Wide-baseline obstacle mapping using monocular camera for unmanned surface vehicle

Chen, Jiaying; Wang, Han

2019

Chen, J. & Wang, H. (2019). Wide-baseline obstacle mapping using monocular camera for unmanned surface vehicle. 2019 IEEE 15th International Conference on Control and Automation (ICCA), 512-517. https://dx.doi.org/10.1109/ICCA.2019.8899983

https://hdl.handle.net/10356/148385

https://doi.org/10.1109/ICCA.2019.8899983

Wide-Basline Obstacle Mapping Using Monocular Camera for Unmanned Surface Vehicle

Chen Jiaying; Wang Han

2019

Chen Jiaying & Wang Han (2019). Wide-Basline Obstacle Mapping Using Monocular Camera for Unmanned Surface Vehicle. 2019 IEEE 15th International Conference on Control and Automation (ICCA), 512-517.

Wide-Baseline Obstacle Mapping Using Monocular Camera for Unmanned Surface Vehicle

Jiaying Chen[1] and Han Wang[1]

***Abstract*— In this paper, we applied a static obstacle mapping system for USV. This system consists of a monocular camera, GPS and compass. Obstacle map can be built while the USV is traveling. To increase the accuracy of the system, we apply feature matching after image transformation and introduce a threshold to filter motion parallax computation results. Moreover, we collect radar data as ground truth to evaluate the system performance and accuracy. According to the results of our experiments, this system can map the static obstacles with high accuracy and efficiency.**

I. INTRODUCTION

Unmanned surface vehicles (USVs) refers to any vehicle that operates on the sea without a crew. USVs can be applied widely both in the military and civilian service; for example, USVs can be used for sea surface surveillance, detect irregular ship movement and smuggle boat.

In this paper, we apply an obstacle mapping system [1] for USV by only using a monocular camera, GPS and compass. Compared to the binocular stereo vision system [2], using one camera greatly reduce the costs and eliminate the problem for calibration. To obtain obstacles map, visual odometry (VO) of USV should be known. Conventional method such as 8-point algorithm [3] , the 7-point algorithm [4], and PnP algorithm [5] can be applied to compute camera translation and rotation. However, these conventional methods are not suitable for the sea environment, because there are very few features that can be detected. For the sea images, the major part of the images is uniform sea and sky. Therefore, we use another method to compute VO. The camera translation can be obtained from GPS data. The angles of camera rotation can be obtained from sea horizon line detection (roll and pitch) and compass or IMU (yaw). After VO is obtained, pith and roll angles are used to do image transformation. Then, the 3-dimensional sea environment can be treated as 2D space. ORB method is applied to detect matched features for a pair of images after transformation. We calculate the obstacle depth using motion parallax method with matched features. We introduce a threshold in the motion parallax computation process and increase the accuracy. Finally, these feature points are reconstructed to the geodetic coordinates (latitude and longitude) and plot on Google Map. To evaluate the performance and accuracy of this system, we also collect the radar data as ground truth.

The paper is structured as follow: Sect.2 introduces camera rotation and image transformation. Sect.3 introduces motion parallax method. Sect.4 briefly outlines feature matching. Sect.5 analyzes experiments data and discuss. Sect.concludes.

*Singapore Technologies Electronic Ltd.

[1]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 chenjy, HW@ntu.edu.sg

II. CAMERA ROTATION AND IMAGE TRANSFORMATION

A. Camera Rotation

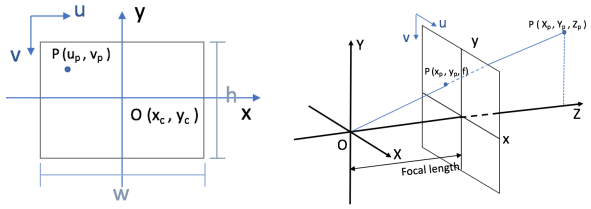


Fig. 1: Point P in different coordinate system

In this section, we need to do coordinates conversion and rotation. In order to have a better understanding of the following sections. Firstly, we briefly state the conversion of four coordinates that are world coordinate, camera coordinate, image coordinate, the pixel coordinate. As shown in Fig.1, suppose there is a point $P(X_p, Y_p, Z_p)$ in world coordinate is project to an image on point $p(x_p, y_p, f)$. The conversion between world coordinate and camera coordinate is:

$$\begin{pmatrix} x_p \\ y_p \\ f \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} X_p \\ Y_p \\ Z_p \end{pmatrix} \tag{1}$$

where f is the focal length of the camera.

Suppose, the point in the pixel coordinate is $p(u_p, v_p)$. The conversion between pixel coordinate and image coordinate is shown as:

$$x_p = u_p - x_c, \quad y_p = -(v_p - y_c) \tag{2}$$

where x_c and y_c represent center of image in pixel coordinate. If the size of image is $w \times h$, then $x_c = \frac{w}{2}$, and $y_c = \frac{h}{2}$.

Moreover, we define the rotation angles. Roll angle denoted by α represents a rotation around the z-axis, Yaw angle denoted by β represents a rotation around the y-axis, Pitch angle indicated by γ represents a rotation around the x-axis. Now, lets obtain the rotation angles in our experiments.

Although camera motion can be obtained directly from the IMU and GPS sensor, IMU used in the experiments is very cheap which means that the accuracy of the IMU is not high. Gyroscope and accelerometer measures the roll and pitch angles, and the magnetometer sensor in the IMU measures the yaw angle that is reliable. We only used

IMU as a compass and the yaw angle obtained from IMU. Furthermore, we obtained roll and pitch angles from sea horizon line or vanishing line. The methods applied to detect horizon line are structured edge detection [6] used for edge map computation as well as RANSAC method [7]used for fitting a straight line from edge map. As shown in Fig.2, we suppose the monocular camera is aligned with the world coordinate system; hence the sea horizon line should be at the middle row of the image. If there is no rotation, the normal vector $\boldsymbol{n}$of the sea surface plane in camera coordinate system should be as same as the normal vector in the world coordinate system which is along Y-axis. When there exists rotation around Z-axis and X-axis, the sea surface plane will tilt, and the vanishing line in the image will change. Denoted two points $p_1(x_1, y_1)$and $p_2(x_2, y_2)$are on the vanishing line. Vector $\boldsymbol{Op}_1$ and $\boldsymbol{Op}_2$ are $(x_1, y_1, f)^T$ and $(x_2, y_2, f)^T$ respectively. The normal vector of the sea surface plane is : $\boldsymbol{n} = \boldsymbol{Op}_1 \times \boldsymbol{Op}_2 = (n_1, n_2, n_3)^T$. The roll angle α can be computed as:

$$\alpha = \arctan\left(\frac{n_1}{n_2}\right) \tag{3}$$

The pitch angle γ can be computed as:

$$\gamma = \arctan\left(\frac{n_3}{\sqrt{n_1^2 + n_2^2}}\right) \tag{4}$$

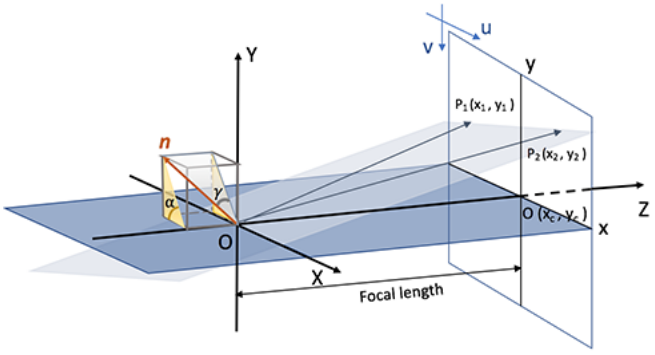


Fig. 2: Normal vector of sea horizon line

Compared with roll and pitch angles obtained by the IMU sensor, Fig.3 illustrates both rotation angles derived from the IMU sensor and horizon line detection. There exist significant differences between the IMU sensor and horizon line measured rotation angles.

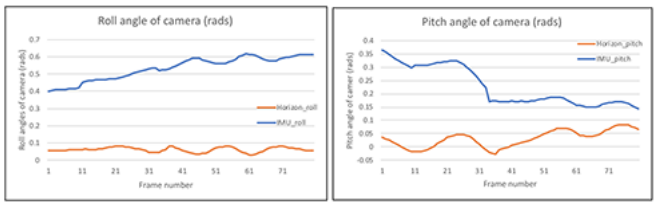


Fig. 3: Roll and pitch angles obtained from vanishing line and IMU

B. Image Transformation

To do the image transformation based on pitch and roll angles, we need to find the general perspective transform F of a quadrilateral.

$$M = F(w, h, \alpha, \gamma, f_v) \tag{5}$$

which produce a 4×4 warp matrix M by which to transform the vertices of a $w \times h$ image, given the vertical field of view f_v, and camera rotation roll and pitch angles α, γ. From Fig.4 , when the roll angle $\alpha = \arctan(\frac{w}{h})$and the pitch angle $\gamma = \frac{f_v}{2}$, the extreme case occurs, that is, which results in the biggest image. We list each corner position in image coordinate is listed in Table I.

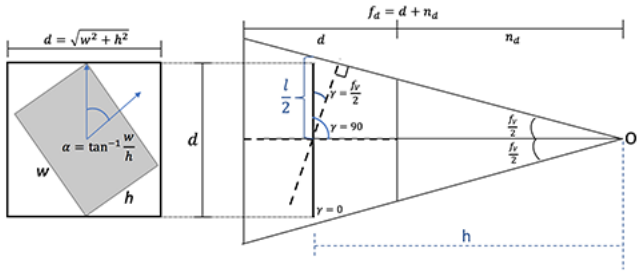


Fig. 4: Image transformation based on roll and pitch angles

TABLE I: Corners in Image Coordinate System

Corner	Image coordinate system
Top Left	$(-\frac{w}{2}, \frac{h}{2}, 0)$
Top Right	$(\frac{w}{2}, \frac{h}{2}, 0)$
Bottom Left	$(-\frac{w}{2}, -\frac{h}{2}, 0)$
Bottom Right	$(\frac{w}{2}, -\frac{h}{2}, 0)$

More specifically, we seek

$$F = PTR_\gamma R_\alpha \tag{6}$$

where R_α is rotation matrix around z-axis, R_γ is rotation axis around x-axis, T is the translation matrix that displace the coordinate system down the z-axis and P is the projection matrix for a square view point with vertical field of view f_v. R_α and R_γ can be easily obtained as below:

$$R_\alpha = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{7}$$

$$R_\gamma = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma & 0 \\ 0 & \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

To calculate the translation matrix T, firstly, we define a variable d representing the side length of the square that would contain any rotation of image.

$$d = \sqrt{w^2 + h^2} \tag{9}$$

Then, we find the hypotenuse h of a right triangle with an opposite side $\frac{d}{2}$, subtending an angle $\frac{f_v}{2}$.

$$h = \frac{d}{2\sin\frac{f_v}{2}} \tag{10}$$

where h denotes the translation of image coordinate down the z-axis. $\mathbf{T}$ matrix is defined as:

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -h \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -\frac{d}{2\sin\frac{f_v}{2}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

Moreover, we need to find the projection matrix P. we define two more value n_d and f_d, which are the distances to the near plane and far plane respectively.

$$n_d = h - \frac{d}{2} \qquad f_d = h + \frac{d}{2} \tag{12}$$

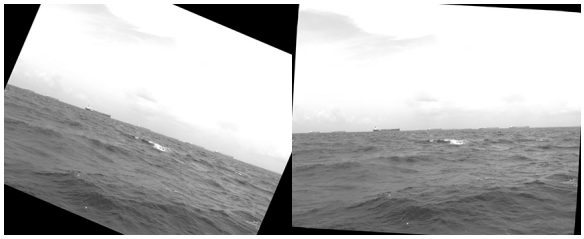
Given the aspect ratio is 1, we can find the projection matrix as follows:

$$\begin{aligned} \mathbf{P} &= \begin{bmatrix} \frac{n_d}{n_d \tan\frac{f_v}{2}} & 0 & 0 & 0 \\ 0 & \frac{n_d}{n_d \tan\frac{f_v}{2}} & 0 & 0 \\ 0 & 0 & -\frac{f_d+n_d}{f_d-n_d} & -\frac{2f_d\times n_d}{f_d-n_d} \\ 0 & 0 & -1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} \cot\frac{f_v}{2} & 0 & 0 & 0 \\ 0 & \cot\frac{f_v}{2} & 0 & 0 \\ 0 & 0 & 1 & -\frac{f_d+n_d}{f_d-n_d} & -\frac{2f_d\times n_d}{f_d-n_d} \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{13}$$

The length of square output image l can be obtained from the distance between the center point of the image and the point directly above it on the upper plane of the viewing frustrum, which is $h\tan\frac{f_v}{2}$. Hence the square length l is twice that distance.

$$l = 2h\tan\frac{f_v}{2} = 2\frac{d}{2\sin\frac{f_v}{2}}\tan\frac{f_v}{2} = d\sec\frac{f_v}{2} \tag{14}$$

We select two images from our data to do the image transformation by using the roll angles obtained from the IMU sensor and horizon line detection respectively as shown in Fig.5.The horizon line rotated by IMU measured roll angle is not appear horizontally well in the image; therefore, we can infer that the rotation angles calculated by horizon line are better than obtained from IMU. Hence, we use the roll and pitch angles derived by horizon line detection instead of the IMU sensor.



(a) IMU (b) Vanishing line

Fig. 5: Image transformation using roll angles obtained from IMU and vanishing line

III. MOTION PARALLAX

A translation of the camera causes a practical translation of the object relative to the camera, and resulting image motion of points and lines reveal their 3-dimensional geometries. This fact is known as motion parallax.

We suppose that the camera translates from O to O' with rotation R. Since we use pitch and roll angles to do the image transformation before, we only need to consider the rotation against the Y-axis with angle β, as shown in Fig.6. The vector $\overrightarrow{O'P}$ is equally rotated R^{-1} with reference to the new frame.

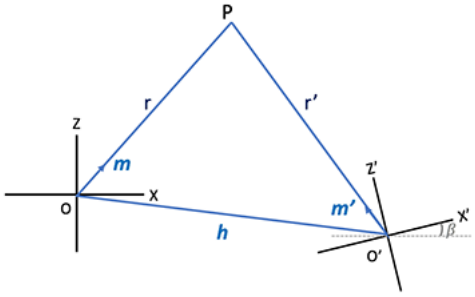


Fig. 6: Illustration of motion parallax

Therefore, we have the vector relation:

$$\overrightarrow{O'P} = R^{-1}(\overrightarrow{OP} - \overrightarrow{OO'}) \tag{15}$$

Let $\boldsymbol{h}$ denotes translation vector $\overrightarrow{OO'}$, set $\boldsymbol{h} = (h1, h2, h3)^T$, where $\boldsymbol{h}$ is the translation vector in the camera coordinate; however, we can only obtain the translation vector $\boldsymbol{H} = (H_1, H_2, H_3)^T$in world coordinate from GPS. We only consider the 2D case which means that $h2$ and H_2 are zeros. Then, we can calculate translation vector $\boldsymbol{h}$ by using yaw angle β of the camera to the world north.

$$\boldsymbol{h} = \begin{bmatrix} h1 \\ h3 \end{bmatrix} = \begin{bmatrix} \cos\beta & \sin\beta \\ -\sin\beta & \cos\beta \end{bmatrix} \begin{bmatrix} H_1 \\ H_3 \end{bmatrix} \tag{16}$$

We assume that a world point P has been observed twice as (x, y) at local frame with the origin O, (x', y') at local frame with the origin O'. For convenience, we convert the image points into unit vectors:

$$\boldsymbol{m} = \frac{1}{\sqrt{x^2+y^2+f^2}} \begin{pmatrix} x \\ y \\ f \end{pmatrix} = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} \tag{17}$$

and

$$\boldsymbol{m}' = \frac{1}{\sqrt{x'^2 + y'^2 + f^2}} \begin{pmatrix} x' \\ y' \\ f \end{pmatrix} = \begin{pmatrix} m'_1 \\ m'_2 \\ m'_3 \end{pmatrix} \tag{18}$$

Hence (15) can be written as $r'\boldsymbol{m}' = R^{-1}(r\boldsymbol{m} - \boldsymbol{h})$. Then, we can compute the motion parallax. Let,

$$\boldsymbol{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = r\boldsymbol{m} - r'R\boldsymbol{m}' - \boldsymbol{h} \tag{19}$$

In the presence of noise, a may not be a zero vector. Therefore, we proceed to look for the minimum value of $\|\boldsymbol{a}\|^2$ in an attempt to find the optimal r and r'. Define the residual $\mathbf{E}$ as:

$$\mathbf{E} = \boldsymbol{a}^T\boldsymbol{a} = \begin{pmatrix} a_1, a_2, a_3 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \|\boldsymbol{a}\|^2 \tag{20}$$

Take the first derivative of $\mathbf{E}$ with respect to r:

$$\begin{aligned} \frac{\partial E}{\partial r} &= \boldsymbol{a}^T \frac{\partial \boldsymbol{a}}{\partial r} + \frac{\partial \boldsymbol{a}^T}{\partial r}\boldsymbol{a} \\ &= 2\boldsymbol{a}^T\boldsymbol{m} \end{aligned} \tag{21}$$

Substitute $\boldsymbol{a} = r\boldsymbol{m} - r'R\boldsymbol{m}' - \boldsymbol{h}$ into (21) and set it to zero, We can get:

$$r - r'(\boldsymbol{m}, R\boldsymbol{m}') - (\boldsymbol{h}, \boldsymbol{m}) = 0 \tag{22}$$

Repeat the above steps for r', we can get:

$$r(\boldsymbol{m}, R\boldsymbol{m}') - r' - (\boldsymbol{h}, R\boldsymbol{m}') = 0 \tag{23}$$

From (22)and (23), $r$and r' can be solved:

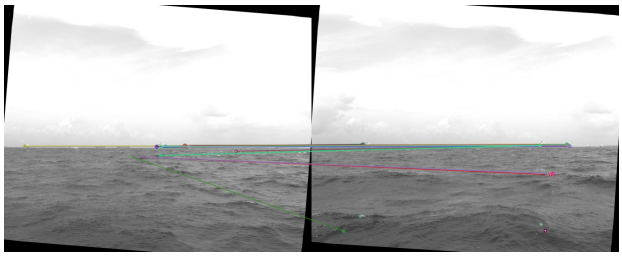
$$r = \frac{(\boldsymbol{h}, \boldsymbol{m}) - (\boldsymbol{m}, R\boldsymbol{m}')(\boldsymbol{h}, R\boldsymbol{m}')}{1 - (\boldsymbol{m}, R\boldsymbol{m}')^2} \tag{24}$$

$$r' = \frac{(\boldsymbol{m}, R\boldsymbol{m}')(\boldsymbol{h}, \boldsymbol{m}) - (\boldsymbol{h}, R\boldsymbol{m}')}{1 - (\boldsymbol{m}, R\boldsymbol{m}')^2} \tag{25}$$

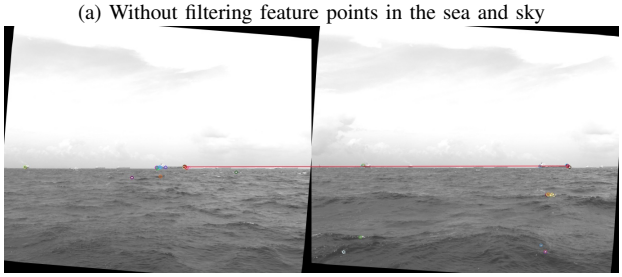
After that, we substitute the solved r and r' and compute the residual $\mathbf{E}$ again. $\mathbf{E}$ ideally should equal to zero, but it is impossible due to noise in reality. We can use value $\mathbf{E}$ to estimate the accuracy of r and r'. In the experiments, we set $\mathbf{E}$ as a threshold to improve the obstacle detection performance, which will be discussed later.

IV. FEATURE MATCHING

As mentioned in the motion parallax section, $\boldsymbol{m}$ and $\boldsymbol{m}'$ are obtained from matched features in a pair of images. To detect matched features, we apply the oriented Fast and rotated BRIEF (ORB) method [8]. ORB method is based on the FAST keypoint detector [9] and the visual descriptor BRIEF (Binary Robust Independent Elementary Features)[10]. Compared to SIFT method, it is faster and more efficient. Since ships are mainly located near the sea horizon, we filter out the matched features in the sky or sea and focus on the matched features closed to the horizon line.



(a) Without filtering feature points in the sea and sky



(b) Filtering out feature points in the sea and sky

Fig. 7: ORB feature matching

V. EXPERIMENTAL RESULTS

A. Dataset

Since there is no public data set for vision-based obstacle mapping in a maritime environment, we evaluate this system by our own collected data. The dataset includes:

- Seq-1: contains 90 consecutive grey frames. A stationary boat about 80m away from USV appears in each frame, with radar data as ground truth.
- Seq-2: contains 100 consecutive grey frames. A stationary ship about 300m away from USV appears in each frame, with radar data as ground truth.
- Seq-3: contains 800 consecutive grey frames. Seq-2 is cut form Seq-3. This sequence shows the scenes when the USV is traveling along a loop on the sea.

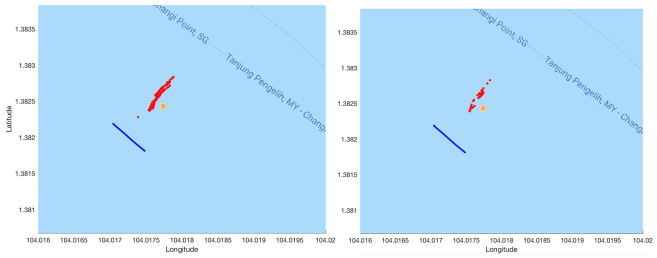
B. Performance Evaluation

Firstly, we evaluate the relationship between baseline and the variances and accuracy of our proposed system by analyzing Seq-1 data. We use different image-steps to represent the different size of the baseline. Image-steps means the number of images between the pair of images used in motion parallax computation. A smaller image-steps means a smaller baseline. According to our USV locations in geodetic coordinates, we can get the location of the obstacle or feature points in geodetic coordinates by triangle geometry. The variances of the latitudes and longitudes with each feature point are calculated respectively. The accuracy of the obstacle locations is also computed using the radar data. From Table II, we can deduce that the obstacle detection with a larger baseline leads to smaller variances and higher accuracy. However, the image-steps value cannot be set too large. If the number of images that obstacles appear in is less than image-steps, the obstacle will not be detected. We draw the obstacle map for all reconstructed points on Google Map as shown in Fig. (8).

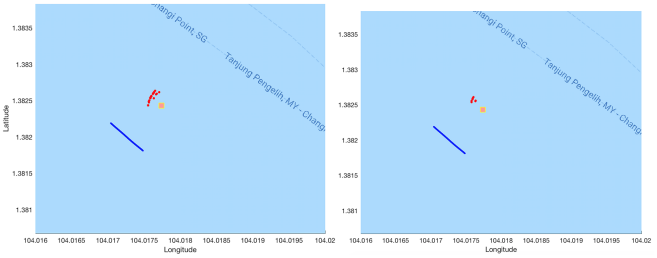
TABLE II: The variances and accuracy of obstacle location (feature points) in Seq-1 with different image-steps. The smallest variance and the highest accuracy are highlighted.

image-steps	Var Lat	Var Lon	accuracy (%)
10	1.3249E-08	5.9431E-09	84.2691082
30	1.0469E-08	5.0493E-09	83.9373327
50	2.9625E-09	1.5971E-09	94.1982197
70	**4.4972E-10**	**4.849E-10**	**96.9281328**

(a) Static obstacle in Seq-1



(b) image-steps=10 (c) image-steps=30



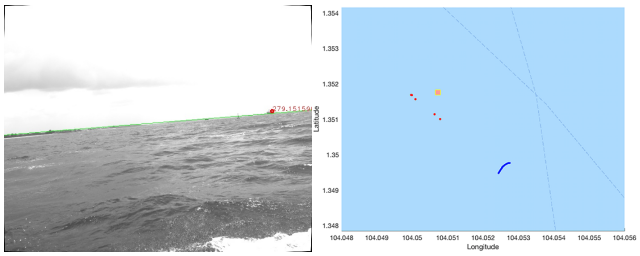
(d) image-steps=50 (e) image-steps=70

Fig. 8: Obstacle mapping results of Seq-1. The red points are the obstacle (feature points) location, the blue curve is the trajectory of our USV, and the orange square is the radar data.

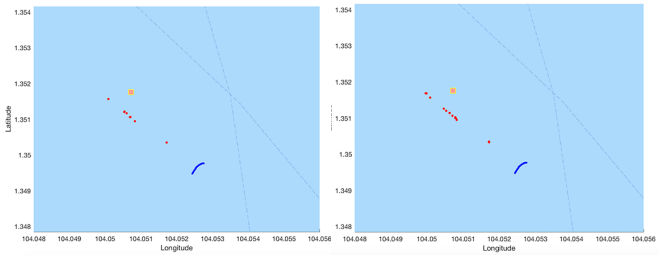
Secondly, we set a threshold based on the residual **E** value and evaluate its effect on the variances and accuracy of our system by analyzing Seq-2 data. We set image-steps equal to 50 and use different threshold values. Table *III* shows the experiments results with Seq-2. Similar to experiments in Seq-1, feature points on the obstacle in the images are reconstructed to the geodetic coordinates and drawn on Google map as shown in Fig. (9). From Table *III*, we can deduce that the reconstruction with a smaller threshold lead to smaller variances and higher accuracy.

TABLE III: The variances and accuracy of obstacle (feature points) location in Seq-2 with a different threshold. The smallest variance and the highest accuracy are highlighted.

Threshold	Var Lat	Var Lon	accuracy (%)
30	**1.0266E-07**	**1.49383E-07**	**85.19132688**
70	1.1861E-07	2.12864E-07	77.58565148
no limit	1.9227E-07	3.30576E-07	75.23393382



(a) Static obstacle in Seq-2 (b) threshold=30



(c) threshold=70 (d) no threshold

Fig. 9: Obstacle mapping result of Seq-2. The red points are the obstacle (feature points) location, the blue curve is the trajectory of our USV, and the orange square is the radar data.

The result of Seq-3 is shown in Fig. (10). The middle image shows the obstacle mapping; the surrounding images show the corresponding original images. This obstacle map is drawn with image-steps=50 and threshold =30. From this figure, we can see that some of the obstacles can be effectively mapped, especially for the nearby obstacles. The variances of reconstructed feature points of near obstacles is small. However, when the obstacles are far away from our USV, it is difficult for feature points to be matched correctly. As the results, the variances of feature points on the obstacle map increase and the accuracy of the reconstructed locations decrease compared to the nearby obstacles.

From the above experiments results analysis, we can conclude that our obstacle mapping system is susceptible to feature matching. For the stationary obstacles within 1000m, the variances of reconstructed locations are small, and the accuracy is high. For moving obstacles or distant obstacles, the reconstructed positions are not reliable due to incorrect or weak feature matching.

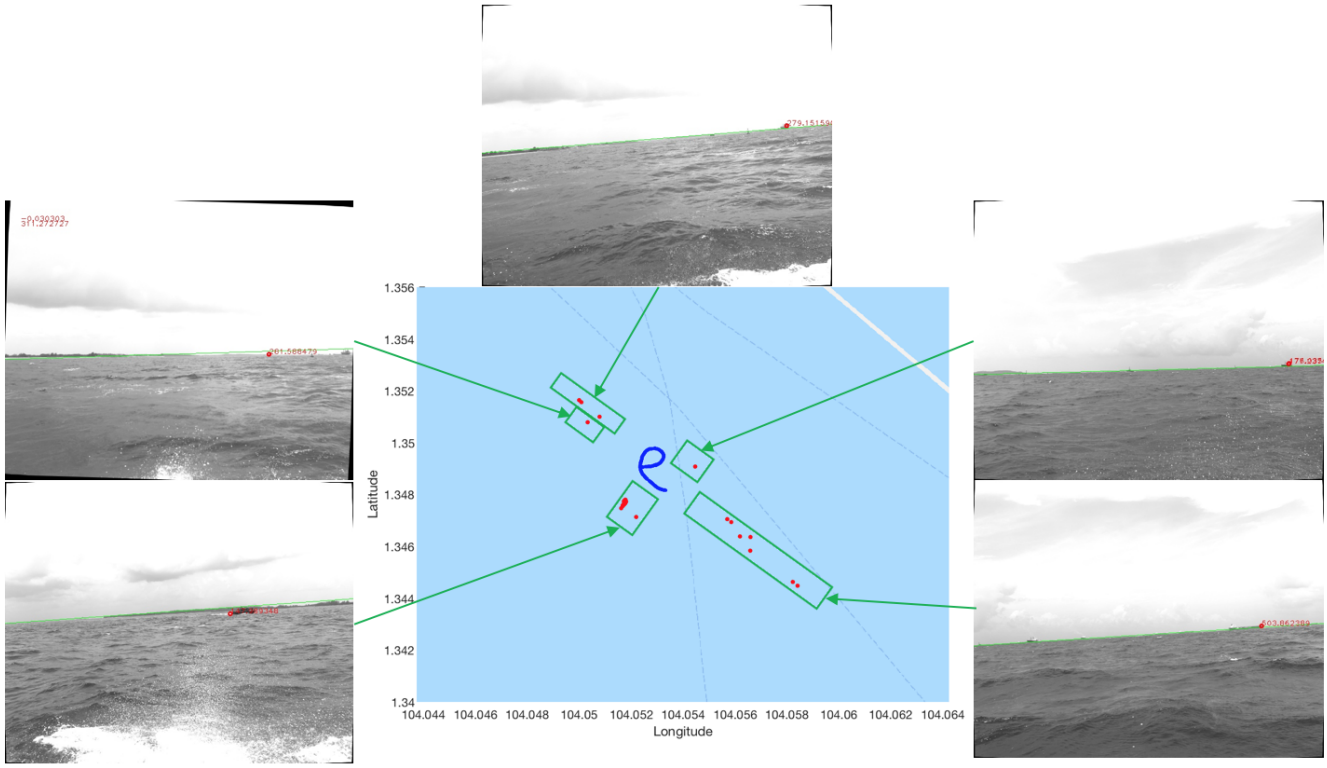


Fig. 10: The result obstacle map of Seq-3. The red points are the reconstructed feature points from obstacles, the blue curve is the trajectory of our USV

VI. CONCLUSIONS

In this paper, we apply a new obstacle mapping system for USV. We use roll and pitch angles to do image transformation so that we can simplify the sea environment to 2-dimensional space. Then, we use the matched feature points obtained by ORB method to compute motion parallax. We also set a threshold based on residual value **E** to increase the accuracy. Finally, by using Google Map API, we draw the obstacle map of reconstructed feature points. From our experiments data analysis, for nearby stationary obstacles, this system has good performance, which means the system can map obstacles with high accuracy and small variances.

REFERENCES

[1] X. Mou and H. Wang, "Wide-baseline stereo-based obstacle mapping for unmanned surface vehicles," *Sensors*, vol. 18, no. 4, p. 1085, 2018.

[2] B.-S. Shin, X. Mou, W. Mou, and H. Wang, "Vision-based navigation of an unmanned surface vehicle with object detection and tracking abilities," *Machine Vision and Applications*, vol. 29, no. 1, pp. 95–112, 2018.

[3] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, no. 5828, p. 133, 1981.

[4] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[5] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International journal of computer vision*, vol. 81, no. 2, p. 155, 2009.

[6] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1841–1848.

[7] 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*, vol. 24, no. 6, pp. 381–395, 1981.

[8] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE international conference on*. IEEE, 2011, pp. 2564–2571.

[9] 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*, vol. 32, no. 1, pp. 105–119, 2010.

[10] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*. Springer, 2010, pp. 778–792.