A SCENE-ASSISTED POINT-LINE FEATURE BASED VISUAL SLAM METHOD FOR AUTONOMOUS FLIGHT IN UNKNOWN INDOOR ENVIRONMENTS

S. Cheng [1], J. Yang [1], Z. Kang [1] *, P. H. Akwensi [1]

[1] Department of Remote Sensing and Geo-Information Engineering, School of Land Science and Technology, China University of Geosciences, Xueyuan Road, Beijing, 100083 CN – 18101361751@163.com, jtyang66@126.com, zzkang@cugb.edu.cn, ahtimeless@outlook.com

Commission IV, WG IV/5

KEY WORDS: Unmanned aerial vehicles, Autonomous flight, Simultaneous localization and mapping, Scene interpretation

ABSTRACT:

Since Global Navigation Satellite System may be unavailable in complex dynamic environments, visual SLAM systems have gained importance in robotics and its applications in recent years. The SLAM system based on point feature tracking shows strong robustness in many scenarios. Nevertheless, point features over images might be limited in quantity or not well distributed in low-textured scenes, which makes the behaviour of these approaches deteriorate. Compared with point features, line features as higher-dimensional features can provide more environmental information in complex scenes. As a matter of fact, line segments are usually sufficient in any human-made environment, which suggests that scene characteristics remarkably affect the performance of point-line feature based visual SLAM systems. Therefore, this paper develops a scene-assisted point-line feature based visual SLAM method for autonomous flight in unknown indoor environments. First, ORB point features and Line Segment Detector (LSD)-based line features are extracted and matched respectively to build two types of projection models. Second, in order to effectively combine point and line features, a Convolutional Neural Network (CNN)-based model is pre-trained based on the scene characteristics for weighting their associated projection errors. Finally, camera motion is estimated through non-linear minimization of the weighted projection errors between the correspondent observed features and those projected from previous frames. To evaluate the performance of the proposed method, experiments were conducted on the public EuRoc dataset. Experimental results indicate that the proposed method outperforms the conventional point-line feature based visual SLAM method in localization accuracy, especially in low-textured scenes.

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are becoming increasingly popular and crucial autonomous platforms for many applications ranging from hazard monitoring, search and rescue operations, emergency response, Special Weapons and Tactics (SWAT) support to intelligence, surveillance, and reconnaissance (ISR). It can be noted that the operation environment, such as indoor, a group of complex urban buildings or woods, becomes more diverse and complex. Since Global Navigation Satellite System (GNSS) in complex dynamic indoor or outdoor environments may be unavailable, visual Simultaneous Localization And Mapping (SLAM) systems have gained importance in robotics and its applications in recent years. Simultaneous localization and mapping (SLAM) refer to the phenomenon where a robot moving in an unknown environment estimates its self-localization according to the surrounding environment and using its self-localization to establish a surrounding environment map. The localization and mapping become a process of correlation and interaction (Klein and Murray, 2007). Using the visual camera as a sensor to locate and sense the environment is called V-SLAM.

Compared with other sensors, visual cameras are cheaper, more intuitive, lower in power consumption, and images can provide abundant information. Thus, it has gradually become irreplaceable in the SLAM community to date (Fuentes et al., 2015). Common visual sensors mainly include monocular cameras (Andrew et al., 2007), binocular cameras (Victor et al., 2015), depth cameras (Renato et al., 2015; Hu et al., 2012) and so on. Andrew et al. (2007) developed the first monocular vision SLAM system called MonoSLAM that uses extended Kalman filter to estimate camera motion. At the same time, Klein and Murray (2007) introduced the PTAM system which is the first monocular vision SLAM system which is based on key frame BA and simultaneous tracking and mapping, making real-time V-SLAM a reality. Mur et al. (2015) designed the highly influential open source monocular ORB-SLAM system, and Mur and Tardós (2017) expanded it into an open source ORB-SLAM2 system which support binocular vision and RGB-D cameras the following year. Li et al. (2017) proposed a monocular VINS SLAM system in INS-vision fusion. They are also open sourcing the world's first vision-IMU fusion SLAM system on mobile phone and Linux system, which can be run on IOS devices and work well on UAV control.

Most feature-based SLAM systems use point feature tracking to estimate camera motion, such as Scale-Invariant Feature Transform (SIFT) (David, 1999), Speeded Up Robust Features (SURF) (Bay et al., 2003), ORiented Brief (ORB)（Rublee et al. 2011）, along a sequence of images. The maturity of point feature extraction and matching allowed point features to become widely used in inter-frame tracking in SLAM topics. ORB-SLAM is the most representative, effective and visual camera-adapted point feature tracking SLAM. Indeed, point features over images might be scarce or not well distributed in low-textured scenes, which makes the behavior of these approaches deteriorate. Compared with point features, line features as higher-dimensional features can provide more environmental information in complex scenes. Zhang et al. (2015) in the

* Corresponding author.

StructSLAM system proved the irreplaceable advantage of line feature in indoor environment by replacing point features with line features.

The previous work concluded that the combination of both point and line segments enable visual SLAM system to robustly work in a wider variety of scenarios. Gomez et al. (2016) described PL-SVO, a visual odometer based on point-line feature, which uses the photometric difference between pixels of three-dimensional line segments to estimate pose increment. The author described in detail camera motion of the nonlinear minimum projection error estimation with joint point-line features. Gomez et al. (2017) then introduced point-line feature in loop detection to realize the binocular stereo SLAM system PL-SLAM which combined point-line feature. Meanwhile the authors (2016) introduce a VO system to weight the errors of different features according to their covariance matrices. Pumarola et al. (2017) introduced real-time monocular visual SLAM, which combines point and line features for localization and mapping. Di et al. (2016) obtained the inverse of the error as the weights of different data sources in RGB-D SLAM and achieved good results. Wang et al. (2018) introduced the line feature angle as one of the parameters of the re-projection error, and designed the PL-SLAM method to adjust the weight ratio of the point-line based on the estimation of the camera state residual.

As a matter of fact, line segments are usually sufficient in any human-made environment, even in low textured scenes, while the quality and quantity of the detected point decreases in low-texture environments, which suggests that scene characteristics remarkably affect the performance of point-line feature based visual SLAM system. In this paper, we develop a scene-assisted point-line feature based visual SLAM method for autonomous flight in unknown indoor environments. The rest of this paper is organized as follows. Section 2 describes the proposed method in detail. Section 3 presents the experimental results and analysis for evaluating the proposed method. This paper concludes with a discussion of future research considerations in Section 4.

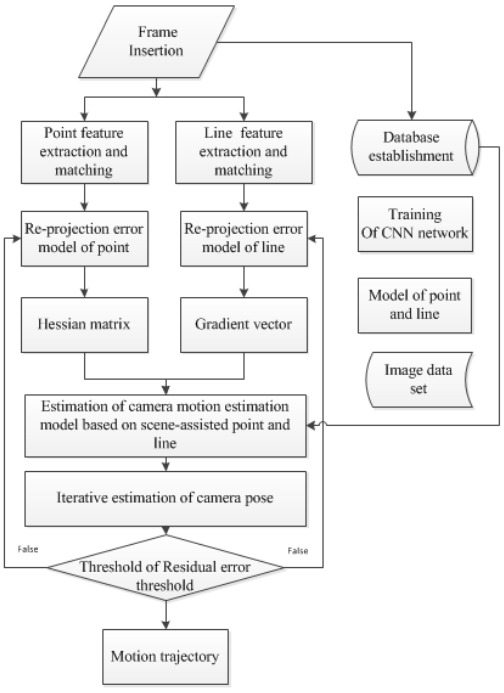


Fig. 1. Flowchart of our proposed visual SLAM method.

2. METHODOLOGY

Fig. 1 illustrates the flow chart of the proposed method, which consists of the following three parts: (1) the establishment of point-line features based on a multi-feature tracking model; (2) the establishment of CNN-based point and line weighted model; (3) the establishment of adaptive weighted Gauss-Newton estimation model. Key algorithms are given in detail below.

2.1 Tracking model of point and line features

The maturity and easy parameterization of point feature extraction and matching techniques make them widely used in the inter-frame tracking of visual SLAM systems. However, in low-texture areas, relying on single-point feature tracking is often poor. But line features are widely present in artificial indoor scenes and both can effectively make up for each other's respective shortcomings. In this section, we establish a tracking model based on both point and line features, and use the acquired camera image features (points, line features) to establish a relationship between a series of stereo frames.

2.1.1 Tracking model of Point Features： In terms of point feature extraction and matching, the frequently-used algorithms include SURF, SIFT, ORB and so on. The SIFT algorithm compared to the ORB feature algorithm has high precision but complex computation, thus cannot meet the real-time requirements of SLAM. By integrating FAST feature points with BRIEF descriptors, the ORB feature algorithm generates improved and optimized features. Therefore, the ORB algorithm exhibits good tracking effect and good real-time performance, and can realize real-time effective point feature tracking. At the same time, in order to reduce the number of tracking outliers, we only consider the measurement of the best match in the left and right images (Gomez et al.,2017). Finally, in order to minimize false matches, we removed the matching points in the descriptor space where the matching distance is less than four times the minimum matching distance. Fig. 2 shows a schematic diagram of extracting image features using ORB.

2.1.2 Tracking model of Line Segment Features： ORB-SLAM is one of the most representative SLAM systems based on ORB features. The system builds a tracking model based on ORB features, which can effectively realize simultaneous localization and mapping of unknown regions in most scenarios. However, in weak texture scenes, the point features are limited. For example, when stairs are used, effective tracking cannot be established based on single point feature establishment. As shown in Fig. 3, the number of effective matching point features is small. Obviously, in such a scenario, the estimation of inter point frame motion by a single point feature cannot meet the accuracy requirement. In order to fully analyse the scenes that easily make the performance of the ORB-SLAM system degrade, we use the ORB-SLAM system to experiment on different

Fig. 2 ORB feature extraction

Fig. 3 ORB point feature matching in low-texture scene

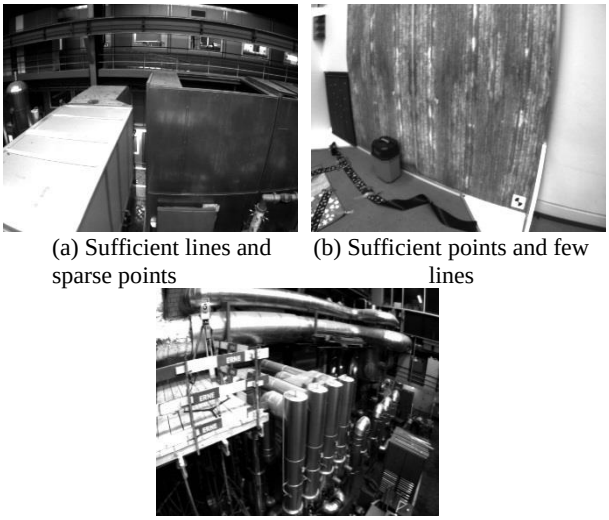
Fig. 4 ORB-SLAM lost tracking and scene correlation analysis

Fig. 5 LSD feature extraction

scenarios. The experimental results are shown in Fig. 4. The results show that the light mutation, narrow field of vision and low texture areas are likely to cause the locking of the ORB-SLAM system to be lost.

By fully analysing the ORB-SLAM, we can conclude that in the low-texture region, the inter-frame matching effect tends to be poor due to the scarcity of its point features. The line features are widely used in most scenes, especially in artificial scenes such as stairs and walls. The line features are very rich and can effectively compensate for the sparseness of point features in such scenes to adapt to a wider range of scenarios.

In line feature extraction, we use the Line Segment Detector (LSD) line extraction algorithm (Grompone et al., 2010) to extract line segment features. The algorithm has high precision, real-time and repeatability. For line feature tracking, we use Line Band Descriptor (LBD). a algorithm establishes line feature matching between adjacent frames. Similar to the point feature, it is necessary to detect whether the two-frame matching is the best match for each other. As shown in Fig. 5, it can be seen that the line features also have good performance in scenes such as white walls.



(a) Sufficient lines and sparse points (b) Sufficient points and few lines

(c) Sufficient points and lines

Fig. 6 Three types of the typical scenes

2.2 Establishment of point and line feature weighted model based on CNN network

As mentioned earlier, line segments are usually sufficient in any human-made environment, even in low textured scenes, while the quality and quantity of the detected point decreases in low-texture environments, which suggests that scene characteristics remarkably affect the performance of point-line feature based visual SLAM system. For instance, in human-made environment with low texture, line segments can provide rich structural information and the detection of the line segment is more robust than that of the point feature. According to our experience and the scene characteristics analysis, the current frame can be classified into three types: (a) sufficient lines and sparse points, (b) sufficient points and few lines and (c) sufficient points and lines. Fig. 6 shows three types of the typical scenes. In such situation, if the current frame captured from the camera corresponds to the ‘sufficient lines and sparse points’, the weight of the line segment features should be larger than that of the point features. If the current frame captured from the camera belongs to the ‘sufficient points and few lines’, the weight of the line segment features should be smaller than that of the point features. If the current frame captured from the camera belongs to the ‘sufficient points and lines’, the weight of both line segment features and point features should be equal. Consequently, we weight the point features and line segment features at the subsequent re-projection error model. Due to the superior classification performance and the real-time efficiency, we fine-tune a CNN-based network model based on resnet-50 (He et al., 2016) and conduct the pre-trained CNN-based model for classifying the current frame.

2.3 Adaptive weighted re-projection error model

In order to estimate the motion of the camera relative to the previous frame, we first project the key points and lines from the previous frame to the current frame, and establish a re-projection error model based on point and line features. After the new image frame is inserted, the current image frame is classified using the pre-trained CNN-based network model for weighting both the points and the line segments. According to the type of the current frame, we establish a weighted Gauss-Newton estimation method for iterative estimation.

2.3.1 Estimation model of point features： According to the

difference between the re-projected 2D position and the matching point 2D position, the error distance between the re-projected points and their corresponding observation points on the current frame is minimized, and a least-squares error estimation model based on the point feature is established. The two core parameters of the Gauss-Newton estimation method, the Hessian matrix $\boldsymbol{H}$ and the Gradient vector $\boldsymbol{g}$, are constructed. First, in the re-projection error model, the error of the i[th] point feature can be described as follows:

$$e_p^i(\boldsymbol{\zeta}) = \boldsymbol{K}*\boldsymbol{T}(\boldsymbol{\zeta}).P_{\text{xyz\_world}} - p \tag{1}$$

where $\boldsymbol{\zeta}$ is a six-dimensional Lie algebra vector representing the motion of the camera, $\boldsymbol{K}$ is the camera's internal reference matrix, and $\boldsymbol{T}(\boldsymbol{\zeta})$ is the transformation matrix between two frames. $P_{\text{xyz-word}}$ refers to the world coordinate system of SLAM. Generally, the camera coordinate system of the first frame represents the world coordinate system of SLAM, p is the coordinate of the current frame. $e_p^i(\boldsymbol{\zeta})$ is the resultant error vector, $p(x, y)$ is the corresponding observed point of the re-projected point. $p'(x', y')$ According to the chain rule, we solve the Jacobian matrix $\boldsymbol{J}$:

$$\boldsymbol{J} = \frac{\partial e_p^i e(\boldsymbol{\zeta})}{\partial p'} \frac{\partial p'}{\partial P'} \frac{\partial P'}{\partial \boldsymbol{\zeta}} = \frac{\partial e_p^i e(\boldsymbol{\zeta})}{\partial p'} \frac{\partial P'}{\partial \boldsymbol{\zeta}} \tag{2}$$

$$\frac{\partial P'}{\partial \boldsymbol{\zeta}} = \begin{bmatrix} f_x \frac{1}{Z'} & 0 & -f_x \frac{X'}{Z'^2} & -f_x \frac{X'Y'}{Z'^2} & f_x(1+\frac{X'}{Z'^2}) & -f_x \frac{Y'}{Z'} \\ 0 & f_y \frac{1}{Z'} & -f_y \frac{Y'}{Z'^2} & -f_y(1+\frac{Y'}{Z'^2}) & f_y \frac{X'Y'}{Z'^2} & f_y \frac{X'}{Z'} \end{bmatrix} \tag{3}$$

$$\frac{\partial e_p^i e(\boldsymbol{\zeta})}{\partial p'} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4}$$

According to the formulas (2), (3), and (4), the Jacobian matrix $\boldsymbol{J}$ can be obtained, and the Gauss-Newton method uses $\boldsymbol{J}^T\boldsymbol{J}$ as the approximation of the Hessian matrix.

$$\begin{cases} \boldsymbol{H}_P^i = \boldsymbol{J}_p{}^T \boldsymbol{P} \boldsymbol{J}_p \\ \boldsymbol{g}_P^i = -\boldsymbol{J}_p{}^T \boldsymbol{P} e_p^i(\boldsymbol{\zeta})' \end{cases} \tag{5}$$

Therefore, the $\boldsymbol{H}$ matrix and the $\boldsymbol{g}$ gradient vector of the frame point cloud set can be obtained, and the feature points extracted from the frames are independent and equal, hence the matrix P is defined as:

$$\boldsymbol{P} = \begin{vmatrix} \frac{1}{1+\| \boldsymbol{e}_p^i \|} & 0 \\ 0 & \frac{1}{1+\| \boldsymbol{e}_p^i \|} \end{vmatrix} \tag{6}$$

$$\begin{cases} \boldsymbol{H}_P = \sum_{i=1}^{n} \boldsymbol{H}_p^i \\ \boldsymbol{g}_p = \sum_{i=1}^{n} \boldsymbol{g}_p^i \end{cases} \tag{7}$$

2.3.2 Estimation model of line features： Similar to the point feature, the line feature can establish a camera trajectory tracking model based on the two ends, and estimate the camera trajectory using the Gauss-Newton method. The coefficient model is outlined as:

$$e_l^i(\boldsymbol{\zeta}) = \begin{bmatrix} e^1(\boldsymbol{\zeta}) \\ e^2(\boldsymbol{\zeta}) \end{bmatrix} = \begin{bmatrix} a*x_{p'} + b*y_{p'} + c \\ a*x_{q'} + b*y_{q'} + c \end{bmatrix} \tag{8}$$

where p' , q' are respectively:

$$\begin{cases} p'(x_{p'}, y_{p'}) = \boldsymbol{K} * \exp(\boldsymbol{\zeta}^\wedge).P_{\text{XYZ-word}} \\ q'(x_{p'}, y_{p'}) = \boldsymbol{K} * \exp(\boldsymbol{\zeta}^\wedge).Q_{\text{XYZ-word}} \end{cases} \tag{9}$$

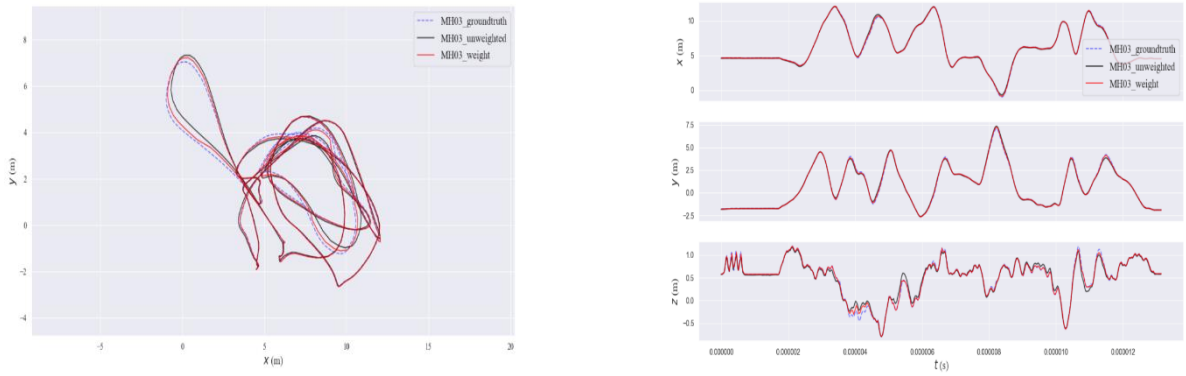
In equations (8) and (9), a, b, and c are line feature coefficients, points p and q are the two endpoints of the line segment. p' and q' are the two endpoints of the re-projected line segment, where $e^1(\boldsymbol{\zeta})$ and $e^2(\boldsymbol{\zeta})$ are the distances of p' and q' to the line segment pq, respectively. Similar to point feature derivation, according to the chain rule:

$$\boldsymbol{J} = \begin{bmatrix} \frac{\partial e^1(\boldsymbol{\zeta})}{\partial p'} \frac{\partial p'}{\partial P'} \frac{\partial P'}{\partial \boldsymbol{\zeta}} \\ \frac{\partial e^2(\boldsymbol{\zeta})}{\partial q'} \frac{\partial q'}{\partial Q'} \frac{\partial Q'}{\partial \boldsymbol{\zeta}} \end{bmatrix} = \begin{bmatrix} \frac{\partial e^1(\boldsymbol{\zeta})}{\partial p'} \frac{\partial p'}{\partial \boldsymbol{\zeta}} \\ \frac{\partial e^2(\boldsymbol{\zeta})}{\partial q'} \frac{\partial q'}{\partial \boldsymbol{\zeta}} \end{bmatrix} \tag{10}$$

$$\frac{\partial e^1}{\partial p'} = \frac{\partial e^2}{\partial q'} = [a, b] \tag{11}$$

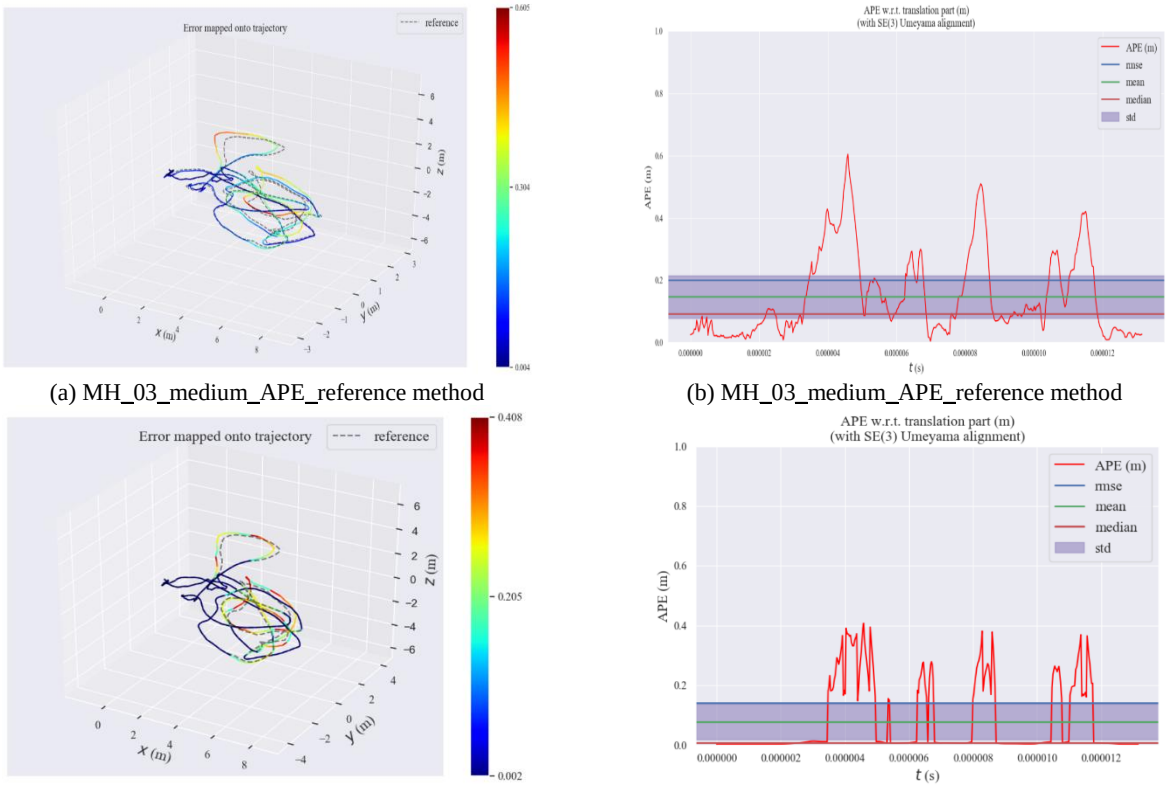
$$\begin{cases} \boldsymbol{H}_l^i = \boldsymbol{J}_l^T \boldsymbol{P} \boldsymbol{J} \\ \boldsymbol{g}_l^i = -\boldsymbol{J}_l^T \boldsymbol{P} e_l^i(\boldsymbol{\zeta})' \end{cases} \tag{12}$$

Also, similar to point features, line features are equal and independent of each other, so $\boldsymbol{P}$ is defined as:



(a)MH_03_trajectory (b) MH_03_trajectory XYZ

Fig. 7 MH_03_medium_trajectory



(a) MH_03_medium_APE_reference method (b) MH_03_medium_APE_reference method

(c) MH_03_medium_APE_our method (d) MH_03_medium_APE_our method

Fig. 8 MH_03_medium_APE

$$\boldsymbol{P} = \begin{vmatrix} \dfrac{1}{1+\| e^1(\zeta) \|} & 0 \\ 0 & \dfrac{1}{1+\| e^2(\zeta) \|} \end{vmatrix} \tag{13}$$

$$\begin{cases} \boldsymbol{H}_l = \sum_{j=1}^{n} \boldsymbol{H}_l^j \\ \boldsymbol{g}_l = \sum_{j=1}^{n} \boldsymbol{g}_l^j \end{cases} \tag{14}$$

2.3.3 Estimation model of point and line: According to the nonlinear least squares error model established by the point and line features, and the respective W_p and W_l returned by the database, an estimation model of adaptive joint point-line features can be established. We apply a new reconstructed Hessian matrix and gradient vector to estimate the camera pose based on Gauss-Newton estimation.

$$\begin{cases} \boldsymbol{H} = \boldsymbol{H}_P * W_P + \boldsymbol{H}_l * W_l \\ \boldsymbol{g} = \boldsymbol{g}_P * W_P + \boldsymbol{g}_l * W_l \end{cases} \tag{15}$$

3. EXPERIMENTAL VALIDATION

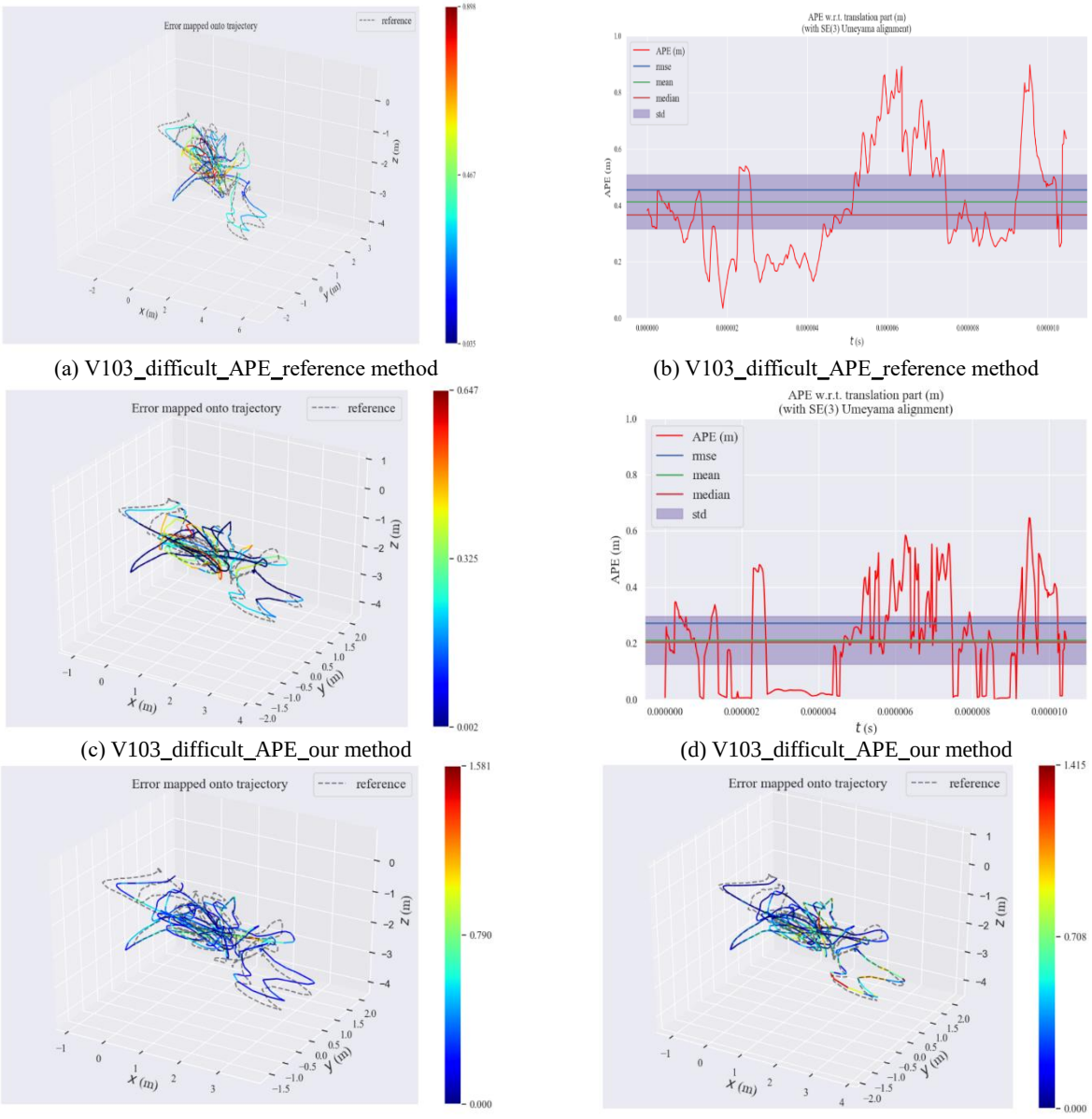
In this section, we test the performance of our proposed method using the EuRoc data set for test positioning results testing. The EuRoC MAV data set consists of 11 stereo sequences covering three different environments (Burri et al., 2016): two indoor rooms and one industrial scene. According to the flight speed, lighting conditions and texture conditions of the drone, different data sets are presented. Each data set provides a complete image frame and accurate groundtruth, and provides important parameters for capturing the camera's internal information and other sensors.

First of all, without considering the problem of point-line weighting, experiments are carried out using the power of the dotted line, and then the comparative experiment based on the CNN-based adaptive weighted PL-SLAM system is proposed. The main scope of comparison includes: the main comparison ranges include estimated trajectory versus groundtruth relative pose error (RPE) and absolute pose error (APE). In our implementation, the weights for the different types of scenes are determined by the following criteria: (a) Sufficient lines and sparse points ($W_P = 0.25$, $W_l = 0.75$); (b) Sufficient points and few lines ($W_P = 0.75$, $W_l = 0.25$); (c) Sufficient points and lines ($W_P = 0.5$, $W_l = 0.5$). All experiments were performed on the same computer (Intel(R) CORE(TM) I5-4200 CPU @2.5GHz, and 8G RAM without GPU parallelization).

Fig. 7 shows a comparison of the results of the trajectories in the MH_03_medium room. The dashed line represents the dataset groundtruth, the black line represents the reference method's trajectory, and the red line represents the trajectory of the our method. As can be seen, the visual SLAM method using the scene-assisted red line feature produces a trajectory that is closer to the groundtruth compared to the reference method. Thus, the maximum APE of the reference method is -0.605 m, and that of our method is -0.408 m.

In order to fully verify the accuracy of our proposed algorithm and the ability to adapt to more complex environments, we selected V103_ difficult room for experiments, the experimental method is the same as above. The comparison is as shown in Fig. 9.

From the experimental results of Figs 7, 8, and 9, it is not difficult to conclude that the trajectory obtained by the reference



(a) V103_difficult_APE_reference method (b) V103_difficult_APE_reference method

(c) V103_difficult_APE_our method (d) V103_difficult_APE_our method

(e) V103_difficult_RPE_reference method (f) V103_difficult_RPE_our method

Fig.9 V103_ difficult APE and RPE comparison chart

method SLAM system has a large offset during aircraft turn when the scene is not assisted. As shown in Fig. 9 (a), (b), (c), and (d), the reference method has a maximum deviation of -0.898 meters when the aircraft is sharply turning, and based on our proposed algorithm, the maximum error can be reduced to -0.647 meters at the aircraft turn scene. Fig. 9 (a), (b) and (e) represent the reference methods being compared with the proposed method Fig. 9(c), (d) and (f). The comparison in Fig. 9 proves that the proposed method can detect the aircraft in advance when faced with complex scene changes like fast turning of the aircraft, etc. The environmental information enables the system to make timely and effective adjustments, improving the accuracy of the algorithm's results.

4. CONCLUSION

In this paper, we propose a SLAM method based on scene perception for autonomous flight positioning of auxiliary aircraft. When facing different complex scenes, the aircraft can autonomously perceive the scene information according to the image information returned by the camera, and adjust the weight ratio in the points. This is undoubtedly very important for the autonomous flight of the aircraft. When a certain feature in the scene is sparse, the proportion of the feature in the pose estimation is reduced in time, thus the lack of equal weight ratio of the traditional method is improved. In the follow-up study, we will continue to discuss how to effectively locate the sparse regions of feature pairs. In addition, we will introduce inertial measurement unit into the current system, where the integration of the V-SLAM system and inertial measurement unit interacts with the scene for autonomously evaluating the accuracy of the inertial navigation in during positioning and adjusting the proportion of the V-SLAM system and the inertial navigation system in time to realize the fusion of the inertial navigation and the V-SLAM system as an online system.

References

Andrew, J. D., Ian, D. R., Nicholas, D. M., and Olivier, S., 2007. MonoSLAM: Real-Time Single Camera SLAM. *IEEE transactions on pattern analysis and machine intelligence*,vol. 29, pp.1052-1067.

Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V., 2008. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding.* Vol, 110, pp.346–359

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M. W., and Siegwart, R., 2016. The EuRoC micro aerial vehicle datasets. *Int. J. Robot. Res*., vol. 35, pp.1157–1163.

David G. L., 1999. Object recognition from local scale-invariant features. *International Conference on Computer Vision, Corfu, Greece*, pp. 1150-1157.

Di, K., Zhao, Q., Wan, W., Wang, Y., and Gao, Y., 2016. RGB-D SLAM based on extended bundle adjustment with 2D and 3D information. *Sensors*, vol. 16, 1285.

Fuentes P., J., Ruiz A., J., and Rendón M., J. M., 2015. Visual simultaneous localization and mapping: A survey. *Artif. Intell. Rev,* vol. 43, pp.55–81.

Gomez O., R., Moreno, F. A., Scaramuzza, D., and Gonzalez J., J., 2017. PL-SLAM: A Stereo SLAM System through the Combination of Points and Line Segments *arXiv*:1705.09479.

Gomez O., R., and Gonzalez J., J., 2016. Robust stereo visual odometry through a probabilistic combination of points and line segments. 2016 *IEEE International Conference on Robotics and Automation (ICRA)*, 16–21, pp. 2521–2526.

Gomez-O, R., Briales, J., and Gonzalez-Jimenez, J., 2016. PL-SVO: Semi-direct Monocular Visual Odometry by combining points and line segments. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4211–4216.

Grompone, V. G. R., Jakubowicz, J., Morel, J. M., and Randall, G., 2010. LSD: A fast line segment detector with a false detection control. *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, pp. 722–732.

Hu, G., Huang, S., Zhao, L., Alempijevic, A., and Dissanayake, G., 2012. A robust RGB-D SLAM algorithm. *In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),Vilamoura, Portugal,* 7–12, pp. 1714–1719.

He K, Zhang, X. Y., Zhang, S. Q., and Sun, R. J., 2016. Deep residual learning for image recognition. *IEEE conference on computer vision and pattern recognition*, pp.770-778.

Klein, G., and Murray, D., 2007. Parallel tracking and mapping for small AR workspaces (PTAM). *IEEE and ACM International Symposium on Mixed and Augmented Reality, Washington, DC, USA*, 13–16, pp. 1–10

Kurt, K., and Motilal, A., 2008. FrameSLAM: from Bundle Adjustment to Real-time Visual Mappping *IEEE Trans.Robot*.24, pp. 1066-1077.

Mur-Artal, R., J, M. M. Montiel., and J. D. Tardós., 2015. ORBSLAM: a Versatile and Accurate Monocular SLAM System, *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163.

Mur-Artal, R., and Tardós, J. D., 2017. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot*, vol. 33, pp.1255–1262.

Li, P. L., Qin, T., Hu, B. T., Zhu, F. Y., and Shen, S. J., 2017. Monocular Visual-Inertial State Estimation for Mobile Augmented Reality.*ISMAR2017.*

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M.W., and Siegwart, R., 2016. The EuRoC micro aerial vehicle datasets. *Int. J. Robot. Res*. vol. 35, pp. 1157–1163.

Pumarola, A., Vakhitov, A., Agudo, A., Sanfeliu, A., and Moreno-Noguer, F., 2017. PL-SLAM: Real-time monocular visual SLAM with points and lines. In Proceedings of the 2017 *IEEE International Conference on Robotics and Automation (ICRA*), Singapore, vol. 29, pp. 4503–4508.

Renato, F. S. M., Richard, A .N., Hauke, S., Paul, H. J. K., and Andrew, J. D., 2013. SLAM++: Simultaneous global location and mapping at the level of objects. *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1352-1359.

Rublee, E., Rabaud, V., and Konolige, K., 2011. ORB: An efficient alternative to SIFT or SURF. *IEEE International Conference on Computer Vision (ICCV)*, 6–13, pp. 2564–2571.

Victor, H. S., Felipe, G. B., and Eduardo, T. B., 2015. A SoC With FPGA Landmark Acquisition System for Binocular Visual SLAM. *IEEE DOI 10.1109/LARS-SBR*.

Wang, R. Z., Di, K. C., Wan, W. H., and Wang, Y. K., 2018. Improved Point-Line Feature Based Visual SLAM Method for Indoor Scenes. *Sensors,* vol.18, no.10, pp.3559.

Zhang, G., Jin, H.L., Lim, J., and Suh, I.H., 2015. Building a 3-d line-based map using stereo SLAM. *IEEE Trans. Robot*. vol. 31, pp.1364–1377.