Coupling of localization and depth data for real-time mapping

Evgeny Tsykunov*, Valery Ilin†, Stepan Perminov‡, Aleksey Fedoseev§, Elvira Zainulina¶
Perception in Robotics course, Skoltech, Moscow, Russia

Emails: *evgeny.tsykunov@skolkovotech.ru, †valery.ilin@skoltech.ru, ‡stepan.perminov@skoltech.ru, §aleksey.fedoseev@skoltech.ru, ¶elvira.zainulina@skoltech.ru

***Abstract*—We propose to couple two types of Intel RealSense cameras (tracking T265 and depth D435i) in order to build an online 3D occupancy mapping of the indoor environment.**

***Index Terms*—Mapping, Navigation, Cameras, Robot vision systems**

I. Objective

Create a 3D occupancy map in real-time using combination of tracking and depth cameras data. The problem can be referenced as a mapping with known poses [1].

II. Introduction

Our motivation is to navigate an autonomous robot in unknown environment. Robot needs to get a map in order to perform planning. For this we are going to develop an algorithm that will fuse data from 2 different Intel RealSense cameras into 3D occupancy map, such as OctoMap [2]. We propose to generate 3-dimensional rather than 2-dimensional map in order to make the solution more general and fit more number of applications (from aerial robotics to underwater robots). In order to fulfil that, we do not set any other constraints such as limit or allowed computational power.

In contrast to [2], Guizilini ae al. in [3] proposed to create a continuous occupancy map using Hibert maps in real-time. Another existing technology that generates height map for walking robot is presented in paper of Bayer et. al. [4]. The height map is more simple to build than 3D occupancy map like OctoMap [2], therefore it can be accomplished on a lightweight sing-board computer in real time.

In order to be able to navigate a robot in real time we need to update the map fast enough [3]. That is extremely important for robots with fast dynamics, such as drones. In order to meet this requirements, we propose to implement mapping using acceleration with CUDA GPU. In the work of Pan et. al. [5] presented approach that allows to make GPU acceleration and multi-threaded parallelism for map calculation. In their approach the used data from LiDAR sensor, so our final solution may have a different approach.

III. Methodology

Our preliminary algorithm is provided bellow:

1) Literature review for real-time methods and libraries for occupancy mapping;
2) Extracting of the tracking (pose) data from T265 and depth data from D435 Intel RealSense cameras;
3) Synchronizing the data from cameras (by timestamps or other method);
4) Build an algorithm (GPU accelerated) that is able to constantly update a prior occupancy map in real-time using the fusion of current data from depth and tracking cameras.
5) (Optional) SLAM: To update a localisation data (treat it as an odometry) from T265 with a depth data from D435i.
6) (Not sure about this – from optional) Trajectory planning: make a recommendations to robot, how avoid collisions.

Project hardware: T265 & D435i Intel RealSense cameras, Jetson Nano.

Project software: Open source python/C++ libraries (Open3D, Octomap, PCL, etc.)

Project data-set: collected indoor data-set from the cameras.

IV. Milestones Corresponding dates

Preliminary Project Pipeline:

	Actions	Deadlines
1	Learning of how to work with sensors, which data to obtain	13/03/2020
2	Learn how to synchronize the data from sensors	16/03/2020
3	Literature review, collection of data-set from sensors	17/03/2020
4	Try different open source libs (Open3D, Octomap, PCL, etc.)	18/03/2020
5	Develop a mapping algorithm	20/03/2020
6	Accelerate the mapping algorithm with GPU	22/03/2020
7	Optional task: SLAM algorithm	25/03/2020
8	Make a presentation	25/03/2020
9	Final project presentation	26/03/2020

V. Optional work: tasks after the completion of milestones

- Make a localisation from 2 RealSense cameras (using raw IMU data): but we actually already have precise sensor for this.
- Accelerate the map exploration of the robot by using auto-encoders/GANs that complete map and predict its optimal trajectory [6].
- Make a collision avoidance algorithm: interesting task for us, not a goal of the course

VI. Current work and discussions

Current status:

1) Currently we can get data from our sensors synchronously. To achieve this we build Python application based on Observer programming pattern and multithread approach. Code is presented by link[1].

2) We collected a dataset from both sensors and used it in current experiments.

3) We visualized camera frames and poses and computed a set of transformation matrices using data from T265 sensor. To achieve that we transformed obtained unit-norm quaternions to rotation matrices and combined them with corresponding transition vectors.

4) We visualized and tried to analyse points and images from D435 sensor. We tried to map points from one frame to another with SIFT algorithm. It helps to detect key-points of the frame and derive its transformation. As there are outliers in the mapping its improvement can be done by RANSAC algorithm. Transformation matrices can help to restore map in the initially defined coordinate system.

Work in the coming days:

1) For the mapping we are going to start from simple joint point cloud representation. Probably then someone will implement OctoMap or other more advanced representation of 3D data.

2) Finally we are going to try to optimize the trajectory from T265 sensor and build a map at the same time. We are going to do that in the following way. Every time when we register a new point cloud from D435 we use a transformation matrix from T265 as an initial guess for optimization. T265 provides good initial guess, that is why we are able to implement computationally intensive and precise methods for point cloud alignment. As a result we can get a good update of the transformation matrix, like in PS4 Task 2, which can be used for optimization of the initial trajectory received from T265 sensor.

References

[1] F. Homm, N. Kaempchen, J. Ota, and D. Burschka, "Efficient occupancy grid computation on the GPU with lidar and radar for road boundary detection," in *2010 IEEE Intelligent Vehicles Symposium*, IEEE, jun 2010.

[2] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.

[3] V. Guizilini and F. Ramos, "Towards real-time 3d continuous occupancy mapping using hilbert maps," *The International Journal of Robotics Research*, vol. 37, pp. 566–584, may 2018.

[4] J. Bayer and J. Faigl, "On autonomous spatial exploration with small hexapod walking robot using tracking camera intel realsense t265," in *2019 European Conference on Mobile Robots (ECMR)*, pp. 1–6, IEEE, 2019.

[5] Y. Pan, X. Xu, Y. Wang, X. Ding, and R. Xiong, "Gpu accelerated real-time traversability mapping," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 734–740, IEEE, 2019.

[6] R. Shrestha, F. Tian, W. Feng, P. Tan, and R. Vaughan, "Learned map prediction for enhanced mobile robot exploration," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 1197–1204, May 2019.

[1] https://github.com/IlinValery/perception_final_2020