Research of SLAM for Indoor Environment based on Kinect

Di Liu[a], Ri Pan[b] and Yajun Zhang[c*]

School of Mechanical and Electrical Engineering, Beijing University of Chemical Technology, Beijing 100029, China.

[a]ldi2009@163.com, [b]panri@mail.buct.edu.cn, [c*]zhyj@mail.buct.edu.cn

Corresponding Author: Yajun Zhang

Keywords: SLAM, Kinect, ORB, loop closure detection, pose graph optimization.

Abstract. The research of mobile robot's intelligence currently is a hot topic, and the basic and key technology to achieve the intelligence is Simultaneous Localization and Mapping(SLAM). In order to solve the problem of localization for mobile robots based on vision, A method of SLAM based on Kinect is proposed. Firstly, the successive images are captured by Kinect. Secondly, the frame-to-frame alignments are performed based on ORB(Oriented FAST and Rotated BRIEF) features between frames, and the relative motion transformations are computed via the PnP algorithm. Thirdly, the key-frames are defined by the relative motion estimated between frames, then loop closure detection and global graph optimization are performed to efficiently decrease the accumulative error of poses and achieve a global consistence trajectory. Finally, the OctoMap are applied to represent the environment with less data amount. Experimental results show the feasibility and effectiveness of this method.

1. Introduction

The research of mobile robot's intelligence has become a trend, and SLAM is the basic and key technology to achieve that, which aims to estimate the trajectory of mobile robots by the information of sensors mounted on robots and incrementally map the explored environment simultaneously. With the development of computer vision technology, the research of SLAM based on camera is becoming more and more popular because of the characteristics of cheap, lightweight and easy to mount[1].

Kinect is a depth camera that can capture the color images along with per-pixel depth information in high rate, which provides the advantage for its application in SLAM. Henry et al[2] proposed a method combining visual features and RGBD-ICP to create and optimize a pose graph. The method employed *Surfel* elements to represent the environment, however the real-time performance was poor. Newcombe et al[3] presented a novel modeling method for indoor environment, which was able to fine rendering the environment with high precision and good effect, but its computation expense was very huge. Based on the above analyses, a method of SLAM based on Kinect is proposed, combining the fast ORB feature and OctoMap map framework to represent the environment effectively.

2. System Architecture

The system architecture is shown in Fig.1. It mainly includes frame-to-frame alignment, key-frame selection, loop closure detection, pose graph optimization and mapping.

The frame-to-frame aims to estimate the relative motion between frames. The operations are based on the fast ORB features. Wrong matches are rejected by RANSAC and relative motion

transformation can be estimated by means of the PnP algorithm.

Key-frame selection mainly aims to decided which frames should be used to estimate the trajectory and build the map of environment, which can efficiently decrease the data amount of map. The basic of selection is the relative motion between frames.

Loop closure detection and graph optimization aim to decrease the accumulative errors of poses and achieve a global consistence trajectory respectively.

Mapping is the process to build the map of explored environment, which is accomplished on the basic of global consistence trajectory provided by graph optimization and represented by OctoMap.

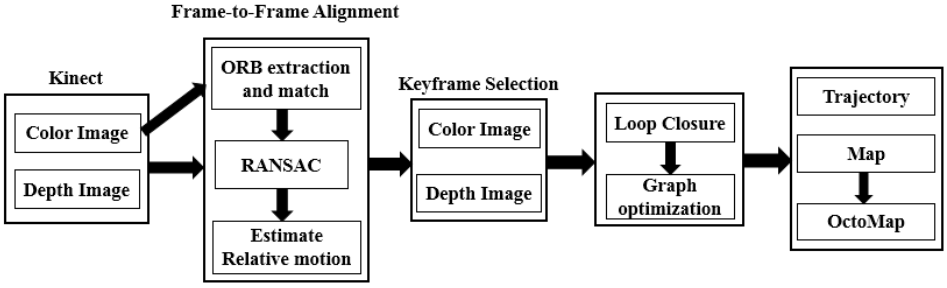

Fig.1. System architecture

3. Frame-to-Frame Alignment

The poses of Kinect during the motion can be initially estimated by frame-to-frame alignments. Frame-to-frame consists in feature extraction and match, relative motion estimation.

3.1 Feature Extraction and Match

Taking into account the real-time performance, ORB feature is chosen to perform frame-to-frame alignment, which is invariant to rotation and scale(in a limit range), having a good real-time and overall performance[4].

Considering the real time capability and validity, a feature extraction strategy based on a grid is proposed: an image is divided into multi-grid before extraction, and ORB features in every grid are extracted and then matched between frames.

The Brute Force search method is applied to match two sets of feature points between two frames. In addition, the RANSAC and geometrical constraint are applied to reject the outliers.

3.2 Relative Motion Estimation

Based on the features matched between frames, the relative motion transformation can be estimated. The PnP algorithm is chosen to estimate the relative motion because of taking into account the re-projection errors. More precisely, the error function is the form as follows:

$$T_{k-1}^{k} = \arg\min_{T} \sum_{i} || p_k^i - {p'}^{i}_{k-1} ||^2 \tag{1}$$

Here, k is the timestamp, i means the *i*th feature point, p_k^i is the feature points set in frame I_k, ${p'}^{i}_{k-1}$ is the re-projection of the 3-D points corresponding with the p_{k-1}^i into image I_k according to the transformation T. This problem can be solved by an non-linear optimization using an iteration method.

3.3 Key-frame Selection

Key-frame technology is used to decrease the data redundancy. The key-frames are selected based on the motion between frames, the measurement of motion is defined as follows:

$$e = \|\Delta t\| + \min(2\pi - \|R\|, \|R\|) \tag{2}$$

where R and t represent the rotation and translation between frames respectively.

The overall procedure for selecting an key-frame is shown in Fig.2.

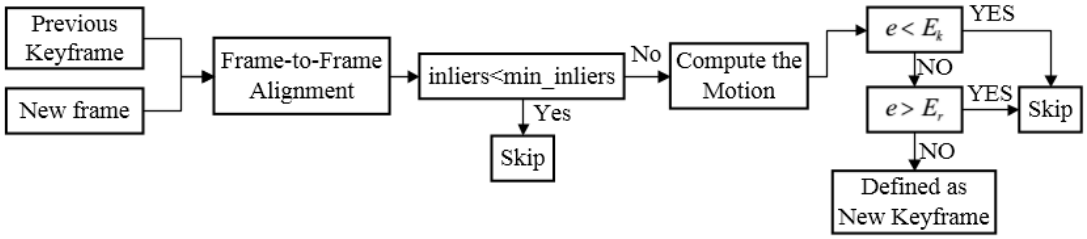

Fig.2. The procedure of key-frame selection

where *inliers* is the number of matches after RANSAC, *min_inliers* is the threshold for minimum inliers required for the frame-to-frame alignment, *e* is the motion between frames described as (2), E_k is the threshold for the minimum motion and E_r for the maximum motion. Every new coming frame should be decided whether it is a key-frame or not by the procedure described above.

4. Loop Closure Detection and Graph Optimization

Alignment between the successive frames is a good method for tracking the Kinect pose over moderate distances. However, errors in alignment, noise and quantization in depth values will cause the estimation of Kinect's poses to drift over time, leading to inaccuracies in the map. Therefore, all the frames should be taken into account other than just the adjacent ones to increase the accuracy and reduce the drift. This problem is known as the loop closure detection and pose graph optimization. On the specific implementation, two different loop closure strategies are proposed: the nearby loop closure and the random loop closure.

For each key-frame, the nearby loop closure is always performed to estimate the motion between key-frames, then the random loop closure will be performed to reduce the drift more efficiently.

A pose graph structure is applied to represent the relationships between frames, with nodes corresponding to the poses of Kinect and edges corresponding to geometric constraints. In detail, the optimization is achieved by minimizing the error function of the form:

$$\min F(x) = \sum_{k=1}^{n} e_k(x_k, z_k)^T \Omega_x e_k(x_k, z_k) \tag{3}$$

where, $e_k(x_k, z_k)$ is an error vector that measure how well the pose x_k satisfy the constraint z_k, Ω_x is the information matrix that shows a prior knowledge about the constraint z_k.

The procedure described so far can achieve a globally consistent trajectory. Based on this trajectory, the map of environment can be built by projecting the original points into the world coordinate frame. The OctoMap framework[5] is employed to represent the environment, with the advantages of having a compact structure, being able to map overlap regions well and can easily be used in application like navigation and path planning.

5. Experimental Results and Discussion

In order to test the performance of the method, a public dataset fr1_room provided by the Computer Vision Group is adopted, the images(color and depth images) are captured by Kinect with a 640×480 resolution and 30 fps. The program is tested on a PC running the i3-M350 CPU and ATI HD5650 GPU under Ubuntu operating system.

The results of experiment are shown in Fig.3, Fig.4, Fig.5 and Fig.6. The pose graph shown in Fig.3 contains 72 nodes and 455 edges; and the size of the point cloud shown in Fig.5 is 125Mb, but the size is just 3.4Mb when using the OctoMap, reducing the data amount to 1/36 of the original. The errors of the estimated trajectory to the ground truth is shown in Table 1.

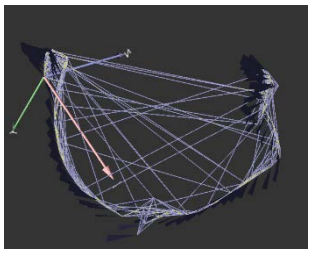

Fig.3 Loop closure result

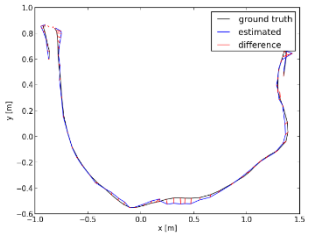

Fig.4 The estimated trajectory

Fig.5 Point Cloud

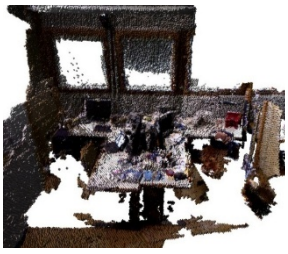

Fig.6 OctoMap

Table 2 Accuracy evaluation of the algorithm on 400 frames of the dataset

ATE mean error [m]	0.028264 m
ATE median error [m]	0.024512 m
ATE maximum error [m]	0.122537 m
ATE minimum error[m]	0.004597 m
Time cost[s/frame]	0.14s/frame
Dataset description	fr1_room, 13.60s, 5.17m, 400 frames

6. Summary

In this paper, a SLAM system based on Kinect and the fast ORB feature is proposed. The data amount are decreased by key-frame selection and OctoMap framework, and the accumulative errors can be effectively decreased by the loop closure, finally the global consistent trajectory and map would obtained by pose graph optimization. Experimental results showed that the mean error of estimated trajectory was 0.028m in the trajectory length of 5.17m, and reducing the amount of data to 1/36 of the original by using the OctoMap mapping framework.

Acknowledgement

This work was financially supported by the Fundamental Research Funds for the Central Universities (YS1403, ZY1521); the China Postdoctoral Science Foundation (2015M570920); State Key Laboratory of Organic-Inorganic Composites (Open issue 201601011); Beijing Natural Science Foundation (2131003).

Reference

[1]. Durrant-Whyte H, Bailey T, et al. Simultaneous localization and mapping: part I[J]. IEEE Robotics & Automation Magazine. Vol. 12 (2006) No. 2, p. 99-110.

[2]. Henry P, Krainin M, Herbst E, et al. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environment[J]. International Journal of Robotics Research. Vol. 31 (2012) No. 5, p. 647-663.

[3]. Newcombe R A, et al. KinectFusion: Real-time dense surface mapping and tracking[C]. IEEE International Symposium on Mixed and Augmented Reality. Switzerland, 2011, p. 127-136.

[4]. Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF[J]. Proceedings. Vol. 58 (2011) No. 11, p. 2564-2571.

[5]. Hornung A, Kai M W, Bennewitz M, et al. OctoMap: An efficient probabilistic 3D mapping framework based on octrees[J]. Autonomous Robots. Vol. 34 (2013) No. 3, p. 189-206.