Otto-von-Guericke-University Magdeburg

Faculty of Computer Science

Institute for Intelligent Cooperating Systems

Comparison of Real-Time Plane Detection Algorithms on Intel RealSense

Bachelor Thesis

Author:

Lukas Petermann

Examiner:

Prof. Frank Ortmeier

2nd Examiner

M.Sc. Marco Filax

Supervisor:

M.Sc. Maximilian Klockmann

Magdeburg, 32.13.2042

Contents

1 Introduction

2 Background

"Summary", What is to be expected in this chapter Propably a bad idea to have started with this before i finished reviewing the literature. In this chapter we will present relevant literature needed to completely understand the proposed concept of chapter 3.

2.1 SLAM

Simultaneous Localization and Mapping (SLAM) is an important research topic in the field of modern robotics. As the name name suggests, SLAM consists of two main components, namely Localization and Mapping. Localization on its own is the problem of estimating the robot's location, while the process of building a model of the environment can be described as Mapping. Therefore, solving the SLAM problem includes a robot moving through an environment and building a map while correctly estimating its own location.

Why SLAM is needed

No matter if its a car needing to be able to safely navigate urban and dynamic areas, full of pedestrians and other cars, or a robotic vacuum cleaner that has to clean small, indoor areas, being able to sufficiently solve the SLAM problem at hand is a necessary step for the robot to become fully autonomous [3].

How SLAM works

State of the Art

"Summary", What is to be expected in this chapter

2.2 Plane Description

2.2.1 Formulas and what not

2.3 General approaches of PDA

3 types of approaches [6, 8]

RANSAC

What is Ransac, how does it work RANdom SAmple Consensus (RANSAC)[4] is a very popular approach used for the parameter estimation of mathematical models in data sets with a variable number of outliers [14, 12, 11]. The RANSAC algorithm consists of two stages. First, a hypothesis is generated by taking a random subset of data and fitting the desired model to it. In the context of plane detection, a minimum of three points need to be sampled [14]. The next step is used to verify the previously calculated model. This is done by calculating both the number of points that are consistent with the model, henceforth called inliers, and the number of points that are considered outliers, i.e. points that do not lie within a certain threshold of the model. If the amount of inliers from the current model exceeds the amount of inliers of the previously best-fitting model, both the model and the amount of associated inliers are updated. The described algorithm can be found in Algorithm 1.

Algorithm 1: RANSAC

Data: $P \neq \emptyset$, $N > 0$
Result: best fitting model m_{best}
$m_{best} \leftarrow \emptyset$;
$i_{max} \leftarrow 0$;
while $N > 0$ **do**
    $p \leftarrow$ random sample of P;
    $m \leftarrow$ fit model to p;
    $i \leftarrow$ amount of inliers in m;
    if $i > i_{max}$ **then**
        $i_{max} \leftarrow i$;
        $m_{best} \leftarrow m$;
    end
end

Ransac in the context of plane detection

Hough Transform

What is HT, how does it work generally? The Hough Transform is used for the detection of parameterized objects in two- or three-dimensional space.

HT in the context of Plane Detection (i.e. 3D HT)

Region Growing

I am not happy with this

what is RG, how does it work Originally, Region Growing (RG) has been introduced as a method of image segmentation [1]. The only necessary input to perform region growing is a set of pre-determined seeds, which can be chosen manually or by automated procedures. The general idea of the Region Growing algorithm is to gradually increase the size of regions by appending adjacent pixels depending on a membership criterion. In the original implementation, an adjacent pixel x is only added to a region if all labelled neighbors of this pixel have the same label. After that, the mean gray-scale value of the entire corresponding region is updated. Lastly, all adjacent pixels of x whose pixel value is also within the predetermined threshold or are still unset are added to the list.

3 Concept

Description of following Chapter. (Structure and what not. Write this after the rest is in place.)

3.1 Problem Description

"We want to find planes quickly, etc. Also, we have the following requirements: ...

3.2 SLAM

Since we use CAMERA and want REQ, we have a couple options... These are ... We choose X because...

3.3 Plane Detection

SLAM returns OUTPUT, thus we have these options ... [explain] ... we choose these because

3.4 Plane Description

We want to use found planes, we have options of how to describe them, we use X because

3.5 Summary

to solve PROBLEM we use SLAM, compare PDAS with respect to REQUIREMENTS

4 Implementation

What this chapter will cover In this chapter we will cover the realization of the aforementioned concept in chapter 3.

4.1 Used Sensors

There exists a broad range of sensors applicable for SLAM algorithms. Depending on the specific algorithm, Lidar [2], monocular [9] stereo [13] or even a combination of multiple Cameras can be integrated [5]. Different types of cameras ultimately lead to different kinds of input. Lidar, for example, returns dense Point Clouds, whereas a RGB-D camera would return colorful images. Of course, cameras are not the only sensors used in SLAM algorithms. Many systems make use of an inertial measurement Unit (IMU) [7, 10]

For this work we will be using the Intel RealSense T256[1] tracking camera as well as the Intel RealSense D455[2] depth camera. Not only do both have a built-in IMU, they also both have two imagers, which classifies them as stereo cameras. Another advantage of combining a fish-eye tracking camera (T256) with a RGB-D camera (D455) is that they support each other in situations a robot with only one of them would be unable to handle well.

4.2 Architecture

ROS

librealsense

We integrate RTAB-MAP into our architecture like this:...

[1]https://www.intelrealsense.com/tracking-camera-t265/
[2]https://www.intelrealsense.com/depth-camera-d455/

5 Evaluation

Introduction

Description of following Chapter. (Structure and what not. Write this after the rest is in place.)

5.1 Experiments

We previously compiled a list of algorithms to compare under also previously defined Requirements. To be able to precisely compare them, we do $STUFF. We perform the experiments on $DATASETS To visualize results, we integrate RVIZ in our ROS Architecture.

5.2 Results

We conducted experiments for each selected algorithm. In this section, we will report and thoroughly analyze the results for each experiment with respect to the metrics we introduced previously in 3.

5.2.1 $ALG1

5.2.2 $ALG2

5.2.3 $ALG3

Summary

5.3 Summary

This chapter introduced experiments necessary for verification of the problem this paper is dedicated to solve. We derived and analysed results in 5.2.

6 References

Bibliography

[1] R. Adams and L. Bischof. "Seeded region growing". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16.6 (June 1994), pp. 641–647. ISSN: 1939-3539. DOI: 10.1109/34.295913.

[2] David Droeschel and Sven Behnke. "Efficient Continuous-Time SLAM for 3D Lidar-Based Online Mapping". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 5000–5007. DOI: 10.1109/ICRA.2018.8461000.

[3] H. Durrant-Whyte and T. Bailey. "Simultaneous localization and mapping: part I". In: *IEEE Robotics Automation Magazine* 13.2 (June 2006), pp. 99–110. ISSN: 1558-223X. DOI: 10.1109/MRA.2006.1638022.

[4] Martin A. Fischler and Robert C. Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Commun. ACM* 24 (1981), pp. 381–395.

[5] Adam Harmat, Inna Sharf, and Michael Trentini. "Parallel Tracking and Mapping with Multiple Cameras on an Unmanned Aerial Vehicle". In: *Intelligent Robotics and Applications*. Ed. by Chun-Yi Su, Subhash Rakheja, and Honghai Liu. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 421–432. ISBN: 978-3-642-33509-9.

[6] Rostislav Hulik et al. "Continuous plane detection in point-cloud data based on 3D Hough Transform". en. In: *Journal of Visual Communication and Image Representation* 25.1 (Jan. 2014), pp. 86–97. ISSN: 10473203. DOI: 10.1016/j.jvcir.2013.04.001.

[7] Stefan Leutenegger et al. "Keyframe-Based Visual-Inertial SLAM using Nonlinear Optimization". en. In: *Robotics: Science and Systems IX*. Robotics: Science and Systems Foundation, June 2013. ISBN: 978-981-07-3937-9. DOI: 10.15607/RSS.2013.IX.037. URL: http://www.roboticsproceedings.org/rss09/p37.pdf.

[8] Frederico A. Limberger and Manuel M. Oliveira. "Real-time detection of planar regions in unorganized point clouds". en. In: *Pattern Recognition* 48.6 (June 2015), pp. 2043–2053. ISSN: 00313203. DOI: 10.1016/j.patcog.2014.12.020.

[9] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Transactions on Robotics* 31.5 (Oct. 2015), pp. 1147–1163. ISSN: 1941-0468. DOI: 10.1109/TRO.2015.2463671.

[10] Raúl Mur-Artal and Juan D. Tardós. "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras". In: *IEEE Transactions on Robotics* 33.5 (Oct. 2017), pp. 1255–1262. ISSN: 1941-0468. DOI: 10.1109/TRO.2017.2705103.

[11] Bastian Oehler et al. "Efficient Multi-resolution Plane Segmentation of 3D Point Clouds". en. In: *Intelligent Robotics and Applications*. Ed. by Sabina Jeschke, Honghai Liu, and Daniel Schilberg. Vol. 7102. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 145–156. ISBN: 978-3-642-25488-8. DOI: 10.1007/978-3-642-25489-5_15. URL: http://link.springer.com/10.1007/978-3-642-25489-5_15.

[12] Bo Sun and Philippos Mordohai. "Oriented Point Sampling for Plane Detection in Unorganized Point Clouds". en. In: arXiv:1905.02553 (May 2019). arXiv:1905.02553 [cs]. URL: http://arxiv.org/abs/1905.02553.

[13] Vladyslav Usenko et al. "Direct visual-inertial odometry with stereo cameras". en. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. Stockholm: IEEE, May 2016, pp. 1885–1892. ISBN: 978-1-4673-8026-3. DOI: 10.1109/ICRA.2016.7487335. URL: http://ieeexplore.ieee.org/document/7487335/.

[14] Michael Ying Yang and Wolfgang Forstner. "Plane Detection in Point Cloud Data". en. In: (), p. 16.