ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Autonomous Systems Lab
Prof. Roland Siegwart

Semester Thesis

Online Estimation of Camera Extrinsics using Map Information

Autumn Term 2023

Supervised by:
Cornelius Von Einem
David Hug

Author:
Nicolina Spiegelhalter

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Online estimation of camera extrinsics using map information

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Spiegelhalter	Nicolina Helen

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
Zürich, 23.01.2023	N. Spiegelhalter

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.

Contents

Preface

I would first like to extend my sincerest appreciation to Prof. Dr. Roland Siegwart for giving me the opportunity to write this thesis within the Autonomous System Lab at ETH Zürich. Furthermore, I would also like to express my deepest gratitude to Cornelius von Einem and David Hug for their unwavering support throughout the duration of this project. Their guidance and expertise have been instrumental in allowing me to acquire a wealth of professional knowledge, and I am truly grateful for their contributions.

Abstract

The primary task of vision-based driver support systems is to assist the driver in operating a vehicle safely, which includes raising an alarm in case of a potential collision. In this work, we present an online method for extrinsically calibrating a monocular camera installed on a track vehicle for named support systems. The method utilizes a preinstalled GPS to self-localize the vehicle and a front-mounted camera to search for anomalies within the area between the railway tracks. In order to determine this area of interest, the railway tracks are projected into the camera frame, which are known from a given map. For a precise projection, we require the relative pose between the GPS and the camera. Our proposed method utilizes the horizontal vanishing point, the visual detection of wooden sleepers between the tracks and electricity poles to perform the camera calibration. This process can be applied to both moving track vehicles and existing data sets. The estimated pose shows a significant improvement compared to the absence of calibration, and with an added safety margin, we believe the method to be applicable in real-world scenarios.

Symbols

Symbols

φ, θ, ψ	roll, pitch and yaw angle
λ	depth of image point

Indices

f_u, f_v	Focal length in pixels for u-axis and v-axis
h	Height
$H_{i/j}$	Homogeneous transformation matrix from j-frame to i-frame
h_{offset}	Height offset along y-axis of camera from camera to GPS
hl_{offset}	Horizontal offset along x-axis of camera from camera to GPS
K	Camera intrinsic matrix
l_{offset}	Longitudinal offset along z-axis of camera from camera to GPS
M	Perspective projection matrix
R	Rotation matrix
t	Translation vector
u_0, v_0	Coordinates of principal point of camera
${}_ix$	Value along x-axis expressed in i-frame
${}_iy$	Value along y-axis expressed in i-frame
${}_iz$	Value along z-axis expressed in i-frame

Acronyms and Abbreviations

ETH	Eidgenössische Technische Hochschule
FoV	Field of view of camera
GPS	Global Positioning System
IMU	Inertial Measurement Unit
ROI	Region of interest

Chapter 1

Introduction

With the increasing amount of traffic in cities, the demand for safe and frequent public transport has grown significantly [1]. This fuels the interest in having a vision-based driver support system for tramways and trains, as they improve safety with collision assessment protocols and can, therefore, potentially even lead to full autonomy. This development could additionally deem itself more profitable. With the demanded higher frequency of trains, no additional drivers would be required, and the cameras used are low cost and can easily be integrated into existing infrastructures [2].

To ensure the successful implementation of self-driving track vehicles, collision prevention is a key priority. This requires looking as far ahead as possible along the tracks, as the long braking distances necessitate early detection of obstacles. The region of interest (ROI) is consequently the area in between the railway tracks ahead of the track vehicle. Finding the ROI can be achieved by either detecting the railway tracks in every frame from the bottom of the image frame to as far ahead as possible or by projecting the known map of railway tracks into the image. The first method must be able to handle, e.g., illumination changes and occlusions, whereas the second method doesn't have these limitations.

In this thesis, we focus on the second method, where knowing the camera's pose at the front of the track vehicle is crucial for an accurate projection of the railway tracks. However, determining the extrinsics is complicated because cameras are often detached and reattached differently due to the lack of mounting fixtures, and an on-site calibration process is deemed cumbersome. This means a fast online computation of the camera's relative position and relative orientation with respect to the global positioning system (GPS) after the camera has been newly attached to a track vehicle is needed. It is given that the GPS is already preinstalled on every vehicle for self-localization.

This paper assumes that a single camera is attached to the train, which has already been intrinsically calibrated. Assuming that the monocular camera and GPS are installed on one rigid body, the extrinsics, once computed, can continuously be used until the camera is dismounted from the train. There are three main approaches to finding the extrinsics of a monocular camera. The first approach relies on instruments such as calibration boards or markers placed in the surroundings of the camera [3]. This was dismissed as we wanted to use the visual information collected during the drive.

The second approach trains a convolutional neural network to determine the ex-

trinsics of the camera [4]. This is not feasible in our case due to the lack of labeled data for track vehicles.

The third approach leverages the geometry of the image, a process called photogrammetry. Here some authors made use of the three vanishing points present in an image to calibrate the camera [5] [6] [7] [8]. This only works well if all vanishing points lie in the finite space from the image. This isn't the case in our visual information, but inspiration was taken out of this approach.

We propose a method that estimates the six parameters of the pose in a multi-staged process, where some leverage the location of one of the horizontal vanishing points visible in the image [9] and the others utilize the visual detection of distinctive landmarks. The detection of the used horizontal vanishing point is rather complicated in, e.g., construction sites but not so much in track scenes, as it lies at the intersection of the linearly extended railway tracks. The computed estimates are averaged over several frames to account for the potential slight lateral movement of the train due to uneven railway tracks. Validated on the alignment of the projected railway tracks to the tracks in the image, applying the proposed method led to a large improvement. This method has the advantage of also being applicable to existing data sets.

The thesis is structured by first giving background information on the theory used. It is followed by a detailed approach on how the parameters are estimated in chapter 3. The results of the proposed method are described in chapter 4 and discussed in the last chapter, where also an outlook is given.

Chapter 2

Background

2.1 Projective Geometry

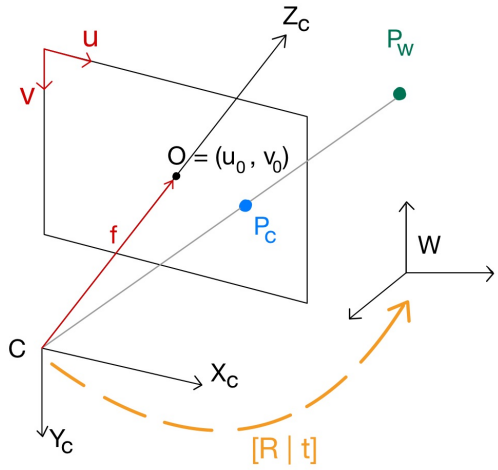


Figure 2.1: Visualization of projective geometry

The projection of a world point $P_w \in \mathbb{R}^{3\times1}$ into the image frame requires the multiplication with the perspective projection matrix $M \in \mathbb{R}^{3\times4}$ [10]. M is defined as the multiplication of the camera intrinsic matrix $K \in \mathbb{R}^{3\times3}$ with the rotation matrix $R \in \mathbb{R}^{3\times3}$ and horizontally concatenated translation vector $t \in \mathbb{R}^{3\times1}$ of the camera in the world frame. To use these matrices, P_w has to be converted into homogeneous coordinates by vertically appending a "1".

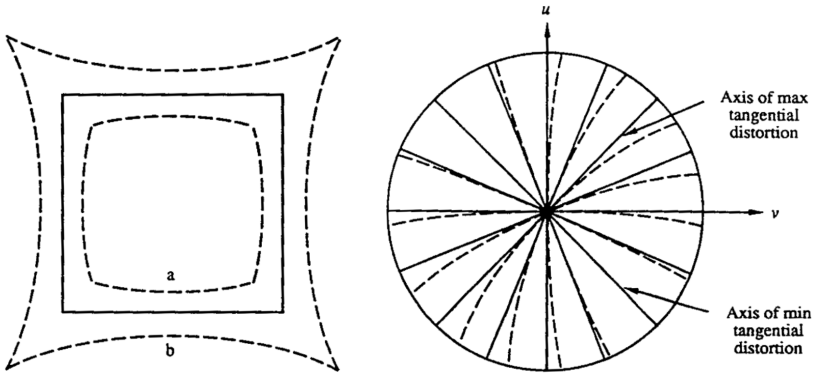
$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = M \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = K \, [R \mid t] \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = K \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} \tag{2.1}$$

The matrix K containing the intrinsic parameters of the camera is shown in equation 2.2, where α is the focal length in pixels and (u_0, v_0) are the coordinates of the optical center, also known as the principal point, in pixels. If the pixels are not perfectly square, α_u and α_v will not have the same values. The focal length in pixels α_u or α_v can be converted into meters with the respective conversion factor k_u or k_v.

$$K = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \alpha_i = k_i \cdot f, \quad i \in \{u, v\} \tag{2.2}$$

In the rest of the report, for clarity purposes, the focal length in pixels will be referred to as f with the subscript of the according direction $\{u, v\}$.

If the camera's lens is distorted, this needs to be corrected, prior to the conversion of a world point to an image point, to be able to compute precise dimensional measurements. The two types of distortion are radial and tangential [11]. Different transformation equations must be applied to find the new coordinates of the undistorted image points for these two types of distortions.



(a) (b)

Figure 2.2: a) Solid lines: no distortion, dashed lines: radial distortion, where a: negative and b: positive, (b) Solid lines: no distortion, dashed lines: tangential distortion[11]

2.2 Vanishing Points

The world frame can be described by three mutually orthogonal axes. All lines running in the direction of one of these axes are parallel. When observing an image of the world, this property is no longer maintained. The lines are still perceived as straight but no longer parallel.

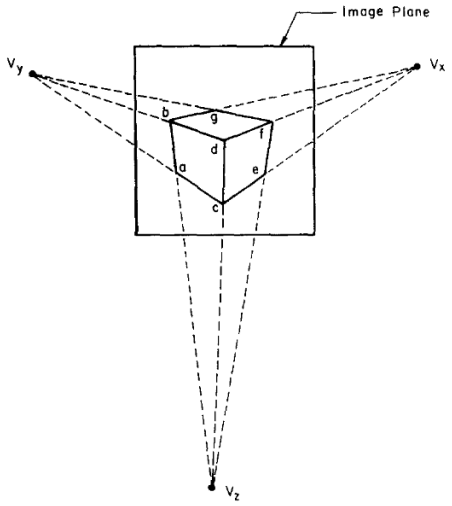


Figure 2.3: Vanishing points in a perspective image [12]

As can be seen in the perspective Figure 2.3, the lines parallel in a plane converge to one point, called vanishing point [5]. This property is reversible, meaning if two lines converge to a vanishing point, they are parallel in the world frame. When assuming perspective projection, every image contains three vanishing points $\{V_x, V_y, V_z\}$, which can lie outside of the image plane or even converge in infinity depending on the position of the camera with respect to the world frame. Any straight line that is parallel to the image plane has a vanishing point in infinity [13]. The two vanishing points $\{V_x, V_y\}$ are referred to as horizontal vanishing points in Figure 2.3, and V_z as the vertical vanishing point. A vanishing line is defined as a line that contains two vanishing points. Therefore in a perspective image, three vanishing lines exist. In a projective image, the horizon line is where all planes parallel to the ground plane meet.

Chapter 3

General Approach

When a camera is randomly attached at the front of a track driving vehicle, its pose relative to the preinstalled GPS shown in blue in Figure 3.1 is unknown. To determine these six degrees of freedom, three for the position and three for the orientation, a combination of visual-, map-, and GPS information is leveraged. With this knowledge, we close the circle, from the world frame to the camera frame, via the GPS and back.

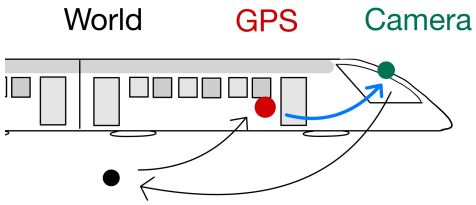


Figure 3.1: Relative poses between the world, GPS, and camera

These six parameters are estimated in a multi-stage process, where they are computed over several frames and then averaged. The parameters are labeled as visualized in Figure 3.2.

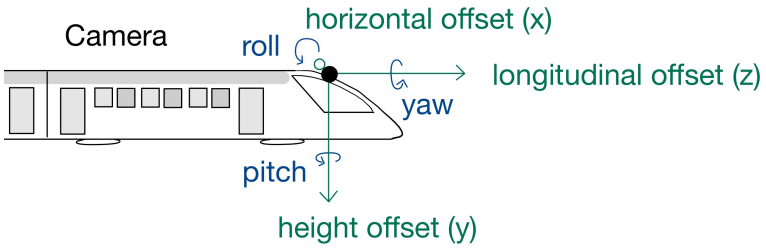


Figure 3.2: Labels of estimated parameters

With prior knowledge, it is obvious that the monocular camera is attached at the front of the train and, therefore, must have a longitudinal offset to the GPS. As the GPS location is unknown, the magnitude of this parameter is obscure. This also holds for the other two translational camera offsets to the GPS. Therefore, the camera frame is initialized at the origin of the GPS frame. It is initialized with a

rotation of $-90°$ around the z-axis followed by a $-90°$ intrinsic rotation around the x-axis. This way, the z-axis points forward, which is the standard way of defining a camera frame. Any estimate is made relative to the initial camera frame, where position estimates are added to the corresponding axis in the camera frame, and the rotations are extrinsically added to the camera frame.

The order in which the parameters are computed is of relevance, as some are dependent on others. This dependency is visualized in Figure 3.3.

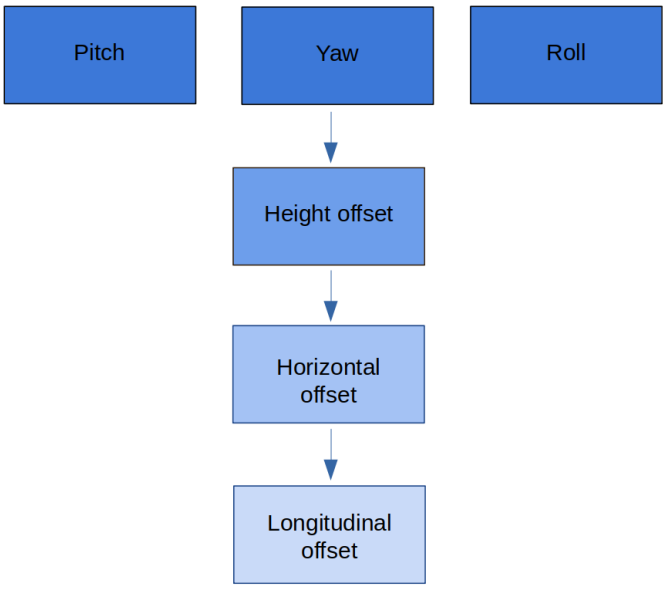


Figure 3.3: Visualization of parameter dependencies and their consequential order

The process is subdivided into two steps; where first, the railway tracks are visually detected, and this information is then used to compute the parameters.

3.1 Rail Detection

The basis of the pose estimation is a successful railway track detection. For our purposes, only a small part at the bottom of the image frame had to be detected. In this part of the image, the railway tracks can be approximated as straight lines [14], simplifying their detection.

Herefore, after having turned the current frame into a gray image, it was blurred with a 3×3 Gaussian kernel. For the determination of the most fitting edge detection algorithm, a comparison was done between the most commonly used algorithms, the Canny, LOG, Prewitt, Roberts, and Sobel edge detectors depicted in Figure 3.4. We obtained the best results with the Canny edge detector, which has also been implemented by other authors [15].

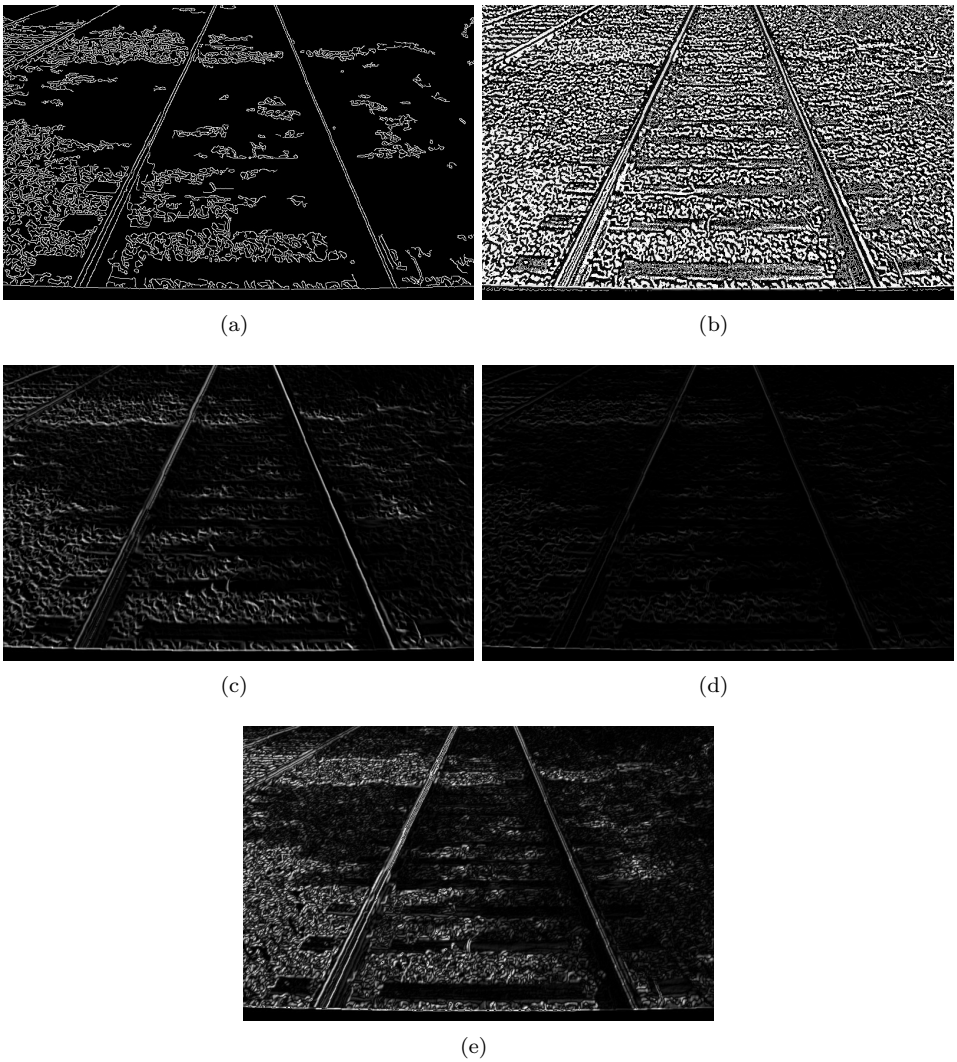


Figure 3.4: (a) Canny (b) LOG (c) Prewitt (d) Roberts (e) Sobel

After applying the Canny algorithm on the image, a probabilistic Hough Transform [16], which allows searching for a predefined minimum length and maximal pixel gap, was performed. The resulting line candidates were further filtered with pre-

knowledge of the train tracks geometry as proposed by Wang [3]. This filter limited the admissible gradient for the detected rail and only allowed rails that crossed the bottom frame of the image. This last constraint had the benefit of filtering out railway tracks that ran parallel to the currently traveled-on tracks of the vehicle. As a final decision, when multiple line candidates for the rails were left, the candidates closest to the horizontal middle of the frame were chosen. This last decision was based on the fact that the standardized width of tracks is measured on the inside of the rails, which is leveraged later. If no appropriate candidates were determined the frame was skipped.

3.2 Parameters Estimation

In the following section, the different approaches used to compute the individual parameters are described, where first, the translational ones are addressed, followed by the rotational ones.

3.2.1 Height Offset Estimate

The difference in height h_{offset} between the GPS and camera was measured along the y-axis of the camera. Determining this height offset depended on having previously estimated φ_{roll} of the camera.

With an approach inspired by Ross [17], the radius r was calculated in Equation 3.1 using the focal length in pixels f_v along the vertical direction, the ratio w_0 between the perceived width of train tracks at the bottom of the image frame and the real-world standardized width of the train tracks W. This ratio was inserted into Equation 3.2 with the vertical field of view (FoV) of the camera, and the precomputed camera roll φ_{roll} to obtain the height of the camera.

$$r = \frac{f_v \, W}{w_0} \tag{3.1}$$

$$h_c = r \;\; cos(\frac{\pi}{2} - \varphi_{roll} - FoV_v) \tag{3.2}$$

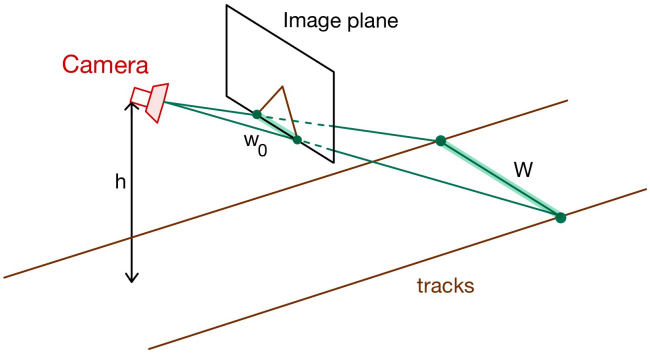


Figure 3.5: Geometry of perceived width of tracks and their real-world width

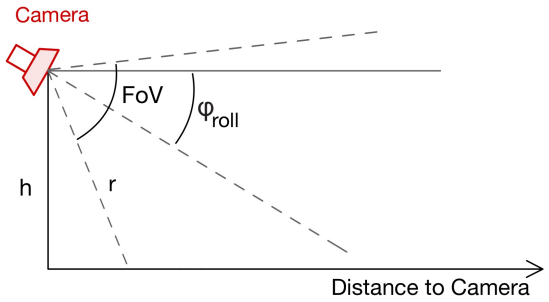


Figure 3.6: Geometry used for the estimation of h_c [17]

Leveraging the elevation at the current train position [18], the h_c was converted into the vertical height difference between the camera and GPS h_{offset} in Equation 3.3.

$$h_{offset} = h_c + h_{elevation} - h_{GPS} \tag{3.3}$$

3.2.2 Horizontal and Longitudinal Offset Estimate

The estimation of the horizontal offset along the x-axis of the camera and the longitudinal offset along the z-axis of the camera was computed similarly. The difference in the visual perception to the known location of a distinctive landmark was leveraged. This process depends on all rotational estimates and the height offset of the camera.

As a distinctive object, many options existed, such as a turnout, curve, or pole, where the visual detection of a turnout has been researched [19] [17] [20] and the detection of curved tracks far ahead has also been done [14]. The choice of the landmark was set on an electricity pole as it could be reduced to one fixed location at its contact point to the ground. Once the pole had been visually detected, the depth of its ground contact point in pixels had to be found, as it had to be converted into the GPS frame. Following this was a comparison between the visually detected pole in the GPS frame and its known location in the GPS frame.

Visual Pole Detection

To visually detect the electricity poles, HSV color ranges trimmed to the pole colors in different illumination cases were applied to the image. This was done, as a simple Hough Transform searching for vertical lines didn't suffice in finding lines that would run along the complete pole front. The top part of the poles was easily distinguished from the sky, but not so for the bottom part, depending on the pole's background.

The images were converted from an RGB color space into an HSV color space, separating the pixel color information into three layers: hue, saturation, and value. All pixels that fell within predefined HSV ranges were kept white, while all others were turned black. This process was followed by a component analysis that searched for groups of pixels with a certain area. This area was then compared to a predefined threshold to determine whether this group was desired.

Figure 3.7: Image after application of HSV range filters and component analysis

The image with the remaining component had to be searched for the component representing the pole. This was done by searching the image for vertical lines with a Hough Transform that fulfilled constraints on its position. These constraints were set with the prior knowledge of the pole location next to the railway tracks. This filtered out any other objects that could have been detected, e.g., trees.

This process only worked when the train was driving on a straight track, as only then were the poles observed as vertical and could be distinguished from the train tracks, which could have been included in the color-filtering process by chance. The component this vertical line belonged to was found by computing its distance to the centroid of the component and choosing the component with the smallest distance. The ground contact point of the pole was then found by first finding the top pixel with a simple search through the component's pixels. The bottom pixel was chosen as the lowest one with the same u-coordinate as the top pixel.

Depth Computation

In a setup with stereo cameras, performing the depth estimation is done by focusing on the same scene from different perspectives [21]. In others, with only a monocular camera, a laser could be used to compute the depth [22], [23]. Some even apply a deep convolutional neural network in a monocular setup [24]. In our case, if the pole detection was successful, the lowest point should be on the ground, which can be leveraged to compute the depth λ to convert the pixel to the camera frame [9].

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = K \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} \tag{3.4}$$

If the matrix notation of Equation 3.4 is written out in single equations and solved for the camera coordinates, we obtain the following three equations:

$$X_c = \frac{Z_c(u - u_0)}{\alpha_u} \quad Y_c = \frac{Z_c(v - v_0)}{\alpha_v} \quad \lambda = Z_c \tag{3.5}$$

With the precomputed pose estimates of the camera, the homogeneous transformation matrix $H_{w/cam}$ from the camera to the world frame can be used to convert a point P_c in homogeneous coordinates on the image frame to a point P_w in homogeneous coordinates in the world frame:

$$\begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = H_{w/cam} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 & h_4 \\ h_5 & h_6 & h_7 & h_8 \\ h_9 & h_{10} & h_{11} & h_{12} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \tag{3.6}$$

If Equation 3.5 is plugged into Equation 3.6, the third row is taken, solved for Z_c and our knowledge of $Z_w = h_{elevation}$ is used, we obtain Equation 3.7.

$$Z_c = (h_{elevation} - h_{12})/(h_9 \frac{(u - u_0)}{\alpha_u} + h_{10} \frac{(v - v_0)}{\alpha_v} + h_{11}) \tag{3.7}$$

As we know $\lambda = Z_c$, we can now use this depth to find the bottom point of the pole expressed in the GPS frame.

Height Criterion

As a condition of whether the pixels indicating the highest and lowest point of the pole are detected correctly and the frame can be used, the height of the pole in the world can be computed and compared to our prior knowledge of its height.

With our computed depth λ, we can find the distance of the pole from the base point of the camera D' with our precomputed height estimate of the camera h_c by simply applying Pythagoras theorem. When dividing this diagonal distance D' by the cosine of α, we obtain the horizontal distance D.

$$D' = \sqrt{\lambda^2 - h_c^2} \qquad D = \frac{D'}{cos(\alpha)} \tag{3.8}$$

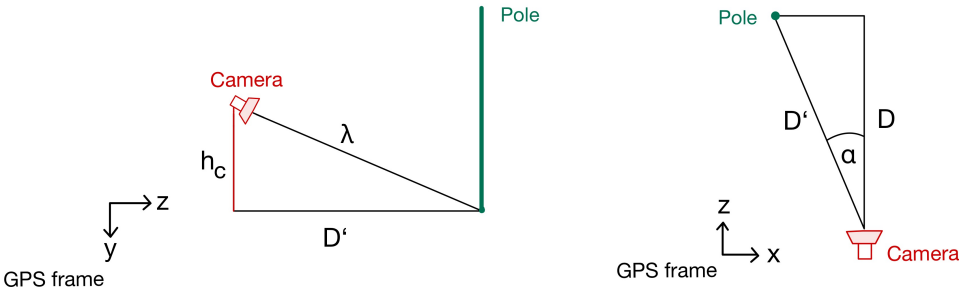


Figure 3.8: a) Top view of the camera to pole geometry, (b) Side view of the camera to pole geometry

To determine the angle α visible in Figure 3.8 b), the distance T seen in red in Figure 3.9 can be determined and then converted to an angle by determining the percentage this distance takes up of the vertical FoV_u [9].

$$T = u_0 - u_{pole} \qquad \gamma = \frac{FoV_u}{2\ u_0} \qquad \alpha = \gamma \cdot T \tag{3.9}$$

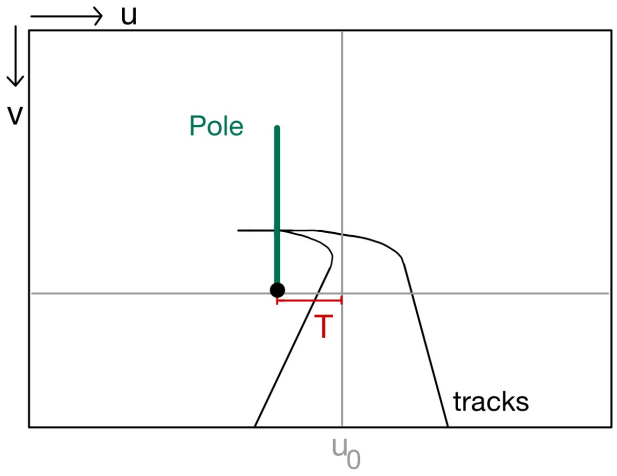


Figure 3.9: Exemplary image of parameters used to compute α

Using the horizontal distance D and the height of the pole in pixels h_p, by subtracting the u-components of the highest from the lowest pixel of the pole, the height of the pole in the world frame H can be computed. An aspect ratio can be set up, as the pole is an object that touches the ground [25]. The approach used is the not-approximated approach proposed by Kar[26]. The definition of the top pixel P_t and bottom pixel P_b given in Equation 3.10 can be subtracted to get a definition

for the height of the pole in pixels h_p and solve it for the height of pole in meters H in Equation 3.12.

$$P_b = f\frac{-h_c/D + tan(\varphi_{roll})}{1 + (h_c/D)\ tan(\varphi_{roll})} \qquad P_t = f\frac{-(h_c + H)/D + tan(\varphi_{roll})}{1 + (h_c/D)\ tan(\varphi_{roll})} \tag{3.10}$$

$$h_p = P_t - P_b \tag{3.11}$$

$$H = \frac{h_p(D + h_c\ tan(\varphi_{roll}))}{f_v} \tag{3.12}$$

This height H can be compared to the prior knowledge of the pole's height. If their difference is over a predefined threshold, this estimate can be discarded. The threshold can be set depending on how precise the previous estimates of the relative camera pose are.

Horizontal Offset Estimate

After a successful pole detection and depth computation, the horizontal offset hl_{offset} can be computed. For this, after the visually detected ground contact point of the pole is expressed in the GPS frame and the same for the known location of the pole, their y-coordinates can be subtracted from each other:

$$hl_{offset} = {}_{GPS}\ y_{cam} - {}_{GPS}\ y_{world} \tag{3.13}$$

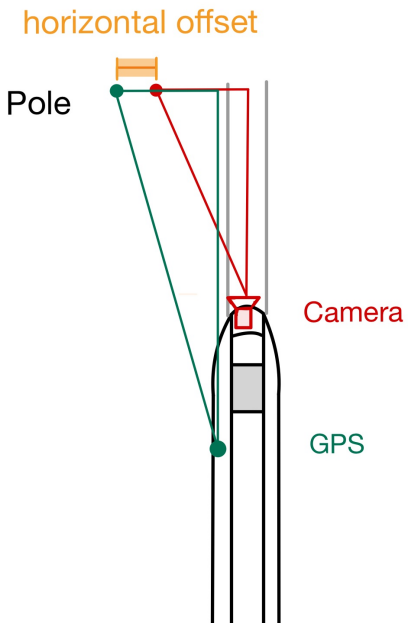


Figure 3.10: Geometry used to detect hl_{offset} between camera and GPS

Longitudinal Offset Estimate

The same holds for the longitudinal offset l_{offset}. As the camera is located at the front of the vehicle, meaning in front of the GPS, the x-coordinate of the object seen from the camera would be smaller than its actual x-coordinate in the GPS frame. This difference is used to compute the longitudinal offset between the camera and GPS.

$$l_{offset} =_{GPS} x_{world} -_{GPS} x_{cam} \tag{3.14}$$

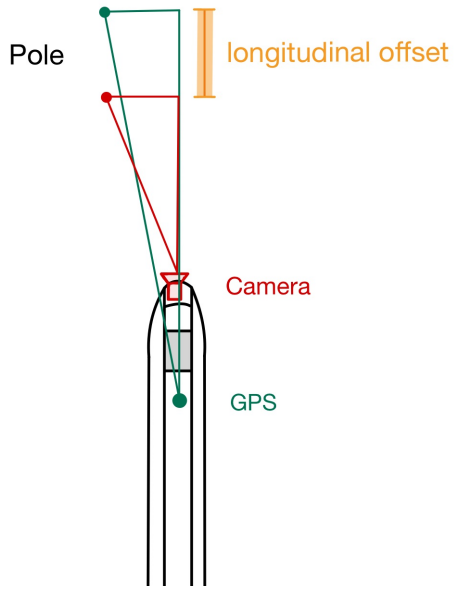


Figure 3.11: Geometry used to detect l_{offset} between camera and GPS

3.2.3 Roll and Pitch Offset Estimate

These two estimates are computed by leveraging the properties of the vanishing point in the image. As a rail has a width and the railway detection can have some slight variations, to make the location of the vanishing point more robust than simply taking the point where the two detected railway tracks intersect, the following process is proposed.

Vanishing Point Detection

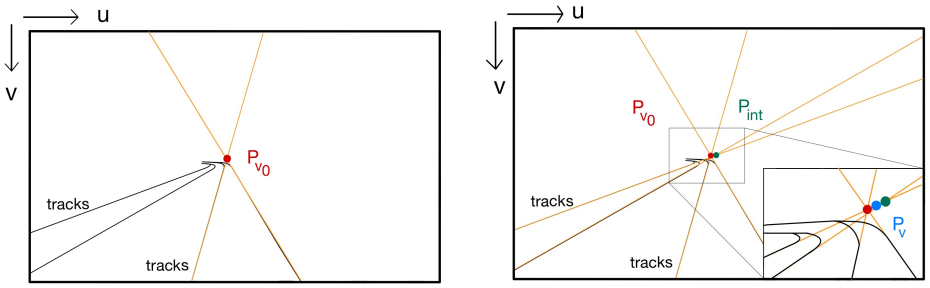


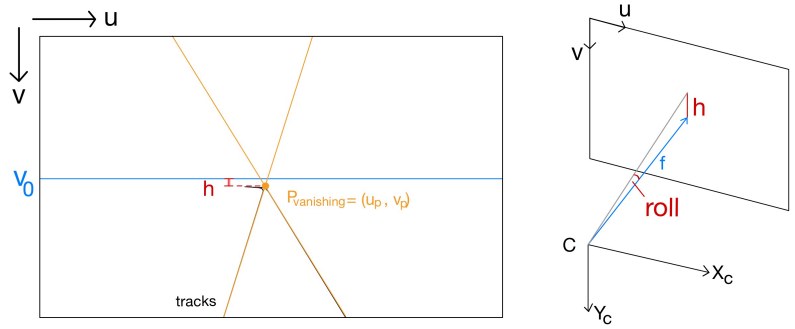
Figure 3.12: a) P_{v_0} at linearly extended railway tracks, (b) additional lines converging to the horizontal vanishing point

The intersection of the detected railways, denoted as P_{v_0}, is utilized as prior for the vanishing point, as depicted in Figure 3.12 a). However, to improve robustness, additional straight lines, represented by the orange lines in Figure 3.12 b) that should converge to the same vanishing point, are incorporated into its calculation. The enlarged portion of Figure 3.12 b) illustrates how the new vanishing point, denoted as P_v, is determined by taking the midpoint of the prior vanishing point P_{v_0} and the intersection point P_{int} of the newly added orange lines, both represented in blue and green respectively. To differentiate between lines converging to the various vanishing points present in an image, only those lines that intersect in close proximity to P_{v_0} are used. This method also has the added benefit of excluding lines from, for example, junction tracks that could potentially interfere with accurate vanishing point detection.

Roll Offset Estimate

In Equation 3.15, the height of the vanishing point v_p is taken to compute the roll together with the middle of the vertical height of the image v_0 and the focal length f_v in pixels in the vertical direction as proposed by Ross [17].

$$\varphi_{roll} = arctan(\frac{v_0 - v_p}{f_v}) \tag{3.15}$$



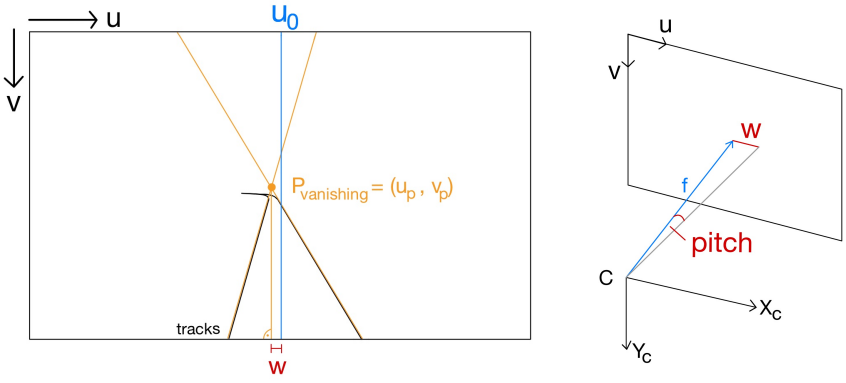
(a) (b)

Figure 3.13: a) Visualization of offset h and vanishing point, (b) visualization of focal length f and φ_{roll}

Pitch Offset Estimate

For the pitch offset, the difference w between the u_p-coordinate of the vanishing point and the horizontal middle line of the image u_0 is detected. This is then converted into an angle with the focal length f.

$$\theta_{pitch} = arctan(\frac{u_p - u_0)}{f_u}) \tag{3.16}$$



(a) (b)

Figure 3.14: a) Visualization of offset w and vanishing point, (b) visualization of focal length f and θ_{pitch}

3.2.4 Yaw Offset Estimate

For this estimation, a new Hough Transform of the image was computed with a smaller minimum line length. The idea was to detect the top and bottom contouring lines of the wooden sleepers between the railway tracks seen in Figure 3.15 in red. The candidate lines obtained were further filtered to be valid if their gradient was in a limited range, based on the assumption that the camera was reasonably positioned to have a low rotational angle around this axis. Additionally, only lines were accepted that were found within a predefined range of the horizontal middle of the detected railway tracks. The gradient of all valid candidate lines was computed and averaged to obtain ψ_{yaw}.

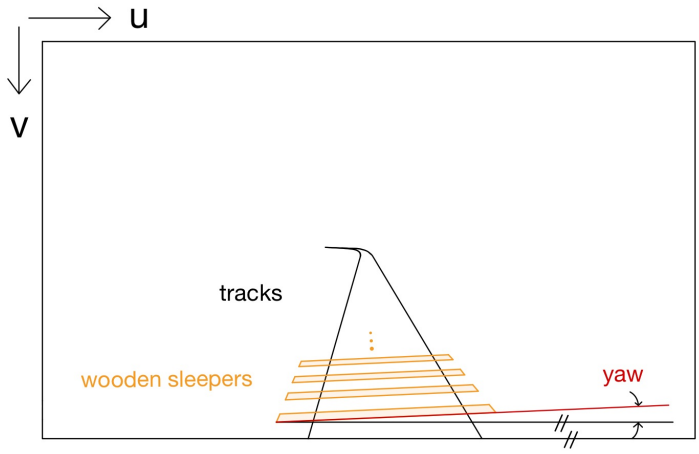


Figure 3.15: Geometry used for the estimation of ψ_{yaw}

Chapter 4

Results

The approach proposed above was applied to two pre-existing data sets of a Tatra-tram driving in Potsdam, Germany. This information led to the knowledge of the standardized railway track width of 1.435m, the height of the tram of 3.1m, and its width of 2.2m [27],[28]. These two data sets were from two cameras attached at the front of the tram, which gave us some prior knowledge of the accuracy of the obtained estimates. For other pre-existing data sets, there is a complete lack of ground truth.

Figure 4.1: Setup of the two cameras used for methodology validation

4.1 Height Offset Estimate

Table 4.1 shows the statistical summary, meaning the median and variance of the h_{offset} across 70 frames of each camera.

$\mathbf{h}_{offset}$	**Camera 1**	**Camera 2**
Median [m]	-1.05	-0.99
Variance [m^2]	$5.1 * 10^{-4}$	$2.99 * 10^{-4}$
Sample Number	70	70

Table 4.1: Statistical summary of h_{offset} of Camera 1 and Camera 2

Due to the lack of ground truth, the only manner of validating the height offset estimates of the individual cameras was by comparing the computed height of the camera with the ground truth camera setup height of 1.55m. The error was computed with Equation 4.1, where the absolute difference to the ground truth was calculated.

$$Error = |1.55 - h_c| \tag{4.1}$$

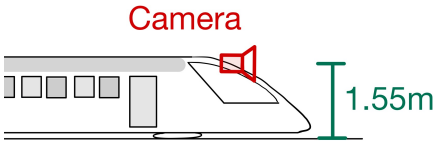


Figure 4.2: Visualization of camera height for setup

In Table 4.2, the individually computed values of the camera's height are presented. Compared to each other, they differ by 0.03m, where Camera 1 is 0.05m off of the ground truth value, and the error of Camera 2 is 0.13m.

Camera height	**Camera 1**	**Camera 2**
Median [m]	1.48	1.42
Error [m]	0.07	0.13

Table 4.2: Median of camera heights and corresponding error

To get the required height offset estimate, the height of the GPS off the floor was subtracted from the camera height. To determine this height, the floor elevation at the current position was leveraged. As the GPS is rigidly mounted on the tram, this value should be stable and would only vary depending on the preciseness of the floor elevation data. In Table 4.3, the statistics on the GPS height computed over 700 frames are visualized.

	GPS height from floor
Median [m]	0.43
Variance [m^2]	$9 * 10^{-4}$
Sample Number	700

Table 4.3: Statistical summary of GPS height from floor

4.2 Horizontal Offset Estimate

Likewise, without the ground truth values, the manner of validating the horizontal offset estimates hl_{offset} of the two cameras was by leveraging the known distance of 0.3m between them. By subtracting the horizontal offset values of the two cameras, this difference was compared to the known gap. Additionally, with the known width of the tram of 2.2m, a permissible upper bound on the estimates could be placed.

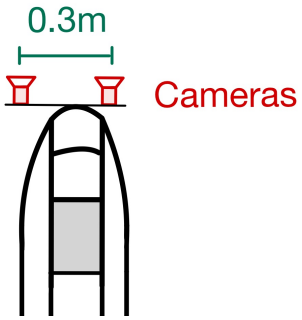


Figure 4.3: Visualization of the distance between the two cameras in setup

Table 4.4 shows the statistical summary of the horizontal offset estimate of both cameras. With these values, the cameras were separated by a distance of 0.31m.

$\mathbf{hl}_{offset}$	**Camera 1**	**Camera 2**
Median [m]	1.28	0.97
Variance [m^2]	$3 * 10^{-3}$	0.05
Sample Number	3	5

Table 4.4: Statistical summary of hl_{offset} of Camera 1 and Camera 2

4.3 Longitudinal Offset Estimate

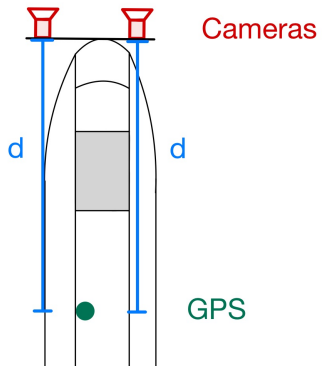


Figure 4.4: Visualization of distance between camera and GPS for setup

The longitudinal offset estimate was difficult to validate as no range on its value was known. The only prior knowledge given by the camera setup was, that these values should be of the same magnitude, as the two cameras were installed next to each other.

The following Table 4.5 represents the statistical data of the estimates made. By comparing the median of the l_{offset} estimates, the cameras had a difference of $0.44m$.

	Camera 1	**Camera 2**
Median [m]	0.04	0.48
Variance [m^2]	$3.2 * 10^{-2}$	$2.5 * 10^{-2}$
Sample Number	3	4

Table 4.5: Statistical summary of l_{offset} of Camera 1 and Camera 2

4.4 Rotational Estimates

The computed rotational estimates were compared to the ground truth in two different manners, which will be explained in the following two sections.

In both cases, the rotational estimates were converted into one rotation matrix, which was then transformed into three Euler angles using a predefined order. The same was done for the ground truth rotation matrix. The obtained three Euler angles per rotation matrix were compared to each other by taking their respective squared difference. This calculation is visualized in Equation 4.2, where the values from the ground truth have a subscript t, and the computed estimates a subscript e.

$$Error = \sqrt{(x_t - x_e)^2 + (y_t - y_e)^2 + (z_t - z_e)^2} \tag{4.2}$$

4.4.1 Rotation leveraging IMUs

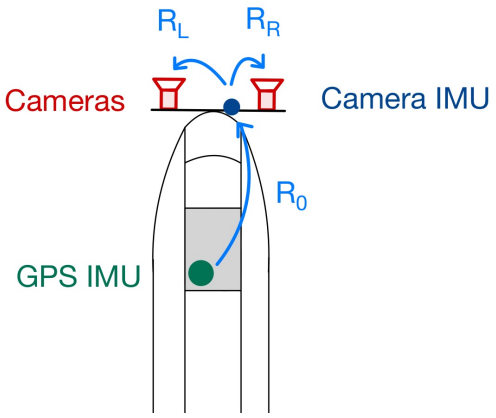


Figure 4.5: Visualization of rotations R_0, R_L and R_R

This first validation method compared the ground truth rotation matrix from the GPS to the respective camera. In detail, this entails the single steps from the GPS to the GPS-IMU, to the camera-IMU, and then to the corresponding camera. The rotation angle from the GPS to the GPS-IMU is zero, which is expressed as an identity matrix. The rotation from the GPS-IMU to the camera-IMU is defined as R_0, and the rotation from the camera-IMU to the left camera as R_L and R_R for the right camera.

$$R_{TOT} = R_{GPS/GPS\,IMU}\; R_0\; R_i \qquad i \in \{L, R\} \tag{4.3}$$

With this approach of comparing the estimated rotation per camera to the ground truth, the error in Table 4.6 was obtained and calculated with Equation 4.2.

	Camera 1	**Camera 2**
Error [°]	1.1	2.44

Table 4.6: Rotation errors of Camera 1 and Camera 2

4.4.2 Relative Rotation between Cameras

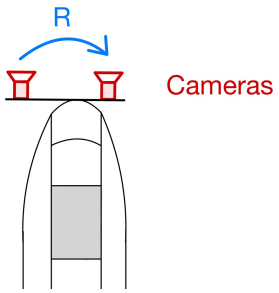


Figure 4.6: Visualization of relative rotation from Camera 2 to Camera 1

The second manner of validating the rotational values was by computing the relative rotation from the left camera to the right camera and comparing it to the ground truth attained from the calibration report of the camera setup. Table 4.7 presents the computed error by using the same error metric as applied to the first validation manner.

	Camera 1 to Camera 2
Error [°]	0.49

Table 4.7: Rotation error of Camera 1 and Camera 2

4.4.3 Statistics on Roll, Pitch, and Yaw

As in the previous validation manners, the individually computed rotational values roll, pitch, and yaw were inspected as a whole in a rotation matrix. Table 4.8 gives an overview of the statistics of the single values. Presented are the median and the variance over a certain sample number. The wooden sleepers needed for the yaw weren't found as often as the vanishing point needed for the roll and pitch, leading to a lower sample number.

		Camera 1	**Camera 2**
Roll			
	Median [°]	-0.31	0.73
	Variance	$4.46 * 10^{-4}$	$4.59 * 10^{-4}$
	Sample Number	70	70
Pitch			
	Median [°]	-0.48	1.57
	Variance	$9.92 * 10^{-5}$	$7.89 * 10^{-5}$
	Sample Number	70	70
Yaw			
	Median [°]	0.0	0.0
	Variance	0	0
	Sample Number	30	30

Table 4.8: Statistical summary of roll, pitch, and yaw of Camera 1 and Camera 2

4.5 Railway Map Projection

A method to ascertain all estimated parameters at once was to project the known coordinates of the railway tracks into the image with the computed pose. The accuracy of the estimated pose could be determined by the degree of alignment between the projected railway tracks and those present in the image.

The left side of Figures 4.7 and 4.8 feature the projected railway tracks from the map without any pose estimates, whereas on the right-hand side, the figures depict the projected railway tracks using the respective pose.

(a) (b)

Figure 4.7: Projection of railway track into the frame of Camera 1 (a) with no estimated pose, (b) with estimated pose

(a) (b)

Figure 4.8: Projection of railway track into the frame of Camera 2 (a) with no estimated pose, (b) with estimated pose

In order to determine which parameters were most in need of adjustment, they were altered manually and used to project the railway tracks into the image. Analysis of both camera views revealed that possibly only two parameters required larger modifications. In both cases, this specifically meant a reduction in horizontal offset by approximately 0.3m and an increase in camera pitch rotation angle by approximately 2° for Camera 1 and approximately 4° for Camera 2, to achieve the projection seen in Figure 4.9.

(a) (b)

Figure 4.9: Railway projection with corrected parameters (a) for Camera 2, (b) for Camera 1

Chapter 5

Conclusion

This thesis presents a comprehensive method for determining the relative pose between a monocular camera mounted at the front of a track vehicle to a preinstalled GPS. This method can be applied online and also to pre-existing data sets. The process of parameter determination was performed in a multi-staged fashion. The horizontal vanishing point found at the intersection of the railway tracks was utilized for the roll, pitch, and height estimates. For the other parameters, distinctive landmarks were leveraged, wooden sleepers for the yaw, and electricity poles for the horizontal and longitudinal estimates. As there is a complete lack of ground truth, the proposed approach was applied to two data sets for which some prior knowledge of their parameters existed.

The results were, on the one hand, validated with prior knowledge of the translational estimates and by leveraging the IMUs for the rotational estimates. On the other hand, they were validated as a whole in the degree of alignment of the railway track projections with the estimated pose to the railway tracks seen in the image.

When analyzing the results of the rotational estimates with the IMUs, distinguishing the source of error between the three angles was rather difficult as these three values were transformed into one matrix. The rotation error was found to be smaller for Camera 1 than for Camera 2. This is thought to partially originate in a less precise roll estimate for Camera 2, as the thereon-depending height estimate was also less accurate. When validating the rotation results with the railway projection, both cameras seemed to also have an error in their pitch. The reason for the incorrect roll and pitch estimates lies in the inaccurate vanishing point detection, which would originate in a faulty railway track detection. The railway tracks have a width, and depending on their detection, we have slight variations.

When examining the translational estimates, the height offset error for Camera 1 was 0.07m and 0.13m for Camera 2. A source of error could be the dependency on elevation data, which had a precision of 1m. The impact of this error was analyzed by observing the variance of the GPS height, which should be zero but had a value of $9*10^{-4}$ m^2. Additionally, the error of the camera roll propagated to this estimate.

Even though the difference between the horizontal offset estimates nearly corresponded to the known gap, this didn't help decipher whether these two values should be shifted more to the left or right. What became evident with the manually corrected railway projection was that, indeed, they needed to be moved by approximately 0.3m. This error is suspected to originate in their dependency on the roll, pitch, yaw, and height offset estimates.

The longitudinal offset estimates were found to be less precise, as they were dissimilar by a value of 0.4m. This can also be explained by their dependency on all five prior pose estimates. Even a slight change in roll and pitch had a large impact on the depth computation needed for this parameter. As we observed in the manually corrected projection, the pitch for both cameras was off, and additionally, the roll was slightly off as the thereof resulting height estimates were also slightly amiss.

Generally, it can be concluded that with precise vanishing point detection, the error in roll and pitch can be decreased. This would lead to smaller propagated errors, impacting the height-, horizontal- and longitudinal offsets. Nevertheless, when comparing the degree of alignment of the tracks a large improvement was seen. Adding a safety margin to the projected railways, the possibility of the methods application in the real world exists.

In future work, an option to improve the visual electricity pole detection could be to train a convolutional neural network to obtain the optimal set of hue, saturation, and value ranges. If this were to be precise, then the detection of the longitudinal offset wouldn't have to be restricted to situations where the track vehicle drives on straight tracks, and the estimate could be computed more often. Additionally, a prior frame evaluation could be done to select keyframes automatically. This could be implemented in the manner proposed by Schneider [29], where a score is computed depending on the amount of information visible in a frame. When the score passes a predefined threshold, the frame is marked as a keyframe.

Bibliography

[1] L. Dell'Olio, A. Ibeas, and P. Cecin, "The quality of service desired by public transport users," *Transport Policy*, vol. 18, no. 1, pp. 217–227, 2011.

[2] D. Trentesaux, R. Dahyot, A. Ouedraogo, D. Arenas, S. Lefebvre, W. Schön, B. Lussier, and H. Chéritel, "The autonomous train," in *2018 13th Annual Conference on System of Systems Engineering (SoSE)*. IEEE, 2018, pp. 514–520.

[3] Z. Wang, X. Wu, Y. Yan, C. Jia, B. Cai, Z. Huang, G. Wang, and T. Zhang, "An inverse projective mapping-based approach for robust rail track extraction," in *2015 8th International Congress on Image and Signal Processing (CISP)*. IEEE, 2015, pp. 888–893.

[4] H.-Y. Lin *et al.*, "3d object detection and 6d pose estimation using rgb-d images and mask r-cnn," in *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2020, pp. 1–6.

[5] B. Caprile and V. Torre, "Using vanishing points for camera calibration," *International journal of computer vision*, vol. 4, no. 2, pp. 127–139, 1990.

[6] C. Bräuer-Burchardt and K. Voss, "Image rectification for reconstruction of destroyed buildings using single views," in *Virtual and Augmented Architecture (VAA'01)*. Springer, 2001, pp. 159–170.

[7] L. Grammatikopoulos, G. Karras, and E. Petsa, "An automatic approach for camera calibration from vanishing points," *ISPRS journal of photogrammetry and remote sensing*, vol. 62, no. 1, pp. 64–76, 2007.

[8] R. Cipolla, T. Drummond, and D. P. Robertson, "Camera calibration from vanishing points in image of architectural scenes." in *BMVC*, vol. 99. Citeseer, 1999, pp. 382–391.

[9] E. Namazi, R. Mester, C. Lu, and J. Li, "Geolocation estimation of target vehicles using image processing and geometric computation," *Neurocomputing*, vol. 499, pp. 35–46, 2022.

[10] K. M. Dawson-Howe and D. Vernon, "Simple pinhole camera calibration," *International Journal of Imaging Systems and Technology*, vol. 5, no. 1, pp. 1–6, 1994.

[11] J. Weng, P. Cohen, M. Herniou *et al.*, "Camera calibration with distortion models and accuracy evaluation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 14, no. 10, pp. 965–980, 1992.

[12] M. J. Magee and J. K. Aggarwal, "Determining vanishing points from perspective images," *Computer Vision, Graphics, and Image Processing*, vol. 26, no. 2, pp. 256–267, 1984.

[13] Z. Liu and T. Chen, "Viewing transform of image based on vanishing point," in *2010 International Conference on Intelligent Computing and Integrated Systems*. IEEE, 2010, pp. 180–184.

[14] B. T. Nassu and M. Ukai, "Rail extraction for driver support in railways," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 83–88.

[15] E. Namazi, R. N. Holthe-Berg, C. S. Lofsberg, and J. Li, "Using vehicle-mounted camera to collect information for managing mixed traffic," in *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE, 2019, pp. 222–230.

[16] J. Matas, C. Galambos, and J. Kittler, "Robust detection of lines using the progressive probabilistic hough transform," *Computer vision and image understanding*, vol. 78, no. 1, pp. 119–137, 2000.

[17] R. Ross, "Vision-based track estimation and turnout detection using recursive estimation," in *13th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2010, pp. 1330–1335.

[18] L. und Geobasisinformation Brandenburg. Digitales geländemodell.

[19] Z. Qi, Y. Tian, and Y. Shi, "Efficient railway tracks detection and turnouts recognition method using hog features," *Neural Computing and Applications*, vol. 23, no. 1, pp. 245–254, 2013.

[20] R. Ross, "Track and turnout detection in video-signals using probabilistic spline curves," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2012, pp. 294–299.

[21] C. Shan-shan, Z. Wu-heng, and F. Zhi-lin, "Depth estimation via stereo vision using birchfield's algorithm," in *2011 IEEE 3rd International Conference on Communication Software and Networks*. IEEE, 2011, pp. 403–407.

[22] F. Sun, Y. Zhou, C. Li, and Y. Huang, "Research on active slam with fusion of monocular vision and laser range data," in *2010 8th World congress on intelligent control and automation*. IEEE, 2010, pp. 6550–6554.

[23] Z. Yan, X. Xiaodong, P. Xuejun, and W. Wei, "Mobile robot indoor navigation using laser range finder and monocular vision," in *IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003*, vol. 1. IEEE, 2003, pp. 77–82.

[24] X. Yin, X. Wang, X. Du, and Q. Chen, "Scale recovery for monocular visual odometry using depth estimated with deep convolutional neural fields," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5870–5878.

[25] H. E. Burton as translator, "The optics of euclid," in *Journal of the Optical Society of America*, 1943, p. Vol. 35 Nr. 5.

[26] A. Kar, S. Tulsiani, J. Carreira, and J. Malik, "Amodal completion and size constancy in natural scenes," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 127–135.

[27] M. Dittrich. Strassenbahnen in potsdam.

[28] R. Leichsenring. Tatra kt4d linienwagen.

[29] T. Schneider, M. Li, M. Burri, J. Nieto, R. Siegwart, and I. Gilitschenski, "Visual-inertial self-calibration on informative motion segments," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 6487–6494.