Map Generation in Autonomous Racing

A Comparision of a Classic Heuristical Algorithm and Machine Leaning

Alexander Seidler

Bachelor's Thesis

Department of Computer Science
Multimedia Information Processing Group
Kiel University

Advised by: Prof. Dr. Reinhard Koch
Lars Schmarje, M.Sc.

January 2022

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Die eingereichte schriftliche Fassung der Arbeit entspricht der auf dem elektronischen Speichermedium.
Weiterhin versichere ich, dass diese Arbeit noch nicht als Abschlussarbeit an anderer Stelle vorgelegen hat.

Alexander Seidler
23. 01. 2022

Abstract

- advancing technology in automation of driving and in controlled environment racing - reconstruction of abstract racing map from camera and lidar input, using slam output or using direct output - implemented in two ways a classical approch using foo bar and heuristics - and machine learning approach using mlp, cnn, etc.

Acknowledgements

Optionale Danksagungen

Table of contents

Todo list

Chapter 1

Introduction

1.1 Motivation

Autonomous racing sets a competition driven framework for the exploration of autonomous driving which plays important role in automation of modern transport. One example of such competition is Formula Student Driverless (FSD). one team that competes in this counteraction is rosyard, the Formular Student team of the Kiel university

in autonomous racing cars map generation crucial Problem
basis for autonomous driver to decide its path
look at different approaches for using data provided by simultaneous localization and mapping (SLAM) turn into usable map data for driver using a classical algorithm and a machine learning approach
this thesis systematically compares these 2 approaches for the creation of map in autonomous racing

0.5p

1.2 Related Work

einordnung in andere, was soll erforscht werden und oh wunder wird von mir gemacht

andere arbeiten nur slam und karte erkennen zusammenm
oder neural networks, auf starße aber nicht pylonen eigentlichen
aber ich beides

1p

1.3 Thesis Structure

in the following chapter basics and technical baground explained
in 3rd chapter the details of the classical and machine leaning (ML) approach, as well as their implementation presented
in the 4th chapter the approaches are evaulated and compared, and in the last chapter the results are summarized and several improvements and ideas for future work are listed

Chapter 2

Foundations and Technologies

2.1 Rosyard

Autonomous racing in track, first round second round

0.5p

2.1.1 The Rosyard Pipeline

1p

Dockerized

2.2 Machine Leaning

0.5p

2.2.1 Deep Learning and MLPs

1p

Deep learning and multi layer perceptron (MLP)s,

2.2.2 Convolutional Neural Networks

0.5p

convolutional neural network (CNN)

2.3 Discrete Curvature

Discrete curvature applies the concept of curvature from a continuous curve to a discrete curve called a polyline.

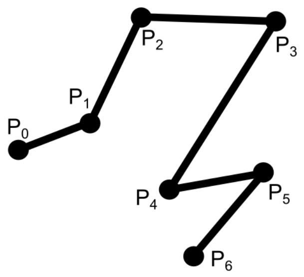

Fig. 2.1 A polyline over the vertices P_0 to P_6

A polyline is a series of line segments and is determined by a sequence of points $(P_0, ..., P_n)$ $n \in \mathbb{N}$ where each line segment connecting a pair of adjacent points $[P_i, P_{i+1}]$ $i \in \mathbb{N}_{\leq n}$ forms a vertex in the polyline.

In the continuum[source:wiki] "the curvature at a point of a differentiable curve is the curvature of its osculating circle" which more formally can be defined in terms of the unit tangent $\vec{T}$ and the arc length s: [1] [2]

$$\kappa = \left\| \frac{d\vec{T}}{ds} \right\|$$

This definition however cannot be used directly to determine the curvature of points in a polyline, given its non-continuous nature. All straight segments would have a curvature of 0 while the curvature in the edges would diverge to infinity. A new definition must be used to determine the curvature of a series line segments, which in turn can be used to approximate this series. A different definition can be derived from the quotient of the circular angle and the arc length:

$$\kappa = \frac{d\varphi}{ds}$$

Using this idea we can define the curvature from a point A, a heading $\vec{h}$ in that point and a point B as the reciprocal of the radius of the circle passing though A and B and being tangent to $\vec{h}$ in A.

Thus we can calculate the reciprocal of the radius of this circle as follows:

Since $\vec{h}$ is tangent $\gamma = 90^\circ - \alpha$ and $180^\circ = 2\gamma + \beta$ thus $\beta = 2\alpha$

The length of the secant $s := |\vec{AB}|$ can be calculated as $s = 2r \cdot sin(\frac{\beta}{2}) \Leftrightarrow \frac{1}{r} = \frac{2sin(\alpha)}{s}$

Using this method we can calculate the average curvature of the curve that is tangent in A to $\vec{h}$ and passing though B, which approximates the polyline connecting these points.

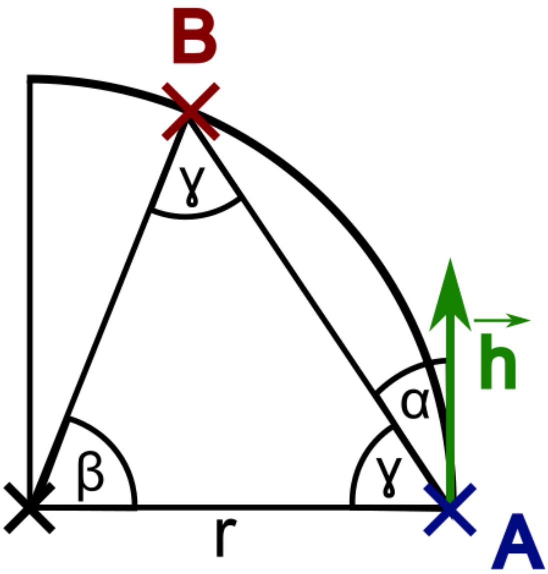

Fig. 2.2 Points A with heading $\vec{h}$ and B in circle with radius r, implying a curvature in point A of $1/r$. The circle center and B and A form an isosceles triangle with base angle γ and vertex angle β

The heading $\vec{h}$ can also be derived using the next point after A leading to B. Doing this for different distant points B on a polyline gives us a suitable approximation for the course of a polyline starting from point A. Of course this neglects the shape of the polyline completely, which fails to detect S-curves between point A and B, this however imposes no problem if we choose a fairly small distance between point A and B such that the variance of the curvature for intermediate points is non-significant.

2.4 Simultaneous Localization and Mapping

0.5p

SLAM

2.5 Development Environment Used

web technologies

1p

2.5.1 Pyodide

Pyodide [3] python script direkt im browser ausgeführt wird, und man dafür keinen extra server braucht um den python teil auszuführen, dadurch kann alles statically geserved werden, z.b. auf github pages: https://dsalex1.github.io/BachelorThesisRaceyard/

0.5p

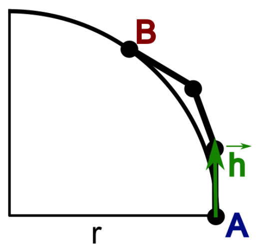

Fig. 2.3 Example curvature of $1/r$ *approximating a polyline leading from A to B, the circle corresponding to the curvature has the radius r*

Chapter 3

Methods

3.1 Classical Approach

3.1.1 Basis - Master Project by Vaishnav/Agrawal

based on master project Vaishnav/Agrawal

Schwächen bsheriger Algorithmus, Rot anstatt Gelb, Schwarz nicht klassifizierte Punkte, Grau gänzlich nicht detektierte Punkte, schwarze Linie ground truth, Der Algorithmus macht die grüne Linie daraus. wo die Pylonen perfekt erkannt wurden, grüne Linie exakt auf ground truth, da wo die Farbe nicht erkannt wurde, passieren ganz komische dinge und da wo oben rechts gar keine Roten erkannt wurden weicht es sehr vom eigentlichen Track ab. Eingabekarte: https://gyazo.com/645d505ed8a7b1ae5bd6ac8d6a867348 ERgebnis: https://gyazo.com/43237dba0fb02314743c860a9360e520

0.5p

3.1.2 First Improvement - Guessing Missing Points

1. readding missing points

erster Ansatz, Punkte auf einer Seite fehlen, also orthogonal zur Tangente eines Pylons mit dem Abstand der Trackbreite kein andersfarbiger Pylon, dann ergänzt https://gyazo.com/38577684278cf937f5

2p

3.1.3 Second Improvement - Covariance Filtering

2. ansatz Covarianzen rumgespielt, nur Punkte mit entsprechend kleiner Covarianz mit covarianz filter: https://gyazo.com/8c58c259adede0fb5b1eb977704b9655 echte daten: https://gyazo.com/60c4223b

1p

3.2 Machine Learning Approach

3.2.1 Idea and Input/Output Design

curevature to points 2-8m further down the midline using immediate environment to predict the curvature of the line segment current on

3.2.2 Modelling as Image Regressing Problem Using an CNN

mapping of input values to have a flatter distribution of values

Chapter 4

Discussion

4.1 Evaluation

4.1.1 Classical Approach

uncerternity threshhold in covariances auswirkungen analzsieren
sicherheit in der karte vs rauschen

zeitlicher verlauf nicht verfolgbar weil slam ids nicht matchen können beschrieben, weil particles getrennt kann man nicht gut mathcen, zu inperformant/wenn springen dann gar nicht

evaluation wie weit voraus notwendign sinnvoll etc 1p

Metric - Deviation From ground truth (GT)

examples for successful failed detection of the centerline 1p

Metric - Driving Test

letting driver test according to algorithm 1p

4.1.2 Machine Learning Approach

low number of training samples already great results
different parameters different learning results 2p

Metric - Driving Test

letting driver test according to algorithm 1p

4.2 Comparison of Approaches

ml more useful in first when there is no map data available, more robust for less accurate map data

less plannig ahead possible, work needed to generate map afterwards

classical resulting in complete map where planning can be done extensively, but very fragile

Chapter 5

Conclusion

5.1 Summary

Fasse nochmal alle Ergebnisse der Arbeit zusammen. 0.5p

5.2 Future Work

5.2.1 SLAM

Probleme des SLAMs wären wahrscheinlich noch verrauschte Position, Drifts und Doppelterkennungen. 0.5p

5.2.2 Classical Algorithm

- usage in first round - use proximity to car to add points Idee: neue Punkte nur in der Nähe des Autos zur Karte hinzufügen, - use orientation relative to car to recolor 0.5p

5.2.3 Machine Learning Algorithm

- factoring in derivation from track center - aus bild direk tkarte schätzem 0.5p

5.2.4 Other Improvements

cnn als preprocessing vor slam, bounding box von cones zu ist cone ja nein und farbe 0.5p

5.3 Outlook

where this work leads to

References

[1] (2016). Definition of curvature. [online] https://tutorial.math.lamar.edu/classes/calciii/curvature.aspx.

[2] Aupetit, B. (1991). *A Primer on Spectral Theory*. Springer-Verlag, New York.

[3] development team, T. P. (2021). pyodide/pyodide.

Appendix A

Abbreviations

GT ground truth 9

MLP multi layer perceptron 3

ML machine leaning 2

CNN convolutional neural network 3

SLAM simultaneous localization and mapping 1

Appendix B

TrackVisualizerJS Documentation