

## Research of SLAM for static environment based on Kinect

Hong-Yan Chen<sup>a</sup> and Ya-Jun Zhang<sup>b</sup>

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

*Email:*<sup>a</sup>*chy1100824@126.com,* <sup>b</sup>*zhyj@mail.buct.edu.cn*  
*www.buct.edu.cn*

Intelligent mobile robots require autonomous navigation and localization in the environment. In order to realize the function of localization and navigation, the robot often needs to obtain the 3D map of the environment. 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 continuous images are obtained by Kinect sensor. Secondly, to find the matching points between the two frames via SIFT feature matching method, then RANSAC algorithm is used to remove the errors of matching, and the relative motion transformations are computed via the PnP algorithm. Thirdly, loop closure detection and global graph optimization are used to eliminate the accumulated robot pose errors and achieve a continuous trajectory. Finally, the 3D map of the environment is obtained. Experimental results show the feasibility and effectiveness of this method.

*Keywords:* SLAM; Kinect; Loop Closure Detection; Pose Graph Optimization.

### 1. Introduction

In recent years, it has been a hot research topic for the autonomous navigation and object recognition in unknown environments. The research of SLAM provides a good direction for this problem. It aims to estimate the trajectory of the robot and the 3D map of the environment by using the information provided by the robot vision sensors. Visual SLAM technology is becoming more and more popular because of the large amount of information in the visual sensor system and the development of the computer vision technology [1].

Microsoft developed Kinect sensor in 2010 which combined with a common camera and laser sensors. The information of environment can be easily obtained by Kinect with the continuity color and depth pictures. Therefore, it is widely used in the SLAM research field. Newcombe et al [2] 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 RGB and Depth feature and point cloud map framework to represent the environment effectively.

## 2. System architecture

A classical SLAM system can be conceptually divided into two parts: the front-end and the back-end. The front-end abstracts sensor data into models that are amenable for estimation, while the back-end performs inference on the abstracted data produced by the front-end [3]. A classical SLAM system is shown in Fig.1.

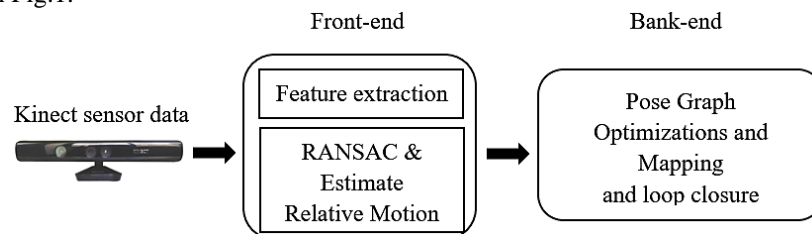


Fig. 1 classical SLAM system architecture

In this paper, a full SLAM method is proposed on the basis of the above. The full SLAM includes several parts as those: a frame matching and error elimination, keyframe selection for loop closure detection and global graph optimization, trajectory generation and map construction, which is show in Fig. 2.

Features extraction is aimed to match two images fast by the feature points. But this kind of matching has many false matches. The RANSAC algorithm is used to solve this problem. Finally, the correct matching of feature points is obtained. At this time, the relative motion between frames is computed by the PnP algorithm. The main purpose of Keyframe Selection is to avoid frequent updating the map by to remove frames which are too close. Loop closure and graph optimization with g2o [4] aim to decrease the accumulative errors of poses and achieve a global consistence trajectory respectively, otherwise it is useful to reduce the trajectory drift and possibly correct it. By the function of localization and mapping, a global consistence trajectory and a globally consistent 3D point cloud map are built when the robot is to explore an environment. At the same time, the octoMap map is also built to decrease the data space. In addition to the octoMap can be used to navigate, but this function is not researched in this paper.

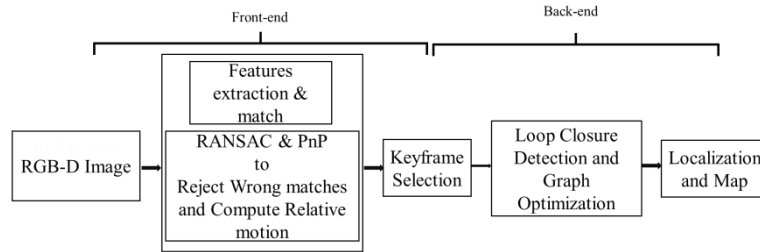


Fig.2. full SLAM system architecture

### 3. Front-End

The front-end abstracts sensor data into models that are amenable for estimation.

#### 3.1. Features Extraction and Match

SLAM consists in the simultaneous estimation of the state of the robot and the map of the environment. On the simultaneous estimation of the state of the robot is to solve such a problem: given two frames of images and to get the relative motion of them, because of the motion of an image contains the motion of the camera or robot.

The feature-based method is applied to match two sets of feature points between two frames. Assume that we want to match two frames  $F_1$  and  $F_2$ , and we obtain the characteristic points of the two groups:

$$P = \{p_1, p_2, \dots, p_N\} \in F_1 \quad (1)$$

$$Q = \{q_1, q_2, \dots, q_N\} \in F_2 \quad (2)$$

where,  $p_N$  and  $q_N$  are feature points set in frame point set  $R^{3 \times 3}$ .

The aim is to derive a rotation matrix  $R$  and a displacement vector  $t$ :

$$\forall i, p_i = Rq_i + t \quad (3)$$

However, in practice due to errors, equality is not possible. So, actually, it is a nonlinear least squares problem as follow:

$$\min_{R, t} \sum_{i=1}^N \|p_i - (Rq_i + t)\|_2 \quad (4)$$

The problem (4) is commonly solved via successive linearization. In addition, the RANSAC and geometrical constraint are applied to reject the outliers.

#### 4. Back-End

The back-end performs inference on the abstracted data produced by the front-end.

##### 4.1. Keyframe selection

It is unwise to put every frame on the map. Because if the frame and next frame is very close, it will lead to frequent updates the map. Thus, it is not only a waste of time, but also occupy a lot of storage space. So Keyframe technology is used to decrease the data redundancy by adding a key frame every other period of time, and finally the keyframe is added to the map only.

##### 4.2. Loop closure detection and graph optimization

The essence of loop closure detection is to identify place the robot had arrived. Loop detection can effectively increase the accuracy of the results and reduce the drift over time.

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.

Graph optimization is to show the general optimization problem in the form of graph. Graph is composed of vertices and edges. The edge representation some relation between vertices. The core of SLAM is to calculate the trajectory and map of the robot according to the observed data. About graph optimization, vertices corresponding to the poses of Kinect and edges corresponding to geometric constraints. It can be expressed as

$$z_k = h(x_k) \quad (5)$$

Considering the errors it expressed as

$$e_k = z_k - h(x_k) \quad (6)$$

In detail, the optimization is achieved by minimizing the error function of the form:

$$\min F(x) = \sum_{k=1}^n e_k(x_k, z_k)^T \Omega_x e_k(x_k, z_k) \quad (7)$$

Where,  $e_k(x_k, z_k)$  is an error vector that measure how well the pose  $x_k$  satisfy the constraint  $z_k$ ,  $\Omega_x$  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] are 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 feasibility of this method, the public data set NYU Depth Datasets v2 [7] and fr1\_room Datasets provided by the computer vision group [6] are used to verify the method proposed in this paper. Those pictures are composed of RGB and depth charts which are captured by Kinect, with  $640 \times 480$  resolution and 30 fps. The experiment is tested on a notebook PC with Intel® Core™ i3 M380 CPU, 2.53GHz, 6GB RAM, NVIDIA GeForce GT 520M GPU, and under Ubuntu 14.04 LTS Linux operating system and the program procedure is described as below.

### 5.1. Experiment of NYU v2-Café dataset

The result of experiment using NYU dataset are show in figure as below.



Fig. 3(a) Point Cloud



Fig. 3(b) OctoMap

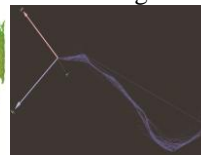


Fig. 4(a) loop closure result

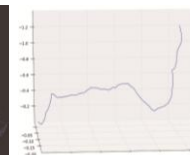


Fig. 4(b) Robot trajectory

### 5.2. Experiment of TUM-fr1\_room dataset

The result of experiment using fr1\_room dataset are show in figure as below. Picture (a) is obtained by official standard data, and picture (b) is obtained by full SLAM method proposed in this paper.

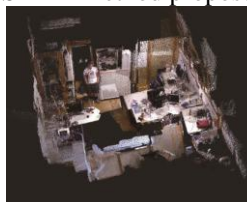


Fig. 5 Point Cloud

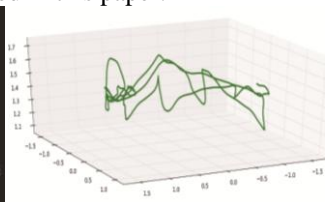


Fig. 6(a)

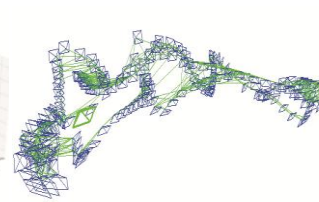


Fig. 6(b)

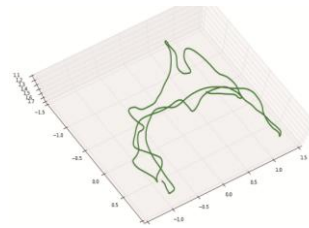


Fig. 7(a)

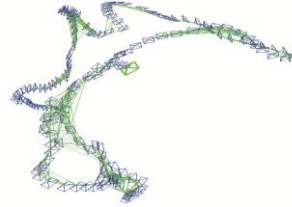


Fig. 7(b)

The results of the experimental output the three-dimensional map of the input images and a set of trajectory of the camera, and compared with the standard trajectory given by the official show the feasibility and effectiveness of the experiment.

## 6. Summary

In this paper, a SLAM system based on RGB-D image is proposed. This approach, keyframe are selected to reduce update frequency of the map, reduce the amount of calculation and compute time, at the same time it also save the storage space, through the loop closure effectively reduced drift errors over time. Finally, global graph optimization scheme to robot trajectory in the environment and obtain a 3D map of the environment. So the problem about Localization and Mapping can be solved effectively by the method proposed in this paper.

## References

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. Newcombe R A, Izadi S, Hilliges O, et al. KinectFusion: Real-time dense surface mapping and tracking[C]. IEEE International Symposium on Mixed and Augmented Reality. Switzerland, 2011, p. 127-136.
3. Cadena C, Carlone L, Carrillo H, et al. Simultaneous Localization And Mapping: Present, Future, and the Robust-Perception Age[J]. 2016.
4. R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), Shanghai, China, May 2011.
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.

6. ComputerVisionGroup[EB/OL].[http://vision.in.tum.de/data/datasets/rgbd\\_dataset/download](http://vision.in.tum.de/data/datasets/rgbd_dataset/download).