ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0058

Research on visual SLAM algorithm for point-line feature fusion in structured Environment

Youquan Zhang[1, 2, a], Sanpeng Deng[1, 2, b] and Yuming Qi[1, 2, c]

[1] Institute of Robotics and Intelligent Equipment, Tianjin University of Technology and Education, Tianjin 300222, China;

[2] Tianjin Key Laboratory of Intelligent Robot Technology and Application, Tianjin 300222, China.

[a] zh_youquan@163.com, [b] 37003739@qq.com, [c] chiym@163.com

Abstract

The image preprocessing method of visual SLAM algorithm determines the quality of data association in SLAM. At present, the visual feature matching algorithm based on feature point method is relatively mature and applicable to a wide range of applications, but the algorithm based on point feature has a large amount of calculation and requires a large space for map storageIn structured artificial environment, only use the characteristic point matching is easy to appear a large number of false matching, this paper puts forward a kind of point, line feature fusion visual SLAM algorithm, in this environment the midline characteristics is rich, has a good illumination invariant and rotation invariance, affected by the environment is small, can good recovery in map building structural information in the environment Experiments on EUROC data set show that the localization accuracy of the proposed SLAM algorithm is significantly higher than that of the pure feature point method.

Keywords

Visual SLAM; Line feature extraction; Point and line feature fusion.

1. Introduction

Characteristic point by point feature extracting in the image and calculate the descriptor, match again, pose estimation, according to the correlation information algorithm is more complex Direct method directly using the image pixels with strong gradient, pose estimation error by minimizing the luminosity, although this part of the large amount of calculation, but it need not calculate descriptor, save a lot of time About SLAM scheme based on feature point method, In 2007, Davison A J proposed the first real-time monocular visual SLAM system Mono SLAM [2], which tracked sparse feature points at the front end through extended Kalman filter EKF [3] as the back end to achieve real-time positioning and mapping Mur-ArtalIn 2015, orB-SLAM was proposed by et al. [4], which is one of the most perfect systems in CURRENT SLAM systems. It innovatively uses tracking mapping loopback to detect the pose estimation of the three-thread system, which has good positioning accuracy and robustness and can ensure global consistency In July 2020, they opened source ORB-Slam3 [5], which greatly improved the performance compared to the previous version. Based on this, this paper integrates line features to make the system perform better in structured environment.

2. Description of line characteristics

When the camera moves, the number of point and line features extracted from the surrounding environment is constantly changing. Generally speaking, the number of line features extracted from

ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0058

the structured environment is large, and line features are less affected by light changes and have good Angle invariance. As a large-scale feature, line features can describe the surrounding environment in detail.

2.1 Line feature extraction

In the current Line feature detection algorithm, LSD[6](Line Segment Detector, LSD) is the most widely used algorithm, its accuracy can reach sub-pixel level Similar characteristics in its main idea is to specific area of the pixel to merge, and the detection error of certain constraints, ensure that the correct line segment detection LSD does not need to adjust the additional parameters extraction speed, line accuracy is high and the main direction of high performanceThe detection effect is shown in Figure 1.

Fig. 1 LSD algorithm detects line features

2.2 Line feature descriptor

Feature extraction is followed by feature description and matching. Lilian Zhang et al. improved MSLD descriptor [7] by adding Gaussian weight coefficient and proposed LBD descriptor [8], which has faster calculation speed, rotation invariance and better matching effect.

2.2.1 Line segments support strip representation of domains

An LBD descriptor is a detailed description of a Line Support Region (LSR). A Line Support Region is divided into a group of strips $\{B_1, B_2, ..., B_m\}$. The LSR region is divided into m strips, each of which has a pixel width of w. Similar to MSLD, it defines two directions to form a local 2D coordinate system to distinguish parallel lines with opposite gradient directions and keep the rotation of descriptors unchanged. Let's say the line is going in the direction of d_L, The orthogonal direction $d_\perp$ is defined as the direction clockwise and perpendicular to d_L, and the midpoint of the strip is defined as the origin of the local coordinate system. The global Gaussian function f_g acting on the $d_\perp$ direction of LSR makes it insensitive to small changes away from the line along the $d_\perp$ direction. By applying the local Gaussian function f_g to adjacent strips, the edge effect is reduced and the sudden change of descriptor is avoided when pixels move from one strip to the next.As shown in Figure 2, the line support domain is divided into five strips, each with a width of 3.

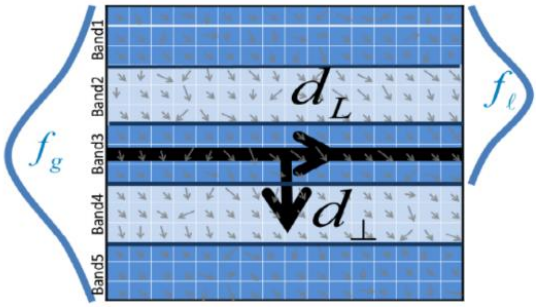


Fig. 2 The line support region of the LBD descriptor

ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0058

2.2.2 Construct the LBD strip descriptor

The descriptor BD_j of each strip consists of strips BD_{j-1} and BD_{j+1} of the two nearest adjacent rows. When the top strip B_1 and the bottom strip B_m are calculated, the parts outside the LSR region are not considered in the algorithm. The descriptors of all strips are combined, and the total descriptors are expressed as (1):

$$LBD = (BD_1^T, BD_2^T, ..., BD_m^T)^T \tag{1}$$

The line supports the gradient projection of each pixel in the domain into this local two-dimensional coordinate system as (2):

$$g' = (g^T \bullet d_\perp, g^T \bullet d_L)^T \triangleq (g'_{d_\perp}, g'_{d_L})^T \tag{2}$$

Where, g is the pixel gradient in the image.

The calculation of BD_j is essentially to describe the Gaussian weight in the four directions of upper, lower, left and right, and the k-th line of the strip is expressed as (3) :

$$\begin{aligned} v1_j^k &= \lambda \sum_{g'_{d_\perp}>0} g'_{d_\perp}, \quad v2_j^k = \lambda \sum_{g'_{d_\perp}<0} g'_{d_\perp} \\ v3_j^k &= \lambda \sum_{g'_{d_L}>0} g'_{d_L}, \quad v4_j^k = \lambda \sum_{g'_{d_L}<0} g'_{d_L} \end{aligned} \tag{3}$$

Where, $\lambda = f_g(k) f_l(k)$ is the Gaussian coefficient.

By accumulating gradient information of all rows associated with strip BD_j, a strip descriptor matrix BDM_j was constructed to represent the description information of the j-th strip.

$$BDM_j = \begin{pmatrix} v1_j^1 & v1_j^2 & ... & v1_j^n \\ v2_j^1 & v2_j^2 & ... & v2_j^n \\ v3_j^1 & v3_j^2 & ... & v3_j^n \\ v4_j^1 & v4_j^2 & ... & v4_j^n \end{pmatrix} \in R^{4n}, n = \begin{cases} 2w, j = 1 \text{ or } m \\ 3w, \text{else} \end{cases} \tag{4}$$

Calculating the mean variance M_j^T and the mean vector S_j^T of BDM_j yields the descriptor $BD_j = (M_j^T, S_j^T)^T$.Therefore, the general descriptor is:

$$LBD = (M_1^T, S_1^T, M_2^T, S_2^T, ..., M_m^T, S_m^T)^T \in R^{8m} \tag{5}$$

In order to improve the accuracy and speed of calculation, the floating-point descriptors are replaced by binary descriptors, and hamming distance is used as the distance measurement to judge the similarity quickly, thus improving the matching efficiency.

2.3 Algorithm detailed process

Let's define the total state vector to be $X = \left[S_1^T, S_2^T, ..., S_n^T, F_1^T, F_2^T, ..., F_m^T\right]^T$, Where, F_i^T is the coordinates of map points in dimension 3×1, S_i is the state vector of camera in dimension 6×1, and it is the pose of camera in frame i, expressed as:

$$S_i = \left[R_w^{c_i T}, {}^w P_{c_i}^T\right]^T \tag{6}$$

Where, $R_w^{c_i}$ represents the rotation of the world system and the camera system, and ${}^w P_{c_i}^T$ represents the position of the camera in the world system.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0058

Optimization is carried out by minimizing the residuals of all feature points and feature lines:

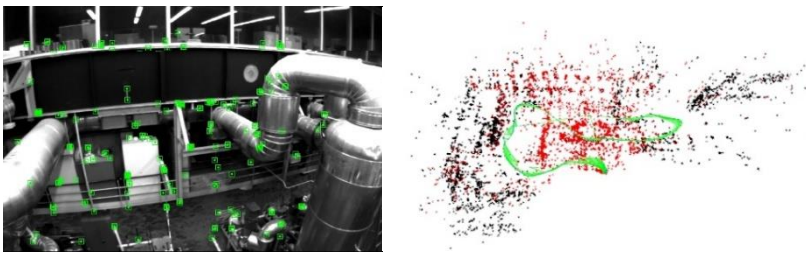
$$X^* = \arg\min_X ((f_i - \pi(S_i F_i))^T \sum_I (f_i - \pi(S_i F_i)) + (l_i - K_l H_{cw} L_w)^T \sum_K (l_i - K_l H_{cw} L_w)) \quad (7)$$

Where, f_i represents camera observation, $\pi(\bullet)$ represents projection from camera system to image plane, H_{cw} represents change matrix between world system and camera system, l_i represents observation of line features, L_w represents reprojection from line features to image plane, and $\sum$ represents covariance matrix.

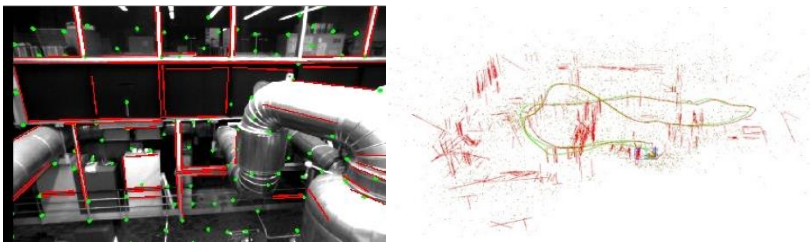
In feature location estimation, due to the influence of noise, the reprojection of feature points and feature lines will not coincide with their observed values in the image. The nonlinear optimization problem as shown in Equation (7) is constructed to solve the pose and position of the camera as well as the position of feature points and feature lines. The whole problem is solved by the Gauss-Newton method in G2O [9] library.

3. Experimental results and analysis

In order to quantitatively evaluate the visual front-end method of point-line fusion proposed in this paper, the proposed point-line feature fusion visual SLAM algorithm and THE ORB-Slam2 [10] algorithm were tested on the open source data set EUROC data set respectively. The image feature detection effect and graph construction of the two algorithms on the EUROC data set are shown in Figure 3.



(a) Location and mapping of ORB-SLAM2

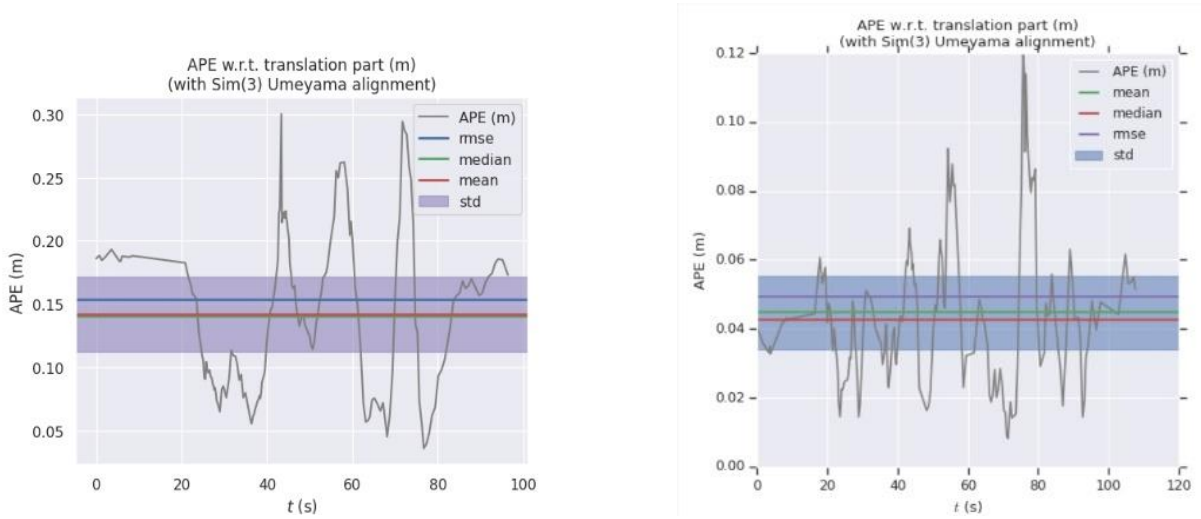


(b) Location and mapping of point-line fusion algorithm

Fig. 3 Two algorithms SLAM cases

In this paper, when comparing the effects of ORB-SLAM2 based on feature point method and the algorithm in this paper on the data set, the true value of sensor motion track provided by the data set is loaded, and the positioning error curves of the two algorithms are drawn, as shown in Figure 4.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202111_7(11).0058



(a) ORB-SLAM2 (b) point-line fusion algorithm

Fig. 4 Positioning error curve

4. Conclusion

In order to solve the problem of low localization accuracy of visual SLAM system based on feature point method in highly structured scenes, we propose a vision front-end tracking strategy that integrates point and line features. Experiments on EUROC data set show that the localization accuracy of the proposed algorithm is significantly higher than that of the pure feature point method The next step is to integrate IMU into the system proposed in this paper, so that the system can effectively cope with the scene of fast camera movement or low light.

References

[1] HAOMIN L, GUOFENG Z, HUJUN B. A survey of monocular simultaneous localization and mapping[J]. Journal of Computer-Aided Design & Computer Graphics, 2016.

[2] DAVISON A J, REID I D, MOLTON N D, et al. MonoSLAM: Real-time single camera SLAM[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2007, 29(6):1052.

[3] Bloesch M, Burri M, Omari S, et al.Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback[J].The International Journal of Robotics Research,2017, 36 (10): 1053-1072.

[4] MUR-ARTAL R, MONTIEL J M M, TARDOS J D. ORB-SLAM: a versatile and accurate monocular SLAM system[J]. IEEE Transactions on Robotics, 2015, 31(5):1147-1163.

[5] Campos C, Elvira R, Rodr ́guez J, et al.ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM[J],2020.

[6] GIOI R G V, JAKUBOWICZ J, MOREL J M, et al .LSD: A fast line segment detector with a false detection contorl[J]. IEEE Trasactions on Pattern Analysis & Machine Intelligence,2010,32(4):722-732.

[7] WANG Z, LIU H,WU F. MSLD: A robust descriptor for line matching[J]. Pattern Recognition, 2009,42(5): 941-953.

[8] ZHANG L, 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):794-805.

[9] Kummerle R, Grisetti G, Strasdat H, et al. G2o: A general framework for graph optimization[C]. IEEE International Conference on Robotics & Automation,2011.

[10] Mur-Artal R, Tard ́os J. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras[J]. IEEE Transactions on Robotics,2017.