MONOCULAR VISUAL SLAM ALGORITHM FOR AUTONOMOUS VESSEL SAILING IN HARBOR AREA

Sh. Wang
College of Navigation
Dalian Maritime University
Dalian,China
wangshaobo1@163.com

Yi. Zhang
College of Navigation
Dalian Maritime University
Dalian,China
zhangyj@dlmu.edu.cn

F. Zhu
College of Navigation
Dalian Maritime University
Dalian,China

***Abstract*—In recent years, Self-driving technology received extensive attention of the society from all walks of life. The self-driving cars developed by major technology companies in the whole world have been able to drive on the road. At the same time, the concept of autonomous vessel has been proposed in the past few years. Without any prior information, the SLAM (Simultaneous Localization and Mapping) algorithm, comprises the simultaneous estimation of the state of a robot equipped with on-board sensors and the construction of a model (the map) of the environment that the sensors are perceiving, SLAM is becoming one of the most important components of the driverless sensing module. In this paper, the First part introduces the concept of the autonomous vessel, and then discusses its perception module in detail. After that, it summarizes relevant concepts about SLAM algorithms. The Second part, this paper makes a specific analysis of the current two kinds of monocular visual SLAM(V-SLAM) methods and collects the video data in the harbor environment to carry out experiments, then analyzes and records the relevant experimental results. In the third part, through the analysis and comparison of the experimental results, we discussed the possibility of V-SLAM method applied to autonomous vessel, as well as the problems that may arise in the actual process and attempt to propose some prospective solutions.**

Keywords—autonomous vessel; V-SLAM; feature matching; optical flow; bundle adjustment

I. INTRODUCTION

A. The Concept of Autonomous Vessel

The enormous progress made in electronic sensors, telecommunications, and computer science technologies continue to motivate the development of autonomous vehicles, such as driverless cars. In the area of marine transport, many companies and research institutes have begun to work on the development of autonomous vessels. According to the report of Munich Allianz Insurance Company, 75%-96% of the ship accidents were caused by human errors. Most of the errors were due to fatigue driving or the sailor lack of experience. In addition, the crew usually suffer from the threat of piracy and bad weather. As a result, autonomous vessel will become safer and more efficient.

This paper is sponsored by the National Natural Science Foundation of China (NO.51479021, NO.51679025), and the Fundamental Research Funds for the Central Universities (Grant No.3132016321)..

Driverless technologies mainly include four aspects: situation awareness, navigation and positioning, path planning, decision-making and control. The traditional on-board equipment, such as AIS (Automatic Identification System), RADAR (Radio Detection and Ranging), etc., can provide information about the surrounding environment and other ships, they assist the mariner in making navigational decisions. However, for autonomous vessel, perception means that the sailor's "lookout" is replaced by various sensors. For example, the detection and tracking tasks by visible light cameras, infrared cameras that can work in poor visibility, and short-range RADAR and LIDAR, etc., the sensing module requires the fusion of multiple sensors to ensure the safety of navigation in many aspects. In addition, the traditional two-dimensional electronic charts cannot more specifically reflect the navigation environment, especially in complex water areas such as harbors and narrow waterways. The three-dimensional real-time environment mapping can solve this problem in a good manner. The vessel always equipped with a satellite positioning system, but there is a large risk in the autonomous vessel due to the loss of GPS signal, it will affect navigation safety heavily, so vision-based positioning can be used as an auxiliary method.

B. Visual Simultaneous Localization and Mapping Algorithm

SLAM refers to the body which carries a specific sensor, without any prior knowledge of the environment, estimating its own pose and position during moving, at the same time, building a map of surrounds[1]. If the sensor is mainly a camera, it is called visual SLAM. The basic principle of visual SLAM is multi-view geometry[2]. A classic visual SLAM framework as shown in Fig.1, The front end is also known as VO (visual odometry). VO can estimate the motion between adjacent images and describe the appearance of the surrounding map. The back end component named optimization, in order to get consistent trajectories and maps, it accepts the camera pose measured at different timestamps and loop-closing detection information then optimizes them; loop-closing is to determine if the carrier has arrived at a previous position. At first, SLAM is mainly used in indoor navigation for robots. With the continuous development of the algorithm, visual SLAM is gradually applied to large-scale scenes.

The visual SLAM algorithm has a variety of sensor solutions. Currently widely used are RGB-D depth sensor, stereo camera, and monocular camera. Among them, RGB-D sensors such as Microsoft's Kinect/Kinect V2 can directly obtain spatial depth information, but the RGB-D sensor has a small field of view, the depth range is generally limited to 0.5~5m and the anti-interference capability in outdoor environments is insufficient, so it is generally used in indoor environments. Stereo camera can estimate depth and work outdoors, but stereo vision requires a lot of calculations, in addition, it always needs a precise camera calibration. The monocular camera is small and low cost, it is more convenient to carry and install on-board, however, it is more challenging in implementing the SLAM algorithm[3]. This paper mainly studies the application of monocular visual SLAM algorithm in harbor area.

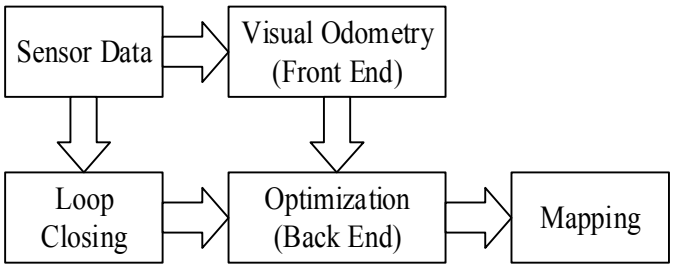

Fig. 1. The Framework of SLAM

II. Related Work

For the application of SLAM in driverless, Bresso, Guillaume, et al.[4]discussed the relation between them, and combined with the actual needs of autonomous driving, explored the establishment of multi-subgraphs, place recognition, multi-vehicle map-building, and other issues, they also proposed six SLAM technology indicators in the context of driverless applications: **Accuracy**(refers to the result of localization need to satisfy the standards), **Scalability**(refers to the capacity of the vehicle to handle large-scale autonomous driving), **Availability**(refers to the SLAM algorithm could be used right away for autonomous driving if sufficiently accurate), **Recovery**(refers to the ability to localize the vehicle inside a large-scale map), **Updatability**(refers to the identification of permanent changes between the map and the current observation), **Dynamicity**(refers to how the SLAM approach is able to handle dynamic environments and changes). Moratuwage, Wilesoma[5] proposed a multi-ship cooperation sea-based SLAM algorithm called CSLAM. They first analyzed the performance of the EKF-SLAM algorithm on a single unmanned surface vehicle, and then correlates the multi-ship data. Although the sensor is a 2D laser, it is an unusual SLAM experiment on the sea surface. South Korea's J.Han[6] and others carried an Omnidirectional camera and a LIDAR on an unmanned surface vehicle. Experiments were conducted in areas where GPS signals were limited, such as under bridges. The three-dimensional model of the bridge can be accurately drawn and the vehicle's position can be located in a local environment. Thomas Kriechbaumer and Kim Blackburn[7] equipped stereo camera on unmanned surface vehicle and used the feature-based SLAM method to test on a 663-meter-long canal. This paper compared the positioning results with the ground truth position, found the error ±0.067 m. George Terzakis in his doctoral dissertation[8] used a sea-surface vehicle developed at Plymouth University named *Springer* to implement the monocular visual SLAM algorithm. He still used the feature-based SLAM algorithm. In addition, he analyzed the sea-surface environment, in the end, this vision-based position method can guarantee the safety of the *Springer* under missing GPS signals condition. These algorithms can be referenced during the research of V-SLAM algorithm in harbor area, but most of them were carried on unmanned surface vehicles in small range water area, the V-SLAM algorithms used are mostly based on feature points, so the monocular SLAM algorithm in the harbor area is still worth exploring.

III. Feature-Based Method

The feature-based method uses image feature points to perform image matching which can realize the camera pose estimation. After that, it triangulates the matched feature points to create cloud map points. Common features include but are not limited to SIFT, SURF, BRIEF, FAST, ORB, etc. Hartmann et al.[9] performed a detailed comparative analysis of the common image features used in the V-SLAM algorithms. Combined with the needs of the V-SLAM algorithm, the ORB feature are extremely fast to compute and match, while they have good invariance to viewpoint, so it is It is more popular these years[10]. The developed feature-based V-SLAM algorithm mostly uses multi-thread processing, namely tracking, local map building and loop-closing detection. In the entire algorithm flow, each thread uses the same feature descriptor, which also increases the camera's relocalization performance (making it possible to recover after point tracking is lost). Therefore, feature-based V-SLAM has good map reusability. In addition, algorithm uses the BA (Bundle Adjustment) method based on key frames to optimize data, and it takes certain criteria to filter point clouds and key frames strictly. Compared with the previous methods such as the EKF filter, the amount of calculation in this type of optimization process is kept at a reasonable operating range. What's more, the algorithm accuracy is Higher, detailed algorithm framework is shown in Fig. 2.

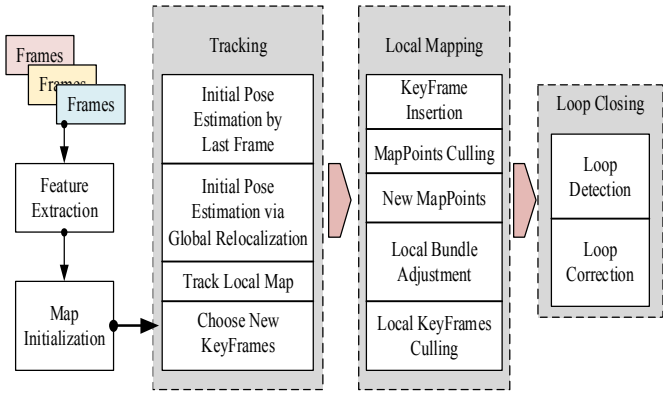

Fig. 2. The Multi-Thread Algorithm for Feature-Based Algorithm

A. Map Initialization

The goal of the map initialization is to compute the relative pose between two frames to triangulate an initial set of map points. The general initialization method is to run two operation models at the same time, that is, the homography matrix facing the planar view and the fundamental matrix

facing the non-planar view. According to some criteria, selecting an appropriate initialization model.

B. Tracking

The images captured by monocular camera are just a two-dimensional projection of the three-dimensional space, so in order to recover the three-dimensional structure, the camera's perspective must be changed. The object distance related to its observed moving speed, thus the disparity value can be obtained through the knowledge of epipolar geometry, triangulating matched points to determine the spatial scale factor. Fig. 3 illuminates the tracking process. T1, T2, T3, and T4 replace keyframes at different times. Each colored dot in the figure represents a map point and its projection on each frame. During the camera movement, map points are observed by cameras at different times, and if the points quality is better, then preserved, such as 3,4,5; some map points are only observed by one or two frames, such as 1,2, in order to maintain the sparse property of map and reduce the amount of calculations in the optimization thread. Points 1 and 2 will be filtered out; in addition, there are new map points such as 6 coming to the map.

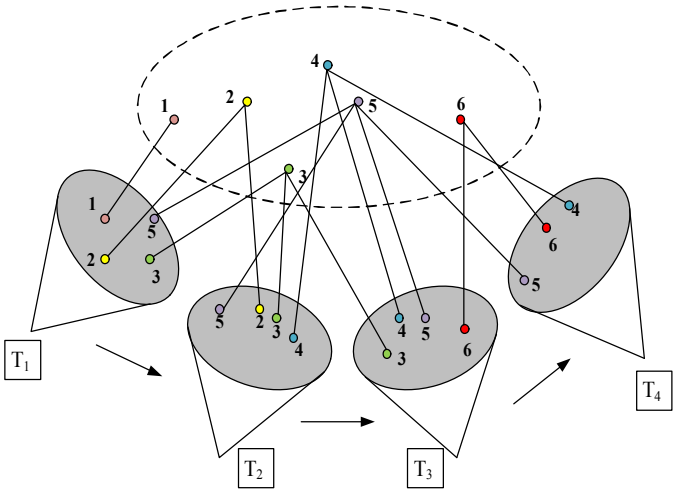

Fig. 3. The illustration of Tracking Thread

Each frame contains a camera motion parameter including information such as the camera's position and orientation, usually expressed as a 3×3 rotation matrix R_i and 3D translation variables T_i, R_i and T_i transform the 3D point pi in the world coordinate system into the local coordinate system C_j corresponding to the camera position parameters:

$$\left(X_i, Y_i, Z_i\right)^T = R_i\left(p_i - \mathrm{T}_i\right) \tag{1}$$

Then the point was projected into the image:

$$h_i = \left(f_x X_i / Z_i + c_x, f_y Y_i / Z_i + c_y\right)^T \tag{2}$$

In this formula, f_x, f_y are respectively the focal length of the image along x, y axis. c_x, c_y is the lens center position in the image. According to (1)(2), it can be seen that the projection position $\mathbf{h_i}$ of the three-dimensional points in the image can be expressed as a function of C_j and p_i:

$$h_i = h\left(p_i, C_j\right) \tag{3}$$

According to (3) at this time, V-SLAM needs to match image points corresponding to the same field in different images. The algorithm uses the optimization model BA(Bundle Adjustment) to minimize the reprojection error and expect to find an optimal set of camera parameters and three-dimensional points. The reprojection error also refers to the second projection:

- The first projection refers to the three-dimensional point projection into the image.
- Then using triangulation to calculate the location of 3D points.
- Finally, using the calculated 3D point coordinates and the camera matrix to perform the second projection.

To make the distance between two projection points as small as possible, solving the following objective function:

$$\arg\min_{p_1 \cdots p_m, C_1 \cdots C_n} \sum_{i=1}^{m}\sum_{j=1}^{n} \| h\left(p_i, C_j\right) - a_i \|_{\sum_{ij}} \tag{4}$$

The symbol a_i represents the position of the observed image point, the process of solving (4) is called BA. There are two main methods for solving BA: Gauss-Newton method and Levenberg-Marquardt (LM algorithm), which are no longer illuminate here because of the complexity of these method. Tracking thread contains the following sections:

1) Initial pose estimation

As explained above, if the tracking is successful after initialization, we usually use the constant speed motion model to predict the pose of the current camera and the points corresponding to the feature points of the previous frame of image are searched for the matching points in the current frame. After that using BA to realize local optimization. If not find the matching points of the current frame, it will use the points near the corresponding points; if the expanded search range still fails to track the feature points, indicating that the previously assumed constant speed motion model is invalid, then we take the global relocalization method. At present, The main method for relocalization is bag of words model, at the same time, there were also novel methods like deep learning.

2) Tracking local map

A set of keyframes Ki are included in the local map. They share common map points with the current keyframe. All the map points in this group of Ki are searched in the current frame and the inappropriate points are removed. The camera pose is finally optimized by obtaining all the map cloud points in the current frame. The purpose of this process is to obtain more

matching points between the current frame and the local map, so as to optimize the current frame pose in a better manner.

3) New keyframes

The last step of tracing thread, we must determine whether the current frame can be used as a keyframe. Using keyframes can speed up the computational efficiency of the algorithm. In the case of difficult external conditions (such as: rotation, rapid camera movement), the algorithm can still achieve robust tracking, while at the same time, When the same environment is revisited, the size of the map is controllable, which is conducive to the long-term work of the system.

C. Local Mapping

- Key frame Insertion: Add a key frame K_i and calculate the bag of words model about this key frame and then generate a new map point using the triangulation method.
- Map Point Culling: In order to preserve these map points, we must take a strict test for them in the first three key frames. The test ensures that the remaining points can be tracked, not produced by error data.
- New Map Points: The new map points are created by triangulating the feature points. The depth information, disparity, re-projection error, and scale consistency in the camera coordinate system need to be reviewed, after that the new point can insert the map as a new point.
- Local Bundle Adjustment: This manner is mainly used to optimize the current processed key frame K_i and the map points. During optimization, all the observation data marked as invalid will be discarded.
- Local Key Frame Culling: If 90% of the key frame K_i can be simultaneously observed by at least three other key frames, then K_i is considered to be redundant and we remove it. When the algorithm is running in the same scene, the number of key frames is controlled in a limited situation. Only when the scene content changes, the number of key frames will increase.

D. Loop-Closing

Although the back-end can improve the accuracy of the algorithm, only the adjacent key frame data is difficult to eliminate the cumulative error, the loop-closing module gives a long-term constraint beyond the adjacent frame, the key of loop-closing is how to effectively detect the camera passing through the same place. If it can be successfully detected, then it provides more conditions for back-end optimization. If the loop-closing is successful, we can get globally consistent trajectories and maps.

E. Experiment

Here is the simple experiment as shown in Fig. 4, Fig. 5, we record data in the harbor, and Fig.4 shows the feature points in each frame, Fig. 5 shows the map points of the environment.

Fig. 4. The Feature Points in Each Frame

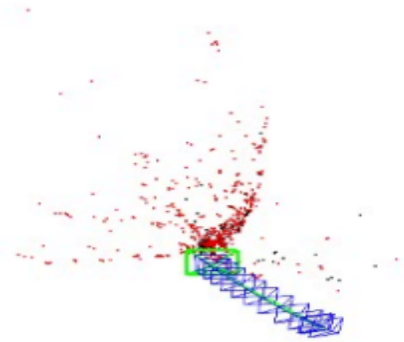

Fig. 5. The Map Points Produced by Feature-Based Method

IV. Optical Flow Method

Optical flow is used to describe the motion of pixels between images over time, it estimates camera motion by using the luminance information difference of each pixel in adjacent frame images. Optical flow V-SLAM method is robust around the environment where lack of scene texture and with sparse feature points[11]. The optical flow method puts data association and pose estimation in a unified nonlinear optimization. However, the feature-based algorithm solves step by step, that is, the data association is first obtained by matching feature points, then estimating the pose based on the association of these points.

The optical flow method will always solve a more complex optimization problem, namely minimizing the photometric error. Each 3D point, starting from a host frame, is multiplied by a depth value and then projected to another target frame, thereby creating a projection residual. As long as the residuals are within a reasonable range, these points can be considered as being projected from the same point. In this type of method, it will try to project each point to all frames, calculate its residual in each frame, but does not care about the one-to-one correspondence between points and points. The back end of the optical flow method uses a sliding window consisting of several key frames. This window is always present in the entire algorithm, and there is a set of methods to manage the new data and remove old data.

A. The Optical-flow algorithm flow

As shown in Fig. 6, After processing each frame, the algorithm stores the data of each frame in the type of Frame-Hessian structure. In monocular V-SLAM, when all map points are observed at the beginning, there is only a 2D pixel coordinate whose depth is unknown, which is called immature point in the optical flow, along with the camera's motion, the algorithm tracks the immature points on each image. If the depth of the immature points is converged during the tracking process, we can get the three-dimensional coordinates of the immature points and form a normal map point.

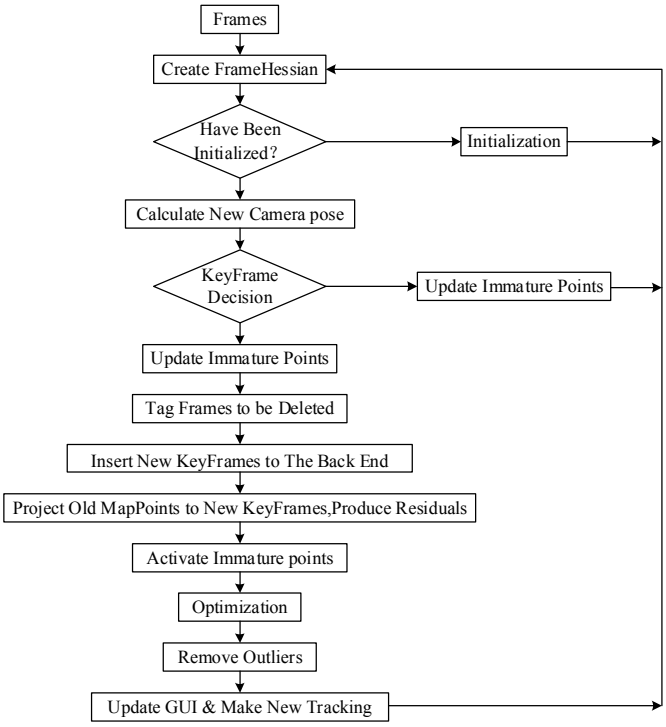

Fig. 6. The Algorithm Flow of Optical Flow Method

The optical flow method is still based on key frames. For non-key frames, the algorithm only calculates the pose, and uses this frame to update the depth estimation of each immature point. The main processing for each key frame is: adding new residuals, removing the wrong residuals, and extracting new immature points. In the optimization part, it also uses BA method, but the objective function is not to minimize the re-projection error, but to minimize the photometric energy function. In addition, the back-end uses the sliding window optimization algorithm. These are explained below.

B. Photometric Energy Function

In feature-based method, we obtain the positions of pixels by a pair of matching points through feature matching, then we can calculate the position of the re-projection. In the optical flow method, there is no feature tracking, it is not clear that the two points correspond to the same point in space. Therefore, the optical flow method is based on the current camera pose estimate to find the corresponding pixel position. The better camera pose estimation value we got and the higher accuracy we realized, this optimization problem is solved by minimizing the photometric error. In simple terms, it is the brightness error of the corresponding pixel:

$$e = I_1(p_1) - I_2(p_2) \tag{5}$$

This formula is very simple. I represents the image and p represents the pixel. However, in practice, obtaining photometric errors is much more complicated:

$$E_{pj} = \sum_{p \in N} w_p \left\| (I_j[p'] - b_j) - \frac{t_j e^{a_j}}{t_i e^{a_i}} (I_i[p] - b_i) \right\|_\gamma \tag{6}$$

As shown in (6), It defines the photometric error function about p and p'. It may also be called the energy function E_{pj}. I_i, I_j are frames corresponding to p and p' respectively. N is the error calculation model for point p, t Represents the exposure time, $\|\cdot\|_\gamma$ is the Huber function. In practice, the pixel intensity calculation function I requires camera photometric calibration[12]. Affected by various parameters, and in order to make the above formula to be established when the exposure time is unknown, an affine brightness transfer function is introduced here:

$$Y_{affine} = e^{-a_i}(I_i - b_i) \tag{7}$$

In this equations, a_i and b_i are transformation parameters. Finally, all formulas are integrated by a gradient-dependent weight coefficient w_p. By summing all photometric errors, the total photometric error is obtained, which is the photometric energy function:

$$E_{photo} = \sum_{i \in K} \sum_{p \in P_i} \sum_{j \in obs(p)} E_{pj} \tag{8}$$

Here, i has traversed all frames, p has traversed all pixels in frame i, j over all frames which can observe p.

C. Sliding Window Optimization Algorithm

In the optical flow method, the back-end aims to optimize the pose and map by minimizing the energy function (8). However, as the amount of optimization gradually increases, the amount of calculation of the algorithm also increases, so it is impossible to add variables without limitation, the sliding window algorithm can solve this problem. As shown Fig. 7, the window can hold 4 frames.

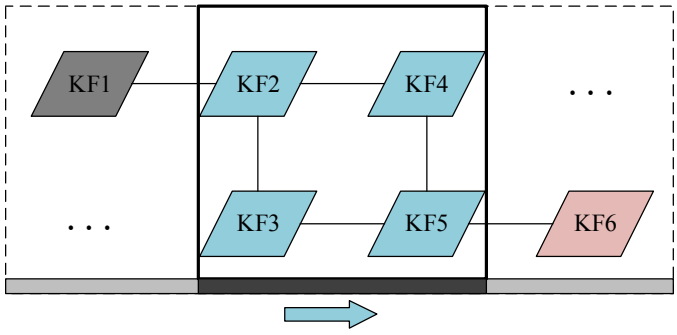

Fig. 7. The Illustration of Sliding Window Algorithm

In the sliding process, KF1 is removed, and at the same time, KF6 will join in the next moment. The algorithm only

optimizes the frames in the window, thus fixing the calculation amount. But this will produce a very intuitive problem, the original constraint relationship between KF1 and KF2 is destroyed, resulting in a very serious loss of information, so the core problem of sliding window algorithm is fixed at the same time, but also fully Reserved information.

Now defining the state relationship between frames:

$$z_{ij} = h_{ij}(x_i, x_j) + \delta_{ij} \tag{9}$$

x is the camera pose, **z** is the correspondence between adjacent poses and δ represents the random noise of the environment. Now, we define $x = \{x_a, x_b, x_c\}^T$. x_a denotes the variable to be eliminated, x_b denotes a variable that has a constraint relationship with x_a, x_c denotes another variable. Here we define a constraint relationship: $z = \{z_m, z_n\}$, z_m represents the constraint between x_a and x_b and z_n represents the constraint between x_b and x_c. Now we want to eliminate x_a to optimize x_b and x_c. In order to preserve the information z_m, the right way is to store z_m as a priori information of x_b, that is to tell x_c that x_b and x_a had an agreement before, expressed as the probability x_b under z_m conditions:

$$p(x_b \mid z_m) = \int_{x_a} p(x_b, x_a \mid z_m) d_{x_a} \approx \mathcal{N}(\hat{x}_b, \Lambda_t^{-1}) \tag{10}$$

The above equation encapsulates the constraints between x_a and x_b as prior information for $x_b - \mathcal{N}(x_b, \Lambda_t^{-1})$, in order to solve this priori information, we simply need to solve:

$$\arg\min_{x_b, x_a} \sum_{i,j \in z_m} \frac{1}{2} \| z_{ij} - h_{ij}(x_i, x_j) \|^2_{\Lambda_{ij}} \tag{11}$$

When we solve this nonlinear least square problem, we can get the information matrix as follows:

$$H = \begin{pmatrix} H_{aa} & H_{ba}^T \\ H_{ba} & H_{bb} \end{pmatrix} \tag{12}$$

And we do Schur Complement, then get $\hat{x}_b, \Lambda_t^{-1}$:

$$\begin{aligned} (H_{bb} - H_{ba} H_{aa}^{-1} H_{ba}^T) \hat{x}_b &= b_b - H_{ba} H_{aa}^{-1} b_a \\ \Lambda_t &= (H_{bb} - H_{ba} H_{aa}^{-1} H_{ba}^T) \end{aligned} \tag{13}$$

The prior information is confirmed, so solving x_a, x_b, x_c full-state problems can eliminate x_a without lose information:

$$\hat{x} = \arg\min_x \frac{1}{2} \| \hat{x}_b - x_b \|^2_{\Lambda_t} + \sum_{(i,j) \in z_n} \frac{1}{2} \| z_{ij} - h_{ij}(x_i, x_j) \|^2_{\Lambda_{ij}} \tag{14}$$

The above process is also called marginalization. With this method, the optical flow can maintain the prior information between frames and apply this information to the solution of BA.

D. Experiment

We use the same dataset to test optical flow method in harbor area. Fig. 8 shows the optical flow and Fig. 9 shows map points.

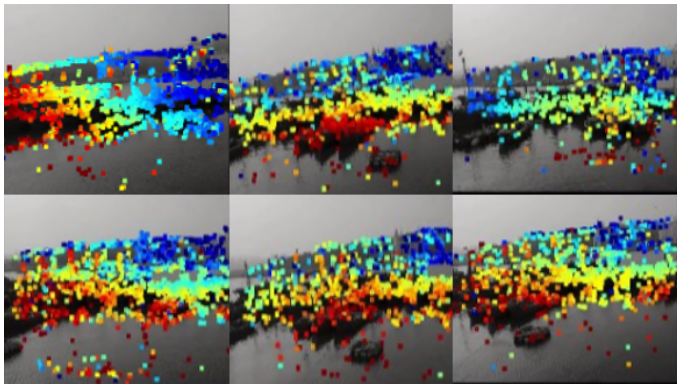

Fig. 8. The Optical Flow Shown in Each Frame

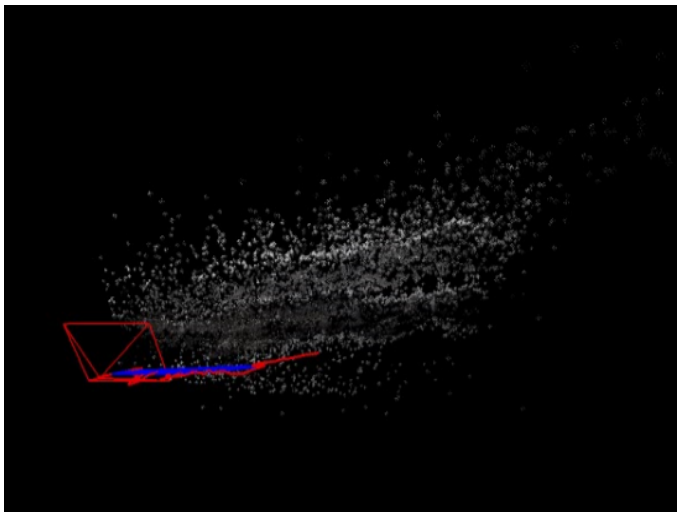

Fig. 9. The Map Points Produced by Optical Flow Method

V. Conclusion and Future Work

The biggest difference between the optical flow method and the feature-based method is that the optical flow method deals with the data association problem in a more elegant way. The feature-based method needs to rely on the feature extractor and the correct feature matching in order to calculate the camera motion correctly. When the environment has more corner points and the texture is good, the feature-based method may be a good choice, but this kind of environment also applies to the optical flow method. According to the experimental results, under the large-scale scene of the harbor, most of the images are occupied by the sea surface, with fewer corner points, and there are many repeated textures in the environment. So, feature-based method will perform badly. In contrast, the optical flow method is much better. The optical flow method gives us the ability to track smooth blocks, but it has high

requirements on the quality of the image. It also lacks the rotation invariance and illumination invariance than the feature-based method, which bring a great challenge to the calibration of the onboard camera. Feature matching, corner detection algorithms have become more and more sophisticated, and the accuracy of focus matching has become higher. This has made the feature-based method more accurate in solving the pose of the camera, although this paper does not compare the positioning accuracy between two kinds of methods, it is not difficult to find that the optical flow method only requires that the previous point has a reasonable projection residual in the current image. The projection success depends on our judgment of the map point depth and the camera pose. Although the back-end optimization method is further improved, it is difficult to improve the accuracy of the pose resolution.

On the other hand, the optical flow method lacks loop-closing detection mechanism and inevitably produces a cumulative error. In terms of map reusability, the optical flow method is difficult to reuse the current map unless all the keyframes are stored. Even if all the keyframes are stored, a very accurate initial estimate of the current pose is required. However, this is very difficult. The feature-based method is relatively simple on map reusability, it only needs to store all the feature points, and the current image can be characterized to solve the camera pose. Therefore, the optical flow method is more suitable for positioning continuous images, and the feature point method is more suitable for global matching and loop-closing detection.

Future Work: Providing the three-dimensional environment model for autonomous vessels sailing in the harbor area and simultaneously determining the relative position of the vessel in the local environment is our goal. After the discussion of this two V-SLAM algorithms, we will analyze the accuracy of pose calculations and the accuracy of the environmental model more quantitatively in next step. What's more, we will improve the accuracy and robustness of the system through various methods:

- We will improve and combine feature-based method and optical flow method. Considering the disadvantages of sparse feature points in the harbor waters, we will try to use edge features to improve the robustness of the feature point method, optimize the accuracy of the pose method for the feature-based method. The optical flow method helps us to find more map cloud points, therefore, the front-end should continue to maintain the excellent characteristics of the optical flow method, and the back-end pose optimization is added to the edge feature to correct the minimum photometric error results of the direct method, preserving and storing edge and intersection features and provide a closed-loop detection mechanism for the optical flow method to eliminate Cumulative error.
- We will explore the "vision +" multi-sensor SLAM model, such as vision + IMU and vision + LIDAR, to reconstruct the scene more perfectly with multiple sensors and improve the pose accuracy.
- Add a map optimization and processing module. We expect that maps can be used for positioning, navigation, obstacle avoidance, and interaction. It is difficult to save and reuse maps constructed by the optical flow method. Sparse maps based on the feature point method are difficult to use for these functions. Therefore, how to effectively processing maps is one of the future research tasks. For example, the feature points are treated as OCTOMAP dense maps and additional information such as scale distances is used to facilitate the path planning of autonomous vessels.

In addition, there are many problems that need to be solved, such as how to deal with dynamic targets in the port environment, how to combine SLAM with MOT and how to understand the scenarios, such as identifying beacons. how to combine with existing electronic charts and so on, these issues require us to work harder.

REFERENCES

[1] Cadena C, Carlone L, Carrillo H, et al. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. IEEE Transactions on Robotics, 2016, vol.32(6), pp.1309-1332.

[2] R.Hartley and A.Zisserman, Multiple View Geometry in Computer Vision, 2nd ed., Cambridge, U.K.: Cambridge Univ. Press, 2004.

[3] Younes G, Asmar D, Shammas E, et al. Keyframe-based monocular SLAM: design, survey, and future directions[J]. Robotics and Autonomous Systems, 2017, vol.98, pp. 67-88.

[4] Bresson, G., Alsayed, Z., Li, Y., & Glaser, S. Simultaneous Localization And Mapping: A Survey of Current Trends in Autonomous Driving. IEEE Transactions on Intelligent Vehicles.

[5] Moratuwage, M. D. P., Wijesoma, W. S., Kalyan, B., & Dong, J. F.. Collaborative multi-vehicle localization and mapping in marine environments. Paper presented at the Oceans, 2010.

[6] Han, J., & Kim, J. (2013, Oct. 30 2013-Nov. 2 2013). Navigation of an unmanned surface vessel under bridges. Paper presented at the 2013 10th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI).

[7] Kriechbaumer, T., Blackburn, K., Breckon, T. P., Hamilton, O., & Rivas, C. M. Quantitative Evaluation of Stereo Visual Odometry for Autonomous Vessel Localisation in Inland Waterway Sensing Applications. Sensors, 2015,vol.15(12), 31869-31887.

[8] Terzakis, G. . Visual Odometry and Mapping in Natural Environments for Arbitrary Camera Motion Models. PhD Thesis, Plymouth University, 2016.

[9] Hartmann, J., Klussendorff, J. H., & Maehle, E. A comparison of feature descriptors for visual SLAM. European Conference on Mobile Robots, ECMR 2013.

[10] R. Mur-Artal, J. M. M. Montiel, and J. D. Tard´ os, "ORB-SLAM: a versatile and accurate monocular SLAM system," IEEE Transactions on Robotics, 2015, vol. 31, no. 5, pp. 1147–1163.

[11] Engel, J., Koltun, V., & Cremers, D. Direct Sparse Odometry. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017.

[12] Engel, J., Usenko, V., & Cremers, D. A Photometrically Calibrated Benchmark For Monocular Visual Odometry.2016.