



IT Systems Engineering
Universität Potsdam

Bachelor's Thesis

Racing Line Detection - Recording and Comparison of Racing Lines

Ideallinienerkennung - Aufnahme und Vergleich von Ideallinien

by

Tim Oesterreich

Potsdam, June 2016

Supervisor

Prof. Dr. Christoph Meinel,
Philipp Berger, Patrick Hennig

Internet-Technologies and Systems Group

Disclaimer

I certify that the material contained in this dissertation is my own work and does not contain significant portions of unreferenced or unacknowledged material. I also warrant that the above statement applies to the implementation of the project and all associated documentation.

Hiermit versichere ich, dass diese Arbeit selbständig verfasst wurde und dass keine anderen Quellen und Hilfsmittel als die angegebenen benutzt wurden. Diese Aussage trifft auch für alle Implementierungen und Dokumentationen im Rahmen dieses Projektes zu.

Potsdam, June 27, 2016

(Tim Oesterreich)

Kurzfassung

deutsche Zusammenfassung... Lorem ipsum dolor sit amet, consectetur adipiscing elit. In erat mauris, faucibus quis pharetra sit amet, pretium ac libero. Etiam vehicula eleifend bibendum. Morbi gravida metus ut sapien condimentum sodales mollis augue sodales. Vestibulum quis quam at sem placerat aliquet. Curabitur a felis at sapien ullamcorper fermentum. Mauris molestie arcu et lectus iaculis sit amet eleifend eros posuere. Fusce nec porta orci.

Integer vitae neque odio, a sollicitudin lorem. Aenean orci mauris, tristique luctus fermentum eu, feugiat vel massa. Fusce sem sem, egestas nec vulputate vel, pretium sit amet mi. Fusce ut nisl id risus facilisis euismod. Curabitur et elementum purus. Duis tincidunt fringilla eleifend. Morbi id lorem eu ante adipiscing feugiat. Sed congue erat in enim eleifend dignissim at in nisl. Donec tortor mauris, mollis vel pretium vitae, lacinia nec sapien. Donec erat neque, ullamcorper tincidunt iaculis sit amet, pharetra bibendum ipsum. Nunc mattis risus ac ante consequat nec pulvinar neque molestie. Etiam interdum nunc at metus lacinia non varius erat dignissim. Integer elementum, felis id facilisis vulputate, ipsum tellus venenatis dui, at blandit nibh massa in dolor. Cras a ultricies sapien. Vivamus adipiscing feugiat pharetra.

Abstract

//TODO

Motor racing is all about getting around a track as fast as possible. One of the most important and most driver-influenced steps in achieving best times is the racing line, especially hitting the ideal racing line through corners.

Being able to track and compare difficult corners helps professional racing drivers to gain the necessary tenth of a second to have an advantage over the competitors and racing beginners to learn track driving more quickly and efficiently.

This paper investigates ways of automatically recording the racing line from within the car using different methods like Visual Odometry and lane detection and use the extracted data to compare and analyse the driving behaviour of different drivers.

Contents

1. Introduction	1
1.1. Motivation	1
1.2. Project Scope	1
1.3. Problem Introduction/Description/Context	1
2. Related Work	2
2.1. Racing Line calculation in Video Games	2
3. Algorithm/Concept	3
3.1. Recording	3
3.1.1. Visual Odometry	3
3.1.2. Lane Detection	4
3.2. Comparing	5
3.2.1. Visual Comparison of Racing Line with help of sensor data	5
3.2.2. Mathematical Calculation of ideal line given a certain track	5
3.2.3. Mathematical Comparison of two paths	5
4. Implementation	7
4.1. Hardware Setup	7
4.2. CUDA-accelerated Algorithms	7
5. Evaluation	7
5.1. Visual Odometry to record Racing Lines	7
5.2. Usefulness of Racing Line Comparison	7
6. Future Work	9
6.1. Open Research Questions	9
6.2. Extensions	9
7. Conclusion	10
Bibliography	11
A. Appendix	12

1. Introduction

1.1. Motivation

The trajectory around a track that allows a given vehicle to traverse the circuit in the minimum amount of time is called the ideal racing line. Needless to say, in a race this is the path every driver wants to take. In reality, however, this isn't an easy task. In order to achieve best times, a racing driver has to remember exactly which corner of the track comes next, how fast they can take this corner, see how the car is positioned on the track and process many more information.

1.2. Project Scope

1.3. Problem Introduction/Description/Context

2. Related Work

2.1. Racing Line calculation in Video Games

- vamos racing simulator

3. Algorithm/Concept

As this Bachelor Thesis tackles two problems, the algorithms can be divided into two categories as well. The first category consists of racing line recording concepts and the second one of ways to compare these recordings.

3.1. Recording

3.1.1. Visual Odometry

Visual Odometry (VO) is a concept that most camera-based robots use for navigation. It uses two consecutive camera frames to calculate the rotation and translation of the camera between them. The advantage over traditional tracking systems like GPS is, that it isn't dependent on any external sources, like satellites or radio towers, to determine the position of the object. Also it is capable of much higher polling rates than most GPS receivers, which normally poll at 1 Hz, so they get 1 positional update per second. The potential speed of visual odometry is linked to the framerate of the camera. That way, a camera that records at 30 frames per second enables Visual Odometry to estimate a position 30 times a second. The crude algorithm behind VO determines and stores recognisable features in one frame, using a feature detection algorithm. At first we need two consecutive, rectified camera images. It is important that the images are rectified, because otherwise the edges of the image would be curved and produce wrong results. Our solution used the Harris corner detection method for determining features. Finding corners in an image is a very important prerequisite for the final position determination, because it allows us to calculate direction vector to another point. This other point being the same feature in the following camera frame, where the Harris detector is used, as well. Having two sets of corners from either image we can track each feature from one frame to another.

- Lucas-Kanade?
- Jacobian?

Now that we have an end point to every (possible) start point, we can construct a set of directional vectors. Removing outliers (i.e. vectors that are much longer or go in a vastly different direction than most vectors) an approximation translation and rotation the camera conducted during the two frames can be calculated. Outlier removal is important, because otherwise the movement of other cars on the track would cause the algorithm to think the camera was moving in a different direction than it actually was. If the taken path includes loops it is possible to optimize the result by using loop detection. If any features are redetected at a later point in time and the current position is close to the position the features were redetected, the path can be stretched in a way that it connects to the earlier path. This can reduce errors that accumulated due to small estimation errors in the previous steps.

3.1.2. Lane Detection

Lane Detection uses markings on a road to determine its lateral boundaries in relation to the recording camera. It is often used in autonomous driving and driving assistance systems, because the position of the car in between lane markings gives a lot of information about the direction the car is traveling. In our case, not only can we determine a traveling direction, but also the position on the lane. By tracking the distance to the left and right lane, a deviation to the middle of the lane can be calculated. If the driven track is known or the camera is capable of determining a scale (e.g. by using a stereo camera), the deviation can even be expressed as a metric length, which is helpful for extracting additional information, like speed and acceleration, later on.

- possibility to improve recording results on known track
- track distance to left and right lane edge
- possibility to determine accurate horizontal position on track

3.2. Comparing

3.2.1. Visual Comparison of Racing Line with help of sensor data

- overlay racing lines with different lap times
- highlight relevant locations, like corners, showing detailed information about throttle and brake behaviour and curvature through the corner
- fastest way through corner is tradeoff between distance travelled and curvature, with low curvature allowing the driver to carry more speed to exit of the corner

3.2.2. Mathematical Calculation of ideal line given a certain track

- calculating smooth curves around an edged representation of the track using Bézier curves
- naively not an optimized track, but fair approximation
- simulating a chain of springloaded masses -> ideal line calculated by iterating multiple times and taking the line that uses the least force to keep masses in bounds of track (//this seems quite complex, not really sure how this works yet)

3.2.3. Mathematical Comparison of two paths

- convert pair of consecutive points into list of vectors; each vector is distance between points and angle to x-axis
 - $\text{double dx} = \text{endPoint.X} - \text{startPoint.X};$
 - $\text{double dy} = \text{endPoint.Y} - \text{startPoint.Y};$
 - $\text{double magnitude} = \text{Math.Sqrt}((\text{dx} * \text{dx}) + (\text{dy} * \text{dy}));$
 - $\text{double direction} = \text{Math.Atan2}(\text{dy}, \text{dx}) * (180 / \text{Math.PI});$
- combine vectors that have the same direction, by generating a new vec-

tor with the direction of the old ones and the sum of their lengths. This indicates a straight, which is not that interesting for comparison

- Now we can compare the angle for each section of a corner

4. Implementation

4.1. Hardware Setup

4.2. CUDA-accelerated Algorithms

5. Evaluation

5.1. Visual Odometry to record Racing Lines

- Pro:
 - Many Datapoints (in theory up to Framerate positions per second)
 - Capable of detecting fairly slight movements
 - No need for external sensors (apart from camera which already exists)
 - works in every environment (especially where there is no or poor GPS signal)
- Con:
 - uses fairly expensive operations
 - in praxis mobile-range computers will only reach up to 10 fps
 - no constant speed (slows down in areas with many feature points (like forests))
 - fairly error prone
 - rounding errors and inaccurate calculations add up and distort the racing line

5.2. Usefulness of Racing Line Comparison

- professional drivers can achieve similar lap times with very different racing lines

- however amateur and intermediate drivers can learn a lot from more skilled drivers

6. Future Work

6.1. Open Research Questions

6.2. Extensions

7. Conclusion

References

A. Appendix