma^2}}, \tag{4.12}$$

where $I(n)$ is image intensity; a is a normalization constant; x^* is the center location of the tracked obstacle; σ is a scale parameter. Instead of the unreliable scale

updating scheme in [4], in our work, the scale parameter σ_{t+1} at time $t+1$ is updated by

$$\sigma_{t+1} = s_t \sigma_0 = \frac{d_0}{d_t} \sigma_0, \tag{4.13}$$

where d_0 is the initial depth of the tracked obstacle; d_t is the depth of the tracked obstacle in the previous frame; σ_0 denotes the initially defined scale parameter.

Depth estimation for a tracked obstacle is illustrated in Figure 4.5. While the obstacle (yellow rectangle) is being tracked in the rectified left image, the searching region (area between blue lines) for template matching in the rectified right image is defined automatically. Since both the stereo images are rectified, the matched patch resides at the same row of the template patch. Also, the column location of the matched patch in the right image cannot be larger than that of the template, and the left boundary of the searching region is set to 0. Therefore, the searching region for template matching is constrained to be very small, which dramatically reduces the burden of computing.

Ensuring robustness. For compensating a possible error in image rectification, we enlarge the search region a little at the top, bottom, and at the right boundary of the search region by a few pixels. Only using the tracked region to do template matching may cause a mismatch in some cases. Thus, we propose to apply the xSobel operator first (being a simple and robust edge detector) on the tracked region to select some edge feature points, and then we generate a number of smaller templates around those selected feature points. After doing constrained template matching for all those small templates, a more robust and accurate target distance is obtained in general using the distance median for all the matches.

4.4 Experimental Results

Our experiments demonstrate a robust and accurate performance. Experimental results with our own dataset also verified the high time-efficiency (about 10Hz) of our proposed system.

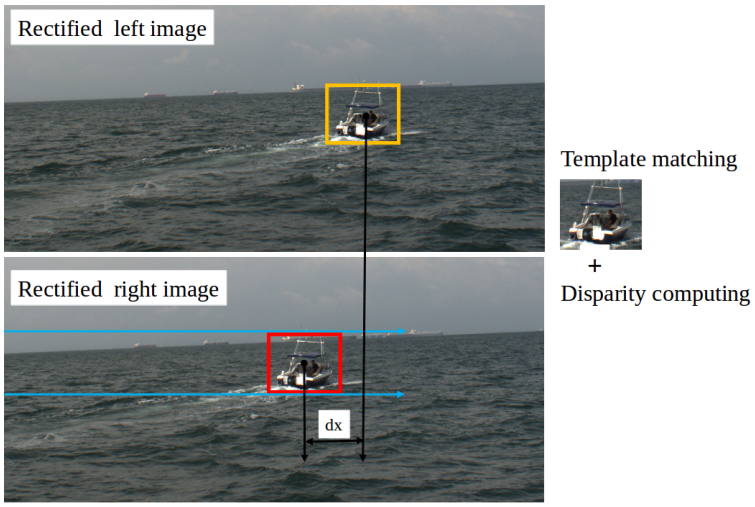


Figure 4.5: Depth estimation for the tracked obstacles.

4.4.1 System Setting

The proposed detection and tracking system for an USV is composed of a pair of Point Grey CCD cameras with a focal length of 2,820 pixels. The two cameras are mounted in parallel, facing forward, and about 2.9 meters apart on the USV, which is the base line length of the stereo rig. The PC of this system is characterised by a 3.40GHz Intel Core i7-3770 CPU. All the reported experiments in this paper were conducted on this PC.

Stereo calibration. The intrinsic parameters of cameras are calibrated using the conventional checkerboard method. The extrinsic parameters of cameras are calibrated using the method of [72] due to the large baseline, which is manually measured by a ruler.

Parameters Setting. In Section 4.3.2: $N = 5$, $D_1 = 50m$, $D_2 = 500m$, $D_3 = 100m$. In Section 4.3.3: $D_o = 100m$, $D_{diff} = 100m$. The initial scale parameter σ_0 for scale adapting in Section 4.3.4 is set to $\frac{w_0+h_0}{2}$, where w_0 and h_0 are the width and height of the initial tracked bounding box.

4.4.2 Depth Estimation Evaluation

We first evaluate the accuracy of depth estimation in our proposed system. In this evaluation, the target for detection and tracking by our USV is an active boat

equipped with GPS. We also have GPS on our USV. The GPS data of our USV and the target boat are logged at the same time, so that the relative distance (ground truth) between them can be calculated.

Dataset. The dataset in this evaluation contains eight long data sequences with 4,800 pairs of stereo frames in total. Each video consists of 600 stereo frames, and each frame is of HD resolution 2736×2192 pixels. We selected challenging scenarios. Those eight videos are characterised as follows:

1. Video #1 shows a target boat that travels towards our USV while the USV stays at a defined spot; the target boat approaches the seashore and travels at a long range.

2. Video #2 shows a target boat that travels in front of the USV, which follows the target boat are closer distance. Rough sea-foam is caused by the movement of the boat.

3. Video #3 shows a target boat that travels towards the open sea and which moves away from the USV, it also passes by very close to container ships.

4. Video #4 shows a target boat that travels towards the open sea and which travels near to container ships.

5. Video #5 shows a target boat that approaches towards the USV, with the open sea as the background, and travels at a long range.

6. Video #6 shows a scene where the USV follows a target boat that crosses right to left at farther distance.

7. Video #7 shows a target boat which approaches towards the USV, with the seashore as background.

8. Video #8 shows a target boat that approaches towards the USV, with container ships as the background.

Figure 4.6 presents the travelling paths of the active target boat and of our USV at map segments of Singapore harbour. The eight different videos are collected with

GPS stamps in challenging scenarios, and we calculate trajectories of the travelling boat and of the USV using the GPS data.

We illustrate the better performance of our developed system for challenging frames by showing a few examples of such situations in Figure 4.7. Figures 4.7(a), 4.7(b), and 4.7(c) show that detection and tracking of the target boat are progressing well under challenges defined by approaching the seashore and dynamic noise such as sea-foam on the water and swell of waves. Figures 4.7(d), 4.7(e), and 4.7(f) show detection and tracking under challenges defined by approaching the container ships and travelling along the container ships. Figures 4.7(g), 4.7(h), and 4.7(i) illustrate successful detection and tracking for challenges defined by travelling along several container ships and also travel at far range. Figures 4.7(j), 4.7(k), and 4.7(l) show that detection and tracking are progressing well under challenges defined by going along the seashore while container ships travel from right to left at long range.

We compare the distance, estimated by our system, with ground truth. The ground truth was obtained through GPS devices mounted on both boats. See Figure 4.8 for a visualisation of this comparison. GPS trajectories of the target boat and the USV are shown on the map; ground truth is shown as black line, the computed distance by red dots, and the optimised distance values as blue marks. We conclude from this performance analysis that distance estimation is more accurate when the target boat is within 300 meters distance to the stereo vision system. We can also observe that distance estimation appears in discrete steps (also known as cardboard effect); the well-known reason is that disparity values are discrete instead of continuous, and calculated disparity is less accurate for distant objects than for close objects. This cardboard effect can be observed more clearly for distances close to 500 meters.

4.4.3 Multiple-Obstacle Tracking Evaluation

In this part, we evaluated the ability of multiple-obstacle detection and tracking in our system. The targets are two active boats moving in different scenarios, such as travelling towards the open sea, approaching towards the USV, and moving cross each other. We also evaluated the abilities of occlusion handling and scale adapting

during the tracking of multiple obstacles.

Dataset. Since there are few public datasets for stereo vision based maritime scene, in this experiment, we collected three sequences (S_#1, S_#2, S_#3) using our stereo vision system in USV for testing. Each sequence consists 600 pairs of stereo frames with size 2736×2192 pixels. These sequences are characterised as follows:

1. S_#1 shows two boats move from near to far $(100m - 500m)$ at the same time, and no occlusion.

2. S_#2 shows two boats move from far to near $(200m - 30m)$ at different time, and no occlusion.

3. S_#3 shows two boats move towards each other along horizontal axis of the image, and occlusion happens when they cross each other.

Therefore, S_#1 and S_#2 are mainly used for testing the performance of scale adapting, while S_#3 is used for occlusion handling test.

As with the experiments in Chapter 3, we evaluate the performance visually. Obstacle tracking results for each frame are classified as tracked bounding box b being either *correct* (close fit to the ground truth) or *not correct* (not fit well to the ground truth) accordingly, and assigned numeric values $\varrho(b)$ is in the set $\{0, 1\}$ respectively. Finally, the assigned values $\varrho(b)$ is reassigned according to the number of tracked bounding boxes in its frame.

$$\varrho(b) = \begin{cases} 1 & \text{if } b \text{ is identified as being } \textit{correct} \\ 0 & \text{if } b \text{ is a case of } \textit{not-correct} \end{cases} \tag{4.14}$$

$$C_f = \sum_{k=1}^{n} \varrho(b_k)/n \tag{4.15}$$

where n denotes the total number of tracked bounding boxes in each frame. Value C_f is expressed as percentage to the *correct detection rate* as shown in Table 4.1. The values in this table indicate a better performance of our proposed approach compared to the state-of-the-art tracking method [4].

Table 4.1: Comparison of multiple-obstacle tracking results with different methods.

	STC [4]	Proposed approach
Sequence	Correct tracking rate	Correct tracking rate
S_#1	90.6 %	99.3 %
S_#2	72.5 %	93.7 %
S_#3	66.8 %	90.5 %

Scale adapting. Figure 4.9 and Figure 4.10 illustrate the compared tracking results for scale adapting with STC [4] and our method. Four frames are taken from S_#1 and S_#2 respectively for exampling. Each tracked obstacle is assigned an unique ID and specific color for its bounding box. Meanwhile, the distance of the tracked obstacle to the USV is displayed on top of the bounding box. In Figure 4.9, when the boats travel away from us, the proposed approach performs slightly better than the compared STC tracker. However, when the boats approach towards us, as shown in Figure 4.10, the proposed approach performs much better than STC, in other words, the proposed approach outputs bounding boxes fitting the obstacles better.

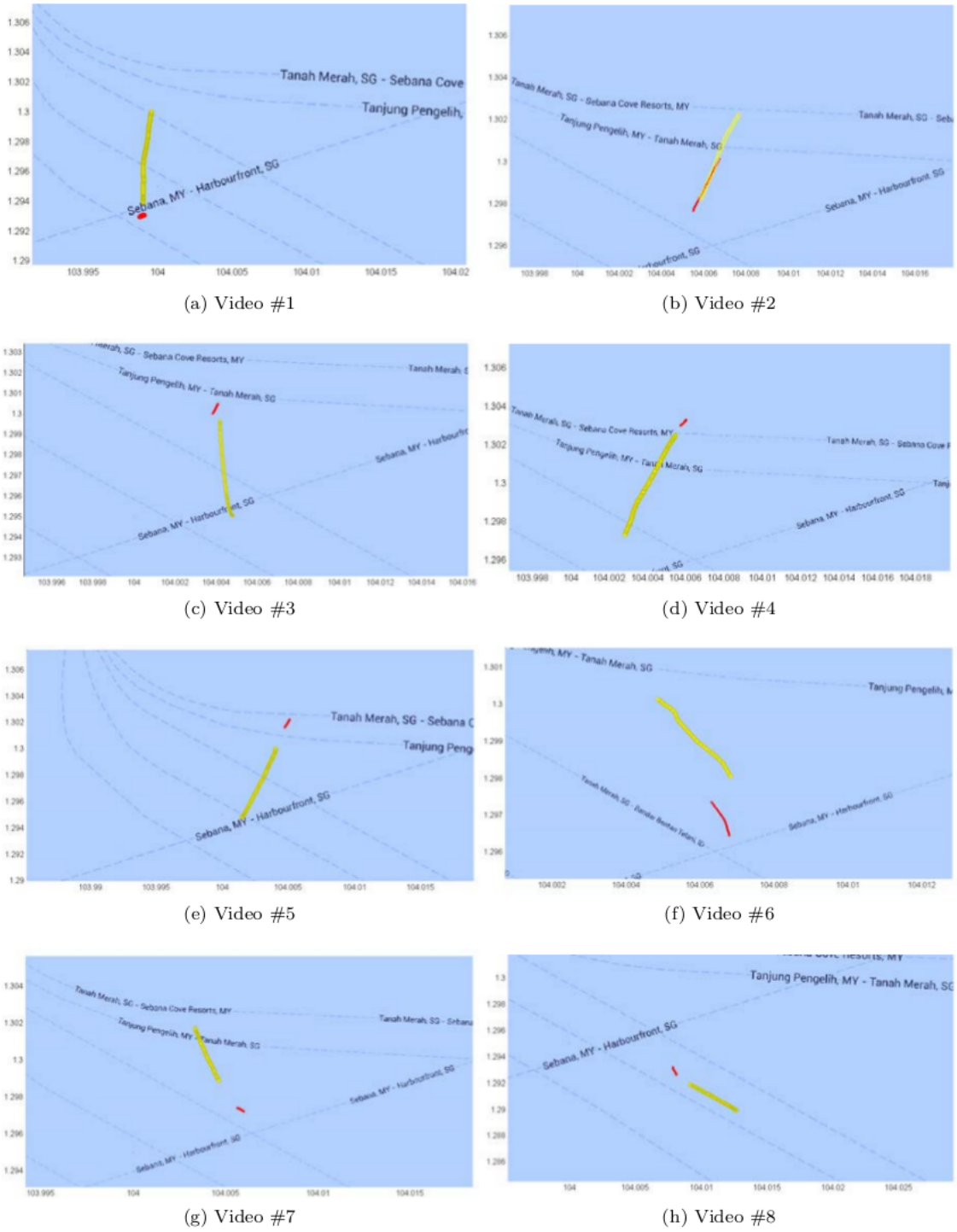
Occlusion handling. The occlusion handling results are shown in Figure 4.11. Three frames are sampled from S_#3, and they indicate the scenarios of a target boat begin to be occluded, totally occluded, and reappearing, respectively. One can see that in the 173th frame, the two boats start to cross each other; In the 183th frame, the boat with ID 1 is totally occluded by the boat with ID 2, and the STC tracker [4] for ID 1 is misclassified to ID 2, while in our method, the tracker for ID 1 is removed to avoid misclassification; In the 237th frame, the reappeared boat ID 1 is detected and tracked again, and without occlusion handling, the newly detected obstacle is assigned a new ID 3 by STC tracker [4], but with our approach, the original ID is preserved and reassigned to it, which keeps the ID of an obstacle consistent during tracking.

4.5 Concluding Remarks

We have developed a real-time long-range obstacle detection and tracking system for USVs. For obstacle detection, an image-pyramid approach has been proposed to achieve real-time processing speed with high resolution image. For obstacle tracking, we have presented a novel multiple-obstacle tracking framework. Stereo vision is adopted to retrieve the depth of the tracked obstacles, which provides the prior knowledge for scale adapting and occlusion handling. The superior experimental performance (accurate, robust, and fast) of our proposed approach has been demonstrated with our own maritime dataset.

The proposed approach greatly reduces the false positives caused by noise (white wake, waves, sun reflection, etc.) in obstacle detection compared to that of the image-based methods. Nevertheless, the large baseline makes the stereo rig really bulky and hard to calibrate.

The processing speed is influenced by the number of obstacles in tracking. That means, when there are many obstacles in a frame are being tracked, the processing speed would drop and be slow. In that case, the real-time performance cannot be achieved, due to the increased computing burden. Moreover, we consider the obstacle is occluded or reappears only based on the depth information, which may be insufficient and weaken the reliability of the occlusion handling. In the future work, the appearance information can be added to enhance the robustness of the occlusion handling.



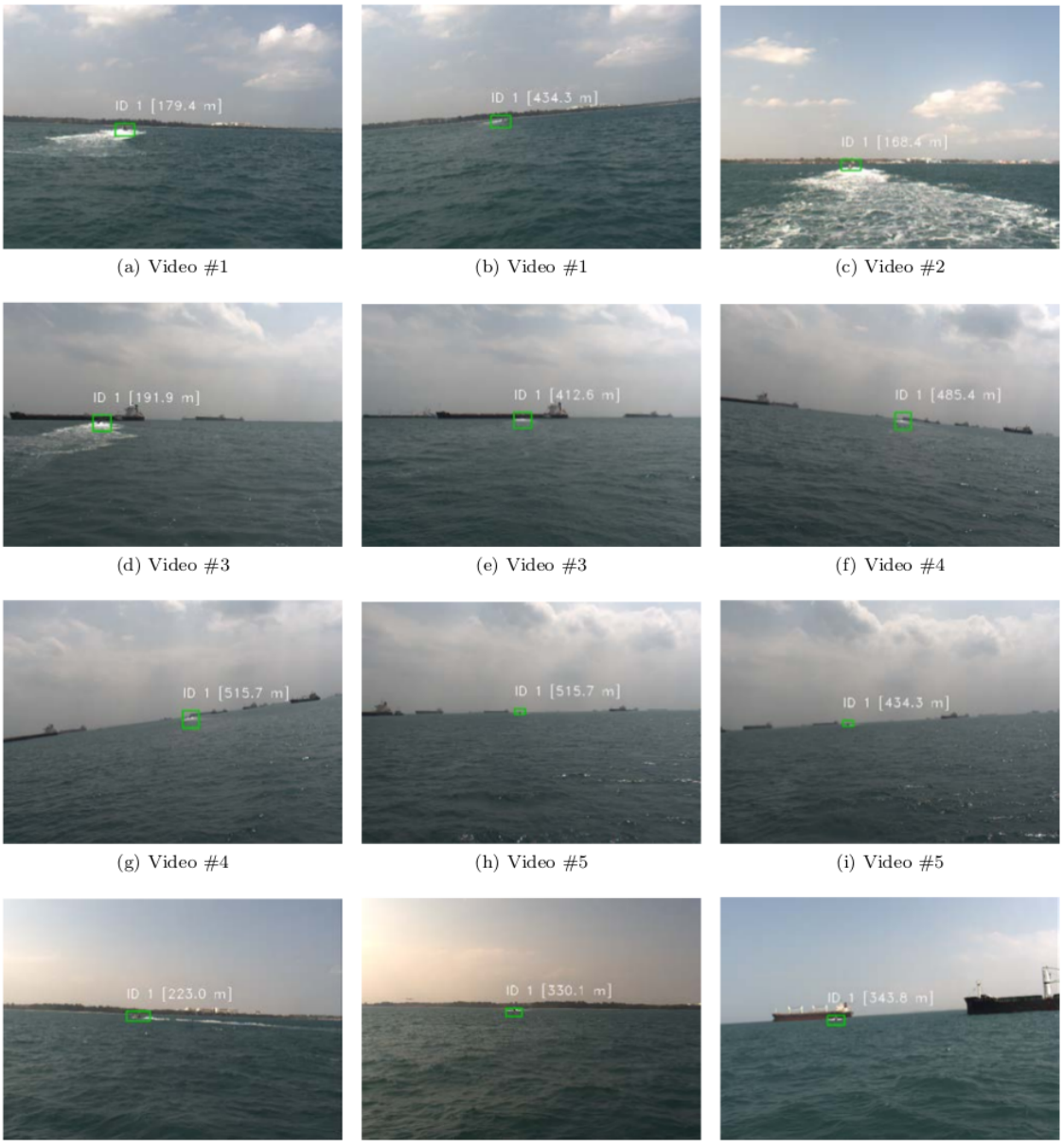
(a) Video #1 (b) Video #2

(c) Video #3 (d) Video #4

(e) Video #5 (f) Video #6

(g) Video #7 (h) Video #8

Figure 4.6: Fragments of a map of Singapore harbour, also showing GPS trajectories of an active target boat and of our USV. Our USV is shown in red, and the target boat in yellow.



(a) Video #1 (b) Video #1 (c) Video #2

(d) Video #3 (e) Video #3 (f) Video #4

(g) Video #4 (h) Video #5 (i) Video #5

(j) Video #6 (k) Video #7 (l) Video #8

Figure 4.7: Selected results for detection and tracking for challenging frames. The detected and tracked target is shown by a green bounding box. The distance is computed and displayed in real time.

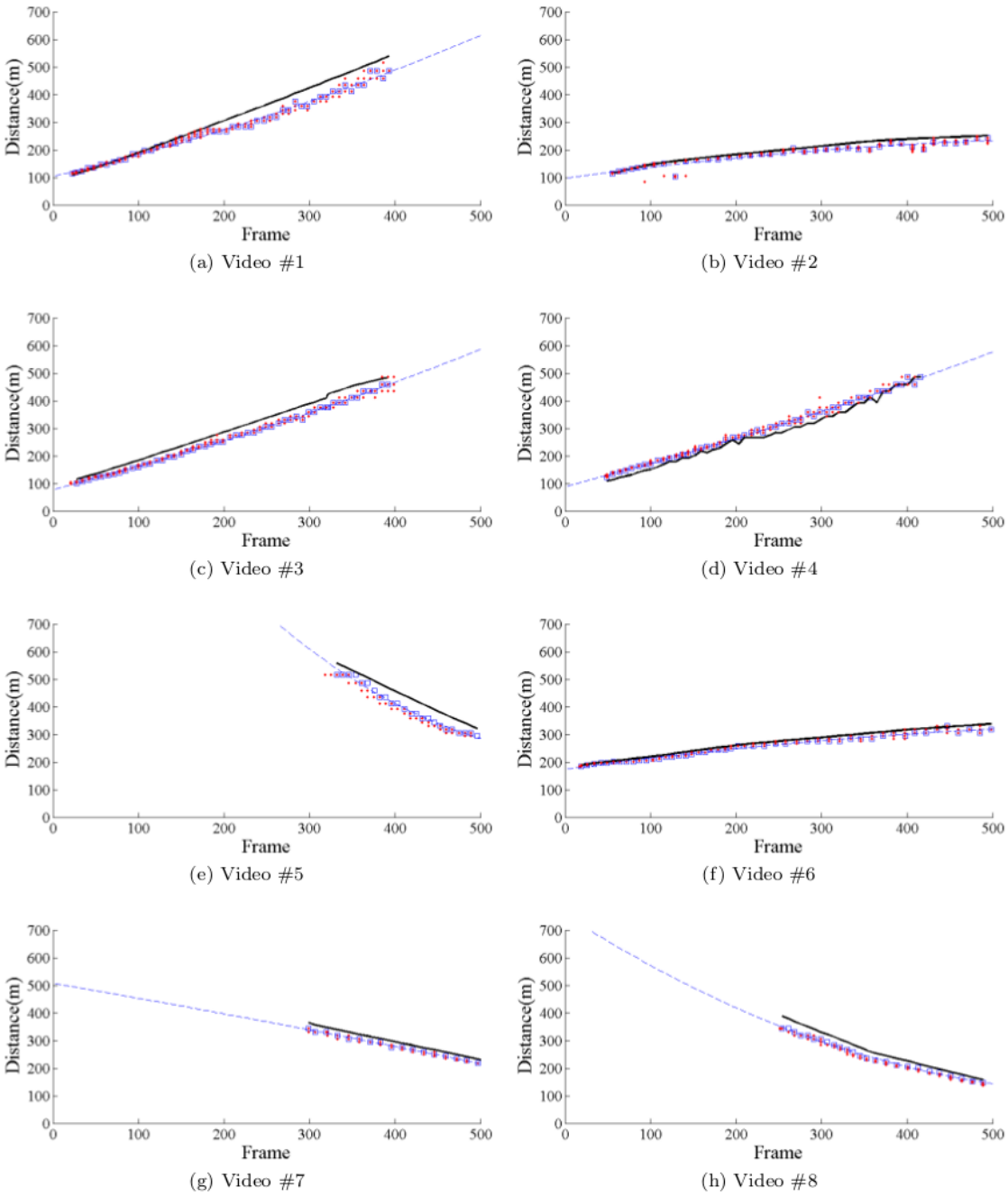


Figure 4.8: Comparison of distance estimation and ground truth for the selected eight videos. Ground truth is shown as black line, the estimated distance by red dots, and the optimised distance value as a blue mark.

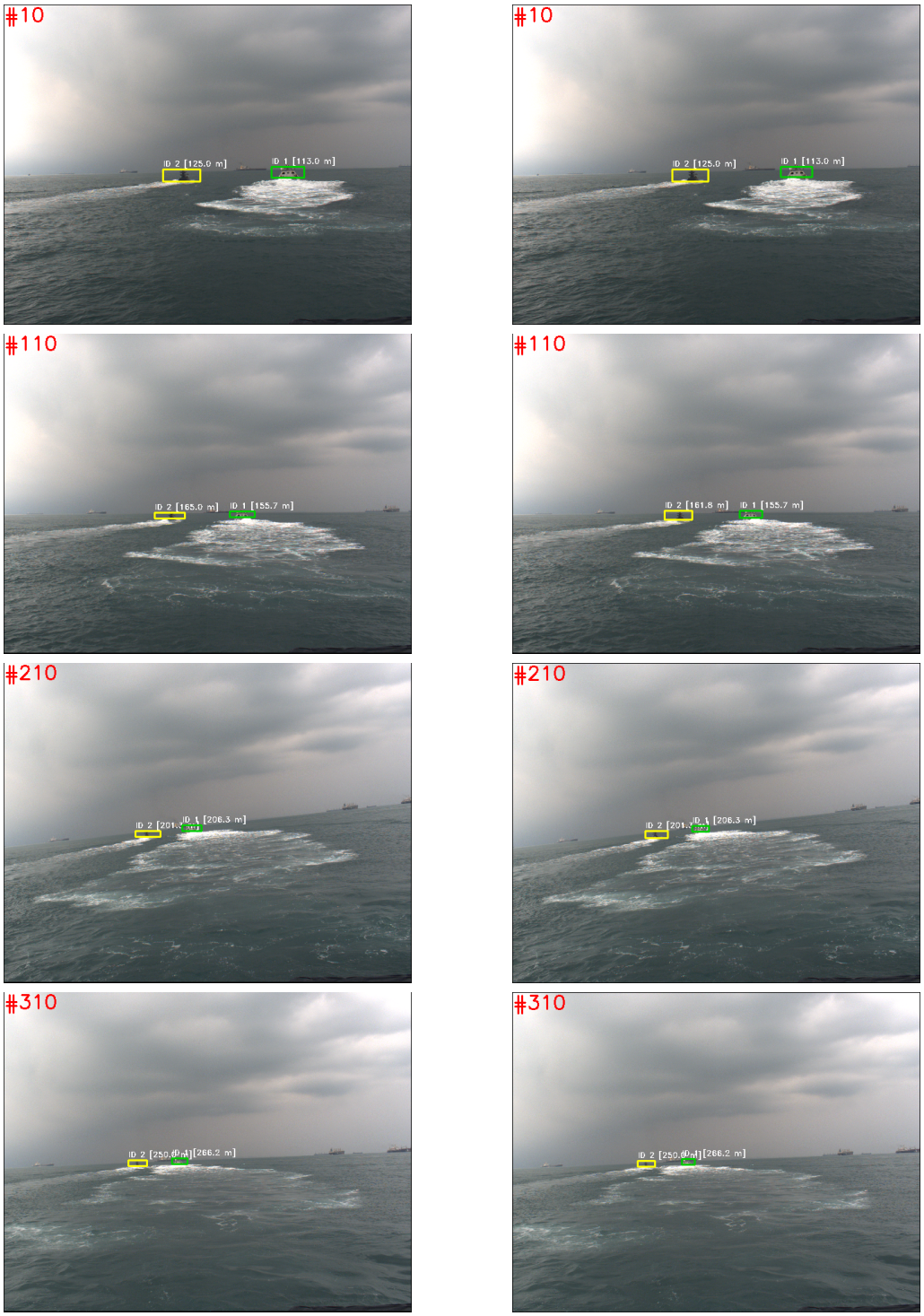


Figure 4.9: Tracking results with S_#1. The first column shows the performance of [4], and the second column shows the performance of the proposed approach.

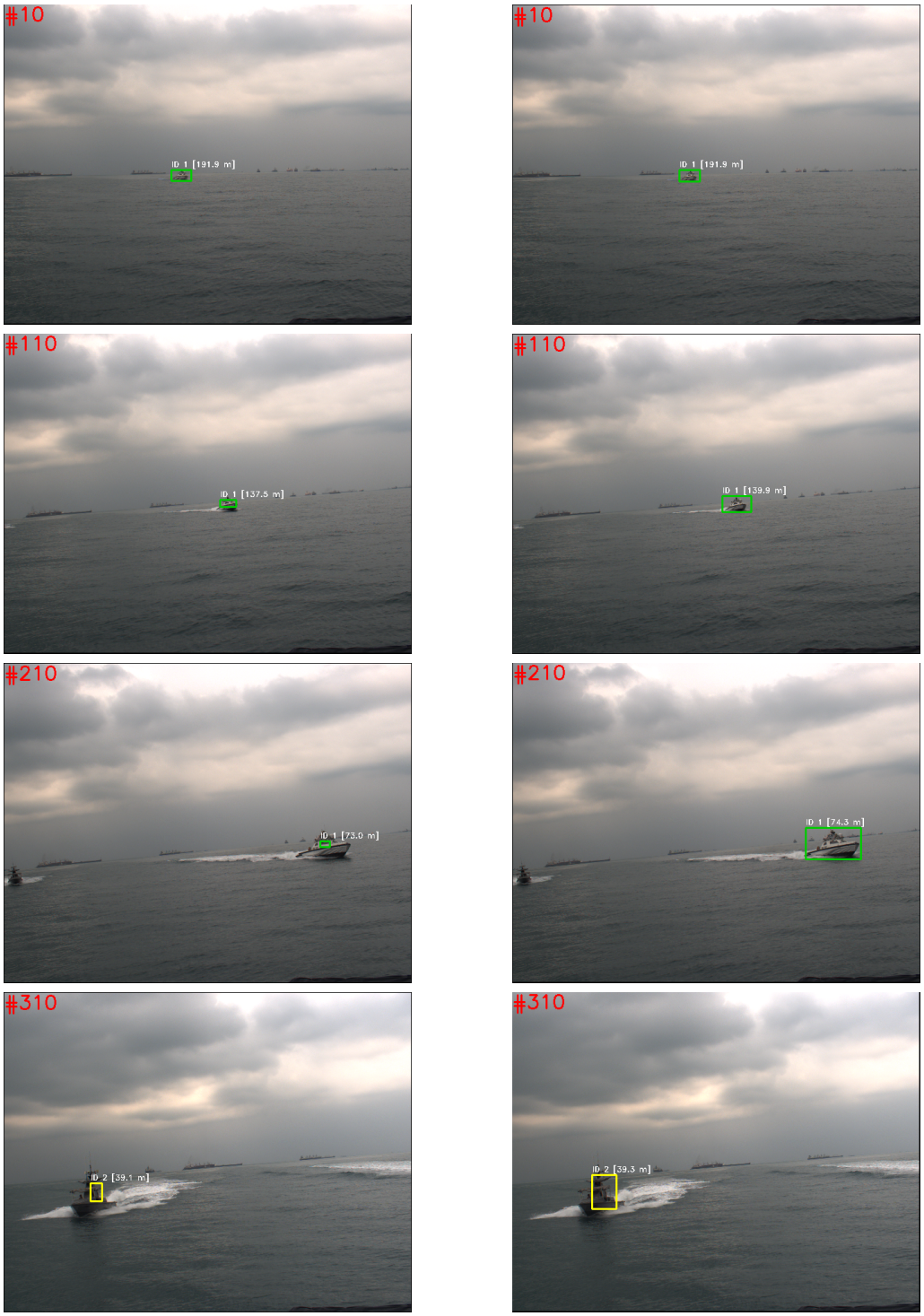


Figure 4.10: Tracking results with S_#2. The first column shows the performance of [4], and the second column shows the performance of the proposed approach.

Figure 4.11: Tracking results with S_#3. The first column shows the performance of [4], and the second column shows the performance of the proposed approach.

Chapter 5

Enhance Visual Obstacle Detection in Open Sea by Fusing 2D and 3D Clues

5.1 Motivation and Objective

In Chapter 3 and Chapter 4, it can be seen that the image-based methods (2D) and the binocular stereo vision (3D) both have their pros and cons. With 2D information, all salient candidate obstacles in an image can be detected regardless of their distances, however, noise might be included in the results, such as white wake, waves and water speckles. With 3D information, obstacles in short range can be detected with high accuracy by clustering points protrude from the sea surface, though this results in less noise, the distant obstacles are hard to be detected, because their depth may not be valid. Since 2D and 3D can compensate the drawbacks of each other, the performance of obstacle detection can be further improved by combining the advantages of 2D and 3D clues.

This chapter addresses the issue of enhanced obstacle detection for USV in the open sea environment. We propose a novel approach to improve the final detection performance by combining two candidate detection phases based respectively on 2D and 3D data in a binocular vision system. Figure 5.1 gives an overview of our approach. In the 2D candidate obstacle detection, the horizon line is first detected

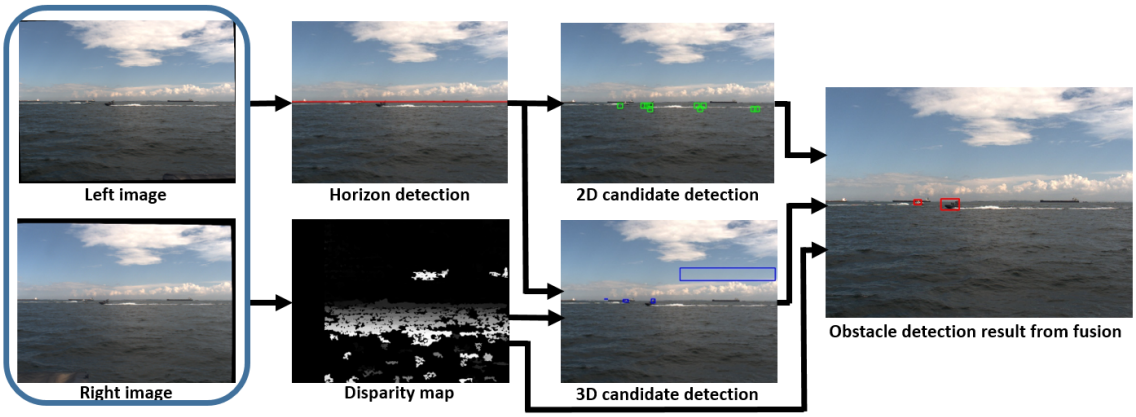


Figure 5.1: The proposed approach for obstacle detection by fusing 2D and 3D clues.

in left image with the prior knowledge of vanishing line, which is estimated from the 3D points obtained from sparse stereo matching. Thereafter, candidate obstacles are detected in the area below the horizon line using the proposed global sparsity potential (GSP). In the 3D candidate obstacle detection, a Bayesian plan-view map based method is applied on the point cloud obtained by dense stereo matching. At last, a fusion scheme that leverages both the confidence scores of candidate obstacles and their distances is proposed to achieve final results. Experiments on our own dataset demonstrate the efficiency of the proposed algorithm, which is more accurate and more robust than the merely 2D or 3D data based methods.

5.2 Horizon Detection

In this work, the horizon in the rectified left image is detected. And we model it as a straight line, because our USV is facing the open sea, and there is no distortion in the image after rectification of stereo pairs.

Basically, the proposed approach for horizon detection consists two phases: coarse estimation and fine detection. Figure 5.2 shows the flowchart of the proposed approach for horizon detection. Given a rectified stereo pair, the sparse 3D point cloud can be obtained by stereo matching the respectively detected 2D features in left and right images, and then the sea surface plane is estimated based on the derived point

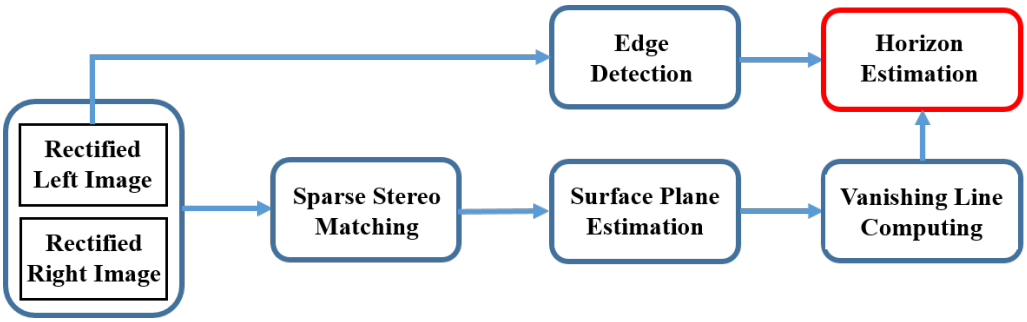


Figure 5.2: Flowchart of the proposed approach for horizon detection.

cloud. According to the perspective geometry, the vanishing line, which is a straight line that the estimated sea surface plane projecting in the left image, is computed. In the meanwhile, edge features in the left image are detected. Finally, the horizon line in the left image is estimated by combining the edge map and the vanishing line.

5.2.1 Sea Surface Plane Estimation

As is known, a plane can be only determined with three 3D points that are not co-linear. Thus, the first step to estimate the sea surface plane is getting adequate number of 3D points from the given rectified stereo pair.

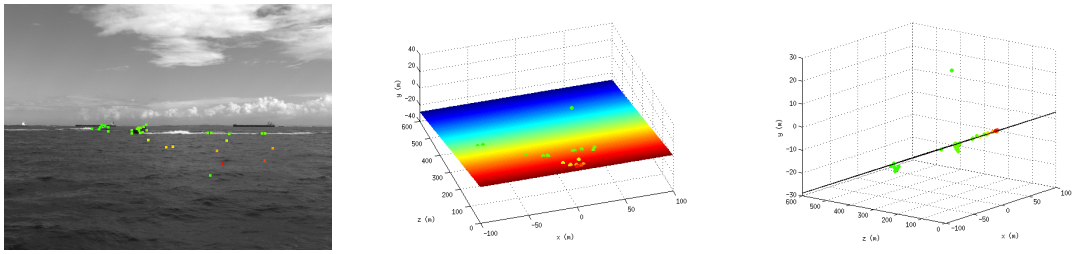
To this end, the dense stereo matching methods, such as block matching (BM) [83] and semi-global block matching (SGBM) [73], can be applied to compute the disparity map, which then renders a corresponding 3D point cloud via perspective transformation. Although the dense stereo matching algorithms generate maximum number of 3D points, there usually exist more or less error points, because the appearances or features of some image blocks from the sea surface are quite similar, which might lead to mismatching and result in 3D points with large errors. With these error points, one would not estimate a reliable sea surface plane.

The counterpart of dense stereo matching is sparse stereo matching, which first respectively detects strong features in the given stereo pair, and then finds matches in these two sets of features from the image pair, instead of looking for matches at every pixel location as the dense methods do. Hence, the sparse methods give a

spare disparity map and point cloud that only has valid values in those locations of matched features. Besides the much higher computing speed compared to the dense approaches, the sparse approaches have the advantage of less error points, since the strong features are more distinct and their matchings are more reliable.

Therefore, based on the above analyses, the sparse stereo matching methods are more suitable to the maritime scenario. Nevertheless, the conventional 2D features like FAST [86] detected in the maritime image always distribute clusteringly around the region with high intensity contrast, such as the regions of obstacles and wake, but very few features in the sea surface region, which means insufficient 3D points from the sea can we obtain to estimate the sea surface plane. The work of [87] provides a solution to this issue. In [87], a new feature detector exFAST is proposed to produce a less clustered feature distribution, together with the proposed dense consistency check during the process of feature matching, it leads to more accurate matching results. So, in this work, we apply the sparse stereo matching approach of [87] to get 3D points for estimating the sea surface plane. An example of extracted 3D points is shown in Figure 5.3(a), in which only points whose depths are within 600 meters are kept and projected to the rectified left image representing by the warm to cold colors according to their depth values from small to large.

The sea surface plane is then fitted using RANSAC [84] on the extracted 3D points. Though RANSAC is a good estimation method to exclude outlier points, its accuracy would drop when the number of outliers is large. Since the points with large depth values have little probabilities to be from the sea surface, while usually from the sky area, throwing away these points could reduce some outliers for RANSAC, and thus improve the estimating accuracy. The fitted sea surface plane from the 3D points in Figure 5.3(a) are illustrated in Figures 5.3(b) (top view) and 5.3(c) (side view). It can be observed that almost all the points from the sea surface lie in the fitted plane.



(a) (b) (c)

Figure 5.3: Sea surface plane estimation. (a) shows the extracted 3D points via sparse stereo matching; (b) and (c) respectively show the top and side views of the fitted sea surface plane from points in (a).

5.2.2 Horizon Detection Using the Prior Knowledge of Estimated Vanishing Line

Vanishing line Suppose the unit normal of the estimated sea surface plane is $\mathbf{n} = (n_1, n_2, n_3)^T$, then its vanishing line in the image plane is $n_1x + n_2y + n_3f = 0$, where f is the focal length of the camera. Theoretically, the horizon in image is just the vanishing line of the sea surface plane. However, due to the errors caused in camera calibration, feature matching or plane fitting, the estimated sea surface plane always biases the actual plane more or less, which thus results in an inaccurate vanishing line or horizon. For example, one can observe in Figure 5.4, the computed vanishing line (green) lies above the true horizon. Therefore, the computed vanishing line can not be directly taken as the horizon for the following processes. Nevertheless, we can take the computed vanishing line as a coarse estimation, since it has a high probability of closing to the true horizon.

To get a more accurate horizon, we propose a new horizon detection approach based on edge map by using the prior knowledge of computed vanishing line to restrict the searching area of horizon edges. Figure 5.4 illustrates this new approach: After obtaining the vanishing line of sea surface plane, it is placed on the Canny edge [11] map of the rectified left image as a virtual line to guide the searching of edge points that belong to the horizon; Moreover, we found that there are usually no edge features in the neighborhood of horizon edges, unless obstacles or wakes appear near the horizon. So, the edge points can be taken as being from horizon

with high probabilities as long as 1) they are within a predefined distance to the virtual line (computed vanishing line) and 2) they are the only one edge point in the local area of their columns. As shown in the top-right picture of Figure 5.4, starting from each point $p(x, y_x = -\frac{n_1}{n_2}x - \frac{n_3}{n_2}f)$ on the virtual line, we sum the pixel values $v_p(x, y) \in \{0, 1\}$ in the binary edge map at the same column x with rows $y \in [y_x - \frac{h}{2}, y_x + \frac{h}{2}]$, where $\frac{h}{2}$ is an vertical offset from point $p(x, y_x)$. This procedure can be formulated in Equation (5.1):

$$N_x = \sum_{y=y_x-\frac{h}{2}}^{y_x+\frac{h}{2}} v_p(x, y), \quad x = [1, 2, ..., W], \tag{5.1}$$

where N_x is the sum of pixel values or total number of edge points at column x and rows from $y_x - \frac{h}{2}$ to $y_x + \frac{h}{2}$; W is the width of the edge map. In the right two pictures of Figure 5.4, it can be seen that along the virtual line, edge points p_x in column x with $N_x > 1$ are abandoned, because they are most likely from obstacles or wakes and may cause a lot of noise to horizon estimation; while edge points p_x in column x with $N_x = 1$ are kept to form the candidate points set P for final horizon estimating, which is formulated as

$$P = \bigcup_{x=1, N_x=1}^{W} p_x. \tag{5.2}$$

Finally, RANSAC is applied on the candidate points P to fit the final horizon. Compared with the left two pictures of Figure 5.4, one can observe that the final detected horizon is more accurate than the computed vanishing line.

5.3 2D Obstacle Detection

In the 2D image, we constrain the detecting range for candidate obstacles to the region below the horizon. Knowing the location of horizon in Section 5.2, as shown in Figure 3.3, the region of interest (ROI) can be cropped via affine transformation. According to the perspective geometry in 2D images, the resolution of the observation that is close to the horizon is smaller than that of the observation close to the image bottom. Thereafter, similar to [2], we analyse the ROI using variable size

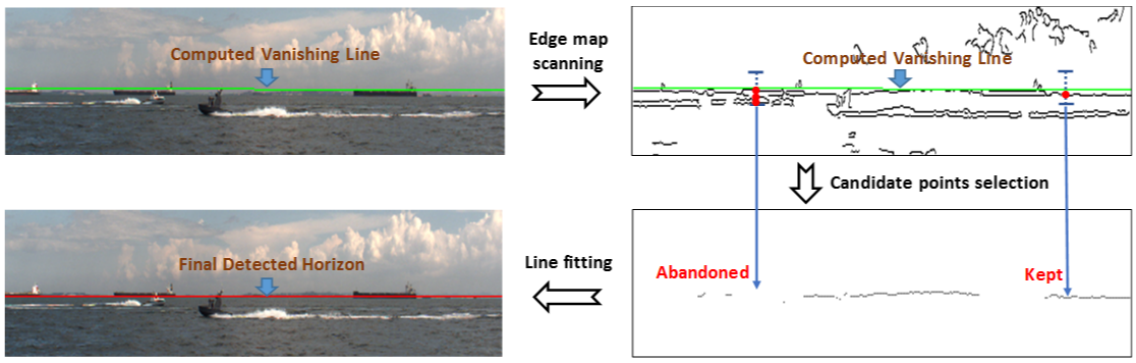


Figure 5.4: Horizon detection using the prior knowledge of estimated vanishing line.

image windows with an overlap rate of α and an expansion rate of β; the windows are square, and minimum size is $w \times w$ pixels. As shown in the middle image of Figure 3.1, each window (white rectangle) represents an image patch.

The feature of each image patch f is calculated via the gray-level co-occurrence matrix-based texture analysis [50], in which all image patches are first resized to the same size ($w \times w$) and then, f is represented by a four-dimensional vector: $f = [Energy, Entropy, Contrast, Homogeneity]$. The calculation of f is same as the definition in Chapter 3 (Equations (3.1)-(3.4)).

Global sparsity potentials . In our previous work [51] for detecting obstacles in 2D image, the sparsity of the appearance of a patch in the entire sea area instead of the local neighborhood is calculated to separate the obstacle patches from the sea patches, and we referred to this new sparsity measure as global sparsity potentials (GSPs). The GSP of an image patch in [51] is measured as the average distance to all other patches, and then it compute the mean feature of patches with smaller GSP values, finally, patches with larger distance to the mean are taken as obstacles. One drawback of [51] is that it is not easy to find a threshold to separate the outliers, because the mean GSP distance varies much with the number of outliers.

Different from [51], in this work, we set a threshold τ_1 to check the similarities of each patch to the whole patches set, and then the GSP of each patch is computed by taking the average of all its similarity measures. Equations (5.3) - (5.5) formulate

this process.

The GSP G_i of the ith image patch is measured as

$$G_i = \frac{1}{N-1} \sum_{j=1, j \neq i}^{N} \omega_{ij}, \tag{5.3}$$

where N is the total number of patches sampled in an image; ω_{ij}, as formulated in Equation 5.4, is the similarity measure of the ith image patch to the jth image patch.

$$\omega_{ij} = \begin{cases} 1, & \text{if } D(f_i, f_j) > \tau_1; \\ 0, & \text{otherwise.} \end{cases} \tag{5.4}$$

where τ is a positive value threshold; f_i and f_j are respectively the features of the ith patch and the jth patch; $D(\cdot)$ is the Mahalanobis distance between two features, and defined as

$$D(f_i, f_j) = \sqrt{(f_i - f_j)^T C^{-1} (f_i - f_j)}, \tag{5.5}$$

in which C is the covariance matrix of the features set in an image.

From Equations (5.3) - (5.5) one can observe that the larger GSP value G_i of a patch, the more sparsity or distinct the patch is. Instead of taking the mean distance of a patch to the whole patches set as the GSP measure as [51] did, it is more robust to measure the GSP of a patch by its similarity rate to the whole patches set as formulated in Equation (5.3). Finally, patches with $G_i > \tau_2$ are kept as candidate for obstacles in 2D image.

5.4 3D Obstacle Detection

The stereo rig used in this chapter is same as that in Chapter 4, so we adopt the similar process for 3D candidate obstacle detection as presented in Section 4.2.2. Since images with normal size are processed in this chapter, rather than the HD images used in Chapter 4, here we do not need to down sample the original image, neither the fine obstacle detection as that Section 4.2.3 does. Following the procedures described in Section 4.2.2, we can obtain the corresponding bounding boxes for detected 3D candidate obstacles.

5.5 Fusion of 2D and 3D Clues

In Section 5.3, we have obtained the candidate bounding boxes for obstacles in 2D image, and in Section 5.4, we have the depth map corresponding to the 2D image. Thus, by placing those candidate bounding boxes in 2D image to the depth map, one can get the depth value of each pixel in the original bounding boxes. Thereafter, the mean depth $\bar{d}_i$ and its depth variance σ_i^2 of each bounding box can be calculated.

Intuitively, a bounding box on an obstacle should have similar depth values for each pixel in it, which means smaller depth variance. On the contrary, if the depth values in a bounding box vary much, i.e. larger depth variance, that means the bounding box has a low probability to contain an obstacle, but may contain the water which caused wrong stereo matching, or two obstacles far away from each other. So the bounding boxes with large depth variances are rejected. Nevertheless, although some bounding boxes have smaller depth variance, they may contain a number of invalid depth values, which means points are too far away from the camera, and such bounding boxes are also rejected. Then, the retained bounding boxes are grouped if they have big enough overlap with others.

The confidence score ζ_{2D} of a detected obstacle in 2D detection is defined as

$$\zeta_{2D} = a \sum_{i=1}^{N_{2D}} \frac{1}{\sigma_i^2} G_i, \tag{5.6}$$

where a is a constant positive value; N_{2D} is the total number of bounding boxes grouped for this detected obstacle; σ_i is the stand deviation of depth in each bounding box; G_i is the GSP of each bounding box, and is defined in Equation (5.3). From Equation (5.6), one can observe that ζ_{2D} is not only determined by the GSPs of 2D image patches, but also the depth variance derived from the 3D cue. A 2D detecting result with more distinct appearance (larger G_i) and more smooth depth (smaller σ_i) gives more confidence to be an obstacle.

The confidence score ζ_{3D} of a detected obstacle in 3D detection is defined as

$$\zeta_{3D} = b \sum_{i=1}^{N_{3D}} \varrho_i, \tag{5.7}$$

where b is a constant positive value; N_{3D} is the total number of 3D points projected back in the 3D detected bounding box of an obstacle; ϱ_i is the probability value

of each 3D point, which is the same value as computed in Equation (4.8). Hence, the more 3D points with high probabilities a detected 3D obstacle has, the larger confidence score is given to it. In other words, a detected 3D obstacle with few number of high probability points might be a noise, which can be rejected due to a small confidence score.

The fusion of detecting results from 2D and 3D is done if there exists a correspondence between them, i.e. a big enough overlap between their bounding boxes. If a 3D detection does not have its 2D detection correspondence, then the confidence score of 2D is set to 0, and vice versa. As is known, scenes nearby show higher definition in a 2D image, while scenes far away show lower definition. Thus, the reliability of an obstacle detected in 2D can be modeled by direct proportional to its distance. For instance, an obstacle detected in 2D with short distance might be a patch from the sea wave, which is very easy to be mistaken as obstacles in nearby view. On the contrary, far away obstacles with low definition are less likely to be contaminated by noise, so more reliable. On the part of 3D, things are opposite. 3D points at short distance are more reliable than the ones with long distance, because the stereo matching needs strong information or high definition for robust and accurate performance. So the reliability of an obstacle detected in 3D can be modeled by inverse proportional to its distance.

The confidence score ζ_{fusion} of fusion is thereby defined as

$$\zeta_{fusion} = \lambda d_{2D} \zeta_{2D} + \frac{\eta}{d_{3D}} \zeta_{3D}, \tag{5.8}$$

where λ and η are respectively two constant positive values; d_{2D} is the average distance of the projected 3D points in an obstacle of 2D detection; d_{3D} is the average distance of 3D points in an obstacle of 3D detection.

At last, the final detecting results are obtained by merging the detections of 2D and 3D only when $\zeta_{fusion} > \tau_{fusion}$, where τ_{fusion} is a constant real value threshold.

5.6 Experimental Results

The stereo vision system of this project is composed by a pair of Point Grey CCD cameras with focal length in pixels of 704.7 and 706.2 respectively. Their average

are taken as the focal length of the stereo in this work, i.e. $f = 705.5$. The intrinsic parameters of cameras are calibrated using the conventional checkerboard method. The extrinsic parameters of cameras are calibrated using the method of [72] due to the large baseline $2.90m$, which is manually measured by a ruler. All experiments in this part were conducted on a PC equipped with i7, 3.40GHz CPU.

5.6.1 Dataset

Since there are few available public datasets for stereo vision based obstacle detection in maritime scenes, we built our own dataset. Our stereo vision based obstacle detection dataset consists two sequences (S_#1, S_#2). Each sequence contains 300 pairs of RGB frames with size 684×548. The obstacles in this dataset are two moving target boats, which have both short and long distances (from 100m to 500m or so) to the camera boat (USV). The sequences render different challenges for the task of obstacle detection:

- S_#1 characterizes the detection ability in long distance. The two target boats move towards to each other, and less white wake appears in images due to the long distance.
- S_#2 contains a lot of white wake noise generated by the fast moving target boats, whose distances are around $100m$. Some frames in this sequence have no obstacles.

The ground truth of obstacles in each left image are manually labelled with rectangle bounding boxes.

5.6.2 Performance Evaluation

Same as [88], the results of detection are considered as true or false positives according to their overlap rate with the labelled ground truth. This overlap rate γ is defined as

$$\gamma = \frac{B_d \cap B_{gt}}{B_d \cup B_{gt}}, \tag{5.9}$$

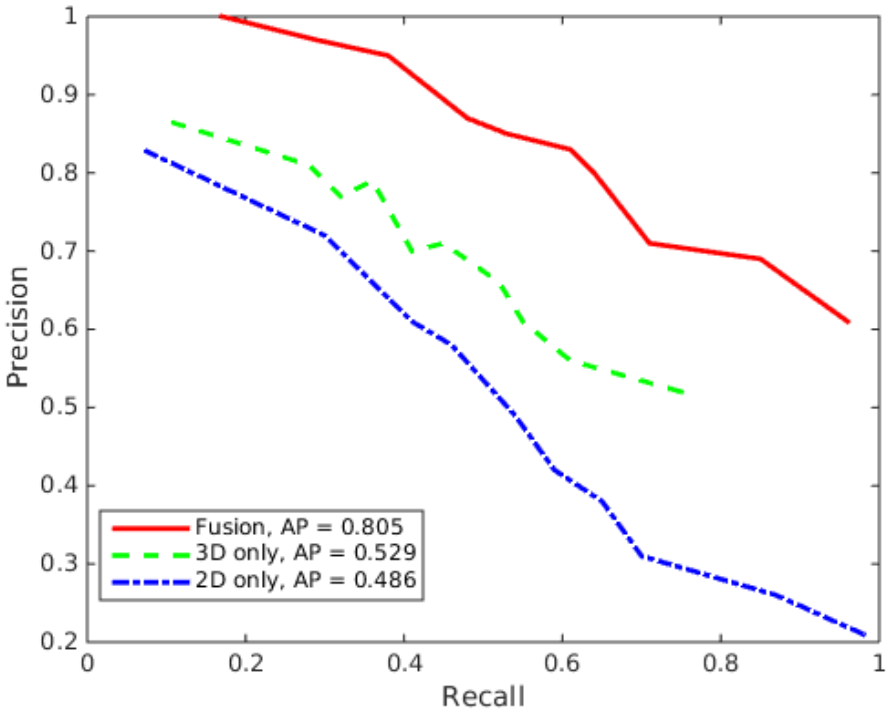


Figure 5.5: Precision-Recall curves and AP scores for obstacle detection on our dataset using different methods.

where B_d is the bounding box of the detected obstacle, and B_{gt} is the bounding box of labelled ground truth. A correct detection is assigned when $\gamma \geq 0.5$.

Figure 5.5 shows the comparison of Precision-Recall curves and average precision (AP) scores for obstacle detection results on our own dataset using the 2D only detection (blue line), 3D only detection (green line), and the proposed fusion method (red line). It can be seen that our proposed fusion method outperforms both the 2D only detection and 3D only detection.

Figure 5.6 illustrates the performance of these three methods. The first row shows the single obstacle scenario; The second row is the case of two obstacles are close to the camera, and many white wake noise appear; The third row shows the long distance case, where relative few white wake appear; The forth row exhibits the case when two obstacles are close to each other; The last row shows the scenario of no obstacle appearing in the image. It can be observed that there are many false alarms

in the 2D only detection (first column), and the 3D only detection (second column) always can not contain the whole obstacle. On the contrary, the proposed method that combines the advantages of both 2D only and 3D only methods, exhibits better performance than the other two.

5.7 Concluding Remarks

A novel approach of enhancing vision based obstacle detection for USV in open sea environment by fusing 2D and 3D clues has been presented in this paper. We proposed to combine the 2D and 3D clues in a weighting model, which gives more weights to the 2D detecting results when obstacles are distant, while gives more weights to the 3D detecting results when obstacles are nearby. The final detecting results are obtained when the confidence score of fusion is larger than a predefined threshold. In experiments, this new approach demonstrates better performance with higher accuracy and robustness than the methods based only on 2D or 3D information. Besides the fusion framework, we also made contributions in horizon detection and candidate obstacle detection in 2D image. The horizon detection is enhanced by using the prior knowledge of estimated vanishing line, and a new GSP measure is proposed to improve the robustness of 2D candidate obstacle detection.

This approach is tailored for the open sea environment, in which the horizon can be seen and usually small number of obstacles appear in images. To perfect the algorithm, a detection including the coastal lines would be studied in the future work. Moreover, the restriction of the requirement for horizon detection would be released with more powerful 2D detection methods.

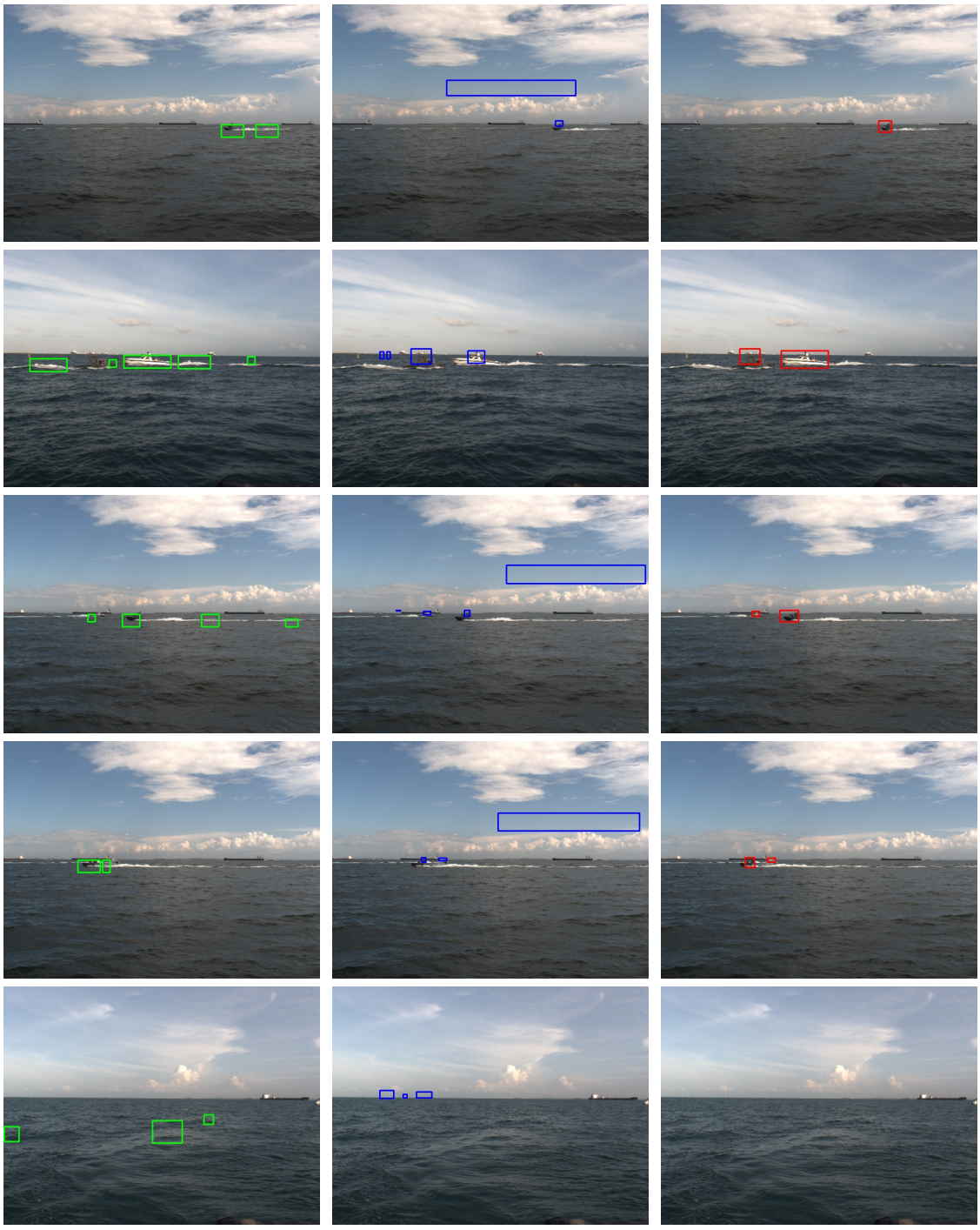


Figure 5.6: Obstacle detection results using only 2D image (first column, green), only 3D point cloud (second column, blue), and the proposed approach of fusion with 2D and 3D clues (third column, red).

Chapter 6

Monocular Vision Based Obstacle Mapping for Unmanned Surface Vehicles

6.1 Motivation and Objective

In a binocular stereo vision system, the depth Z of a point is computed as

$$Z = \frac{fB}{d}, \tag{6.1}$$

where B is the baseline; f is the focal length of the camera in pixels; d is the disparity value. Assuming there is a small disparity error Δd added to the disparity d, then the depth error ΔZ is derived as

$$\Delta Z = \frac{Z^2}{fB}\Delta d. \tag{6.2}$$

In Equation (6.2), one can see that the depth error ΔZ is inverse proportional to the length of baseline B by supposing the focal length f of the camera is fixed. In other words, larger baseline yields smaller depth error at a same depth value. Thus, as in Equation (6.1), under the same disparity d, a larger depth Z can be obtained with a larger baseline B.

In Chapter 4 and Chapter 5, we have built the binocular stereo vision system with very large baseline B and focal length f, and the ranging ability is achieved

within 500 meters. However, to further enlarge the ranging ability within 1,000 meters, it is really hard to continue developing the binocular stereo vision system, which would result in an even bulky structure on the boat and a big problem for calibration. Therefore, to get rid of these problems, in this Chapter, we propose a motion stereo vision based system, in which only one camera is used and the depth is restored via the USV's own motion. The motion of USV can provide large enough baseline, so that the ranging ability can even reach $1,000+$ meters.

After reconstructing the points from obstacles, it is necessary to build an obstacle map for the navigation of USV. Conventional SLAM methods [76, 78] applied in UGV and UAV assume plenty features can be detected in each frame, however, this is not suitable to the open sea environment, which usually has very few features can be detected. In this chapter, we propose a new solution to compute the visual odometry for triangulating and mapping points from obstacles in the maritime environment. The translation of the camera is computed using recorded GPS data. The rotation of the camera is recovered by the detected horizon line (roll and pitch) and a compass or IMU (yaw). Then, by matching the detected feature points between two frames, the depth of each feature point can be calculated. To achieve more accurate and robust performance, multiple frame pairs are leveraged to synthesize the final reconstruction results in a weighting model. Finally, these feature points from obstacles are projected to a Google map. Experimental results demonstrate the efficiency of our system. To the best of our knowledge, we are the first to handle the task of monocular vision based obstacle mapping in maritime environment.

6.2 Motion Parallax

In a monocular vision based system, a 3D motion of the camera causes an effective translation and rotation of the object relative to the camera, and the resulting image motion of points and lines reveal their 3D geometries. This fact is known as motion parallax, which is the main theory behind our proposed system.

As shown in Figure 6.1, suppose the camera mounted on a vehicle undergoes a translation $\boldsymbol{h}$ followed by a rotation $\boldsymbol{R}$ from O to O'. A world point P has been observed twice as (x, y) and (x', y') respectively in the images. Therefore, we have

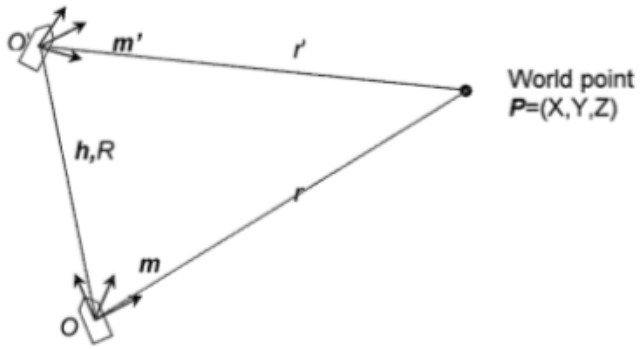


Figure 6.1: Illustration of motion parallax.

the following vector relation:

$$\vec{O'P} = \boldsymbol{R^{-1}}(\vec{OP} - \vec{OO'}). \tag{6.3}$$

For convenience, we convert the image points in O and O' into unit vectors as

$$\boldsymbol{m} = \frac{1}{\sqrt{x^2 + y^2 + f^2}} \begin{pmatrix} x \\ y \\ f \end{pmatrix}, \tag{6.4}$$

$$\boldsymbol{m'} = \frac{1}{\sqrt{x'^2 + y'^2 + f^2}} \begin{pmatrix} x' \\ y' \\ f \end{pmatrix}. \tag{6.5}$$

Thus, Equation (6.3) can be rewrote as

$$r'\boldsymbol{m'} = \boldsymbol{R^{-1}}(r\boldsymbol{m} - \boldsymbol{h}), \tag{6.6}$$

where r and r' are the distances between the world point P and the camera at location O and O' respectively. As proven in [89], solving Equation (6.6) using the least square method, we can find the optimal solution for r and r' that minimizes the error between the two sides of the equation. The final solution is obtained as

$$r = \frac{(\boldsymbol{h}, \boldsymbol{m}) - (\boldsymbol{m}, \boldsymbol{Rm'})(\boldsymbol{h}, \boldsymbol{Rm'})}{1 - (\boldsymbol{m}, \boldsymbol{Rm'})^2}, \tag{6.7}$$

$$r' = \frac{(\boldsymbol{m}, \boldsymbol{Rm'})(\boldsymbol{h}, \boldsymbol{m}) - (\boldsymbol{h}, \boldsymbol{Rm'})}{1 - (\boldsymbol{m}, \boldsymbol{Rm'})^2}, \tag{6.8}$$

where $(\cdot, \cdot)$ is the inner product of two vectors.

With Equations (6.11) and (6.12), the same feature point appears in the previous frame and the current frame can be reconstructed respectively. The following issues are that how to retrieve the translation vector $\boldsymbol{h}$, rotation matrix $\boldsymbol{R}$, and unit vectors $\boldsymbol{m}$ and $\boldsymbol{m'}$.

6.3 Visual Odometry

Visual odometry (VO) refers to computing the translation and rotation of a camera mounted on a vehicle by analysing the associated camera images. Conventional methods like 8-point algorithm [90], 7-point algorithm [91], 5-point algorithm [92], RANSAC [84], etc., are all based on image feature detection and matching to compute the essential matrix, from with the rotation and translation (up to scale) are then recovered. Although these methods are widely used in UGV and UAV nowadays, they need plenty strong features to be detected and matched to render a good estimation of VO. However, in the case of sea environment, there is usually very few features can be detected, because the sea water and sky take most part of the image, and their appearance are mostly uniform. Even some features can be detected on the wave, they are not stable and give unreliable feature matches. For features detected on the sky (clouds), though they can be taken as stable points since the clouds move very slow and can be considered as stationary in a short time period, they are too far away and hard to accurately recover the translation of the camera. Therefore, the conventional VO methods are not suitable to a USV in the sea environment.

Using measurement sensors of GPS and IMU, we can directly get the motion of the camera. Nevertheless, the accuracy of these sensors are not that high. To solve this problem, we propose using horizon line to recover the roll and pitch angles in the rotation. The yaw angle is measured by our IMU, which works as a compass. The translation of the USV is computed directly from the recorded GPS data. To reduce the error produced by GPS, we take two frames with a large baseline (translation) to reconstruct the feature points, so that the influence of GPS error can be weakened.

In the IMU (myAHRS+) used in our system, the yaw angle is measured by

(a) IMU (b) Horizon line

Figure 6.2: Visual comparison of roll correction using IMU and horizon line.

magnetometer sensor, which is more reliable. While the roll and pitch angles are measured by gyroscope and accelerometer, which are less reliable through our comparison in experiments. Figure 6.2(a) shows the roll angle corrected image using the measurement from IMU. It can be seen that the horizon line in the image is not horizontalized well.

To compensate the low accuracy of roll and pitch angles rendered by IMU in our system, we proposed to compute the roll and pitch from the horizon line. Therefore, the first step is to accurately detect the horizon in the image. In this work, structured edge detection [93] is applied to derive the edge map of the original image. After that, RANSAC method is used to fit a straight line (horizon) from the edge points.

As shown in Figure 6.3, with the horizon or vanishing line, the normal vector $\boldsymbol{n}$ of the sea surface plane in camera coordinate can be calculated, and it is same with the normal vector in world coordinate. Denote the image center as (x_c, y_c); focal length of the camera f; two different points (x_1, y_1) and (x_2, y_2) on the horizon line. Thus, the two vectors connecting the origin of camera coordinate and the two different points on horizon line are respectively $\boldsymbol{v_1} = (x_1 - x_c, y_1 - y_c, f)^T$ and $\boldsymbol{v_2} = (x_2 - x_c, y_2 - y_c, f)^T$. Hence, the normal vector of the sea surface plane is calculated as the cross product of $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$: $\boldsymbol{n} = \boldsymbol{v_1} \times \boldsymbol{v_2} = (n_1, n_2, n_3)^T$. Finally, the roll α and pitch β angles of the camera can be computed as

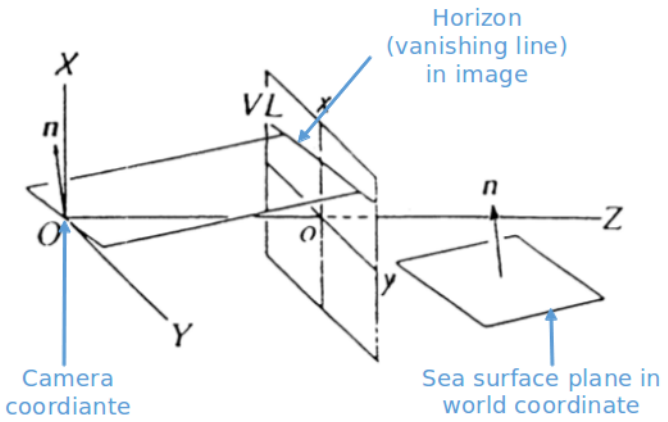
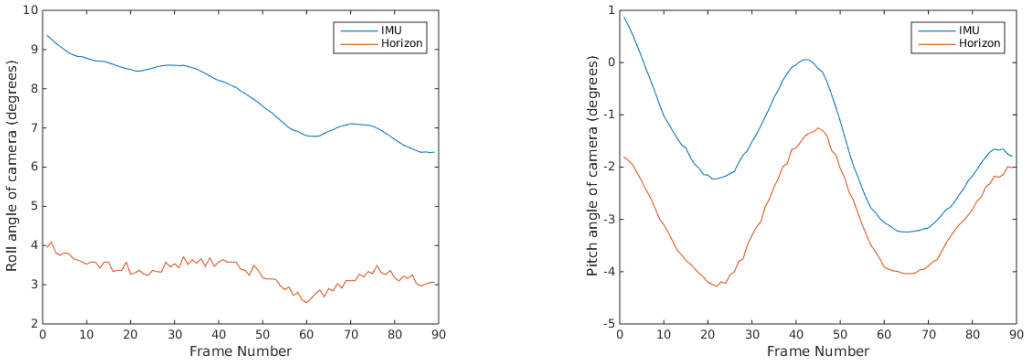


Figure 6.3: Sea surface plane estimation from horizon line.



(a) Roll

(b) Pitch

Figure 6.4: Roll and pitch angles of camera obtained from IMU and horizon line.

$$\alpha = \arctan(\frac{n_1}{n_2}), \tag{6.9}$$

$$\beta = \arctan(\frac{n_3}{\sqrt{n_1^2 + n_2^2}}). \tag{6.10}$$

Figure 6.4 illustrates the comparisons of roll and pitch angles obtained from IMU and horizon line. One can see that there are large gaps between the rotation measurements from IMU and horizon line. Furthermore, It can be seen in Figure 6.2, the image after roll correction using horizon line are much better than that using IMU. Therefore, in this work, we chose to derive the roll and pitch angles of the camera mounted on USV from horizon line instead of IMU.

6.4 From 3D to 2D

After getting the roll and pitch angles of the camera, with respect to which one can correct the image so that the roll and pitch angles equal to 0. That means the USV is moving on a 2D flat plane. Thereby, our problem can be simplified to the 2D case by the roll and pitch correction. On the 2D plane, there exists only the yaw rotation between two frames.

Suppose the rotation matrices for roll and pitch correction in the previous and the current states are $\boldsymbol{R_1}$ and $\boldsymbol{R_1}'$ respectively; The relative rotation matrix of yaw angle from previous state to current state is $\boldsymbol{R_2}$. Then the Equations (6.11) and (6.12) can be rewrote as

$$r = \frac{(\boldsymbol{h}, \boldsymbol{R_1 m}) - (\boldsymbol{R_1 m}, \boldsymbol{R_2 R_1' m'})(\boldsymbol{h}, \boldsymbol{R_2 R_1' m'})}{1 - (\boldsymbol{R_1 m}, \boldsymbol{R_2 R_1' m'})^2}, \tag{6.11}$$

$$r' = \frac{(\boldsymbol{R_1 m}, \boldsymbol{R_2 R_1' m'})(\boldsymbol{h}, \boldsymbol{R_1 m}) - (\boldsymbol{h}, \boldsymbol{R_2 R_1' m'})}{1 - (\boldsymbol{R_1 m}, \boldsymbol{R_2 R_1' m'})^2}. \tag{6.12}$$

In Equations (6.11) and (6.12), $\boldsymbol{h}$ is the translation vector in camera coordinate, while we can only get the translation vector $\boldsymbol{H}$ in world coordinate from GPS. Consider the 2D case and suppose $\boldsymbol{h} = (X_c, Z_c)^T$, $\boldsymbol{H} = (X_w, Z_w)^T$. Knowing the yaw angle θ of the camera to the world North from the compass, $\boldsymbol{h}$ can be computed as

$$\boldsymbol{h} = \begin{bmatrix} X_c \\ Z_c \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} X_w \\ Z_w \end{bmatrix} \tag{6.13}$$

6.5 Feature Matching

To find the matched feature points in a pair of images, it generally goes three steps: feature detection, feature tracking, and feature matching.

Feature Detection. To detect features from the image, we use ORB (oriented FAST and rotated BRIEF) [77], which is basically a fusion of FAST keypoint detector [94] and BRIEF descriptor [95] with many modifications to enhance the performance. It first uses FAST to find keypoints, then applies Harris corner

Figure 6.5: ORB feature detection, tracking, and matching. The straight lines in the image connecting the matched features.

[96] measure to find top N points among them. It also uses pyramid to produce multiscale-features. Its descriptor is the modified BRIEF descriptor.

Feature Tracking. The detected features are tracked by calculating their optical flow using the iterative Lucas-Kanade method with pyramids [40].

Feature Matching. To prevent the feature drift during tracking, we do feature matching for the tracked features in the pair of images that are selected to reconstruct the feature points. In this way, the drifted features can be filtered out, and only features with strong matching (closer distance) are retained. The matching process is in a brute-force manner.

Figure 6.5 shows the final matched features in a pair of images. It can be observed that the features are mainly located at the strong corners of the obstacle.

6.6 Obstacle Mapping

Points reconstructed from the current frame with only one pair of frames may not be reliable or accurate enough, due to the errors come from different sources, including feature matching, horizon detection, GPS and IMU sensors. To improve the accuracy of reconstruction, we propose to do the points reconstruction in current frame with multiple pairs of frames.

Denote the current frame as I_c, and several previous frames as $I_i, i = (1, 2, ..., m)$, where m is the total number of selected previous frames. With each pair of frames $\{I_c, I_i\}$, the depth r_{ci} and r'_{ci} of a pair of matched feature points in the previous and the current frame can be triangulated using Equations (6.11) and (6.12), respectively. Knowing the locations in world coordinate of the pair of frames from GPS, together with r_{ci} and r'_{ci}, the 3D location P_{ci} of this matched feature point in world coordinate can be calculated by basic techniques of triangle geometry.

Therefore, in total we can get m measurements for a same feature point in the current frame. Then, we give each of the m measurements a weight ω_{ci}, and the final determined 3D location P_c of this feature point is synthesized by computing the weighted sum of the m measurements. This process is formulated in Equation (6.14).

$$\begin{cases} P_c = \sum_{i=1}^{m} \omega_{ci} P_{ci}, \\ \sum_{i=1}^{m} \omega_{ci} = 1. \end{cases} \tag{6.14}$$

Finally, the obstacle map is built by projecting the reconstructed 3D points using Equation (6.14) to a grid map or Google map.

6.7 Experimental Results

The proposed motion stereo system in this work is composed by one Point Grey grasshopper3 CCD camera with image size 2736×2192, frame rate 13Hz, and focal length 2840; Camera lens with horizontal field of view (FOV) 50 degrees; A Trimble SPS351 GPS unit with frequency 10Hz; A myAHRS+ sensor module with frequency 100Hz; A laptop equipped with i7, 2.60GHz CPU.

Since the frequency of GPS is lower than that of the camera, we do a linear interpolation as formulated in Equation (6.15) so that each frame has a GPS data.

$$L_{img} = \frac{T_{img} - T_{GPS1}}{T_{GPS2} - T_{GPS1}} (L_{GPS2} - L_{GPS1}) + L_{GPS1}, \tag{6.15}$$

where L_{img} is the interpolated location of current frame; T_{img} is the time stamp of current frame; T_{GPS1} and T_{GPS2} are the GPS time stamp that are before and after T_{img}, respectively; L_{GPS1} and L_{GPS2} are the respective GPS locations at T_{GPS1} and T_{GPS2}.

The myAHRS+ sensor has a quite high frequency, so we just find the yaw data that has closest time stamp to that of the video frame, and assign it to that video frame.

6.7.1 Dataset

There is no public dataset for vision based SLAM in maritime environment, so we evaluate the proposed algorithm on our own dataset, which includes three image sequences (Seq-1, Seq-2, Seq-3) captured using the proposed motion stereo system from a moving USV in open sea. The dataset is characterized as follows:

- Seq-1 contains 100 continuous gray frames; A stationary buoy with about 100 meters away from the USV appears in each frame.
- Seq-2 contains 90 continuous gray frames; A stationary tanker with about 500 meters away from the USV is shown in each frame.
- Seq-3 is the full sequence, from which Seq-1 and Seq-2 are cut out. This sequence contains 1,185 continuous gray frames, and presents the scenes while the USV is travelling along a circle route on the sea.

Seq-1 and Seq-2 are used for quantitative evaluation of the stability of our proposed reconstructing approach that leverages multiple frame pairs. Seq-3 is used for visual evaluation of the performance of the proposed obstacle mapping with a full moving loop of USV.

Currently, we do not have ground truth locations of the obstacles in this dataset, so we just evaluate the stability of our propose approach.

6.7.2 Performance Evaluation

In this part, we evaluate the stability of our proposed system by examining the variance of the mapped feature points in our dataset. In the following experiments, we take equal weights in Equation (6.14) for a simple evaluation.

First, we need to find a good value of m in Equation (6.14), so that the reliable reconstruction results can be obtained. Table 6.1 shows our trials with Seq-1. We

Table 6.1: Reconstructed location (latitude and longitude) variances of two feature points in **Seq-1** using different number (m) of frame pairs.

	Feature Point #1		Feature Point #2	
m	Var Lat ($\times 10^{-10}$)	Var Lon ($\times 10^{-10}$)	Var Lat ($\times 10^{-10}$)	Var Lon ($\times 10^{-10}$)
1	0.2211	0.3366	0.1907	0.2953
5	**0.1764**	0.2758	**0.1679**	**0.2568**
10	0.2062	**0.2723**	0.2232	0.2573
15	0.2483	0.2733	0.2939	0.2741

Table 6.2: Reconstructed location (latitude and longitude) variances of two feature points in **Seq-2** using different number (m) of frame pairs.

	Feature Point #1		Feature Point #2	
m	Var Lat ($\times 10^{-7}$)	Var Lon ($\times 10^{-7}$)	Var Lat ($\times 10^{-7}$)	Var Lon ($\times 10^{-7}$)
1	0.0555	0.9332	0.0288	0.8539
5	0.0291	0.5546	0.0140	0.4926
10	**0.0194**	**0.4209**	**0.0095**	**0.3895**
15	0.0441	0.7795	0.0226	0.7079

select two feature points from the obstacle in image, and each of these two features are tracked frame by frame and reconstructed to the geodetic coordinate (latitude and longitude) with different number of frame pairs. Finally, the variances of the reconstructed locations with each feature in the latitude (Var Lat) and longitude (Var Lon) dimensions are calculated, respectively. One can see from Table 6.1 that the reconstruction with $m = 5$ and $m = 10$ pairs of frames get smaller variances than that with $m = 1$ or $m = 15$. The reason is that the reconstruction using only one pair of frame is not reliable enough, while using too many pairs of frames may introduce some noises to the computing.

The top row of Figure 6.6 illustrates the two features mentioned in Table 6.1. As is seen, the two features are extracted from different parts of the buoy, and their

estimated distances are displayed. Comparing the distributions of mapped feature points in the middle row to that in the bottom row, we can observe that the feature maps reconstructed from 5 pairs of frames are less scattered than that with one frame pair.

Similar to the experiments in Seq-1, we also tried different m values with Seq-2. It can be seen in Table 6.2 that smaller location variances can be obtained by using 10 pairs of frames for reconstruction. Furthermore, Figure 6.7 shows the two features in Table 6.2, and compares the distributions of mapped points using one frame pair (middle row) with that using 10 frame pairs (bottom row), which again demonstrates less scattered results.

Figure 6.8 and 6.9 show the final results for obstacle mapping with Seq-1 and Seq-2, respectively. It compare our proposed approach that leverage multiple frame pairs to reconstruct the feature points with that using single frame pair. The difference is hard to be seen in Figure 6.8, because the approach is more reliable to the near-field obstacles, which have a larger position difference in the frame pairs, so be more tolerant to noise. Thus, the difference between using single frame pair and using multiple frame pairs is very small, but in Table 6.1, we can see this difference. The obstacle in Seq-2 is a distant tanker, so it can be observed in Figure 6.9 that the distribution of reconstructed features points using multiple frame pairs is less scattered than that using single frame pair.

The resulted obstacle map with Seq-3 is shown in the middle figure of Figure 6.10, and the corresponding obstacles in original images are shown in its surrounding. This obstacle map is obtained with $m = 10$ and range limit 3,000 meters. It can be seen that some of the stationary obstacles can be effectively mapped after a full travelling loop of the USV. In Figure 6.10, the bottom-left image shows a stationary buoy with measured distance around 100m, and its corresponding points in the obstacle map have a very small variance, due to the fact that the shorter the distance, the stronger the detected features, and the more accurate the feature matching. The images in middle-right, top-right, and top-middle show stationary tanker and boats with distance around 500m to the USV, and these obstacles demonstrate larger but acceptable variances in the obstacle map. A large container-ship is respectively

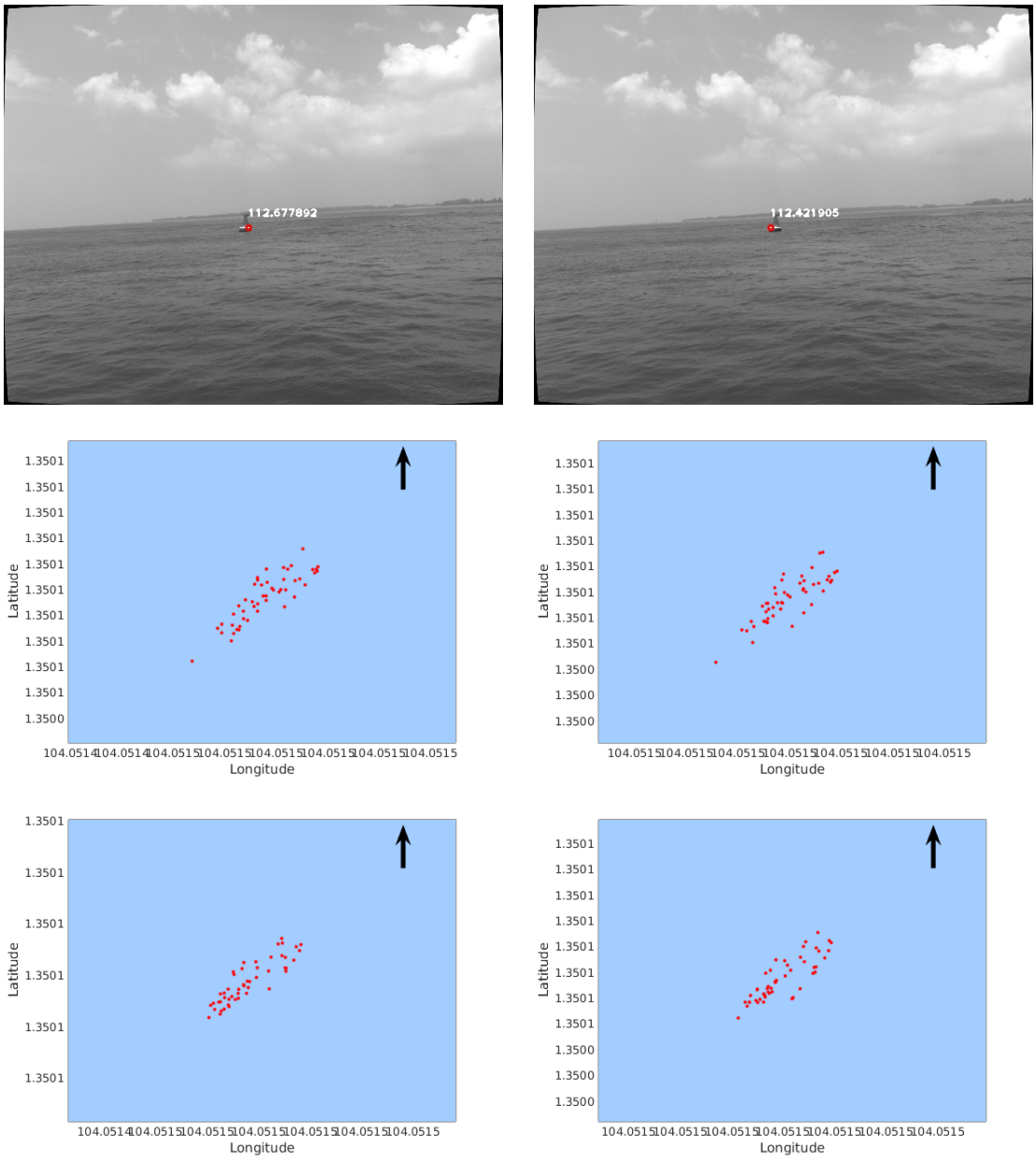


Figure 6.6: Distribution of mapped feature points (red) in **Seq-1**. The left column shows the case of Feature Point #1; the right column, the case of Feature Point #2. The top row shows the features on images with their distances displayed in meters. The middle row shows the reconstructed points using 1 pair of frames. The bottom row shows the reconstructed points using 5 pair of frames.

shown in the bottom-middle and top-left images with very large distance ($>$ 1,000m) to the USV, and they are mapped with very big variances, because the detected

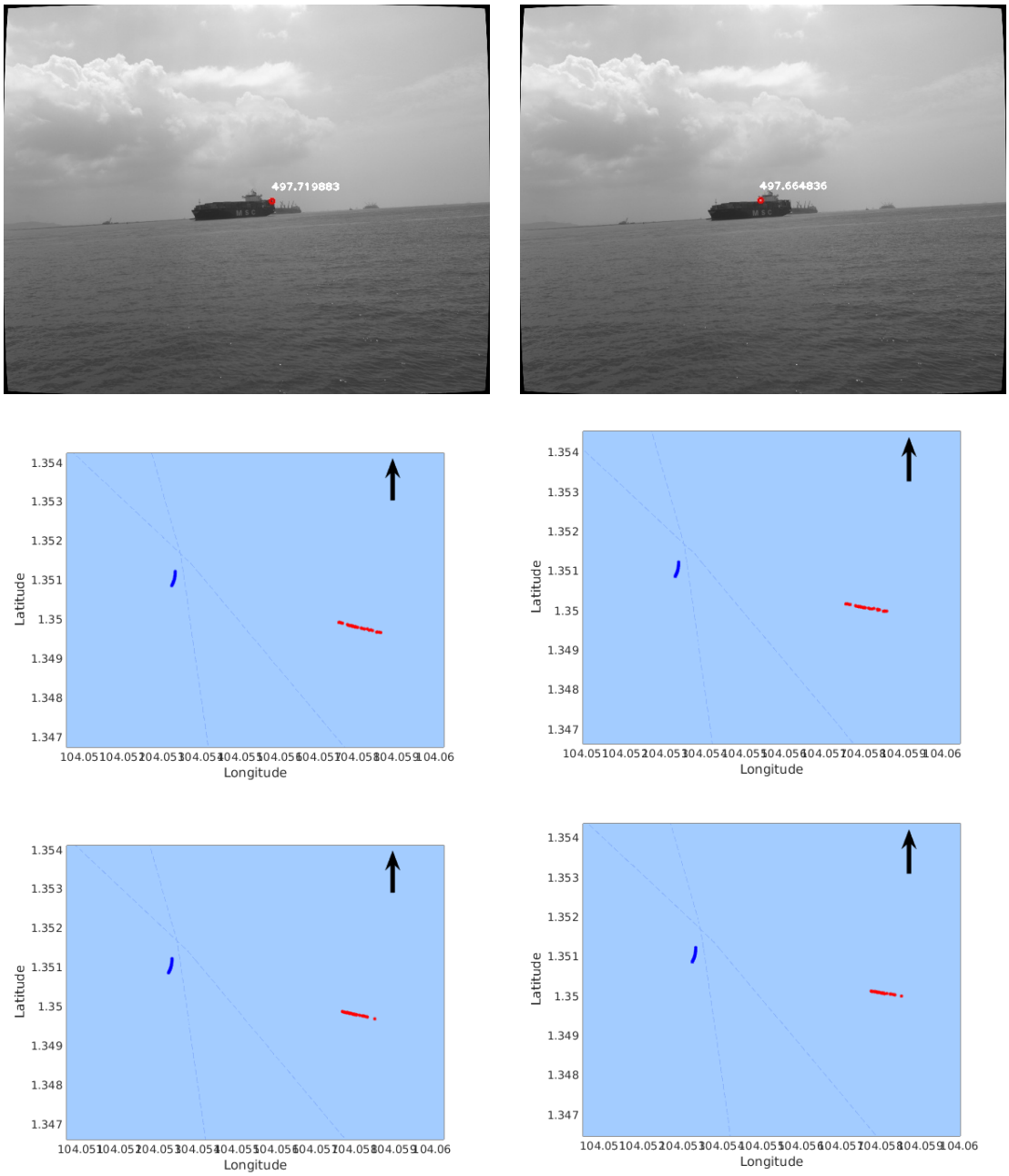
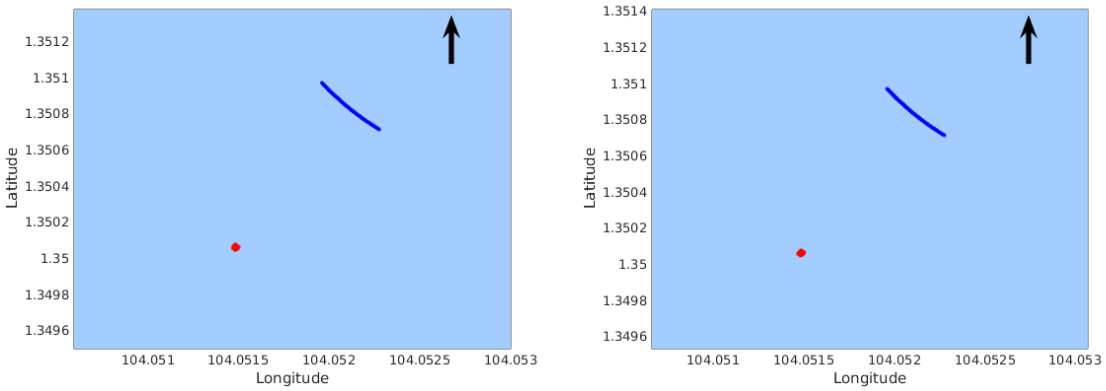
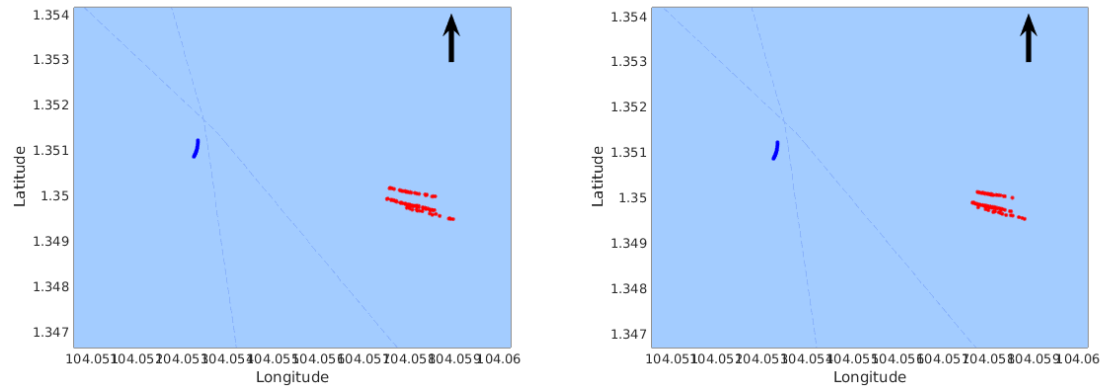


Figure 6.7: Distribution of mapped feature points (red) in **Seq-2**. The left column shows the case of Feature Point #1; the right column, the case of Feature Point #2. The top row shows the features on images with their distances displayed in meters. The middle row shows the reconstructed points using 1 pair of frames. The bottom row shows the reconstructed points using 10 pair of frames.



(a) Triangulation using 1 frame pair. (b) Triangulation using 10 frame pairs.

Figure 6.8: Obstacle mapping result of **Seq-1**. The red points are the reconstructed feature points from obstacles; The blue curve presents the trajectory of our USV.



(a) Triangulation using 1 frame pair. (b) Triangulation using 10 frame pairs.

Figure 6.9: Obstacle mapping result of **Seq-2**. The red points are the reconstructed feature points from obstacles; The blue curve presents the trajectory of our USV.

features from the distant obstacles are very weak, and hard to be matched correctly, thus resulted in unreliable distance estimation. The similar case also appears in the middle-left image, which shows the scenario of seashore. A moving obstacle case is illustrated in the bottom-right image, which contains a fast moving boat with computed distance around 500m, however, by visually comparing with the stationary tanker (estimated distance around 500m) on the left of this image, one

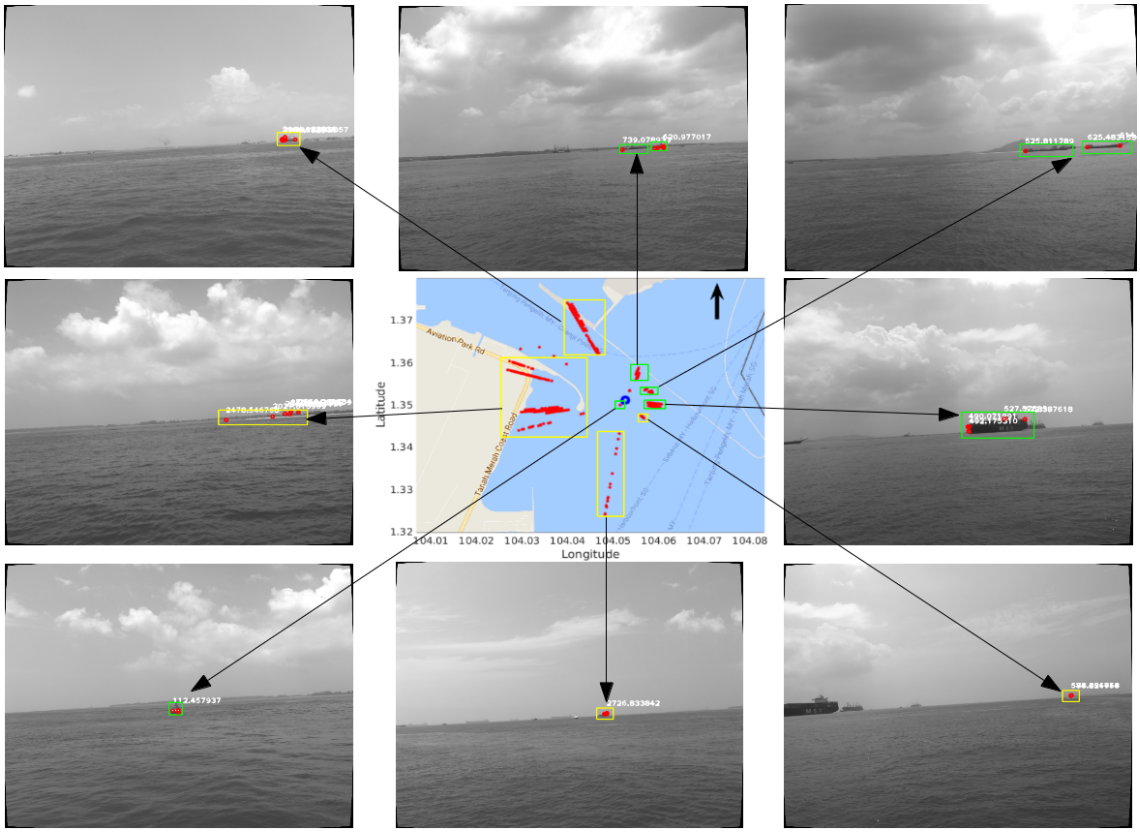


Figure 6.10: Resulted obstacle map (middle) after a full moving loop of the USV (**Seq-3**). The corresponding obstacles in original images are shown surroundingly with an arrow linking each of them to the map. In the obstacle map, the blue circle represents the trajectory of the USV, and the red points represent the mapped feature points from obstacles. The green rectangles are manually drawn to illustrate the stationary obstacles, while the yellow rectangles are manually drawn to show the distant obstalces with large mapping variances and the moving obstacles.

can observe that the moving boat is much farther away from the USV than the tanker, so the mapped points from this moving boat are not reliable.

Therefore, it can be concluded from Figure 6.10 that the proposed obstacle mapping system for USV is very sensitive to the feature matching. Nearby stationary obstacles can be robustly mapped with small variances, because strong features can be detected and then matched with high accuracy, while distant stationary obstacles might be mapped with large variances due to the weak features and less reliable matching. However, no matter strong or weak features, moving obstacles would

be mapped wrongly. Another important reason caused the large mapping variance of distant obstacles lies in the inherent nature of stereo vision. In Equation (6.2), there exists: $\Delta Z \propto Z^2$. Thus, larger depth yields larger reconstruction error, which results the larger variance in obstacle mapping. The reliable mapping range of the proposed system would be from 100m to 1,000m.

6.8 Concluding Remarks

An new approach for monocular vision based obstacle mapping using motion stereo has been presented. The translation of the camera is computed from recorded GPS data, and the rotation of the camera is obtained from a compass (yaw) and the detected horizon line (roll and pitch). Knowing the geometry of two images at different locations, a pair of matched feature points in the two images can be triangulated using basic linear algebra. Moreover, an approach of using multiple frame pairs to enhance the stability of reconstruction is proposed. Finally, the triangulated points are projected to a Google Map.

The proposed approach gets rid of the bulky structure and difficult calibration in binocular stereo. On the contrary, it is very convenient to setup and just needs a simple calibration for the intrinsic parameters of the camera. Its ranging ability is approximately from 100m to 1,000m. However, it can only reconstruct points from static scenes, and the USV should be in the state of travelling. Furthermore, measurement noise come from GPS, compass, horizon detection, and especially feature matching might influence the accuracy of this approach.

The proposed approach relies on the horizon line to be detected. A future work would be detecting the water line that segments the sea surface from the seashore, so that the proposed approach could work when the camera is close and facing the seashore. Another future work would be integrating with the image-based methods for obstacle detection, so that the points in the resulted obstacle map could be clustered easily. In addition, more quantitative evaluation for the proposed method will be conducted in the future by comparing with the recorded Radar data (ground truth).

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this study of vision based obstacle detection and mapping for USV, three algorithms for improving the performance of obstacle detection and one algorithm exploring the monocular vision based obstacle mapping have been presented. The proposed approaches have 4 key contributions.

Firstly, we proposed a new sparseness measure, global sparsity potential (GSP), for the appearance of an image patch. Using GSP, we proposed a new image-based approach for obstacle detection in maritime scenes. In this approach, image patches with a relatively small GSP value are considered the main cluster (i.e., sea surface), while their outliers, which have a relatively large GSP value and a relatively large Mahalanobis distance with respect to the mean feature of the sea surface), are considered the obstacles. It outperforms one conventional method using feature space reclustering and one state-of-the-art method for saliency detection in our experiments. Although this proposed image-based method is convenient, economic, and fast-processing, it suffers from false positives come from white wake, waves, sun reflection, etc.. Moreover, the distance of the detected obstacles can not be estimated.

Secondly, a novel framework for real-time binocular stereo based long range obstacle detection and tracking in maritime environment was presented. In that framework, we proposed to detect obstacles in an image-pyramid manner to speed

the processing up, and we also proposed a new solution for multiple-obstacle tracking with scale adapting and occlusion handling. Experimental results demonstrated its high accuracy and robustness. This proposed approach greatly reduces the false positives caused by noise (white wake, waves, sun reflection, etc.) in obstacle detection compared to that of the image-based methods. Nevertheless, the large baseline makes the stereo rig really bulky and hard to calibrate. The processing speed is influenced by the number of obstacles in tracking. That means, the more obstacles in tracking, the lower processing speed. In that case, the real-time performance cannot be achieved, due to the increased computing burden. Moreover, we consider the obstacle is occluded or reappears only based on the depth information, which may be insufficient and weaken the reliability of the occlusion handling. In the future work, the appearance information can be added to enhance the robustness of the occlusion handling.

Thirdly, to further enhance the visual obstacle detection in open sea, a fusion method that leverage both the advantages of 2D based detection and 3D based detection was proposed. With 2D clue, we can detect all salient candidate obstacles in images no matter how far they are, however, false positives might be included in the results. With 3D clue, we can detect obstacles in short range by clustering points protrude from the sea surface, though this results in less noise, obstacles far away are hard to be detected, because their depth may not be valid. Since 2D and 3D can compensate the drawbacks of each other, we proposed to combine the 2D and 3D clues in a weighting model, which gives more weights to the 2D detecting results when obstacles are far away, while gives more weights to the 3D detecting results when obstacles are nearby. In addition, an improved horizon detection algorithm by fusing 2D and 3D clues was also proposed in this approach. The proposed method was demonstrated by experimental results with superior performance for obstacle detection than the methods based on only 2D or 3D information. Besides the same cons shared with the previously proposed binocular stereo system, this approach gets a lower processing speed due to the large amount of data to be processed.

Lastly, a new motion stereo based approach for obstacle mapping in open sea environment was presented. To the best of our knowledge, we are the first to address

the problem of monocular vision based obstacle mapping for USV. In this approach, the bulky structure and difficult calibration in binocular stereo are avoided, instead, the translation of the camera is computed from recorded GPS data, and the rotation of the camera is obtained from a compass (yaw) and the detected horizon line (roll and pitch). Knowing the geometry of two images at different locations, a pair of matched feature points in the two images can be triangulated using basic linear algebra. In addition, we exploited multiple pairs of frames to triangulate more accurate and robust 3D information of feature points. Obstacles mapped on Google Map in experiments demonstrated the efficiency of this proposed approach. The ranging ability of the proposed approach is greatly enlarged compared to the binocular stereo, however, it can only reconstruct the points from static scenes, and the USV should be in the state of travelling to provide large enough baseline for frame pairs. Furthermore, measurement noise come from GPS, compass, horizon detection, and especially feature matching might influence the accuracy of this approach.

7.2 Future Work

The proposed framework opens an exciting area of research on obstacle detection and mapping in maritime environment. Some suggested future work in this field include:

1. Exploring machine learning methods

   The detection methods proposed in this study output the locations for all the obstacles, but can not differentiate and recognize what a specific obstacle is. To make our system be more intelligent, machine learning methods, such as deep R-CNN [97], random forest [98], etc., have to be exploited to improve the detection performance and let the USV know what kind of object each detected obstacle is and then make the corresponding response.

   Another advantage of using machine learning resides in solving the occlusion problem during object tracking. When an occluded object reappears in the image, we can recover its ID before being occluded by checking its appearance with the trained model using machine learning.

2. Addressing complex environments

   One limitation of the proposed methods for obstacle detection is that the horizon line or sea surface plane needs to be obtained as a preliminary requirement. Therefore, the methods in this study may not work well when the image plane is close to the seashore, which produces much noise for sea surface plane estimation. To solve this problem, a combination of 3D point cloud processing and machine learning can be studied. That means we can detect the scene of seashore or the sea surface in 2D image using machine learning, thereby, the 3D points from the sea surface can be selected from the point cloud to fit the plane.

   Furthermore, a robust system should also work under other complex environments, such as varying visibility and weather conditions, including rain, fog, strong wind, high waves, etc.. Each of these scenarios will be investigated in the future.

3. Further optimizing the monocular vision based obstacle mapping

   In the part of obstacle mapping, the roll and pitch angles of the camera are estimated from the detected horizon line. However, when the camera is facing the seashore, in which scenario the horizon line can not be detected using image processing methods, the proposed system would fail to work. One promising solution for this issue is to apply image segmentation or machine learning methods to extract the coastal line that separates the sea surface from the seashore. Then, recover the roll and pitch angles with respect to this coastal line.

   Another future work would be integrating with the image-based methods for obstacle detection, so that the points in the resulted obstacle map could be clustered easily.

   Moreover, in triangulating the feature points of the proposed motion stereo system, multiple pairs of frames are leveraged with equal weight. To further optimize the process of triangulation, each pair of frames should take different weights according to the measurement variances in GPS, IMU, horizon

detection, and feature matching. In this way, more accurate and robust triangulation results could be expected to obtain.

4. Fusing motion stereo with binocular stereo

   Motion stereo is able to reconstruct points with large distances, however, it can only reconstruct the static scenes, so our proposed mapping approach is limited to stationary obstacles. It can be fused with binocular stereo, which is suitable to reconstruct both the dynamic and the static scenes in the near-field environment, to achieve an improved obstacle map. That means building an obstacle map which contains points with short distances from binocular stereo and points with larger distances from motion stereo.

   Another advantage of combing motion stereo with binocular stereo is that the roll and pitch angles of the camera can be recovered by the fitted 3D sea surface plane from binocular stereo in case that the horizon line can not be detected in a single 2D image.

Bibliography

[1] F. Liu and D. Zhang. 3D fingerprint reconstruction system using feature correspondences and prior estimated finger model. *Pattern Recognition*, 47(1):178–193, 2014.

[2] P. Voles, A. A. W. Smith, and M. K. Teal. Nautical scene segmentation using variable size image windows and feature space reclustering. *Proceedings of European Conference on Computer Vision (ECCV)*, pages 324–335, 2000.

[3] S. Frintrop, T. Werner, and G. M. Garcia. Traditional saliency reloaded: a good old model in new shape. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 82–90, 2015.

[4] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M. Yang. Fast visual tracking via dense spatio-temporal context learning. *Proceedings of European Conference on Computer Vision (ECCV)*, pages 127–141, 2014.

[5] C. Almeida, T. Franco, H. Ferreira, A. Martins, R. Santos, J. Almeida, J. Carvalho, and E. Silva. Radar based collision detection developments on USV ROAZ II. *OCEANS*, pages 1–6, 2009.

[6] C. Onunka and G. Bright. Autonomous marine craft navigation: On the study of radar obstacle detection. *Proceedings of the IEEE International Conference on Control Automation Robotics & Vision*, pages 567–572, 2010.

[7] M. Schuster, M. Blaich, and J. Reuter. Collision avoidance for vessels using a low-cost radar sensor. *IFAC Proceedings Volumes*, 47(3):9673–9678, 2014.

[8] FURUNO. FURUNO Marine Radar. *Furuno Electric Co. Ltd*, http://www.furuno.com/en/products/radar.

[9] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabaly, and C. Quek. Video processing from electro-optical sensors for object detection and tracking in a maritime environment: a survey. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–24, 2017.

[10] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[11] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:679–698, 1986.

[12] S. Suzuki. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.

[13] S. Fefilatyev, D. Goldgof, M. Shreve, and C. Lembke. Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system. *Ocean Engineering*, 54:1–12, 2012.

[14] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[15] S. D. Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing*, 33(1):116–125, 1986.

[16] D. Hermann, R. Galeazzi, J. C. Andersen, and M. Blanke. Smart sensor based obstacle detection for high-speed unmanned surface vehicle. *IFAC-PapersOnLine*, 48(16):190–197, 2015.

[17] S. Fefilatyev and D. Golgof. Detection and tracking of marine vehicles in video. *Proceedings of International Conference on Pattern Recognition*, pages 1–4, 2008.

[18] J. Woo, J. Lee, and N. Kim. Obstacle avoidance and target search of an autonomous surface vehicle for 2016 Maritime RobotX challenge. *Underwater Technology*, pages 1–5, 2017.

[19] T. H. Tran and T. L. Le. Vision based boat detection for maritime surveillance. *Proceedings of the IEEE International Conference on Electronics, Information, and Communications*, pages 1–4, 2016.

[20] P. KaewTraKulPong and R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. *Video-based surveillance systems*, 1:135–144, 2001.

[21] Z. Zivkovic. Improved adaptive Gaussian mixture model for background subtraction. *Proceedings of IEEE Conference on Pattern Recognition (ICPR)*, pages 28–31, 2004.

[22] O. Barnich and M. V. Droogenbroeck. ViBe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image Processing*, 20(6):1709–1724, 2011.

[23] L. Li, W. Huang, I. Y. Gu, and Q. Tian. Foreground object detection from videos containing complex background. *Proceedings of ACM International Conference on Multimedia*, pages 2–10, 2003.

[24] P. F. Felzenszwalb and D. P. Huttenlocher. Image segmentation using local variation. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 98–104, 1998.

[25] D. Socek, D. Culibrk, O. Marques, H. Kalva, and B. Furht. A hybrid color-based foreground object detection method for automated marine surveillance. *Lecture Notes in Computer Science*, 3708:340, 2005.

[26] D. Frost and J. R. Tapamo. Detection and tracking of moving objects in a maritime environment using level set with shape priors. *EURASIP Journal on Image and Video Processing*, 42(1):1–16, 2013.

[27] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, pages 1151–1163, 2002.

[28] A. Tsai, A. Yezzi, W. Wells, C. Tempany, D. Tucker, A. Fan, W. E. Grimson, and A. Willsky. A shape-based approach to the segmentation of medical imagery using level sets. *IEEE Transactions on Medical Imaging*, 22(2):137–154, 2003.

[29] A. Borji, M. Cheng, H. Jiang, and J. Li. Salient object detection: A survey. *arXiv preprint arXiv:1411.5878*, 2014.

[30] T. Cane and J. Ferryman. Saliency-based detection for maritime object tracking. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 18–25, 2016.

[31] J. Zhang and S. Sclaroff. Exploiting surroundedness for saliency detection: A boolean map approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(5):889–902, 2016.

[32] A. Sobral, T. Bouwmans, and E. ZahZah. Double-constrained RPCA based on saliency maps for foreground detection in automated maritime surveillance. *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 1–6, 2015.

[33] E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):117–142, 2011.

[34] C. Li, Z. Cao, Y. Xiao, and Z. Fang. Fast object detection from unmanned surface vehicles via objectness and saliency. *Proceedings of the IEEE Chinese Automation Congress*, pages 500–505, 2015.

[35] C. L. Zitnick and P. Dollar. Edge boxes: Locating object proposals from edges. *Proceedings of European Conference on Computer Vision (ECCV)*, pages 391–405, 2014.

[36] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. *Proceedings*

of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2083–2090, 2013.

[37] R. Achanta and S. Susstrunk. Saliency detection using maximum symmetric surround. *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 2653–2656, 2010.

[38] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[39] H. Wang, Z. Wei, S. Wang, C. Ow, K. Ho, and B. Feng. A vision-based obstacle detection system for unmanned surface vehicle. *Proceedings of IEEE Conference on Robotics, Automation and Mechatronics*, pages 364–369, 2011.

[40] J. Y. Bouguet. Pyramidal implementation of the Lucas Kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*, 2000.

[41] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 246–252, 1999.

[42] D. Bloisi and L. Iocchi. ARGOS–A video surveillance system for boat traffic monitoring in Venice. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(7):1477–1502, 2009.

[43] Y. Wang, R. Tan, G. Xing, J. Wang, X. Tan, X. Liu, and X. Chang. Aquatic debris monitoring using smartphone-based robotic sensors. *Proceedings of International Symposium on Information Processing in Sensor Networks*, pages 13–24, 2014.

[44] M. Kristan, V. Sulic Kent, S. Kovacic, and J. Pers. Fast image-based obstacle detection from unmanned surface vehicles. *IEEE Transactions on Cybernetics*, 43(3):641–654, 2015.

[45] S. Z. Li. Markov random field modeling in image analysis. *Springer Science & Business Media*, 2009.

[46] A. Diplaros, N. Vlassis, and T. Gevers. A spatially constrained generative model and an EM algorithm for image segmentation. *IEEE Transactions on Neural Networks*, 18(3):798–808, 2007.

[47] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[48] J. A. Hartigan and M. A. Wong. Algorithm AS 136: A K-Means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[49] O. Gal. Automatic obstacle detection for USV's navigation using vision sensors. *Robotic sailing*, pages 127–140, 2011.

[50] R. M. Haralick, K. Shanmugam, and I. H. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621, 1973.

[51] X. Mou and H. Wang. Image-based maritime obstacle detection using global sparsity potentials. *Journal of Information and Communication Convergence Engineering*, 14(2):129–135, 2016.

[52] D. Bloisi, L. Iocchi, M. Fiorini, and G. Graziano. Camera based target recognition for maritime awareness. *Proceedings of International Conference on Information Fusion*, pages 1982–1987, 2012.

[53] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[54] T. Sadhu, A. B. Albu, M. Hoeberechts, E. Wisernig, and B. Wyvill. Obstacle detection for image-guided surface water navigation. *Proceedings of the IEEE International Conference on Computer and Robot Vision*, pages 45–52, 2016.

[55] C. M. Bishop. Pattern recognition and machine learning. *Springer*, 2006.

[56] F. Bousetouane and B. Morris. Fast CNN surveillance pipeline for fine-grained vessel classification and detection in maritime scenarios. *Proceedings of the*

IEEE International Conference on Advanced Video and Signal Based Surveillance, pages 242–248, 2016.

[57] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[58] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005.

[59] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[60] R. Girshick. Fast R-CNN. *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[61] Y. Tang. Deep learning using linear support vector machines. *arXiv preprint arXiv:1306.0239*, 2013.

[62] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014.

[63] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[64] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg. SSD: Single shot MultiBox detector. *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 21–37, 2016.

[65] J. Larson, M. Bruch, and J. Ebken. Autonomous navigation and obstacle avoidance for unmanned surface vehicles. *In Defense and security symposium*, pages 623007–623007, 2006.

[66] T. Huntsberger, H. Aghazarian, A. Howard, and D. Trotz. Stereo vision-based navigation for autonomous surface vessels. *Journal of Field Robotics*, 28(1):3–18, 2011.

[67] M. Harville. Stereo person tracking with adaptive plan-view templates of height and occupancy statistics. *Image and Vision Computing*, 22(2):127–142, 2004.

[68] R. Muñoz-Salinas. A bayesian plan-view map based approach for multiple-person detection and tracking. *Pattern Recognition*, 41(12):3665–3676, 2008.

[69] H. Wang, Z. Wei, C. S. Ow, K. T. Ho, B. Feng, and J. Huang. Improvement in real-time obstacle detection system for USV. *Proceedings of the IEEE International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 1317–1322, 2012.

[70] H. Wang and Z. Wei. Stereovision based obstacle detection system for unmanned surface vehicle. *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pages 917–921, 2013.

[71] H. Wang, X. Mou, W. Mou, S. Yuan, S. Ulun, S. Yang, and B. S. Shin. Vision based long range object detection and tracking for unmanned surface vehicle. *Proceedings of the IEEE International Conference on Robotics, Automation and Mechatronics (RAM)*, pages 101–105, 2015.

[72] H. Wang, W. Mou, X. Mou, S. Yuan, S. Ulun, S. Yang, and B. S. Shin. An automatic self-calibration approach for wide baseline stereo cameras using sea surface images. *Unmanned Systems*, 3(4):277–290, 2015.

[73] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

[74] R. Gladstone, Y. Moshe, A. Barel, and E. Shenhav. Distance estimation for marine vehicles using a monocular video camera. *Proceedings of European Signal Processing Conference (EUSIPCO)*, pages 2405–2409, 2016.

[75] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[76] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. ORB-SLAM: A versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[77] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 2564–2571, 2011.

[78] J. Engel, T. Schops, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. *Proceedings of European Conference on Computer Vision (ECCV)*, pages 834–849, 2014.

[79] Y. Yan, B. S. Shin, X. Mou, W. Mou, and H. Wang. Efficient horizon detection on complex sea for sea surveillance. *International Journal of Electrical, Electronics and Data Communication*, 3(12):49–52, 2015.

[80] N. Razavi, N. S. Alvar, J. Gall, and L. Van Gool. Sparsity potentials for detecting objects with the Hough transform. *Proceedings of BMVC*, pages 1–10, 2012.

[81] T. Deselaers and V. Ferrari. Global and efficient self-similarity for object classification and detection. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1633–1640, 2010.

[82] B. S. Shin, J. Tao, and R. Klette. A superparticle filter for lane detection. *Pattern Recognition*, 48(11):3333–3345, 2015.

[83] K. Konolige. Small vision systems: Hardware and implementation. *Robotics Research*, pages 203–212, 1998.

[84] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[85] J. Tao, M. Enzweiler, U. Franke, D. Pfeiffer, and R. Klette. What is in front? multiple-object detection and tracking with dynamic occlusion handling. *Computer Analysis of Images and Patterns*, pages 14–26, 2015.

[86] E. Rosten and T. Drummond. Machine learning for high speed corner detection. *Proceedings of European Conference on Computer Vision (ECCV)*, pages 430–443, 2006.

[87] K. Schauwecker, R. Klette, and A. Zell. A new feature detector and stereo matching method for accurate high-performance sparse stereo matching. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5171–5176, 2012.

[88] J. Xu, K. Kim, Z. Zhang, H. Chen, and Y. Owechko. 2D/3D sensor exploitation and fusion for enhanced object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 764–770, 2014.

[89] K. Kanatani. Geometric computation for machine vision. *Oxford University Press*, 1993.

[90] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1981.

[91] R. Hartley and A. Zisserman. Multiple view geometry in computer vision. *Cambridge University Press*, 2003.

[92] D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004.

[93] P. Dollar and C. L. Zitnick. Fast edge detection using structured forests. *IEEE Transactions on PAMI*, 37(8):1558–1570, 2015.

[94] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):105–119, 2010.

[95] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary robust independent elementary features. *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 778–792, 2010.

[96] C. Harris and M. Stephens. A combined corner and edge detector. *Alvey Vision Conference*, 15(50):10–5244, 1988.

[97] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[98] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

Appendix A

Author's Publications

Journal Papers

[1] Xiaozheng Mou and Han Wang, "Wide-baseline stereo based obstacle mapping for unmanned surface vehicles," *Sensors*, 2018. (accepted)

[2] Bok-Suk Shin, Xiaozheng Mou, Wei Mou and Han Wang, "Vision-based navigation of an unmanned surface vehicle with object detection and tracking abilities," *Machine Vision and Applications*, 29(1), pp.95-112, 2018.

[3] Xiaozheng Mou and Han Wang, "Image-based maritime obstacle detection using global sparsity potentials," *Journal of Information and Communication Convergence Engineering*, 14(2), pp.129-135, 2016.

[4] Han Wang, Wei Mou, Xiaozheng Mou, Shenghai Yuan, Soner Ulun, Shuai Yang and Bok-Suk Shin, "An automatic self-calibration approach for wide baseline stereo cameras using sea surface images," *Unmanned Systems*, 3(4), pp.277-290, 2015.

[5] Yan Yan, Bok-Suk Shin, Xiaozheng Mou, Wei Mou and Han Wang, "Efficient horizon detection on complex sea for sea surveillance," *International Journal of Electrical, Electronics and Data Communication*, 3(12), pp.49-52, 2015.

Conference Papers

[1] Xiaozheng Mou, Bok-Suk Shin and Han Wang, "Hierarchical RANSAC for accurate horizon detection," *in Proceedings of 24th Mediterranean Conference on*

Control and Automation (MED'16), pp.1158-1163, 2016.

[2] Xiaozheng Mou, Han Wang and Kart-Leong Lim, "Scale-adaptive multiple-obstacle tracking with occlusion handling in maritime scenes," *in Proceedings of IEEE International Conference on Control and Automation (ICCA)*, pp.588-592, 2016.

[3] Bok-Suk Shin, Xiaozheng Mou and Han Wang, "Generic horizon feature detection for unmanned surface vehicle," *in Proceedings of IEEE International Conference on Future Information & Communication Engineering (ICFICE)*, pp.315-316, 2016.

[4] Kart-Leong Lim, Han Wang and Xiaozheng Mou, "Learning Gaussian mixture model with a maximization-maximization algorithm for image classification," *in Proceedings of IEEE International Conference on Control and Automation (ICCA)*, pp.887-891, 2016.

[5] Xiaozheng Mou and Han Wang, "Global sparsity potentials for obstacle detection from unmanned surface vehicles," *in Proceedings of IEEE International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pp.1-6, 2015.

[6] Han Wang, Xiaozheng Mou, Wei Mou, Shenghai Yuan, Soner Ulun, Shuai Yang and Bok-Suk Shin, "Vision based long range object detection and tracking for unmanned surface vehicle," *in Proceedings of IEEE International Conference on CIS-RAM)*, pp.101-105, 2015.

[7] Han Wang, Wei Mou, Xiaozheng Mou, Shenghai Yuan, Soner Ulun, Shuai Yang and Bok-Suk Shin, "An automatic self-calibration approach for wide baseline stereo cameras using sea surface images," *in Proceedings of IEEE International Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pp.1-6, 2015.

[8] Yan Yan, Bok-Suk Shin, Xiaozheng Mou, Wei Mou and Han Wang, "Efficient horizon detection on complex sea for sea surveillance," *in Proceedings of International Conference on Science, Technology and Management (ICSTM)*, pp.14-17, 2015.

[9] Xiaozheng Mou and Han Wang, "An improved approach for depth data based face pose estimation using particle swarm optimization," *in Proceedings of International Conference on Computer Vision Theory and Applications (VISAPP)*, pp.534-

541, 2014.

[10] Xiaozheng Mou and Han Wang, "A fast and robust head pose estimation system based on depth data," *in Proceedings of IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp.470-475, 2012.