Höhere Technische Bundeslehranstalt
und Bundesfachschule
im Hermann Fuchs Bundesschulzentrum

Autonomous Car Mapping and Tracking

Diploma Documentation

School autonomous focus on Mobile Computing and Software Engineering

Performed in school year 2019/2020 by:

Alexander Voglsperger (AV), 5AHELS

Simon Moharitsch (SM), 5AHELS

Advisors:

Dipl. Ing. Müller Gerhard

April 3, 2020

Thema:
Autonomous Car Mapping and Tracking

Subtopics and Editor:

Implementing SLAMS and DeepTAM, Image Pre-Processing
Alexander Voglsperger, 5AHELS
Advisors: Dipl. Ing. Müller Gerhard

Implementing DeepTAM, Gathering Trainingdata
Simon Moharitsch, 5AHELS
Advisors: Dipl. Ing. Müller Gerhard

Projectpartner:

Designation:	Johannes Kepler University - Artificial Intelligence Lab
Address:	Altenberger Straße 69
ZIP, location:	4040 Linz, Austria
Contact person:	Dr. Nessler Bernhard
Phone:	+43 (0)732 2468 4539
E-Mail:	nessler@ml.jku.at

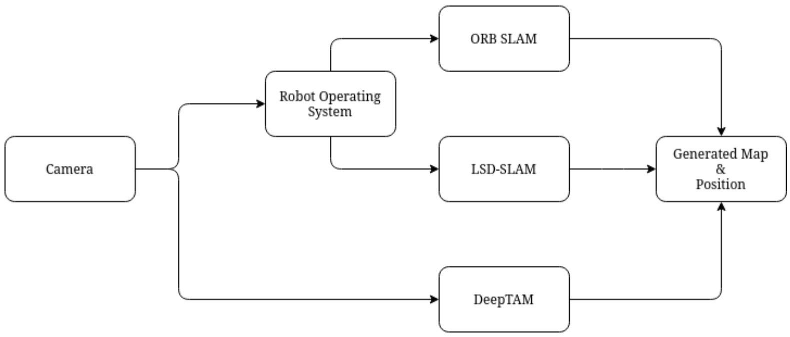
Höhere technische Bundeslehranstalt
und Bundesfachschule Braunau
Elektronik und Technische Informatik
School autonomous focus on Mobile Computing
and Software Engineering

DIPLOMA DOCUMENTATION

Author	Alexander Voglsperger, Simon Moharitsch
Vintage School year	5AHELS 2019/2020
Topic of the diploma documentation	Autonomous Car Mapping and Tracking
Cooperation-partner	Johannes Kepler University - Artificial Intelligence Lab
Task definition	The task of this work is to get depth information out of a video stream in real time. The considered field of application is an autonomous driving car that uses several cameras to orientate itself while driving. In a first approach the Robot Operating System (ROS) sends the video of a camera to two Simultaneous Localization And Mapping (SLAM) algorithms which gather depth information out of the images. This results in a point cloud which represents the detected surroundings. Another approach is DeepTAM, a method which implements artificial intelligence methods to estimate distances between objects in each frame of a video stream. As DeepTAM has not been updated for a while there are many compatibility issues with newer drivers, newer Tensor Flow framework and other required libraries. Thus the task is to fix the compatibility issue.
Realization	Simultaneous Localization And Mapping (SLAM) algorithms are implemented on the free available platform Robot Operating System (ROS). ORB-SLAM and LSD-SLAM are two specific SLAM methods. These two SLAMs have been modified to get executable program code on ROS. Several performance tests and a comparison between ORB-SLAM and LSD-SLAM are also part of this work.
Outcome	In this thesis the results of ORB-SLAM and the LSD-SLAM are compared. The ORB-SLAM is working, but will not produce a detailed map if prominent points are missing in the video. Prominent points are characterized by sharp edges or corners of an object. SLAM methods use these prominent points to get the depth information in a 2D video stream. The LSD-SLAM does not work as good as the ORB-SLAM when the camera only has a axial movement in the sense that it requires both axial and rotational movement to work and thus would generate an incorrect map. Otherwise the LSD-SLAM only works on cars when a wide-angle camera lens is used. DeepTAM works on Ubuntu 16.04. Since this is not up to date, most libraries need to be downgraded, which creates driver problems. To get DeepTAM installed on the current version of Ubuntu, the source code of DeepTAM has to be adapted.

Illustrative graph, photo (incl. explanation)

Structure of the flowchart:



Accessibility of diploma thesis

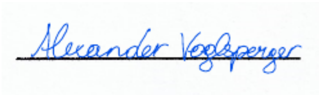
HTL Braunau archive, or `https://diplomarbeiten.berufsbildendeschulen.at/`

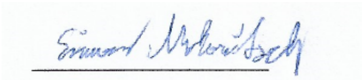
Approval (date / signature)

Examiner	Head of College / Department

Statement

I declare in lieu of oath that I have written this diploma thesis independently and without outside help, have not used sources and aids other than those stated directly and have made the sources used verbatim and in terms of content taken as such recognizable.

Braunau/Inn, 03.04.2020	Alexander Voglsperger	
Location, Date	*Author*	*Signature*

Braunau/Inn, 03.04.2020	Simon Moharitsch	
Location, Date	*Author*	*Signature*

Contents

Abstract x

Summary xi

1 SLAM[AV] 1
1.1 What is SLAM? . . . 1
1.2 Application . . . 1
1.3 History . . . 1
1.4 Existing Methods . . . 2

2 Robot Operating System[AV] 3
2.1 What is the Robot Operating System? . . . 3
2.2 Design . . . 3
2.2.1 Topics . . . 4
2.2.2 Nodes . . . 4
2.2.3 Services . . . 4
2.2.4 Parameter server . . . 5
2.3 Licenses and OS . . . 5
2.4 Tools . . . 5
2.4.1 Rosbag . . . 5
2.4.2 RQt . . . 5
2.4.3 CatKin . . . 5
2.4.4 Rviz . . . 6
2.4.5 Roslaunch . . . 7

3 Artificial Neural Networks[SM] 8
3.1 What is a Artificial Neural Networks? . . . 8
3.2 Areas of Application . . . 8
3.3 Components of an ANN . . . 8
3.3.1 Neurons . . . 8
3.3.2 Connection and weights . . . 8
3.3.3 Propagation function, activation function and Bias . . . 9
3.4 Organization . . . 9
3.4.1 Feed Forward ANN . . . 9
3.4.2 CNN . . . 10
3.5 Encoder-Decoder-Based Architecture . . . 11

4 Deep Learning[SM] 12
4.1 What is Deep Learning? . . . 12
4.2 Supervised Learning . . . 12
4.3 Semi-Supervised Learning . . . 12

4.4 Unsupervised Learning . . . 12
4.5 Applications . . . 12

5 Basler Camera^AV . . . 13
5.1 Information . . . 13
5.2 Prerequisites . . . 13
5.3 Running pylon-ros-camera node . . . 13
5.3.1 Compiling node . . . 14
5.3.2 Using Pylon-Ros-Camera Node . . . 14

6 ORB-SLAM2^AV . . . 15
6.1 What is ORB-SLAM? . . . 15
6.2 How does the ORB-SLAM work? . . . 15
6.2.1 Extracting Keypoints . . . 15
6.2.2 Loop-closing and Bundle Adjustments . . . 16
6.2.3 Localization . . . 17
6.2.4 Input/Output . . . 17

7 LSD-SLAM^AV . . . 18
7.1 What is LSD-SLAM? . . . 18
7.2 Difference Feature-Based and Direct . . . 18
7.3 How does the LSD-SLAM work? . . . 19
7.3.1 Components that make up the LSD-SLAM . . . 19
7.3.2 Depth Map Estimation . . . 19
7.3.3 Map optimization . . . 20
7.3.4 Input/Output . . . 20

8 DeepTAM^SM . . . 21
8.1 What is DeepTAM? . . . 21
8.2 Tracking . . . 21
8.2.1 Network Architecture . . . 21
8.3 Mapping . . . 22
8.3.1 Network Architecture . . . 23
8.3.2 Training . . . 23

9 Workflow . . . 24
9.1 Used Hardware^AV . . . 24
9.1.1 Raspberry Pi . . . 24
9.1.2 PC . . . 25
9.2 Used Software^AV . . . 25
9.3 Setup^AV . . . 26
9.4 Streaming video from Pi to PC^AV . . . 26
9.4.1 Enabling Camera . . . 26
9.4.2 Python Script . . . 26
9.5 Receiving images on PC and Laptop^AV . . . 29
9.5.1 MJPG-Stream receiver . . . 29
9.6 Cameras^AV . . . 29
9.6.1 Calibration . . . 30
9.7 Things to keep in mind . . . 31

9.8 Running Monocular ORB-SLAM2^{AV} 32
9.8.1 Calibration file 32
9.8.2 Launching ORB-SLAM2 33
9.9 Running Stereo ORB-SLAM2^{AV} 34
9.9.1 Hardware setup 34
9.9.2 Calibration 34
9.9.3 Software setup 34
9.9.4 Launching 35
9.10 Result with ORB-SLAM 36
9.11 Running LSD-SLAM^{AV} 37
9.11.1 Errors when building from source original source 37
9.11.2 Installing LSD-SLAM from fixed repository 40
9.11.3 Things to keep in mind 41
9.12 DeepTAM^{SM} 42
9.12.1 Errors when building from source original source 42
9.12.2 Errors when building with Script 43

10 Fazit und Persönliche Erfahrungen 44
10.1 Fazit 44
10.2 Persönliche Erfahrungen 44
10.3 Ausblick 44

Glossary 45

Abbildungsverzeichnis 48

Quelltextverzeichnis 49

Authors 51

Abstract

An autonomous driving car has to orientate in the immediate vicinity. For this purpose the car needs a map and the own position in the map. Why is this such an important task when we already have a technology like Global Positioning System (GPS)? GPS provides routing data, but no live details what happens on the street, like an ongoing construction or a pedestrian crossing the street. To get this information the car has to collect visual data from its environment in real-time. This thesis examines two approaches of visual data processing to generate a map and locate the car inside this map. One approach is Simultaneous Localization And Mapping (SLAM), the other one uses artificial intelligence. The platform for ORB-SLAM and LSD-SLAM is the Robot Operating System (ROS), which handles the publish/subscriber based communication between the camera and the SLAM algorithms.
DeepTAM uses Deep Neural Networks which can learn to predict the distance between objects in an image. The prediction is based on training data. The accuracy of the result depends on the amount of training data. It is a system for keyframe-based dense camera tracking and depth map estimation that is entirely learned. This thesis shows how to implement the different algorithms in ROS and points out the pros and cons of each method compared to the others and gives an outlook into future methods.

Ein selbstfahrendes Auto muss sich in der unmittelbaren Umgebung orientieren. Dazu benötigt das Auto eine Karte der Umgebung und die eigene Position. Warum ist dies so eine wichtige Aufgabe, wenn wir bereits über Technologien wie das Global Positioning System (GPS) verfügen? GPS liefert zwar die Standortdaten, aber keine aktuellen Details darüber, was auf der Straße passiert, wie z.B. eine Baustelle oder ein Fußgänger der die Straße überquert. Um diese Informationen zu erhalten, muss das Auto visuelle Daten aus seiner Umgebung in Echtzeit sammeln. In dieser Arbeit werden zwei Ansätze der visuellen Datenverarbeitung untersucht, um eine Karte zu erstellen und das Auto in dieser Karte zu lokalisieren. Ein Ansatz ist Simultaneous Localization And Mapping (SLAM), der Andere verwendet eine künstliche Intelligenz. Die Plattform für den ORB-SLAM und den LSD-SLAM ist das Robot Operating System (ROS), das die Publisher/Subscriber-basierte Kommunikation zwischen der Kamera und den SLAM-Algorithmen übernimmt. Der DeepTAM verwendet ein Deep Neural Network, welches trainiert wurde den Abstand zwischen Objekten in einem Bild vorherzusagen. Die Genauigkeit der Vorhersage beruht auf der Anzahl der Trainingsdaten. Der DeepTAM verwendet ein angelerntes System welches mit Keyframe-basiertem Dense Camera Tracking und Tiefenschätzung eine Karte generiert. Diese Arbeit zeigt die Funktionsweise der verschiedenen Algorithmen, beleuchtet deren Vor- und Nachteile und gibt einen Ausblick auf zukünftige Methoden.

Summary

This paper starts with the introduction into Simultaneous Localization and Mapping (SLAM) in general and the ORB-SLAM and LSD-SLAM in more detail.
Both methods need camera data. Therefore, the Basler camera is explained in chapter 5. This image data needs to be collected and transferred to the SLAM methods. For this task the Robot Operating System (ROS) is used. Chapter 2 explains ROS, which provides the platform for the different SLAM methods. ROS is a publish/subscribe based system. In chapter 6 a more detailed look into the ORB-SLAM is taken to make it easier to understand the concept of the LSD-SLAM in chapter 7. The introduction into ORB-SLAM and LSD-SLAM includes how the algorithm work, what data they require and in which way they can be used. Both algorithms provide the current position and a pointcloud which can be transferred into ROS which has a tool called Rviz 2.4.4 to visualize the pointcloud with the current position in it.
In contrast to the first two methods, DeepTAM works with Artificial Intelligence. Therefore, Artificial Neural Networks (ANN) are needed which are explained in chapter 3. This chapter explains the components and the most important types of Artificial Neural Networks for DeepTAM: Feed Forward Neural Networks and Convolution Neural Networks. The next chapter is Deep Learning. There are three different ways Deep Learning can be implemented: Supervised, Semi-Supervised and Unsupervised. This all leads to chapter 8 where DeepTAM itself is explained.
Chapter 9 deals with the project workflow where the used hardware and software is listed, setup, and how the errors where fixed during the implementation.