FAKULTÄT FÜR
INFORMATIK

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

SLAM ALgos includes table of different ones

PDAs includes table of different ones

DataSets includes table of different ones

3 Concept

Introduction

What this chapter is about This chapter covers...

Chapter Structure It is Structured as follows ...

3.1 Used Sensors

Why these ones

open source ros wrapper realsense-ros

Detailed Information i guess?

3.2 SLAM

Why SLAM is needed

map building reduces complexity for plane detection

SLAM SOTA

RTAB-MAP, ORBSLAM etc..

bc we are using visual, stereo etc.

We choose RTAB-MAP bc.. just makes sense because its already included in realsense-ros and also has a wide variety of options regarding output formats

3.3 Requirements

Problem/Use case We want to find planes quickly..

Definition Real-time

Because camera runs at 30fps .. Daher definieren wir in dieser arbeit "echtzeit" als

Precision

What does precision mean in general
What is a "good" precision Werden wir $später definieren fokus wird auf echtzeit gelegt wird und wir gucken was wir im rahmen der echtzeit im besten fall an präz raus holen können dazu sind allein die angegeben werte nicht 100%ig aussagekräftig weil größtenteils verschiedene datensätze benutzt wurden somit werden wir in dieser arbeit einen einheitlichen vergleich von PDAs unter besonderer berücksichtigung des echtzeitkriteriums machen

3.4 PLane Detection Algorithms

Intro?

SOTA aufgrund von $SACHEN filtert sich die gesamte liste runter auf diese enge auswahl $DARUM werden wir den fokus auf diese N algorithmen legen

3.5 Summary

we want to answer the question if real time plane detection is viable on OTS hardware like the intel realsense. to answer the q, we selected $SLAM which gives us $OUTPUT which we use as input for out $PDAs

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 [5] stereo [7] or even a combination of multiple Cameras can be integrated [3]. 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) [4, 6]

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

In diesem kapitel werden zuvor ausgewählte algorithmen einheitlich verglichen und die resultierenden ergebnisse ausgewertet.

5.1 Evaluation Protocol

Das Ziel der Arbeit ist es, die Frage zu beantworten, ob präzise Ebenenfindung in Echtzeit möglich ist. Das Problem bei der Auswahl des "besten" algos ist, dass jeder Algorithmus auf anderen Daten getestet wurde, was die Vergleichbarkeit dezimiert. Daher wird diese Evaluation auf einem öffentlich erhältlichen Datensatz ausgeführt, sowie einem "live" test unterzogen. Der Datensatz besteht aus einer Anzahl indoor Aufnahmen mit dazugehöriger Ground Truth, beides in Punktwolken-Form.

grundlegend:
- Problem: niemand nimmt den gleichen Datensatz
- daher alle aufm selben testen für einheitlichen vergleich
- quantitativ: datensatz+GT
- qualitativ: "live" test in der FIN, überlagern der ebenen mit punktwolke
- diese metriken sind wichtig:

5.1.1 Metrics

grundlegend:
- Laufzeit in relation zu len(PC) und #planes
- Präzision, Recall: "best match" berechnen. wie? i guess über MSE, wenn ich eh die cornerpoints habe. Dafür muss ich jedoch eine maximale entfernung angeben, bis eine ebene kein match hat
- f1 i guess, tut genauso weh, es raus zu lassen, wie eine weitere spalte in die tabelle zu klatschen. man kann es ja eh aus P und R berechnen
- MSE: durchschnittlicher abstand der cornerpoints

Für die Bewertung eines Algos werden zum einen verschiedene Laufzeiten betrachtet. Wie lange dauert das finden von Ebenen in relation zur Größe der Punktwolke, wie lange dauert es pro Ebene.

Dazu werden wir die Qualität der Algorithmen bewerten. Dafür berechnen wir die *precision*, *recall*, den *f1-score* sowie eine Durchschnittliche Abweichung der Ebene von der gegebenen Ground truth über den *Mean-Squared-Error*(MSE).

Erklärung der Metriken im Background nehme ich an? Um eine gefundene Ebene einer Ebene der Ground truth zuzuordnen wird wird einfach die ebene mit der geringsten Abweichung (geringster MSE) ausgewählt. Ist der MSE aller gefundenen Ebenen zu groß, wird die bestimmte Ebene als "nicht gefunden" gewertet.

Was, wenn zu viele gefunden wurden? 1. wurden N ebenen doppelt gefunden oder Falsche ? Daraus ergeben sich die Werte für *precision*, *recall* und *f1-score*.

Im anschluss betrachten wir das Maß an Korrektheit der Anordnung innerhalb der Scene. Im grunde berechnet sich dieses Maß erneut durch den MSE. quasi 3d variante zu IoU

5.1.2 Dataset

Wir benutzen den *Stanford Large-Scale Indoor Spaces 3D Dataset* (S3DIS)[1]. Der Datensatz beinhalt viele verschiedene Räume aus verschiedenen Gebäuden. Diese Indoor Umgebungen werden als Punktwolken dargestellt. Dazu werden Ground truths in form von annotierten punktwolken bereitgestellt, welche in insgesamt 12 Kategorien fallen. Da für diese arbeit allein die Ebenen interessant sind, werden wir aus diesen GTs manuell planare Strukturen extrahieren und durch ein convexes Polygon(meistens 4 Ecken?) beschreiben.

5.1.3 Real-Life Test

Bei dem Live test wird im Gebäude der FIN ein Datensatz aufgenommen. Der Scan wird $STUFF beinhalten. Zu diesem Scan gibt es keine Ground Truth, daher wird neben der Laufzeitanalyse lediglich eine qualitative Analyse der Präzision vorgenommen. Dafür überlagern wir die Punktwolke mit den gefundenen Ebenen.

5.2 Results

5.2.1 Results Dataset

Alle N Sequenzen des Datensatzs wurde X mal von jedem Algorithmus berechnet.

Hier sind die Ergebnisse:

5.2.2 Results Real-Life Test

Der FIN Datensatz wurde auch X mal von jedem Datensatz berechnet.

Hier sind die Ergebnisse:

6 Conclusion

Possibly redundant with results section in evaluation

7 References

Bibliography

[1] Iro Armeni et al. "3D Semantic Parsing of Large-Scale Indoor Spaces". In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. 2016.

[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] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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/.