CMPUT 631: Project Proposal

Performance Enhancement of RTAB-map Utilizing ORB-SLAM2 Depth Information

Islam Ali
iaali@ualberta.ca

Zifei Jiang
zifei.jiang@ualberta.ca

Abstract

SLAM (Simultaneous Localization and Mapping) became one of the main building blocks of any modern robotic system due to of the inevitable need for robust and reliable localization and mapping engine in the robotic perception subsystem. A number of trials have been done during the last few decades to provide reliable solutions but each suffered a downside in a certain domain or under certain environmental conditions. Two famous SLAM algorithms (namely RTAB-map and ORB-SLAM2) are considered the industry standard as they provide state-of-the-art performance. However, each suffer critical performance issues in some testing scenarios. In this work, we explore the possibility of enhancing the performance of RTAB-map by utilizing the depth information of ORB-SLAM2. The proposal starts by defining the problem in hand and by giving a quick background and literature review of the efforts exerted in this track. Then, the methodology and procedure are discussed thoroughly to provide a road map of the project activities. Finally, the proposal conclude by providing a mechanism for performance evaluation as well as discussing the significance of this work.

Problem Definition

Recently, modern autonomous robotic systems are playing a vital role in a wide spectrum of industrial and non-industrial applications. One problem that is related to such application is robot perception of the environment, as such how it models the environment and how it localize itself in it accurately and reliably. To answer these questions, SLAM was introduces a few decades ago as a method for simultaneous localization and mapping. Two of the major advancements in this field was introduced by the introduction of two state-of-the-art algorithms which are *RTAB-map* [1] and *ORB-SLAM* with its two versions [2][3].

Background

SLAM (Simultaneous Localization and Mapping) is the process of generating a model of the environment (a map) and localizing the camera inside it [4]. By localization we refer to the estimation of the 6DoF of the moving camera (orientation and position). Many sensors are used in SLAM such as LiDARs, Cameras, and Radars [5]. In this work, we focus on the usage of vision-based SLAM and possibly its integration with Laser-based SLAM.

The conventional pipeline of SLAM consists of two major stages, the first one is responsible for sensor abstraction and processing, which includes features extraction and tracking as well as any long-term data association steps such as bundle adjustment or loop closure [5][4]. The second step is to use the associated data to estimate a map or a model of the environment and, at the same time, estimate the 6DoFs (complete pose) of the camera inside the generated map reliably.

Literature Review

Visual SLAM

A wide spectrum of research has been conducted into the SLAM problem with the objective of enhancing its performance, allow for long-term operation, and enable portable and limited resources applications to utilize it in real-time. SLAM algorithms can be categorized to either feature-based method and direct methods [6]. Feature-based methods depend on extracting features and tracking them through different captured images. The estimation of the camera motion and the map is then determined using either filter-based methods (such as EKF in MonoSLAM) or using bundle adjustment optimization (such as BA used in PTAM). It was shown that BA method can provide better performance due to its ability to use more feature key points when compared to the complex EKF-based solution [7].

On the contrary, direct feature-less methods are usually used on images directly without any abstraction [6] in order to achieve real-time operation and eliminating the accumulated error usually occurring in the aforementioned feature-based methods. Direct methods differ, mainly, in the density of the map and can be categorized based on that to either dense, semi-dense, or sparse methods [6]. Methods such as DTAM and LSD-SLAM are examples of the trials done to achieve direct SLAM by means of dense and semi-dense maps while other methods such as SVO and DVO are more advanced steps towards fully direct SLAM algorithms utilizing spare maps [4][6].

RTAB-Map

RTAB-Map, (Real-Time Appearance-Based Mapping), is an open source Graph-Based SLAM system. RTAB-Map recently has been used as a platform to comparing various of SLAM algorithms due to its deep integration with ROS [1]. Based on ROS, developers are able to modify and extend the system easily. In the mean time, RTAB-Map provide two classic visual odometry algorithms Frame-To-Map(F2M) and Frame-To-Frame(F2F) [8]. F2F method registers the new frame against last frame while F2M registers the new frame against a local map features created from past key frames. The mapping process of RTAB-Map is separated from odometry part and to reduce the calculation burden. RTAB-Map create Point Cloud using the depth image directly without any optimization.

ORB-SLAM 1 & 2

ORB-SLAM 1 is a monocular SLAM system proposed in 2015 [2]. It use ORB features for all nodes in the SLAM system such as tracking, mapping, relocalization and loop closing. ORB features provide high calculation efficiency to enable the real time operation in large

Odometry	KITTI Sequence											time(msec)
	00	01	02	03	04	05	06	07	08	09	10	
F2F	1.4	14.5	4.7	0.4	0.2	0.72	1.8	0.6	5.8	2.2	3.0	61
F2M	1.0	4.7	4.7	0.3	0.2	0.5	0.8	0.5	3.8	2.8	0.8	82
ORB-RTAB	1.0	5.3	4.4	0.2	0.2	0.5	0.6	0.5	3.0	1.5	0.9	175
ORB-SLAM2	1.3	10.4	5.7	0.6	0.2	0.8	0.8	0.5	3.6	3.2	1.0	-

Table 1: Table 1. Baseline for KITTI Dataset

environment. Based on ORB-SLAM 1, ORB-SLAM 2 is developed to work with RGBD and stereo camera. ORB-SLAM 2 has three main parallel threads: tracking, local mapping and loop closing. Since ORB-SLAM 2 optimize both state of the camera and key points in the map, it achieves the state-of-art in accuracy in SLAM systems.

RTAB-Map vs. ORB-SLAM 2

A number of trials have been made to evaluate the performance of SLAM algorithms (specially RTAB-map and ORB-SLAM 1&2) utilizing unified test procedures.

Research Methodology and Procedure

Baseline Evaluation

The baseline is the accuracy of RTAB-Map and ORB-SLAM2 on KITTI and TUM datasets. Table 1 present some results from multiple papers. The time measurement depends on the computer configurations, thus we will measure the time again on our own configurations.

Depth Information Generation from ORB-SLAM2

RTAB-Map team incorporate ORB-SLAM2 by using ORB-SLAM2 as a odometry input, and disabling loop closing and full bundle adjustment of ORB-SLAM2. Local bundle adjustment of ORB-SLAM2 is still working, which make the modified ORB-SLAM2 similar to F2M.

Because ORB-SLAM2 optimize the feature points in the map, it can provide us with more accurate depth information for the feature points. In our project, we will utilize depth information after optimization, to perform a more accurate odometry estimation.

System Validation

In order to have a unified procedure for evaluating the system performance, publicly available datasets with ground truth are utilized to. The datasets were selected so that they can exploit the system boundaries in a wide spectrum of scenarios and conditions. For that purpose, KITTI dataset [9] was selected to cover scenarios in outdoor conditions with random moving objects, and TUM RGB-D dataset [10] was selected to cover more static and indoor environment conditions.

Performance Evaluation

The three corner stones of performance in computer science are accuracy, efficiency, and storage requirements. In this work, we focus on addressing *accuracy* aspects of the system which are defined by the relation between the generated trajectory and ground truth data.

The *Absolute Trajectory Error (ATE)* was defined in [11] as a performance metric to stand on the accuracy of a SLAM system vs. a ground truth. The ATE is defined by:

$$ATE(t_i) = \|(x^*_{t_i}, y^*_{t_i}) - (x_{t_i}, y_{t_i})\| \tag{1}$$

where $(x^*_{t_i}, y^*_{t_i})$ are the ground truth coordinates at t_i , and (x_{t_i}, y_{t_i}) are the coordinates generated by the SLAM algorithm under test at the same epoch. For a more detailed representation of this performance metric, statistical functions are applied to this metric such as Root Mean Square (RMS), mean, median, variance, and standard deviation. Additionally, the maximum and minimum ATE are reported to define the system limits while operating.

Significance of Proposed Research

As discussed in previous sections, both RTAB-map and ORB-SLAM2 are state-of-the-art algorithms and are able to provide accurate localization and mapping information under a number of conditions. Integration of both algorithms can extend the boundaries of each and can yield a more robust and stable SLAM system with the ability to work in even more challenging conditions. Such improvement can contribute in the all time quest of having an out-of-the-box SLAM system [4] that can be accurate and optimized for long-term operation.

References

[1] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[3] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[4] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[5] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[6] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual slam algorithms: a survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, p. 16, 2017.

[7] H. Strasdat, J. M. Montiel, and A. J. Davison, "Visual slam: why filter?," *Image and Vision Computing*, vol. 30, no. 2, pp. 65–77, 2012.

[8] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part ii - matching, robustness, and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.

[9] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[10] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 573–580, IEEE, 2012.

[11] M. Filipenko and I. Afanasyev, "Comparison of various slam systems for mobile robot in an indoor environment," in *2018 International Conference on Intelligent Systems (IS)*, pp. 400–407, Sep. 2018.