Dense 3D Environment Reconstruction with an RGB-D Camera for Mobile Robot

Chih-Hsuan Chen

***Abstract*— In this paper, we present an environment reconstruction system to generate an indoor 3D map for mobile robots. Using an RGB-D sensor, the robot doesn't need the initial odometry. Furthermore, the system can be used for reconstruction of a 3D environment model by manually. We optimize our approach to reach 10Hz for the front-end and 1Hz for the back-end to fulfill the applications for the mobile robot. Our final goal is to develop the 3D SLAM systems which combine the Region Based Convolution Neural Network (MASK R-CNN) for creating a 2D semantic image and a 3D semantic map. The experimental results demonstrate some preliminary results for 3D reconstruction with an RGB-D camera for creating the point cloud map and the OctoMap for the mobile robot (Care-O-bot 4) in an indoor environment.**

I. Introduction

To explore an unknown indoor environment, a mobile robot needs to create a map and localize itself in the map simultaneously. This procedure, called simultaneous localization and mapping (SLAM), is challenging and difficult to deal with visual SLAM. The major challenge with visual SLAM is due to the uncertainty of measurements, varying light conditions, and noise from the sensor. The camera can be described the estimated poses of the robot from RGB-D data can create a 3D model of an indoor environment at the same time.

Mobile robots typically use wide range sensors such as 2D laser scanners for measuring the indoor environment with very high accuracy. The state-of-the-art laser-based SLAM (simultaneous localization and mapping) are known as [1] [2]. To estimate of a camera on the robot motion is known as visual odometry[3].

In this paper, we present an approach with the small improvement to build a 3D map and localize in the map simultaneously based on RGB-D data illustrated in Fig. 2. The 3D environment reconstruction system can be slipped into 3 parts, Front-End, Back-End, and mapping.

In the Front-End part, it can be divided into three subfunction, which includes feature extraction, feature matching and pose estimation. The features from RGB images around the corner and edge can be extracted. Once we collect all features, it can be applied these features key-points for matching with the pervious image. In our approach, we selected the Oriented FAST and Rotated BRIEF (ORB) feature extraction. Based on the features matching results, we can estimate the 3D poses of the any two corresponding frames using Efficient Perspective-n-Point (EPnP) [8]. Therefore, the robot is evaluated the transformation of each frames based these correspondences.

Fraunhofer Institute for Manufacturing Engineering and Automation IPA, Stuttgart, Germany
chih-hsuan.chen@ipa.fraunhofer.de

Fig. 1: The RGB-D Camera mounted on the mobile robot Care-O-bot 4 [15]

As we mentioned the major problem in previous section, it will be accumulated the estimation error and cause the accumulating drift problem. In order to resolve this problem, we need to optimize the pose estimates between frames. In the Back-End part, the approach is applied General Graph Optimization (g2o) library which is open source framework for optimizing nonlinear error functions [14] to reduce the accumulating drift ,and the approach is applied loop closure for detecting the previous scene to provide optimizing loop[12]. For the mapping part, the point cloud map (PCL) and OctoMap[13] are utilized to express the 3D environment reconstruction. The preliminary results for 3D reconstruction are presented in Section IV. Finally, we conclude with some future works in Section V.

II. Related Work

A classical approach to visual SLAM, the Mono SLAM is the first real-time monocular visual SLAM system proposed by A.J. Davison [4]. MonoSLAM is using EKF filter as backend and using sparse feature extraction as frontend.

The Parallel Tracking and Mapping (PTAM) is proposed by Klein [5]. It achieves not only the tracking and mapping parallel, but it also introduces the nonlinear optimization instead of traditional optimization such as EKF filter or particle filter. After PTAM, many kinds of research in the field of visual SLAM are using nonlinear optimization as a backend.

The class of algorithms known as iterative closest point (ICP) [10], minimize the distance between two sets of point cloud, which can be generated from two raw scans. The ICP is applicable when we have in good initial guess, otherwise it is likely to be stuck into a local minimum.

ORB-SLAM [6] is known as backend and inherited from PTAM. Comparing with PTAM, there are several advantages for visual SLAM. It supports three types consisting of RGB-D cameras, stereo camera and Monocular camera. Instead of using Scale-invariant feature transform (SIFT) or speeded-up robust features (SURF) feature extraction, the frontend of ORB-SLAM is using ORB feature extraction. It could reduce the computation time and also improve consistency with rotation and zooming. It also uses loop closure to decrease the accumulating error from pose estimation.

III. 3D Reconstruction

The approach of 3D visual SLAM system consists of third main part, Front-End, Back-End, and mapping. The system architecture is illustrated in Fig. 2.

A. Front-End

Our implementation of the front-end consists of the third parts, feature extraction, feature matching and pose estimation. We are mainly using functions from OpenCV [7]. First, we extract ORB features which based on the FAST detector and the BRIEF descriptor proposed by Rublee et al [9], which can be determined landmarks by extracting descriptor vector from RGB image. Once we have the descriptors, feature matching will become a very critical port. Generally, it solves the data association for the landmarks by providing a measure for similarity in visual SLAM system. To match a pair of descriptors, we use Fast Library for Approximate Nearest Neighbors (FLANN) method [7] in case of large of matching point, it takes lower computation time than using brute-force matcher [7].

After we have feature matching results, we can utilize the widely known Random Sample Consensus (RANSAC) [8] for estimating ego-motion. Generally, the model is evaluated by measuring the error for each pose. Consequently, this separates the dataset into two subsets. The inliers can be fitting to the model and the outliers should be ignored. We also propose to use the keyframe to express the most representative frame for Back-End, loop closure, and mapping.

B. Back-End

The estimated ego-motion from front-end comes with an accumulating drift. The back-end of the SLAM system is dealing with the noise problem.

To minimize the error, the graph optimization bases on constraints between the nodes. We introduce the loop closure detection without making an assumption on the path. It is possible to check if the current frame matches with previous ones. The observation of a common point is seen in the past. It can trigger the new link between two poses that were separated. Once the graph has been initialized with the poses

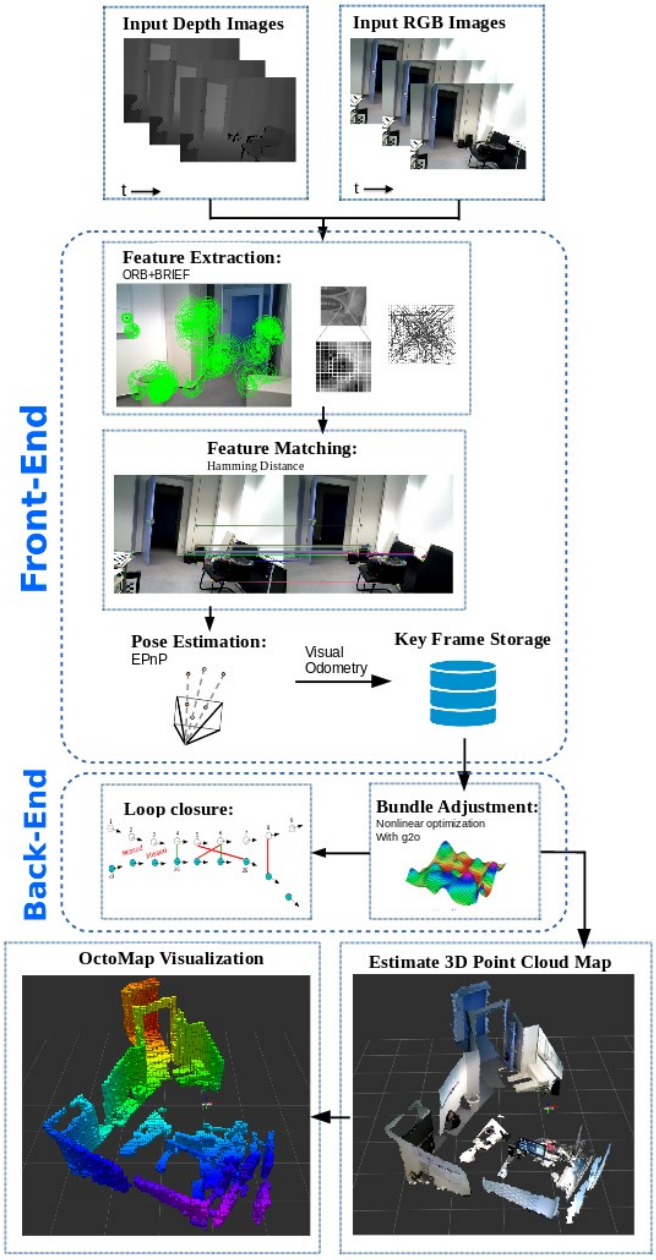
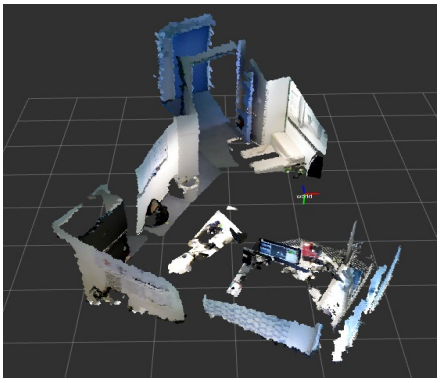


Fig. 2: System Architecture Diagram: Processing of the RGB-D data

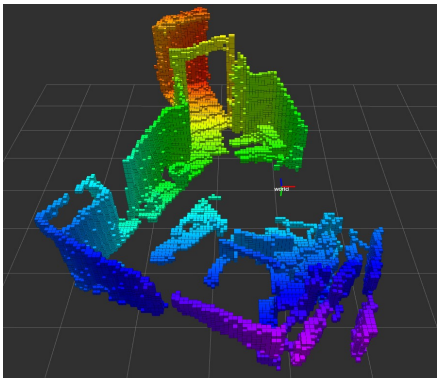
and the constraints from the loop closures, it can trigger the optimization. To resolve bundle adjustment, the VSLAM can be defined as a least squares optimization of an error function, and can be described by a graph model. We use g2o which is an open-source library for the optimization process and to minimize the error. This is the method chosen to solve the graph problem.

C. Mapping

At the end of 3D environment reconstruction, the overall map can be built from the sequence of data. We exploit two methods to represent the map, one is the 3D point cloud map, and other is the OctoMap which can be used to overcome the limitations of point cloud representation with reduced memory resource.



(a) Mapping after graph optimization



(b) Post-processing Octomap

Fig. 3: Map representation for our approach

We use the octree-based framework Octomap in an efficient tree structure that requires less memory consumption than PCL map, but the resolution of Map will also decrease. The figure illustrates how the RGB data and depth information can be used to compute the sequence of 3D transformations, and estimation of the robot poses. Subsequently, the system can be created both the Point-Cloud Map and OctoMap.

IV. Results

In this section, we show the preliminary experiential results illustrated in Fig.3. The data stream acquired from an RGB-D camera. Our approach computes the 6 DoF robot poses including trajectory and orientation and conduct a 3D map. The 3D reconstruction for indoor environment uses the Care-O-bot 4. The input data consisting of RGB and depth information for our approach is captured from an Asus Xtion camera, which is mounted on the robot. The map obtained from office and lab at Fraunhofer IPA. The preliminary results are running on XMG notebook with Intel Core i7-6820 4-cores. All software packages were developed using ROS indigo with Ubuntu 14.04, and OpenCV 2.

Fig 3(a) shows the result after graph optimization tracking created by point-cloud map with loop-closure. Thought the results are satisfying for small drift. Fig 3(b) shows the 3D Octomap after post-processing. The Octomap is valuable for exploration and robot navigation tasks.

V. Conclusion and Future Works

In this paper, we presented an approach for dealing with 3D environment reconstruction for mobile robot applications. We use a feature-based 3D SLAM approach with graph optimization to achieve the 3D reconstruction of an indoor environment. In the future, we are planning to combine the region based convolution neural network (MASK R-CNN) [11] for creating 2D semantic images and 3D semantic point cloud maps. The maps are allowed to provide more information for interacting with the indoor environment. Further results and experiments will be also integrated and tested with Care-O-bot 4.

Acknowledgment

The author has received funding from the European Union's Horizon 2020 framework programme for research and innovation under the Marie Sklodowska-Curie Grant Agreement No. 642667 (SECURE) and gratefully acknowledges the support.

References

[1] Hauke Strasdat, J. M. M. Montiel, and Andrew J. Davison. Scale Drift-Aware Large Scale Monocular SLAM. In Matsuoka, Yoky and Durrant-Whyte, Hugh F. and Neira, José, editor, Robotics: Science and Systems. The MIT Press, 2010.

[2] Kurt Konolige and Motilal Agrawal. FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping. IEEE Transactions on Robotics, 24(5):1066–1077, 2008.

[3] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2004

[4] A.J Davsion and I. Reid, et. al:"Real-tiem single camera SLAM", IEEE Transactions on pattern Analysis and Machine Intelligence, vol. 20, no. 6 pp. 1052-1067, 2007

[5] G. Klein and D. Murray, "Parallel tracking and mapping for small at workspaces," in Mixed and Augmented Realty, 2007. ISMAR 2007. 6th IEEE and ACM international Symposium on , pp.225-234, IEEE, 2007.

[6] R. Mur-Artal, J. Montiel, and J.D tardos, "Orb-slam: a versatile and accurate monocular slam system, " arXiv:1502.00956, 2015.

[7] G. Bradski and A. Kaehler. Learning OpenCV: Computer Vision with
the OpenCV Library. O'Reilly Media, 2008.

[8] D. Nister, "Preemptive RANSAC for live structure and motion estimation," in Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV), 2003

[9] C. Wu, "SiftGPU: A GPU implementation of scale invariant feature
transform (SIFT)," http://cs.unc.edu/~ccwu/siftgpu, 2007.

[10] A. W. Fitzgibbon, “Robust registration of 2d and 3d point sets,” Image Vision Comput., vol. 21, no. 13-14, pp. 1145–1153, 2003.
[11] K. He, G. Gkioxari, P. Dollar, R. Girshick, “Mask R-CNN, ” IEEE International Conference on Computer Vision, 2017.
[12] F. Endres, J. Hess, J. Sturm, D. Cremers, W. Burgard, “3D Mapping with an RGB-D Camera, ” IEEE Transactions on Robotics, vol. 30, no. 1, 2014.
[13] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “OctoMap: An efficient probabilistic 3D mapping framework based on octrees,” Autonomous Robots, 2013.
[14] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard, “g2o: A General Framework for Graph Optimization, ” IEEE International Conference on Robotics and Automation, pp. 3607–3613, 2011.
[15] Care-O-bot 4, https://www.care-o-bot.de/en/care-o-bot-4.html