Proceedings of 2021 IEEE
International Conference on Mechatronics and Automation
August 8 - 11, Takamatsu, Japan

Study on Positioning for The Spherical Amphibious Robot Based on Visual-Inertia Localization

Jian Guo[1] , Xiangyu Chen[1]
[1]Tianjin Key Laboratory for Control Theory&Applications
In Complicated systems and Intelligent Robot Laboratory
Tianjin University of Technology
Binshui Xidao Extension 391,Tianjin,300384,China
jianguo@tjut.edu.cn;
794415466@qq.com

Shuxiang Guo[1,2*]
[2] Department of Intelligent Mechanical Systems Engineering,
Faculty of Engineering
Kagawa University
Takamatsu,Kagawa,Japan
*corresponding author :
guo@eng.kagawa-u.ac.jp

and Jigang Xu[3*]
[3]Unit68709
Qinghai Haidong,810700,China
*corresponding author :
xujigang216@163.com

***Abstract* –Simple visual odometer such as ORB_SLAM2 algorithm has its limitations, in a short time to move fast, the image is blocked, the image features are sparse can not work stability. The inertial measurement unit (IMU) could calculate the trajectory well in a short time, but it was easy to accumulate too much error in a long time. So we used the visual positioning information to estimate the zero deviation of the IMU, reduce the divergence and accumulation error caused by the zero deviation of the IMU. And the IMU can also provide the positioning for the vision when moving fast. In this paper, two localization algorithms are fused by untraceable Kalman filter to form a more robust localization algorithm. According to the motion model of IMU, the pose was obtained by integrating it. In the visual part, fast and effective ORB feature points were selected for feature tracking and the camera pose was solved. Finally, the visual inertial odometer solution propose in this paper was experimented and verified. The comparative experiments are carried out on the TUM data set first, then on the mobile robot. The hardware platform of the experiments was a mobile robot equipped with a camera and IMU, and the experimental environment is a general indoor environment. The effectiveness and reliability of the visual inertial odometer designed in this paper on mobile robot are verified through experiments. Compared with the pure visual odometer, the scheme is stable, faster and practical.**

Index Terms – ORB SLAM2, Inertial Measurement Unit, Untraced Kalman filter , The spherical underwater robot

I. INTRODUCTION

Visual SLAM was implemented mainly with the help of filters when it was first developed in the early years.Subsequently, the nonlinear optimization of visual SLAM system based on the minimization of key frame and cost function is gradually developed. Parallel Tracking and Mapping(PATM) proposed by Klein et al. and open source is the first nonlinear monocular SLAM system.Moreover, the key frame mechanism is introduced so that the VO system does not have to deal with every frame of image, which greatly strengthens the real-time performance of VO. In addition, PTAM realizes the parallelization of tracking and mapping, and divides the tracking part and mapping part into two parts, which contributes the concept of front and rear end for modern visual SLAM. Since PTAM has been used for some years, its defects are more obvious now, such as small application scenarios, easy tracking loss, and no loop detection to eliminate cumulative errors. [1]-[7]

In 2015,Mur-Artal et al. proposed and open-source ORB-SLAM, the benchmark work of modern visual SLAM system.ORB-SLAM has made many optimizations and improvements on the basis of PTAM.The main features are as follows: ORB feature points with vision and rotation invariance are selected as feature points;The ORB-SLM has a closed-loop detection mechanism to detect previously traveled areas and eliminate error accumulation;ORB-SLAM automatically selects two frames to complete the monocular initialization by searching the maximum apparent parallax.Loop detection is used to eliminate cumulative errors.

In visual SLAM, their pose estimation depends on the feature points in the environment. When the camera moves quickly or the Angle of view changes too much, positioning failure will occur.This is a fatal shortcoming in commercial applications. In addition, monocular SLAM cannot determine the scale of the trajectory, which is also a source of error. [8]-[11]

In our real life, robots often do not carry only one sensor. For example, the driverless car, which is very popular recently, also uses the scheme of multi-sensor fusion.This is because in a complex environment, a single sensor is difficult to achieve a stable and ideal positioning effect. After the IMU is fused with the camera, a scheme combining the advantages of the two is expected to be obtained. Therefore, the navigation and positioning method integrating vision and inertia has become one of the current research hotspots in this context. However, the calculation of VIO is very complicated, mainly because the IMU measures the acceleration and angular velocity, so we have to build the motion model of the IMU. At present, VIO can be divided into two fusion methods of loose coupling and tight coupling, and back-end processing methods can be divided into two methods of filtering and optimization. But in theory, more accurate results should be obtained by adopting tight coupling and optimization.

However, in VIO, due to the high frequency of IMU data and the image data, the system has a large amount of computation. Therefore, the effect of the optimization based fusion algorithm is not significantly better than that of the filtering algorithm in the case of limited computing capacity.

Mourikis et al. proposed a tightly coupled filtering based method, MSCKF, in 2017.MSCKF solves the traditional method based on extend kalman filter VIO dimension problems too much, when SLAM MSCKF not add the feature points to the state vector, but the camera pose of different time to join the state vector, the feature points would be more than one camera to see, thus in geometric constraints between the state of multiple cameras, after using geometric constraint observation model was constructed to update EKF.

Stefan Leutenegger proposed a visual inertial fusion method, OKVIS, in 2015.Unlike MSCFK, which uses Kalman filter optimization, OKVIS uses keyframes and nonlinear optimization to estimate the pose. In addition, OKVIS combines the camera reprojection error and the IMU integration error into a back-end optimization function, integrating the IMU information and visual information into one optimization problem. But the shortcoming is that this method does not add loop detection to achieve global optimization, so positioning errors will inevitably accumulate over time.

Called VINS - Mono is the Hong Kong university of science and technology in 2017, Dr Qin Tong a VIO algorithm is put forward and open source, its front end USES GFTT feature points extraction combined with multilayer KLT light flow tracking, after to get a better effect of feature matching, using loosely coupled initialized data visual inertia, the backend is adopted key frames and the optimization of sliding window method, and use the Ceres library of nonlinear optimization in searching the posture, update another loopback detection based on DBoW2 can eliminate the accumulated error, can have better global consistency.

In recent years, MSF and SSF developed by ETH Zurich in Switzerland have adopted the loose coupling scheme. SSF, for example, SSF first on camera image processing, complete position, after the camera visual part is used to estimate the results with the measured value of the IMU joint construction of state variables, using EKF to forecast the quantity of state and after update, the whole process of filtering and fusion visual observation as auxiliary to correction of IMU integral value, the multi-sensor fusion algorithm based on EKF is relatively small amount of calculation, but is not accurate, because in nonlinear system is more serious, because EFK is depend on the state of linearization to spread so mean and covariance of EKF estimation are not accurate. In this paper, an improved ORB-SLAM2 algorithm based on untracked Kalman filter is proposed to achieve a more accurate positioning effect on robots with limited computing power.

II. THE SPHERICAL UNDERWATER ROBOT

With the expansion of human exploration, more and more researchers devote themselves to the development and application of mobile robots.Mobile robot has high application value in narrow and strange environment.The spherical amphibious robot is a kind of robot that can move on land and underwater. It is equipped with sensors such as camera and IMU unit so that it can move and collect image data in complex environment.

The spherical amphibious robot is shown in Figure 1.The robot has legs, and two servo motors on each leg are used to control the movement of the legs. When the controller controls the four legs to work together, the robot can move on the land. The gait of the robot is shown in Figure 2.The tiptoe of each leg is equipped with a water jet motor. When the robot is underwater, the direction of the water jet motor can be adjusted by the motor to realize the underwater movement of the robot.

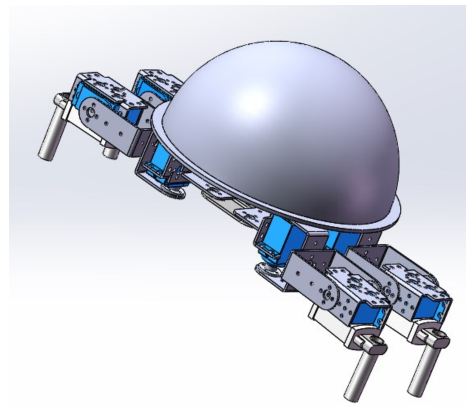

Fig.1 The spherical underwater robot's structure

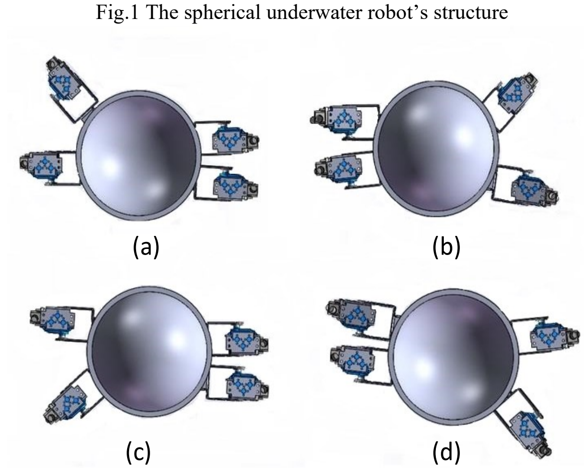

Fig. 2 Robot's gait

III. VISUAL-INERTIAL LOCALIZATION RESEARCH

A. *ORB-SLAM2 Visual Odometer Model*

ORB-SLAM2 algorithm is implemented by three parallel threads: tracing, local mapping and loop detection.In the tracking thread, image feature points are collected and pose information is estimated.After receiving the image information of the camera, the tracking thread will perform the following steps on each frame of the camera image: [12]-[19]

1). Mage Preprocessing

This step will carry out gray conversion, ORB feature extraction, image edge calculation and other operations on the collected image information.The algorithm extracts FAST corner points from an 8-layer image pyramid.In order to ensure uniform distribution of feature points, the ORB-SLAM2 algorithm divides each layer of images into grids and extracts at least 5 corner points from each grid.Then detect each grid corner points, if the number of corners is not enough, adjust the threshold.If no corner points can be detected in some cells, the number of corner points extracted will be reduced accordingly.Finally, the direction and ORB feature descriptor are calculated according to the reserved corners of FAST.The ORB feature descriptor will be used for all subsequent feature matching of the algorithm.

2). Estimate The Initial Pose

If a frame image tracking success, the algorithm with the movement rate constant model to predict the current position of the camera (i.e., think the camera is in constant motion), and then search a frame of image feature points on the map the corresponding matching point cloud point and the current frame image, finally using search to match point to solve the position by the PnP.However, if not enough matching points are found to solve the pose, the algorithm will enlarge the search scope, search whether the points near the map cloud point have matching points in the current frame image, and then optimize the camera pose at the current moment by finding the corresponding matching point pair.If the feature points can not be tracked after the search scope is expanded (then the motion model is invalid), then the word bag (BOW) vector of the current frame image is calculated, and several key frames are selected as alternative matching frames by using the BOW dictionary.Then, the ORB features corresponding to the map cloud points are calculated in each alternative key frame.Then, PNP algorithm is executed for each alternative key frame in turn to calculate the pose of the current frame.If we find a pose that covers enough valid points, we search for more matching cloud points corresponding to that keyframe.Finally, the location of the camera is further optimized based on all the matching points found. If there are enough valid data, the tracking program will continue to execute. The track thread flow for ORB-SLAM2 is shown in Figure 3.

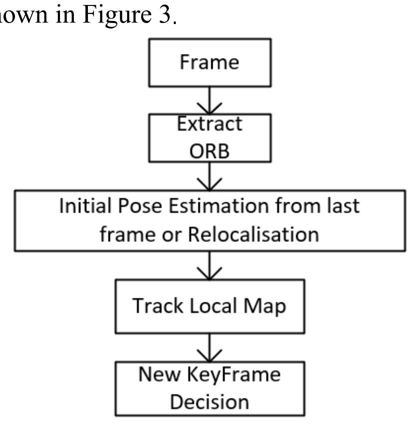

Fig.3 Track thread flow

After the above steps, the front end of ORB-SLAM2 algorithm can calculate the current robot pose.However, if the camera moves quickly or the Angle of view changes too much, the positioning failure may occur.

B. Establish The IMU Model

1). The Coordinate System

In the Inertial navigation of robot, the Earth-Centered Inertial (ECI) Frame is generally used as the reference coordinate system.The center of the earth is taken as the origin point, the northward axis is the Z axis, the X-Y plane is the equatorial plane, and the X axis points to the Vernal Equinox point (i.e. the intersection point of the centre-earth line and the equator in the annual spring equinox).The coordinate system is shown in Figure 4.

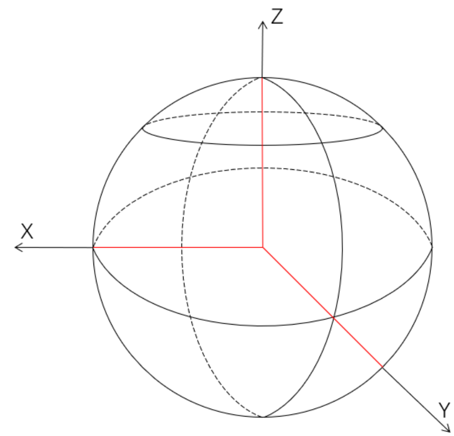

Fig.4 The ECI coordinate system

2). The IMU Model

The general IMU is a six-axis sensor, including three-axis acceleration and three-axis angular acceleration.If the influence of scale factor is ignored and only white noise and deviation random walk are considered, then the angular velocity and acceleration obtained by IMU are:

$$\tilde{\omega}^b = \omega^b + b^g + n^g \tag{1}$$

$$\tilde{a}^b = q_{bw}(a^w + g^w) + b^a + n^a \tag{2}$$

The superscript g represents the gyroscope, a represents the accelerometer, w represents the world coordinate system, and b represents the IMU body coordinate system.The real value of the IMU is ω, a , and the measured value is $\tilde{\omega}, \tilde{a}$.

The derivative of position is the velocity, the derivative of velocity is the acceleration $a$measured by IMU, and the angular velocity of rotation is the angular velocity ω measured by IMU. Therefore, the motion model of IMU can be expressed as

$$\begin{aligned} \dot{p}_{wb_t} &= v_t^w \\ \dot{v}_t^w &= a_t^w \\ \dot{q}_{wb_t} &= \frac{1}{2}\Omega(\omega) q_{wb_t} \end{aligned} \tag{3}$$

Where

$$\Omega(\omega)=\begin{bmatrix}\omega^{\wedge} & \omega \\ -\omega^{T} & 0\end{bmatrix} \tag{4}$$

According to the above derivative relation, the motion model of IMU in continuous time can be deduced.Suppose from the Pose, Velocity and Quaternion at any time of I, by integrating the measured value of IMU, the PVQ formula at the time of j can be obtained:

$$\begin{aligned}p_{wb_j} &= p_{wb_i}+v_i^w\Delta t+\iint_{t\in[i,j]}(q_{wb_t}a^{b_t}-g^w)\delta t^2\\ v_j^w &= v_i^w+\int_{t\in[i,j]}(q_{wb_t}a^{b_t}0-g^w)\delta t\\ q_{wb_t} &= \int_{t\in[i.j]}\frac{1}{2}\Omega(\omega)q_{wb_t}\delta t\end{aligned} \tag{5}$$

Since our measurement data are discrete, Euler's method is used to discretization Formula (5). The positions from two adjacent moments k to k+1 are calculated with the measured value ω, a at the k moment.The kinematics formula of IMU in the discrete state is obtained as follows:

$$\begin{aligned}p_{wb_{k+1}} &= p_{wb_i}+v_k^w\Delta t+\frac{1}{2}a\Delta t^2\\ v_{k+1}^w &= v_k^w+a\Delta t\\ q_{wb_{k+1}} &= \frac{1}{2}\Omega(\omega)q_{wb_t}\Delta t\end{aligned} \tag{6}$$

C. Kalman filter setup and sensor fusion

Both SLAM algorithm and inertial navigation are nonlinear systems.State estimation is a difficult problem in nonlinear systems.Kalman Filter (KF) is only applicable to linear systems.The Extended Kalman Filter (EKF) linearizes nonlinear systems using Taylor expansion.However, the error of EKF in a strongly nonlinear system is very large.At the same time, it is necessary to calculate the first-order partial derivatives, that is, Jacobian matrix, when updating the system, which undoubtedly increases the amount of calculation and aggravates the burden of the processor.In this paper, a novel filtering algorithm Unscented Kalman Filter (UKF) is used to replace the traditional EKF. Its calculation accuracy is higher than EKF and the calculation of Jacobian matrix is omitted. UKF USES the statistical linearization technique, we called this linearization method of nondestructive transformation (unscented transformation) this technique mainly through n acquisition in the prior distribution of points (we call them the sigma points) of linear regression to linearize nonlinear function of random variables, since we consider is an extension of the random variable, so the linearization than Taylor series linearization strategy used by (EKF) is more accurate.Like EKF, UKF is mainly divided into forecast and update.

Let the system state matrix be：

$$x=\begin{bmatrix}p\\ v\\ q\end{bmatrix} \tag{7}$$

Assume that the uncertainty of the system is

$$P=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix} \tag{8}$$

The algorithm flow of untraced Kalman filter is as follows:

Step 1: Initialize system state x_k, P_k

The first sensor data received after system startup is used to initialize the system.

Step 2: Based on state x_k, P_k generates Sigma point X_k

Through the states x_k, P_k, we find a set of certain Sigma points, and the true mean and covariance of the position and pose can be accurately estimated by means of the mean and covariance of the Sigma points after transformation.

Step 3: According to the model to predict the future point of Sigma $X_{k+1|k}$

The Sigma point obtained at time k was substituted into the model to calculate the Sigma point at time k+1

Step 4: $x_{k+1|k}, P_{k+1|k}$ is predicted according to the state generated by the predicted Sigma point $X_{k+1|k}$

Step 5: The predicted Sigma point $X_{k+1|k}$ is converted to the predicted measurement $Z_{k+1|k}$ B when the measurement arrives.

Step 6: According to the predicted measurement value $Z_{k|k+1}$ and difference of the real measured value update system state $x_{k+1|k+1}, P_{k+1|k+1}$

IV. EXPERIMENTAL TEST AND RESULT ANALYSIS

A. Run TUM Dataset to Get Data

We first validate the reliability of the algorithm by using a data set on a computer to simulate reality. We use the technical university of Munich (TUM) data set, which provides a series of image streams and imu data for each frame. This experiment is based on the xyz data set in tum, which is collected by the camera in the context of a desk environment. By running the original algorithm and the improved algorithm, the data is compared. The original algorithm is compared to the real trajectory below. You can see that the trace of the original algorithm is moving, and there are some errors that make the trajectory inaccurate. Fig. 5 is the comparison between ORB-SLAM2 algorithm and the actual trajectory.

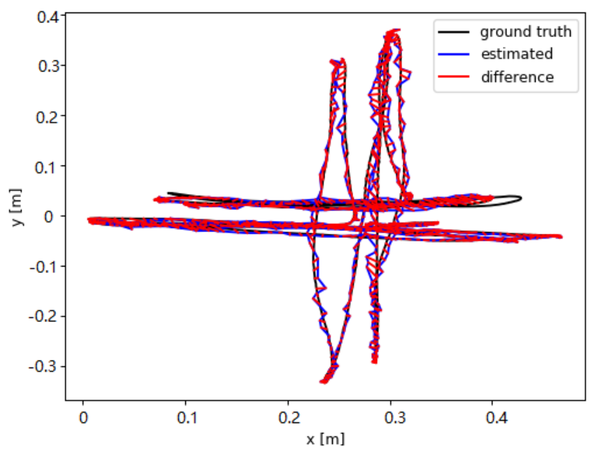

Fig.5 ORB-SLAM2 algorithm compared with the actual trajectory

The comparison between the improved algorithm and the real trajectory is shown in the following figure.It can be seen that compared with the original algorithm, the trajectory tracking of the improved algorithm is closer to the real trajectory, and the error is also less than the original algorithm. Fig. 6 is a comparison between the improved algorithm and the actual trajectory

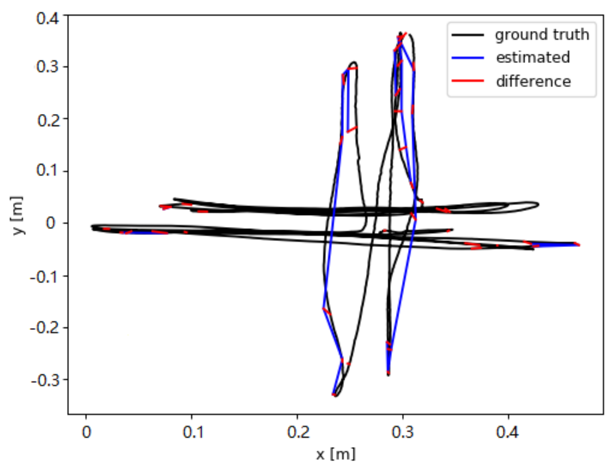

Fig.6 Comparison between the improved algorithm and the actual trajectory

By comparing the error between the estimated trajectory and the real trajectory in the two simulations, we can see the data as shown in the TABLE I below

TABLE I
Absolute error data comparison

	REMS	Means	Maximum	Minimum
Before improvement	1.200	0.969	2.880	0.0771
After improvement	1.017	0.912	1.837	0.0229

It can be seen that the improved algorithm is better than the original algorithm, and the effectiveness of the improved algorithm is verified in the simulation experiment.

B. Experiments with The Real Environment

We tested the algorithm in a real environment.The spherical amphibious robot carries an RGB-D camera for image acquisition. The size of the camera is 165x40x30mm and the maximum power is 2.5W, which is suitable for use in the robot. At the same time, the camera can capture 640x480 images at a speed of 30 frames per second.After pictures are collected by the camera, they are processed by the Raspberry Pi 4B on the robot. The Raspberry Pi 4B used in this experiment has a dominant frequency of 1.5GHz and a memory of 4GB.We will control the robot to circle around the site, and the estimated trajectory diagram obtained by using the improved algorithm is shown in the figure below.

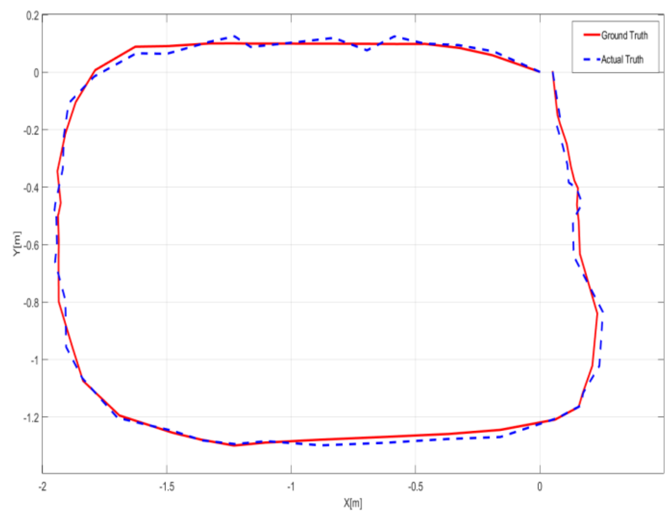

Fig.7 The trajectory obtained by the robot running ORB-SLAM2 algorithm

Then we use the improved algorithm to introduce the IMU information. The estimated trajectory obtained this time is shown in the figure below.

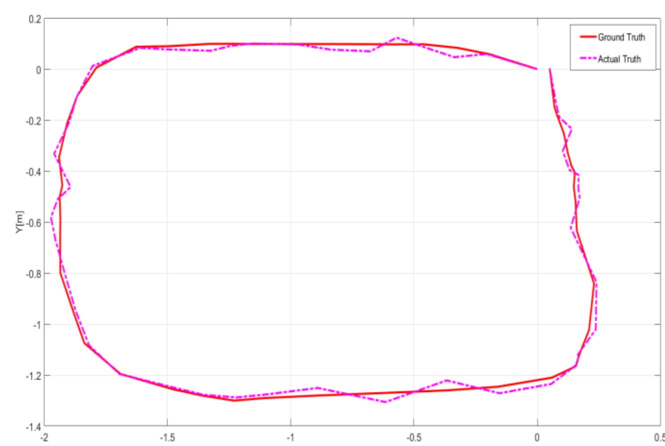

Fig.8 The trajectory of the robot running the improved algorithm

By comparing the errors of each algorithm in the two experiments with the real trajectory, the data are shown in the following TABLE II. It can be seen that the improved algorithm is more accurate.

TABLE II
Absolute error data comparison

	REMS	Means	Maximum	Minimum
Before improvement	1.523	1.153	2.467	0.0842
After improvement	1.012	1.095	1.944	0.0369

V. CONCLUSIONS AND FUTURE WORK

In this paper, the positioning accuracy of the robot was improved by combining visual information and inertia information through untraced Kalman filter. Traditional visual positioning could not be used in places with little texture. It was easy to be lost in high-speed motion, while inertial navigation performs well in high-speed motion, but has zero drift error when it is stationary. This paper fuse visual navigation and inertial navigation by untraceable Kalman filter algorithm to improve the positioning accuracy. Compared with the traditional Extended Kalman Filter (EKF), the untraceable Kalman Filter (UNKF) had better linearization of the nonlinear system, and reduced the computation amount because it did not need to calculate the Jacobian matrix, so it could run on the robot with little computation amount. In the future, we will build and optimize the visual-inertia model to improve the positioning accuracy.

ACKNOWLEDGMENTS

This research is supported by National Natural Science Foundation of China (61703305), Key Research Program of the Natural Science Foundation of Tianjin (18JCZDJC38500) and Innovative Cooperation Project of Tianjin Scientific and Technological (18PTZWHZ00090).

REFERENCES

[1] W. Lee, K. Eckenhoff, P. Geneva and G. Huang, "Intermittent GPS-aided VIO: Online initialization and calibration", Proc. 2020 IEEE International Conference on Robotics and Automation (ICRA), pp.5724-5731, 2020.
[2] P. Xie, W. Su, B. Li, R. Jian, R. Huang, S. Zhang and J. Wei, "Modified keyframe selection algorithm and map visualization based on ORB-SLAM2", Proc. 2020 4th International Conference on Robotics and Automation Sciences (ICRAS), pp.142-147, 2020.
[3] H. Xu, C. Yang and Z. Li, "OD-SLAM: Real-Time localization and mapping in dynamic environment through Multi-Sensor fusion", Proc. 2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM), pp.172-177, 2020.
[4] M. Zhang, S. Han, S. Wang, X. Liu, M. Hu and J. Zhao, "Stereo visual inertial mapping algorithm for autonomous mobile robot", Proc. 2020 3rd International Conference on Intelligent Robotic and Control Engineering (IRCE), pp.97-104, 2020.
[5] C. Lee, J. Peng and Z. Xiong, "Asynchronous fusion of visual and wheel odometer for SLAM applications", Proc. 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), pp.1990-1995, 2020.
[6] P. Li, B. Leng and H. Fu, "Autonomous positioning of omnidirectional mobile robot based on visual inertial navigation", Proc. 2020 39th Chinese Control Conference (CCC), pp.3753-3758, 2020.
[7] O. C. B. Silveira, J. G. O. C. de Melo, L. A. S. Moreira, J. B. N. G. Pinto, L. R. L. Rodrigues and P. F. F. Rosa, "Evaluating a visual simultaneous localization and mapping solution on embedded platforms", Proc. 2020 IEEE 29th International Symposium on Industrial Electronics (ISIE), pp.530-535, 2020.
[8] J. Henawy, Z. Li, W. Y. Yau and G. Seet, "Accurate IMU factor using switched linear systems for VIO", IEEE T IND ELECTRON, pp.1, 2020.
[9] T. Sun, Y. Liu, Y. Wang and Z. Xiao, "An improved monocular Visual-Inertial navigation system", IEEE SENS J, pp.1, 2020.
[10] W. Huang, H. Liu and W. Wan, "An online initialization and Self-Calibration method for stereo Visual-Inertial odometry", IEEE T ROBOT, vol.36; 36, no.4; 4, pp.1153-1170, 2020.
[11] L. Jin, H. Zhang and C. Ye, "Camera intrinsic parameters estimation by visual－inertial odometry for a mobile phone with application to assisted navigation", IEEE/ASME Transactions on Mechatronics, vol.25; 25, no.4; 4, pp.1803-1811, 2020.
[12] C. Chu and S. Yang, "Keyframe-Based RGB-D Visual-Inertial odometry and camera extrinsic calibration using extended kalman filter", IEEE SENS J, vol.20; 20, no.11; 11, pp.6130-6138, 2020.
[13] C. Li, L. Yu and S. Fei, "Large-Scale, Real-Time 3D scene reconstruction using visual and IMU sensors", IEEE SENS J, vol.20; 20, no.10; 10, pp.5597-5605, 2020.
[14] J. H. Jung, S. Heo and C. G. Park, "Observability analysis of IMU intrinsic parameters in stereo visual－inertial odometry", IEEE Transactions on Instrumentation and Measurement , vol.69; 69, no.10; 10, pp.7530-7541, 2020.
[15] X. Liu, L. Chen, X. Wang, H. Kuang and X. Ma, "An improved ORB-SLAM algorithm for mobile robots", Proc. 2019 Chinese Control And Decision Conference (CCDC), pp.5585-5590, 2019.
[16] J. H. Jung, J. Y. Chung, J. Cha and C. G. Park, "Rapid initialization using relative pose constraints in stereo Visual-Inertial odometry", Proc. 2019 IEEE 15th International Conference on Control and Automation (ICCA), pp.969-974, 2019.
[17] C. L. Xu, D. K. Qu, C. D. Wu, Z. Z. Li, P. Di, J. L. Song and D. J. Tian, "Research on ORB-SLAM Autonomous Navigation Algorithm", Proc. 2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), pp.1182-1186, 2019.
[18] Z. Yu and H. Min, "Visual SLAM algorithm based on ORB features and line features", Proc. 2019 Chinese Automation Congress (CAC), pp.3003-3008, 2019.
[19] X. X., C. W., L. Z. and S. Q., "A double stage EKF-based stereo visual inertial odometry",in 2019 IEEE International Conference on Real-time Computing and Robotics (RCAR),pp.947-952, IEEE, 2019.