Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute for
Dynamic Systems and Control
Institut für Dynamische Systeme
und Regelungstechnik

Noah Isaak

Free Space Segmentation for Gokart Application

V-Disparity

Semester Project

Institute for Dynamic Systems and Control
Swiss Federal Institute of Technology (ETH) Zurich

Supervision

First Supervisor
Prof. Dr. Second Supervisor

June 2019

IDSC-XX-YY-ZZ

Abstract

Road and ground detection are key tasks for an autonomous ground vehicle. These computations should be robust and preferably be performed in real-time. This paper aims to show the implementation of the V-Disparity method for ground detection. The approach is based on classic computer vision and does not incorporate learning methods. Basis of the method is a disparity map, for which a row-wise histogram is computed. This V-disparity histogram robustly preserves geometric scene features and can be used for various tasks. Experimental results however show shortcomings of the implementation and how they could be overcome. In the following, a second pipeline is introduced. A mapping from a Velodyne Vld-16 Lidar to a camera is computed. The binary obstacle - ground mask which is computed from the lidar's point cloud can then be projected onto the camera image. These labels can then be used for machine learning tasks.

Keywords: Ground detection, V-Disparity.

Contents

Chapter 1

Introduction

The task of segmenting a scene into ground, obstacles and other labels is a well-researched topic. It is a fundamental part of any autonomous ground vehicle, and lays the basis for many different tasks, such as planning, safety features or scene understanding. Over the last few years, solutions to this task, which are based on machine learning methods, have deemed themselves to be robust and accurate. Their ability to generalise and be applied to a variety of scenes make them a reliable choice. There however, exist many approaches to the problem, which are based on classic computer vision, with implementations dating many years back.
The goal of this semester project is the implementation of a robust and accurate free space detection pipeline based on classic computer vision. The results of this pipeline can be used to e.g. label data for machine learning. If the pipeline does not perform as expected, a second pipeline is introduced, which in a broader sense is a external calibration of the lidar and the onboard camera. The need for a lidar sensor highlights the necessity of multiple sensors to create a robust pipeline.

This semester project is part of a more extensive research into autonomous driving, based a self-driving Gokart. The Gokart is basis for a variety of research topics, such as planning, control systems and sensor fusion. The testing environment is a modular indoor Gokart track. The ground is flat, with no uphill or downhill sections. The track is not affected by weather conditions.

1.1 Related Work

1.1.1 V-Disparity

Related work on this topic is extensive. After limiting my research to the method of V-Disparity, early implementations can be dated back to 2002, with Labayrade et al. [1] providing a robust and accurate method for road detection, even for non-flat geometries. A study on the U-V Disparity method can be found in [2], providing a real-time implementation of the stereovision based scene analysis. The method was improved and adapted since it's introduction, becoming more robust and computationally efficient, as shown in [3]. Aside from free space detection, U-V-Disparity can be used solely for obstacle and pedestrian detection, with U-V-Disparity acting as the underlying basis for a SVM Classifier, where the extraced ROIs are used for training. Iloie et al. implemented this framework in [4].

1.1.2 Lidar to camera projection

Different approaches exist to calibrate a range sensor, in our case a Velodyne Vld-16 Lidar, with a camera. Geiger et al. [5] propose an automatic camera and range sensor calibration, using checkerboard patterns using 3D correspondence matching. A refined method, proposed by Dhall et al. [6], uses Aruco markers and 3D-3D point correspondences, to increase the accuracy and robustness of the method. The code implementation of this method is available as a ROS package on https://github.com/ankitdhall/lidar$_{c}amera_{c}alibration$.

1.2 Structure of this report

This report is structured as follows. In 2 both, the V-Disparity method and Lidar to Camera projection are explained in theory. Section 3 discusses the implementation of the above methods, relevant parts of the code and additionally the mounted hardware. Results are presented in section 4, followed by a conclusion and proposed future work in 5.

Chapter 2

Method

2.1 V-Disparity

This chapter explains the theoretical basis of the V-disparity method. To begin with, the basis of the V-Disparity method, the disparity map, is outlined.

2.1.1 Disparity Map

It is assumed, that the reader has a general understanding of stereovision systems. Given two calibrated stereo cameras, the respective rectified images can be used to calibrate a disparity for each pixel. These is achieved with different block-matching algorithms, such as the StereoSGBM algorithm implemented in the popular OpenCV library. Given a disparity map, the baseline and focal length, a depth map can be calculated and used for further processing. The V-Disparity method however only makes use of the disparity map.

2.1.2 V-Disparity

Once a disparity map $\Delta(u,v)$ was computed, a V-Disparity histogram can be constructed. For each row u, a histogram of the occuring disparities in this row is computed. The histogram values represent the occurence of the respective disparity in the row, where each bin is represented by a pixel. Given a plane in a scene, the projection of the plane onto the V-Disparity image has a useful property. A plane will be projected as a linear curve in the V-Disparity image. This simplifies the extraction of the respective plane in the V-Disparity image, as a e.g. Hough Line Transform can be applied to detect the lines. Consequently, the detection of straight lines in the V-Disparity corresponds to detection of planes in the scene. A scene is therefore made up of planes, where vertical planes can be understood as obstacles, horizontal planes as the road when flat, or as a set of oblique planes when the road is non-flat. Hu et al. [2] offer an in-depth analysis of the projection of the three above types of planes.

Because the most prominent plane in usually represented by the ground, it will be detected as the line with the most votes in the V-Disparity image. Horizontal or vertical lines can be dismissed, as the disparity gradient of the road leads to a skewed line. Horizontal lines can be associated with obstacles and used for obstacle detection. Even though Non-flat road geometry will not be considered in this project, it should be noted that the road in that case can be approximated by a series of planes, which will then be projected as a piecewise linear curve in the V-Disparity image.

Figure 2.1

Once the line has been fitted in the V-Disparity image, the disparity values for the road surface are known. Extracting the road in the image domain is straightforward. For each row, the values which lie within a threshold of the value of the extracted line are part of the road, all other pixels are masked as non-road.

Oniga et al. [7] show a camera image, the corresponding dispartiy map and the V- and U-Disparity respectively.

2.2 Lidar Camera Projection

The lidar scans its surrounding and generates a point cloud of the structures and objects around it. To begin with, a simple bidary classification was put in place. The classifier distinguishes obstacles and ground, via a simple height threshold. Therefore, all points above a certain height threshold are classified as obstacles. This is a rough approximation and by no means sufficient for higher level classification and detection task, but for a first implementation of the proposed method it adequate. If one only wants to project the points belonging to the ground onto the camera frame, the point cloud can be compressed to a 2D occupancy map, which will be used later. The full point cloud can be used as well, if needed. At first, the point cloud and occupancy map will be sparse, after a few laps however, it should be dense and contain all relevant points and it doesn't need to be updated anymore. After the point cloud/occupancy map acquisition, the points can be projected onto the camera frame.

Before a point in camera coordinates can be projected onto the camera frame, the respective point in world coordinates needs to be converted to camera coordinates.
Let X_w be a point in world coordinate frame.
Let X_c be a a point in camera coordinate frame.
The two points relate to each other by a translation and rotation. To convert X_w to X_c, a translation and rotation, in that order, need to be applied.

$$X_c = R(X_w - C) \tag{2.1}$$

Where all points are in non-homogeneous coordinates, R and C with respect to the world coordinate system.

In homogenous coordinates, equation 2.1 can be written as follows.

$$\widetilde{X}_c = \left[\begin{smallmatrix} R & -RC \\ 0 & 1 \end{smallmatrix}\right] \widetilde{X}_w \tag{2.2}$$

To project lidar scanning points onto a camera image, the camera's intrinsic parameters should be known, namely the focal length f_x and f_y in x and y direction respectively and the camera's optical center c_x and c_y. All can be summarised in, what is called, the intrinsic matrix K of the camera.

$$K = \left[\begin{smallmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{smallmatrix}\right] \tag{2.3}$$

For most cameras, the distortion parameters need to be considered, to correct the pixel projection. OpenCV considers radial and tangential factors. More on that can be found on the OpenCV documentation site.

The final projection of the point in camera coordinates onto the camera frame is done via simple matrix multiplication.

$$x = K[I|0]\widetilde{X}_c = KR[I| - C]\widetilde{X}_w \tag{2.4}$$

Where K denotes the intrinsic matrix from 2.3 and I the identity matrix.

To determine the rotation and translation matrix, the kart's position and orientation with respect to the world coordinate frame needs to be known. Also, the relativ position and orientation of the camera to the kart's center needs to be measured or determined by a different method. Measuring can lead to large projection errors, as the method is inherently sensitive to even small errors. A different method to find the respective rotation and translation matrix is proposed by Dhall et al. [6].

All points which belong to the ground and are projected onto the camera image can be interpreted as a mask, covering all points in the camera image which belong to the ground.

Chapter 3

Implementation

3.1 V-Disparity Implementation

3.2 Hardware

The Gokart is fitted with a ZED Stereolabs camera, see Figure 3.2, which was the basis for the V-Disparity method. The camera comes with a ZED SDK, which provides many functionalities, such as disparity map generation and point cloud computation. The Python API enables quick and intuitive access to these functionalities. In addition, a Velodyne VLD-16 Lidar is mounted on top of the Gokart, which can be seen in Figure 3.1. It offers 16 scanning lines with high accuracy. More infos on the used hardware can be found on the manufacturer's website.

3.3 Code implementation

3.3.1 V-Disparity

The method was implemented in the Python framework, while making use of the popular OpenCV library. OpenCV was used, mainly because of the existing methods already implemented, such as the probabilistic houghline transform, which was used for line-fitting. The OpenCV aims at real-time computer vision programming functions, which is a crucial property for autonomous vehicles.

Given the ZED Camera and the ZED SDK, data can be streamed from the Camera with few lines of code. Depending on the application, the resolution, framerate and other parameters can be adjusted accordingly. For disparity computation, range and quality can be modified, to fit the requirements. The disparity map is computed as float32. For visualisation purposes, a normalisation and conversion to uint8 type needs to be done. For higher precision, the float32 disparity map can be used for further processing. To ease up on computation, the uint8 disparity map was used for the V-Disparity method. The number of histogram bins for every row histogram is then set to 255, one for every disparity value present in the normalised disparity image. OpenCV's calcHist function was used. To extract the regions of interest, in our case the ground, OpenCV's Houghline transform can be used, to fit the line in the image. In order to fine tune the line fitting, probabilistic houghline transform was used, where minimum line length and maximum line gap can be adjusted. This line can then be backprojected into the original camera image, to generate the mask.

Figure 3.1

Figure 3.2

For the current implementation, the disparity map and camera images are streamed at a resolution of (672, 376). The same resolution is used for the V-Disparity Histogram and Line Fitting. For mask generation, the resolution was downscaled by a factor of two, to ease up on computation and enable real-time application. Using higher resolution could improve results, but drastically lowers the framerate

In order to avoid speckles and disjointed mask segments, all mask pixels are labeled according to their connectedness, using OpenCV's connectedComponents function. Only the largest connected region is kept, all other labels are discarded.

3.3.2 Lidar to camera projection

Euler angles where used to determine the rotation matrices. After constructing the rotation vector and determining the relative angles, the vector was transformed to a 3x3 rotation matrix. Because OpenCV only uses Rodrigues Vector notation for rotation, the rotation matrix had to be converted to Rodrigues as well. This was easily done with an already implemented OpenCV function.

Chapter 4

Results

Results have been visually evaluated, as there is no ground truth available for this method. The evaluation have shown, that for this project's implementation, the V-Disparity method does not provide the desired results. Parameters can be tuned for single frames to offer a useful ground mask, however the method does not generalize well for a series of frames, or video stream. The generated mask lacks temporal coherency, and for mid-range distances (>4m), important features are not detected, such as various obstacles or road boundaries.

The reason for this can be found in the generated disparity map. Without post processing, the backprojection of the road requires an accurate disparity map. However the quality of the disparity map diminishes for further distances, edges are not well preserved and artifacts in low-textured areas lead to inaccurate masks. As the disparity values for low-textured areas are hard to determine via block matching and the underlying disparity computation code is closed source, the exact reason for the error is hard to determine.
Adjustments of the parameters can lead to better results. An extensive study on the quality and error of the ZED Camera was carried out by Ortiz et al. [8].

Chapter 5

Conclusion

The evaluation of the proposed V-Disparity method for free-space detection shows the need for an auxiliary sensor, in our case a range finding lidar sensor. The stereocamera on it's own does not provide the desired quality of the road mask. To complete the task of creating a useful road mask, a second method was implemented, which projects scanned lidar points into the camera frame. This was done by first externally calibrating the two sensors, after meticulously measuring the relative pose and orientation. This approach was then compared to a automatic calibration method, which is available online.

5.0.1 Future Work

For future work, the disparity map generation should be improved. The quality of disparity maps generated by recent neural networks seem promising, such as PSMNet by Jlia et al. [9]. Resolving the bottleneck of the disparity map can shed light on the performance of the actual V-Disparity method. Keeping the initial ground mask undersegmented, and refining it by using it as seeds for region growing algorithms, may improve results. Augmenting the code to be more computationally efficient should be considered as well. Using for loops in Python increases computing time, replacing loops with other functionalities can help.

Appendix A

AppendixChapter

Bibliography

[1] R. Labayrade, D. Aubert, and J. . Tarel, "Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation," in *Intelligent Vehicle Symposium, 2002. IEEE*, vol. 2, June 2002, pp. 646–651 vol.2.

[2] Z. Hu and K. Uchimura, "U-V-disparity: An efficient algorithm for stereovision based scene analysis," *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2005, pp. 48–54, 2005.

[3] S. Kakegawa, H. Matono, H. Kido, and T. Shima, "Road Surface Segmentation based on Vertically Local Disparity Histogram for Stereo Camera," *International Journal of Intelligent Transportation Systems Research*, vol. 16, no. 2, pp. 90–97, 2018.

[4] A. Iloie, I. Giosan, and S. Nedevschi, "UV disparity based obstacle detection and pedestrian classification in urban traffic scenarios," *Proceedings - 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing, ICCP 2014*, pp. 119–125, 2014.

[5] A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3936–3943, 2012.

[6] A. Dhall, K. Chelani, V. Radhakrishnan, and K. M. Krishna, "Lidar-camera calibration using 3d-3d point correspondences," *CoRR*, vol. abs/1705.09785, 2017. [Online]. Available: http://arxiv.org/abs/1705.09785

[7] F. Oniga, E. Sarkozi, and S. Nedevschi, "Fast obstacle detection using U-disparity maps with stereo vision," *Proceedings - 2015 IEEE 11th International Conference on Intelligent Computer Communication and Processing, ICCP 2015*, pp. 203–207, 2015.

[8] L. E. Ortiz, E. V. Cabrera, and L. M. Gonc, "Depth Data Error Modeling of the ZED 3D Vision Sensor from Stereolabs," vol. 17, no. 1, pp. 1–15, 2018.

[9] J. Chang and Y. Chen, "Pyramid stereo matching network," *CoRR*, vol. abs/1803.08669, 2018. [Online]. Available: http://arxiv.org/abs/1803.08669

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institute for Dynamic Systems and Control

Prof. Dr. R. D'Andrea, Prof. Dr. E. Frazzoli, Prof. Dr. C. Onder, Prof. Dr. M. Zeilinger

Title of work:

Free Space Segmentation for Gokart Application

V-Disparity

Thesis type and date:

Semester Project, June 2019

Supervision:

First Supervisor
Prof. Dr. Second Supervisor

Student:

Name:	Noah Isaak
E-mail:	nisaak@student.ethz.ch
Legi-Nr.:	13-929-476
Semester:	FS 2019

Statement regarding plagiarism:

By signing this statement, I affirm that I have read and signed the Declaration of Originality, independently produced this paper, and adhered to the general practice of source citation in this subject-area.

Declaration of Originality:

`https://www.ethz.ch/content/dam/ethz/main/education/rechtliches-abschluesse/leistungskontrollen/declaration-originality.pdf`

Zurich, 25.6.2019: ______________________________