IOP Conference Series: Materials Science and Engineering

An unmanned aerial vehicle pose estimation system based on SLAM

An unmanned aerial vehicle pose estimation system based on SLAM

Xinglei Dou[1], Yuzhou Huo[1], Yongchang Liu[1] and Xin Wang[2,3*]

[1] College of Software Engineering, Jilin University, Changchun, Jilin, 130012, China

[2] College of Computer Science and Technology, Jilin University, Changchun, Jilin, 130012, China

[3] Key Laboratory of Symbolic Computation and Knowledge Engineer of Ministry of Education, Jilin University, Changchun, Jilin, 130012, China

[*]Corresponding author's e-mail: w_x@jlu.edu.cn

Abstract. This paper presents an unmanned aerial vehicle (UAV) pose estimation system based on monocular simultaneous localization and mapping (SLAM) guided by the desired shot. The system enables UAV to automatically adjust the pose to achieve a shot close to the desired shot provided by the user. The SLAM module in the system includes ORB feature-based visual odometry and Levenberg-Marquardt method-based optimizer. To ensure the reliability of the camera pose estimation result, the bag of words model is used to select an image which has enough good matches with the desired shot. The experimental results prove that the system is valid and effective.

1. Introduction

Unmanned aerial vehicles (UAVs) have been widely used in aerial photography, geographical exploration and emergency response in disaster areas. However, UAVs are difficult to manipulate. It is hard to reach the ideal pose and get the desired shot in a short time. In the paper, a UAV pose estimation system is designed to solve the problem. The system is based on monocular simultaneous localization and mapping (SLAM) and is guided by the desired shot.

1.1. System overview

The purpose of the system is to enable a drone to adjust its pose to capture a shot that is closed to the desired shot provided by the user. The system is designed for drones with a monocular camera and a Pixhawk flight control[1].

The system structure is shown in Figure 1. With a 3DR radio telemetry, the ground control station (GCS) can establish a connection with the airborne flight control. The user can monitor the status of the drone and export flight missions to the drone using methods implemented in GCS core. The image transmission equipment transmits the content captured by the camera to the picture transmission receiver in real time. The GCS obtains the video stream by accessing the web server hosted by the image transmission receiver.

The SLAM and pose estimation module is the core of the system. The sequence of images acquired by the monocular camera is used to conduct pose estimation of the drone and construction of the surrounding map. The drone's target pose is estimated based on the results obtained from SLAM and the desired shot provided by the user. The desired shot acquisition process is detailed in 2.3. Finally, the target pose is exported to the flight control as a mission by the GCS. The flight control guides the drone to the target pose.

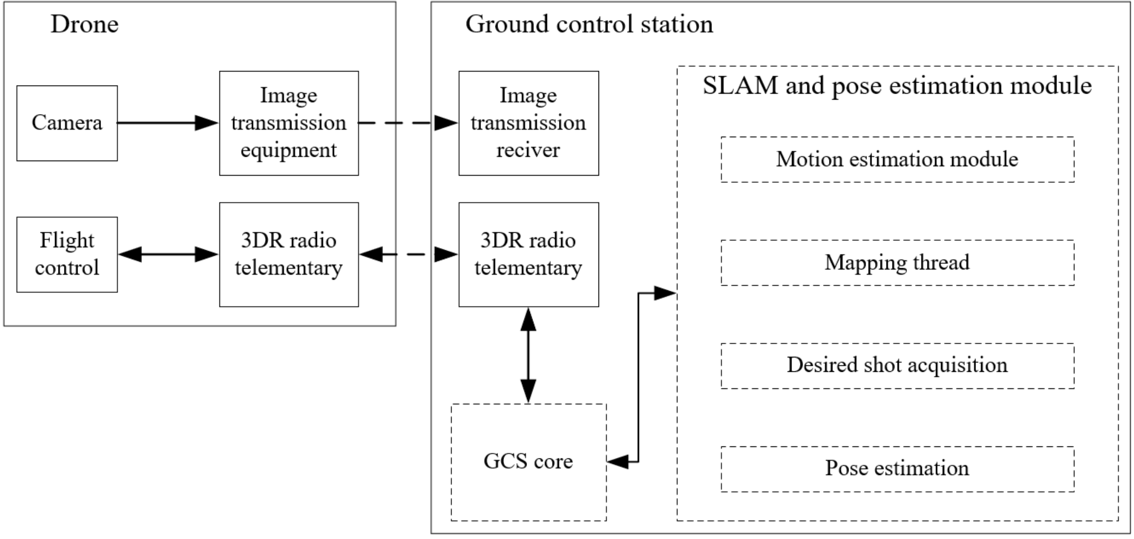

Figure 1. System overview.

The module has two parallel threads: motion estimation thread and mapping thread. The first one extracts ORB feature points, solves the motion of the camera between the current frame and the previous frame. The second one optimizes camera pose and map structure by minimizing re-projection errors and reconstructing the map.

1.2. Related works

When it comes to SLAM, A.J. Davison proposed MonoSLAM[2] in 2007, which was the first real-time monocular vision SLAM system. The keyframe mechanism of PTAM [3-4] proposed the parallelization of tracking mapping process and was considered as the most accurate SLAM method from a monocular video in real time. ORB-SLAM [5] was a well-known successor to PTAM with good hardware versatility. Based on ORB features [6], the non-linear algorithm prevented the cumulative error effectively. It could quickly recover from losing track as well. However, the ORB feature detection was time-consuming. The three-threaded structure imposed a heavy burden on CPU, which made it difficult to transplant to embedded terminals.

Moreover, there were also many other works such as LSD-SLAM[7] and SVO[8]. LSD-SLAM used the direct method to track, which enabled it to be insensitive to feature missing areas. However, it was unreliable when the camera moved fast. The most significant advantage of SVO was its speed. Taking advantage of the sparse direct method, it did not have to compute descriptors, nor did it need to process as much information as dense and semi-dense so that it could achieve real-time performance easily.

2. Methods

2.1. Visual odometry

Visual odometry (VO) is the front part of visual SLAM. It estimates the rough camera motion according to the parallax of adjacent images and provides a good initial value for further processing.

From the perspective of feature extraction, the implementation of VO can be divided into two categories: feature-based methods and direct methods. The former works well even there is much noise and the camera moves fast; The latter does not need to extract features and can build dense maps but has the disadvantages of heavy computation and poor robustness. Therefore, feature-based methods have long been (until now) considered as the mainstream method of VO due to its robustness. It runs stably and is a mature solution at present.

On account of the ORB can be calculated faster, which enables it runs real-timely even in a portable device. It also has excellent uniformity and stability. Hence, ORB is used for it compromises well between quality and performance.

By matching these feature points in adjacent images, the motion of the camera between the two frames can be estimated. Specifically, essential matrix E is solved by constraint of epipolar line, which means for two pictures, the location of cameras and the position of one feature point in the world coordinate system are coplanar points.

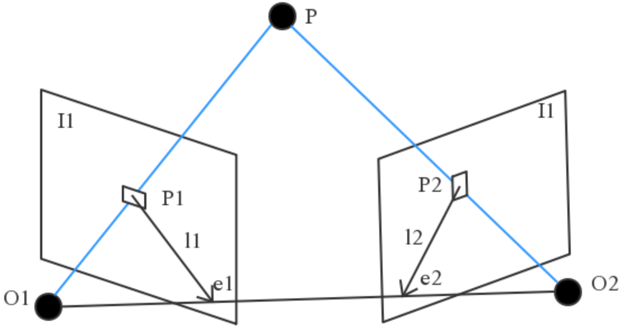

Figure 2. Epipolar Constraint.

For monocular SLAM, triangulation is needed to figure out the depth of the feature points for map initialization. The depth of a point is determined by the angle of two different connections between it and observation points.

After those preparatory work, we solve the 3d-2d point pair motion with Perspective-n-Point (PnP). In that way, we can minimize re-projection errors.

2.2. Non-linear optimization

Due to the accumulation of long-term errors, the trajectory of the camera and the map structure may be biased and inaccurate. Global optimization is needed to optimize the camera trajectory and map structure.

There is a generic abstract representation for SLAM problem. The variables to be optimized is formulated as:

$$X = \left[x_1, \cdots, x_\mathrm{N}, y_1, \cdots, y_\mathrm{M}\right] \tag{1}$$

Where x_n denotes the pose of the camera, y_m denotes the coordinates of the landmarks.

The motion model x_k denotes that the camera's pose changes from x_{k-1} to x_k at time k because of the motion u_k. The observation model $z_{k,\mathrm{j}}$ denotes the pixel coordinate of y_j's projection to the image plane of pose x_k at time k.

$$\begin{cases} x_k = f(x_{k-1}, u_\mathrm{k}) + w_k \\ z_{k,\mathrm{j}} = g(x_k, y_\mathrm{j}) + v_{k,\mathrm{j}} \end{cases} \tag{2}$$

where w_k and $v_{k,\mathrm{j}}$ are both random errors subjected to a Gaussian distribution with a mean of 0.

$$w_k \sim N(0, R_k), v_{k,\mathrm{j}} \sim N(0, Q_{k,\mathrm{j}}). \tag{3}$$

Thus, the conditional probability of $z_{k,\mathrm{j}}$ is:

$$P(z_{\mathrm{j},k} \mid x_k, y_\mathrm{j}) = N(g(x_k, y_\mathrm{j}), Q_{k,\mathrm{j}}). \tag{4}$$

Now we are solving an argmax problem. To maximize (4) is to minimize the negative logarithm of (4). Define the error between the truth and the estimated result as:

$$\begin{aligned} e_{u,k} &= x_k - f(x_{k-1}, u_k). \\ e_{y,\mathrm{j},k} &= z_{k,\mathrm{j}} - h(x_k, y_\mathrm{j}). \end{aligned} \tag{5}$$

Now the problem is described as:

$$(x,y)^* = \arg\min\left(e^T_{y,j,k} Q^{-1}_{k,j} e_{y,j,k}\right) \tag{6}$$

A least squares problem is constructed:

$$J(x) = \sum_k e^T_{u,k} R^{-1}_k e_{u,k} + \sum_k \sum_j e^T_{y,k,j} Q^{-1}_{k,j} e_{y,k,j} \tag{7}$$

The Levenberg-Marquardt method is then applied to solve the least squares problem. The algorithm is implemented in g2o framework[9], which is utilized to solve the optimization problem.

2.3. Desired shot acquisition

The overall goal of camera pose estimation is to make the camera shot similar to the desired shot when it reaches the target pose solved by camera pose estimation. The desired shot acquisition process allows the user to interact with the system and generate the desired shot easily.

It is vital that the desired shot is generated from the scene that is part of the map constructed by SLAM. To obtain the pose of the camera, the desired shot needs to be matched with the map to solve a PnP problem. If a scene was not a part of the map constructed by SLAM, there would be a high probability that the pose estimation fails.

There is a keyframe set in the system which contains almost all scenes in the map. Take a frame selected from the keyframe set as the reference frame. The reference frame is perspective-transformed based on the four anchors specified by the user in the frame. Perspective transformation is the projection of a picture onto a new view plane. It is a mapping of two-dimensional (x, y) to three-dimensional (X, Y, Z) to another two-dimensional (x', y') space. The result is obtained by multiplying the perspective transformation matrix by each pixel in the reference frame.

$$\begin{bmatrix} \mathrm{x}' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{8}$$

Firstly, calculate a perspective transform with four vertexes of the reference frame and the corresponding anchors specified by the user by solving a linear system of equations. Secondly, apply the perspective transform to the reference frame. The perspective-transformed frame is the same size as the reference frame. Pixels without foreground are filled with RGB(0,0,0) by default. Then the desired shot is obtained.

2.4. Pose estimation

It is necessary to establish a matching relationship between the feature points on the image and the three-dimensional points in the map to solve the PnP problem.

When the desired shot is provided, the system finds a keyframe which is most similar to the desired shot and use these two images to perform feature points extraction and matching. Hence the matching relationship between the feature points in the desired shot and the three-dimensional points of the map can be solved. The bag of words module implemented in DBoW2[10] is used to find the images with the highest similarity to the desired shot. A feature point matching test should be performed between the image selected by DBoW2 and the desired shot to ensure the result is reliable.

Then the system utilizes the matched features to solve the PnP problem to obtain the camera pose of the desired shot.

3. Results

The TUM data set[11] is used to test the system. This data set provides a sequence of images of a room, including depth images and RGB images. Only the RGB images are needed here.

Run the system with the TUM data set and view the ground truth and estimation trajectory for comparison in figure 3.

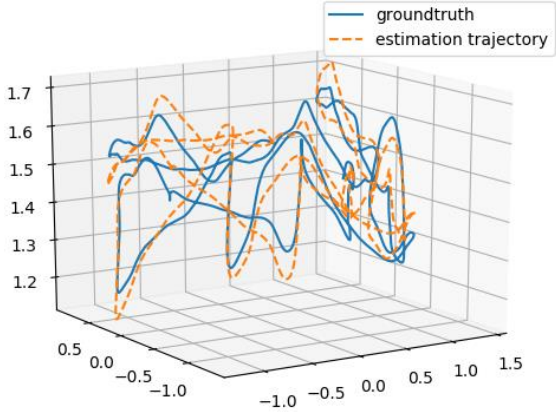

Figure 3. Estimated trajectory and ground truth.

In order to view the result of camera pose estimation clearly, the SLAM was run on the first 100 frames of [11]. The frames in the subset are of good continuity and stability. A frame from the subset is selected as the reference frame. The desired shot is generated by applying a perspective transform to the reference frame. As Figure 4 shows, the left one is the reference frame, and the right one is the desired shot.

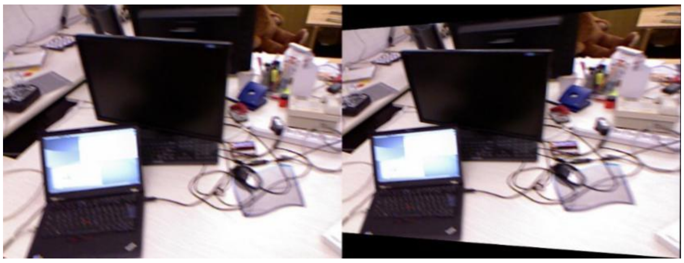

Figure 4. Reference frame(left) and the desired shot(right).

Then the bag of words model is applied to select the frame with the highest similarity to the desired shot from the subset. After feature matching, the estimated camera pose is obtained by solving a PnP problem. Figure 5 shows the result of the camera pose estimation.

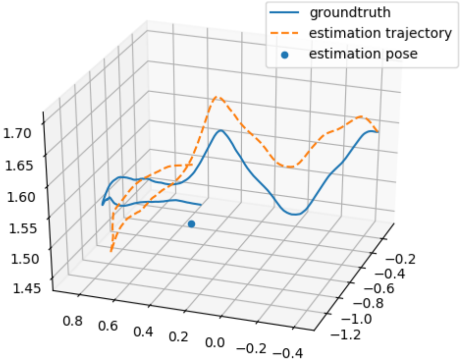

Figure 5. Result of camera pose estimation.

4. Conclusions

In this work, we have presented a result-oriented UAV pose estimation system. Compared with other SLAM implementation, there is no loop closing detection in the system, which may lead to errors in global map construction. The system enables the user to adjust the camera pose in a simple and innovative way. The main idea of the system is to solve a PnP problem to estimate the camera pose using the desired shot and constructed map structure. As shown in the experimental results, the system is effective and reliable.

References

[1] Meier, L., Honegger, D. and Pollefeys, M., (2015) PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In: IEEE International Conference on Robotics & Automation. Seattle. pp. 6235-6240.

[2] Davison, A.J., Reid, I.D., Molton, N.D. and Stasse, O. (2007) MonoSLAM: Real-Time Single Camera SLAM. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29: 1052-1067.

[3] Klein, G. and Murray, D., (2008) Parallel Tracking and Mapping for Small AR Workspaces. In: IEEE & Acm International Symposium on Mixed & Augmented Reality. Nara. pp. 225-234.

[4] Klein, G. and Murray, D., (2008) Improving the Agility of Keyframe-Based SLAM. In: European Conference on Computer Vision. Marseille. pp. 802-815.

[5] Murartal, R., Montiel, J.M.M. and Tardos, J.D. (2017) ORB-SLAM: a Versatile and Accurate Monocular SLAM System. IEEE Transactions on Robotics, 31: 1147-1163.

[6] Rublee, E., Rabaud, V., Konolige, K. and Bradski, G.R., (2012) ORB: an efficient alternative to SIFT or SURF. In: International Conference on Computer Vision. Barcelona. pp. 2564-2571.

[7] Engel, J., Schöps, T. and Cremers, D., (2014) LSD-SLAM: Large-Scale Direct Monocular SLAM. In: Computer Vision – ECCV 2014. Switzerland. pp. 834-849.

[8] Forster, C., Pizzoli, M. and Scaramuzza, D., (2014) SVO: Fast Semi-Direct Monocular Visual Odometry. In: IEEE International Conference on Robotics & Automation. Hong Kong. pp. 15-22.

[9] Kuemmerle, R., Grisetti, G., Strasdat, H., Konolige, K. and Burgard, W., (2011) g2o: A general framework for graph optimization. In: IEEE International Conference on Robotics and Automation (ICRA). Shanghai. pp. 3607–3613.

[10] Galvez-Lopez, D. and Tardos, J.D. (2012) Bags of Binary Words for Fast Place Recognition in Image Sequences. IEEE Transactions on Robotics, 28: 1188-1197.

[11] Sturm, J., Engelhard, N., Endres, F., Burgard, W. and Cremers, D., (2012) A benchmark for the evaluation of RGB-D SLAM systems. In: IEEE/RSJ International Conference on Intelligent Robots & Systems. Vilamoura. pp. 573-580.