Postprint of Iterative Data Association Algorithm in Line-Feature-Based Monocular SLAM

Authors: Xinyu Wei, Huang Jun, Yang Xiaofei, Peng Junjie

Date: 2018-11-29T00:00:00+00:00

Abstract

To address the data association problem in monocular SLAM (Simultaneous Localization and Mapping) based on line features, an iterative data association algorithm based on verification of endpoint patches of line segments is proposed. The algorithm obtains nearest neighbor association pairs for line features based on two metrics: approximate collinearity and approximate endpoint coincidence, employs a directional matching verification mechanism based on endpoint patches of line segments to eliminate outliers from the nearest neighbor association pairs, and simultaneously improves data association accuracy through iterative processing by comprehensively utilizing both geometric constraints between line features and image similarity constraints. The proposed algorithm was tested on public datasets, and comparative experimental results with existing line feature data association algorithms demonstrate that it achieves favorable performance in both the number of line feature association pairs and association accuracy while satisfying system real-time requirements.

Full Text

Preamble

Vol. 37 No. 1
Application Research of Computers
ChinaXiv Cooperative Journal

Iterative Data Association Algorithm for Line-Based Monocular SLAM

Wei Xinyu[1,2], Huang Jun[2], Yang Xiaofei[2], Peng Junjie[1]
(1. School of Computer Engineering & Science, Shanghai University, Shanghai 200444, China;
2. Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai 201210, China)

Abstract: Aiming at the data association problem in line-based monocular SLAM (Simultaneous Localization and Mapping), this paper proposes an iterative data association algorithm based on endpoint patch verification. The algorithm obtains nearest-neighbor association pairs for line features according to two metrics: approximate collinearity and approximate endpoint coincidence. It then employs a directional matching confirmation mechanism based on line endpoint patches to eliminate erroneous pairs from the nearest-neighbor associations, while using an iterative processing approach to improve data association accuracy. The algorithm comprehensively utilizes both geometric constraints between line features and image similarity constraints. Tests on public datasets and comparative experiments with existing line feature data association algorithms demonstrate that the proposed algorithm achieves good performance in both the number of line feature association pairs and association accuracy while satisfying real-time system requirements.

Keywords: simultaneous localization and mapping; data association; line segment feature; iterative matching; feature matching

---

0 Introduction

With the rapid development of augmented reality, UAV navigation, and related fields, visual SLAM systems have advanced significantly in recent years. Among these, feature-based visual SLAM systems have demonstrated excellent performance in accuracy and robustness, giving rise to numerous feature-based visual SLAM systems [1,2]. The most commonly used features in these systems are point features. However, point-based SLAM systems suffer from a scarcity of features in texture-sparse environments such as indoor man-made environments and corridors. Moreover, since points have zero dimension in a mathematical sense, 3D point-based maps cannot represent multi-level structural information, often requiring additional extraction of planes, lines, and other features for applications like augmented reality. Some researchers have attempted to employ alternative features in visual SLAM systems, with line-based systems being the mainstream approach [3–6]. This is because, on one hand, 3D line-based maps can provide richer environmental structural information; on the other hand, texture-sparse scenes (such as corridors and empty rooms) commonly encountered in visual SLAM applications typically contain abundant structural features with numerous extractable line features.

Compared with point features, the development of line-based SLAM has been relatively slow. Existing monocular line-based SLAM systems all have certain limitations: some have special requirements for the application environment [4,7,8], while others can only handle scenes with extremely sparse line segments [3,5,9]. An important reason for these problems lies in the inherent difficulties of data association in current monocular line-based SLAM.

Data association is one of the major challenges in SLAM and has a significant impact on the robustness and stability of SLAM systems. It refers to the process of establishing correspondences between map data and current sensor input data in SLAM systems. In visual SLAM systems, the data association problem for point features has been thoroughly studied. However, due to issues inherent to line features such as unstable endpoints and over-segmentation, data association for line features requires further investigation.

Currently, there is no universal data association method for monocular line-based SLAM, but existing algorithms can be roughly categorized into three types: The first type projects 3D line segments from the map onto the 2D image plane using the current camera pose prediction, then obtains 3D-2D line segment matching pairs using nearest-neighbor matching [4]. The second type uses image patches (Patch) around 2D line segment endpoints or other sampling points [8,9] and line segment descriptors [10] to obtain associations, comparing the line segment descriptors or sampling point patches of already-associated 2D line segments with those of candidate 2D line segments to obtain association pairs, with MSLD [11] and LBD [12] being the primary line segment descriptors. The third type combines the first two methods [3,5], using nearest-neighbor matching to obtain a candidate set of 2D line segments, then applying endpoint patch or line descriptor-based methods to obtain final matching results.

Among these, the first type is the fastest but suffers from excessively high error rates in line-dense scenarios or when pose prediction errors are large. Moreover, determining the nearest-neighbor line segment remains an unsolved problem. The second type offers higher accuracy but involves greater computational cost. Additionally, due to the poor repeatability of line segment endpoints obtained by current line feature extraction algorithms, relying solely on image features makes it difficult to obtain sufficient matching pairs. While the third type improves upon the problems of single methods, existing implementations still involve substantial computation, and the number of correct matching pairs obtained remains small when prediction pose errors are large.

This paper investigates the association problem between 3D line features in the map and 2D line segments in images for monocular line-based SLAM, proposing an iterative line feature association algorithm based on line endpoint patch verification. The algorithm integrates nearest-neighbor matching and line endpoint patch matching, improving the nearest-neighbor line segment definition in nearest-neighbor matching and the patch search strategy in patch matching. It reduces the high error rate of nearest-neighbor matching methods through patch verification and addresses the problem of high input data noise in association algorithms using an iterative optimization framework.

---

1 Problem Definition

The application scenario for the data association problem discussed in this paper is a monocular line-based SLAM system using a graph optimization framework.

We define the data association problem in monocular line-based SLAM systems as follows: Given a 3D line feature set $\mathcal{G} = \{g_1, g_2, ..., g_n\}$ and a 2D line segment set $\mathcal{L}_c = \{l_1, l_2, ..., l_m\}$ in the current frame, find correct matching pairs between these two sets.

In SLAM systems, the data association module' s primary function is to provide input data for optimizing camera pose and 3D map features. We represent the data association algorithm results as $\{(g_i, l_j)\}$, where each specific association pair (g_i, l_j) has an association distance $d(g_i, l_j, \mathcal{T}_c, \mathcal{K})$, with $\mathcal{T}_c$ being the current frame pose and $\mathcal{K}$ the camera intrinsic matrix. We represent the keyframe pose set as $\{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_m\}$. The data association results are used in the following two optimization processes in SLAM systems:

$$\mathcal{T}_c^* = \arg\min_{\mathcal{T}c} \sum_{g_i \in \mathcal{G}} d(g_i, l_j, \mathcal{T}_c, \mathcal{K}) \tag{1}$$

$$(\mathcal{T}^*, \mathcal{G}^*) = \arg\min_{\mathcal{T},\mathcal{G}} \sum_{\mathcal{T}i \in \mathcal{T}} \sum_{g_j \in \mathcal{G}} d(g_j, l_i, \mathcal{T}_i, \mathcal{K}) \tag{2}$$

Equation (1) optimizes the current camera pose, while equation (2) optimizes keyframe poses and 3D line features in the map. The principles of these two optimization processes are basically the same. Without loss of generality, the following sections describe the data association algorithm using the process of optimizing the current camera pose as an example.

---

2 Iterative Line Feature Association Algorithm Based on Patch Verification

The iterative line feature association algorithm based on patch verification first obtains candidate matching pairs through nearest-neighbor line definition, then uses a fixed-direction image patch search strategy to eliminate false matches. Subsequently, it optimizes the current camera pose using the obtained matching pairs, and finally iterates the above process until optimization converges. The following sections detail each step of the algorithm.

2.1 Initial Data Association Based on Nearest-Neighbor Matching

Obtaining initial association results involves two steps: narrowing the candidate association set and determining the nearest-neighbor matching object.

a) Narrowing the candidate association set. In nearest-neighbor matching, 3D line features are first projected onto the current camera plane to obtain 2D projection line segments, which are then used to calculate candidate association line segment sets in the image. To accelerate matching speed, this paper uses line segment angle and distance attributes to narrow the candidate association set.

The projection process employs the Plücker coordinate-based projection method from [13]. After obtaining the 2D projection line segment of a 3D line, the 2D line segments in the image are divided into multiple sets to improve search speed for candidate line segments corresponding to the projection line segment. This paper uses two attributes—the line segment angle and the distance from the line segment' s supporting line to the image center—to determine candidate matching sets. The line segment angle refers to the angle between the 2D line segment' s supporting line and the positive x-axis of the image coordinate system. 2D line segments are divided into n sets $\mathcal{A}_i$ $(i = 1, 2, ..., n)$ by angle (0-90 degrees), with a maximum angle difference of $90/n$ degrees within each set. They are also divided into m sets $\mathcal{D}_i$ $(i = 1, 2, ..., m)$ by the distance from the supporting line to the image center. In datasets with image dimensions of 640$×$480 pixels, the maximum distance difference within each set is 40 pixels. For a projection line segment l_p to be matched, we first obtain its angle value β and distance to the image center d. Using threshold values β_{max} and d_{max} for the corresponding attribute differences, we determine candidate sets $\mathcal{A}_t$ and $\mathcal{D}_t$. The final candidate 2D line segment set is $\mathcal{L}_t = \mathcal{A}_t \cap \mathcal{D}_t$. It should be noted that although two attributes are used for classification, the time complexity of the classification algorithm remains constant. Using only a single attribute would result in excessively large candidate matching sets in scenes with many co-linear line segments or uneven line segment distribution, which would significantly increase computation time in subsequent steps.

b) Determining the nearest-neighbor matching object. After obtaining the candidate 2D line segment set, we need to identify the nearest-neighbor matching line segment within this set. The key challenge lies in defining the nearest-neighbor line feature. Unlike point features, defining the nearest-neighbor line feature is a complex problem [14]. Since the distance between line segments is typically represented by the distance from segment endpoints to the other segment' s supporting line, this measurement is multi-dimensional and cannot intuitively determine the nearest-neighbor line segment. The nearest-neighbor line segment must be calculated based on the line segment' s own attributes.

We represent the 2D projection line segment of a 3D line segment as l_p, and the 2D line segments extracted from the image as l_a and l_b. The distance between 2D line segments is defined as:

$$d(l_a, l_b) = d_{P1}(l_a, l_b) + d_{P2}(l_a, l_b) + 0.5 \cdot d_{L1}(l_a, l_b) + 0.5 \cdot d_{L2}(l_a, l_b) \quad (3)$$

where $d_{P1}(l_a, l_b)$ and $d_{P2}(l_a, l_b)$ represent the distances from the two endpoints of line segment l_a to the supporting line of l_b, calculated using the same method as in [8]. Taking $d_{P1}(l_a, l_b)$ as an example, in equation (4), L_1 is an endpoint of line segment l_a (represented in homogeneous coordinates), and the supporting line of l_b is represented as $\mathcal{L}_b$. $d_{L1}(l_a, l_b)$ and $d_{L2}(l_a, l_b)$ represent the distances between corresponding endpoints of line segments l_a and l_b. $\bar{l}$ represents the average length of all 2D line segments in the current image plane, and l_a represents the length of line segment l_a. This weighting makes long line segments, which contribute more to the pose optimization process, easier to match.

This paper defines two proximity attributes for line segments: one is the collinearity attribute, quantified by the endpoint-to-line distance $d_P(l_a, l_b)$; the other is the position coincidence attribute, represented by the distance between corresponding endpoints $d_L(l_a, l_b)$. Typical line segment distance definitions generally take the form $d(l_a, l_b) = d_{P1}(l_a, l_b) + d_{P2}(l_a, l_b)$. In line-dense scenes, traditional definitions cannot distinguish between collinear or approximately collinear line segments, while using only endpoint distances cannot robustly handle unstable line segment endpoints.

After obtaining the candidate 2D line segment set, we calculate the distance $d(l_p, l_i)$ between each 2D line segment l_i in the set and the 3D line feature' s projection line segment. We select the 2D line segment with the minimum distance, and if this minimum distance is less than threshold d_{max}, we consider this 2D line segment as the associated feature of the 3D line feature L_i. In practice, this paper uses parameters $n = 18$, $m = 20$, $\beta_{max} = 6$ degrees, and $d_{max} = 40$ pixels.

2.2 False Association Elimination Based on Line Endpoint Patches

Due to errors in 3D line segments and predicted camera poses in SLAM systems, the matching pairs obtained by nearest-neighbor matching inevitably contain false associations that severely affect the accuracy of current frame pose estimation. This paper employs a line endpoint image patch-based false match elimination strategy to handle these erroneous associations.

A patch is a square image block around a 2D point. Using patches near line segment endpoints or midpoints for line matching is a common practice [8,15]. Such methods can be summarized in the following steps (taking line segment endpoints as an example):

However, in the patch search stage, calculating the Zero-Mean Sum of Squared Differences (ZMSSD) between the candidate 2D line segment endpoint patch and the projection line segment endpoint patch may lead to elimination of many correct associations due to unstable line segment endpoints. To address this issue, this paper defines a "relaxation range" for line segment endpoints. The relaxation range is a line segment of length S pixels that is collinear with the original line segment and centered at the original endpoint, as shown in Figure 2 [Figure 2: see original paper]. Here, l is the original line segment, and the relaxation range for endpoint l^* is the line segment l^{**}. The length S is defined as:

$$S = \begin{cases} 10 & \text{if } l \geq 20 \\ \frac{l}{2} & \text{if } l < 20 \end{cases} \tag{4}$$

where l is the length of the original 2D line segment, with a maximum S value of 10 pixels. When l is too small, the value of S varies with l. The relaxation range makes the false elimination algorithm more robust, ensuring that correct association pairs are not erroneously eliminated when endpoint errors do not exceed the relaxation range.

This paper applies the line endpoint patch-based matching method to eliminate errors from nearest-neighbor matching results. The specific steps are:

a) Obtain the patch corresponding to the 3D line feature. First, select the keyframe F_b corresponding to the 3D line feature. The keyframe with observation angle closest to that of the 3D line feature in the current frame is selected as the best observation keyframe F_b. The observation angle is defined as the angle between the current frame' s camera principal axis and the plane passing through the camera optical center and the 3D line, as shown in Figure 1 [Figure 1: see original paper]. Here, z is the camera principal axis (the line passing through optical center C and perpendicular to the image plane), π is the plane passing through optical center C and 3D line L, and the observation angle is the angle between z and π. After obtaining the keyframe, two 8$×$8 image blocks are extracted from the endpoints of the 2D line segment l_b corresponding to L in that keyframe. The subsequent processing is identical to the patch extraction process in PTAM [1], with implementation details available in [1].

b) Within the relaxation range of the 3D line feature' s associated 2D line segment endpoints, perform patch search along the direction collinear with the relaxation range. If the ZMSSD value between the searched image patch and the original 3D line feature' s corresponding patch is less than the threshold, the association is confirmed as correct; otherwise, this false association is eliminated.

2.3 Iterative Association Algorithm

The iterative association algorithm uses the above nearest-neighbor matching method to obtain candidate matching pairs, then employs the line endpoint patch-based false association elimination algorithm to remove incorrect pairs. The current frame pose is optimized using the obtained associations, and the entire process is iterated until the matching error rate in the optimization process falls below a threshold.

In the pose optimization process, the objective function is formulated as equation (1). To avoid adverse effects from unstable line segment endpoints during optimization, the distance function in the objective function uses the line endpoint-to-line distance described in equation (3). This paper uses the LM algorithm in g2o [16] to implement camera pose optimization. The iteration termination criterion is that the inlier ratio in the camera pose optimization process exceeds 70% of the inlier ratio in the nearest keyframe, where inliers are defined as association pairs with distances less than the threshold after pose optimization. Considering the real-time requirement of SLAM systems, the maximum number of iterations is set to 3.

The iterative line feature association algorithm based on patch verification is presented as Algorithm 1.

Algorithm 1: Iterative Line Feature Association Algorithm Based on Patch Verification
Input: 2D line segment set $l = (l_1, l_2, ..., l_n)$, 3D line segment set $L = (L_1, L_2, ..., L_m)$, current camera pose P
Output: 3D-2D line segment matching pairs $\mathcal{G}$

```
function IterativeLineAssociation(l, L, P)
  repeat
    G $\leftarrow$ LineAssociation(l, L, P)
    P $\leftarrow$ OptimizePose(P, G)  // Optimize camera pose according to equation (1)
    if Inliers(P, G) then    // Determine loop termination based on inlier count
      break
  until maximum iterations reached
  return G
end function

function LineAssociation(l, L, P)
  result $\leftarrow$
  for i = 1 to m do
    lp $\leftarrow$ Project3DLine(l[i], P)    // Project 3D line segment to image plane
    ls $\leftarrow$ Get2DLineSet(lp, l)    // Obtain reduced candidate line segment set
    for each lb in ls do
      d $\leftarrow$ Distance(lp, lb)
      if d < dmax then
        pt $\leftarrow$ GetPatch(lp)
        fPatch $\leftarrow$ MatchPatch(pt, lb) // Perform patch verification as describe
        if fPatch then
          result $\leftarrow$ result  {(L[i], lb)}
        end if
      end if
    end for
  end for
  return result
end function
```

3 Experiments and Results Analysis

3.1 Experimental Design

To verify the accuracy and efficiency of the proposed iterative line feature association method, we conducted comparative experiments with similar algorithms on public datasets.

The experimental platform uses an Intel i7-3700 CPU at 3.4GHz with 4GB memory. The experimental data employs the TUM RGB-D dataset [19]. We selected four scenarios from the dataset as experimental data for the data association algorithms, shown in Figure 4 [Figure 4: see original paper]. From top to bottom and left to right, the typical image frames are from fr2_{xyz}, fr3_{{{long}}*{{office}}}, fr3*{{{str}}*{{tex}}}*{far}, and fr2_{rpy} scenes, which we name scenes 1-4 in this order.

The SLAM system used in experiments adopts the same framework as ORB-SLAM [2], replacing point feature operations with corresponding line feature operations and removing the loop closure detection module. The fast EDLine [17] algorithm is used to extract 2D line segments from images. We also modified the local map tracking module in the original system's tracking thread and added an association error evaluation module after it. The local map tracking module simultaneously uses three line feature association algorithms, feeding the data to be processed into all three algorithms in parallel. The algorithms are:

- **Algorithm 1:** The data association algorithm proposed in this paper
- **Algorithm 2:** Iterative nearest-neighbor matching algorithm, which removes the patch verification step from our algorithm and is used in [5]
- **Algorithm 3:** LBD descriptor-based association algorithm, used in [10]

Since the 3D map in SLAM systems is built and updated in real-time, the input data for data association algorithms inevitably contains error terms that are difficult for the association error evaluation module to completely eliminate. To accurately evaluate the accuracy and efficiency of data association algorithms in SLAM applications, the input data for all three algorithms is identical for each frame in our experimental scheme. The association error evaluation module evaluates the output data of the three line feature association algorithms using the same method.

We adopt the parameters from [18] as evaluation metrics, including matching line segment length ratio M, average projection error G, and sum of matched line segment lengths in minimum angular regions L. The specific calculation methods are:

$$M = \frac{\sum_{i,j} l_{ij}}{\sum_i l_i^I} \tag{5}$$

$$G = \frac{1}{N}\sum_{i,j}\frac{d(g_i, l_j)}{l^I_{ij}} \tag{6}$$

$$L = \sum_{k=1}^{3} l_k \tag{7}$$

where l_{ij} represents the length of 2D line segments in the association pairs obtained by the algorithm, l^I_i represents the length of 2D line segments extracted from the image, and l^I_{ij} represents the length of 2D projection line segments of 3D line segments that can be projected onto the current frame. The matching line segment length ratio M describes whether the algorithm can obtain sufficient matching pairs.

The average projection error G describes the average projection error of association pairs obtained by the line feature association algorithm. In equations, l_1, l_2, l_3 represent the sum of lengths of 2D line segments in different angular intervals within association pairs. This paper restricts 2D line segment angles to 0–180 degrees, dividing them into four intervals following similar processing methods in literature: the horizontal direction interval includes angles (0°, 22.5°) and (157.5°, 180°); the vertical direction interval includes (67.5°, 112.5°); remaining directions are divided into two intervals (22.5°, 67.5°) and (112.5°, 157.5°). The sum of lengths of associated line segments in each interval is denoted as l_i ($i = 1, 2, 3, 4$). After removing the maximum value l_{max}, the remaining three items are summed to obtain metric L, which represents the algorithm' s performance in handling line segments of different orientations.

Reference [18] sets corresponding thresholds for these three parameters, considering a frame' s data association to have failed when any metric cannot meet its threshold. This paper uses the same method to calculate the false association frame rate. Since [18] is based on a binocular camera system that can directly obtain depth information for 3D line segments, while our monocular system cannot directly acquire depth information and has relatively larger errors in 3D line segments, the input data for the data association algorithm contains significant errors. Therefore, this paper adopts more relaxed threshold settings for evaluation metrics G and M.

In addition to these metrics, we also statistically analyze the average computation time per frame and average number of association pairs obtained by different data association algorithms. To further analyze the algorithms, the experiment records the value of parameter G for each frame.

The experimental scheme is shown in Figure 3 [Figure 3: see original paper]. The local map tracking module takes as input the 3D line segment set in the local map and the 2D line segment set in the current frame, outputting matching pairs between 3D line features and 2D line segments. Four metrics for each data association algorithm are recorded.

3.2 Experimental Results and Analysis

Table 1 shows the results recorded by the association error evaluation module for each algorithm. The false association frame rate is the proportion of frames where any of the three evaluation metrics (G, M, L) exceeds its threshold. The average projection error is the average value of metric G across all frames.

From the perspective of false association frame rate and average projection error G, our algorithm achieves the lowest values across all four dataset scenarios, significantly improving association accuracy. In terms of association pair count, our algorithm yields slightly fewer pairs than Algorithm 2, but with the fewest false associations. Excessive false association pairs can significantly reduce SLAM system robustness. Regarding computation time, our algorithm runs within 30ms per frame in all four experimental scenes. The algorithm runs longer in scene 3 (which has the most 2D line features) and scene 2 (which contains loop closure). The closed-loop camera motion in scene 2 adds many occluded 3D line features to the input data for the line feature data association algorithm. However, due to the candidate association set reduction strategy described in Section 2.1 and the iterative optimization method in Section 2.3, the algorithm scale does not expand dramatically when the number of input 2D or 3D line features increases, ensuring real-time performance.

Among other algorithms, Algorithm 2 is the fastest but its false association frame rate varies significantly across different scenes, making it difficult to handle line-dense scenarios. Algorithm 3 is a commonly used LBD descriptor-based association method with relatively stable false association frame rates across scenes, but it is time-consuming and yields the fewest association pairs. As shown in single-frame results, this algorithm does not handle input data errors and texture-sparse line segments well.

Figure 5 [Figure 5: see original paper] shows system screenshots during experiments, where A, B, C, D represent the input image and association results from Algorithms 3, 2, and 1, respectively. In the association result images, thick solid lines represent 2D projection lines of 3D line segments whose association pair distances are within a certain threshold, considered correct associations; thick dashed lines represent 2D projection lines whose association pair distances exceed the threshold, considered false associations; thin solid lines represent 2D line segments extracted from the image. Black arrows in images B and C indicate regions for focused analysis.

Image A is from scene 2 of the dataset. Since Algorithm 3 relies solely on the LBD line descriptor, when the optimization results of 3D line segments in the map contain errors, Algorithm 3 struggles to eliminate such erroneous associations. The 2D projection line indicated by the arrow in image B is projected from a 3D line segment with large errors in the map, which Algorithm 3 fails to eliminate. Algorithm 2 detects association pairs based only on projection line distances, obtaining many false association pairs when line segments are densely distributed. The region indicated by the arrow in image C has dense line segments, where Algorithm 2 obtains numerous false associations, while our proposed algorithm can eliminate these false associations using patch verification.

To analyze the impact of different line feature association algorithms on SLAM system performance, we conducted experiments using local map tracking modules based on Algorithms 1, 2, and 3 in the same SLAM system. Due to Algorithm 2' s excessively high false association frame rate in scene 2, experiments were only performed on dataset scenes 1, 3, and 4. The relative errors of camera poses obtained by the SLAM system are shown in Table 2 .

Table 2: Relative Pose Error of Different Data Association Algorithms

Dataset Scene	Algorithm 1	Algorithm 2	Algorithm 3
Scene 1	RMSE (m/s): 0.052, Median (m/s): 0.045RMSE (deg/s): 2.29, Median (deg/s): 1.95	RMSE (m/s): 0.061, Median (m/s): 0.054RMSE (deg/s): 2.42, Median (deg/s): 2.17	RMSE (m/s): 0.058, Median (m/s): 0.051RMSE (deg/s): 2.35, Median (deg/s): 2.05
Scene 3	RMSE (m/s): 0.071, Median (m/s): 0.063RMSE (deg/s): 3.15, Median (deg/s): 2.87	RMSE (m/s): 0.082, Median (m/s): 0.074RMSE (deg/s): 3.52, Median (deg/s): 3.21	RMSE (m/s): 0.079, Median (m/s): 0.071RMSE (deg/s): 3.38, Median (deg/s): 3.05
Scene 4	RMSE (m/s): 0.048, Median (m/s): 0.041RMSE (deg/s): 1.95, Median (deg/s): 1.72	RMSE (m/s): 0.055, Median (m/s): 0.048RMSE (deg/s): 2.18, Median (deg/s): 1.95	RMSE (m/s): 0.053, Median (m/s): 0.046RMSE (deg/s): 2.07, Median (deg/s): 1.84

The pose trajectory relative errors in Table 2 demonstrate that our line feature association algorithm improves the overall performance of monocular line-based SLAM systems. Compared with the iterative nearest-neighbor association algorithm that uses only line feature geometric relationships and the LBD descriptor-based association algorithm, our proposed algorithm exhibits higher stability and accuracy.

4 Conclusion

Line feature data association is a crucial problem in monocular line-based SLAM. In recent years, monocular SLAM systems combining points and lines have attracted significant attention due to their accuracy and robustness advantages [10,15,20]. However, these fundamental problems related to line features limit the exploitation of line features' advantages. This paper proposes a novel line feature data association algorithm for challenging monocular line-based SLAM systems. The proposed algorithm integrates nearest-neighbor matching and false association elimination based on line endpoint image patches, using an iterative processing approach. Experiments on public datasets demonstrate that the proposed iterative line feature association algorithm achieves good performance in both association accuracy and association pair count.

Future work will further investigate joint association algorithms for point and line features to improve the performance of point-line fusion SLAM.

References

[1] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces [C]// Proc of IEEE and ACM International Symposium on Mixed and Augmented Reality. Piscataway, NJ: IEEE Press, 2007: 225–234.

[2] Mur-Artal R, Montiel J M M, Tardós J D. ORB-SLAM: a versatile and accurate monocular SLAM system [J]. IEEE Trans on Robotics, 2015, 31(5): 1147-1163.

[3] Perdices E, López L M, Cañas J M. LineSLAM: visual real time localization using lines and UKF [C]// Proc of the 1st Iberian Robotics Conference. 2014: 282-291.

[4] Zhang Guoxuan, Suh I H. SoF-SLAM: segments-on-floor-based monocular SLAM [C]// Proc of IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, NJ: IEEE Press, 2010: 2083-2088.

[5] Zhang Lilian, Koch R. Hand-held monocular SLAM based on line segments [C]// Proc of Machine Vision and Image Processing Conference. Piscataway, NJ: IEEE Press, 2011: 7-14.

[6] Wu Tao, Sun Fengchi, Yuan Jin, et al. An active SLAM approach based on line segment feature in indoor environment [J]. Robot, 2009, 31(2): 166-70.

[7] Zhang Guoxuan, Suh I H. A vertical and floor line-based monocular SLAM system for corridor environments [J]. International Journal of Control Automation & Systems, 2012, 10(3): 547-557.

[8] Zhou Huizhong, Zou Danping, Pei Ling, et al. StructSLAM: Visual SLAM With Building Structure Lines [J]. IEEE Trans on Vehicular Technology, 2015, 64(4): 1364-1375.

[9] Smith P, Reid I, Davison A. Real-Time Monocular SLAM with Straight Lines [C]// Proc of British Machine Vision Conference. Berlin: Springer, 2006: 17-26.

[10] Pumarola A, Vakhitov A, Agudo A, et al. PL-SLAM: Real-time monocular visual SLAM with points and lines [C]// Proc of IEEE International Conference on Robotics and Automation. Piscataway, NJ: IEEE Press, 2017: 4503-4508.

[11] Wang Zhiheng, Wu Fuchao, Hu Zhanyi. MSLD: a robust descriptor for line matching [J]. Pattern Recognition, 2009, 42(5): 941-53.

[12] Zhang Lilian, Koch R. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency [J]. Journal of Visual Communication & Image Representation, 2013, 24(7): 791-802.

[13] Zhang Lilian, Koch R. Structure and motion from line correspondences: Representation, projection, initialization and sparse bundle adjustment [J]. Journal of Visual Communication & Image Representation, 2014, 25(5): 763-768.

[14] Bartoli A, Sturm P. The 3D line motion matrix and alignment of line reconstructions [J]. International Journal of Computer Vision, 2004, 57(3): 195-210.

[15] Gomez-Ojeda R, Briales J, Gonzalez-Jimenez J. PL-SVO: semi-direct monocular visual odometry by combining points and line segments [C]// Proc of IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, NJ: IEEE Press, 2016: 4211-4216.

[16] Kümmerle R, Grisetti G, Strasdat H, et al. G2o: a general framework for graph optimization [C]// Proc of IEEE International Conference on Robotics and Automation. Piscataway, NJ: IEEE Press, 2011: 3607-3613.

[17] Akinlar C, Topal C. EDLines: A real-time line segment detector with a false detection control [J]. Pattern Recognition Letters, 2011, 32(13): 1633-42.

[18] Witt J, Weltin U. Robust stereo visual odometry using iterative closest multiple lines [C]// Proc of IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, NJ: IEEE Press, 2013: 4164-4170.

[19] Sturm J, Engelhard N, Endres F, et al. A benchmark for the evaluation of RGB-D SLAM systems [C]// Proc of IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, NJ: IEEE Press, 2012: 573-580.

[20] Yang Shichao, Scherer S. Direct monocular odometry using points and lines [C]// Proc of IEEE International Conference on Robotics and Automation. Piscataway, NJ: IEEE Press, 2017: 3871-3877.

Note: Figure translations are in progress. See original paper for figures.