2024 IEEE 18th International Conference on Control & Automation (ICCA)
June 18-21, 2024. Reykjavík, Iceland

SLAM in Low-Light Environments Based on Infrared-Visible Light Fusion

Haiwei Wang, Chenqi Gao, Tianyu Gao, Jinwen Hu, Zhao Xu, Junwei Han, Yan Zhu and Yong Wu

Abstract— **Traditional visual Simultaneous Localization and Mapping (SLAM) techniques are difficult to obtain effective information in non-ideal environments such as changing light or full of smoke, which leads to the performance degradation of SLAM algorithms. To overcome the aforementioned challenges, this paper proposes a visual SLAM front-end system based on infrared-visible light fusion. The system achieves precise optimization of camera poses and map point locations in non-ideal environments by jointly optimizing the reprojection errors of visible light image point features and infrared image edge features. In addition, this article further improves the robustness of the algorithm in non-ideal environments through back-end optimization of infrared-visible light and Inertial Measurement Unit (IMU) tight coupling.**

I. Introduction

With the rapid advancement of robotics and artificial intelligence technologies, autonomous driving technology is demonstrating tremendous potential and broad application prospects. Simultaneous Localization And Mapping (SLAM), as a key technology in the field of autonomous driving, plays a crucial role in vehicle positioning and navigation. It has become a research hotspot that is currently receiving joint attention from the academic and industrial communities [1].

Currently, based on the different types of sensors used, SLAM technology can be categorized into two main types: visual SLAM and laser SLAM. However, the high cost and long development cycle of LiDAR, as well as issues such as the lack of semantic information in point cloud maps constructed by LiDAR, make it difficult for LiDAR to be widely adopted in applications. Visual sensors, compared to LiDAR, can perceive a richer set of environmental information and offer advantages such as portability, easy installation, and lower costs. Therefore, visual SLAM is considered an important developmental direction in the current field of SLAM. Currently, most research on visual SLAM is based on visible light cameras, and it can only operate effectively under good ambient lighting conditions. However, in scenes with insufficient lighting, lighting variations, or the presence of smoke, causing a decline in visual quality, visible light cameras can be severely disrupted, leading to a degradation in the system's localization performance or even a failure to function properly. In comparison to visible light cameras, thermal infrared cameras can maintain efficient performance in low light and nighttime conditions. Their imaging principle, based on the temperature of target surfaces, allows them to operate without being constrained by lighting conditions. In scenarios with light absence or smoke diffusion, thermal infrared cameras can still maintain stable imaging quality, demonstrating all-weather capabilities [2]. In summary, in military, security, firefighting, and other fields, infrared SLAM technology provides a crucial solution for positioning in complex environments.In addition, because inertial measurement unit (IMU) are capable of flexibly measuring angular velocity and linear acceleration with changes in motion speed, without being restricted by motion scenes and speed changes [3]. Integrating IMU into visual SLAM systems can fully leverage the respective advantages of cameras and IMU. By fusing visual information and inertial information, robustness in mapping and localization can be enhanced.

To enhance the accuracy and reliability of visual SLAM localization in low-light conditions, this paper proposes a visual-inertial SLAM system that fuses infrared, visible light, and IMU data. In the visual frontend, edge features are extracted from the infrared images, and they are fused with the ORB features [4] from visible light images. Simultaneously, in the backend, the IMU is introduced to form a complementary advantage with the visual system, enhancing the system's robustness. The proposed visual-inertial SLAM system in this paper demonstrates the capability to operate robustly under various lighting conditions, providing high-precision positioning results and achieving all-weather functionality.

The organization of the paper is as follows: Section II introduces a visual SLAM front-end algorithm framework based on infrared-visible light fusion, specifically designed for low-light conditions. Building upon Section II, Section III integrates IMU information and proposes a tightly-coupled back-end optimization algorithm based on infrared-visible light-IMU fusion. In Section IV, an experimental platform is established, and the proposed algorithms are tested and analyzed using publicly available datasets and on-campus measured datasets. Finally, a summary of the entire paper is provided in Section V.

This work is funded by National Natural Science Foundation of China (52372434) and Aeronautical Science Foundation of China (2019ZA053008, 20185553034).

H. Wang is with AVIC XAC Commercial Aircraft CO. LTD. whw01421204@163.com

C. Gao, T. Gao, J. Hu and J. Han are with the School of Automation, Northwestern Polytechnical University, 710129 Xi'an, China. gaochenqi159@mail.nwpu.edu.cn; gao_tianyu98@mail.nwpu.edu.cn; hujinwen@nwpu.edu.cn; jhan@nwpu.edu.cn

Z. Xu, Y. Zhu and Y. Wu are with the School of Electrinics And Information, Northwestern Polytechnical University, 710129 Xi'an, China. zhaoxu@nwpu.edu.cn; zhuyan@nwpu.edu.cn; yongwu@nwpu.edu.cn

Jinwen Hu is the corresponding author.

2024 IEEE 18th International Conference on Control & Automation (ICCA) | 979-8-3503-5440-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCA62789.2024.10591931

II. Visual SLAM Frontend Based on Infrared-Visible Light Fusion

Infrared images rely on the thermal radiation and temperature variations of objects in the environment. Typically, infrared images exhibit low texture, low contrast, and contain significant noise, resulting in a low signal-to-noise ratio. Consequently, traditional feature point detection methods may struggle to detect a sufficient number of feature points in infrared images, leading to failure in visual tracking in the frontend and an inability to accomplish the tasks of localization and mapping. Due to the nature of infrared images capturing the distribution of surface temperatures of objects, significant temperature changes are usually manifested at the edges of objects. This makes edge features more pronounced in infrared images compared to point features. Therefore, in the frontend of infrared visual SLAM, selecting edges as features is considered more reliable.This paper proposes a visual SLAM front-end framework based on infrared-visible light fusion. Edge features are extracted from the infrared images, point features from the visible light images, and these two types of features are fused. This strategy allows the features from both types of images to complement each other, enabling the system to work under different lighting conditions. Consequently, this enhances the system's accuracy and robustness in localization. The specific algorithm framework is illustrated in Fig. 1.

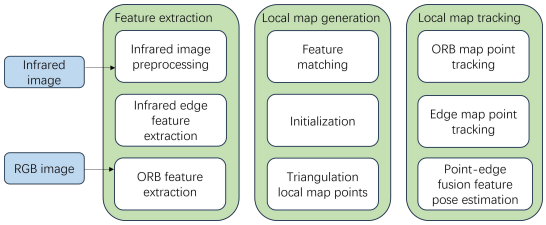

Fig. 1. Visual SLAM front-end algorithm framework for infrared-visible light fusion

A. Feature Extraction and Matching

The front-end system of the visual SLAM system designed in this paper extracts point features from visible light images, selecting the widely used ORB features for visible light image representation. For infrared images, the paper extracts edge point features for matching and tracking. This system enhances the robustness and localization performance of visual SLAM in low-light scenarios. The process flow for matching using edge features in infrared images is illustrated in the following diagram, with its effectiveness shown in Fig. 2. Infrared images suffer from weak texture and low contrast, which can result in an insufficient number of features if feature extraction is directly applied to the original image. This deficiency can adversely affect subsequent feature matching and tracking processes. Therefore, preprocessing of the infrared images is necessary before feature extraction. The raw images output by the infrared camera are 14-bit monochrome images. However, most image processing algorithms are designed to handle 8-bit images. Therefore, it is necessary to scale the 14-bit infrared images to 8-bit images before further processing. Next, the paper applies the Contrast Limited Adaptive Histogram Equalization (CLAHE) [5] algorithm to enhance the contrast of the infrared images. To further improve the image quality, the processed images undergo Gaussian smoothing. After preprocessing the infrared image, this paper adopts the Difference of Gaussian (DoG) [6] algorithm for edge feature extraction from infrared images.

In order to improve the accuracy of edge matching, this paper uses the pixel information of the local neighborhood of the edge point to construct an edge descriptor.When performing edge matching, because the pose estimation is relatively rough, the position of the projection point will also be inaccurate, which will cause edge mismatching. Therefore, using the distance from points to tangents instead of point-to-point distance can effectively reduce the impact of mis-matching on pose optimization accuracy. Subsequently, initialization provides a foundation for the system to achieve continuous localization and map building functions. The map initialization process provides initial estimates of camera motion poses and scale information, forming the initial local map points.

B. Pose Estimation Based on Infrared-Visible Light Feature Fusion

This section proposes a feature fusion pose estimation algorithm. The specific method involves establishing the reprojection error for both point features and edge features, jointly constructing error functions for both types of features, optimizing these error functions using nonlinear optimization methods, and thus solving for the camera pose.

1) Visual feature reprojection error: The algorithm presented in this paper utilizes bundle adjustment to minimize the reprojection error of map points for both point features and edge features, thereby optimizing the camera pose and map point positions. The construction method of reprojection error for point features is consistent with the construction method for ORB features. The construction process of reprojection error for edge features is similar to that of point features. According to Section A, the normal distance between two matching edge points should be used instead of the Euclidean distance to construct the reprojection error. The schematic diagram is shown in Fig. 3.

In the imaging plane, there are m edge features and n edge feature points. One of the edges, denoted as E, contains an edge feature point P. The projection of point P in reference frame I_r is denoted as P_1^r. The rough estimate of the camera pose from the reference frame to the current frame is denoted as T_r^c. Point P_1^r is transformed to the current frame I_c through T_r^c, resulting in the projected point $\hat{P}_2^c$ in the current frame. Using the edge feature matching method, the edge point P_2^c that matches point $\hat{P}_2^c$ is found. The normal vector of point P_2^c is denoted as n_2^c. Its expression is as follows:

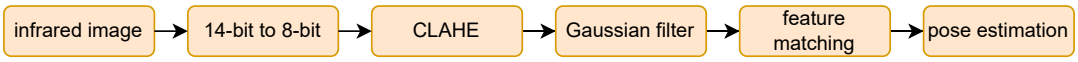

Fig. 2. Infrared image feature matching flow chart

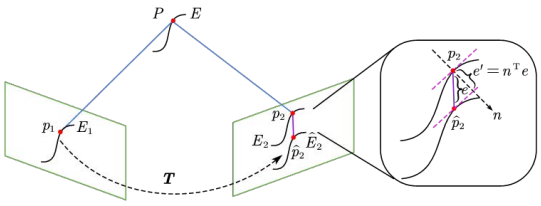

Fig. 3. Schematic diagram of edge point reprojection error

$$\begin{aligned}\{T_r^c, P\}^* &= \arg\min_{T_r^c, P} \frac{1}{2}\sum_{i=1}^{n}\left\|e_{edge}^i\right\|_2^2 \\ &= \arg\min_{T_r^c, P} \frac{1}{2}\sum_{i=1}^{n}\left\|n_i^{cT} e_{point}^i\right\|_2^2,\end{aligned} \tag{1}$$

Using Lie groups and Lie algebras to represent poses, the problem is transformed into an unconstrained least squares problem. Optimization is then performed using the Gauss-Newton method.

2) Pose estimation based on feature fusion: The principle of fusion localization based on point features and edge features is illustrated in Fig. 4.

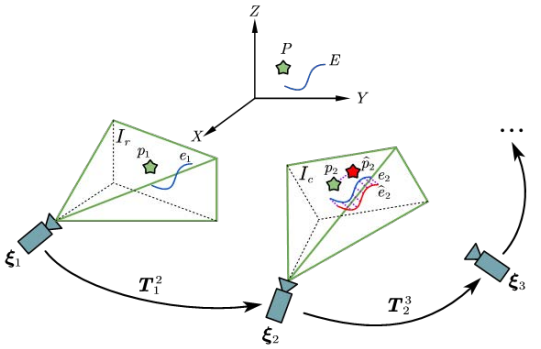

Fig. 4. Fusion positioning of point features and edge features

In space, there are a total of n feature points and m edges. Taking one feature point P and one edge E as examples, their projections in the reference frame I_r and the current frame I_c are denoted as p_1^r, p_2^c and e_1^r, e_2^c respectively. The pose of the camera in the reference frame ξ_1 is known, while the pose in the current frame $\hat{\xi}_2$ is a rough estimate. Based on $\hat{\xi}_2$, the spatial feature point P and edge feature E can be projected onto the current frame, obtaining projections $\hat{p}_2^c$ and $\hat{e}_2^c$. Next, we search for and match the projected points in the current frame, obtaining p_2^c and e_2^c. Due to the inaccuracy of the projection model and the rough estimate of the pose, there exists some error between $\hat{p}_2^c, \hat{e}_2^c$ and p_2^c, e_2^c. By optimizing the error function, we can accurately estimate the pose of the current frame, thereby achieving precise localization. Based on the matching results, the error function for fusion of point features and edge features is constructed as follows:

$$\begin{aligned}\boldsymbol{T}_1^{2*} = \arg\min_{\boldsymbol{T}_1^2} \frac{1}{2}\Bigg(&\sum_{i=1}^{n} w_{ORB}^i \left\|\boldsymbol{e}_{ORB}^i\right\|_2^2 \\ &+ \sum_{j=1}^{m} w_{edge}^j \left\|\boldsymbol{e}_{edge}^j\right\|_2^2\Bigg),\end{aligned} \tag{2}$$

where $\boldsymbol{e}_{ORB}^i$ represents the reprojection error of ORB point features, $\boldsymbol{e}_{edge}^j$ represents the reprojection error of edge features, and w_{ORB}^i and w_{edge}^j are observation weights related to the quantity of features. Based on the obtained Jacobian matrix, (2) is optimized using the bundle adjustment algorithm to obtain the pose transformation $\boldsymbol{T}_1^2$ from the reference frame to the current frame, thus obtaining an accurate estimate ξ_2 of the current frame's pose. The error function constructed above is also applicable to back-end optimization. By using the BA algorithm to jointly optimize the camera poses and map point positions, the accuracy of localization and mapping can be improved.

III. Pose Estimation Based on Vision-IMU Fusion

Traditional visual SLAM systems often face difficulties in feature extraction and matching when dealing with fast motion, low-texture environments, or strong lighting changes, which can affect localization accuracy and even lead to system failure. In these complex scenarios, IMUs, due to their ability to estimate motion states through inertial measurements in a short period, become an effective solution to enhance system robustness and performance. Additionally, the estimation results obtained from the camera can be used to correct the results obtained from the IMU, thereby eliminating the accumulated errors caused by IMU drift.

In the tracking thread of this visual-inertial SLAM system proposed in this paper, the pose estimation is divided into two modes based on whether the map points are updated, and there are differences in the optimization variables and optimization functions between these two modes. The optimization process of the tracking thread is illustrated in Fig. 5.

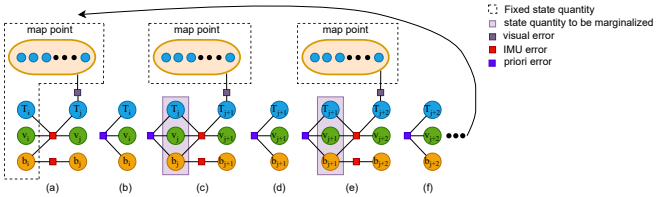

Fig. 5. Optimization of tracking threads

In Fig. 5, (a) when tracking frame j after map points are updated, the state of the previous keyframe i is not optimized but the IMU constraints associated with frame i are retained; (b) the optimization result from (a) is used as prior information for the next optimization step; (c) when map points are not updated, tracking frame $j+1$ involves optimizing the states of both frame j and frame $j+1$, using prior information from frame j and IMU data as constraints; (d) after optimization in (c), frame j is marginalized out, and its optimization result is used as prior information for the next optimization; (e) to (f) repeat the above process until a map update triggered by the loop closure thread or local mapping thread is detected.

When map points are updated during tracking frame j (Fig. 5 (a)), only the state of frame j needs to be optimized. In this scenario, the optimization variables comprise the camera pose, velocity, and IMU biases of frame j, while the optimization function includes the visual reprojection error of frame j and the IMU pre-integration error between the two frames. The optimization variables and functions for this process are as follows:

$$\boldsymbol{S}_j \doteq \left\{\boldsymbol{R}_j, \boldsymbol{p}_j, \boldsymbol{v}_j, \boldsymbol{b}_j^g, \boldsymbol{b}_j^a\right\}, \tag{3}$$

$$\begin{aligned}\boldsymbol{S}_j^* = \arg\min_{\boldsymbol{S}_j} \Bigg(& \boldsymbol{r}_{IMU}(i,j) + \sum_k \boldsymbol{r}_{ORB}(k,j) \\ & + \sum_l \boldsymbol{r}_{edge}(l,j) \Bigg).\end{aligned} \tag{4}$$

Among them, the expressions of each error term are as follows:

$$\begin{aligned}\boldsymbol{r}_{IMU}(i,j) &= \rho\left(\boldsymbol{r}_{I_{i,j}}^T, \sum\nolimits_I, \boldsymbol{r}_{I_{i,j}}\right), \\ \boldsymbol{r}_{ORB}(k,j) &= \rho\left(\boldsymbol{r}_{p_{k,j}}^T, \sum\nolimits_p, \boldsymbol{r}_{p_{k,j}}\right), \\ \boldsymbol{r}_{edge}(l,j) &= \rho\left(\boldsymbol{r}_{e_{l,j}}^T, \sum\nolimits_e, \boldsymbol{r}_{e_{l,j}}\right),\end{aligned} \tag{5}$$

$\boldsymbol{R}_j$ is the rotation matrix from the IMU coordinate system to the world coordinate system under the keyframe j, $\boldsymbol{p}_j$ is the position of the IMU under the keyframe j, $\boldsymbol{v}_j$ is the speed of the camera in the world coordinate system, $\boldsymbol{b}_j^g, \boldsymbol{b}_j^a$ is the offset of the gyroscope and accelerometer in the IMU. ρ is the robust kernel function, $\boldsymbol{r}_{I_{i,j}}$ is the pre-integration error between two frames, $\boldsymbol{r}_{p_{k,j}}$ is the reprojection error of point feature map points, and $\boldsymbol{r}_{e_{l,j}}$ is the reprojection error of edge feature map points. Each of these errors corresponds to a covariance matrix denoted by $\sum_I, \sum_p, \sum_e$. The optimization problem is solved using the Gauss-Newton method, and the optimized result is used as prior information for the next optimization process (Fig. 5 (b)).

When map points are not updated and tracking is conducted for frame $j+1$ (Fig. 5 (c)), optimization is performed for the states of both frame j and frame $j+1$. In this scenario, the optimization variables include the camera pose, velocity, and IMU bias for both frames. Additionally, the optimization function encompasses visual and IMU errors, as well as the prior errors generated at the end of the previous optimization process. The optimization variables and function for this process are as follows:

$$\begin{aligned}\boldsymbol{S}_{j,j+1} \doteq \Big\{ & \boldsymbol{R}_j, \boldsymbol{p}_j, \boldsymbol{v}_j, \boldsymbol{b}_j^g, \boldsymbol{b}_j^a, \boldsymbol{R}_{j+1}, \boldsymbol{p}_{j+1}, \\ & \boldsymbol{v}_{j+1}, \boldsymbol{b}_{j+1}^g, \boldsymbol{b}_{j+1}^a \Big\},\end{aligned} \tag{6}$$

$$\begin{aligned}\boldsymbol{S}_{j,j+1}^* = \arg\min_{\boldsymbol{S}_{j,j+1}} \Bigg(& \boldsymbol{r}_{IMU}(j,j+1) + \sum_k \boldsymbol{r}_{ORB} \\ & (k,j+1) + \sum_l \boldsymbol{r}_{edge}(l,j+1) + \boldsymbol{r}_{prior}(j) \Bigg).\end{aligned} \tag{7}$$

Among them, $\boldsymbol{r}_{prior}(j)$ is the prior error generated by the optimization result of frame j, and its expression is as follows:

$$\begin{aligned}\boldsymbol{r}_{prior}(j) &= \rho\left(\left[\boldsymbol{r}_R^T \boldsymbol{r}_v^T \boldsymbol{r}_p^T \boldsymbol{r}_b^T\right] \sum\nolimits_{prior} \left[\boldsymbol{r}_R^T \boldsymbol{r}_v^T \boldsymbol{r}_p^T \boldsymbol{r}_b^T\right]^T\right), \\ \boldsymbol{r}_R &= \log\left(\tilde{\boldsymbol{R}}_{\boldsymbol{j}} \boldsymbol{R}_{\boldsymbol{j}}\right), \boldsymbol{r}_v = \tilde{\boldsymbol{v}}_j - \boldsymbol{v}_j, \\ \boldsymbol{r}_p &= \tilde{\boldsymbol{p}}_j - \boldsymbol{p}_j, \boldsymbol{r}_b = \begin{bmatrix}\tilde{\boldsymbol{b}}_j^g - \boldsymbol{b}_j^g \\ \tilde{\boldsymbol{b}}_j^a - \boldsymbol{b}_j^a\end{bmatrix}.\end{aligned} \tag{8}$$

Among them, $(\sim)$ represents the estimated state obtained through the previous optimization process, and $\sum_{prior}$ is the Hessian matrix obtained from the previous optimization process. After completing the optimization in this case, the frame j will be marginalized, and its optimization result will be used as a priori information for the next optimization (Fig. 5 (d)). If the map points are not updated, follow the steps from Fig. 5 (e) to (f) for optimization. This means optimizing the state of two frames simultaneously and culling the oldest frame through a marginalization operation but retaining its optimization results as a priori. However, if the map point is updated, follow the steps (a) to (b) in Fig. 5 for optimization again. In this case, since the previous prior information has become invalid, only the state of the latest frame can be optimized.

IV. Experiments and Analysis

This section evaluates and analyzes the performance of the visual-inertial SLAM algorithm proposed in this paper in terms of positioning accuracy. We built an unmanned vehicle experimental platform based on Ubuntu 18.04 ROS architecture for experimental verification. This article uses Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) as evaluation indicators of algorithm positioning accuracy. The experiment is divided into two parts, one based on public data sets and the other on campus actual testing environment.

A. Experiments based on public data sets

This paper conducted performance testing of the algorithm using the M2DGR public dataset [7] recorded by Shanghai Jiao Tong University. The dataset was collected by ground unmanned vehicles equipped with various sensors, including RGB cameras, fisheye cameras, infrared cameras, 32-line LiDAR, event cameras, IMUs, etc., and ground truth positioning was obtained via GNSS. To evaluate the performance and robustness of the proposed algorithm in low-light

conditions, sequences recorded at nighttime were selected for assessment in this section.To validate the advancement and positioning performance of the algorithm, comparisons and analyses were conducted between the proposed algorithm and current mainstream open-source visual solutions such as ORB-SLAM3 [8], VINS-Mono [9], DSO [10], and ROVIO [11] on the test sequences. Fig. 6 illustrates the comparison of trajectory curves generated by different algorithms, and Fig. 7 shows the heat map of the absolute trajectory error between the real trajectory and the estimated trajectory using this algorithm.

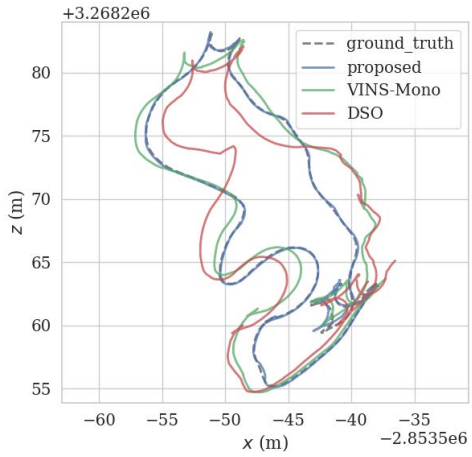

Fig. 6. Comparison of trajectories of different algorithms

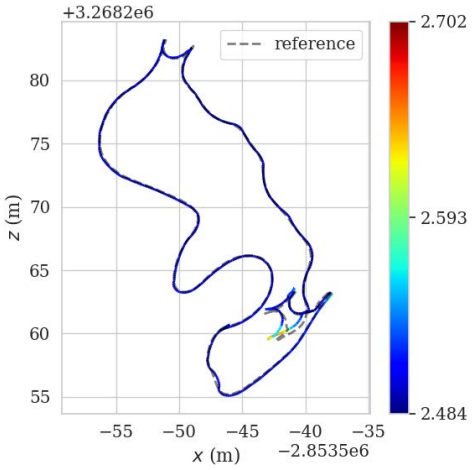

Fig. 7. Absolute trajectory error plot

In Fig. 6, the ground truth trajectory is depicted with a black dashed line, while the trajectories estimated by each algorithm are presented with solid lines of varying colors. A closer alignment between the estimated trajectory and the ground truth trajectory indicates higher positioning accuracy of the algorithm. In Fig. 7, the ground truth trajectory is represented by a gray dashed line, while the colored curves represent the estimated trajectories. The ATE and root mean square error (RMSE) values of RPE for each algorithm are calculated using the evo tool as evaluation metrics for positioning accuracy, and the results are summarized in Table I. In the table, $*$ indicates that the algorithm failed to run in that sequence, resulting in no positioning results. Smaller ATE and RMSE values of RPE in Table I indicate more accurate pose estimation and higher positioning accuracy by the algorithm. From Table I, it can be observed that in low-light conditions, algorithms based on visible light images are susceptible to tracking loss or initialization failure due to illumination variations. Even if positioning is achieved, significant positioning errors may exist. In contrast, the proposed algorithm effectively leverages the advantages of infrared cameras in low-light conditions, successfully achieving positioning in nighttime environments, and achieving the highest positioning accuracy among all algorithms.

B. Experiment based on real campus environment

To further evaluate the performance of the proposed algorithm, we first set up an unmanned vehicle experimental platform, as shown in Fig. 8. Subsequently, multiple datasets were recorded in the nighttime campus environment, and the performance of the proposed algorithm was compared with ORB-SLAM3, VINS-Mono, DSO, and ROVIO on these datasets. Fig. 9 illustrates the comparison of trajectory curves generated by different algorithms, and Fig. 10 shows the heat map of the absolute trajectory error between the real trajectory and the estimated trajectory using this algorithm.

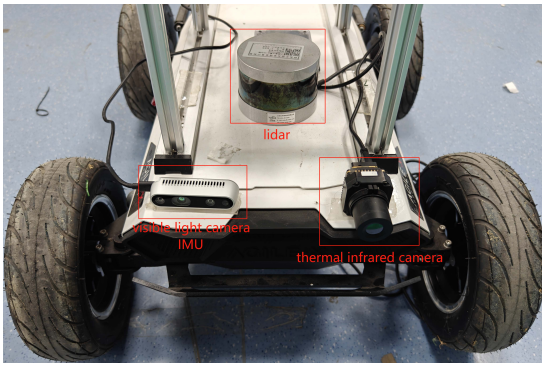

Fig. 8. Unmanned vehicle experimental platform

Based on Fig. 9 and Fig. 10, it can be observed that the proposed algorithm achieves good localization results in real environments, with its estimated trajectory closely matching the ground truth trajectory. The ATE and RMSE values of RPE for each algorithm were calculated using the evo tool as localization accuracy evaluation metrics. The results are presented in Table II.

Based on the analysis of the experimental results from Fig. 9, Fig. 10 and Table II, the proposed algorithm achieves precise localization results in real nighttime environments. In the trajectory comparison plots, it is observed that the estimated trajectory by the proposed algorithm closely aligns with the ground truth trajectory. In the error heatmaps, the trajectory errors of the proposed algorithm are predominantly blue, with minimal occurrence of red regions, indicating overall good localization accuracy without significant offset.

TABLE I

COMPARISON OF POSITIONING ACCURACY OF DIFFERENT ALGORITHMS UNDER THE M2DGR DATA SET (RMSE, UNIT: METERS)

ROVIO		VINS-Mono		DSO		ORB-SLAM3		PROPOSED	
ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE
*	*	12.553	1.304	9.820	1.392	*	*	2.503	0.245

TABLE II

COMPARISON OF POSITIONING ACCURACY OF DIFFERENT ALGORITHMS IN CAMPUS ENVIRONMENT (RMSE, UNIT: METERS)

ROVIO		VINS-Mono		DSO		ORB-SLAM3		PROPOSED	
ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE	ATE	RPE
4.519	4.580	3.968	2.076	*	*	*	*	3.490	1.549

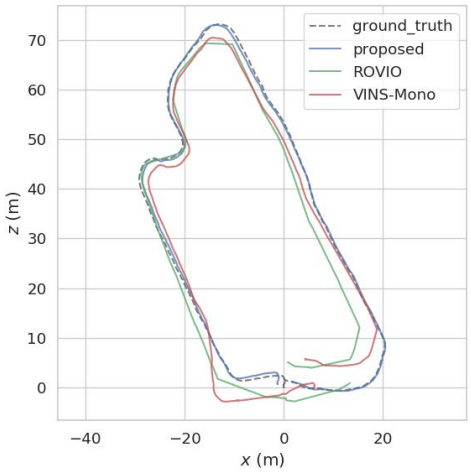

Fig. 9. Comparison of trajectories of different algorithms

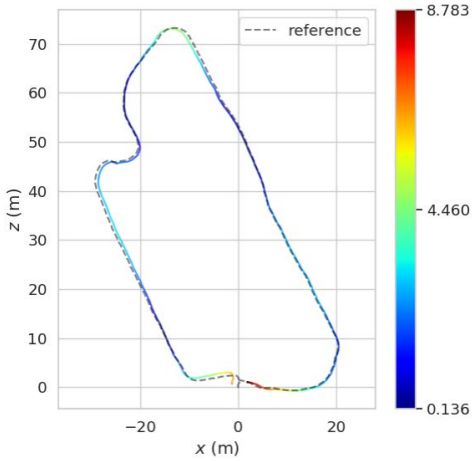

Fig. 10. Absolute trajectory error plot

In summary, the proposed algorithm demonstrates the capability to achieve accurate localization in low-light conditions during nighttime.

V. CONCLUSION

Traditional visual SLAM systems based on visible light images lack robustness when facing changes in environmental lighting conditions, making it difficult to achieve accurate localization in dark, smoky, or other harsh environments, thus limiting their applicability. The visual-inertial SLAM system proposed in this paper, which integrates infrared, visible light, and IMU sensors, addresses the insufficient robustness of the visual frontend of single-sensor systems, thereby improving localization accuracy.

REFERENCES

[1] Singandhupe A, La H M. A review of slam techniques and security in autonomous driving[C]//2019 third IEEE international conference on robotic computing (IRC). IEEE, 2019: 602-607.

[2] He Y, Deng B, Wang H, et al. Infrared machine vision and infrared thermography with deep learning: A review[J]. Infrared physics & technology, 2021, 116: 103754.

[3] Chen C, Zhu H, Li M, et al. A review of visual-inertial simultaneous localization and mapping from filtering-based and optimization-based perspectives[J]. Robotics, 2018, 7(3): 45.

[4] Mur-Artal R, Montiel J M M, Tardos J D. ORB-SLAM: a versatile and accurate monocular SLAM system[J]. IEEE transactions on robotics, 2015, 31(5): 1147-1163.

[5] Reza A M. Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement[J]. Journal of VLSI signal processing systems for signal, image and video technology, 2004, 38: 35-44.

[6] Marr D, Hildreth E. Theory of edge detection[J]. Proceedings of the Royal Society of London. Series B. Biological Sciences, 1980, 207(1167): 187-217.

[7] Yin J, Li A, Li T, et al. M2dgr: A multi-sensor and multi-scenario slam dataset for ground robots[J]. IEEE Robotics and Automation Letters, 2021, 7(2): 2266-2273.

[8] Campos C, Elvira R, Rodríguez J J G, et al. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam[J]. IEEE Transactions on Robotics, 2021, 37(6): 1874-1890.

[9] Qin T, Li P, Shen S. Vins-mono: A robust and versatile monocular visual-inertial state estimator[J]. IEEE Transactions on Robotics, 2018, 34(4): 1004-1020.

[10] Engel J, Koltun V, Cremers D. Direct sparse odometry[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 40(3): 611-625.

[11] Bloesch M, Omari S, Hutter M, et al. Robust visual inertial odometry using a direct EKF-based approach[C]//2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2015: 298-304.