



OTTO VON GUERICKE  
UNIVERSITÄT  
MAGDEBURG

INF

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

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Background</b>	<b>4</b>
<b>3</b>	<b>Concept</b>	<b>5</b>
3.1	Used Sensors . . . . .	5
3.2	SLAM . . . . .	5
3.3	Requirements . . . . .	6
3.4	PLane Detection Algorithms . . . . .	6
3.5	Summary . . . . .	6
<b>4</b>	<b>Implementation</b>	<b>7</b>
4.1	Used Sensors . . . . .	7
4.2	Architecture . . . . .	7
<b>5</b>	<b>Evaluation</b>	<b>8</b>
5.1	Protocol . . . . .	8
5.2	Metrics . . . . .	8
5.3	Dataset . . . . .	9
5.4	Real-Test? . . . . .	9
5.5	Results Dataset . . . . .	9
5.6	Results FIN? . . . . .	9
<b>6</b>	<b>Conclusion</b>	<b>10</b>
<b>7</b>	<b>References</b>	<b>11</b>
	<b>Bibliography</b>	<b>12</b>

# 1 Introduction

## 2 Background

**SLAM ALgos** includes table of different ones

**PDA**s 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 [1], monocular [4] stereo [6] or even a combination of multiple Cameras can be integrated [2]. 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) [3, 5]

For this work we will be using the Intel RealSense T256<sup>1</sup> tracking camera as well as the Intel RealSense D455<sup>2</sup> 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:...**

---

<sup>1</sup><https://www.intelrealsense.com/tracking-camera-t265/>

<sup>2</sup><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 Protocol

Wir evaluieren die Auswahl an PDAs auf einem Datensatz. Der Datensatz wird aus aufnahmen einer indoor umgebung bestehen, dazu muss eine ground truth dazu verfügbar sein. (Da nicht alle PDAs auf dem Selben Datentypen als input arbeiten, muss der Datensatz teilweise in ein anderes Format überführt werden.)

## 5.2 Metrics

Für den Vergleich muss erst geklärt werden, welche Metriken für den vergleich betrachtet werden.

Natürlich ist die benötigte Laufzeit von Interesse. Wir werden also die PDAs auch in der Zeit, die sie benötigt haben um die Ebenen zu finden vergleichen.

Wir wollen dazu ebenso die Präzision der PDAs evaluieren. Dazu vergleichen wir die gefundenen Ebenen mit den Ebenen der GT: Wir suchen für jede GT Ebene ein "best match", sobald vorhanden. Das beinhaltet Differenz der Rotation, Translation und den Flächeninhalt zwischen den Ebenen.

Weichen alle gefundenen Ebenen zu stark von einer Ebene der GT ab, so wird diese als "nicht gefunden" gewertet. Andernfalls gibt es ein "best match" und diese Ebene wird als "gefunden" gewertet. Aus *#gefunden* und *#nichtGefunden* lassen sich die Präzision, sowie Recall und der F1-score berechnen.

Wir werden dazu noch die Intersection-over-Union für jede richtig gefundene Ebene berechnen. Dafür muss ich jedoch einen weg finden, wie ich translation und rotation ausklammern kann. Oder: muss ich die ausklammern? Reicht hier ggf sogar eine qualitative auswertung?



## 5.3 Dataset

disclaimer: meine 2 favoriten der datensätze sind der kinect DS und der SR400. Problem: ich war bisher zu blöd um das SegComp tool zu nutzen. Der ausgewählte Datensatz besteht aus 30 Punktwolken (Kinect) oder 5 Tiefenbildern Problem: (SR400). Aufgenommen wurden indoor scenen und jeder frame besteht aus 307.200 Punkten/640x480 Pixeln. Da nicht alle algorithmen Punktwolken oder respektiv Tiefenbilder nutzen, werden die Daten verlustlos umgewandelt. bei der umwandlung muss die GT ggf angepasst werden, entscheiden nachdem ich ergebnisse gesehen habe

## 5.4 Real-Test?

Um das Echtzeitkriterium genauer zu betrachten nehmen wir einen Datensatz in der fin auf. Da hierbei keine GT vorliegt beschränken wir uns hierbei auf eine qualitative Auswertung mit besonderem fokus auf Laufzeit.

hier beschreiben wie der in der fin aufgenommene datensatz aussieht? Ich muss dann wahrscheinlich manuell annotieren, wie ebenen in den bildern liegen.

meine idee hier ist, mit dem datensatz zu zeigen, dass die Präzision ausreichend ist und dass durch die FIN gezeigt wird, dass das ganze echtzeitfähig ist. Sogesehen statisch für Präzision und dynamisch für laufzeit.

## 5.5 Results Dataset

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

Hier sind die Ergebnisse:

## 5.6 Results FIN?

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] David Droschel 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.
- [2] 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.
- [3] 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>.
- [4] 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/TR0.2015.2463671.
- [5] 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/TR0.2017.2705103.
- [6] 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/>.