

Analysis of lateral vehicle position using naturalistic driving data

Master's thesis in Automotive Engineering

ALVAREZ SALAZAR JORGE ROBERTO
KLEINERT FABIAN

MASTER'S THESIS IN AUTOMOTIVE ENGINEERING

Analysis of lateral vehicle position using naturalistic driving data

ALVAREZ SALAZAR JORGE ROBERTO
KLEINERT FABIAN

Department of Applied Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden 2017

Analysis of lateral vehicle position using naturalistic driving data
ALVAREZ SALAZAR JORGE ROBERTO
KLEINERT FABIAN

© ALVAREZ SALAZAR JORGE ROBERTO, KLEINERT FABIAN, 2017

Master's thesis 2017:32
ISSN 1652-8557
Department of Applied Mechanics
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: +46 (0)31-772 1000

Cover:

Front view of a random event with a comparison of the lane offset signal of the CAN-bus and the lane offset signal from the Video processing.

Chalmers Reproservice
Göteborg, Sweden 2017

Analysis of lateral vehicle position using naturalistic driving data
Master's thesis in Automotive Engineering
ALVAREZ SALAZAR JORGE ROBERTO
KLEINERT FABIAN
Department of Applied Mechanics
Chalmers University of Technology

ABSTRACT

Traffic crashes have a significant impact on humans lives and the world economy. Understanding crash causation is an important step to prevent crashes from happening. One way of gaining information about crashes and human driver behaviour is through the analysis of naturalistic driving data. Often the challenge arises to generate the maximum information from a given database to answer a certain problem or research question.

The thesis covers the research question, of how to analyze large quantities of video material in a quantifiable and efficient way to gain information about the lateral position of a vehicle.

The research addresses that problem by developing a tool to extract the lateral vehicle position of a front view video and using reference data to evaluate its performance and sensitivity. A reliable set of CAN-Bus data serves as reference data for the vehicle position. The vehicle position is extracted from the video through multiple image processing steps. Evaluations are performed to assess precision and accuracy of the method. A sensitivity analysis is conducted to determine the impact of providing correct camera parameters. The method is applied to a new data set to ensure versatility.

The results show that the method is reliable but highly sensitive to the orientation and position of the camera relative to the vehicle. The overall estimation of the precision shows a systematic error in the measurements caused by the camera parameters. The sensitivity evaluation shows that the height and the roll of the camera have a significant impact on the precision of the method.

The developed tool can be used to determine the lateral vehicle position. Tuning of the tool is needed to further improve accuracy and precision of the tool.

Keywords: Naturalistic studies, image processing, lateral vehicle position, camera parameters.

ACKNOWLEDGEMENTS

We would like to thank the Volvo Cars Corporation for giving us the opportunity to work on this project and sponsoring us along the way. A special thank you to our supervisors Daniel Nilsson and Fredrik Granum, and our examiner Marco Dozza for the continuous help and support.

A mis papás, Jorge y Ana, a mi hermana Kary y a mis tías-mamás Titi, Conny y Martita. A todos ellos por su amor y apoyo incondicional en cada locura de mi vida. A mis amigas Karen, Miriely y Rebeca por su amistad y comprensión toda durante mi estancia en Suecia. Till min vän AnnaFrida som har varit min familj här på andra sidan jorden. Finalmente, a CONACYT por ser el patrocinador este sueño.

J.R. Alvarez

Ich möchte meiner Familie und meinen Freunden für die bedingungslose Unterstützung, während meines Studiums und dieser Arbeit danken. Des Weiteren danke ich der Baden-Württemberg Stiftung für die Förderung meines Auslandsaufenthaltes. Die Erfahrungen, welche ich während meines Studiums in Schweden sammeln durfte, werden mich ein Leben lang begleiten und meine zukünftigen Entscheidungen prägen.

Fabian Kleinert

Nomenclature

\bar{E}	Weighted mean
\bar{e}	Mean error of all the events
Δe	Mean squared error of all the events
ϵ_{meters}	Sensibility error in meters
$\epsilon_{percent}$	Sensibility error in percentage
Λ	Standard deviation of the Weighted mean
λ	Arbitrary constant that relates the distance from the camera to the object of interest
σ	Standard deviation
C	Camera center. Principal Point. Location of the camera sensor relative to the image plane
C_x	Horizontal coordinate of the Principal point in pixels
C_y	Vertical coordinate of the Principal point in pixels
CAN	Controller Area Network: A communications network, commonly used in vehicles.
DS	Data Set.
DS^R	Data Set of Reliable cases.
DS^{NR}	Data Set of Non-Reliable cases.
E_i	Error between the <i>LOV</i> and <i>LOC</i> of the event i
EC	Event combining all the curvy road parts
ED_i	Event i during a day drive
EN_i	Event i during a night drive
ES	Event combining all the straight road parts
EV	Ego vehicle: The vehicle which the speaker would be hypothetically driving. The vehicle of interest.
F	Camera Focal length in millimeters
f_x	Horizontal Focal length in pixels per millimeter
f_y	Vertical Focal length in pixels per millimeter
FOT	Field Operational Test: Test to evaluate the performance of a system under realistic conditions.
K	Calibration matrix. Matrix with the properties (in pixels) of a camera.
K_{eyesOR}	Calibration matrix for the videos from the Eyes on Road database.
K_{lytx}	Calibration matrix for the videos from the Lytx database.
LDC	Lane Detection Confidence Signal: Discrete signal that determines how much trustworthy is the <i>LOC</i> .
LOC	Lane Offset CAN: Lane offset signal from the CAN network.
LOC_i^{CAN}	Original Lane Offset signal of the event i .
LOC_i^E	Lane Offset signal of the event i with a process of extrapolation to remove the NaN elements
LOC_i^{NaN}	Lane Offset signal of the event i with the locations of the NaN elements.

LOV Lane Offset Video: Lane offset signal from a video.
 LOV_i^L LOV for the Left side in the frame i .
 LOV_i^R LOV for the Right side in the frame i
 LOV_i^E LOV of the event i with a process of extrapolation to remove the NaN elements
 LOV_i^{NaN} LOV of the event i with the locations of the NaN elements.
 $LOV_{original}$ LOV for the sensitivity evaluation with the default position of the camera
 $LOV_{parameter}$ LOV for the sensitivity evaluation changing one of the parameters of the position of the camera
 LV Leading vehicle(s): The vehicle(s) which the speaker would see from inside of his car.
 NaN Not a Number: Undefined value. Empty element in an array.
 P Camera Matrix. Matrix which maps from pixel coordinates to Real World coordinates
 ROI Region(s) Of Interest: Area in an image where the information is to be extracted.
 U Image Vertical resolution in pixels
 V Image Horizontal resolution in pixels
 W Vector with the Real World coordinates
 w Vector with the pixel coordinates
 X Lateral axis
 X_C Lateral axis with its origin in the center of the camera
 x_p Lateral axis with its origin in the upper left corner of the image
 X_W Lateral axis in the Real World coordinate system. The origin is decided at the user's convenience
 x_d Horizontal coordinate of a pixel (relative to x_p) in a distorted image
 x_u Horizontal coordinate of a pixel (relative to x_p) in an undistorted image
 Y Vertical axis
 Y_C Vertical axis with its origin in the center of the camera
 y_p Vertical axis with its origin in the upper left corner of the image
 Y_W Vertical axis in the Real World coordinate system. The origin is decided at the user's convenience
 y_d Vertical coordinate of a pixel (relative to y_p) in a distorted image
 y_u Vertical coordinate of a pixel (relative to y_p) in an undistorted image
 Z Longitudinal axis
 Z_C Longitudinal axis with its origin in the center of the camera
 Z_W Longitudinal axis in the Real World coordinate system. The origin is decided at the user's convenience
 GUI Grafial User Interface

List of Figures

2.1	Camera Position relative to the vehicle in Real-World coordinate System.	5
2.2	Overall process	8
2.3	Processing the EOR data set	9
2.4	Processing the LOC	11
2.5	Original Lane-Offset Signal	12
2.6	Reliable stretches of the Lane-Offset-Signal	12
2.7	Edited Lane-Offset-Signal	13
2.8	Video processing	14
2.9	Examples of calibration images	15
2.10	Visualization of the camera and image coordinate system.	16
2.11	Radial distortion types.	17
2.12	Tangential distortion types.	17
2.13	Coordinate systems.	18
2.14	Removal of regions of no interest	19
2.15	Vehicle Coordinate systems.	21
2.16	Quantitative Evaluation	23
2.17	Visual Evaluation	27
2.18	Comparison of signals in the frontal view.	28
2.19	Camera Range in real world units	29
2.20	Front view	29
2.21	Top view	29
2.22	Visualization of the front & top view. Top view already with the interpolation	30
2.23	GUI for comparing the CAN-Bus Signal to the Camera Signal	31
2.24	Interface 1, GUI for Lytx videos	32
2.25	Interface 2, GUI for Lytx videos	33
3.1	Effects of Roundabouts	35
3.2	Effects of Crosswalks graph	35
3.3	Effects of Crosswalks Image	36
3.4	Effects of extrinsic parameters	36
3.5	Effects of pitch rate	37
3.6	Histogram of error of a single event	38
3.7	Normal distribution of error of a single event	39
3.8	Evaluation Categories	39
3.9	General error	40
3.10	Daytime error	41
3.11	Nighttime error	41
3.12	Straight road error	41
3.13	Curvy road error	41
3.14	No acceleration error	41
3.15	Acceleration error	41
3.16	Normal distribution for each category	42
3.17	Relative error [angles].	43
3.18	Relative error [distances].	43
3.19	Error [angles].	43
3.20	Error [distances].	43
3.21	Plot from GUI for a Lytx video	44
A1	Undistort function	II

List of Tables

2.1	Video properties	4
2.2	Extrinsic Camera Parameters	5
2.3	Internal Camera Parameters EOR	5
2.4	List of CAN signals and explanation	6
2.5	Video properties	6
2.6	Internal Camera Parameters Internal	7
2.7	Explanation of flow chart shapes	7
2.8	Properties of the Lane-Detection-Confidence Signal	10
2.9	Description & computation of intrinsic parameters.	15
2.10	Explanation of GUI birdvie	31
2.11	Explanation of Interface 1, GUI for Lytx.	33
2.12	Explanation of Interface 2, GUI for Lytx	34
3.1	Analysis of a single event	38
3.2	Analysis of different categories	40

CONTENTS

Abstract	i
Acknowledgements	i
Nomenclature	iii
List of Figures	v
List of Tables	vi
Contents	vii
1 Introduction	1
1.1 Research question and aim	1
1.2 Limitations	1
1.3 Background	1
2 Method	4
2.1 Technical preconditions	4
2.2 Method Structure	7
2.3 Data set processing	9
2.4 LOC processing	11
2.5 Video processing	13
2.6 Evaluations	22
2.7 Versatility	31
3 Results	35
3.1 Qualitative Results	35
3.2 Quantitative Results	38
3.3 Sensitivity results	42
3.4 Versatility results	43
4 Discussion and further work	45
4.1 Qualitative results	45
4.2 Quantitative results	45
4.3 Sensitivity analysis	46
4.4 Versatility	46
5 Conclusions	47
References	48
Appendices	I
.1 Code	II

1 Introduction

Traffic crashes caused 1.25 million road traffic deaths and is in addition to that the leading cause of death among those aged 12-29. Furthermore it costs governments roughly 3% of its GDP. These numbers show the significant human and economic impact and the need for further research to avoid crashes or mitigate its outcome.[WHO 2015]

A crash can be chronologically split into three phases: The pre-crash phase, the in-crash phase and the post-crash phase. Safety measures to counteract crashes can be split into active safety measures, which aim at avoiding an encounter or mitigating its results and passive safety measures, which aim at protecting the vehicle occupants during a collision. Consequently active safety covers the pre-crash and partly the in-crash phase and passive safety covers partly the in-crash and the post-crash phase. Through advances in technology and manufacturing active safety measures are gaining more and more importance in the eyes of researchers and vehicle makers.

The initial step to understanding a crash is understanding its causation. Recent research shows that a high percentage of accidents occur due to driver impairment. This makes studying human driving behaviour immensely important for modern crash research and the development of active safety systems to counteract critical driving scenarios, which are leading to crashes. One way of gaining information about human driver behaviour is through the analysis of naturalistic driving data (NDD).

NDD are collected in real traffic by road users performing their every-day driving activities using instrumented vehicles. It contains information about the driver, the vehicle and the driving environment. A key factor is, that data are unobtrusively acquired. The amount of data and the type of data available vary from data base to data base often depending on the original purpose of a study.[Bärgman 2017]

When analyzing NDD the challenge arises to define what to look for and match it with the type of data provided in a certain data base. Once that is decided, the next step is to find the often very limited number of interesting incidents within the data. These incidents must be found and analyzed according to the research interest. Therefore, especially in a commercial environment, often the challenge arises to generate the maximum information from a given data base to answer a certain problem or research question.

In the particular case, which will be covered in this thesis, the data base is limited to only one camera view of the road laying ahead. The video material from this front view camera shall be used for analyzing the position of the car within its lane in reference to the lane markings on each side. The data can be used for detecting run-of-road incidents or sudden lane changes within a data set. These incidents then can be further used for studying driver behaviour in the pre-crash phase prior to the lane departure. The analysis of driver behaviour can become a prerequisite for setting system requirements and performing safety benefit analysis.

1.1 Research question and aim

1.1.1 Research question

How to analyze large quantities of video material in a quantifiable and efficient way to gain information about the lateral position of a vehicle?

1.1.2 General aim

This thesis aims to provide the lateral position of a passenger vehicle over time within its lane through analyzing and processing video data from a single camera pointed towards the road ahead of the vehicle. (front view)

1.1.3 Specific Aims

- Creation of a tool to extract a lane offset vector from a front view video from the EOR data base

- Evaluation of the precision of the tool for different light, road and driving conditions using corresponding and high confidence CAN-Bus signals from the EOR data base.
- Evaluation of the sensitivity of the tool for changes in extrinsic camera parameters
- Ensure versatility through modifying the tool to enable application to various databases with front view videos available
- Application of the tool to videos from the LYTX data base through a GUI to improve usability.

1.2 Limitations

The data set provided is a subset from the EOR data base. The subset is created based on the following criteria:

- The vehicle model is a Volvo V60 2014 model
- The maximum lane width is four meters
- The lateral movement of the car has to be less than 2.5m/s
- The lateral distance between the car and one of the lane markings has to be less than 0.7m
- The maximum yaw rate is limited to 10°/s
- No direction indicator are used
- All previous criteria are fulfilled for at least a second
- 15s before and after the 1s-event are added

The development of the tool is limited to software standards set by Volvo Cars Corporation to ensure compatibility with existing software. The software used for the project is explained in section 2.1.1.

1.3 Background

1.3.1 Field tests

Field tests are used to extract information in a real-traffic environment. These tests often involve an experimental protocol which may define the driver conditions of the tests such as driver type, itinerary, etc. They may require instrumented vehicles and infrastructures [Dozza 2016].

Field tests can be divided in two main types depending on their purpose. Naturalistic Studies, Field Operational Tests (FOT) [Dozza 2016].

While the purpose of a Naturalistic study is purely observational, the aim of a Field Operational Test is the evaluation of a system [Bärgman 2017]. In some cases both groups can intersect, leading to Naturalistic Field Operational tests (NFOT).

Naturalistic data is a collection of signals taken without any manipulation of the observer. It allows the extraction of information in authentic situations. This realistic feature is crucial in the development of every emergent technology.

In accidentology, naturalistic driving data bases allow car manufactures to understand the driving conditions (driver, road, car, weather, etc.) behind a crash and develop both active and passive safety systems to avoid and/or mitigate car crashes. At a infrastructural level, a naturalistic driving database helps in the mobility analysis and traffic efficiency [Dozza 2016] .

100car, *8truck*, *euroFOT*, *SHRP2* and *LYTX* are some examples of Naturalistic Field Operational Tests.

The way in which the data bases acquire this information is by using a data-logger, which records signals while the vehicle is moving. Most of the material involves the video of the frontal view and may include the view of the inside of the vehicle or the rear view of the car. In these videos, although the frame rate and resolution are set to reduce the file sizes, the amount of data that is acquired is bigger in comparison to the signals of that could be gotten from the CAN bus. Furthermore, in some data bases the only available signal is the recorded video. This property of the videos generates the possibility to extract more information from the driving conditions. However, extracting the relevant parameters involves a tedious, time consuming and low repetitively activities because the current state of the art relies mainly on human eyes. Some of these activities, for example, consist of measuring the position of the vehicle in every frame of a video using rulers or scales. In order to advance in traffic safety research, new methods to automatically process video data and extract vehicle information are needed. This thesis covers that by extracting lane offset from video using different image analysis techniques in order to enable traffic safety analysis, especially in critical situations from the EOR and the LYTX data bases.

1.3.2 Image analysis

In general terms, to achieve automatic video annotation, the basic steps are:

- Highlight and extract the position of the features of interest in each frame.
- Get the equivalences between pixels and scientific units.
- Compute the corresponding transformations from pixel position to scientific units (meters, millimeters, etc).

The first step will depend on the application, but the final steps are a set of equations that are well known in the field of computer vision.

In the automotive sector, there are some works which aim to make the detection and real world measurements more accurate or more efficiently.

1.3.2.1 Real world measure extraction

The work of Akamatsu [Akamatsu, Dong, and Hirota 2013] shows an interesting method to measure the position of moving objects. This paper presents a solution, where the camera is also moving. The presence of an accelerometer beneath the camera carries out the measurement of the kinematic parameters to compensate the relative velocities between the moving camera and the moving object.

In the field of image analysis, interesting reviews have been done. The work of Neethu et al. John, Anusha, and Kutty 2015 seems to be a good method to detect road features by using the vanishing point using a single camera. The disadvantage of this method is that it requires a color video/image. The proposed method is intended to be not susceptible to the amount of channels in the video.

The work of Meng & Wang Meng and Yifei 2016 provides a good starting point in the use of image analysis techniques for the manipulation of Naturalistic Data. In their work, the paths of the *EV* and the *LV* are reconstructed in real world coordinates using videos of naturalistic data.

1.3.2.2 Lane markings detection

In general terms, the most common and robust procedure for line detection in an image seems to be as follows.

1. Pre-process the image to highlight the regions of interest (ROI). This step may include a sequence of different filters or a simple noise reduction.
2. Convert the image to a binary image. This is a black and white image. White pixels are set to be the areas of interest.

3. Convert the areas of interest into lines. There are many methods to carry out this step. The review of Tang [Tan et al. 2013] proposes the use of the image gradient amplitude and direction. The review of Grompone von Gioi [Grompone von Gioi 2014] proposes a method using different considerations in the analysis of gradients. Nowadays, the Hough transform [Hough 1962; Duda and Hart 1972; Fokkinga 2011] is the most common and robust method to extract lines using a parametrization of angles and lengths.

2 Method

2.1 Technical preconditions

2.1.1 Software & Hardware

The computations were done with a laptop of the type Dell Latitude E6230 2.60 GHz, 8 GB of memory, and running Windows 7 Enterprise. The program to keep track of the changes in the scripts was the open source subversion control system TortoiseSVM 1.9.

To make the code compatible with the software used in Volvo Cars Corporation, all the functions and scripts have been developed using Matlab 2013b (The Mathworks, Natick, MA) with the following toolboxes available:

- Image Processing
- Signal Processing
- Application Development

2.1.2 Data set

2.1.2.1 Eyes on Road(EOR)

The EOR study was originally performed to collect a unique data set about driver gaze behaviour in modern driving conditions. It was also the test environment for future driver monitoring camera development and function testing.

Video files

The available data add up to 34h of video material. It is split into 4000 events with an average length of roughly 31 seconds.

The characteristics of the videos are shown in the table 2.1.

Table 2.1: Video properties

Parameter	Value	Units
Resolution	288x355	Pixels
Frame rate	10	fps
Color channels	1	Grayscale

External characteristics

The videos are recorded with a camera, which is oriented towards the windshield to record the road ahead as it is shown in figure 2.1 and is located with the values shown in table 2.2.

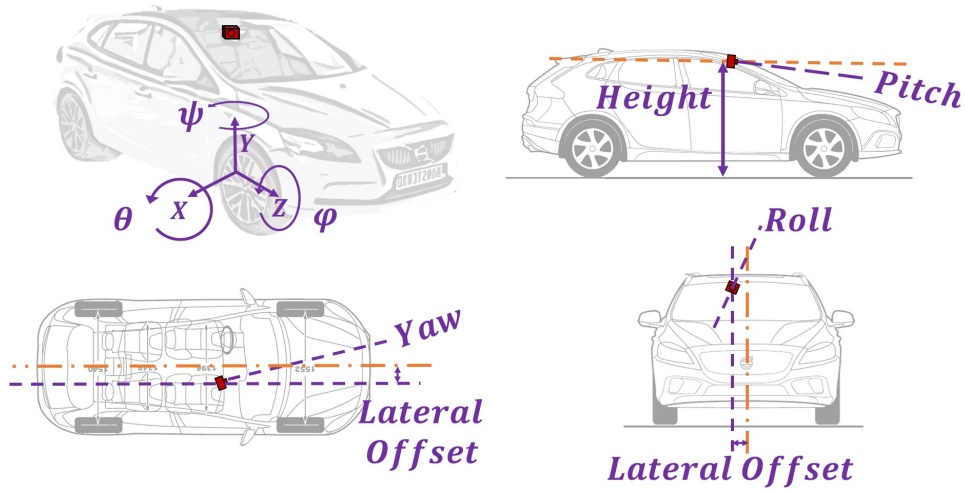


Figure 2.1: Camera Position relative to the vehicle in Real-World coordinate System.

Table 2.2: Extrinsic Camera Parameters

Parameter	Value
Height	1.36 [m]
Lateral Offset	0.19 [m]
Yaw	0 [°]
Pitch	0 [°]
Roll	0 [°]

Internal characteristics

The camera used in the EOR data base to get the videos has the calibration matrix shown in table 2.3.

Table 2.3: Internal Camera Parameters EOR

Parameter	EOR
f_x	255.82
f_y	280.99
C_x	179.39
C_y	143.19

Signal files

A set of CAN-Bus-Signals matching the video data is part of the method. The set contains the following signals:

Table 2.4: List of CAN signals and explanation

Signal name	Description
Vehicle Speed	The vehicle longitudinal speed in kilometer/hour
Yaw Rate	Rate, at which the vehicle is turning around the vertical axis in degrees/second
Left Lane Offset	Position from the center of the vehicle to the inside of the lane marking in meters
Right Lane Offset	Position from the center of the vehicle to the inside of the lane marking in meters
Lane width	The lane width from the inner sides of the lanes in meters
Longitudinal Acceleration	Measurement of the longitudinal translational acceleration
Left Lane Detection Confidence	Measurement for the level of confidence in the Left Lane Offset signal
Right Lane Detection Confidence	Measurement for the level of confidence in the Right Lane Offset signal
Day Light	Measurement for the light environmental conditions, deciding if it is day of night

2.1.2.2 LYTX

LYTX, formerly known as DriveCam, is a company that designs, manufactures and sells road safety programs. In the development of safety programs, LYTX helps in road enforcement to see unsafe driving conditions and behaviours that lead to crashes. Like a flight "black box", an small data recorder constantly collects audio, video, speed and force vectors. For this thesis only the video data is available. [Wolfe and Gwin 2010; Bigelow 2009].

The videos are usually taken with approximately 120-degree field of view out the front windshield, a resolution of 256 x 200 pixels and a frame rate of 4 Hz[Carney et al. 2015].

In this thesis, the data bases of videos will be used at the end, as the main data of interest.

Video files

Five videos with an average length of roughly 15 seconds are available.

The characteristics of the videos are shown in table 2.5.

Table 2.5: Video properties

Parameter	Value	Units
Resolution	352x620	Pixels
Frame rate	4	Hertz
Color channels	3	RGB

External characteristics

The camera orientation is similar to the one explained in section 2.1.2.1. As opposed to the EOR-videos, the exact camera position is unknown and varies for different videos.

Internal characteristics According to Meng [Meng and Yifei 2016], the camera used to create the LYTX videos has the calibration parameters shown in table 2.6 below.

Table 2.6: Internal Camera Parameters Internal

Parameter	LYTX
f_x	534.00
f_y	522.99
C_x	313.90
C_y	174.68



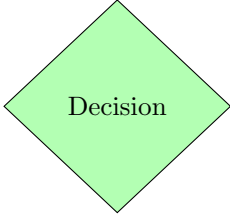
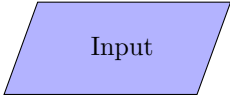


Signal files

No signals are available. Videos are the only source of information.

2.2 Method Structure

The method consists out of several steps illustrated in the flow chart 2.2 below. The different shapes used in the following flow charts are explained in table 2.7. The first step is processing the Lane-Offset-Signal(LOC) from the EOR-CAN-Bus-Signals to assure a high level of reliability of the signal and improve comparability. As a second step the video is processed to generate Lane-Offset-Signal(LOV) from the EOR video data. These signals have to be further processed in a third step. At this point there are comparable lane offset signals. In the following two steps two different types of evaluation are performed. The first one is a visual evaluation. Both the processed LOC as well as the processed LOV are fed into the front view and the aerial view of the corresponding video. These animations can give a visual impression of the accuracy of both signal types. The difference in both signals is further evaluated in different driving, light and road conditions using descriptive statistics. As a final step the tool for the extraction of the lane offset signal is applied to a video from the LYTX data base to ensure versatility.

Table 2.7: Explanation of flow chart shapes

Symbol	Meaning	Explanation
	Terminal	Rounded rectangle signaling the start or end of a process
	Process	Rectangle signaling that something is performed
	Decision	Diamond representing a necessary decision
	Input or Output	Parallelogram showing receiving data and displaying data
	Connector	Arrow, which comes from on symbol and ends at another symbol, signaling that control passes to the symbol the arrow points to.
	complex process	Dashed rectangle showing complex processing steps, which maybe detailed in a separate flow chart

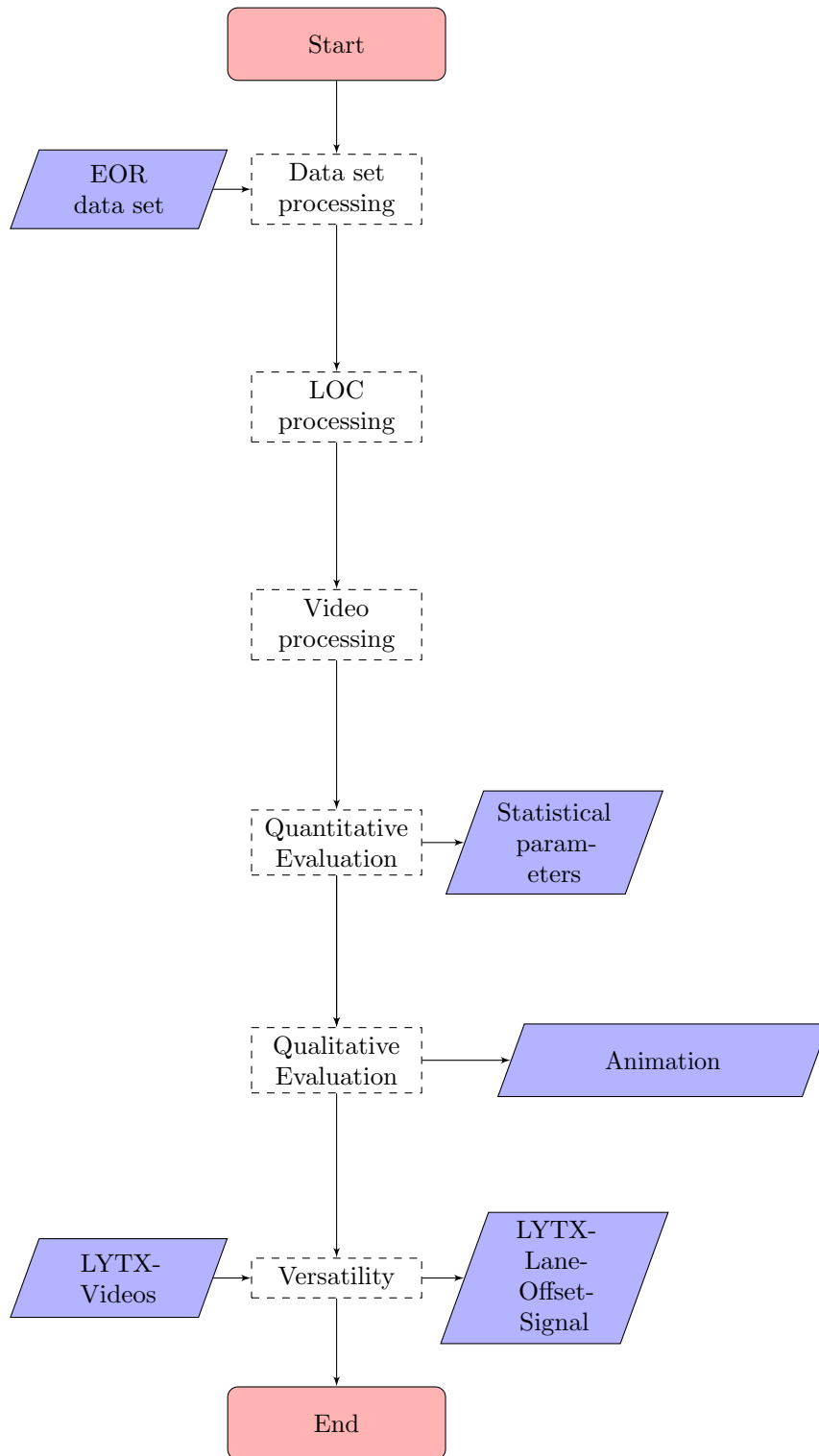


Figure 2.2: Overall process

2.3 Data set processing

The data set has to be processed to assure comparability between LOC and LOV. Figure 2.3 shows the necessary steps. The data has to be converted to LYTX conditions to make the overall method applicable to LYTX video material. Furthermore parts of the data have to be preselected to compare only reliable measurements.

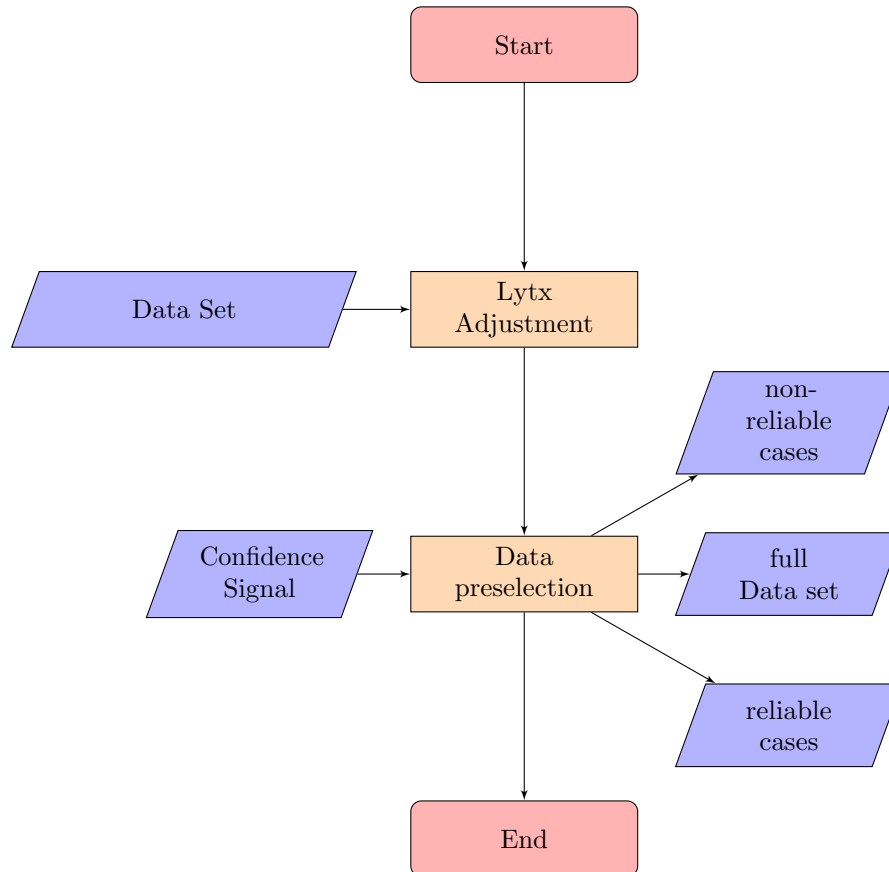


Figure 2.3: *Processing the EOR data set*

2.3.1 Adjustment to LYTX

As mentioned in section 2.1.2.1 the properties of the videos and signals differ from data base to data base. The resolution of EOR videos is lower than the one of LYTX videos, this property does not affect the performance of the method. If it works with a low resolution video, it can also work with higher resolutions. The number of color channels is lower and unique in an EOR video, while a LYTX video can have the three RGB channels. This characteristic causes the same effect as with the resolution. If it works with one channel, it works with three channels. The sampling frequency is a different matter. EOR videos are sampled with 10Hz, while LYTX is sampled with 4Hz. The current method is frequency independent however, to assure that this method works with data bases which were taken with a low sampling rate, both the videos and the signals are re-sampled to have a frequency of 4 Hz.

2.3.2 Event preselection

The LOC is taken as the ground truth to which the LOV is compared to. To compare both reliably, the quality of the LOC is examined through using the Lane Detection-Confidence-Signal (LDC).

The data set contains the LOC for multiple Events

$$DS_i = [LOC_1, LOC_2, \dots, LOC_m] \quad (2.1)$$

The Lane Detection Confidence Signal (LDC) assesses the quality of the LOC during an event. It consists out of 5 different categories.

$$LDC_i = [q_1, q_2, \dots, q_m] \quad q \in 0, 1, 2, 3, 4 \quad (2.2)$$

Table 2.8: Properties of the Lane-Detection-Confidence Signal

Value	Meaning
0	No Confidence
1	Low Confidence
2	Medium Confidence
3	Medium High Confidence
4	High Confidence

The data set is processed using the confidence signal. Only events with a high number of measurements with medium high or high confidence are taken into account for the later process. The specific threshold for the amount of measurements can be set by the user. The initial threshold is set to 60% (60% of the values must have confidence 4 or 5). This creates a balance between on the one hand getting a sufficient amount of reliable measurements and not reducing the data set to a too small amount of events.

In addition to that the user has the option to run the algorithm for non-reliable cases. This can be interesting to see how the algorithm performs for cases when the LOC can not be taken as the truth. In that case it is not possible to examine the error using quantitative statistics. Though animations can give an impression of how good the algorithm performs and if it is even better than the LOC for specific events.

The three resulting data sets are the following:

reliable cases:

$$DS_i^R = [LOC_1, LOC_2, \dots, LOC_{m_1}] \quad (2.3)$$

non-reliable cases:

$$DS_i^{NR} = [LOC_1, LOC_2, \dots, LOC_{m_2}] \quad (2.4)$$

full data set

$$DS_i = [LOC_1, LOC_2, \dots, LOC_m] \quad (2.5)$$

2.4 LOC processing

This section explains the process of generating a clean LOC. The structure of the process can be observed in figure 2.4.

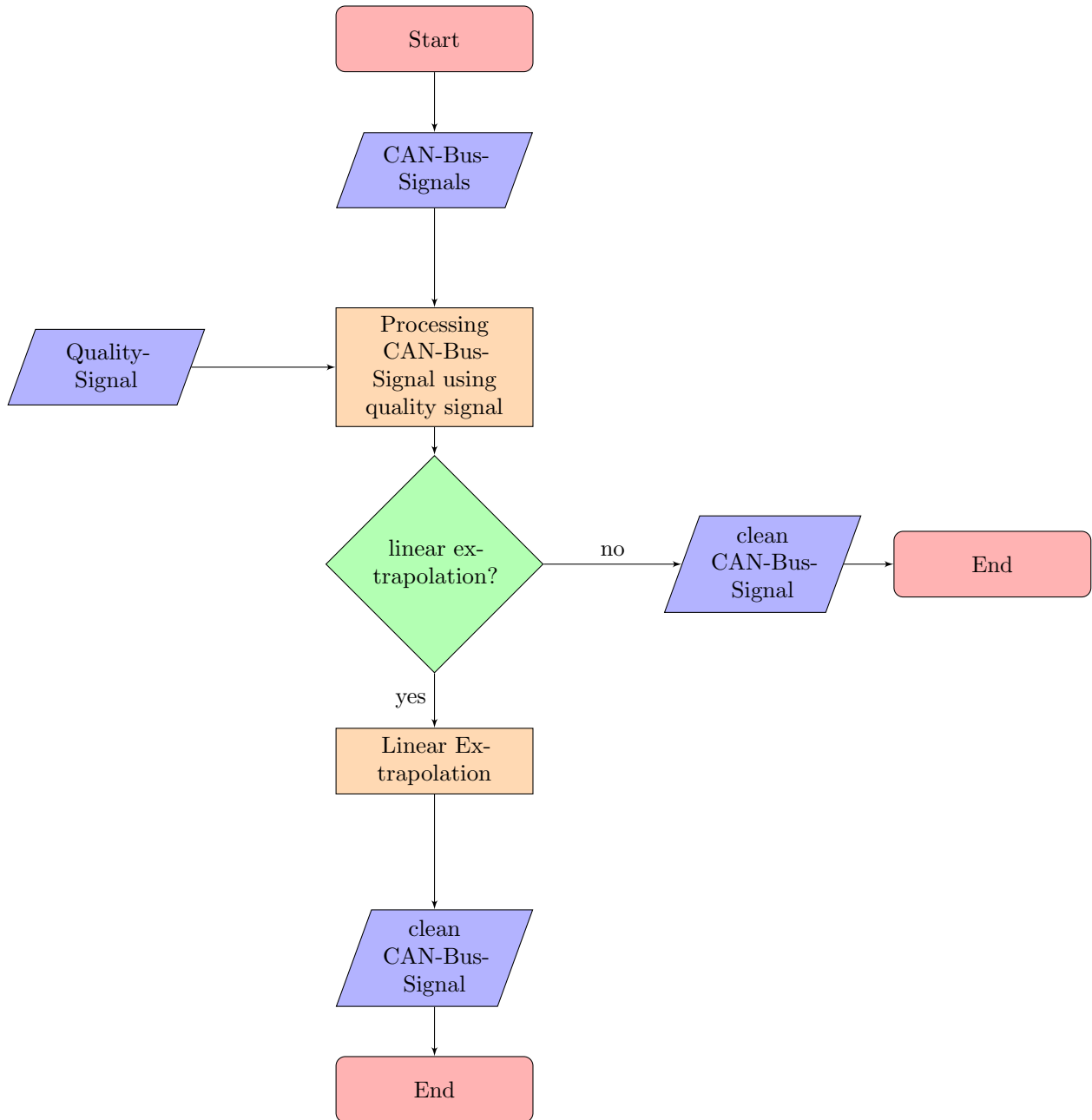


Figure 2.4: *Processing the LOC*

2.4.1 LOC filtering

The LOC derived in section 2.3 is defined as:

$$LOC_i = [loc_1, loc_2, \dots, loc_m] \quad (2.6)$$

The quality signal $LDC_i = [q_1, q_2, \dots, q_m]$, explained in section 2.3.2 is used to detect the stretches of the Lane-Offset Signal, which can be used for a reliable comparison.

The Lane-Detection Confidence Signal was used to only use the parts of the Lane-Offset-Signal, which are assessed medium high or high confidence assessment. Measurements with a lower confidence assessment were set as "Not a Number(NaN)" This assures the use of a reliable signal for further comparisons.

$$LOC_i^{NaN} = [lov_1, NaN, \dots, NaN, lov_m] \quad (2.7)$$

Figure 2.5 shows the original LOC. The remaining parts of the signal after cleaning it with the LCD can be observed in figure 2.6.

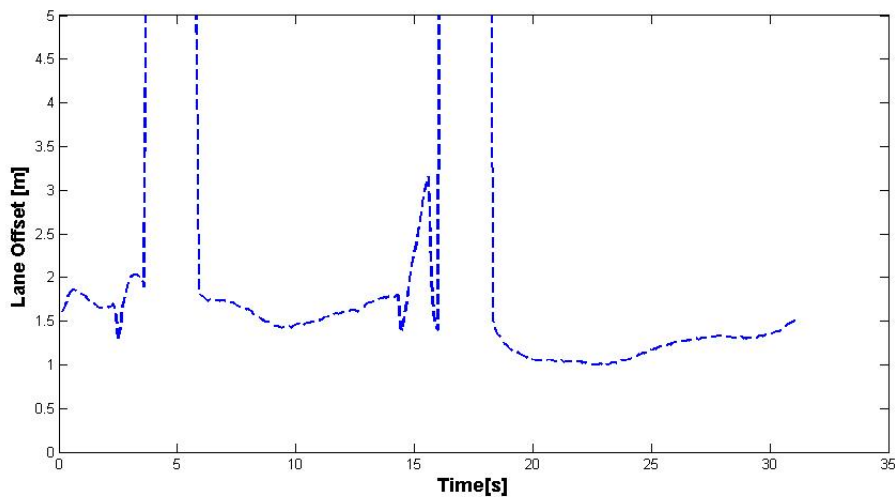


Figure 2.5: *Original Lane-Offset Signal*

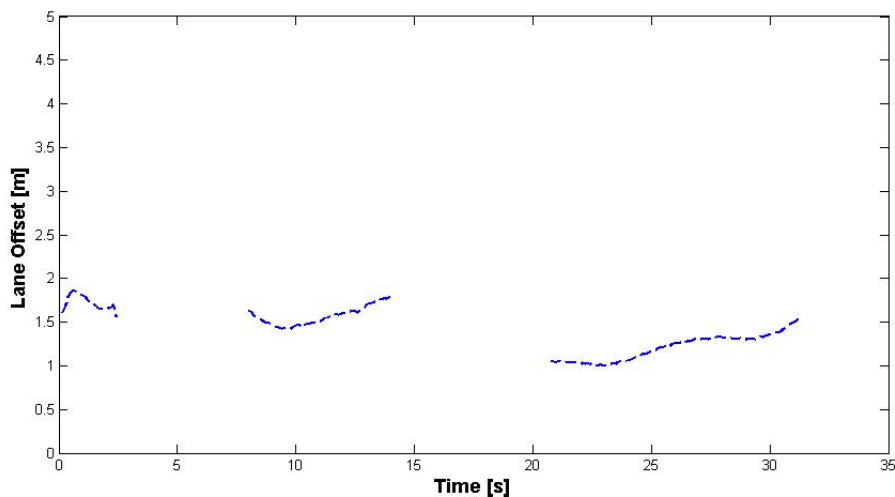


Figure 2.6: *Reliable stretches of the Lane-Offset-Signal*

2.4.2 Linear Inter- and Extrapolation

The user of the program can decide if a linear extrapolation should be performed or not. An extrapolation increases the number of available values for comparing the Lane-Offset-CAN-Signal to the Lane-Offset-Video

Signal. At the same time it creates the risk of comparing less reliable values in stretches of extrapolation or interpolation. Only for signal gaps smaller than 50m it is reasonable to extrapolate and fill the gaps within the signal. If there is a gap at the beginning of the signal the gap is filled by taking the first value after the gap and creating a horizontal line. For gaps at the end of a signal the last value before the beginning of the gap is the basis for the horizontal line. For all other gaps a linear interpolation is performed between the last value before the gap and the first value after the gap. An example can be seen in the following figure 2.7.

$$LOC_i^E = [lov_1, lov_2, \dots, lov_m] \quad (2.8)$$

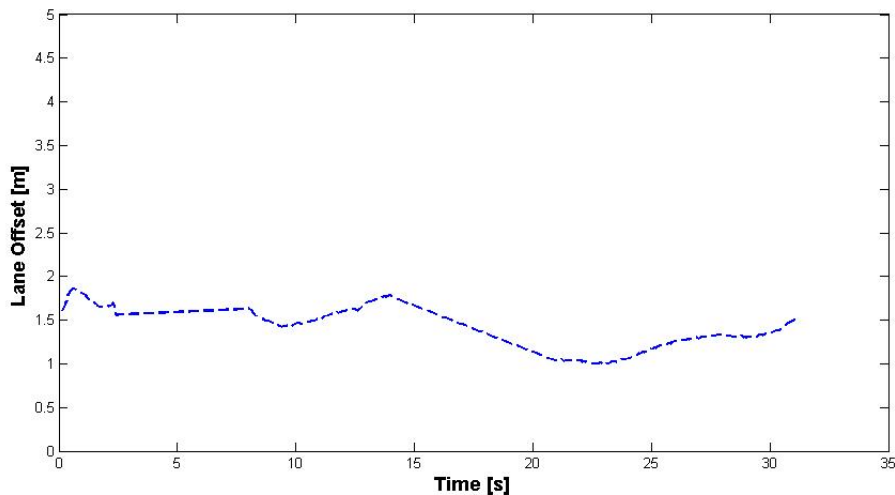


Figure 2.7: *Edited Lane-Offset-Signal*

2.5 Video processing

The videos receive a number of manipulations in this step, namely Pre-processing, Main process and Post-Process. Figure 2.8 shows the process need to extract the LOV from the video data. Before these modifications, some considerations have to be made. This is explained in section 2.5.1.

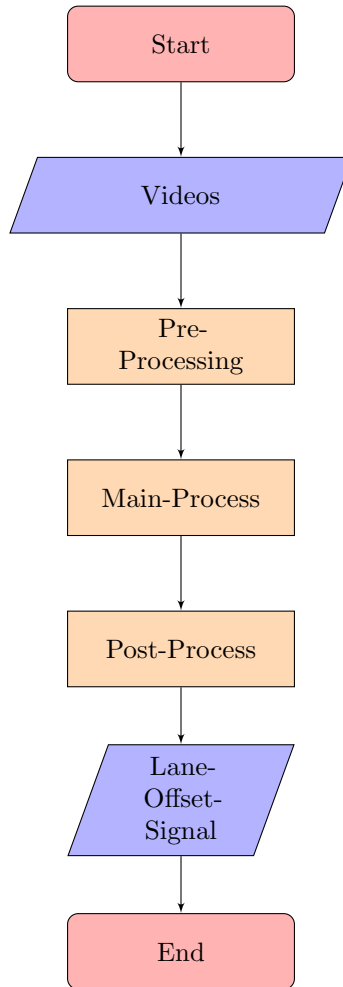


Figure 2.8: *Video processing*

2.5.1 Video data considerations

The characteristics of a video or picture are affected by the conditions, under which it was taken. These conditions involve general aspects as well as more technical ones. Since one of the aims in this work is to obtain measurements from the videos, these modifications in its appearance have to be addressed to increase the overall precision. There are three kinds of parameters that are needed to make a real world representation.

- Intrinsic parameters
- Distortion coefficients
- Extrinsic parameters

The first two are obtained using a method called "calibration", while the other ones have to be provided by the people who took the video/image. The combination of these parameters and the corresponding equations for a real world representation form the "Camera model". In the following section gives an explanation on how each of these parameters can be obtained and how they are used to obtain measurements from videos.

2.5.1.1 Calibration

The step called "Calibration" or 'Camera calibration' is the procedure to get the intrinsic parameters and the distortion coefficients in a camera configuration. The first step in a calibration is taking pictures of a chess board with different orientations and positions. The intention of using a chess board is, firstly, to have the

proper reference between distances in pixels and distances in world units (usually millimeters) and secondly, since the sub elements of a chess board are squares, it is possible to detect deformations in their shapes due to distortion.

If the camera is placed behind 'transparent' surfaces, the chess board has to be placed after this surface and the camera since this material could affect the light rays and cause a distorted appearance. In the present work, the windshield of the car causes this effect.

To cover the complete view range, the more calibration images, the better, however as a general rule of thumb 10-20 images is an acceptable number. In the figure 2.9 some examples of the calibration images are shown.

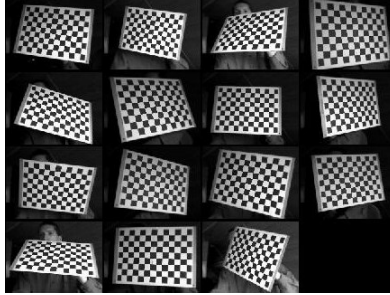


Figure 2.9: *Examples of calibration images*

The second step for the calibration is to use these images to extract the intrinsic and distortion parameters of the camera. The *Camera Calibrator App* of Matlab is the most recurrent option to perform this task but there are other open source apps and scripts that can be found on the web. In this work, the intrinsic and distortion parameters were provided by Volvo Cars as it was explained in section 2.1.2.1 and the open source *camera calibration toolbox* [Bouguet 2017] was used only to verify the correspondence of the values.

2.5.1.2 Camera Model

2.5.1.2.1 Intrinsic Parameters

The intrinsic parameters get their name from the fact that are fully dependant on the internal characteristics of the camera and they can not be changed. The main ones are the focal length and the principal point.

The focal length can be understood as the perpendicular distance between the camera sensor ¹ and the image plane. On the other hand, the principal point corresponds to the vertical and horizontal position of the camera sensor. Manufacturers try to place it as close to the center as possible, but it usually has a small offset.

In general, the units of both the principal point and the focal length are always given in pixels. Although it is possible to get the ideal values by computing the formulas shown in table 2.9, it is recommended to use a calibration tool since they might have a slight variation from these ideal computations.

Table 2.9: Description & computation of intrinsic parameters.

Parameter	Computation	Description
U	-	Image Vertical resolution [pixels]
V	-	Image Horizontal resolution [pixels]
F	-	Camera Focal length [mm]
f_x	V/F	Horizontal Focal length [pixels/mm]
f_y	U/F	Vertical Focal length [pixels/mm]
C_x	$V/2$	Horizontal Principal point [pixels]
C_y	$U/2$	Vertical Principal point [pixels]

¹In the *pinhole camera* or *camera oscura*, the pinhole managed the capture of the image by letting the light go inside the dark room. In the current cameras, the camera sensor performs the same task as the pinhole but with higher control and resolution.

These parameters are generally shown in a matrix form as its shown in equation 2.9. Some of these parameters can be observed in the figure 2.10.

$$K = \begin{bmatrix} fx & 0 & Cx \\ 0 & fy & Cy \\ 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

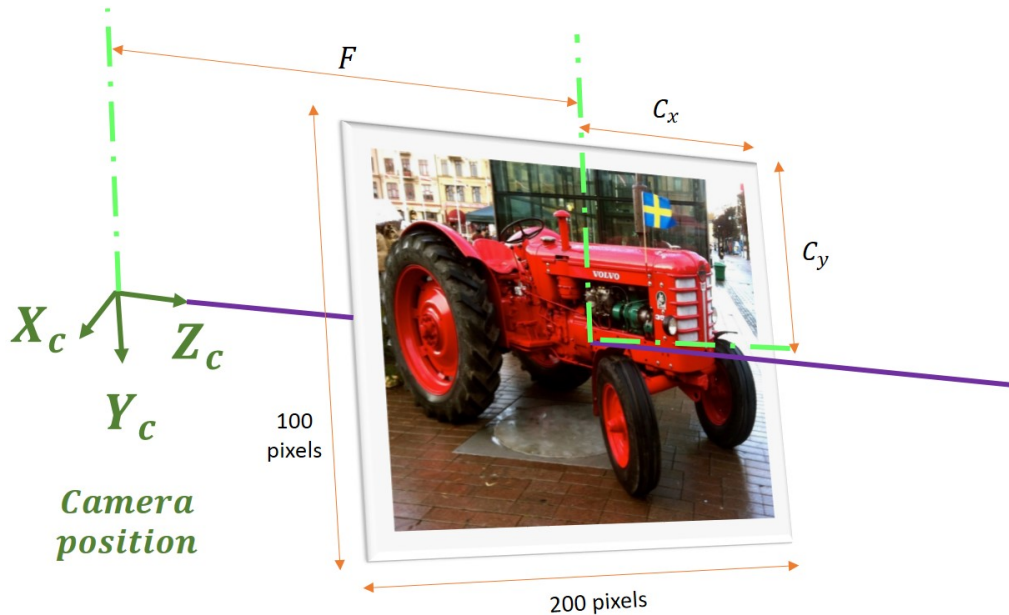


Figure 2.10: Visualization of the camera and image coordinate system.

2.5.1.2.2 Extrinsic Parameters.

The extrinsic parameters are the parameters than depend on the external orientation of the camera, this is, its rotation R and relative position to the ground or its position relative to another reference T . These parameters cannot be obtained from the calibration images and the only way to know them is by asking the people who took the video/image. In this work, these values were given by Volvo Cars as it was explained in the section 2.1.2.1

Intrinsic and extrinsic parameters are usually placed in a matrix form. This matrix, sometimes called camera matrix, performs a mapping between a real world coordinate system and an image coordinate system. The camera matrix can have different nomenclatures and ways to be built, in this work, it is referenced as P and its structure can be seen in the equation 2.10.

$$P = K [R | -R * T] \quad (2.10)$$

2.5.1.2.3 Distortion coefficients.

Distortion is not directly related to the coordinate system transformations, however it has to be addressed before performing the transformations to correct the deformations of the frame. There are two kinds of distortions that can be present in an image. Radial distortion and tangential distortion.

Radial distortion

Radial distortion is presented when lines that are supposed to be straight (either horizontal or vertical) are shown bent when they are closer to the edges. There are two types of radial distortion, positive and negative. The differences in their appearance can be observed in figure 2.11.

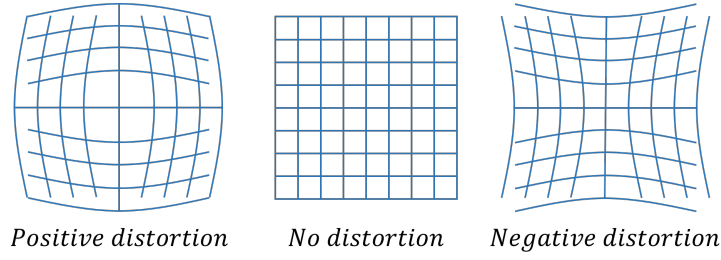


Figure 2.11: *Radial distortion types.*

The process used to undistort an image is by solving the equations 2.11, 2.12, 2.13 for x and y and place the pixel value that was located in the distorted position (x_d, y_d) in its new position (x_u, y_u)

$$x_d = x_u(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (2.11)$$

$$y_d = y_u(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (2.12)$$

$$r^2 = x_u^2 + y_u^2 \quad (2.13)$$

Tangential distortion.

The tangential distortion is presented when there is an angle between the vertical plane and the camera sensor as it is shown in figure 2.12.

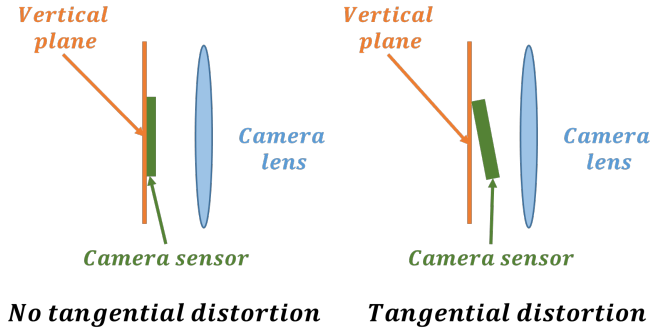


Figure 2.12: *Tangential distortion types.*

The procedure to remove tangential distortion is similar to the radial distortion but with additional terms in the equations 2.11 and 2.12. Leaving the extended versions of the distortion equations as shown in equations 2.14 and 2.15.

$$x_d = x_u(1 + k_1r^2 + k_2r^4 + k_3r^6) + 2p_1x_uy_u + p_2(r^2 + 2x_u^2) \quad (2.14)$$

$$y_d = y_u(1 + k_1r^2 + k_2r^4 + k_3r^6) + 2p_2x_uy_u + p_1(r^2 + 2y_u^2) \quad (2.15)$$

2.5.1.3 Real world representation.

Once all the previous parameters are known, it is possible to derive a measurement in scientific units (e.g. millimetres, meters, feet, inches, etc.). In order to perform this step, it is needed an equation that relates and transforms camera measurements into scientific measurements. In this work these measurements will be called

camera or pixel coordinates and real world coordinates, respectively. The equation that will be used to perform these transformation will be called camera equation and its structure is shown in the equation 2.16

$$\lambda w = PW \tag{2.16}$$

Due to the relationship between the pixel coordinates and the Real World ones not being linear, the easiest way to solve the system is to place the sub elements of w and W in homogeneous coordinates. In consequence, the vector of image coordinates will be defined as $w = (x, y, 1)$ and the vector for the global coordinates will be $W = (X_W, Y_W, Z_W, 1)$. The variable P is the previously mentioned, camera matrix (3-by-4) and since there is a depth (distance from object to the camera plane), its value has to satisfy the relation $\lambda > 0$.

Combining the steps of the previous subsections, the final equation that relates pixel coordinates to Real world coordinates results as follows

$$\lambda \begin{bmatrix} x_u \\ y_u \\ 1 \end{bmatrix} = K [R | -RC] \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \tag{2.17}$$

In figure 2.13 it is possible to observe all the coordinate system changes that are used to get the value of a pixel position in a real world reference.

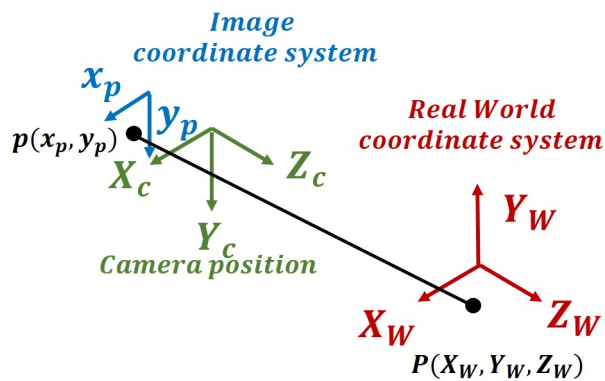


Figure 2.13: *Coordinate systems.*

2.5.2 Video analysis

The video analysis section consists of three subsections, the pre-processing step, the main process and the output signal processing step.

2.5.2.1 Pre-processing

The pre-processing steps are subroutines which their main objective is to carryout modifications to the provided signals/data. Although their presence is not crucial, they increase the speed, accuracy and general performance of the main process. In this work, there are two pre-processing steps. The first one is to remove the distortion of the video and the second one is to remove the irrelevant areas in the view.

2.5.2.1.1 Undistortion In this step the equations 2.14 and 2.15 are solved for every frame of the video considering the intrinsic parameters and the distortion coefficients. To avoid the solving step, a small trick is performed using the pixel coordinates. First, we distort the positions of an empty matrix which has the same size of the image and then we do a backward mapping of the coordinates. Finally we place the pixel value in its new undistorted position.

2.5.2.1.2 Removal of Regions of no interest

In some of the videos, a section of the cluster and the bonnet can be seen in the take. Because of the light reflection in their surfaces, their presence can affect the performance of further processes, for that reason, it is necessary to remove them from the take. The way to do this is to make a comparison of all the frames that are part of the video. The pixels that do not present a significant change in their pixel value are set to zero. To wrap these new empty pixels in a smooth and curved shape, a second order regression is performed using the borders of these empty pixels.

2.5.2.2 Main process

The input of this process will be the cleaned frames of the video. All the steps inside the main process are performed in every frame, ignoring the information of the previous or future frame. The intention of doing this is to make the algorithm independent of the frame rate of the video (the sample rate is different for every data base). The central steps perform the manipulations of the images to detect the lane markings and the calculations to obtain the vehicle position in meters. The output of this step will be two numeric arrays (one for the left and one for right side) with the lateral position of the vehicle in every frame.

2.5.2.2.1 Establish Regions of Interest (ROI's)

In this step it is assumed that all the important information will appear in the lowest part of the image. The sky, the tree tops, electric cables, etc. are irrelevant and have to be removed or ignored. Only a share of the image has to be analyzed and the rest is set to zero. To obtain the value that perfectly divides the ground from the sky is a procedure that might involve the segmentation of the image, or other time consuming algorithms. For simplicity, this value is fixed to 50%. In the figure 2.14 it can be seen the effects of this step.

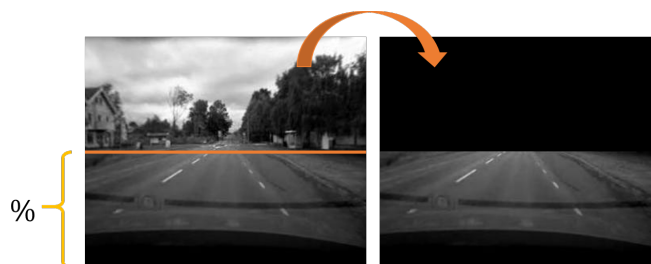


Figure 2.14: *Removal of regions of no interest*

2.5.2.2.2 Binarization

The method that is used to detect lines is Hough transform [Fokkinga 2011]. This function requires a binary image as an input. The 'true' pixels will correspond to the white areas of the lane markings. Convert an RGB or Grayscale image into a binary image can be carried out using in-built functions of Matlab. However, these in-built functions use a constant threshold, no matter the light conditions. Since this is a critical step for the lane marking detection, this step has to be light invariant. More over, this *binarization* has to be done in such way that the pixels that belong to a lane marking, do not be removed and the pixels that do not belong to a lane marking completely disappear from the view.

On the other hand, in some roads there are no lane markings, but there is a clear road which can be perceived by the driver and by the observer. This method has to be able reproduce this cognitive process of detecting the road borders.

One mask ² is the result of highlight the white zones in the image (lane markings), and the other mask is the result of highlight the edges with the shape of a line.

²A mask in computer vision is a binary image

Mask for white zones.

The process to get this mask consists of 4 steps. First, filter the input image with a large horizontal median filter. Second, perform a numerical subtraction between the input image and the filtered image. Then, threshold the image with a double criteria, this means, if a pixel value is greater than the threshold t_1 but lower than the threshold t_2 make the pixel white (or true). Finally, clean the image using different morphological operations.

Mask for road borders.

The mask that preserves the borders of the image is obtained through the logical operation 'and' of two submasks, each submask is obtained through the analysis of the left and right side of the image as it is shown in the equation 2.18. The reason why there is a left and right analysis is because the background of the image is normally different in each side of the view (e.g. bushes in the right side and cars in the left side) and by considering the two thresholds the result is less susceptible to the background conditions.

$$M_{borders} = M_L \cap M_R \quad (2.18)$$

The input image of the two submasks is processed using a filter with the values shown in equation 2.19

$$h = [-1 \quad -1 \quad 0 \quad 1 \quad 1] \quad (2.19)$$

This filter is used to highlight the relevant information from the right side and its flipped version for the left side. The output of this step will produce two different images: I_R for right and I_L for left.

$$I_R = I \otimes h \quad (2.20)$$

$$I_L = I \otimes flip(h) \quad (2.21)$$

The result of processing I_R and I_L will be the left and right binary masks M_R and M_L .

$$M_R = g(I_R) \quad (2.22)$$

$$M_L = g(I_L) \quad (2.23)$$

The function $g(I_X)$ ³ will be performed as follows. The first step will be to divide the image I_X in left and right side producing two sub images I_{XL} and I_{XR} .

With each sub image the optimal threshold [Toran Marti 2010] is computed. Then, the image I_X will be binaryzed with the two optimal threshold values (t_{XL} and t_{XR}). Finally the two images are merged using the 'and' operator as it was shown in equation 2.18.

2.5.2.2.3 Lines detection

The method that it is used to detect the lane markings is the Hough transform method [Fokkinga 2011]. It is used in-built function of the image processing toolbox of Matlab. The input variables that the Hough transform function receives are: a binary image and the angles of the lines that we are looking for. In this case, we are not interested in neither vertical lines (result of poles or fences) nor horizontal lines. There are different methods to get the best angle range for detecting lanes [Satzoda, Suchitra, and Srikanthan 2012; Lopez et al. 2010], however these methods involve a high level of control in the way the videos are taken. In this work since two different sources or videos will be analyzed (EOR and LYTX), a generic constant range of angles was used. Then, the angle range for the right side is set to $-80 > \phi_R > -20$ and for the left side $80 > \phi_L > 20$. An additional parameter can be provided in the Hough transform function, the amount of lines that best fit with the angle criteria. In this work a value of three lines per side is chosen. In case more than one line matched with this criteria, the median of the detected lines will be the final line of the lane marking.

³The sub index X indicates that it is considered any side.

2.5.2.2.4 Conversion to Real World measurements

As it was explained in the section 2.5.1.3, once the camera parameters and the values to convert are known, it is possible to convert a point (in pixels) into a measurement (in meters). The steps to get the lane marking position in real world measurements are as follows:

1. Get the two equation lines (left and right) in pixels.
2. Pick two points from each line equation.
3. Transform the pixel coordinates of these points into real world coordinates.
4. Get the equation lines of the real world coordinates.
5. Evaluate the equation lines at the correct longitudinal offset⁴.

The system coordinates that are used to the coordinate transformations can be seen in the figure 2.15

The longitudinal offset of the camera is known for eyes-on-road but unknown for LYTX, for simplicity this offset was set to 1m.

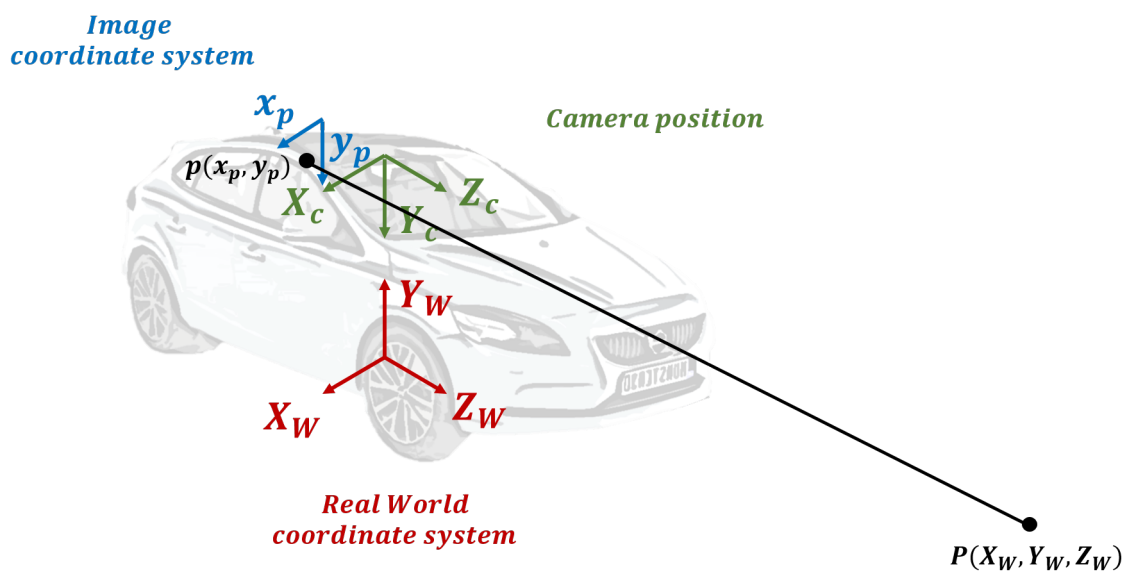


Figure 2.15: Vehicle Coordinate systems.

The lane marking positions of each frame are stacked in two arrays of signals:

- LOV_i^R : Lane Of Video for the Right side in the frame i .
- LOV_i^L : Lane of Video for the Left side in the frame i .

⁴The camera is placed behind the tires and the signals for lane markings are always referenced at at the axis of the frontal tires

2.5.3 LOV processing

2.5.3.0.1 Setting thresholds for detecting unrealistic values.

From the video analysis we receive the raw LOV Signal:

$$LOV_i^R = [lov_1, lov_2, \dots, lov_m] \quad (2.24)$$

The first step, which was used to clean the signal, was setting thresholds to neglect unrealistic values.

Thresholds:

1. $0 < lov_j < 4$
2. $lov_{j+1} - lov_j < 0.1$

If a value within the vector exceeds the certain threshold, it will be replaced by "Not a number(NAN)"

$$LOV_i^{NaN} = [lov_1, NaN, \dots, NaN, lov_m] \quad (2.25)$$

2.5.3.0.2 Threshold for NaN substitution

The stretches of NaNs have to be further analyzed. For the LOV-Signal it is not possible to relate a stretch of NaNs to the speed of the vehicle and therefore the distance travelled by the vehicle. The threshold chosen is twenty NaNs in a row is the maximum stretch, which will be substituted through interpolation, which translates to a signal gap of two seconds.

2.5.3.0.3 Interpolation and Extrapolation

Interpolation and Extrapolation is performed similar to section 2.4.2. The resulting signal is defined as:

$$LOV_i^E = [lov_1, lov_2, \dots, lov_m] \quad (2.26)$$

2.6 Evaluations

This method is evaluated in two different sections. Qualitative and Quantitative evaluations. The Quantitative consists of the numerical analysis of two properties: The first is the precision and accuracy. The second is the sensitivity of the method when the provided camera parameters change. The Qualitative evaluation shows an animation, where both signals, LOC and LOV are fed into the corresponding video.

2.6.1 Quantitative evaluations

The quantitative evaluation will cover the assessment of the error between LOV and LOC and the sensitivity analysis.

2.6.1.1 Precision and Accuracy

Figure 2.16 shows the process of evaluating the Error in different categories.

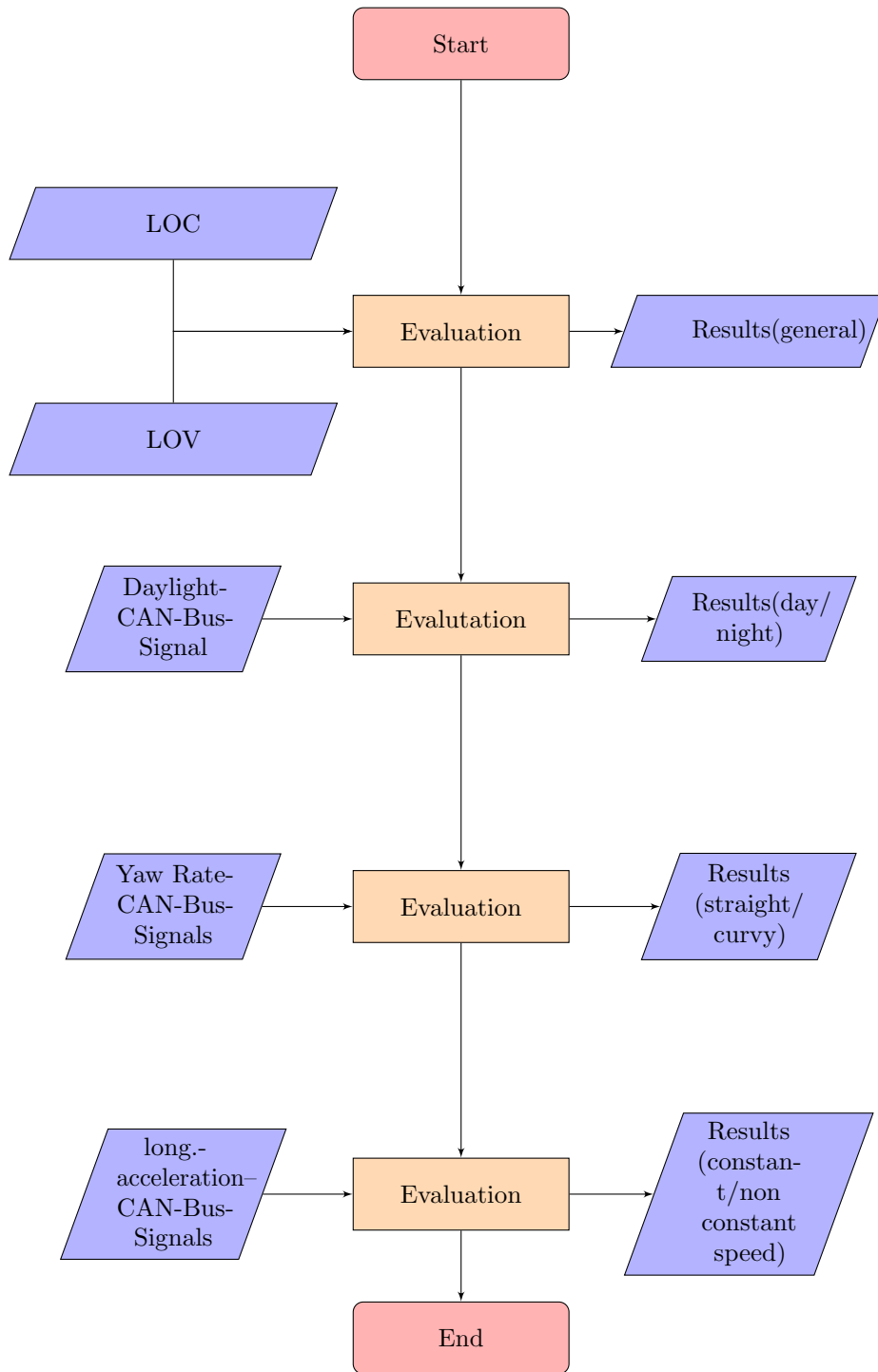


Figure 2.16: *Quantitative Evaluation*

2.6.1.1.1 Estimating the error for each event.

For every event i , the clean Lane-Offset Signal from the CAN-Bus (LOC_i^E) and the clean Lane-Offset-Signal from the Video (LOV_i^E), which are generated in section 2.3 and 2.5, are used. The error between these two signals is calculated:

$$E_i = LOV_i^E - LOC_i^E \quad (2.27)$$

$$E_i = (e_1, e_2, \dots, e_m) \quad (2.28)$$

In the next step the arithmetic mean, the variance and the standard deviation for each event is derived. The harmonic mean is used to counteract outliers within the error vector.

Average:

$$\bar{e} = \frac{1}{m} \sum_{a=1}^m e_i \quad (2.29)$$

mean squared deviation:

$$(\Delta e)^2 = \frac{1}{m} \sum_{i=1}^m (e_i - \bar{e})^2 \quad (2.30)$$

root mean squared deviation:

$$\Delta e = \sqrt{(\Delta e)^2} \quad (2.31)$$

Estimating the mean and the variance

The best estimation for the mean is the average of the data itself:

$$\mu = \bar{e} \quad (2.32)$$

The estimation for the variance is slightly larger than the mean squared deviation:

$$\sigma^2 = \frac{m}{m-1} (\Delta e)^2 \quad (2.33)$$

The standard deviation is estimated using the square root of estimated variance:

$$\sigma = \sqrt{\sigma^2} \quad (2.34)$$

2.6.1.1.2 Estimating the error for the whole data set.

Estimating the mean error of the data set by estimating the weighted average. The weight is assigned according to the number of measurements in each data set.

$$\bar{E} = \frac{\sum_{i=1}^m w_i \mu_i}{\sum_{i=1}^m w_i} \quad (2.35)$$

Estimating the variance of the data set using the weighted estimation.

$$\Lambda = \frac{\sum_{i=1}^m (n_i - 1)\sigma^2}{\sum_{i=1}^m (n_i - 1)} \quad (2.36)$$

The standard deviation is again estimated by taking the square root.

$$\Lambda = \sqrt{\Lambda^2} \quad (2.37)$$

2.6.1.1.3 Estimating the error for different light, road and driving conditions

To get a better understand of how the algorithm is influenced by different light, road and driving conditions, to additional evaluations are performed. In the first on day and night drives are treated separately. The second evaluation takes a look at how the algorithm performs on straight roads in comparison to curvy roads. The final category distinguishes between constant speed and acceleration stretches.

Day drives vs night drives

To separate between day and night drives, the signal Daylight is taken into consideration. Through using the Daylight signal we can split the group of Events.

$$E_i = (ED_a, EN_b) \quad (2.38)$$

Events during day drives: ED_i

Events during night drives: EN_i

Curvy roads vs straight roads

To separate between different road conditions, yaw rate is used as an indicator for the shape of the road. For a yaw rate larger than 5degrees/second. a road is assumed to be curvy. For the road shape it is not possible to split the group of events, because a significant amount of events do have curvy and straight road parts. Therefore the Events themselves are split in to curvy and straight road parts and are added up to Event only including the curvy road parts and a second Event including the straight road parts.

$$EC = (E_{c,1}, E_{c,2}, \dots, E_{c,n}) \quad (2.39)$$

$$ES = (E_{s,1}, E_{s,2}, \dots, E_{s,n}) \quad (2.40)$$

Curvy road parts of an Event: $E_{c,i}$

Straight road parts of an Event: $E_{s,i}$

Event combining all the curvy road parts: EC

Event combining all the straight road parts: ES

Constant speed vs non constant speed

To separate between stretches of constant speed and non constant speed, longitudinal acceleration is used as an indicator for the movement of vehicle. For an longitudinal acceleration larger than 0.1meters/second. The generation of the two separate signals is done similarly to curvy roads vs straight roads.

$$EA = (E_{a,1}, E_{a,2}, \dots, E_{a,n}) \quad (2.41)$$

$$EZ = (E_{z,1}, E_{z,2}, \dots, E_{z,n}) \quad (2.42)$$

Acceleration stretches of an Event: $E_{a,i}$

Constant speed stretches of an Event: $E_{z,i}$

Event combining all the acceleration stretches: EA

Event combining all the constant speed stretches: EZ

2.6.1.2 Sensitivity Evaluation

It is needed to know how a small variation in the position and orientation of the camera change the final result.

The formula 2.44 calculates the relative change. The second evaluation is similar to the first one but in this case, the change is quantified in meters. By doing this, it is possible to notice how the signal increases (or decreases) when the parameters change.

In the two formulas 2.43 2.44, $LOV_{original}$ refers to the LOV using the parameters given by Volvo Cars Corporation. The $LOV_{parameter}$ represents the LOV when only one of the five parameters is changed (Height, lateral offset, yaw, pitch, roll) and the rest keep the value given by Volvo Cars.

$$\epsilon_{percent} = \frac{LOV_{original} - LOV_{parameter}}{LOV_{original}} \times 100\% \quad (2.43)$$

$$\epsilon_{meters} = LOV_{original} - LOV_{parameter} \quad (2.44)$$

These sensitivity evaluations are computed using 90 videos of the 818 available events. In order to avoid a biased estimation, these 90 events are chosen randomly.

2.6.2 Qualitative evaluation

In this work, the CAN-bus data is used as ground truth signal, however the source of these signal has a certain level of confidence. The intention of this section is to observe that in some circumstances the signal of the developed method is better than the ground truth signal. Since there is no variable that can give an estimate of the performance comparison, this evaluation is only qualitative. The way it is done is by showing the two signals (LOV and LOC) in the frontal and top view. By doing this it is possible to observe that in some circumstances the developed method detects a lane marking when the CAN signal could not.

The general structure to get the top and front view is as follows. For the front view (which is in pixels, the real world measurements like the LOC and LOV have to be transformed to pixel coordinates and then merge them for their visualization. For the top view, the images of the front view (in pixels) have to be shown in meters and be transformed into a top view perspective, then this new images are merged with the LOC and LOV signals for their visualization. The figure 2.17 shows the flow chart to get these plots.

In the following sections it is explained the details of the required steps to show the signals in the two perspectives, front and top view as well as the transformation of the signals to match the coordinate systems.

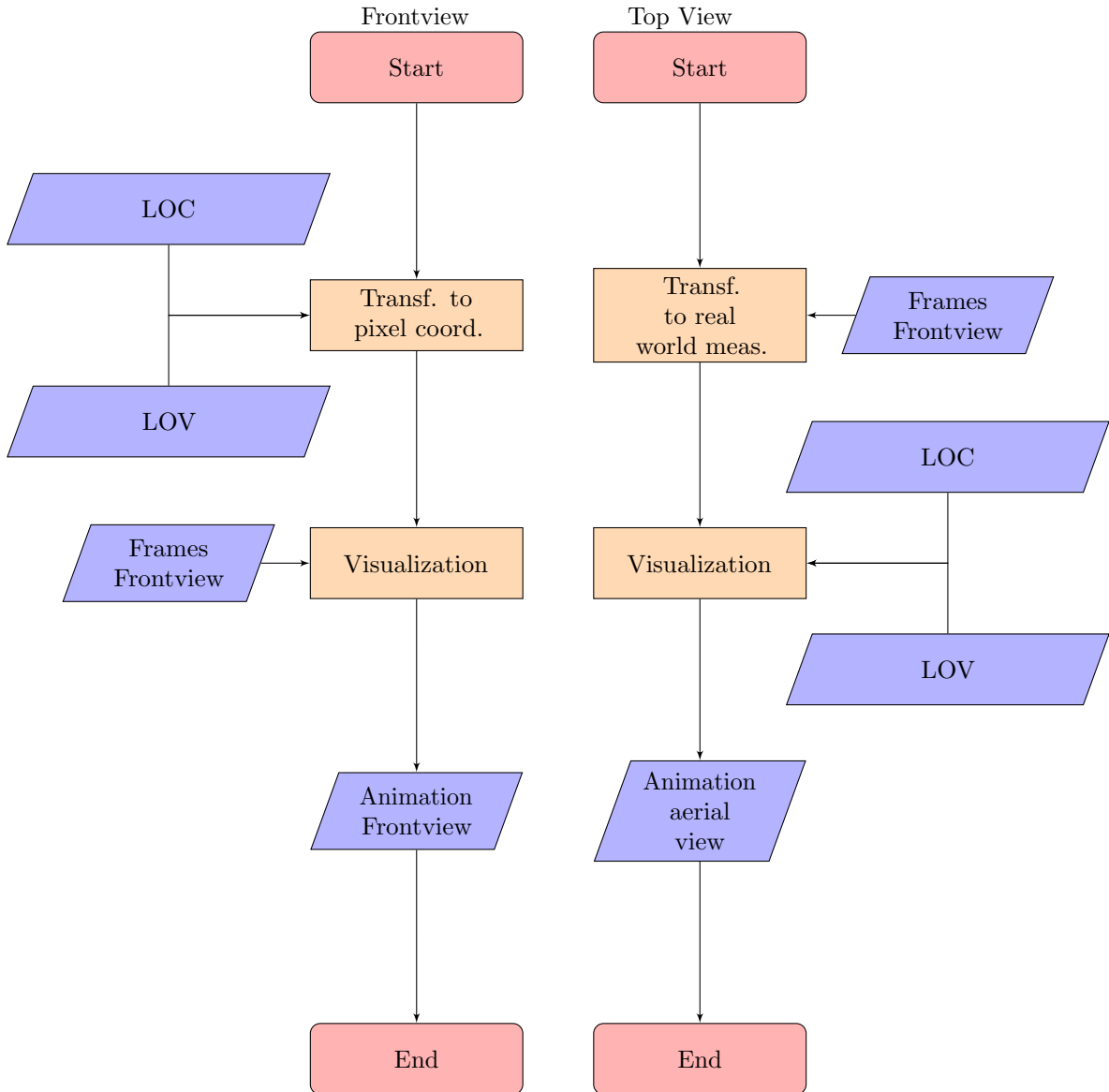


Figure 2.17: *Visual Evaluation*

2.6.2.1 Front view

The perspective of the frontal camera is already part of the provided data, however, in order to observe the difference in the performance of the CAN signals relative to the video signals, a transformation to pixel coordinates is needed.

Although the video signals come from pixel coordinates, a backward computation from real world coordinates to pixel coordinates is needed since the post processing steps modified its values and using the the primitive versions would increase the error in the visual evaluation.

The transformation to convert the signals is carried out using the equation 2.17 but now solving it for x and y . The value of for $Y_W = 0$ for the lane markings located at ground level and $Z_W = 1$ for the signals obtained in the same axis as the frontal axle of the vehicle. The remaining matrices are the same as the one used for the

transformation to real world coordinates. The system coordinates are the same used as in figure 2.13.

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = K [R] - RC \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \quad (2.45)$$

The signals from the video processing step are plotted in magenta and the ones from the CAN bus are in green as it can be observed in figure 2.18.

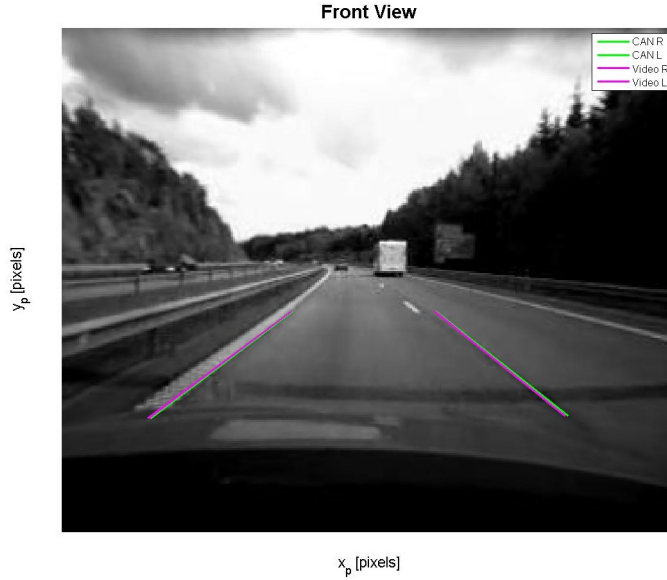


Figure 2.18: Comparison of signals in the frontal view.

2.6.2.2 Top view

The intention of this view is to observe what is detected by the Video and by the CAN, signals. In this view both the images and the signals have to be in meters. The *LOC* and *LOV* signals are already in meters, thus no more manipulations are needed. However, for showing the video in a top view perspective, a transformation of every frame is needed. In computer vision, this step is called homography.

2.6.2.2.1 Homography

This step consists of carrying out the transformations to real world measurements of every pixel location of the take. The aim of this step is the same as the process to convert the lane offset in pixels location into meters. In this case, due to having more than one pair of pixel coordinates but a complete region of interest, a modification to the equations is needed to use matrices instead of one pair of coordinates.

As it was observed in figure 2.14, the relevant information that can be extracted from every frame, is located in the lowest part of the take. For that reason, the transformations are carried out only using the 50% of the image. In figure 2.19 it can be observed, that the range the EOR camera has using only 50% of the frame.

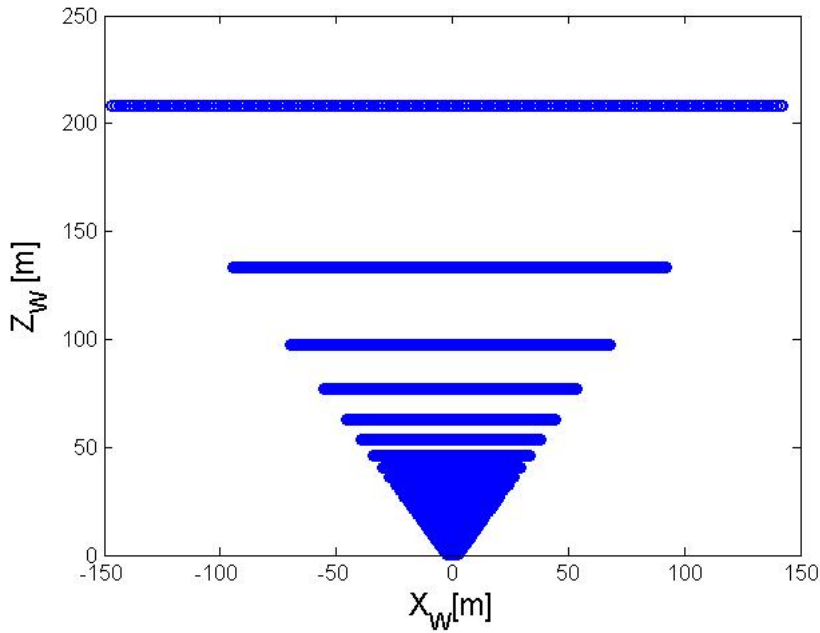


Figure 2.19: *Camera Range in real world units*

As it can be observed, the longitudinal range of the camera can go up to 200 m and lateral range from -150 to 150 m., however, the intention of this visualization is make a comparison of signals, thus, a lateral range of -7.5 to 7.5 m and a longitudinal range of 0 to 25 m. is more than enough.

2.6.2.2.2 Pixel interpolation

When a snapshot to an outdoors landscape is taken, a vanishing point appears. This occurs when infinite and straight lines like a roads, converge in the horizon. When we try to reconstruct the infinite roads from the front view perspective, the discretization of the sampled scene (every pixel is a sample) causes that only some points of the scene can be plotted. As it can be observed in figure 2.21.



Figure 2.20: *Front view*

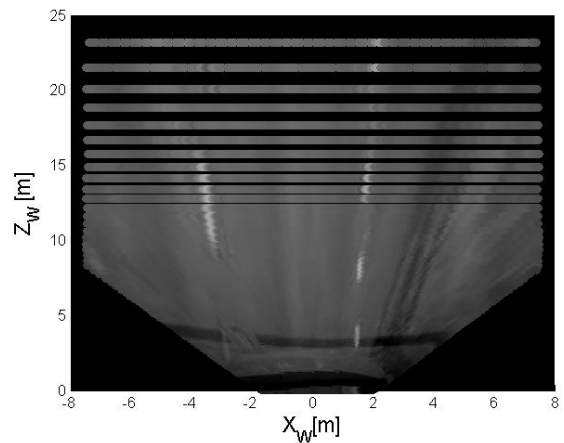


Figure 2.21: *Top view*

In order to fill the empty spaces of the top view, a double interpolation is needed using the position and pixel values of the non-empty locations.

Once the transformation of the frontal view into top view is done, since now both the frames and the signals are in meters, it is possible to fuse the video and signals into one plot. In figure 2.22 this fusion can be seen.

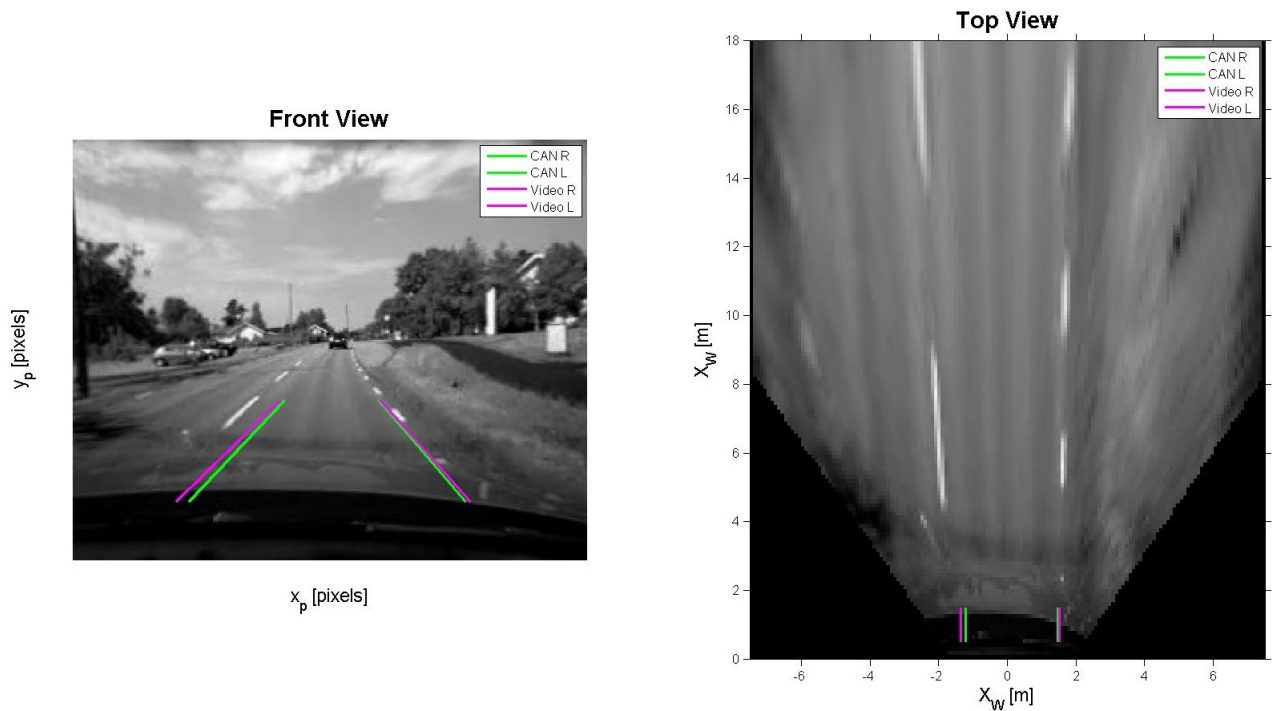


Figure 2.22: Visualization of the front & top view. Top view already with the interpolation

2.6.3 GUI

In order to easily handle and work around with the whole set of videos, a GUI is developed to select different criteria required for the evaluation.

The first step of using the GUI 2.23 is to set the percentage of reliable values that each video has to have using slider 1 (see table 2.11). Depending on the application the user can choose through button group 2 to use reliable values, non reliable values or the whole data set. In the following control, the user can choose to use LYTX settings, which down samples the video and reduces the resolution to match the LYTX sampling conditions. Moreover, the user can show animations of the front and/or the top view using the two additional check boxes(see figure 3.21 .

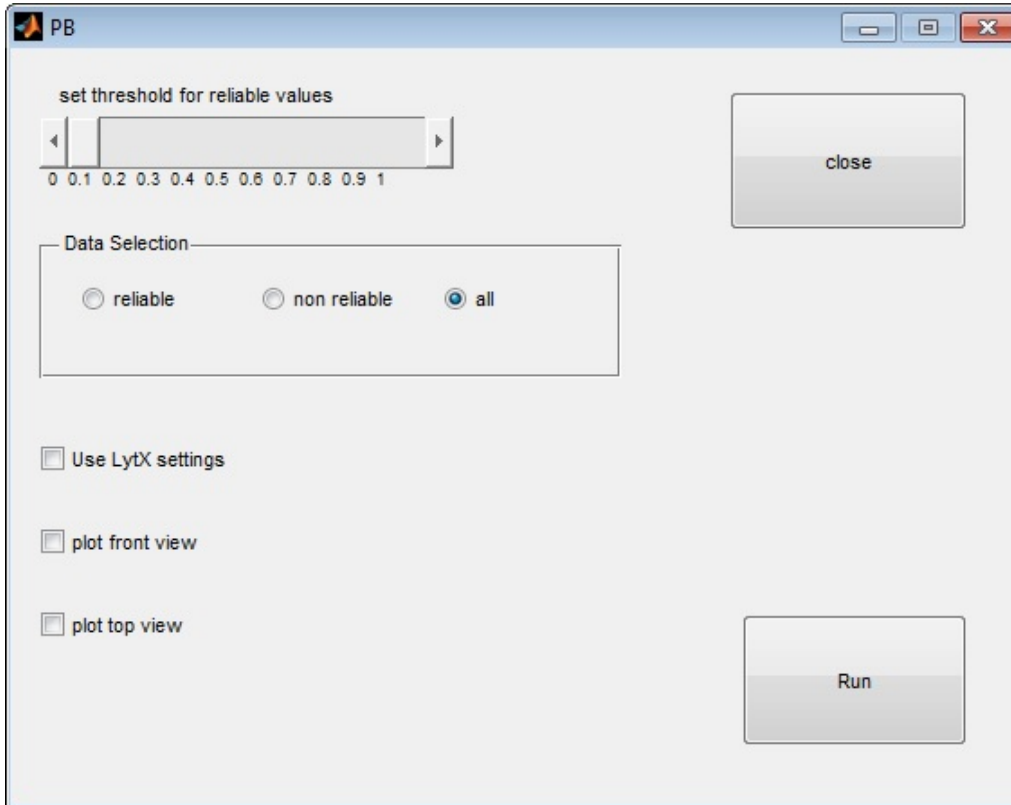


Figure 2.23: GUI for comparing the CAN-Bus Signal to the Camera Signal

Table 2.10: Explanation of GUI birdvie

Number	Description	Explanation
1	Set threshold	By using the slider the user can set a threshold of how much a video has to consist out reliable values.
2	Data Selection	The user can decide, which subset of the videos are going to be processed. Reliable videos are the once, which are over the threshold set by using slider one. Non reliable videos are the once, which are below the threshold. All videos contains the whole set of videos.
3	Use LytX settings	By ticking the check box the user can decide to use LytX settings
4	Plot front view	By ticking the check box the user can decide to show the front view
5	Plot top view	By ticking the check box the user can decide to show the top view
6	Run	By using the button the user starts the computation with the previously defined settings

2.7 Versatility

At this point, this method uses only EOR videos. However, one of the aims is to apply it to different data bases, specifically LYTX. For that reason some modifications have to be done to this method in order to fulfill this requirement. The intrinsic and extrinsic camera parameters are crucial for this method to work. As it has been mentioned in previous sections, or EOR videos these values are provided by Volvo Cars Corporation, however, for LYTX videos the only known values are the intrinsic camera parameters (see section 2.1.2.2).

The position and orientation of the camera are unknown and different even between videos of the same data base. In order to get these parameters, it is needed to develop an easy way to find them. A Graphical User Interface (GUI) is the best solution to do so. In the section 2.7.1 the appearance and functionality of this GUI are explained.

2.7.1 GUI for LYTX Videos

This GUI has two figure interfaces. One is to find the position of the camera that fits the selected video. The second interface is to run the method and give the possibility to the user to show and save the video with the results of the method (*LOV*). The following sections explain two interfaces in detail.

2.7.1.1 Interface 1

The purpose of the interface shown in figure 2.24 is to help the user to define the extrinsic camera parameters. Using button 1 the user loads a video file and displays a random frame. Furthermore a lane offset is placed within the middle of the figure, which is used as a reference. Using the sliders 2-7 the user can change the extrinsic camera parameters and see how the display of the frame changes. The purpose of slider 8 is to give the user the ability to switch between 20 different frames. Button 10 saves the chosen parameters to the current folder. Using button 9 the user can reset all camera parameters to its default values. Pressing button 11 opens a new GUI to process the video with the chosen camera parameters.

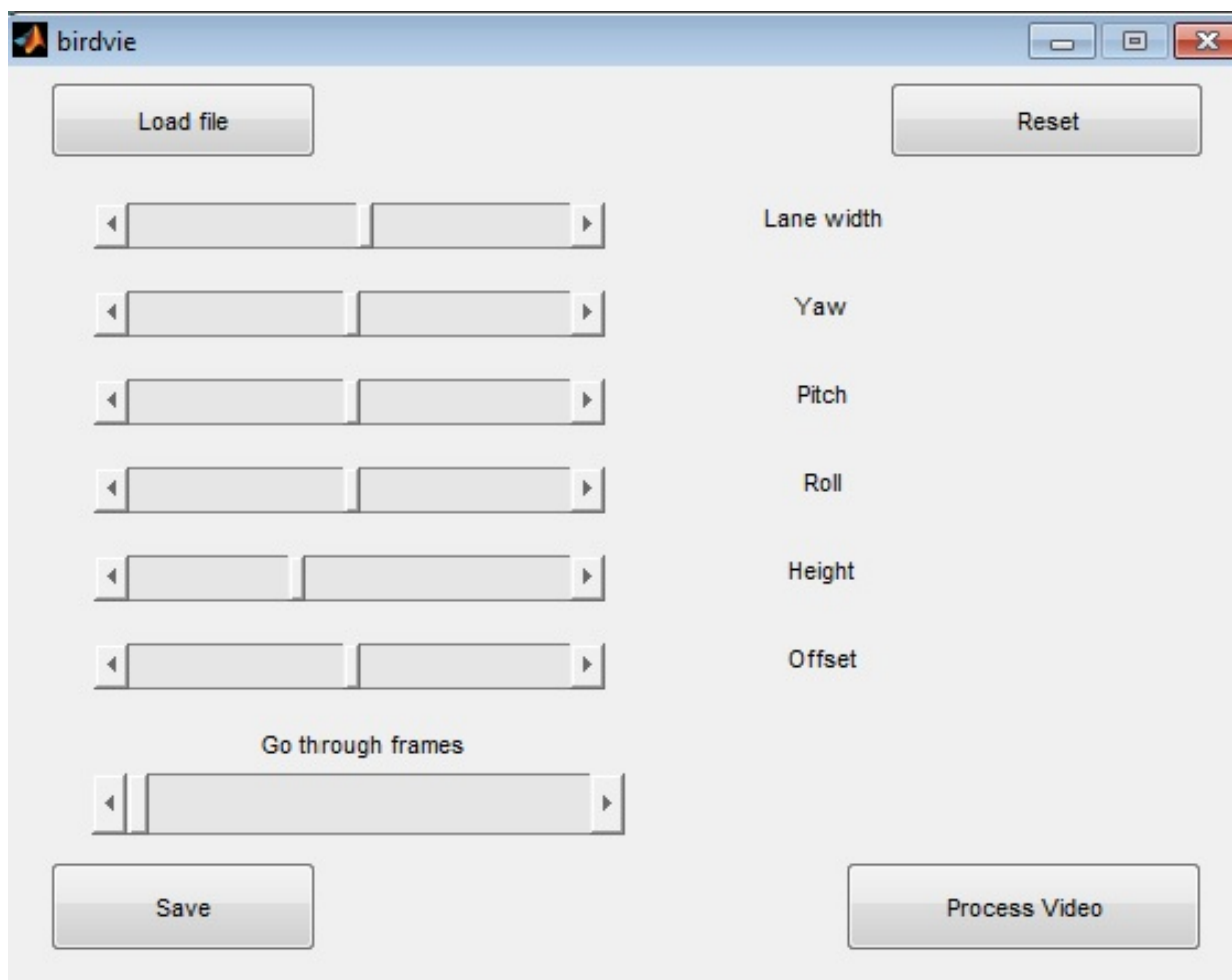


Figure 2.24: *Interface 1, GUI for Lytx videos*

Table 2.11: Explanation of Interface 1, GUI for Lytx.

Number	Description	Explanation
1	Load file	By pressing the button the user can choose a video, of which the front and the top view will be displayed in a separate figure. In addition to that the relative lane position of a vehicle, which is placed in the middle of the lane is fed into both views
2	Lane width	By using the slider the user can adjust the lane width
3	Yaw angle	By using the slider the user can adjust the yaw angle of the camera
4	Pitch angle	By using the slider the user can adjust the pitch angle of the camera
5	Roll angle	By using the slider the user can adjust the roll angle of the camera
6	Height	By using the slider the user can adjust the height of the camera in reference to the the road
7	Lateral offset	By using the slider the user can adjust the lateral offset of the camera in reference to the center of the vehicle
8	Frames	By using the slider the user can switch between twenty random frames of the video
9	Reset	By using the button the user can reset all sliders 2-7 to its defaults values
10	Save video	By using the button the user can save the current extrinsic parameters
11	Process video	By using the button the user opens a new GUI to process the video with the current parameters

2.7.1.2 Interface 2

The purpose of this second interface, shown in figure2.25 is to extract the lane offset from one video using the in GUI 2.24 chose camera parameters. The Gui offers the user to show the top view through check box 1 and the front view through checkbox2. To save the before chosen animation checkbox 3 has to be ticked.

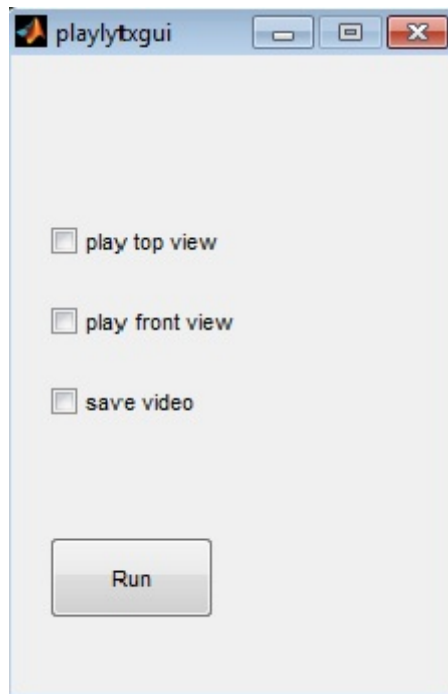


Figure 2.25: Interface 2, GUI for Lytx videos

Table 2.12: Explanation of Interface 2, GUI for Lytx

Number	Description	Explanation
1	play top view	By ticking the check box the user can decide to show the top view
2	play front view	By ticking the check box the user can decide to show the front view
3	save video	By ticking the check box the user can decide to save the previously chosen animations.
4	Run	By using the button the user start the computation, with the previously chosen settings.

3 Results

3.1 Qualitative Results

Some events where the method was affected by different road conditions were selected to analyze the performance of *LOC* and *LOV* visually.

3.1.1 Roundabouts

In presence of roundabouts, the *LOV* performs better than the *LOC*. In the figure 3.1 it is shown the lateral position of the vehicle in the presence of a roundabout. Around $t = 15s$ the *LOC* signals stop, while the *LOV* signals continues with a smooth variation in both sides.

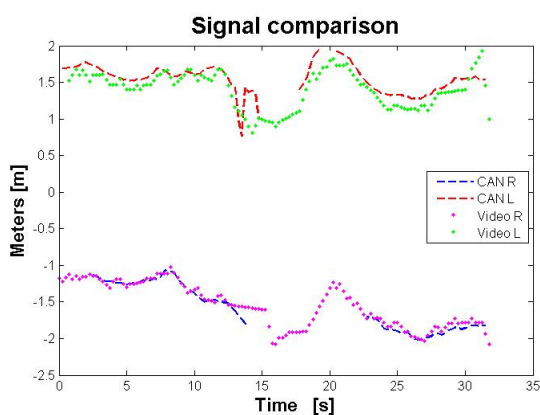


Figure 3.1: *Effects of Roundabouts*

3.1.2 Crosswalks

A similar result is obtained in presence of crosswalks. In figure 3.2 it is shown the *LOV* and *LOC* signals when there are crosswalks. The figure 3.3 shows the corresponding perspective both in front and top view.

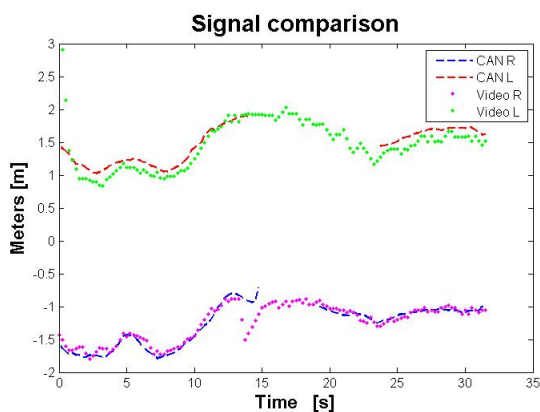


Figure 3.2: *Effects of Crosswalks graph*

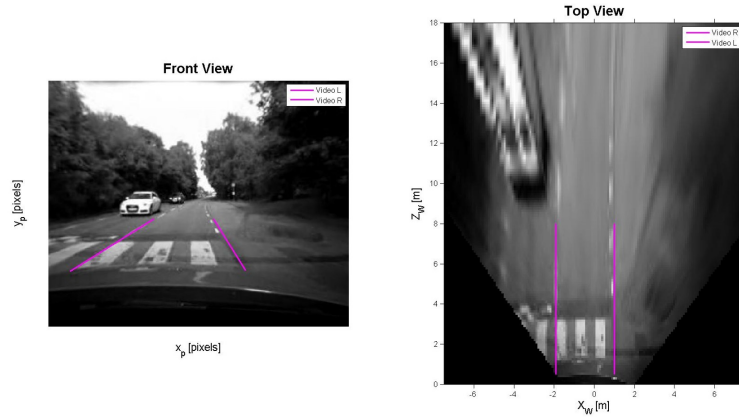
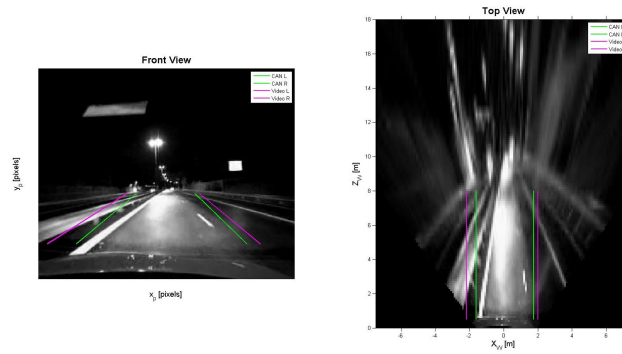


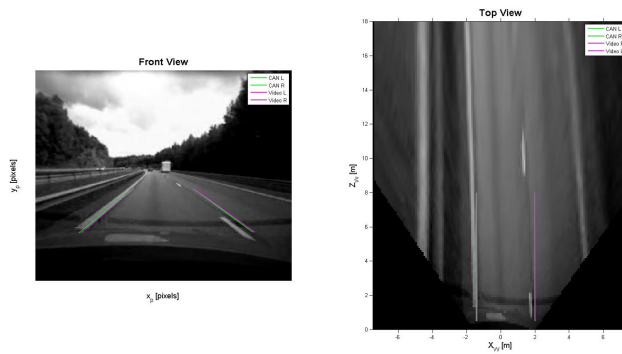
Figure 3.3: *Effects of Crosswalks Image*

3.1.3 Extrinsic parameters

When the extrinsic parameters are wrong or they have an error, it is possible to see it. Figure 3.4 shows both cases, when these parameters are wrong (figure 3.4(a)) and when they are right or closer to their real value (figure 3.4(b)).



(a) Wrong parameters



(b) Right parameters

Figure 3.4: *Effects of extrinsic parameters*

3.1.4 Speed bumps

The presence of speed bumps means a change in the height and pitch of the camera. These modifications affect the precision of both signals *LOV* and *LOC*. The figure 3.5 shows a bouncing sequence caused by a speed

bump.

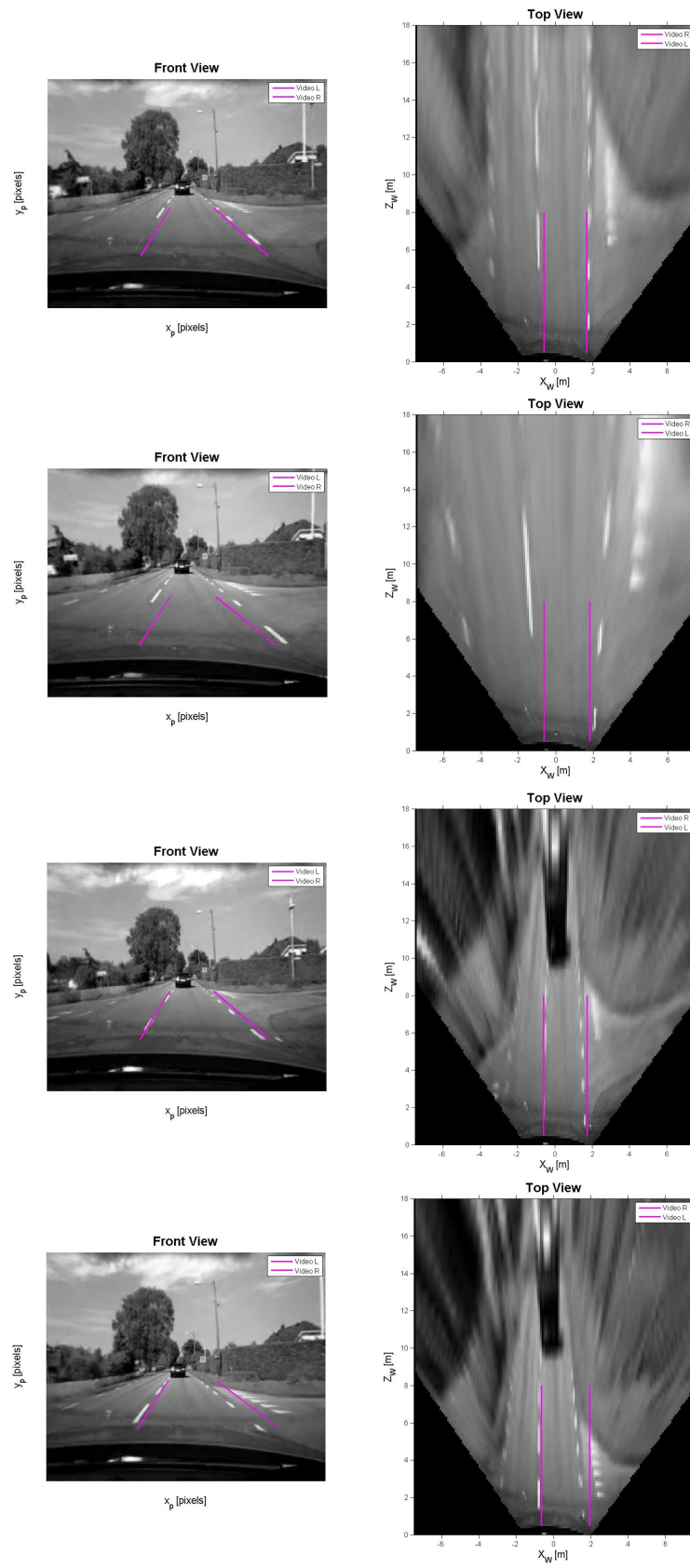


Figure 3.5: *Effects of pitch rate*

3.2 Quantitative Results

3.2.1 Independent events

Every single case of the total number of events of reliable data set is first evaluated individually. A sample case is introduced in the following. The event consists out of 242 comparable measurements. The figure 3.6 shows the distribution of the error for the different measurements in each event. The arithmetic mean and the variance is calculated according to section 2.6.1.1.1. These values are used to generate a normal distribution of the error for the event, which can be seen in figure 3.7.

Table 3.1: Analysis of a single event

Event Code	Number of measurements	mean error	standard deviation
45770700001	242	0.0505	0.0992

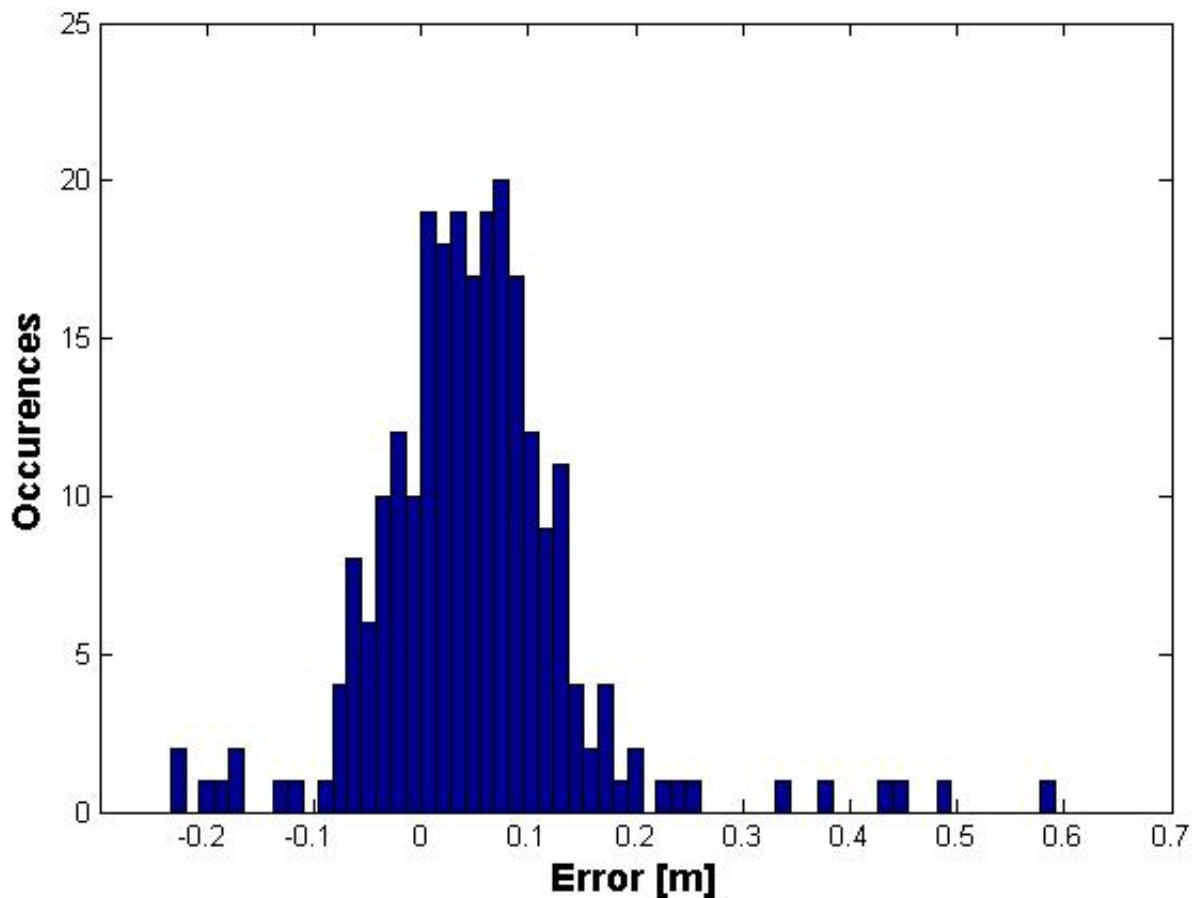


Figure 3.6: Histogram of error of a single event

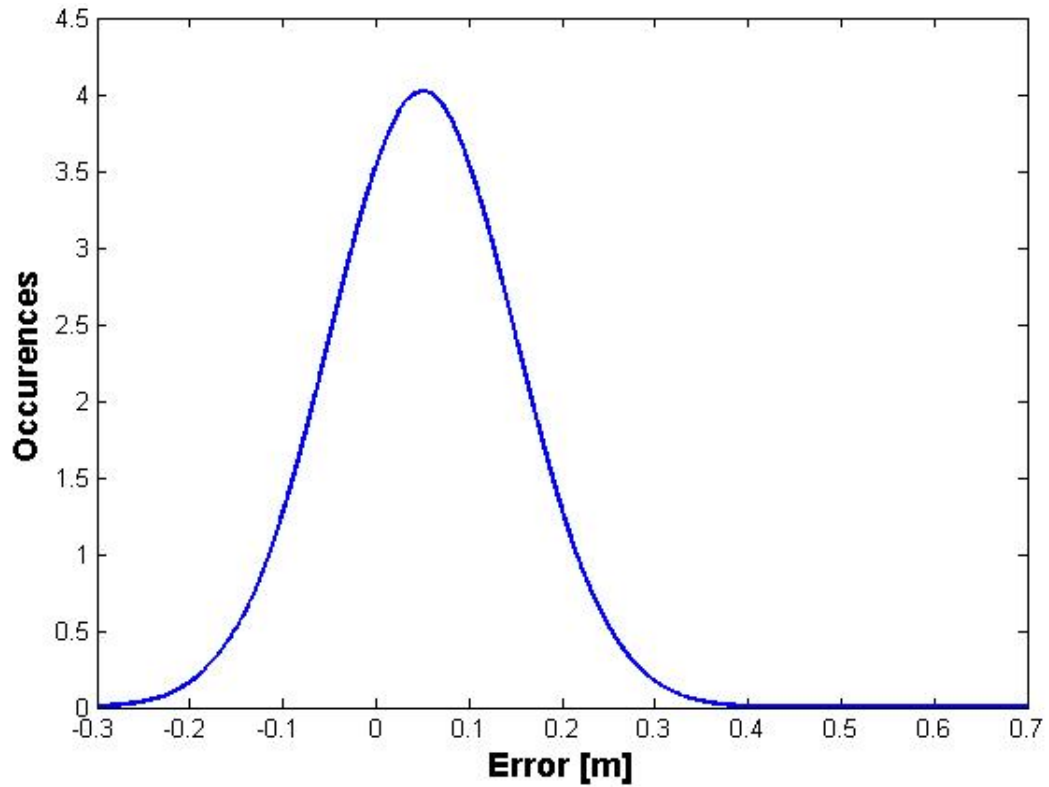


Figure 3.7: Normal distribution of error of a single event

3.2.2 Event categories

The independent events can be classified into categories depending on the time of the day, the road shape and longitudinal acceleration. Figure 3.8 shows the different categories within our set of reliable cases.

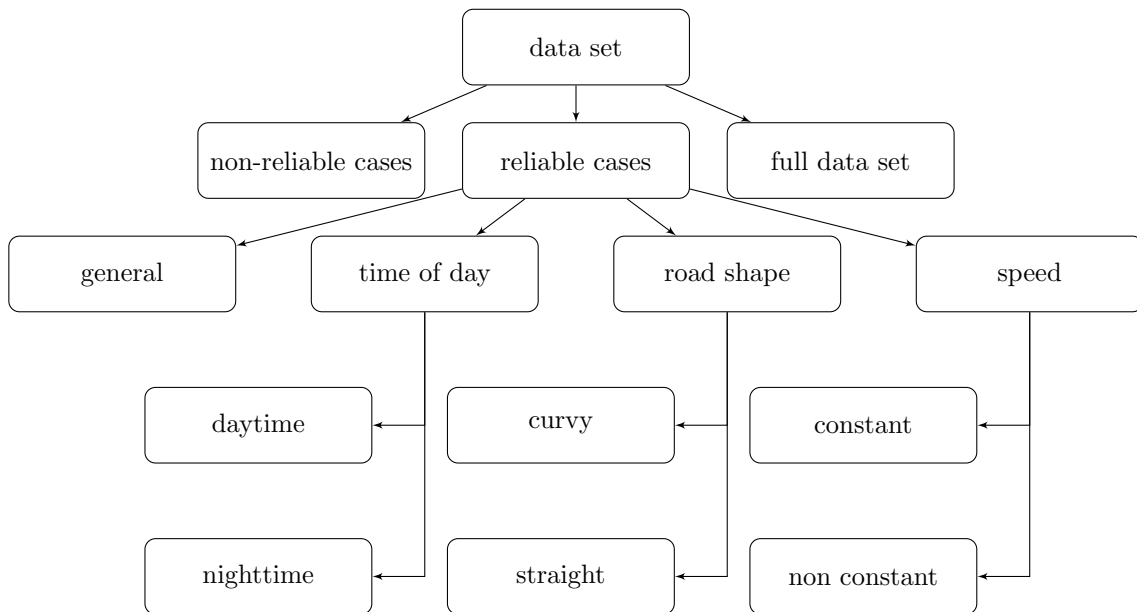


Figure 3.8: Evaluation Categories

3.2.2.1 Results for each category

Table 3.2 shows the different subsets for each category with the number of cases and the total number of measurements. The weighted mean and standard deviation is calculated to further analyze the specific cases.

Table 3.2: Analysis of different categories

type of case	Cases	Measurements	mean	Standard deviation
general	818	173231	0.2557	0.525
daytime	598	126950	0.2464	0.5131
nighttime	171	36243	0.2909	0.557
straight	818	138628	0.2502	0.5184
curvy	677	34603	0.2778	0.5155
constant speed	818	103737	0.2556	0.5125
non-constant speed	816	69494	0.2559	0.5353

Figure 3.9 shows the error distribution for all 818 cases. Furthermore, the error distribution for the different categories is shown in figures 3.10 - 3.15.

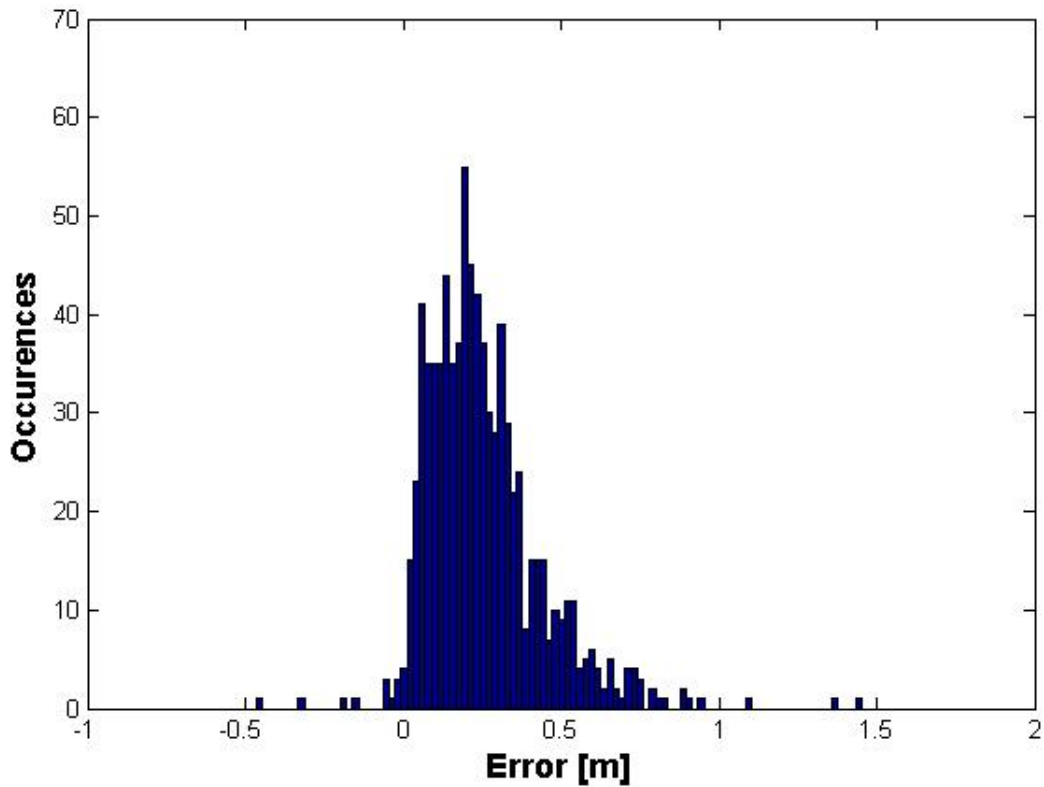


Figure 3.9: *General error*

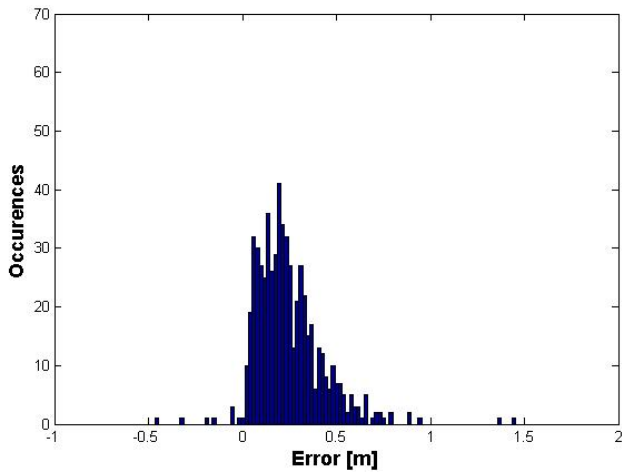


Figure 3.10: *Daytime error*

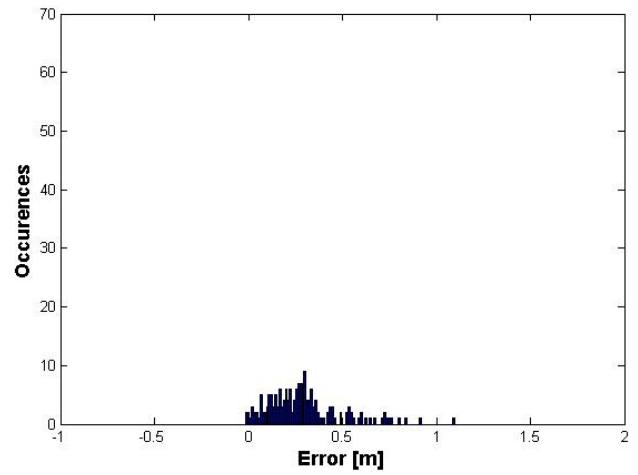


Figure 3.11: *Nighttime error*

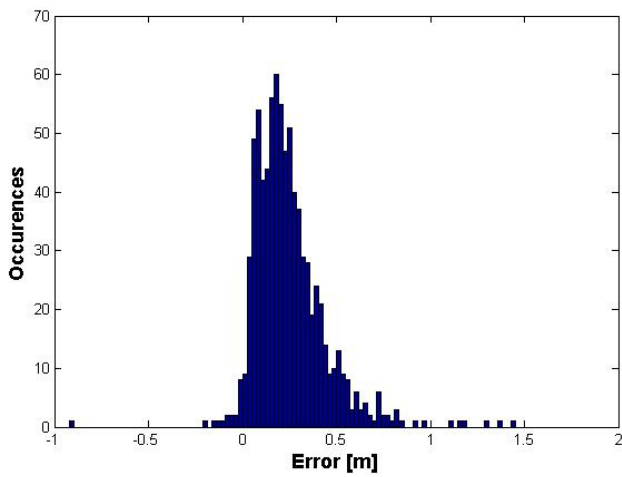


Figure 3.12: *Straight road error*

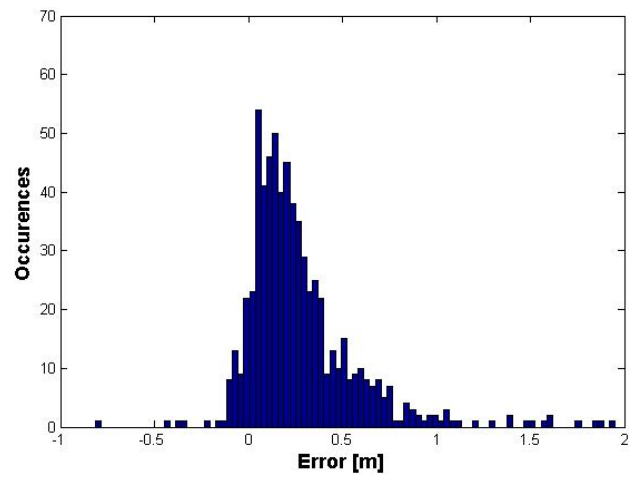


Figure 3.13: *Curvy road error*

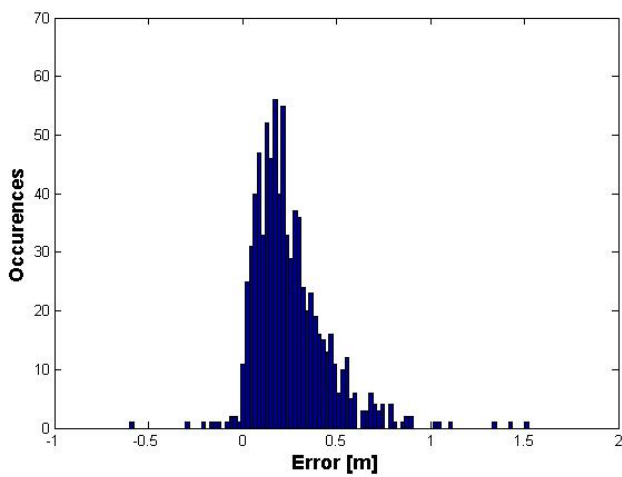


Figure 3.14: *No acceleration error*

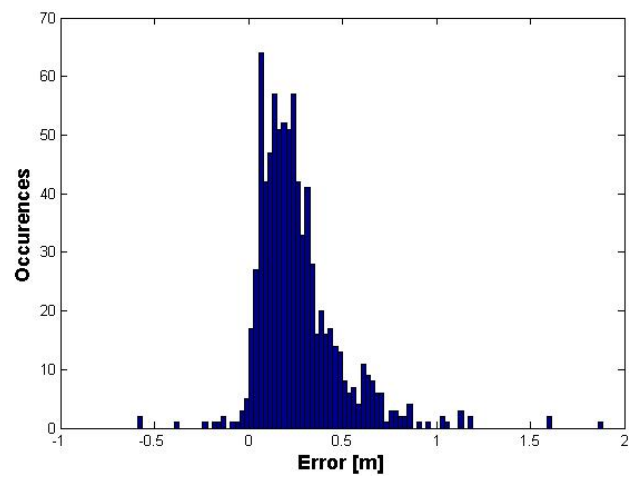


Figure 3.15: *Acceleration error*

Figure 3.16 displays the normal distribution for the general error and each of the six other categories.

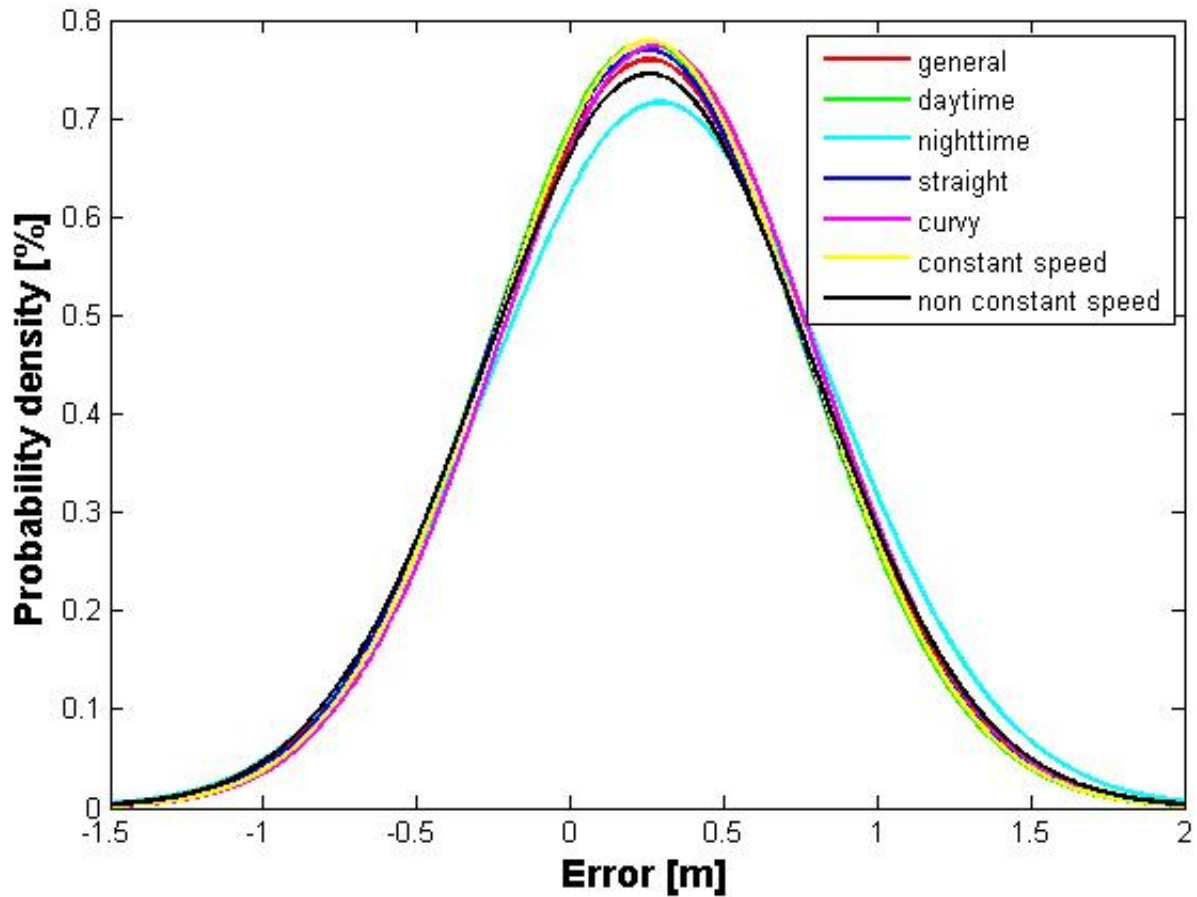


Figure 3.16: Normal distribution for each category

3.3 Sensitivity results

Figures 3.17 and 3.18 show the relative change of the *LOV* when there is a change in the position and orientation of the camera. These plots were obtained using the equation 2.44.

The figures 3.19 and 3.20 show the change in the lateral offset when the position and orientation of the camera change. These plots follow the equation 2.43.

The horizontal axis represents the changes from the original mounting point of the camera.

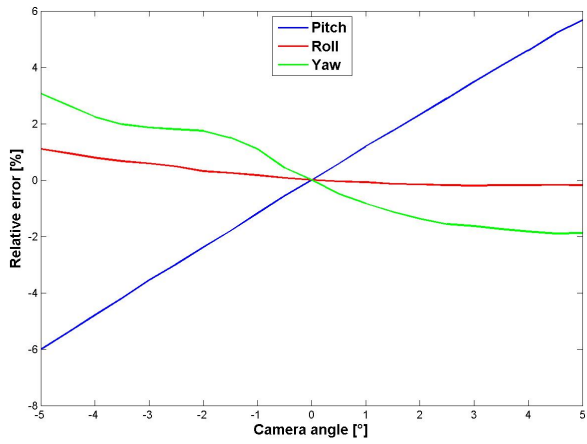


Figure 3.17: *Relative error [angles].*

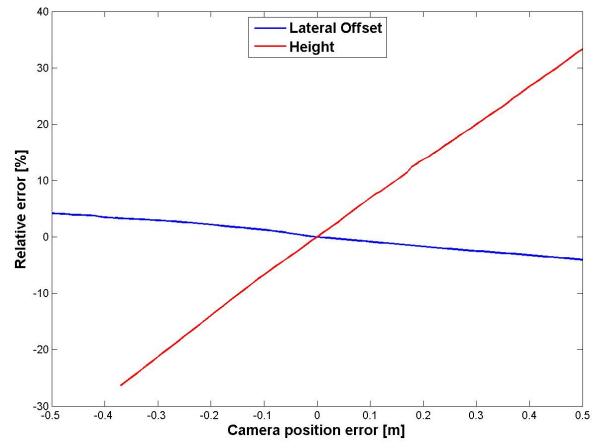


Figure 3.18: *Relative error [distances].*

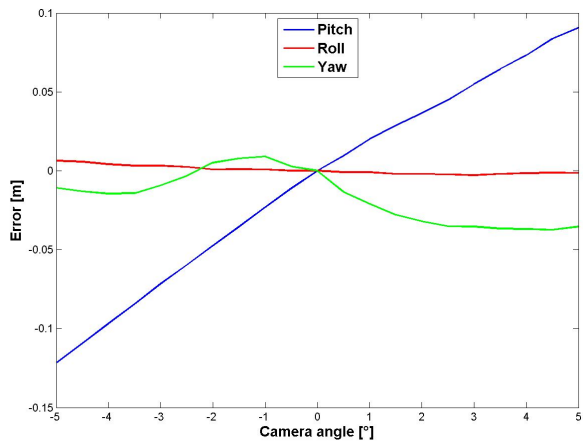


Figure 3.19: *Error [angles].*

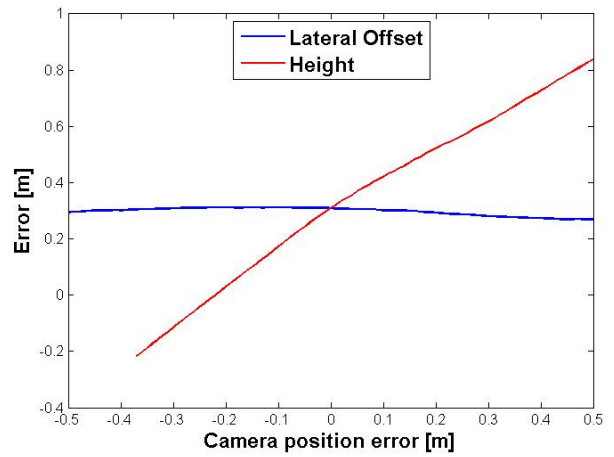


Figure 3.20: *Error [distances].*

3.4 Versatility results

The figure 3.21 shows the result of using the developed method in a LYTX video with the help of the corresponding GUI (see section 2.7).

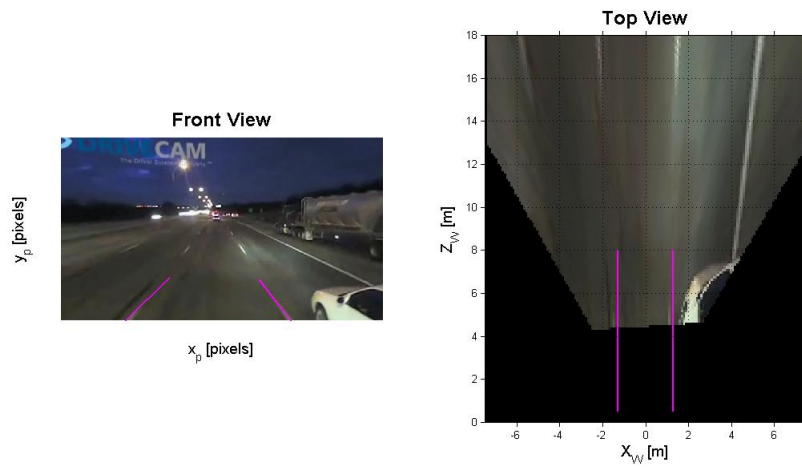


Figure 3.21: *Plot from GUI for a Lytx video*

4 Discussion and further work

4.1 Qualitative results

Plotting the CAN and Video signals in the front and the aerial view can give the user an impression of the accuracy of the *LOV* as well as the *LOC*. Using these plots the user can observe that the method is susceptible to changes in the vertical displacement of the vehicle (speed bumps) and pitch changes (acceleration/deceleration). A possible way to reduce these effects would be extracting the pitch and the horizon line from the vanishing point and produce a video confidence signal similar to the one used with in the CAN network. Also noticeable is the fact that the developed method is better than the CAN signal in the presence of crosswalk markings and roundabouts. Convert this qualitative analysis into a quantitative one, would involve manual labeling of every frame in every video with crosswalks and roundabouts. It is not recommended to carry out this analysis. It would be time consuming and not conclusive since there is not enough video material covering these specific conditions.

4.2 Quantitative results

The discussion of the quantitative results will cover an explanation of the error between *LOC* and *LOV*. It is based on the the results for the estimated arithmetical mean and the estimated standard deviation. The error for the entire subset will be discussed in detail in 4.2.1. Further remarks on the separate categories will be made in section 4.2.2 - 4.2.4.

4.2.1 General error

The accuracy of the tool can be assessed through the mean error. The tool generates a mean error of 0.26m. The accuracy is highly dependent on the extrinsic parameters, which is further explained in section 4.3. The accuracy can be improved by paying closer attention to the position and orientation of the camera. The precision of the tool has been evaluated using the standard deviation.

The standard deviation for the general error is 0.53m. This is caused by many factors, such as the driving motion of the vehicle, the road conditions, the light conditions and the type of lane marking, just to name a few. Nevertheless, for the intended use of the tool this value is tolerable. The standard deviation could be reduced by further investigating the before named factors and reducing their influence.

It is further worth noting that the histogram is not symmetric as seen in figure 3.9. In specific situation the tool detects the road edge or even lane markings of the adjacent lane. This causes a higher spread for values greater than the mean. The unwanted detection can be reduced by finding more effective thresholds, which can lower the amount of false line detection.

4.2.2 Time of day

The results show that the tool performs better in daytime. The mean error is less in daytime compared to nighttime. The standard deviation is greater during nighttime. The tool is not as precise during the night as it is during the day. This is mainly dependent on the light conditions.

4.2.3 Road shape

The results show that the algorithm performs better for straight parts of the road compared to curvy parts in terms of accuracy. In curvy parts of the road the straight line detection overestimates the lane offset. The precision of the tool is similar for straight and curvy parts. No relevant change in the spread of values can be observed here.

4.2.4 Speed

The results show that the tool performs similar during stretches of acceleration and deceleration compared to stretches of constant speed in terms of accuracy. However, the performance differs in terms of precision. During stretches of constant speed the standard deviation is less compared to acceleration and deceleration stretches. This is caused by changes in height and pitch due to changes in speed.

4.3 Sensitivity analysis

The figures 3.17, 3.18, 3.19 and 3.20 show that the parameter that strongly affects precision of the results is the height of the camera. It is recommended to double check the position and orientation of the camera before using the developed method.

The sensitivity analysis was performed changing one parameter at a time. In the future, it would be interesting to perform this sensitivity analysis but now changing all parameters at the same time. Obviously a plot mapping the five dimensions would not be possible to generate, but it would be possible to find the dependency and correlation between them.

4.4 Versatility

The tool can be applied to databases with a front view video available. The challenge for the user will be to set the correct extrinsic parameters for getting sufficient results, due to the high sensitivity of the method. The developed GUI tries to provide an interface where the user can change the parameters quickly and gets to see the changes of these results instantly. Nevertheless the user should be experienced in the field of video analysis and should have a notion of how different extrinsic camera parameters effect the animations to use the tool to its full potential.

5 Conclusions

The goal of the thesis is to develop a method to analyze large quantities of video data in a quantifiable and efficient way to gain information about the lateral position of the vehicle.

Therefore a tool has been created to extract a lane offset signal from the EOR data base. The user is able to successfully extract the signal from the video using the tool.

The performance of the tool has been evaluated in terms of precision and accuracy using a corresponding CAN-Bus Signal from the EOR data base. The tool has been evaluated for different driving, road and light conditions. This can help to distinguish different factors, which influence the performance of the tool.

Further the influence of changes in camera position and camera orientation have been examined. The sensitivity of the tool with regard to changes in the extrinsic parameters is an important measure for understanding the performance of the tool. The tool is highly dependent on correctness of extrinsic parameters to give reliable results.

In addition to that, modifications to the tool have been made to make it applicable to various data bases with front view camera data. For versatility a graphical user interface has been created to set extrinsic camera parameters for data bases, where the exact parameters are not available.

The initial aims have been covered in the course of the thesis. This thesis work can facilitate the analysis of naturalistic driving data and contributes to the field of active safety research.

References

- (2015). 1st ed. World Health Organization.
- Akamatsu, Toshihiro, Fangyan Dong, and Kaoru Hirota (2013). 3D measurement of a moving object using 3D accelerometer attached to moving camera. *2013 IEEE 8th International Symposium on Intelligent Signal Processing*. DOI: 10.1109/wisp.2013.6657485.
- Bärgman, J (2017). “Methods for Analysis of Naturalistic Driving Data in Driver Behavior Research”. PhD thesis. Chalmers University of Technology.
- Bigelow, B (2009). New DriveCam CEO Is Focused on the Road Ahead. *Xconomy*. URL: <http://www.xconomy.com/san-diego/2009/03/26/new-drivecam-ceo-is-focused-on-the-road-ahead/2/>.
- Bouguet, Jean-Yves (2017). *Camera Calibration Toolbox for Matlab*. Intel.
- Carney, C et al. (2015). *Using Naturalistic Driving Data to Assess the Prevalence of Environmental Factors and Driver Behaviors in Teen Driver Crashes*. URL: <https://www.aaafoundation.org/sites/default/files/2015TeenCrashCausationReport.pdf>.
- Dozza, Marco (2016). *Naturalistic and Field Data*.
- Duda, Richard O. and Peter E. Hart (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM* **15**.1, 11–15. DOI: 10.1145/361237.361242.
- Enqvist, Olof (2016). *Camera geometry*.
- Fokkinga, M. (2011). The Hough transform. *Journal of Functional Programming* **21**.02, 129–133. DOI: 10.1017/s0956796810000341.
- Grompone von Gioi, Rafael (2014). A Contrario Line Segment Detection. *SpringerBriefs in Computer Science*. DOI: 10.1007/978-1-4939-0575-1.
- Hough, P.V.C. (1962). *Method and means for recognizing complex patterns*. US Patent 3,069,654. URL: <https://www.google.com/patents/US3069654>.
- John, Neethu, B. Anusha, and Krishnan Kutty (2015). A Reliable Method for Detecting Road Regions from a Single Image Based on Color Distribution and Vanishing Point Location. *Procedia Computer Science* **58**, 2–9. DOI: 10.1016/j.procs.2015.08.002.
- Lopez, A. et al. (2010). Robust lane markings detection and road geometry computation. *International Journal of Automotive Technology* **11**.3, 395–407. DOI: 10.1007/s12239-010-0049-6.
- Meng, L and W Yifei (2016). “Tool development and quantitative analysis for naturalistic Left Turn Across Path/Opposite direction (LTAP/OD) driving scenarios”. PhD thesis. Chalmers University of Technology.
- Satzoda, R. K., S. Suchitra, and T. Srikanthan (2012). Robust extraction of lane markings using gradient angle histograms and directional signed edges. *2012 IEEE Intelligent Vehicles Symposium*. DOI: 10.1109/ivs.2012.6232296.
- Tan, Hui et al. (2013). Research on Lane Marking Lines Detection. *Applied Mechanics and Materials* **274**, 634–637. DOI: 10.4028/www.scientific.net/amm.274.634.
- Toran Marti, Felix (2010). *Optimal threshold*. The MathWorks Inc.
- Wolfe, G and K Gwin (2010). DriveCam Can: Video Technology Reduces Vehicle Crash Risk. *State Magazine*. URL: <http://www.xconomy.com/san-diego/2009/03/26/new-drivecam-ceo-is-focused-on-the-road-ahead/2/>.

Appendices

.1 Code

.1.1 Undistortion

```
function I = undistort(Idistorted , params)
    fx = params.fx;
    fy = params.fy;
    cx = params.cx;
    cy = params.cy;
    k1 = params.k1;
    k2 = params.k2;
    k3 = params.k3;
    p1 = params.p1;
    p2 = params.p2;

    K = [fx 0 cx; 0 fy cy; 0 0 1];

    I = zeros(size(Idistorted));
    [i , j] = find(~isnan(I));

    % Xp = the xyz vals of points on the z plane
    Xp = inv(K)*[j i ones(length(i),1)]';

    % Now we calculate how those points distort i.e forward map them through the distortion
    r2 = Xp(1,:).^2+Xp(2,:).^2;
    x = Xp(1,:);
    y = Xp(2,:);

    x = x.*(1+k1*r2 + k2*r2.^2) + 2*p1.*x.*y + p2*(r2 + 2*x.^2);
    y = y.*(1+k1*r2 + k2*r2.^2) + 2*p2.*x.*y + p1*(r2 + 2*y.^2);

    % u and v are now the distorted coordinates
    u = reshape(fx*x + cx,size(I));
    v = reshape(fy*y + cy,size(I));

    % Now we perform a backward mapping in order to undistort the warped image coordinates
    I = interp2(Idistorted , u , v);
end
```

Figure A1: Undistort function