A High-accuracy and Semi-dense Feature-based VSLAM System

ZHEN LI ^{*†}	Member,	YUREN DU [‡]	Non-member
MIAOMIAO ZHU [†]	Member,	SHI ZHOU [†]	Member
SEIICHI SERIKAWA [†]	Member,	LIFENG ZHANG [†]	Member

(Received April 13, 2021, revised October 8, 2021)

Abstract: SLAM (Simultaneous Localization and Mapping) technology is the basis of a mobile robot, furthermore the most important condition for robot intelligence. VSLAM (Visual Simultaneous Localization and Mapping) [1] technology uses visual sensors to build incremental environment maps and simultaneously realize the self-positioning of a moving robot. Currently, feature-based VSLAM systems tend to build sparse maps, but sparse maps are not accurate enough for navigation and obstacle avoidance applications. Therefore, this paper proposed a high-accurate and semi-dense VSLAM system with a monocular camera. The proposed work used polar line searching and block matching modules to help feature-based VSLAM systems accurately measure depth information. Experimental results show that the proposed semi-dense VSLAM system can reconstruct more dense point cloud maps. The synchronization tracking trajectory accuracy of the system is 9.13% higher than that of the ORB-SLAM2 system.

Keywords: VSLAM, Monocular, Semi-dense, Feature-based

1. Introduction

Simultaneous self-positioning and map reconstruction algorithms can track robots' current position and reconstruct an environment map simultaneously. VSLAM uses visual sensors to perceive the environment around the robot. A real-time, high-accuracy and robust VSLAM algorithm is the fundamental condition for intelligent robots so that these robots can meet the requirements of obstacle avoidance, path planning, and autonomous navigation. Currently, VSLAM technology is mainly applied to self-driving cars, sweeping robots, and unmanned aircraft. Compared with LiDAR, IMU (Inertial Measurement Unit), and GPS (Global Positioning System) sensors, visual sensors could cost less and consume less computation. With the improvement of algorithms and chips' processing performance, the accuracy of VSLAM has also improved a lot. Especially when robots come to the areas where GPS signals are not available, such as indoor rooms and remote mountains, VSLAM systems become irreplaceable. The main research directions of VSLAM can be divided into two parts: direct method VSLAM and feature-based method VSLAM.

LSD-SLAM (Large-Scale Direct Monocular SLAM) system is a representative work of direct method VSLAM. LSD-SLAM system does not need to extract feature points in keyframes, and they can directly extract each pixel point in images. So LSD-SLAM system can easily reconstruct a semi-dense point cloud map of the surrounding environ-

ment. Besides, LSD-SLAM can work stably on a texture-less object and is also stable and reliable in implementing tracking, self-positioning, and reconstruction of large-scale maps [3]. But, the LSD-SLAM system is very sensitive to illumination changes and is not so accurate as feature-based method VSLAM systems.

ORB-SLAM2 (Oriented FAST and Rotated BRIEF SLAM) [6] system is also a representative work based on ORB (Oriented FAST and Rotated BRIEF) feature points. The system needs to extract feature points and descriptors by comparing two neighboring keyframes to ensure the high accuracy of the VSLAM system. The ORB-SLAM2 system is robust, and it can relocate the robot's position due to the loop closing module. In addition, ORB-SLAM2 can be used in complex illumination environments. The main short point of feature-based SLAM is that it can only reconstruct very sparse point cloud maps, which are unsuitable for robotic tasks such as navigation or object interaction. After comparing the above two typical VSLAM systems, an excellent VSLAM system needs to reconstruct a highly accurate and semi-dense point that could map and be robust enough in complex illumination change environments. Based on these requirements, this paper proposed a high-accuracy and semi-dense feature-based VSLAM system. This system is composed of the following four modules [2]: tracking, local mapping, loop closing detection, and semi-dense mapping, and the system structure diagram is shown in Fig. 1.

2. The structure of the proposed VSLAM system

2.1 The tracking module The main purpose of this module is to find more congruent relationships between the

* Corresponding: li.zhen799@mail.kyutech.jp

[†] Kyushu Institute of Technology
(zhang@elcs.kyutech.ac.jp)

[‡] Yangzhou University
(240462160@qq.com)

Figure 1: The structure diagram of the proposed VSLAM system

current frame and the local map to optimize the pose of current frames. The following three movement models in the tracking module are all used to get the pose of the current frame.

(a) Tracking with motion model

Assuming that the object was moving at a uniform speed, the robot pose of the current frame could be estimated by the pose and velocity of the last frame. The speed of the last frame can be calculated by several previous frames. This model is suitable for the cases that camera is moving along with the straight line at a uniform speed.

(b) Tracking with neighboring keyframes model

If the above motion model has failed, you can try to match the last keyframe, because the current frame is closely related to the last keyframe. We can use BoW (Bag of Words) to speed up the matching process. Firstly, we calculate the BoW of the current frame, and set the pose of the last frame as initial pose of current frame; Secondly, the system need to match corresponding feature points according to pose and BoW dictionary of the two neighbouring keyframes; Finally, the feature points matching is used to optimize robot pose.

(c) Relocalization model

If the current frame also fails to match the nearest neighboring keyframe, this means that the current keyframe has been lost. At this point, the system has to match all the keyframes to relocate the current keyframe in the whole map. Firstly, the system calculates the BoW vector of the current keyframe.

Secondly, several keyframes are chosen as alternatives in BoW dictionary. Thirdly, one of keyframes with enough feature points is selected. Finally, the robot pose can be determined by feature points matching. Additionally, the system tends to select the keyframe that has enough interior feature points.

2.2 The local mapping module The current keyframe is inserted into the map, but some keyframes are found redundant after judgement, the system has to delete them for maps reconstructing speed. Due to triangulation errors, the actual one point may be predicted into two points by mistakes. The two points need to be fused into one finally.

The local mapping module is associated with the previous tracking module and the following loop closing module, and keyframes are generated by the tracking module. The method of local BA (Bundle Adjustment) for the local mapping is usually used in no loop closing situations, so that the keyframes queue could maintain idle conditions. So as to make sure the high speed for loading and saving maps.

2.3 The loop closing detection module The loop closing detection module mainly consists of two steps: 1. Loop closing information detection and confirmation; 2. Modify and optimize the robot pose. Regardless of monocular, binocular or RGBD cameras, all of them exist the problem of accumulated measurement drifts [7]. As the extension of robot movement scale, the error of the previous keyframes will be transmitted to the following keyframes, leading to a very large error in the pose of the latest keyframe in world coordinate system. In addition to using other optimization methods to adjust the local and global poses, loop closing detection module is also a good

way to optimize the pose globally.

In other words, when a moving robot returns to the same previous poses, the whole map will form a closed loop. The camera can automatically identify the previous location, and then transmit this information back to the entire map and optimize the map. The whole map would get fairly accurate mapping information. Therefore, loop closing detection module is a very useful method for large-scale [3] map reconstruction.

2.4 The semi-dense mapping module Suppose a sequence of images in a video, and the system got the trajectory of each frame. The procedure takes the first image frame as the reference one, and then the depth information of each pixel in this reference frame will be calculated. Next, the function of completing the process of feature-based and semi-dense mapping in this paper need to be explained:

First, extract the feature points of every image frame and calculate the matching rate between corresponding feature points according to ORB descriptors. Based on the feature points method, the system can track many objects in the 3D space and acquire their poses in different images.

Then, the VSLAM system cannot determine the pose of feature points with only one image frame. The VSLAM system needs to use the transformation relationships with neighboring keyframes to estimate the points' depth information, and the triangulation calculation method is adopted here.

During the semi-dense depth information estimation process, the system cannot compute all the descriptors of every pixel as feature points, which could gain much computation. So the feature points matching becomes an essential question in the semi-dense [8] depth estimation. So this work used polar line searching and block matching techniques inspired by the direct VSLAM method. Once the system acquired the position of feature points in each image, triangulation calculation could determine their depth information, just like the feature-based SLAM systems. However, the difference is that we will use triangulation many times to make the depth estimation converge. We hope the depth estimation can gradually be refined from an uncertain number to a stable number as the measurement times increase. The Gaussian depth filter is used there.

3. The main algorithm of the system

3.1 Polar line searching and block matching As is shown in Fig.2, the system can shot the same point in real 3D space from different view angles. The camera can detect the pixel p_1 from the first shot place. While using a monocular camera, the depth information cannot be measured directly. Based on the fact that depth value is certainly within a fixed region, from zero to infinity. So, the corresponding object points in real 3D space are all distributed on one straight line. As the movement of the robot, the point of intersection between this straight line and the detected image frame could form a polar line on the later image frame. Once the movement trajectory of the robot is obtained by the system, this polar line can also be calculated.

Figure 2: Polar search diagram

In VSLAM systems using feature-based methods, the position of p_2 can be determined through matching the different features of different points. However, ORB descriptors and feature points are not provided here, so the searching of this polar line for points that are similar to p_1 in the second image is the only way in this situation. Specifically, the position of p_2 will be searched from starting place to the ending place along the calculated polar line in the later image frame, pixel by pixel, comparing the similarity of each pixel to p_1 . From this view, this approach is similar to the direct method VSLAM system.

In the systems using the direct methods, it is already known that the effect of comparing the brightness values of one single pixel is not very stable and reliable, because the illumination changing of one single pixel could be very large. So this paper decide to compare the similarities of pixel blocks. First, a block size of $w \times w$ around p_1 point need to be extracted, and then many small pieces of the same size blocks on the polar line also need to be extracted for later similarity comparison, to some extent this method improved the discrimination of matching. This kind of processing is called "block matching".

Let's see this two extreme cases in Fig. 3, pixels' gradient is parallel to polar line direction and vertical to the polar line direction. In the parallel case, we can determine exactly where the highest matching score occurs. Conversely, In the vertical case, even though the blocks have a distinct gradient, when we match the blocks along the polar lines, we find that the match is always the same, which could not get a good match score. In practical situation, the angel of pixel gradient and the polar line is between the above two situations. And the smaller the angle is, the easier the matching will be.

For example, a little pixels' block is taken around p_1 point, and then a bunch of little pixels' blocks on the polar line are also taken. We can call the first selected block around p_1 as $M \in R^{w \times w}$, and call the others as $N_i, i = 1 \dots n$. To calculate the correlation between all these blocks to the

Figure 3: Two extreme cases

first one, we can take the NCC (Normalized Cross Correlation) method: This approach is a bit more complex, calculating the correlation between two small pieces, as is shown in Eq. (1):

$$S(M, N)_{NCC} = \frac{\sum_{i,j} M(i, j)N(i, j)}{\sqrt{\sum_{i,j} M(i, j)^2 \sum_{i,j} N(i, j)^2}} \quad (1)$$

When the correlation value is close to 0, which means the two blocks are not similar. While the correlation value that is close to 1 means having a high level of similarity. There is a contradiction between precision and efficiency in this calculation method. Methods with good accuracy often requires complex calculations, while simple and efficient algorithms usually brings inaccurate results. This requires a good balance in practical engineering.

3.2 Gaussian depth filter Two solutions could be adopted here, the optimization of filters and nonlinear performances. Although the nonlinear optimization effecting is better, considering that the front-end visual odometry [8] has occupied a lot of computational load in the whole system, and a strong real-time [5] requirement, the filtering method with less computational load [10] is more suitable for depth estimation, so Gaussian depth filter is used here. Set the depth value of a pixel to obey this formula, as is shown in Eq. (2):

$$P(d) = N(\mu, \sigma^2) \quad (2)$$

Every time a new data comes in, the depth information will be firstly calculated by the system. Then, the system need to suppose that a set of estimated depth values also subjects to Gauss Distribution, as is shown in Eq. (3):

$$P(d_{obs}) = N(\mu_{obs}, \sigma_{obs}^2) \quad (3)$$

Next, the new estimated depth value need to be fused with the old depth values set, to update the original distribution of d . According to Gauss formula, it could be easily understood that the product of two Gauss Distributions is still a Gauss Distribution. Assuming that the fused d obey the distribution of $N(\mu_{fuse}, \sigma_{fuse}^2)$, then according to the theory of

Figure 4: Uncertainty analysis

Gauss Distributions' production, these formulas can be got as Eq. (4) - Eq. (5):

$$\mu_{fuse} = \frac{\sigma_{obs}^2 \mu + \sigma^2 \mu_{obs}}{\sigma^2 + \sigma_{obs}^2} \quad (4)$$

$$\sigma_{fuse}^2 = \frac{\sigma^2 \sigma_{obs}^2}{\sigma^2 + \sigma_{obs}^2} \quad (5)$$

3.3 Uncertainty analysis Considering the polar line searching, we found the corresponding point p_2 of p_1 , and estimated the depth value of p_1 , which is shown in Fig. 4. We think that the spatial point corresponding to p_1 is P . Therefore, we can call O_1P as p , O_1O_2 as the baseline t , and O_2P as a . And The bottom two angles of this triangle can be called as α and β . Now, considering that there is a pixel size error on the polar line l_2 that makes the β angle becomes β' , and p becomes p' , and the new angle γ occurs. So the error of this single pixel caused from p' to p need to figure out.

This is a typical geometry problem. And the geometry structure of these variates is shown in Eq. (6) - Eq. (8):

$$a = p - t \quad (6)$$

$$\alpha = \arccos \langle p, t \rangle \quad (7)$$

$$\beta = \arccos \langle a, -t \rangle \quad (8)$$

The disturbance of p_2 by one pixel point will cause a change of β , and the camera focal length is f as is shown in Eq. (9), so the formulas can be got as Eq. (10) - Eq. (12):

$$\delta\beta = \arctan \frac{1}{f} \quad (9)$$

$$\beta' = \beta + \delta\beta \quad (10)$$

$$\gamma = \pi - \alpha - \beta' \quad (11)$$

$$\|p'\| = \|t\| \frac{\sin \beta'}{\sin \gamma} \quad (12)$$

In practical engineering, when the uncertainty value is less than a certain threshold, the depth value can be seen as converged. The process for estimating spatial points depth values can be divided into 4 steps.

Figure 5: Sweeping robot equipment

- (a) First, that the set of estimated depth values should be seen as a Gauss Distribution is the premise.
- (b) When a new data is generated, the feature points will be matched through polar line searching and block matching.
- (c) The depth values and uncertainty errors will be calculated by this system.
- (d) Fuse and renew the latest estimated depth values into the whole depth value set. Stop calculation if the depth value is converged, otherwise return to step (b).

4. Simulation

The proposed SLAM system is equipped in a sweeping robot, as shown in Fig. 5. This sweeping robot equipment has 3 layers, the first layer is equipped a monocular camera, the second layer is a control platform, and the third layer can be used to adjust the height of the above camera. The sweeping robot is equipped with a Nvidia's TEGRA X1 processing chip and Kinetic ROS (robot operating system) software.

During the experiment, the ORB-SLAM2 system will be compared with the proposed semi-dense SLAM system based on the feature-based approach. In order to ensure the rigor and fairness of the experiment, TUM (Technical University of Munich) dataset would be used to test these two systems. The TUM benchmark is a powerful tool for evaluating vision odometer and vision SLAM systems. They also provided data sets containing color images, depth images and ground truth trajectories, which are very complex and varied, and could be used to simulate real environmental conditions. So this experiment used the TUM benchmark to compare the maps reconstructed by ORB-SLAM2 system and the proposed SLAM system, as is shown in Fig. 6 and Fig. 7. The semi-dense SLAM system is designed to produce more dense maps that the human eyes can clearly identify. The traditional ORB-SLAM2 system reconstructed

Figure 6: Construction of ORB-SLAM2 system

Figure 7: Construction of proposed system

very sparse maps, and all the sparse points representing different objects are crowding overlapping with each other. So the human eyes can hardly distinguish the appearance of the objects in maps. From the perception ability of human eyes, the proposed semi-dense SLAM system could reconstruct much better point cloud maps than ORB-SLAM2.

Positioning accuracy is very important in the application of 3D point cloud maps reconstruction with monocular camera. The ATE (absolute trajectory error) evaluation indicator could be used to calculate the mean positioning error in the point cloud maps reconstructed by the above two SLAM systems. Table. 1 presents the ATE results of the proposed SLAM system and ORB-SLAM2 system in different datasets, and compared the estimated keyframe trajectory with the ground truth trajectory. As we can see from this table, the ATE results of the SLAM system proposed are slightly better than those of ORB-SLAM2 in static scenarios. Through average calculation of the optimization percentage, the keyframe pose trajectory accuracy of the semi-dense SLAM system is 9.13% higher than that of the traditional ORB-SLAM2 system. But in the datasets of dynamic scenes [4], the accuracy of robot self-positioning trajectory decreased on the contrary.

5. Conclusion

The ORB-SLAM2 system uses a monocular camera to reconstruct sparse point cloud maps, and the robot synchronization tracking trajectory accuracy is not high enough. Human eyes cannot recognize over-sparsed point cloud maps, nor can mobile robots realize navigation, obstacle avoidance, path planning, and other functions in the future. This paper optimized the sparse point cloud maps based on the ORB-SLAM2 system, combined with geometric triangulation, polar line searching, and block match-

Table 1: Absolute trajectory error

Dataset	Proposed work	ORB-SLAM2	Optimize
f1desk	1.37	1.39	1.40%
f2desk	0.86	0.93	7.50%
f1xyz	0.91	1.00	9.10%
f2xyz	0.31	0.33	6.10%
f1floor	2.89	4.17	30.60%
f3str-tex-far	0.92	0.96	4.10%
f3longoffice	3.18	3.89	18.25%
f2deskwithper	0.92	1.42	35.21%
f3walkxyz	1.38	1.30	-6.15%
f3sittingxyz	0.93	0.81	-14.80%

ing methods, measured the pixel points' depth information in high gradient areas, and improved the accuracy of the synchronous self-positioning and tracking trajectory. The improved SLAM system was able to reconstruct semi-dense point cloud maps, and the accuracy of synchronization tracking trajectory was improved by 9.13% compared with ORB-SLAM2. The proposed semi-dense SLAM system can help mobile robots complete more complex works like semantic segmentation, target detection, and 3D environment reconstruction.

References

- [1] J. K. Makhubela, T. Zuva and O. Y. Agunbiade, "A Review on Vision Simultaneous Localization and Mapping (VSLAM)", *2018 International Conference on Intelligent and Innovative Computing Applications (ICONIC)*, pp.1-5, 2018. DOI: 10.1109/ICONIC.2018.8601227
- [2] B. D. Gouveia, D. Portugal and L. Marques, "Speeding up Rao-Blackwellized Particle Filter SLAM with a Multithreaded Architecture", *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp.1583-1588, 2014. DOI: 10.1109/IROS.2014.6942766
- [3] J. Engel, T. Schöps and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM", *European conference on computer vision*, pp.834-849, 2014. DOI: 10.1007/978-3-319-10605-2
- [4] B. Bescos, J. M. Fácil and J. Civera, "DynaSLAM: Tracking, Mapping and Inpainting in Dynamic Scenes", *IEEE Robotics and Automation Letters*. Vol.3, 2018. DOI: 10.1109/LRA.2018.2860039
- [5] J. Tang, L. Ericson and J. Folkesson, "GCNv2: Efficient Correspondence Prediction for Real-Time SLAM", *Computer Vision and Pattern Recognition*, 2019. DOI: 10.1109/LRA.2019.2927954
- [6] R. Mur-Artal and J. D. Tardas, "ORB-SLAM2: An Open-Source SLAM System for Monocular Stereo and RGB-D Cameras", *IEEE Transactions on Robotics*, Vol.33, no.5, pp.1255-1262, 2016. DOI: 10.1109/TRO.2017.2705103
- [7] A. Geiger, J. Ziegler and C. Stiller, "Stereoscan: Dense 3D Reconstruction in Real-Time", *Intelligent*

Vehicles Symposium (IV), pp.963-968, 2014. DOI: 10.1109/IVS.2011.5940405

- [8] J. Engel, J. Sturm and D. Cremers, "Semi-Dense Visual Odometry for a Monocular Camera", *International Conference on Computer Vision (ICCV)*, 2013. DOI: 10.1109/ICCV.2013.183
- [9] E. Rublee, V. Rabaud and K. Konolige, "ORB: An Efficient Alternative to SIFT or SURF", *IEEE Intl. Conf. on Computer Vision (ICCV)*, Vol.13, 2011. DOI: 10.1109/ICCV.2011.6126544
- [10] A. Hornung, K. M. Wurm and M. Bennewitz, "OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees", *Autonomous Robots*, Vol.34, no.3, pp.189-206, 2013. DOI: 10.1007/s10514-012-9321-0

Z. Li (Member) received the B.S. degree in Communication Engineering from Yangzhou University in 2017. He received M.S. degree in Electrical and Electronic Engineering from Yangzhou University in China, and from Kyushu Institute of Technology in Japan in 2020, respectively. He is now a Ph.D candidate in Engineering at Kyushu Institute of Technology in Japan.

His current research interests include machine learning and VSLAM.

Y. Du (Non-member) received the B.S. degree in Optoelectronics from Southeast University in 1983. He received M.S. degree in Electronic and Information Engineering from Zhejiang University in 2004. From 2001 to 2002, he engaged in visual navigation research as a visiting scholar at Intelligent Robot Research Institute of Universität der Bundeswehr München. Since 2001, he

has been an associate professor at Yangzhou University. His current research interests include computer vision and electronic measuring system.

M. Zhu (Member) was born in Shannxi, China, 1981. She received the M.S. degree in engineering from Xi'an University of Architecture and Technology, China, in 2008. Currently, she is a Ph.D. student in Electrical and Electronic Engineering at Kyushu Institute of Technology, Japan. Her research interests include Image Processing, Image Enhancement and Wireless sensor network.

sensor network.

S. Zhou (Member) was born in Hunan, China, 1994. She received the B.S. and M.S. degrees in Mechanical Engineering and Automation from Northeastern University, China, in 2017 and 2020. She is currently a Ph.D student in Electrical and Electronic Engineering at Kyushu Institute of Technology, Japan. her research interests include 3D vision, SLAM, image processing and artificial intelligence.

artificial intelligence.

S.Serikawa (Member) was born in Kumamoto Japan on June, 1961. He received a B.S.degree and M.S. degree in electrical engineering from Kumamoto University in 1984 and 1986. He received a Ph.D. degree in electronic engineering from Kyushu Institute of Technology in 1994. Currently, he is a professor of Dept. of electrical engineering and electronics in Kyushu Institute of Technology, Japan. His current research interests include computer vision, sensors and robotics. He is a member of IEICE, IEEJ, IPSJ, IIAE.

L.Zhang (Member) received the B.S. degree in Electronic Engineering from Southeast University in 1994. He received M.S. and Ph.D. degrees in Electrical Engineering from Kyushu Institute of Technology, in 1999 and 2001, respectively. Now, he has been a professor at Kyushu Institute of Technology. His current research interests include computer vision, image processing, biometrics authentication and remote sensing system. He is a member of IIAE, IEEE, and IEICE.