A visual SLAM method based on point-line fusion in weak-matching scene

Baofu Fang¹,² and Zhiqiang Zhan¹,²

Abstract
Visual simultaneous localization and mapping (SLAM) is well-known to be one of the research areas in robotics. There are many challenges in traditional point feature-based approaches, such as insufficient point features, motion jitter, and low localization accuracy in low-texture scenes, which reduce the performance of the algorithms. In this article, we propose an RGB-D SLAM system to handle these situations, which is named Point-Line Fusion (PLF)-SLAM. We utilize both points and line segments throughout the process of our work. Specifically, we present a new line segment extraction method to solve the overlap or branch problem of the line segments, and then a more rigorous screening mechanism is proposed in the line matching section. Instead of minimizing the reprojection error of points, we introduce the reprojection error based on points and lines to get a more accurate tracking pose. In addition, we come up with a solution to handle the jitter frame, which greatly improves tracking success rate and availability of the system. We thoroughly evaluate our system on the Technische Universität München (TUM) RGB-D benchmark and compare it with ORB-SLAM2, presumably the current state-of-the-art solution. The experiments show that our system has better accuracy and robustness compared to the ORB-SLAM2.

Keywords
Mobile robot, simultaneous localization and mapping, point-line fusion, reprojection error

Introduction
Simultaneous localization and mapping (SLAM) is an extensively researched topic in robotics¹ and has been widely applied in service robot, autonomous driving, Unmanned Aerial Vehicle (UAV), virtual reality, and other fields.²-⁴ The SLAM system mainly contains laser-based and vision-based methods according to its sensor type. In recent years, the theoretic research of laser-based SLAM has achieved extraordinary results in localization,⁵ mapping, autonomous navigation, and independent exploration.⁶ Meanwhile, visual information has the advantage of much quantity information, low cost, and intuitive effects, visual simultaneous localization and mapping (vSLAM) has gradually become the hot research field.⁷

Visual odometry is the process of determining the position and orientation of a robot by analyzing the associated camera images. It has been used in a wide variety of robotic applications. According to its different principles, there are mainly feature-based method and direct method, such as the line segment detector

¹ Key Laboratory of Knowledge Engineering with Big Data, Ministry of Education, Hefei University of Technology, Anhui, China
² School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, Anhui, China

Corresponding author:
Baofu Fang, Key Laboratory of Knowledge Engineering with Big Data, Ministry of Education, Hefei University of Technology, 193 Tunxi Road, AH 230009, China; School of Computer Science and Information Engineering, Hefei University of Technology, 193 Tunxi Road, Hefei, Anhui 230009, China.
Email: fangbf@hfut.edu.cn
(LSD)-SLAM proposed by Engel et al., which is able to build large-scale semi-dense maps, using direct methods instead of bundle adjustment over features. It achieves the reconstruction of semi-dense scene on standard Central Processing Unit (CPU), without Graphic Processing Unit (GPU) acceleration, and guarantees the final stability and real-time, but it still needs to rely on the feature point method for loop detection. Since the direct method is extremely sensitive to illumination and it is difficult to satisfy the strong assumption that the gradation is invariant, the research of feature-based SLAM has become popular currently. In recent years, optimization-based approaches have emerged endlessly due to its superior accuracy per computational unit as compared with filtering-based approaches. The first real-time application was the visual odometry work of Mouragnon et al., followed by the ground-breaking SLAM work of Klein and Murray known as parallel tracking and mapping.

Many algorithms can be performed with only a monocular camera. However, depth information cannot be directly observed, which results in a lack of the scale. In addition, multi-view or filtering techniques are required to produce an initial map, and the pure rotation problem cannot be effectively solved. In recent years, various low-priced depth cameras have been launched, such as Microsoft’s Kinect, Intel’s RealSense, and Asus’s Xtion, which make up the mechanism of loop closing, which can effectively prevent the error accumulated in the loop.

Gomez-Ojeda et al. propose a vSLAM method based on a stereo camera that combines points and line segments to work in a wider range of scenes, especially in scenes where point features are scarce or not evenly distributed. Pumarola et al. propose a visual SLAM based on a monocular camera, which improves the robustness of the system by processing points and lines simultaneously. In addition, it can complete the initialization of the map by processing the lines in three consecutive frames.

The LSD algorithm proposed by Rafael Grompone et al. can obtain detection results with sub-pixel precision in linear time, which can be applied to any digital image without debugging parameters. Its transform extraction speed is far faster than Hough speed, and it has strong robustness. Compared with the Mean-Standard Deviation Line Descriptor (MSDL) algorithm proposed by Zhang et al., the line band descriptor (LBD) algorithm proposed by Zhang et al. is fast and robust, which adds weight coefficients.

**Related work**

The KinectFusion algorithm proposed by Newcombe et al. merges the depth information measured by the sensor and uses the iterative closest point (ICP) to calculate the camera pose. Due to its lack of loop detection, it is often applied to small workplaces. Endres et al. propose using the ICP algorithm to compute frame-to-frame motion, and using the optimization-based method in the back end. Kerl et al. propose a dense vSLAM method that minimizes both photometric and depth error of all pixels, which can make better use of the information available in the image compared to feature-based methods. The ElasticFusion algorithm proposed by Whelan et al. is a dense 3-D reconstruction based on depth cameras which combines with relocation and is suitable for room-sized scene. Raul et al. propose Oriented FAST and Rotated BRIEF (ORB)-SLAM on the basis of their previous work, and the whole system is built around the ORB features, including the visual odometry and the visual vocabulary of loop closing. Although features such as Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF) are better in quality than ORB, their time consumption is difficult to meet the real-time requirements of vSLAM, and GPU acceleration is required. Compared with Harris, ORB has good rotation invariance and can use the pyramid to achieve scale invariance. The system also adds the mechanism of loop closing, which can effectively prevent the error accumulated in the loop.

In this work, we propose a point-line fusion method to deal with weak-matching objects in Red Green Blue - Depth (RGB-D) SLAM, where line features are joined in the feature extraction part. To solve the overlap and branch problem existing in the LSD algorithm, we design an adaptive line extraction method which enhances the reliability of the line features. The camera pose is optimized by minimizing the point-line reprojection error. In addition, we set up an optimization mechanism for the motion-jitter sequences, which can effectively improve tracking success rate and improve the tracking accuracy indirectly. Compared with point-based SLAM, the accuracy and robustness of PLF-SLAM are greatly improved.

The rest of this article is organized as follows. The second section discusses the related work, the third section gives the details of our proposal, the fourth section details the experimental results, and the fifth section presents the conclusions and the future work.
Adaptive line segment extraction

The time needed for feature extraction occupies a large part in the whole algorithm. Efficient feature extraction method is the premise to ensure the real-time performance of the SLAM system. LSD\textsuperscript{17} is a linear-time LSD, which is much faster than the Hough transform\textsuperscript{33} method and requires no parameter tuning. However, compared to extracting ORB features, it takes two to three times longer to extract line features by the LSD method, which will greatly affect the real-time performance of the algorithm. And some of the extracted lines are redundant and useless, resulting in lower reliability. We propose a new line extraction method based on LSD and define a line segment response value function as follows

\[ R_l = \sqrt{(x_{sp} - x_{ep})^2 + (y_{sp} - y_{ep})^2} / \text{Max}(L, W) \]  \hspace{1cm} (1)

where \(sp\) and \(ep\) are the two end points of the extracted line segment, \(L\) and \(W\) are the length and width of the image frame, respectively.

By setting the adaptive threshold, the line segment with response value less than \(\varepsilon_t\) will be eliminated, which increases the extraction speed. The line segments detected by the LSD method often have problems such as overlapping or sub-line segments, as shown in Figure 2.

Among them, Figure 2(a) is the original sample, Figure 2(b) is the line segment extracted by the original algorithm, and Figure 2(c) is the enlarged result of Figure 2(b). The line segment on the floor is mistakenly considered to be the overlap of two or more line segments, and this error occurs in most of the frames, which will affect the reliability of the matching. Therefore, we propose an improved method for this flaw and merge the overlapped line segments through appropriate conditions. First, the above situation is abstracted into a geometric problem, as shown in Figure 3.

In Figure 3, \(d_1\) is the distance from the midpoint on line \(l_1\) to line \(l_2\), we define the conditions for merging the line segments as follows

\[
\begin{cases}
|\text{Dir}(l_1) - \text{Dir}(l_2)| < \varepsilon_1 \\
 d_1 < \varepsilon_2
\end{cases}
\]  \hspace{1cm} (2)

The included angle of the two line segments should be smaller than \(\varepsilon_1\), where \(\text{Dir}(l)\) is the direction angle function of the line segment, and \(\varepsilon_1\) is the angle
experience threshold; $\varepsilon_2$ is the minimum distance threshold from the midpoint of line $l_1$ to line $l_2$. After satisfying formula (2), for case (a), the abscissa or ordinate of one endpoint in $l_1$ must be between the two endpoints of $l_2$, i.e.

$$x_c < x_b < x_d \quad \| \quad y_c < y_b < y_d$$

where $x$ and $y$ are the abscissa and ordinate of the end point of the line segment, respectively. For case (b), the shortest distance from end point $b$ to $c$ is less than the experience threshold $d_{\text{min}}$, i.e.

$$\text{dis}(b, c) < d_{\text{min}}$$

As described above, if the extracted line segments satisfy the above conditions, line segment merging is performed. We propose the least squares method for merging line segments and generate a 2-D point set according to the given line segments.

Point $l_1, l_2 = \{(x_i, y_i) | (x_i, y_i) \in l_1, l_2\}$

Suppose that the approximate line function of the merged line $l_{12}$ be $y_{12} = \varphi_{12}(x)$. To minimize the sum of squared deviation is minimized. The deviation formula is as follows

$$e_i = y_i - \varphi(x_i)$$

Let the fitting formula be

$$y = ax + b$$

Then the sum of squared deviations is as follows

$$e^2 = \sum_{i=1}^{n} (y_i - (ax_i + b))^2$$

Calculating partial derivatives for $a$ and $b$, respectively, we obtain the simultaneous equations

$$\begin{align*}
\left(\sum_{i=1}^{n} x_i^2\right) a + \left(\sum_{i=1}^{n} x_i\right) b &= \sum_{i=1}^{n} y_i x_i \\
\left(\sum_{i=1}^{n} x_i\right) a + nb &= \sum_{i=1}^{n} y_i
\end{align*}$$

Finally, the above formula is solved to obtain the linear function of the merged line segment.

**Line matching method**

Line features can be matched by comparing the similarity of LBD descriptors. The common feature matching methods contain the brute force match (BFM) algorithm and fast library for approximate nearest neighbors (FLANN) algorithm. FLANN can easily lead to matching failure when tracking in low-texture scenes, thus we use BFM in our work.

Compared with the point feature, the BFM algorithm has a higher error rate when matching line features. The main reasons are as follows:

(a) The matched lines have low similarity, and some of them are wrongly matched.
(b) Since the line features of the edge part of the image are often only partially matched in the image, it is difficult to use for the subsequent pose estimation later.
(c) In low-texture scenes, the extracted line features are not reliable enough.

Therefore, we filter the line features extracted by the BFM algorithm and reject the inaccurate matches. All matching pairs are needed to be filtered by the following conditions (Figure 4):

(a) $|\text{Dir}(l_1) - \text{Dir}(l_2)| <$ $\varepsilon_1, \varepsilon_2$. Since the motion between two adjacent frames is small, the angle of the extracted line segment does not change too much. When the difference between the direction angles of the matched line segments is greater than $\varepsilon_1, \varepsilon_2$, the match is regarded as a mismatch.

(b) $\frac{\text{Len}_{l_1} - \text{Len}_{l_2}}{\min\{\text{Len}_{l_1}, \text{Len}_{l_2}\}} < \lambda$, where Len is the length of the detected line segment in the image frame. The line features extracted by the LSD method are not
absolutely equal even in the case of correct matching. When the difference between the lengths of the two lines exceeds a certain range, the matching pair is eliminated to ensure the accuracy of the pose estimation later.

(c) Considering that the edge line features are often partially missing, we propose a method to remove the edge feature line matching pairs. The narrow area around the image is divided into edge areas, when the end point of the matched line falls into the area, the line feature is classified as edge feature and discarded.

In addition, the line feature mentioned above compares the similarity by comparing the Hamming distance between LBD descriptors, and we obtain the minimum descriptor distance by BFM. On this basis, when the minimum descriptor distance exceeds a certain range, the matched line will be discarded.

Point-line reprojection error

The optimization of camera pose first needs to find an error model of the points and lines. The variable of the error model is the camera pose and spatial coordinate of the features. The camera pose is calculated by minimizing the point-line reprojection error. In our work, the distance between the two end points of the spatial line projection and the detected line segments in the image is used as the reprojection error.

As shown in Figure 5, let $P, Q \in \mathbb{R}^3$ be the 3-D end points of the line segment, $p^d, q^d \in \mathbb{R}^2$ be their detected 2D end points. For normalization, let $p^d, q^d \in \mathbb{R}^3$ be their corresponding homogeneous coordinates. From the above, we can define the normalized line coefficient as

$$l = \frac{p^d \times q^d}{|p^d \times q^d|}$$  \hspace{1cm} (8)

After obtaining the normalized line coefficients, we define the point-line error $E_{pl}$ between $P$ and the detected 2D line segment $l$ as the distance from $p^d$ to $l$

$$E_{pl}(\theta, P, l) = (l)^T \pi(\theta, P)$$  \hspace{1cm} (9)

where $\pi(\theta, P)$ represents the projection of $P$ in the image, and $\theta$ represents the rotation matrix $R$ and the translation vector $t$ of the camera pose. Then the line reprojection error $E_l$ is defined as the sum of the point line error $E_{pl}$

$$E_l(\theta, P, Q, l) = E_{pl}^2(\theta, P, l) + E_{pl}^2(\theta, Q, l)$$  \hspace{1cm} (10)

Since there is accumulated errors in camera pose obtained in visual odometry, and the noise cannot be eliminated, the projection of 3-D points and lines on the image is necessarily different from the points and lines detected on the image, so we define the following error formula

$$\theta_{pl} = \arg\min_{\theta} \left\{ \lambda_i \sum_i E_i(\theta, P^i, Q^i, l^i) + \sum_j (x_j - \pi(\theta, X_j)) \right\}$$  \hspace{1cm} (11)

where $i$ is the $i$th line feature, $j$ is the $j$th point feature, and $\lambda_i$ is dynamically adjusted according to the response value of each line feature response value. In our work, the above-mentioned error formula is used to construct a comprehensive model of the point line error and then minimize the error and find the optimal camera pose. The pseudo-code of the whole algorithm is provided in Algorithm 1.

![Figure 5. (a and b) Line reprojection error model.](image_url)
In the tracking process, the camera often moves fast or shakes, which results in blurred images. As shown in Figure 6, it is easy to cause the failure of feature matching, and finally, the tracking will be lost. Tracking loss requires relocation to make adjustments. If the relocation fails after the loss, it will lead to tracking failure. The main reason for the tracking loss is that the texture of the image is not obvious or the feature difference is too large due to motion jittering, which makes it impossible to match correctly.

There are two solutions to the above situation:

(a) Under the premise of obtaining the blur kernel, the image with jittering is applied to the linear convolution, which makes the original blurred image clearer, but calculating the blur kernel is a major difficulty.

(b) Using the Gauss filter to deal with the image captured by the camera, which can reduce the difference between adjacent images and effectively improve the success rate of feature matching. This feature can effectively reduce the probability of tracking loss.

In this article, we propose a method to select motion-blurred images automatically. In tracking, we pre-match the features of the current image and the next image to calculate the success rate of the tracking in advance. When the matching degree of the two images is lower than the given threshold, Gaussian blur is used to reduce the image noise and detail, which reduces the difference between the two images and improves the matching success rate later.

When the size of blur kernel increases in a certain range, the image blur degree will also increase, so the number of feature matching will also increase. But beyond this range, the difference between adjacent images will be larger than the original image, resulting in matching failure. We propose the following improvement methods in Algorithm 2.

---

**Algorithm 1.** Point-line Optimization Algorithm of PLF-SLAM

**Input:** RGB and depth image  
**Output:** Camera pose (R, t)

1. $Imgray \leftarrow Rgb2gray(RGB)$  
2. $CurF \leftarrow ConstructF(Imgray, Depth)$  
3. $MotionJitteringProcess(CurF, nextF)$  
4. $Points \leftarrow ORBExtractor(Imgray, ORB)$  
5. $Lines \leftarrow LineExtractor(Imgray, LSD)$

6. for $i = 1$ to $Nines$ do
7. 
8. if $R_l(line_i) < low_set$ then  
9. 
10. end if
11. end for
12. for $i = 1$ to $Nines$ do  
13. 
14. if $|\text{Dir}(l_i) - \text{Dir}(l_j)| > \varepsilon_1$ \& \& $d_1 > \varepsilon_2$ then  
15. continue
16. end if
17. if $sp_j < x_i < ep_j \& \& sp_j < y_i < ep_j$ then  
18. 
19. if min_dis(point$i$, point$j$) < $d_{min}$ then  
20. 
21. end if
22. end for
23. end for
24. while isTracking() do
25. 
26. MinReprojection(CurF, RefKeyF)
27. 
28. $\theta_{pl} = \arg \min_{\theta} \left\{ \lambda \sum_{i} E_{i}(\theta, P_i, Q_i, l_i) + \sum_{j} (x_j - \pi(\theta, X_j)) \right\}$
29. end for
30. end while

---

**Motion jittering**

Figure 6. (a and b) Blurry image caused by motion jitter.
Through the improvement of the above method, the tracking success rate is obviously improved, and the error of pose estimation is indirectly reduced, which improves the robustness of the SLAM system.

Experiments and results
We experiment and evaluate the performance of our method on the Technische Universität München (TUM)\textsuperscript{34} dataset, which provides several image sequences with accurate ground truths obtained by external motion capture systems. These image sequences also include specific scenes such as normal motion, fast motion, low texture, no structure, etc. We perform our experiments on the computer which has an Intel core i5 (@2.6 GHZ) processor and 8 GB RAM with no GPU acceleration to reduce the running cost of SLAM.

Tracking accuracy
The evaluation of SLAM system performance mainly includes localization accuracy and mapping accuracy, and the latter mainly depends on the former. Therefore, we mainly focus on the localization accuracy of the algorithm in this article. To verify the effectiveness and superiority of the proposed method, we compare it with ORB-SLAM2.

In this article, we select different types of image sequences and carry out multiple groups of the same experiments on each sequence and then take the average value to avoid contingency. Finally, we calculate the root mean square error (RMSE) between the estimated pose and the true pose. We obtain the average of these RMSE data to effectively compare the results. A variety of image sequences such as fr1/360 and fr1/desk are selected as experimental comparison sequences and compared with ORB-SLAM2. The results are presented in Table 1.

Table 1 shows the absolute trajectory error result of different methods for tracking the above dataset sequences separately. From the experimental results, it can be known that the RMSE data obtained by our method is lower than that of ORB-SLAM2 at different degrees in most datasets. As shown in Figure 7(a) and (b), the tracking results of PLF-SLAM are significantly better than ORB-SLAM2 on the fr1/room sequence, which is benefited from the adaptive line segment extraction method proposed in this article. When the number of point features is insufficient, the line features can be simultaneously extracted to participate in the tracking, which makes up for the lack of point features.

In addition, we also compare our method with ORB-SLAM2 based on the original LSD method. It can be seen from the experimental data in the table that, when we use the adaptive line segment extraction method in this article, most of the trajectory errors are significantly reduced, and the pose estimation is more accurate. The reason is that the line features extracted by the original LSD method have a series of situation such as overlap, segmentation. So that the difference in line features extracted between the adjacent images is large, resulting in poor matching. In our work, we deal with the overlap and segmentation, and more stringent conditions are imposed in the matching, which makes the line matching more reliable and lays a good foundation for the back-end optimization.

### Table 1. Results of the comparisons in the TUM data set.

| Sequence        | ORB-SLAM2 | ORB-SLAM2 with LSD | PL-F SLAM | Reduce (%) |
|-----------------|-----------|--------------------|-----------|------------|
| Fr1/360         | 19.60     | 18.7               | 17.03     | 13.11      |
| Fr1/desk        | 1.88      | 1.97               | 1.54      | 18.09      |
| Fr1/floor       | 1.57      | 1.76               | 1.58      | -0.60      |
| Fr1/room        | 7.23      | 6.81               | 5.78      | 20.06      |
| Fr1/xyz         | 0.99      | 0.95               | 0.94      | 5.05       |
| Fr2/desk        | 0.93      | 0.86               | 0.98      | -6.45      |
| Fr2/large_loop  | 21.17     | 23.77              | 20.99     | 0.85       |
| Fr2/pioneer_slam3 | 8.68    | 9.33               | 6.60      | 23.96      |
| Fr2/xyz         | 0.38      | 0.41               | 0.39      | -2.63      |
| Fr3/office_household | 1.13  | 1.01               | 0.93      | 17.70      |

LSD: line segment detector; SLAM: simultaneous localization and mapping. The bold values highlight the experimental results of the method in this paper to show the advantages of the paper.
Motion jitter evaluation

In the second section, when the camera is quickly shaken, the images taken will often affect feature matching due to the motion blur problem, so it is difficult to calculate the camera pose between the current frame and the next frame in the tracking process. When that happens, we need to perform the relocalization method to find the camera pose, otherwise, the entire tracking process will fail.

When the above TUM sequences are tested, it is found that there are different degrees of motion jitter in multiple sets of sequences. In the fr1/desk sequence, ORB-SLAM2 typically has tracking loss during the 170th and 171st images. The reason is that the image of the previous image is drifted due to the camera’s fast jitter, and the difference between the current frame and the next frame is too huge, so that there are not enough matching pairs, resulting in
tracking failure. Although the camera pose can be restored by relocalization, the previous trajectory will be lost, which will inevitably increase the overall trajectory error, as is the fr2/large_loop sequence.

We found two images of the sequence that caused tracking loss and then extracted and matched the features separately. We find that there are few valid matches. For example, the matches of jitter frames in the fr1/desk sequence are only 76, which are difficult to meet the needs of tracking. After our method is used, the matches grow to 117 pairs, and the matching success rate is obviously improved. This is due to the optimization of the jitter images. When the matches are sufficient, the tracking success rate will be greatly improved.

Figure 8 shows the matching of two images in the fr1/desk sequence, where (a) is the matching image before processing and (b) is the processed matching image. It is easy to find that the feature matching after processing by this method is better.

Figure 9(a) shows the comparison between the estimated trajectory and the actual trajectory in ORB-SLAM2. We can find that the tracking in the red rectangle box has been lost. Figure 9(b) shows the comparison between the estimated trajectory and the actual trajectory in PLF-SLAM. It can be seen that the tracking effect of this method in this sequence is obviously better than the former.

Table 2 shows the comparison of the two methods in tracking success rate, which includes multiple groups of image sequences with good tracking and bad tracking. From the table, we can see that our method has good tracking results for most image sequences. Compared with ORB-SLAM2, the tracking success rate has been improved to some extent. This proves that our method improves the robustness of slam system to a certain extent.

### Conclusion

In this article, a point-line fusion SLAM method based on depth camera is presented. Aiming at the unreliability of SLAM based on point feature in low-texture environment, a new line segment extraction, and matching method are proposed, which solve the overlap and branch problem of the original LSD method and improve the reliability of the line matching and robustness. For the motion jitter problem often encountered in tracking, a method of autonomously selecting and optimizing the motion jitter frame is proposed, which greatly improves the success rate of tracking, and the method does not require additional sensor support. We have carried out many experimental evaluations on the TUM dataset, and most of the experimental results have been greatly improved compared with ORB-SLAM2. We have specifically verified the image sequence with motion blur, and the experimental results prove the article. The experimental results verify the rationality and effectiveness. In the future, we will optimize the tracking process and improve the real-time performance of the proposed method by detecting the current tracking environment to choose when to join the line feature estimation. In addition, we will combine the
depth sensor to build a dense point cloud map in real time, and further build an octree-based grid map to enhance the intuitiveness and reusability of mapping for subsequent navigation work and other corresponding work.

Acknowledgments
We would like to express our gratitude to all those who helped us during the writing of this thesis. We acknowledge the help of Professor Wang Hao, who has offered suggestions in academic studies. We owe a special debt of gratitude to Han Jianying, a professor in the same research field. Finally, thanks to all the partners of the lab for their company and support.

Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Natural Science Foundation of Anhui Province [Grant No. 1708085MF146], the university synergy Innovation Program of Anhui Province [GXXT-2019-003], the special fund for basic scientific research in central colleges and universities [Grant No. ACAIM190102], and the Project of Innovation Team of Ministry of Education of China [Grant No. IRT17R32].

ORCID ID
Baofu Fang https://orcid.org/0000-0002-1742-0250

References
1. Garcia-Fidalgo E and Ortiz A. Vision-based topological mapping and localization methods: a survey. Robot Auton Syst 2015; 64: 1–20.
2. Shi C, Lan X, and Wang Y. Motion planning for unmanned vehicle based on hybrid deep learning. In: 2017 International conference on security, pattern analysis, and cybernetics (SPAC), Shenzhen, China, 15–18 December 2017, pp. 473–478. Shenzhen: IEEE.
3. Zhang J, Liu R, Yin K, et al. Intelligent collaborative localization among air-ground robots for industrial environment perception. IEEE T Ind Electron 2019; 66(12): 9673–9681.
4. Von Stumberg L, Usenko V, Engel J, et al. From monocular SLAM to autonomous drone exploration. In: 2017 European conference on mobile robots (ECMR), Paris, France, 6–8 September 2017, pp. 1–8. Paris: IEEE.
5. Wang Z, Zhang J, Chen S, et al. Robust high accuracy visual-inertial-laser slam system. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 4–8 November 2019.
6. Fang B, Ding J, and Wang Z. Autonomous robotic exploration based on frontier point optimization and multistep path planning. IEEE Access 2019; 7: 46104–46113.
7. Quan M, Piao S, and Li G. An overview of visual SLAM. CAAI T Intell Syst 2016; 11(6): 768–776.
8. Engel J, Schöps T, and Cremers D. LSD-SLAM: Large-scale direct monocular SLAM. In: European conference on computer vision. Cham: Springer, 2014, pp. 834–849.
9. Liang M, Min H, and Luo R. Graph-based SLAM: a survey. Jiqiren/Robot 2013; 35(4): 500.
10. Liu R, Zhang J, Chen S, et al. Towards SLAM-based outdoor localization using poor GPS and 2.5D building models. In: 2019 IEEE international symposium on mixed and augmented reality, Beijing, China, 14–18 October 2019, pp. 1–7. Beijing: IEEE.
11. Mouragnon E, Lhuillier M, Dhome M, et al. Real time localization and 3d reconstruction. In: 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR’06), New York, NY, USA, 17–22 June 2006, Vol. 1, pp. 363–370. New York: IEEE.
12. Klein G and Murray D. Parallel tracking and mapping for small AR workspaces. In: Proceedings of the 2007 6th IEEE and ACM international symposium on mixed and augmented reality, Washington, DC, 13–16 November 2007, pp. 1–10. Nara: IEEE Computer Society.
13. Zhang Z. Microsoft Kinect sensor and its effect. IEEE Multimedia 2012; 19(2): 4–10.
14. Keselman L, Iselin Woodfill J, Grunnet-Jepsen A, et al. Intel realsense stereooscopic depth cameras. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, Honolulu, Hawaii, USA, 21–26 July 2017, pp. 1–10. Honolulu: IEEE.
15. Yu Q, Liang J, Xiao J, et al. A novel perspective invariant feature transform for RGB-D images. Comput Vis Image Und 2018; 167: 109–120.
16. Zhang J, Gui M, Wang Q, et al. Hierarchical topic model based object association for semantic SLAM. IEEE Trans Vis Comput Gr 2019; 25(11): 3052–3062.
17. Von Gioi RG, Jakubowicz J, Morel JM, et al. LSD: a line segment detector. Image Process On Line 2012; 2: 35–55.
18. Newcombe RA, Izadi S, Hilliges O, et al. Kinectfusion: real-time dense surface mapping and tracking. ISMAR 2011; 11(2011): 127–136.
19. Rusinkiewicz S and Levoy M. Efficient variants of the ICP algorithm. In: International conference on 3-d digital imaging and modeling, 28 May–1 June 2001, Quebec, Canada, Vol. 1, pp. 145–152. Quebec: IEEE.
20. Endres F, Hess J, Sturm J, et al. 3-D Mapping with an RGB-D camera. IEEE Trans Robot 2014; 30(1): 177–187.
21. Kerl C, Sturm J, and Cremers D. Dense visual SLAM for RGB-D cameras: 2013 IEEE/RSJ international conference on intelligent robots and systems, 3–7 November 2013, Tokyo, Japan, pp. 2100–2106. Tokyo: IEEE.
22. Whelan T, Salas-Moreno RF, Glocker B, et al. ElasticFusion: real-time dense SLAM and light source estimation. Int J Robot Res 2016; 35(14): 1697–1716.
23. Mur-Artal R and Tardos JD. ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Trans Robot 2016; PP(99): 1255–1262.
24. Mur-Artal R, Montiel JMM, and Tardos JD. ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans Robot 2015; 31(5): 1147–1163.
25. Rublee E, Rabaud V, Konolige K, et al. ORB: an efficient alternative to SIFT or SURF. In: 2011 International conference on computer vision, Barcelona, Spain, 6–13 November 2011, Vol. 11, p. 2. IEEE
26. Lowe DG. Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 2004; 60(2): 91–110.
27. Bay H, Tuytelaars T, and Van Gool L. Surf: speeded up robust features. In: *European conference on computer vision*. Berlin: Springer, 2006, pp. 404–417.
28. Shi J. Good features to track. In: 1994 proceedings of IEEE conference on computer vision and pattern recognition (ed Okamura A), Seattle, Washington, 21–23 June 1994, pp. 593–600. Seattle, USA: IEEE
29. Gomez-Ojeda R, Moreno FA, Zuñiga-Noél D, et al. PL-slam: a stereo slam system through the combination of points and line segments. *IEEE Trans Robo* 2019; 35(3): 734–746.
30. Pumarola A, Vakhitov A, Agudo A, et al. PL-SLAM: real-time monocular visual SLAM with points and lines. In: 2017 IEEE international conference on robotics and automation (ICRA) 2017, pp. 4503–4508. Singapore: IEEE.
31. Wang Z, Wu F, and Hu Z. MSLD: a robust descriptor for line matching. *Pattern Recognit* 2009; 42(5): 941–953.
32. Zhang L and Koch R. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. *J Vis Commun Image R* 2013; 24(7): 794–805.
33. Ballard D H. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognit* 1981; 13(2): 111–122.
34. Sturm J, Engelhard N, Endres F, et al. A benchmark for the evaluation of RGB-D SLAM systems. In: 2012 IEEE/RSJ international conference on intelligent robots and systems, Algarve, Portugal, 7–12 October 2012, pp. 573–580. Vilamoura: IEEE.