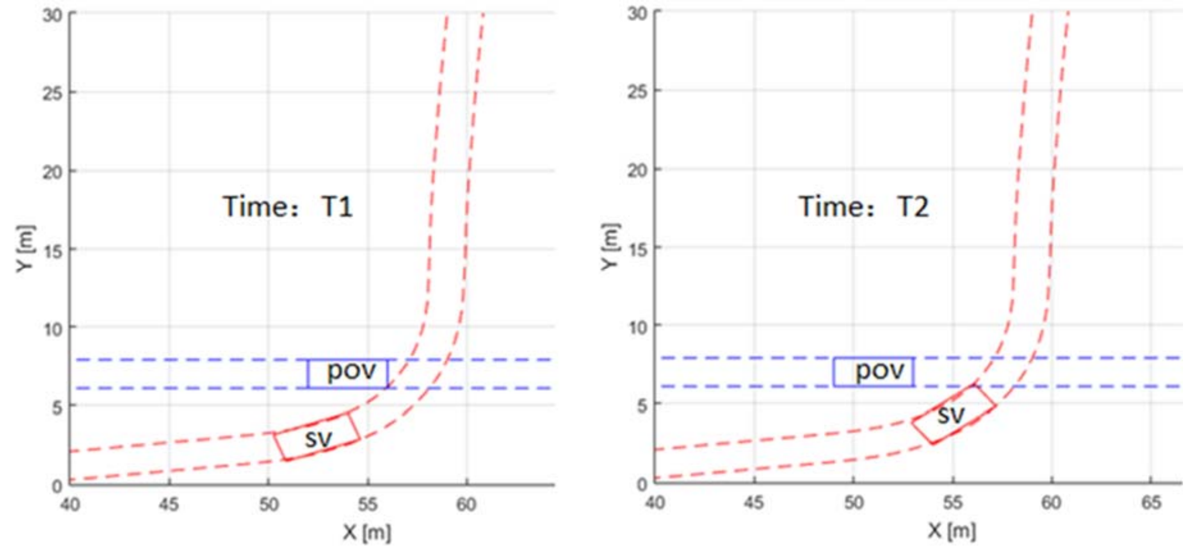


Tool development and quantitative analysis for naturalistic Left Turn Across Path/Opposite direction (LTAP/OD) driving scenarios

Master's thesis in Mechanical Engineering

Lin Meng & Yifeng Wang

Department of Applied Mechanics
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2016

MASTER’S THESIS IN PROGRAMME NAME

Tool development and quantitative analysis for naturalistic Left Turn Across Path/Opposite direction (LTAP/OD) driving scenarios

Lin Meng & Yifeng Wang

Department of Applied Mechanics
Division of Vehicle Safety
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2016

Tool development and quantitative analysis for naturalistic Left Turn Across Path/Opposite direction (LTAP/OD) driving scenarios

Lin Meng & Yifeng Wang

Master's Thesis 2016:53
ISSN 1652-8557
Department of Applied Mechanics
Division of Vehicle Safety
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Cover:
The cover is a simulation result from EuroFOT data set. The kinematic parameters of POV is obtained by utilizing the tool developed in the thesis.

Department of Applied Mechanics
Göteborg, Sweden 2016-06-05

Tool development and quantitative analysis for naturalistic Left Turn Across Path/Opposite direction (LTAP/OD) driving scenarios
[Abstract]
Master's thesis in Master's programme name (if applicable)

Lin Meng & Yifeng Wang
Department of Applied Mechanics
Division of Vehicle Safety
Chalmers University of Technology

Abstract

This thesis aims to develop a tool to obtain key variables of other vehicles from video and apply this tool for the quantitative analysis of driver behaviour for Left Turn Across Path/Opposite Direction scenarios. The variables include relative speed and relative positions between subject vehicle and oncoming vehicle.

Three methods are discussed and implemented in software tool for manual annotation,: a ground points method, a vehicle width method and an optical flow matching method In addition Kalman filter is applied to integrate this three methods together with a constant acceleration model. An experiment shows the range estimation result has an average percentage error of less than 10%, within the range 10m to 50m, and that the speed estimation has around a 10% error at approximately 10m and 20% error around 20m.

Semi-automatic methods for extracting the desired variables is also presented. Based on manually selected tracking region in the first frame, and optical flow computed through the video, the desired (manually selected) region can be tracked. Optical flow vectors on the region has a relationship with motion. Motion estimation is accomplished with a matching process.

After applying the tool on 102 LTAP/OD cases in a subset of EuroFOT data, the Post Encroachment Time was calculated for each. Results show that drivers feel comfortable to turn into the encroachment zone in a range between 2 and 4s after the last oncoming vehicle leaves that zone.

Key words: optical flow, vehicle kinematics, naturalistic driving data, LTAP/OD, manual annotation

Contents

Acknowledgements

We would like to express our deep appreciation for our examiner Jonas Bärgman and supervisors Giulio Francesco Bianchi Piccinini together with Christian Berger. They provided us the possibility and recourses to perform the thesis and they are very supportive throughout the thesis.

We would also like to thank SAFER for providing us a productive working environment and Volvo Car Cooperation that supports us to finish the experiment.

Further we had valuable help from Sergejs Dombrovskis who guided us in GPS data processing and Christian -Nils Boda who contributed to camera calibration. They are really patient and warm-hearted researchers and made our study more effective.

Finally, we would like to thank all our families and friends for supporting us in all situations throughout the thesis.

Göteborg March 2016-06-05

Lin Meng & Yifeng Wang

1 Introduction

1.1 Background

More than 1.2 million people die every year on the roads, making road traffic injuries a leading cause of death all over the world [1]. Globally, road traffic crashes are one of the main cause of death among young people aged between 15 and 29 years in 2012 and bring both huge financial and emotional damage to families, societies and countries.

In Europe, although much progress has been made to improve road safety, more than 25,000 people still died in traffic in 2013 [2], as shown in Figure 1.1 .To address the traffic safety problem through a concrete goal, several countries have engaged in the concept of Vision Zero formulated by Swedish National Road Administration. The statement of this vision is that “No one shall be killed or seriously injured in road traffic systems”, which underlines that the loss of lives in road traffic is unacceptable. To achieve this very ambitious goal, efforts should be made to understand accident causation mechanisms and to improve road infrastructure, vehicle safety design and driving behavior.

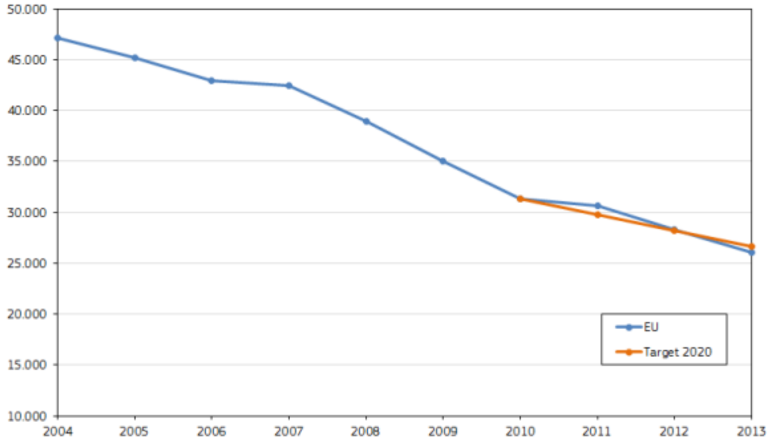


Figure 1.1 Number of road accident fatalities, EU, 2004-2013[2]

Among all the possible solutions to traffic safety issues, autonomous vehicles seem to have high potential and to be supported by nearly all Original Equipment manufacturers and even by some businesses external to the automotive field, like Google [3]. The expected safety benefit of autonomous vehicles is the reduction of crashes, due to smoother traffic flow and elimination of driver errors. However, there are still several challenges that should be taken into account for the design of automated vehicles and one of the most important is how to ensure that the vehicle “behaves” as a human and take into account driver’s comfort zones [4].

In this context, this thesis aims to better understand drivers’ behavior in left turns maneuvers after giving way to oncoming vehicle from the opposite direction at a non-signalized intersection. This knowledge can be used for the design of appropriate trajectories for autonomous vehicle. Besides, it is the aim of this thesis to develop a tool to semi-automatically calculate the speed of the oncoming vehicle and its distance from the vehicle performing the left turn maneuver. The scenario is a good application of this tool.

1.2 Literature Review

1.2.1 Left Turn Across Path / Opposite Direction

Left Turn Across Path (LTAP) scenarios are driving situations in which two vehicles are traveling on the same roadway and one vehicle is entering an intersection to make a left turn across the path of the other vehicle[36]. Among the LTAP scenarios, special attention should be placed on Left Turn Across Path Opposite Direction (LTAP/OD) scenarios because is one of most frequent path crossing types and accounts for more than 27 % of intersection crashes [5]. The LTAP/OD is shown in Figure 1,2, where the left turning black vehicle is called Subject Vehicle (SV) and the oncoming blue vehicle is named Principal Other Vehicle (POV). The same acronyms will be used along this document.

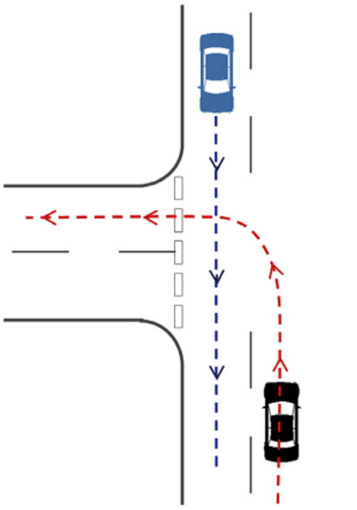


Figure 1.2 Illustration of Left Turn Across Path/Opposite Direction (LTAP/OD) scenario [6]

In order to quantify drivers' behaviors and develop driver models for LTAP/OD, it is necessary to develop measurements which can provide an objective assessment of conflict situations and facilitate the understanding of such scenarios.

Many studies of LTAP/OD scenarios used post-encroachment time (PET) [6,7,8]to define the time gap between the moment when the first vehicle leaves the encroachment zone and the moment when the second vehicle enters the encroachment zone. The encroachment zone is the area defined by the trajectories of SV and POV which is often considered as a rhomboid zone and is depicted in red in Figure 1.3.

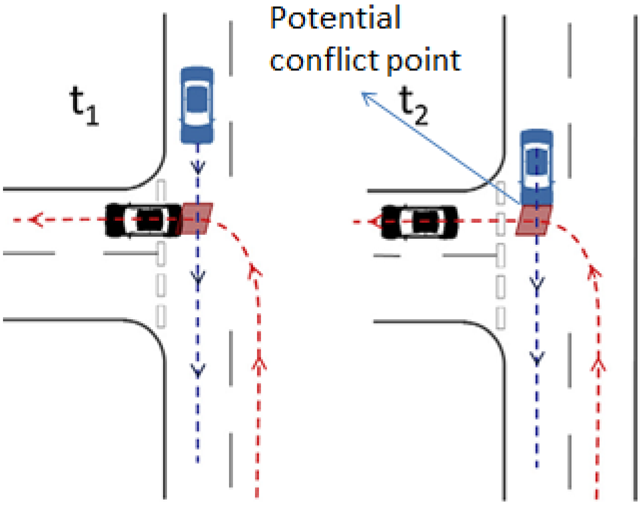


Figure 1.3 Illustration of PET, encroachment zone and potential conflict point. *PET*=$t_2 - t_1$ *[6].*

As can be seen from the definition, the PET is a measure of the available time gap before a crash can occur. Other predictive computed metrics such as trailing buffer (TB) and leading buffer (LB), are also used to estimate the gap time prior to the formation of the conflict configuration [7]. To define these metrics, it is relevant to define the potential conflict point as reference and can often be found at the top right corner of the encroachment zones as illustrated in Figure 1.3 when the SV crosses first. The TB is a measure of the time elapsed from the passing of SV at the potential conflict point and the subsequent arrival of POV at the potential conflict point. The TB is, however, valid only when the SV passes the potential conflict point before the POV: if POV arrives at the potential conflict point before the SV, the time interval is called LB and the potential conflict point is at the bottom left corner. The noticeable difference between PET and TB/LB is that TB/LB utilizes a variable reference speed profile to predict the moment in which the SV or the POV will reach the potential conflict point and this speed profile is generally obtained from a driver model [8]. In some cases, to simplify the calculation or without proper speed estimation model, TB/LB can be calculated with a constant speed and, in such cases, it is called Gap Time (GT) [8].

In the thesis, the PET will be used for the quantitative assessment of drivers' behaviour and, in particular, comfort zone boundaries, in the LTAP/OD scenarios. This choice is motivated by the fact that naturalistic driving data provide the kinematic variables for both SV and POV. The latter should be however extracted by using image processing methods, as described later.

1.2.2 Naturalistic Driving Data

Naturalistic driving data is data collected from Naturalistic Driving Studies (NDS) in real traffic, by road users performing their usual daily activities [9]. As such, NDS provide a great opportunity to understand normal driving and the transition from normal driving to safety critical situations, such as near crashes[1] or crashes[2]. In NDS,

[1] A near-crash may be conceptualized as a situation where the magnitude of the required evasive maneuver approaches the vehicle capabilities

[2] A crash is any contact that the subject vehicle has with another conflict partner, either moving or fixed, at any speed that is observable or in which kinetic energy is measurably transferred or dissipated. This excludes roadway features meant to be driven over such as speed bumps.

the test vehicles are equipped with sensors and video cameras and the data collection is unobtrusive and has little effect on normal driving. A NDS can collect data either continuously during the whole trip, or by triggering only specific events, based on kinematic thresholds (e.g. longitudinal and lateral acceleration). A reference for all NDS is the 100-car naturalistic driving study where 100 cars were driven in real traffic for more than one year and more than 4200 hours driving data were collected [9]. After that, several other NDS have been conducted such as SHRP2 [10].

Similar to NDS, Field Operational Tests (FOT) are often performed in a naturalistic fashion, but with the aim of evaluating a system (e.g. active safety systems) [11]. In Europe, EuroFOT was the first large-scale Field Operational Test, 102 vehicles collecting date for one year in Gothenburg and the aim of EuroFOT is to establish a comprehensive, technical, and socio/economic assessment programme for evaluating the influence of intelligent vehicle systems on safety, the environment and driver efficiency [12].

The main strength of NDS/FOT is that high-resolution driving performance data are coupled with video data [13]. Together, this data provide a rich source of information of drivers' behaviours in a normal driving environment which are valuable for understanding accident causation mechanisms as well as analyses of mobility patterns and traffic efficiency. The primary limitation of these studies regards the participants involved which often are representative of a specific geographical region. Besides, the number of crash events is limited and, therefore, near crashes are used as surrogate of near crashes for further quantitative analyses. The third problem is that the data is collected only for the SV and, therefore, limited information can be extracted for the POV or other relevant road users. This issue will be targeted by this thesis through the development of image processing algorithms to obtain key parameters of the POV and achieve a comprehensive understanding of different driving scenarios.

1.2.3 Driver model

The understanding of driver behaviour is a relevant aspect to take into account for the design of autonomous vehicles. For this reason, driver models are often used to describe driver's behaviours in terms of what he/she usually does and how he/she reacts in a specific driving situation. The interest for this topic is proved by the number of articles recently published on the topic.

Overall, driver models can be classified into two categories: descriptive models and functional models [14]. The first ones describe the driving task focusing on what the driver does. Such models have very limited predictive power, because they do not consider the factors that shape the different behaviours such as driver capabilities, skills, motivation and limitations in different scenarios [15]. By contrast, the second type model attempts to explain drivers' behaviours and predict drivers' performance in demanding or daily routine situations. These models are necessary to understand environmental factors, vehicle factors as well as human factors, and to design active safety systems adapted to driver needs [16].

The data available from NDS provide a unique possibility to develop new driver models and calibrate existing ones. The thesis focuses on building a second type model which describes drivers' left turn behaviours with the consideration the oncoming traffic by analysing the EuroFOT data and this model can be integrated both in path planning and estimation of driver behaviour for autonomous vehicle.

1.2.4 Camera model and calibration

As shown in Figure 1.4 [17], the simplest camera model is the pinhole model: a single ray from a distant object enters the pinhole and, then, is projected onto an image plane. As a result, the image on the image plane is always in focus and the geometrical relationship can be obtained by using the properties of similar triangles.

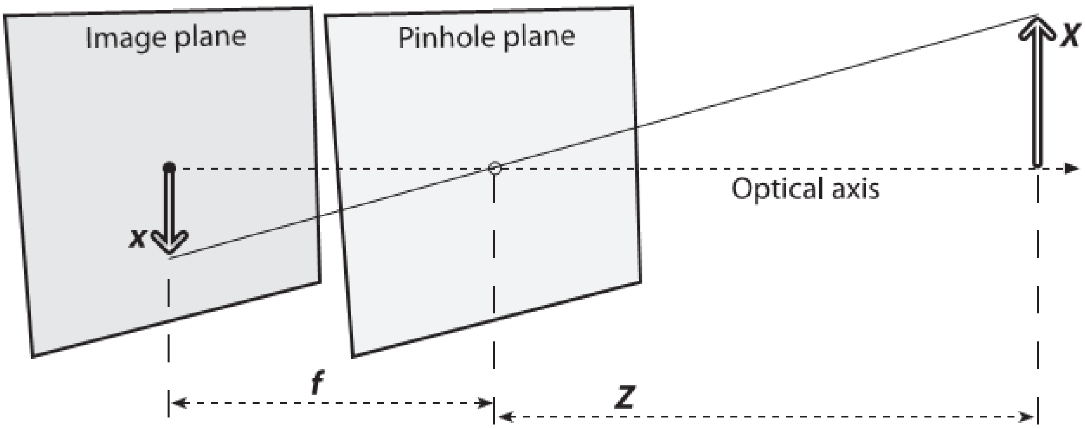


Figure 1.4 Pinhole camera model [17].

However, in this ideal model, very little light goes through a pinhole, thus to form images at a faster rate, a lens is used to gather more light from outside the camera and bend the light to converge at the projection point. In practice, for manufacturing and installation reasons, no lens is perfect, so the cost of this fast imaging is introducing distortions. Two common geometrical distortions are radial and tangential distortions [18]. The first kind of distortions arises because the light farther from the centre of a simple lens is bent too much compared to light that passes closer to the centre. The second-largest distortion is due to the fact that lens are not exactly parallel to the imaging plane as a result of assembling defects. The distortions can be rectified through the calibration of the camera and the process can be performed in Matlab [19,20].

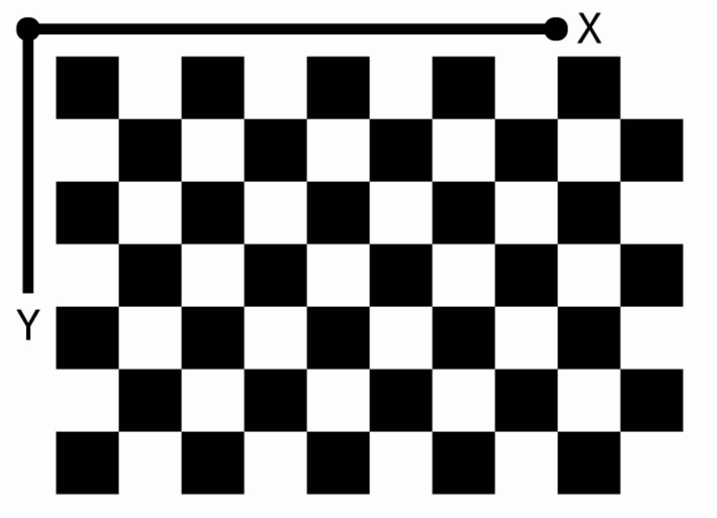


Figure 1.5 Checker board

With a calibration object as shown in Figure 1.5, usually called checker board, the intrinsic matrix M (3×3) and extrinsic matrix W (3×4) of a camera can be obtained by using the formulae below:

$$\mathrm{M} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1.1}$$

$$\mathrm{W} = [\mathrm{R}, \mathrm{T}] = \left[R_x * R_y * R_z, T\right] \tag{1.2}$$

where f_x and $f_y$are focal lengths in two perpendicular directions and c_x and $c_y$model a possible displacement of the centre of coordinates on the image plane (away from the optical axis). As for extrinsic parameters, rotating a coordinate system through each axis can get a 3×3 rotation metrix. The translation vector T presents a shift from one coordinate to anther system whose centre is at another position. In the thesis, only intrinsic parameters from this tool are used while extrinsic parameters are obtained though images captured by the camera since these parameters highly rely on the camera position and orientation.

1.2.5 Range and range rate estimation with single camera

As mentioned earlier, the position of other vehicles on the road is a key information to understand the driver behaviour of road users and to provide important information for the design of active safety systems. A linear model to extract vehicle range and optical parameters by on-screen distance measurement with the estimation of real width of target vehicles was previously built in [21]. On the other hand, Mobileye developed a vision based Adaptive Cruise Control system using a single camera input [22]: in their work, the authors discuss how the imaging geometry affects the range and range rate accuracy with the assumption of knowing the accurate mounting angle. In another work, [23] the authors developed a Forward Collision Warning system based on monocular vision where time to collision was calculated by calibrated camera with vanishing point [24] according to crossing ratio invariability. In [25], it is described a method to reduce the uncertainty of estimating 3D position with the perceived horizon and actual width of vehicles.

This thesis started from previous work [21, 22, 24, 25] and implemented a novel range rate estimation method by using optical flow and integrated these methods through Kaman filter.

1.2.6 Optical flow

Optical flow is the pattern that shows the motion of objects (feature points on the objects, specifically) in a video caused by the relative movement between the camera and the objects. [30] In computer vision, optical flow is extensively used in many areas, such as object detection and tracking, image dominant plane extraction, movement detection, etc. The target features, which are explained later, are tracked over time and their motion is converted into velocity vectors. These vectors represents the motion direction and distance of one pixel which moved between two consecutive frames. [17] Hence, these properties can be used for range and movement detection.

Dense optical flow is a common construction which associates a velocity with every pixel in a frame. In practice, if the tracked object is in single colour, it would be difficult to calculate the change of each pixel in the tracking area. Moreover, this dense method has high computation cost. [17]

An alternative option is known as sparse optical flow. This algorithm specifies the points to be tracked beforehand and then optical flow vectors are computed between the matching points in frames. Such vectors also associates the velocities of the tracked points. If those points are trustworthy, then the tracking process would be relatively more robust and reliable, with less calculation. [17] For this reason, in this thesis work, we try to use sparse optical flow first.

1.2.6.1 Sparse optical flow

Shi and Tomasi corner detector [31] is used to finds the important features (corners) in two consecutive frames. This algorithm gets the corners which are more stable for tracking the object and from the corners, velocity vectors are computed. The result of such method is shown in figure 1.6 and 1.7.

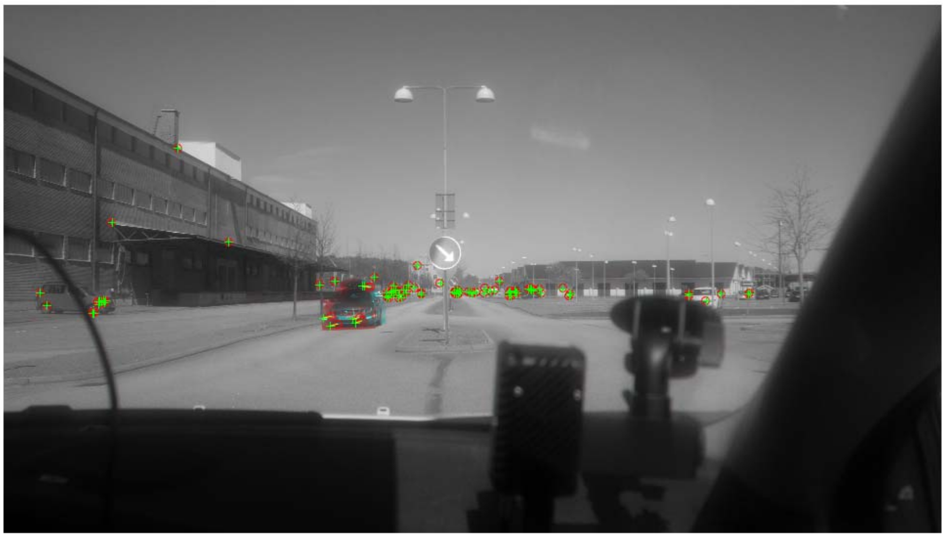


Figure 1.6 Sparse optical flow detection

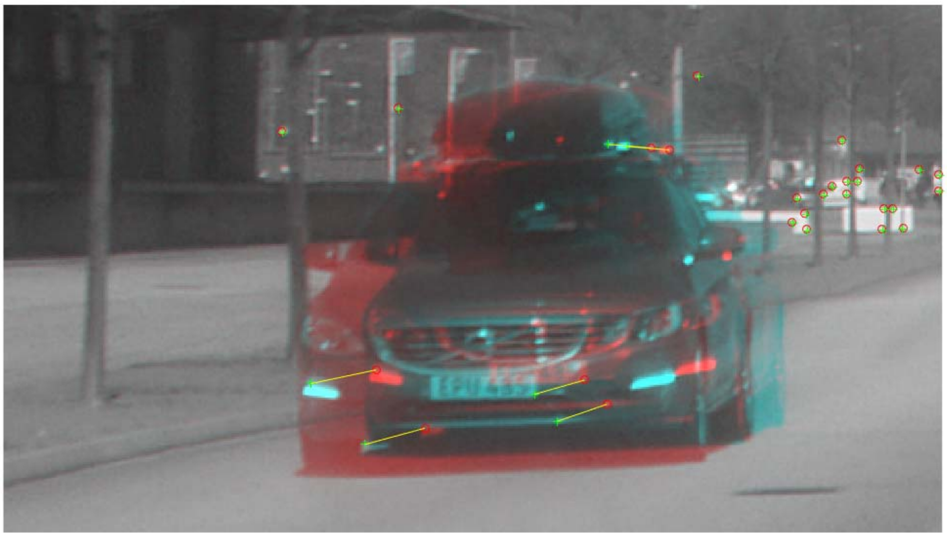


Figure 1.7 Sparse optical flow detection (zoom in)

The Shi and Tomasi detector is modified from the Harris detector [31]. Intuitively, corners are the junctions of contours. They are more stable features over changes of frames. Also, there are always large variations in the neighborhood of the corner points in all directions. Obviously, we can recognize the corner point by looking at intensity values within a small window $w(x, y)$. At corners, shifting the window should yield a significant change at any direction. With a shift of (u, v), the change of intensity is given by

$$E(u, v) = \sum_{x,y} w(x, y)\, [I(x + u, y + v) - I(x, y)]^2 \tag{1.3}$$

where I is the intensity of a position in image. For very distinctive patches, E would be large. By taking the second derivative of E around (u,v)=(0,0), Harris matrix is found and as following,

$$A = \begin{vmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{vmatrix} \tag{1.4}$$

The two eigenvalues of A can be used to detect the corner. One corner can be found if the eigenvalues are distinct large values. As the improvement of Harris detector, Shi and Tomasi corner detector use affine transformation,

$$E(u, v) = \sum_{x,y} w(x, y)\, [I(x, y) - I(A(x, y) - (u, v))]^2 \tag{1.5}$$

instead of simple transformation. Since affine transformation have six variable, we have a 6 by 6 detection matrix. Similarly, if the matrix have large eigenvalues, we can find corners [29].

The next step is to match the same feature point in two frames and generate the velocity vectors as the tracking process. Lucas Kanade algorithm [34] is used to produce the results. For given points (u, v) in frame I_t, the algorithm finds the points $(u + d_x, v + d_y)$ in frame I_{t+1} that minimizes e,

$$e\left(d_x, d_y\right) = \sum_{x=u-w}^{u+w} \sum_{y=v-w}^{v+w} (I(x, y) - I(x + d_x, y + d_y)) \tag{1.6}$$

1.2.6.2 Dense optical flow

Given videos which has low quality, especially low resolution, Shi and Tomasi corner detector cannot detect the features points desirably. In this case, dense optical flow can be considered as an alternative. Different methods have been created to compute the dense optical flow and in this thesis, Farneback algorithm [26] is implemented. It is based on the well-known Yosemite sequence, which shows a relatively higher accuracy than other algorithms [27][37]. Also, this algorithm can be accomplished by means of separable convolution, which makes it fast. [27] Figure 1.8 and 1.9 illustrate the result of this method.

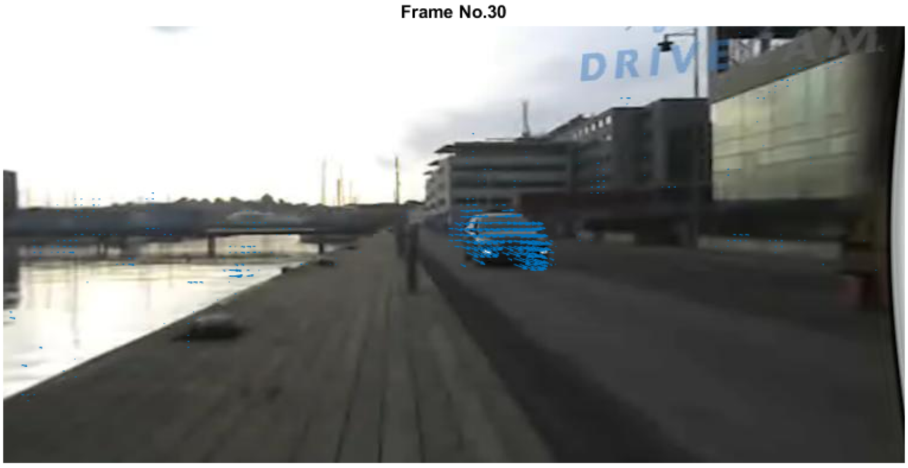


Figure 1.8 Dense optical flow detection

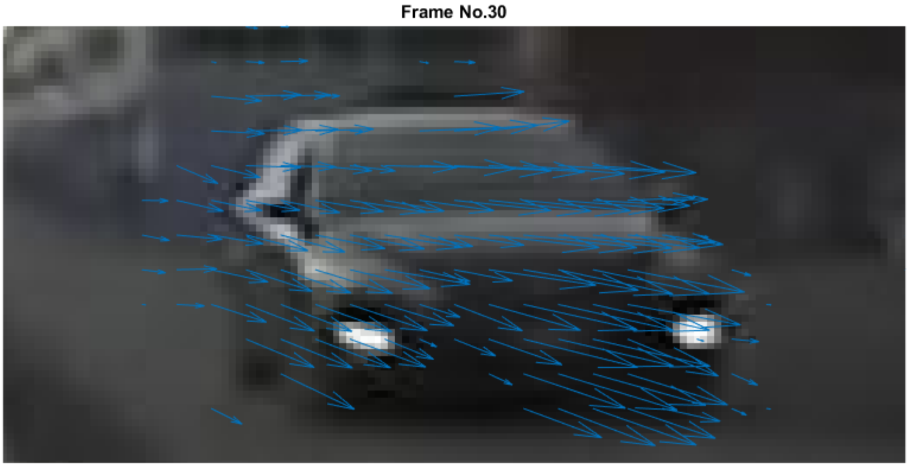


Figure 1.9 Dense optical flow detection (zoom in)

The algorithm starts by calculating 3D orientation tensors from the image sequence, described in Farneback's paper **[5]**. According to the model, the signal is projected as following:

$$f(\boldsymbol{x}) \sim \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} + \boldsymbol{b}^T \boldsymbol{x} + c \tag{1.7}$$

A, b and c are obtained from a weighted least squares approximation (normally as Gaussian) of the signal. From the parameters above, the orientation tensor is obtained by:

$$\boldsymbol{T} = \boldsymbol{A}\boldsymbol{A}^T + \gamma \boldsymbol{b}\boldsymbol{b}^{\boldsymbol{T}} \tag{1.8}$$

where γ is a non-negative weight factor between the even and the odd parts of the signal. Assuming the velocities over a region can be parameterized based on some motion model (affine motion model is used in this method), we can use all the tensors in the region to get the parameters. A 2D velocity vector $(v_x, v_y)^T$, measured in pixels per frame, can be presented with the affine motion model $v_x(x, y) = ax + by + c, v_y(x, y) = dx + ey + f$, where x and y are image coordinates. Such 2D vector can be extended to 3D as

$$\boldsymbol{v} = \boldsymbol{S}\boldsymbol{p} \tag{1.9}$$

where $\boldsymbol{S} = \begin{pmatrix} x & y & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$ and $\boldsymbol{p} = (a\ b\ c\ d\ e\ f\ 1)^T$.

In order to obtain $\boldsymbol{p}$, the cost functions $d(\boldsymbol{v}, \boldsymbol{T}) = \boldsymbol{v^T T v}$ and $d_{tot} = \sum_i d(\boldsymbol{v_i}, \boldsymbol{T_i})$ should be minimized. This function can be also written as following:

$$d_{tot}(\boldsymbol{p}) = \sum_i \boldsymbol{p}^T \boldsymbol{S}_i^T \boldsymbol{T}_i \boldsymbol{S}_i \boldsymbol{p} = \boldsymbol{p}^T \boldsymbol{Q}_{tot} \boldsymbol{p} \tag{1.10}$$

Moreover, instead of computing regions with a consecutive motion, we can obtain a motion model surrounding each point, by replacing the function to

$$d_{tot}(\boldsymbol{p}) = \sum_i w_i d(\boldsymbol{v}_i, \boldsymbol{T}_i) = \sum_i w_i \boldsymbol{p}^T \boldsymbol{S}_i^T \boldsymbol{T}_i \boldsymbol{S}_i \boldsymbol{p} = \boldsymbol{p}^T \boldsymbol{Q}_{tot} \boldsymbol{p} \tag{1.11}$$

where the summation contains the neighbor points of the current point and the weights w_i are given by Gaussian.

1.3 Overall aim and scope

The overall scope of this Master's thesis is twofold: first it aims to develop a tool that can extract the key kinematics and geometric information of an instrumented turning vehicle (subject vehicle; SV) and an oncoming principle other vehicle (POV) from EuroFOT（http://www.eurofot-ip.eu/）and Lytx (www.lytx.com) video data, through a combination of manual annotation and image processing algorithms. Second, it aims to use the extracted information to perform a quantitative analysis of driver behavior (PET) for LTAP/OD scenarios extracted from the EuroFOT dataset.

To achieve the overall scope of the thesis, the following steps have been considered and will be described in the Methods and Results sections:

1) Calibration of video camera and enhancement of image quality.
2) Preparation of Matlab and OpenCV scripts about optical flow and coordinate transformation method in order to get key kinematics and geometric variables.
3) Evaluation of the quality of the method by calibration experiment and 2D LTAP/OD scenarios reconstructions.
4) Description of basic statistic about LTAP/OD scenarios in EuroFOT dataset.
5) Quantitative analysis of retrieved data and metrics to understand how drivers turn left after giving way to oncoming vehicles.

2 Methodology

2.1 Camera calibration

As illustrated in previous chapter, the Matlab tool-Single Camera Calibration App [19] was used to obtain the intrinsic parameters for EuroFOT camera and Lytx camera. The intrinsic matrix of EuroFOT camera is reported in (2.1):

$$\mathrm{M} = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 255.82 & 0 & 179.39 \\ 0 & 280.99 & 143.19 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

On the other hand, the calibration of the Lytx camera used in experiment is reported in (2.2):

$$\mathrm{M} = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 534.75 & 0 & 313.90 \\ 0 & 522.99 & 174.68 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

By applying the calibration in chapter 1.2.4, undistorted images can be obtained. As can been seen from Figure 2.1, the white curved arrow at the bottom left corner of the left image is significantly corrected to the straight one(see red circles in Figure 2.1). This calibration can improve the accuracy of the process to extract the vehicle width from the image.

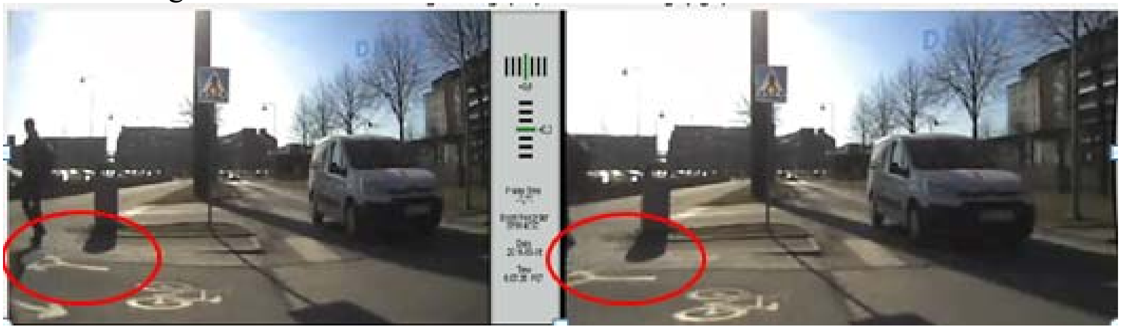


Figure 2.1 Calibration example for Lytx camera

2.2 Manual tool development

2.2.1 Coordinate transformation

Figure 2.2 shows the camera coordinate system and world coordinate system.

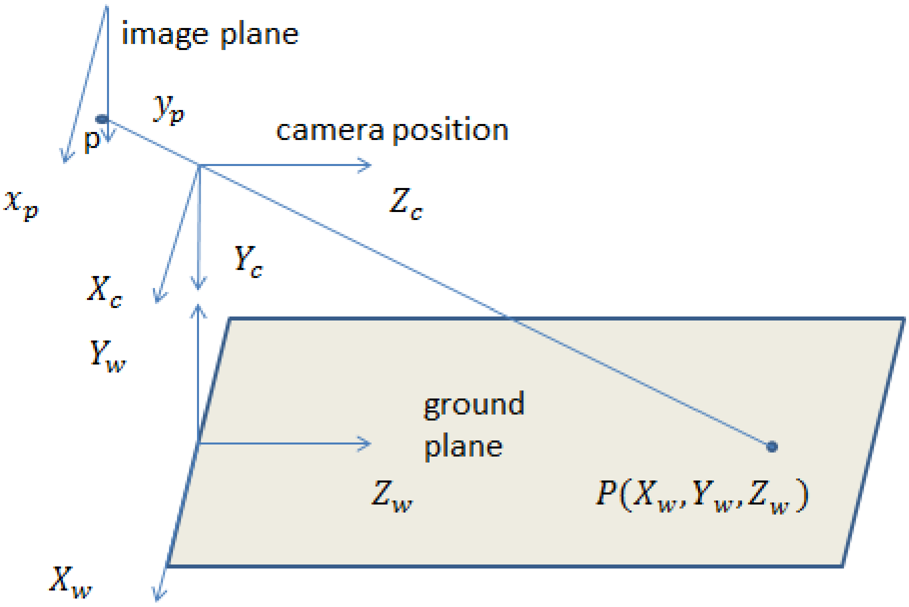


Figure 2.2 Coordinate system

The projection from the tri-dimensional world (P (X_w, Y_w, Z_w)) to the bi-dimensional images (p (x_p, y_p)) can be expressed by the equation reported in [17] and detailed in (2.3):

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = S * M * [R, T] * \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{2.3}$$

In the equation, M, R and T are the 3×3intrinsic calibration matrix, the 3×3 rotation matrix and the 3×1 translation vector respectively. S is an arbitrary scale factor which aims to make explicit that the homography is defined only up to this factor. And conventionally, it is factored out of other matrix.

In the rotation matrix, three angles θ, φ and α rotated around the X_c, Y_c, and $Z_c$axes in sequence are considered. In order to simplify the equation 2.3, φ and α are set to zero. The reason why φ can be set to zero is because the coordinate system can move together with the camera and α can be negligible with the proper camera set up. Moreover, in [21], it was found that θ has the most significant impact on distance estimation compared to other extrinsic parameters. The translation matrix is fully determined by the camera height-H, which means T=(0,H,0)^T. Thus, considering the pitch angle θ and camera height and all intrinsic parameters, the image coordinates p $(x_p, y_p)^T$where a world point P $(X_w, Y_w, Z_w)^T$projects are defined by:

$$x_p = C_x + \frac{f_x X_w}{Z\cos(\theta) + (-h + Y)\sin(\theta)} \tag{2.4}$$

$$y_p = C_y + \frac{f_y\big((h - Y_w)\cos(\theta) + Z_w sin(\theta)\big)}{Z_w\cos(\theta) + (-h + Y_w)\sin(\theta)} \tag{2.5}$$

2.2.2 Range estimation from ground points

2.2.2.1 Basic formulae

Considering equations 2.4 and 2.5, a point on the flat road has a coordinate $(X_w, 0, Z_w)^T$, then a point in 2D image can be translated to 3D world on the road plane by the following equation:

$$X_w = \frac{f_y H(C_x - x_p)}{f_x(C_y - y_p)\cos(\theta) + f_x f_y \sin(\theta)} \tag{2.6}$$

$$Z_w = \frac{H\left((C_y - y_p)\sin(\theta) - f_y \cos(\theta)\right)}{(C_y - y_p)\cos(\theta) + f_y \sin(\theta)} \tag{2.7}$$

In principle, these two equations can be used to estimated range and lateral offset.

2.2.2.2 Theoretical error analysis

As can be seen from equation 2.6 and 2.7, the range between the camera and a selected point on road surface is a function of the pitch angle θ. To estimate the effect of pitch angle on the measurement of range and offset, different values of θ were considered. Figure 2.3 represents the changes in the measurement of the range whereas Figure 2.4 represents the changes in the measurement of the offset.

The scenario of this trial experiment considers the SV standing still and the POV coming from the opposite direction of SV at a constant speed around 20km/h. In this driving situation, comparing two lines where pitch angle equals -0.9 degree and -1,.3degree in Figure 2.3, and a 0.4 degree pitch angle error can lead to 8m error at around 35 m distance while lead to less than 2m error at 20m. As for the velocity estimation, the slopes of these two lines change from 20.7 km/h to 25.2km/h with the consideration that the time interval between two continuous frames is 0.25s. As for the offset error, a 0.4 degree pitch angle error can cause less than 0.5 m at around 2 meters lateral offset. Therefore, the estimation of range and speed need to be improved by increasing the accuracy of pitch angle.

Furthermore, theoretically each line should be a straight line since POV has a constant speed. So the fluctuation of each line shows errors caused by manual selection. The range errors here are always less than 1m and the offset errors are often less than 0.15m throughout the whole trip. So, with stable manual selection, this method can be used to obtain range and offset.

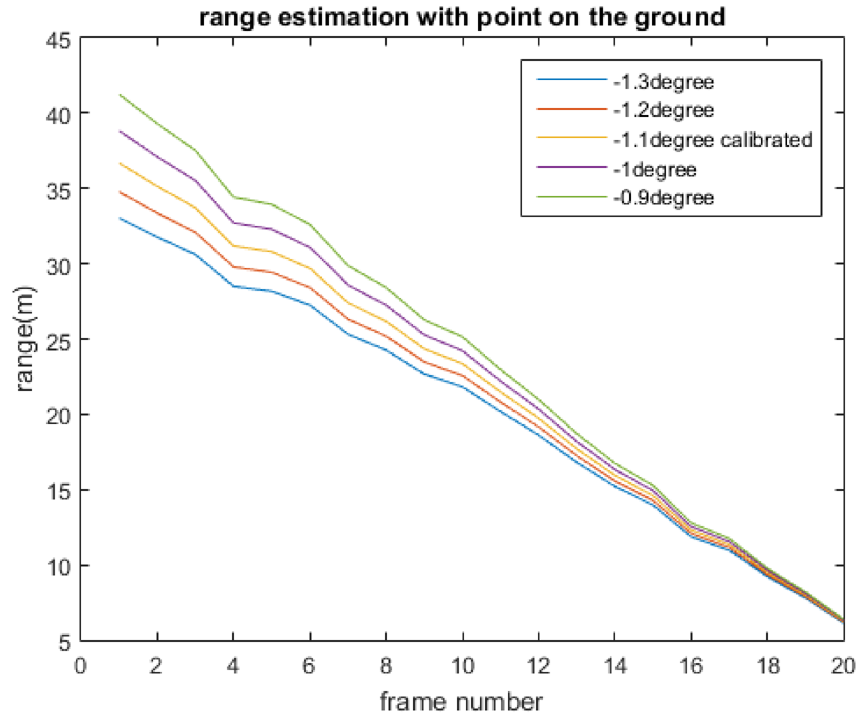


Figure 2.3 Range estimation result from experiment set up

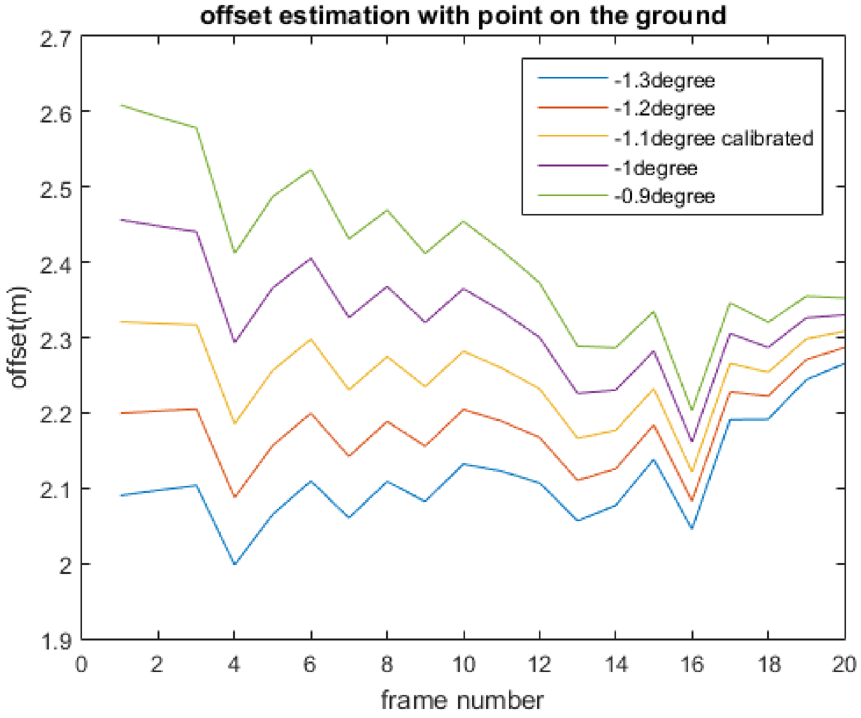


Figure 2.4 Offset estimation result from experiment set up

2.2.2.3 Improvement of ground points method

To have an accurate estimation of the range and the offset, a precise measurement of the pitch angle is required. Unfortunately, this angle can vary due to uneven ground and to the movement of ego-vehicle, especially with large acceleration. Two methods will be described in this thesis to obtain a real time estimation of the pitch angle.

The first method is defined as the vanishing point method, and consider 3D points on the ground at infinity positions whose coordinates in world coordinate are $(X, 0, \infty)^T$. According to equation 2.5, the y coordinate of the horizontal line Figure 2.5 is obtained as 2.8.

$$y_h = C_y + f_y \tan(\theta) \tag{2.8}$$

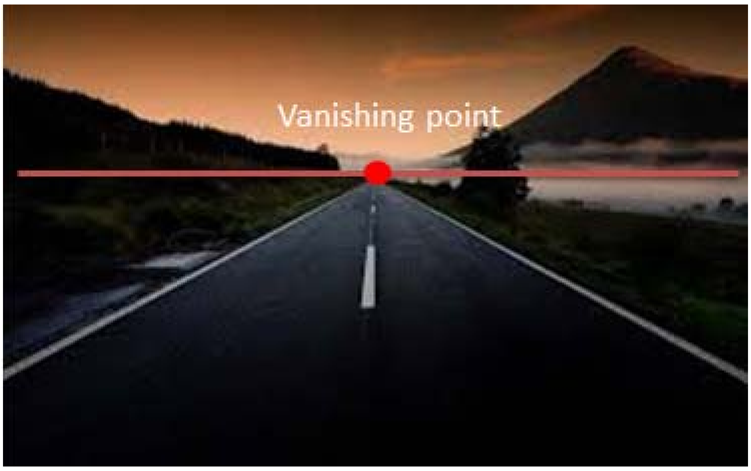
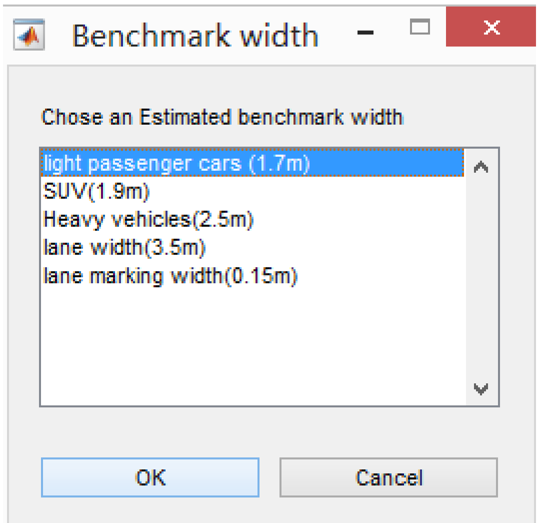


Figure 2.5 Vanishing point and horizontal line

If at least two parallel lane markings are clear enough and could be detected automatically, then the intersection point of these lines in the image is the vanishing point whose vertical coordinate is y_h. Through the calculation of y_h,the pitch angle is obtained.

Unfortunately, many scenarios in the EuroFOT data set do not allow detecting two parallel lines due to non-ideal infrastructures. For this reason, another method is introduced to improve the pitch angle estimation. This method compares the width calculated with the benchmark obtained by considering one of the vehicle types or other standard road markings shown in Figure 2.6.


Benchmark width

Chose an Estimated benchmark width

light passenger cars (1.7m)
SUV(1.9m)
Heavy vehicles(2.5m)
lane width(3.5m)
lane marking width(0.15m)

Figure 2.6 Benchmark width used in the tool

Figure 2.6 shows the process adopted to estimate the pitch angle. The real width of the vehicle is defined by the benchmark and the pitch angle is estimated by comparing the vehicle width selected in Figure 2.6 (red line in Figure 2.7) and the estimated width with different pitch angles (blue line in Figure 2.7). Using more than one benchmark to get a mean value of pitch angle can improve the process.

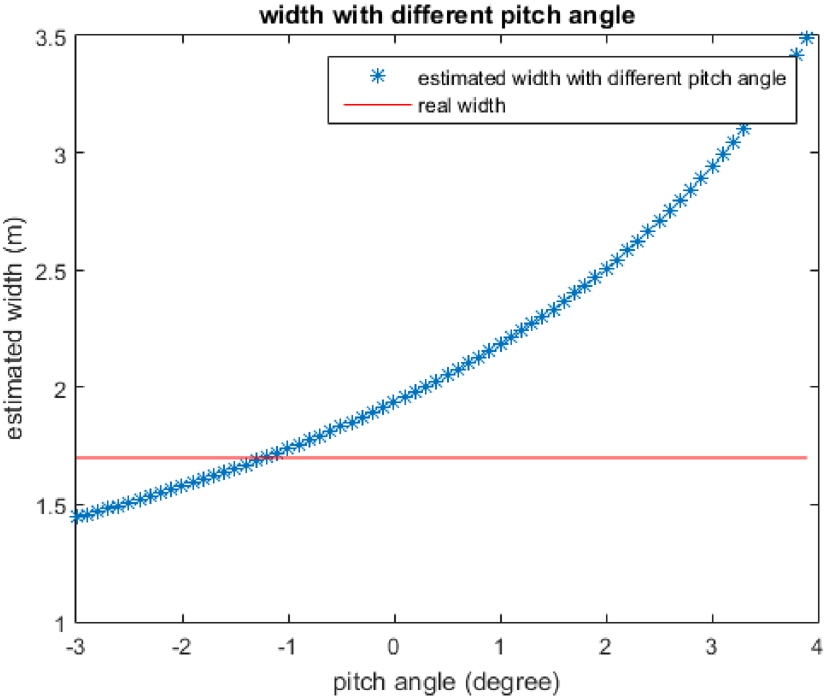


Figure 2.7 Pitch angle calibration with benchmark

2.2.3 Range estimation with vehicle width

2.2.3.1 Basic formulae

According to equation 2.5, supposing the actual vehicle width (W) is known, then the range (Z) between camera and oncoming vehicle can be estimated as follow:

$$\mathrm{Z} = \frac{f_x W}{w_r * \cos(\theta)} + h * \tan(\theta) \tag{2.9}$$

Figure 2.8 shows the result of trial experiment mentioned in 2.2.2.2, range is calculated with different pitch angle according to equation 2.9. Small range variations due to the changes of pitch angle can be ignored, so this method is quite robust.

From equation 2.9, since pitch angle doesn't have a significant influence on the range estimation, the main range errors are due to the actual vehicle width estimation and the vehicle width in pixels.

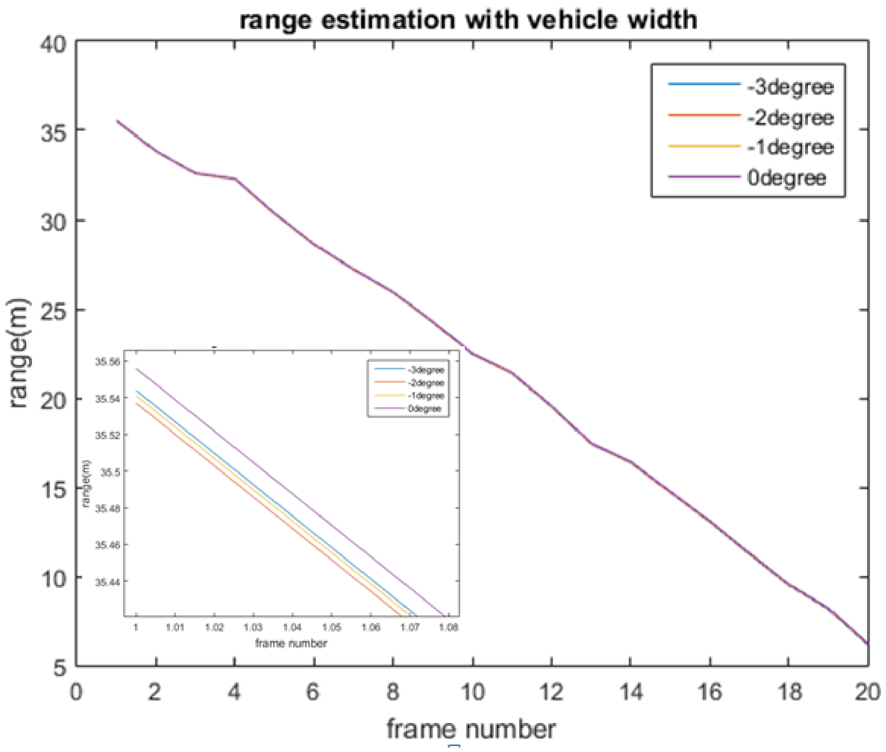


Figure 2.8 Range estimation with different pitch angle.

2.2.3.2 Range Error from real vehicle width estimation

Unfortunately, the actual width of the vehicles in the videos is hard to know. According to the width data of different brand (Table 2.1), since most values are in the range $1.7m \pm 0.1m$ and $1.9m \pm 0.1m$, three width values of different types of vehicles were defined (Table 2.2).

Table 2.1 vehicle dimensions[33]

Make	Model	Length	Width	Height
Volvo	XC90	4950 mm	2008 mm	1776 mm
Tesla	Model S	4970 mm	1964 mm	1445 mm
Volvo	XC60	4627 mm	1891 mm	1713 mm
BMW	X3	4657 mm	1881 mm	1678 mm
Volvo	S60	4628 mm	1865 mm	1484 mm
Volvo	V60	4628 mm	1865 mm	1484 mm
Toyota	RAV4	4605 mm	1845 mm	1675 mm
Renault	Scenic Xmod	4372 mm	1845 mm	1656 mm
Audi	A4	4726 mm	1842 mm	1427 mm
Peugeot	3008	4365 mm	1837 mm	1635 mm
Volkswagen	Passat	4767 mm	1832 mm	1456 mm
Audi	Q3	4388 mm	1831 mm	1608 mm
Ford	C-MAX	4379 mm	1828 mm	1610 mm
Ford	Focus	4360 mm	1823 mm	1469 mm
BMW	3	4633 mm	1811 mm	1429 mm
BMW	3 Touring	4633 mm	1811 mm	1429 mm
Mercedes-Benz	C Estate	4702 mm	1810 mm	1457 mm
Renault	Fluence	4622 mm	1809 mm	1479 mm
Volvo	V40	4369 mm	1802 mm	1445 mm
Nissan	Qashqai	4379 mm	1800 mm	1590 mm

Volkswagen	Golf	4255 mm	1790 mm	1452 mm
Honda	Civic	4370 mm	1770 mm	1470 mm
Nissan	Leaf	4445 mm	1770 mm	1550 mm
Smart	fortwo	2695 mm	1663 mm	1555 mm

Table 2.2 Estimated vehicle width in the tool

Vehicle type	Estimated width (m)
Light passenger cars	1.7
SUV (Sport utility vehicle)	1.9
Heave vehicles (e.g. buses, trucks, …)	2.5

By approximating the real vehicle width with one of the values reported in Table 2.2, an error is introduced in the calculation of the range. Assuming that the width of all SUV is estimated as 1.9m (W_{est}), and the real width of one SUV is 2m (W), the error (m = W $- W_{est}$) is 0.1 m. The examples of these two vehicles can be Volvo XC 60 whose width is 1.891m and Volvo XC 90 with 2.003m width.

Then the range error caused by the 0.1 m error in real width is calculated as：

$$Z_{err1} = Z - Z_{est} = \frac{f_x W}{w_r * cos(\theta)} + h * tan(\theta) - \frac{f_x W_{est}}{w_r * cos(\theta)} + h * tan(\theta)$$

$$= (W - W_{est}) \frac{f_x}{w_r * cos(\theta)} \qquad (2.10)$$

The corresponding percentage error is:

$$P_{err1} = \frac{Z_{err1}}{Z} \approx \frac{(\mathrm{W} - W_{est})}{W} = \frac{0.1}{2} = 5\% \qquad (2.11)$$

Thus theoretically, the percentage error P_{err1} increases linearly with the real width error (W $- W_{est}$) and the range error Z_{err1} also has a linear relationship with the range (Z) when one specific vehicle is tested. As an example, if the actual vehicle width is 0.1m larger than the estimated one and the real range Z is 40 m, the calculated range is about 2m smaller.

2.2.3.3 Range Error from vehicle width in pixels estimation

Another theoretical error comes from the n pixels error of vehicle width (w_r) estimation in an image which can be expressed as follow:

$$Z_{err2} = Z - Z_{est} = \frac{f_x W}{w_r * \cos(\theta)} + h * \tan(\theta) - \frac{f_x W}{w_{rest} * \cos(\theta)} + h * \tan(\theta)$$

$$= \frac{(w_{rest} - w_r) f_x W}{w_r * w_{rest} * \cos(\theta)} \qquad (2.12)$$

$$P_{err2} = \frac{Z_{err2}}{Z} \approx \frac{(w_{rest} - w_r)}{w_{rest}} = -\frac{n}{w_{rest}} \qquad (2.13)$$

Since the relationship between n and w_{rest} is not clear, equation 2.14 can be obtained from the equation 2.12 as:

$$Z_{err2} = -\frac{n}{w_{rest}} * Z = -\frac{n}{w_r - n} * Z \approx -\frac{n}{\frac{f_x W}{Z} - n} * Z$$

$$= -\frac{nZ^2 * \cos(\theta)}{f_x W - nZ * \cos(\theta)} \tag{2.14}$$

Typically n$\approx$ 1, and $f_x W \gg nZcos(\theta)$, so

$$Z_{err2} \approx -\frac{nZ^2 * \cos(\theta)}{f_x W} \tag{2.15}$$

Notes:

1. If the pixel errors n $(w_r - w_{rest})$can be controlled and regarded as a constant value, then this kind of range error increases quadratically with the range. In practice, the pixel errors n shows a slight upward trend with the vehicle width in pixels. However the maximum pixel error is less than 1.8 pixels by manual selection with a range around 5 meters with Lytx camera.
2. Lytx camera with a larger f_x are better than euroFOT camera in terms of reducing this error.

Taking the same example used in 2.2.2.2, when the range Z is around 35m, w_r equals to 24.4 pixels which is the mean value of 10 times manual measurement and the maximum measurement error is 1.2 pixels larger(see Table 2.3), then the Z_{err2} will be 1.64 m.

Table 2.3The same vehicle width in pixels is measured 10 times on one image from Lytx camera. The distance between camera and POV is around 35m.

Times	vehicle width (pixels)
1	23.8
2	24.9
3	24.5
4	25.2
5	23.5
6	23.4
7	23.6
8	24.3
9	25.1
10	25.6

2.2.4 Range rate estimation from scale change

2.2.4.1 Basic formulae

In some vehicular applications like platooning, it is important to estimate the range rate or relative velocity between two vehicles. In layman's term, it is not important to know whether the target vehicle is 35m or 38m away, but whether the difference in distance and speed maintained by the two vehicles. Thus, discussing range rate error is worthwhile.

Let w_r and w_r' be the vehicle width in pixels in two images at two close time points when the target vehicle is at distances Z and Z' respectively.Then the relative vehicle speed in infinitesimal time Δt can be calculated as follow ::

$$v = -\frac{Z' - Z}{\Delta t} = \frac{Z}{\Delta t}\frac{w_r{}' - w_r}{w_r{}'} \tag{2.16}$$

As set by this definition, when two vehicles approach, the relative speed is positive.

2.2.4.2 Range rate error from scale change

Defining scale change S and alignment error A_{err} as follow :

$$S = \frac{w_r{}' - w_r}{w_r{}'} \tag{2.17}$$

$$A_{err} = (w_r{}' - w_r)_{real} - (w_r{}' - w_r)_{est} = (w_r' - w_{rest}') - (w_r - w_{rest}) \tag{2.18}$$

The alignment error is the difference between the real value and the value from measurement of the changing pixels of vehicle width in two images. So the scale error S_{err} can be defined as the alignment error A_{err} divided by vehicle image width:

$$S_{err} = \frac{A_{err}}{w_r{}'} = \frac{A_{err} Z' * \cos(\theta)}{f_x W} \tag{2.19}$$

Suppose the range Z' is accurate, and then the relative speed error from scale change is:

$$v_{err1} = \frac{Z}{\Delta t} S_{err} = \frac{Z}{\Delta t}\frac{A_{err}}{w_r} = \frac{Z * Z'}{\Delta t}\frac{A_{err} * \cos(\theta)}{f_x W} \tag{2.20}$$

Equation 2.20 has some implications:

1. If the Δt is small enough, Z roughly equals to Z'. Then this equation means the velocity error from scale change increases with the range squared as long as the alignment error can be estimated within a certain value.
2. This relative velocity error calculated is independent of the relative velocity.
3. This relative velocity error is inversely proportional to the time window Δt. Selecting two images with long time interval can reduce the error.
4. A narrow field of view camera with a large f_x can increase the accuracy of the relative velocity estimation which means Lytx camera is better than the EuroFOT camera.

As an Example: if the measurement results of Table 2.3 are used here and the worst case is considered, then the vehicle width value in the first image is 1.2 pixels larger than the average one while the second one is 1 pixel smaller. Thus the alignment error is 2.2 pixels and the velocity will be estimated 10.55m/s larger when Δt= 0.25s and Z is 35m . This error on the speed is not acceptable and, therefore, several methods are introduced to improve the estimation.

.

2.2.4.3 Range rate error from range estimation

As discussed in 2.1.4 , the main range error comes from the error of real target vehicle width estimation (Z_{err1}), and the error of n pixels estimated of vehicle length.(Z_{err2}).

$$v_{err2} = \frac{Z_{err1} + Z_{err2}}{\Delta t} * S = \frac{m}{W} * \frac{Z * S}{\Delta t} + \frac{nZ * \cos(\theta)}{f_x W} * \frac{Z * S}{\Delta t}$$
$$= \left(\frac{m}{W} + \frac{nZ * \cos(\theta)}{f_x W}\right) * v \tag{2.21}$$

Clearly, this error increase with the range and relative velocity.

As an example, the parameter in equation 2.21 are set the same values in range error section 2.2.3, the v_{err2} is 0.59 m/s

2.2.4.4 Range rate error from relative acceleration

From the implications of equation 2.20, increasing the time window Δt which is the guidance of average filter design can improve the accuracy of velocity estimation, however, computing velocity using finite differencing equation 2.16 is not accurate with large Δt since it is no longer infinitesimal motion and what we obtain with that equation is average relative velocity.

In case of a constant relative acceleration a, the range and velocity are:

$$Z' = Z - \frac{1}{2}a(\Delta t)^2 - v\Delta t \tag{2.22}$$

$$v = -\frac{Z' - Z}{\Delta t} = \frac{1}{2}a\Delta t + v \tag{2.23}$$

So another error term is:

$$v_{err3} = v - v_{est} = -\frac{1}{2}a\Delta t \tag{2.24}$$

Thus a large Δt and the relative acceleration increase the inaccuracy of relative speed estimation.

As an example: if the oncoming vehicle accelerates to approach the standstill SV with the acceleration equals to 2m/s^2, then the estimated relative velocity is 0.25 m/s larger when the time interval Δt equals 0.25s.

2.2.4.5 Summary of range rate error and improvement

In sum, the range rate error comes from three parts: scale change, range error and relative acceleration. If the small effect of pitch angle is ignored, the total error can be expressed as 2.25:

$$\begin{aligned} v_{err} &= v - v_{est} = v_{err1} + v_{err2} + v_{err3} \\ &= \frac{Z * Z'}{\Delta t}\frac{A_{err}}{f_x W} + \left(\frac{m}{W} + \frac{nZ}{f_x W}\right) * v - \frac{1}{2}a\Delta t \end{aligned} \tag{2.25}$$

The implications of equation 2.25 are the following:

1. The range rate error v_{err} can be positive or negative depending on alignment error A_{err},actual vehicle width error m, width error in pixels n and relative acceleration a. In particular, the range rate error increases towards positive values if these parameters follow the rules in Table 2.3:

Table 2.3 Parameter directions and meaning in equation 2.25

parameters	direction	meaning
A_{err}	+	real difference of width pixels in two images is larger than the measurement one.
m	+	real vehicle width is larger than the estimated width.
n	+	real vehicle width in pixels is larger than the measurement value on the screen..
a	-	The relative speed between to vehicles decreases.

2. From the three examples mentioned in 2.2.4.2, 2.2.4.3 and 2.2.4.4 - which show the value of errors from these three parts - v_{err1} is the largest error and needs improvement. One method is to draw lines (Figure 2.8) that can be the predicted

trajectory of oncoming vehicle to support manually selection. The starting points are the position selected in last image and the new points should be selected around these two lines with almost a same offset. With this improvement, the alignment error A_{err} can be reduced to less than 1 pixel and the v_{err1} at 35 m can be reduced to 4.8 m/s. This value still needs further improvement. For this reason, the Kalman filter is introduced in next section.

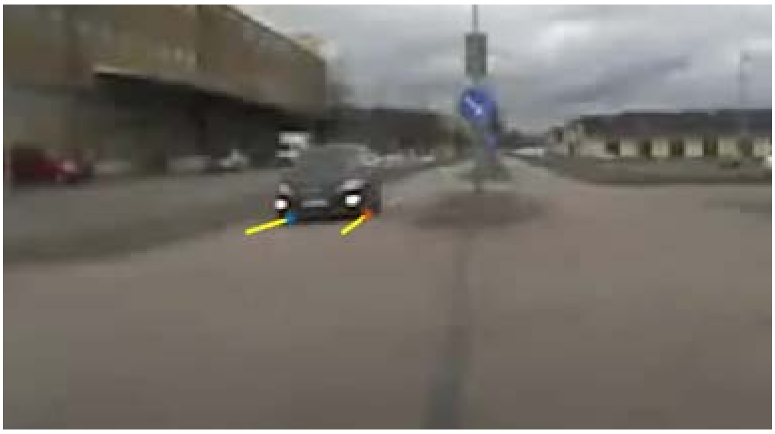


Figure 2.8 Support lines for manual selection of vehicle width

3. There are two terms depending on Δt and Δt is a good indicator to design the moving average filter and decide which time interval between two images are used to estimate a more accurate relative velocity. Considering two terms with Δt, the error of these two terms is :

$$v_{err\Delta t} = \frac{Z * Z'}{\Delta t}\frac{A_{err}}{f_x W} - \frac{1}{2}a\Delta t = \frac{Z * \left(Z - \frac{1}{2}a(\Delta t)^2 - v\Delta t\right)}{\Delta t}\frac{|A_{err}|}{f_x W} - \frac{1}{2}a\Delta t$$
$$= \frac{Z^2}{\Delta t}\frac{A_{err}}{f_x W} - \frac{1}{2}Za\Delta t\frac{A_{err}}{f_x W} - Zv\frac{A_{err}}{f_x W} - \frac{1}{2}a\Delta t \qquad (2.26)$$

In worst case, the possible largest value of $v_{err\Delta t}\ is$:

$$max(\ v_{err\Delta t}) = \frac{Z^2}{\Delta t}\frac{|A_{err}|}{f_x W} + \frac{1}{2}Z|a|\Delta t\frac{|A_{err}|}{f_x W} + Z|v|\frac{|A_{err}|}{f_x W} + \frac{1}{2}|a|\Delta t \qquad (2.27)$$

Then the optimal Δt that can decrease this max $v_{err\Delta t}$ can be calculated by differentiating eq.2.27 and setting the new equation to zero:

$$-\frac{Z^2}{\Delta t^2}\frac{|A_{err}|}{f_x W} + \frac{1}{2}Z|a|\frac{|A_{err}|}{f_x W} + \frac{1}{2}|a| = 0 \qquad (2.28)$$

Now the Δt can be solved with formula (2.29):

$$\Delta t = \sqrt{\frac{2 * Z^2|A_{err}|}{Z|a||A_{err}| + f_x W|a|}} \qquad (2.29)$$

For zero relative acceleration, the optimal Δt is infinity and in practice this value can be limited to a value 2s.

2.2.5 Range rate estimation from optical flow

With the points selected manually, optical flows can be created by using the coordinates of these points on continuous images. The details of how to create this benchmark values and how to match practical values with theoretical ones will be introduced in 2.4.

The difference between manual optical flow and automated optical flow is that the position of tracking point is clear in the manual method. Thus a more accurate offset relative to the camera position can be set. The drawback is that only one point is tracked and the fluctuation of manual selection will influence the match which is the main error of this method.

2.3 Kalman filter

2.3.1 Basic equations

Since the range and range rate information comes from different methods, Kalman filter can be used to integrate these information with several update equations. The predicted and updated equations for discrete Kalman filter are reported in the equations below:

Predicted equations:

$$\hat{X}_{t|t-1} = F_t\hat{X}_{t-1|t-1} + B_t u_t \tag{2.30}$$
$$P_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_t \tag{2.31}$$

Updated equations:

$$K_t = P_{t|t-1} H_t^T \left(H_t P_{t|t-1} H_t^T + R_t\right)^{-1} \tag{2.32}$$
$$\hat{X}_{t|t} = \hat{X}_{t|t-1} + K_t(Y_t - H_t\hat{X}_{t|t-1}) \tag{2.33}$$
$$P_{t|t} = (\mathrm{I} - K_t H_t)P_{t|t-1} \tag{2.24}$$

Where:
$\hat{X}$: Estimated state.
F:State transition matrix (i.e., transition between states).
u: Control variables.
B: Control matrix (i.e., mapping control to state variables).
P: State variance matrix (i.e., error of estimation).
Q: Process variance matrix (i.e., error due to process).
Y: Measurement variables.
H: Measurement matrix (i.e., mapping measurements onto state).
K: Kalman gain.
R: Measurement variance matrix (i.e., error from measurements).
Subscripts are as follows:
t|t: current time period.
t−1|t–1: previous time period.
t|t−1: intermediate steps.

The Kalman filter optimizes the performance (optimal weighting of input) by assuming pre-defined models of a system and the following parts describe how to get models and those parameters in the equations.

2.3.2 Model the state process

A constant acceleration model is applied in the thesis, although in real situations the relative acceleration can change. The method still works effectively and the accuracy can be refined by trusting different measurement models.

The constant acceleration model in the discrete Kalman filter is as follow :

$$z_{t|t-1} = z_{t-1|t-1} + (-v_{t-1|t-1}\Delta t - \frac{1}{2}\mathrm{a}_{t-1|t-1}(\Delta t)^2)$$

$$v_{t|t-1} = v_{t-1|t-1} + \mathrm{a}_{t-1|t-1}\Delta t \tag{2.25}$$

$$\mathrm{a}_{t|t-1} = \mathrm{a}_{t-1|t-1}$$

The directions of speed and acceleration is the same as mentioned in 2.2.4. When the two vehicles come close to each other, the relative speed is positive and the relative acceleration is positive if the acceleration brings the vehicles close to each other in a shorter time. In the formulae, Δt is the sampling time.

So considering the predict equations from 2.2.1:

$$\hat{X}_{t|t-1} = \begin{bmatrix} z_{t|t-1} \\ v_{t|t-1} \\ \mathrm{a}_{t|t-1} \end{bmatrix} \tag{2.26}$$

$$F_t = \begin{bmatrix} 1 & -\Delta t & -\frac{1}{2}(\Delta t)^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{2.27}$$

Control variables B and u are not used and both of them are set to 0.

2.3.3 Model the measurement process

In the thesis, the direct measurement is the coordinate of ground points and the vehicle width in pixels. With the first data source, relative range is obtained through coordinate transformation and relative speed is obtained from optical flow method. With the second measurement, relative range and relative speed are obtained, however, as discussed in 2.24, relative speed is not accurate enough to be directly used in Kalman filter. Thus the update equation will be utilized three times and each update results are used for the next update and the final update results are used in the predict equations.

The first update will use range estimation $z_1$from ground points:

$$\mathrm{Y}_1 = z_1$$
$$\mathrm{H}_{1=}[1\ 0\ 0]$$

The second updates will use range estimation z_2 from vehicle width:

$$\mathrm{Y}_2 = z_2$$
$$\mathrm{H}_{2=}[1\ 0\ 0]$$

With these two updates, a more accurate relative range can be achieved and with this new data, the range rate estimation can be improved.

The final updates will use relative velocity estimation v from optical flow:

$$\mathrm{Y}_4 = v$$
$$\mathrm{H}_{3=}[0\ 1\ 0]$$

2.3.4 Model the noise

As the state process is not well defined which means in real situations the relative acceleration may not be constant, the continuous noise model Q should be as follow:

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & q_f \end{bmatrix} \tag{2.28}$$

Where q_f is the acceleration noise and it means to what extend the constant acceleration is trusted. This continuous process can be approximated by a time discrete process with the assumption q_f is constant and then the discrete Q is :

$$Q = q_f \begin{bmatrix} (\Delta t)^5/20 & (\Delta t)^4/8 & (\Delta t)^3/6 \\ (\Delta t)^4/8 & (\Delta t)^3/3 & (\Delta t)^2/2 \\ (\Delta t)^3/6 & (\Delta t)^2/2 & \Delta t \end{bmatrix} \tag{2.29}$$

Another noise matrix R is from measurement, R is set to r_1, r_2, r_3 respectively for each update. However, r_1, r_2 and r_3 may be not a constant value since the variance of measurement values is dependent on the relative distance as discussed in 2.2.2 and 2.2.3 .All the unsolved parameters can be set properly after the experiment.

2.3.5 Initialization

Basically, if no information about the initial state, then the $\hat{X}_{0|0}$ could be set as $[0\ 0\ 0]^T$, so the state variance matrix P should be set as a large value like :

$$P = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix}$$

The output of the filter will quickly adapt close to the real range, speed and acceleration. Actually, if the initialization value is too bad for adaption, then the first measurement value could be a good choice and a small P can be set to reduce the adaption time. In this thesis, the initial range is utilized the first measurement obtained by vehicle width method while the initial speed and acceleration are both set to 0.

2.4 Semi-automated tool development

2.4.1 Obtaining of optical flow vectors

According to the theory reported in 1.2.6, optical flow vectors can be obtained with either sparse optical flow or dense optical flow, which depends on the quality of detected video. The main issue is how to get the vectors that describe the motion information of the target vehicle.

2.4.1.1 Semi-automatic tracking

To detect the location of the target vehicle, a common method is to generate tracking region based on the optical flow vectors (or features) with similar properties [25]. Another feasible way is template matching. Both these two methods will detect several tracking regions including some mismatching regions. It is difficult to get the regions of interest and, for this reason, here we use a semi-automatic process that can improve the tracking performance.

We chose one frame as the first one and select a rectangle as the tracking region. According to the experience acquired in analyzing videos, the head lights of vehicle show better performance as features and optical flow vectors. Meanwhile, a relatively smaller region leads to a more accurate position. It means that we can estimate the height and offset of the tracking region with a better accuracy in order to generate the benchmark. After that, we can easily find the optical flow vectors whose starting points are in the tracking region. With the ends of these vectors, we generate a new rectangle as the tracking region in next frame. The same process can be done continually until the target is out of the frame. Figure 2.9, 2.10 and 2.11 show the process of tracking the right head light of the target vehicle.

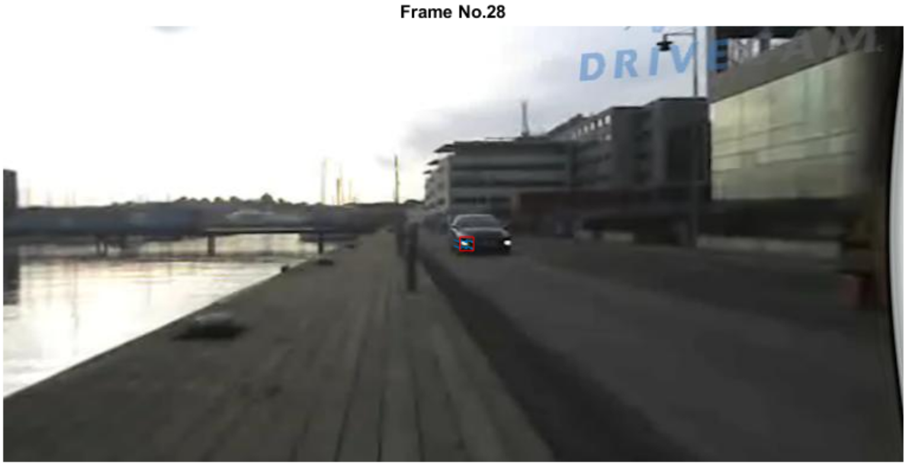


Figure 2.9 Tracking process 1

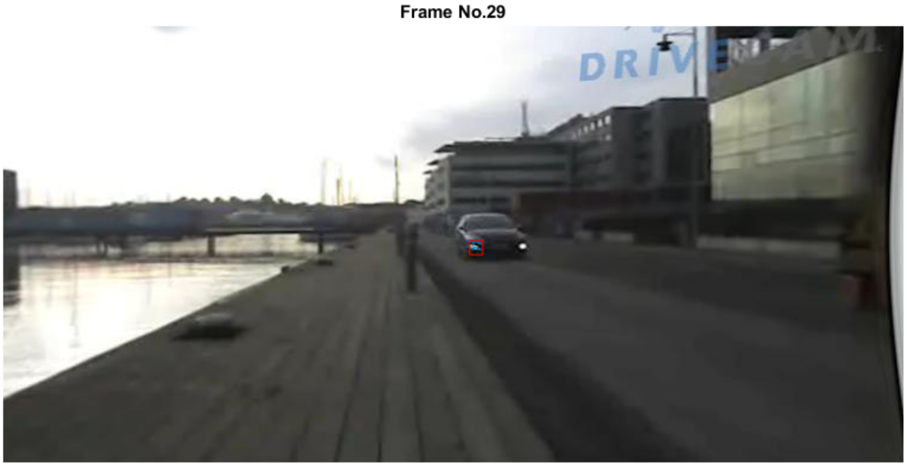


Figure 2.10 Tracking process 2

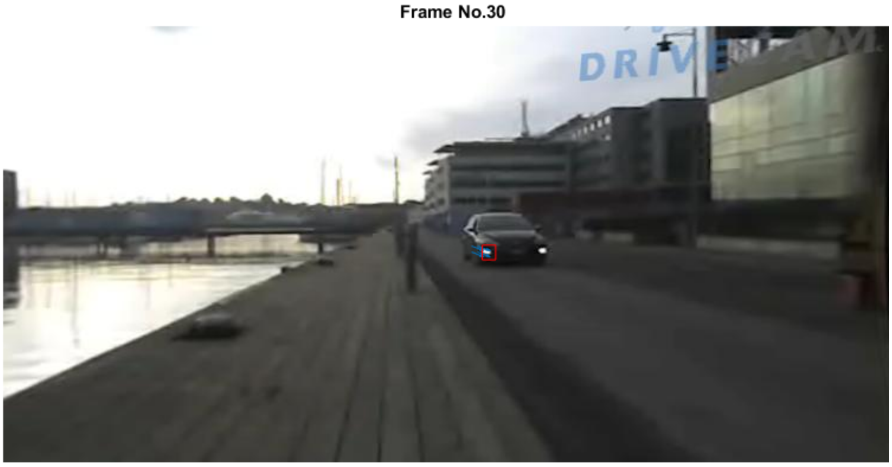


Figure 2.11 Tracking process 3

It should be noticed that the small distance between the target object and the camera may cause wrong tracking in figure 2.12. Unfortunately, in those frames, the theoretical optical flow vectors length is larger than the one in previous frames and this might have a large effect in the matching process, which is describe later. Hence, the proposed solution is to manually select the tracking points when wrong tracking happens.

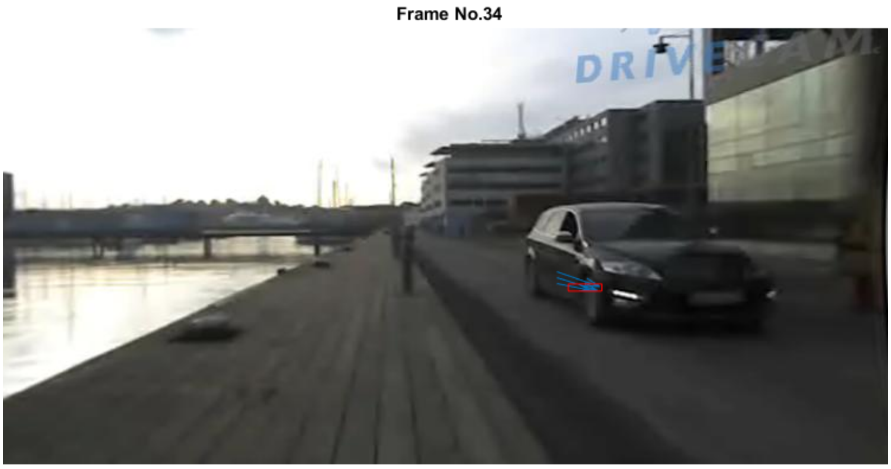


Figure 2.12 Wrong tracking result

2.4.2 Optical flow vector filter

Assuming the oncoming vehicle is traveling in straight forward direction, the optical flow direction of a certain feature (or a small tracking region) can be computed theoretically. With equation (2.3), the optical flow vector's direction of a specific point can be computed as:

$$\alpha = \arctan\left(\frac{f_y(h-Y)cos^2\theta - (Y-h)sin^2\theta}{f_x X_w cos\theta}\right) \tag{2.30}$$

where α is the angle between the direction of the vector and the direction of x-axis. According to the direction, the vectors in the tracking region can be filtered. However, since the offset X_w has a clear influence on the angle and the value of offset will be estimated manually in real measurement, we use boxplot to filter the undesired vectors. The angles of the vectors are draw the boxplot of them and, for the filtering, angles outside the box are eliminated. The remaining vectors are used for following steps.

2.4.3 Theoretical benchmark creation

In this section, it is described how to build a theoretical benchmark which can match the optical flow vector length to the speed of vehicle. Specifically, it is the relative speed between the oncoming vehicle, which is tracked, and the camera. Sampling time of the camera is 0.25s in this thesis.

Assuming the oncoming vehicle is traveling with a constant speed, parallel the ego-vehicle, according to imaging principle of camera, the relationship between the optical flow vector length and the distance from the vehicle to the camera can be calculated theoretically as the benchmark. Given the change of optical flow vector length, we can try to match the change to the benchmark in order to get the speed of the vehicle. Based on the coordinate transform formulae, the parameter, which effects most on the result, is the transvers offset between the target vehicle and the camera, instead of the pitch angle of the camera. The transform formulae are as following:

$$x_p = C_x + \frac{f_x X_w}{Z\cos(\theta) + (-h+Y)\sin(\theta)} \tag{2.31}$$

$$y_p = C_y + \frac{f_y((h-Y)\cos(\theta) + Zsin(\theta))}{Z\cos(\theta) + (-h+Y)\sin(\theta)} \tag{2.32}$$

where C_x and C_y are the coordinate value of the frame centre, f_x and f_y are the focal length of the camera, θ is pitch angle of the camera, h is height of the camera, Z and X_w are the perpendicular and horizontal distance from the target point to the camera and Y is the height of the point. Since θ is almost zero, h and Y don't influence the results too much. However, X_w(also known as offset) and Z have a main effect on the coordinate calculation. For constant X_w, θ ,Y and h, assuming the perpendicular distance from target point to the camera is Z_1 , we can get the position of that point in one frame as (x_1, y_1) with the formulae above. For a given speed v, the real perpendicular distance in the next frame can be calculated as $Z_2 = Z_1 - v * 0.25s$ and the coordinate of the point in next frame is (x_2, y_2). The theoretical optical flow vector between these two consecutive frames is $(x_2 - x_1, y_2 - y_1)$. Here, we define the vector as the optical flow vector of the current frame. Through this method, we can build the benchmark (figure 2.13) of how the constant speed and perpendicular distance effect the vector length. The effect of those parameters is shown in figure 2.14, 2.15, 2.16 and 2.17. The matching process is described as in the next paragraph.

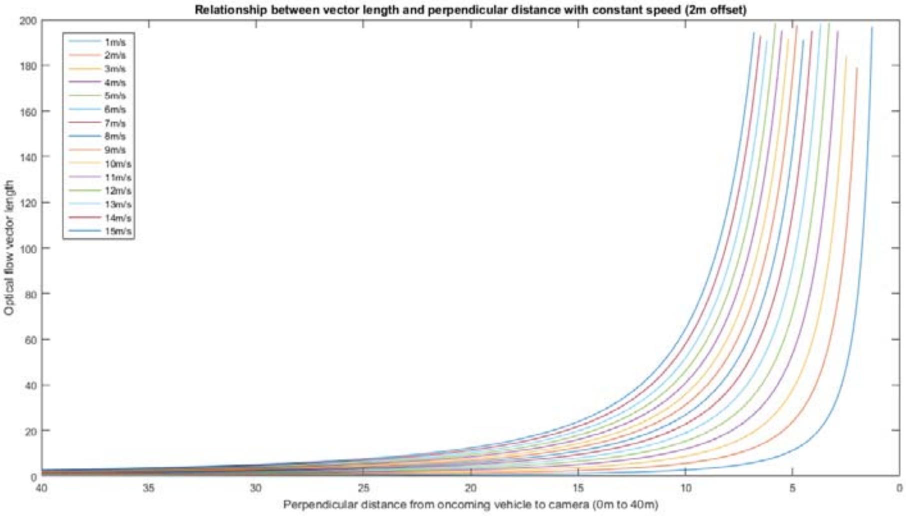


Figure 2.13 Benchmark (constant speed from 1m/s to 15m/s, offset of 2m)

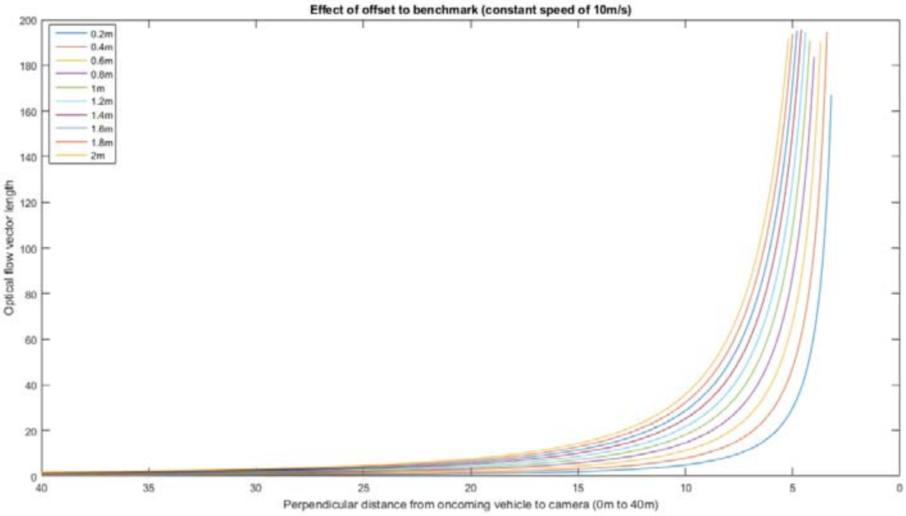


Figure 2.14 Effect of offset (from 0.2m to 2m)

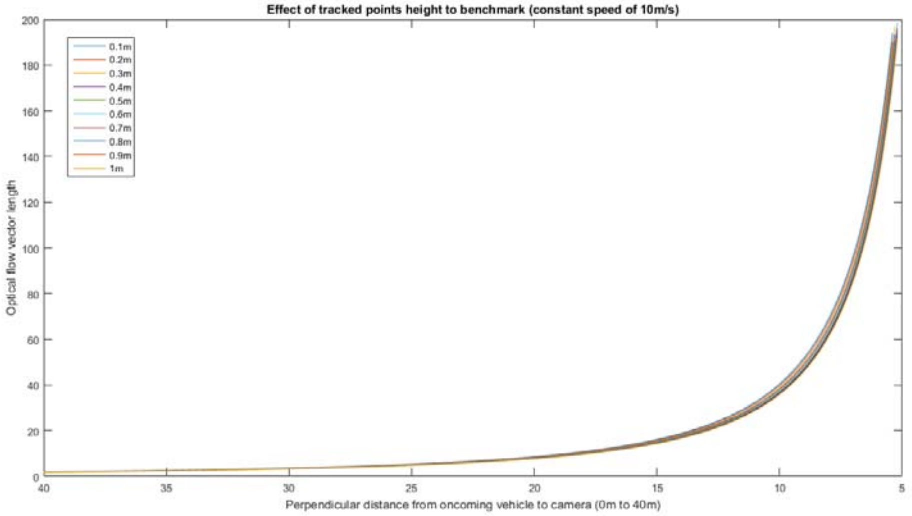


Figure 2.15 Effect of tracked point's height

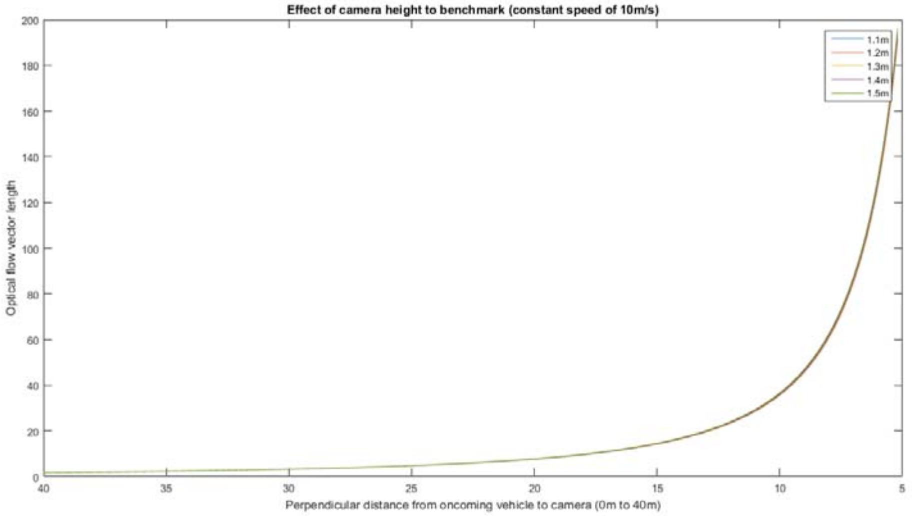


Figure 2.16 Effect of camera's height

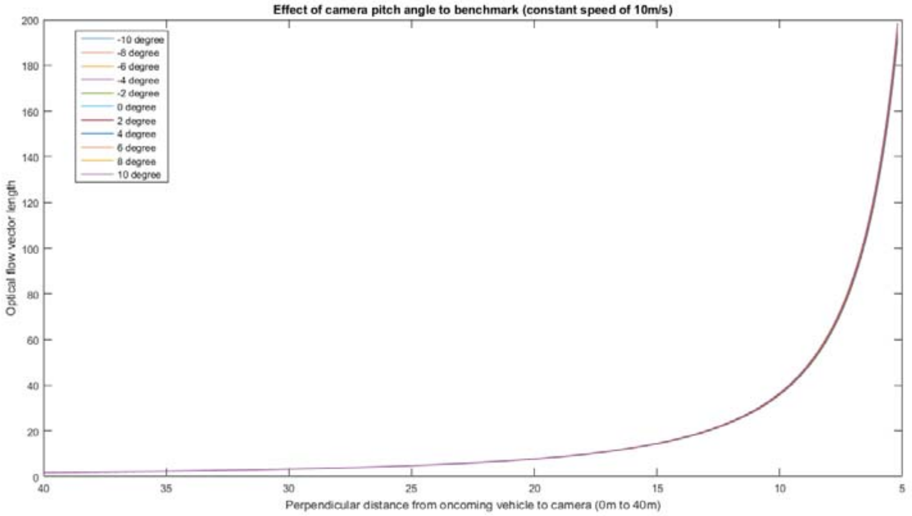


Figure 2.17 Effect of camera's pitch angle

Due to the large effect of offset, on the calculation, it is important to have a value of the offset as close as possible to the real, in order to build a reasonable benchmark for matching process. In experiments, the small camera's horizontal rotating angle has a significant effect a lot on the offset due to large range. Assuming the camera has a horizontal rotating angle α (the angle between the direction of oncoming vehicle and forward direction of the camera, as in figure 2.18), then the 'real' distance (Z') and 'real' offset (X') can be solved as:

$$Z' = (Xtan(\alpha) + Z)\cos(\alpha) \tag{2.33}$$

$$X' = \frac{X}{\cos(\alpha)} - (Xtan(\alpha) + Z)\sin(\alpha) \tag{2.34}$$

Hence, Z' and X' are used to generate benchmark instead. The process of calculating α is explained in next chapter.

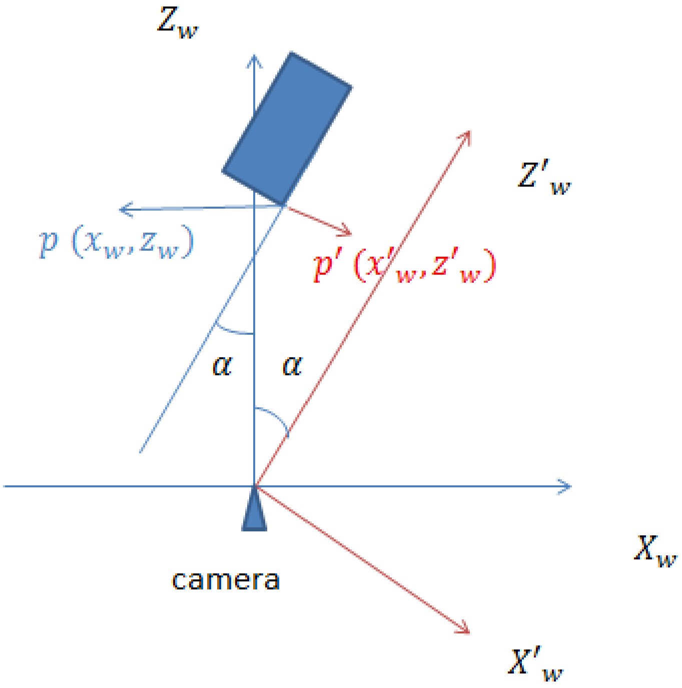


Figure 2.18 Calibration of camera's horizontal rotating angle

If we track one target point on the vehicle in the video, figure 2.18 illustrates the measurement result obtained with 'optical flow' method. Two parts of information that we can use are the vector length at each frame and the time duration between each two consecutive frames. In this process, the points on the measurement curve should be matched on the curves of benchmark as shown in figure 2.13.

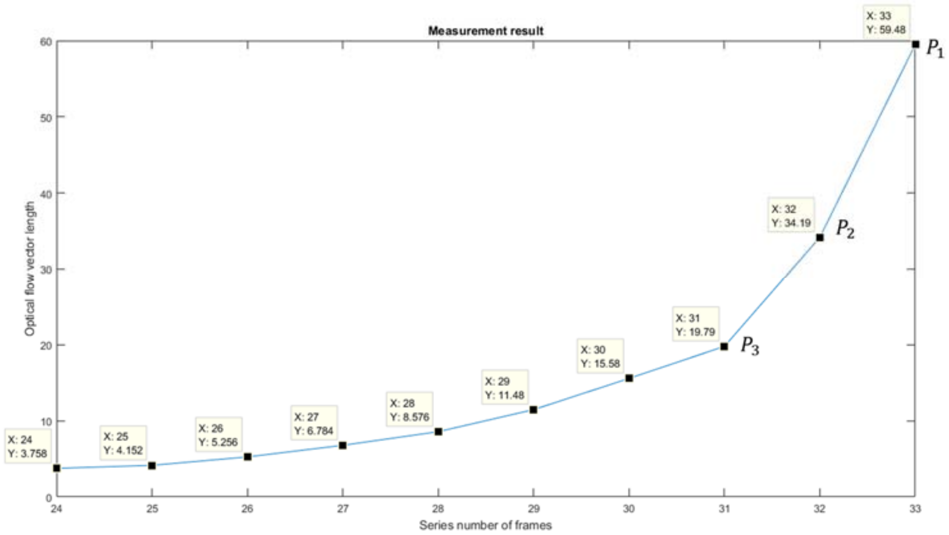


Figure 2.19 Example of one measurement result

Each step can match two continuous points on one curve and we define the matched speed as the speed of latter frame. Two principles should be followed. First, the time duration between the two matched points should be as close as possible to 0.25s. Second, the distance information of each two matched points should be reasonable. For instance, given two measurement points P_1 and P_2 as in figure 2.13, P_1 and P_2 are matched on curve with speed of v_1. We can get the perpendicular distance of these two points as d_1 and d_2. Clearly, the duration between these two points can be solved as: $(d_1 - d_2)/v_1$ and we should make sure that this value is close to 0.25s. As the definition before, we match P_1 to speed of v_1 and distance of d_1 as well. With the same method, P_2 can be matched to v_2 and d_2, with computation over P_2 and P_3. Theoretically, from P_2 to P_1, which means from d_2 to d_1, the target point needs to move at speed of v_2 for 2.5s. That is, $d_2 - 0.25 * v_2$ should be around d_1, such as ± 0.5m. As explanation above, the 1st principle makes the process accurate in two consecutive frames while the 2nd one satisfies two continuous pairs of frames. The process starts from the last point on the measurement curve since it shows higher resolution on the benchmark when the target object is closer to the camera.

To achieve these two principles, several steps need to be considered. First, for each pair of consecutive points, we match them to every curves in benchmark as in figure 2.19. For example, (P_1, P_2) can be matched to those 15 curves as $(p_{1,1}, p_{1,2}), (p_{2,1}, p_{2,2}) \ldots (p_{15,1}, p_{15,2})$, as in figure 2.19. Then we calculate the duration between each pair of matching points and sequence them according to how much are the durations close to 0.25s, from small difference value to larger ones. The result is shown in table 2.1(1). We chose the first value, $v = 6$m/s, as the matching result. After that, we deal with the next pair of points $(p_{x,2}, p_{x,3})$ in table 2.1(2) and find the first value of matching points that satisfies the 2nd principle. It matches $v = 7$m/s in this case. However, if all those data are mismatched, we get back to the previous step and select the next matching points. With the same processes through the measurement, we can finally get the most accurate matching result with reasonable change of position.

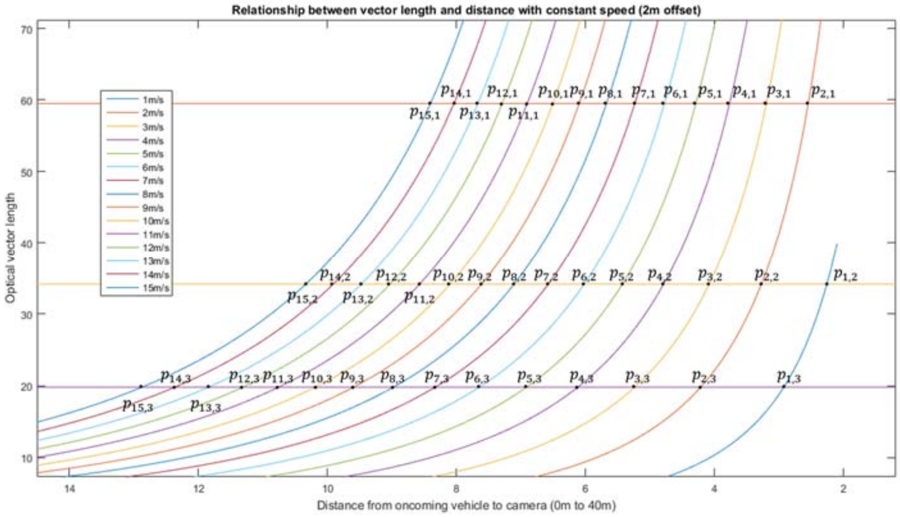


Figure 2.20 Example of matching benchmark

Table 2.1 (1) Example of matching benchmark

Matching speed of P_1 (m/s)	Time duration between P_1 and P_2 (s)	Matching distance of P_1 (m)
6	0.2333	5.5
7	0.2286	6.0
5	0.2800	4.9
8	0.2125	6.5
9	0.2000	6.9
4	0.3000	4.3
10	0.1900	7.4
11	0.1818	7.8
3	0.3333	3.7
12	0.1667	8.3
13	0.1615	8.7
14	0.1571	9.1
15	0.1533	9.5
2	0.4000	3.0
1	0.6000	2.0

Table 2.1 (2) Example of matching benchmark

Matching speed of P_2 (m/s)	Time duration between P_2 and P_3 (s)	Matching distance of P_2 (m)
10	0.2500	9.3
11	0.2364	9.8
9	0.2667	8.7
12	0.2250	10.3
8	0.2750	8.2
13	0.2154	10.8
7	0.2857	7.6
14	0.2071	11.3
15	0.2000	11.8
6	0.3333	6.9
5	0.3400	6.3

4	0.4000	5.5
3	0.4667	4.7
2	0.5500	3.8
1	0.8000	2.6

3 Experiment and Results

3.1 Experiment design

The experiment was supported by Volvo Car Cooperation and Safer. The Lytx camera and a high definition camera were mounted on the SV Volvo V40. The POV was a Volvo V60. The height of both cameras was 1.2 m above ground and the offsets from driver side of these two cameras were 0.83 m and 1.15 m respectively. As shown in Figure 3.1, a base station was used to enhance the accuracy of the (RTK/differential) GPS positioning and to ensure that the positioning error was less than 0.1m. The experiment allowed to obtain 'real' values for the range and offset between SV and POV. The speed information was obtained from Controller Area Network (CAN) data. A manual trigger from the Lytx camera was used to create a time stamp in the GPS data so the data from the SV (including Lytx video) and the POV could be synchronized to GPS time.

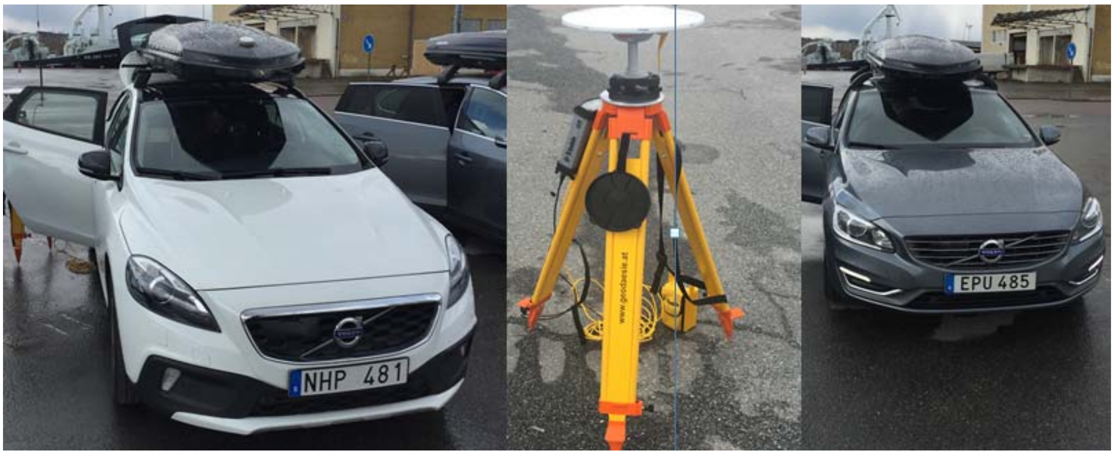


Figure 3.1 Experiment equipment

Table 3.1 showed the 29 test cases implemented in the experiment. Each scenario was conducted once. The offset was defined and measured as the lateral distance between the geometrical centres of two vehicles. Group 1 aimed to understand how different relative speeds influence the methods developed before. Group 2 aimed to evaluate how heading angle influences different methods. The influence of braking and offsets on the three methods was tested in Group 3 and Group 5 respectively. Group 4 was mainly designed for semi-auto tool to understand how the headlight influences tracking. Unfortunately, due to loss of signal of GPS both in SV and POV, 11 cases reported with dark grey background in Table 3.1 were lost.

Table 3.1 Experiment scenarios

Index	SV speed	SV heading angle	POV speed	POV headlight status	offset
1.1	Standstill	0 degree	Constant at 10 km/h	on	around 3m
1.2			Constant at 20 km/h		
1.3			Constant at 30 km/h		
1.4			Constant at 40 km/h		
1.5			Constant at 50 km/h		

1.6			Constant at 60 km/h		
1.7			Constant at 70 km/h		
1.8	Constant at 20 km/h		Standstill		
1.9			Constant at `10 km/h		
1.10			Constant at 20 km/h		
1.11	Constant at 30 km/h		Standstill		
1.12			Constant at 10 km/h		
1.13			Constant at 20 km/h		
2.1	Standstill	10 degree	Constant at 30 km/h	on	around 3m
2.2			Constant at 50 km/h		
2.3		30degree	Constant at 30 km/h		
2.4			Constant at 50 km/h		
3.1	Standstill	0degree	Start at 30 km/h with comfortable braking	on	around 3m
3.2			Start at 30 km/h with hard braking		
3.3			Start at 50 km/h with comfortable braking		
3.4			Start at 50 km/h with hard braking		
3.5	Start at 30 km/h with comfortable braking		Standstill		
3.6	Start at 30 km/h with hard braking		Standstill		
4.1	Standstill	0degree	Constant at 30 km/h	off	around 3m
4.2			Constant at 50 km/h		
4.3			Constant at 70 km/h		
5.1	Standstill	0degree	Constant at 30 km/h	on	around 3m
5.2			Constant at 30 km/h		around 9m
5.3			Constant at 50 km/h		around 6m

3.2 GPS Data processing

DEWEsoft [32] was used to extract GPS data and convert them to Matlab files. During the process, special attention had to be placed on the synchronization due to the fact that the sample frequency of GPS data for the SV changed during the experiment: when the SV stood still the sample frequency was 10 Hz and changed to 100 Hz when the vehicle moved. For the POV, the sample frequency was always constant and equal to 100 Hz.

Figure 3.2 shows an example of obtaining the lateral distance (offset) and longitudinal distance (range) between the SV and POV when both of them are moving. First, the GPS longitudinal and lateral positioning data in WGS84 coordinates were converted into RT90 coordinates. A linear regression of positioning data of either the POV or SV was conducted to obtain the angle $\emptyset$ related to horizontal axis. Then, the distance L between each two synchronized points and angle τ was obtained. Therefore, with L and angle ($\tau - \emptyset$), the offset and range was calculated.

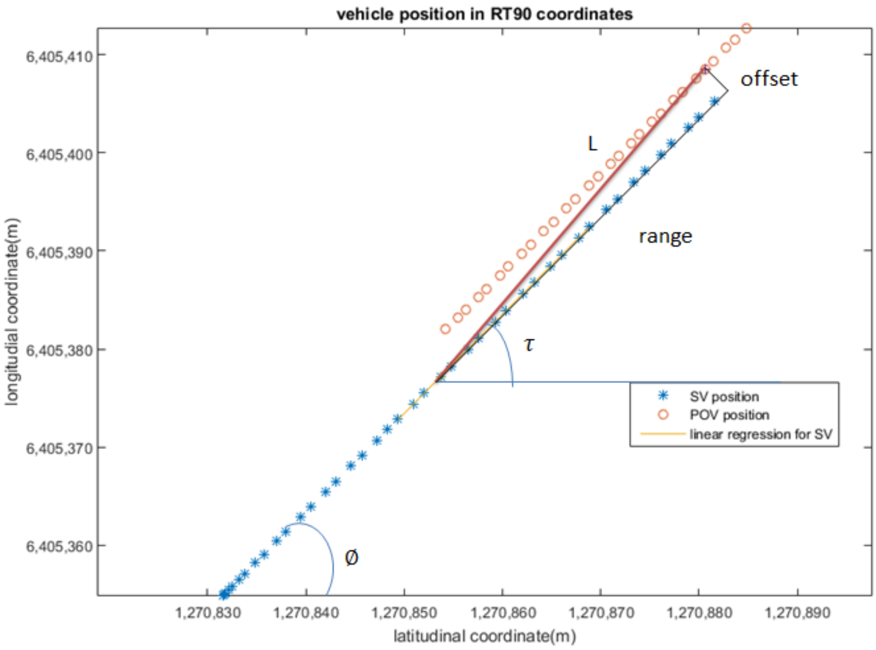


Figure 3.2 Offset and range calculation with GPS data

3.3 Results

All the remaining 19 cases were used for further analysis and each case provided 20 to 30 frames can be measured, thus an overall sample of 285 measurements is used for the analysis of error for the manual tool.

3.3.1 Measurement result of vehicle width method

Table 3.2 and Figure 3.3 show the accuracy of the method that assesses vehicle width in measuring the distance between the camera on the SV and POV.

Table 3.2 Range error analysis for vehicle width method

range (m)	5-10	10-15	15-20	20-25	25-30	30-40	40-50	＞50
number of measurements	20	34	36	35	35	58	40	26
mean error(m)	2.57	1.29	1.56	1.29	1.25	-0.08	-0.69	-0.54
Standard deviation(m)	2.35	2.07	2.27	2.04	2.37	2.92	3.55	4.62
mean absolute error(m)	2.82	1.81	1.96	1.72	2.00	2.27	2.95	3.87

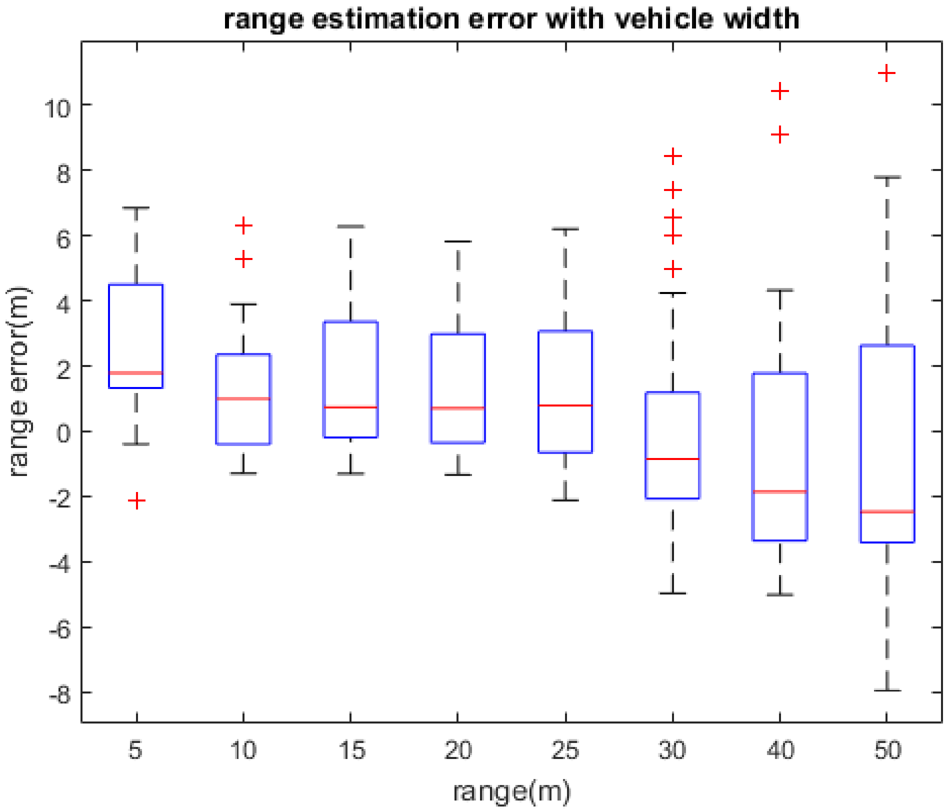


Figure 3.3 Range error distribution at different ranges

\

In general (except for the range 5-10 m), based on the mean absolute error in table 3.2, the method achieves more accurate results when the actual range is smaller. Also, the standard deviation of the errors shows that this method is more stable when the actual range is small, again excluding the range 5-10 m.

In the very short range (5-10m), the range estimation is worse because it is hard to obtain accurate vehicle width in pixels(no distinct width to latch on to) as can be seen from Figure 3.4..

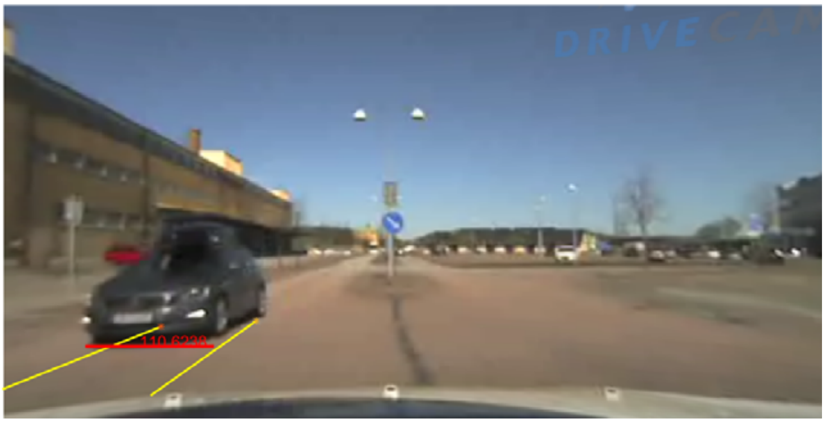


Figure 3.4 Difficulty in the vehicle width estimation in pixels with short range

3.3.2 Measurement result of ground point method

Table 3.3 and Figure 3.5 show statistical analysis of the range error with ground point methods. From the table, it is clear that this method is accurate and stable when two vehicles get closer to each other, since both the mean absolute error and standard deviation become smaller.

On the other hand, the range error is much larger compared to the vehicle width method when the actual range is larger than 30 m. Besides, when the range is larger than 30m, the range error seems to be always negative, meaning that the estimated range seems to be always smaller than the real values. This trend can be explained by the annotator selecting points slightly in front of the 'correct' ground points.

Table 3.3 Range error analysis for ground point method

range (m)	5-10	10-15	15-20	20-25	25-30	30-40	40-50	＞50
number of measurements	20	34	36	35	35	58	40	26
mean error(m)	0.17	-0.02	-0.14	-0.19	-0.43	-2.18	-5.17	-8.40
Standard deviation(m)	1.69	1.80	1.96	2.22	2.43	3.35	6.84	5.31
mean absolute error(m)	1.40	1.43	1.74	1.89	1.95	3.34	7.33	8.51

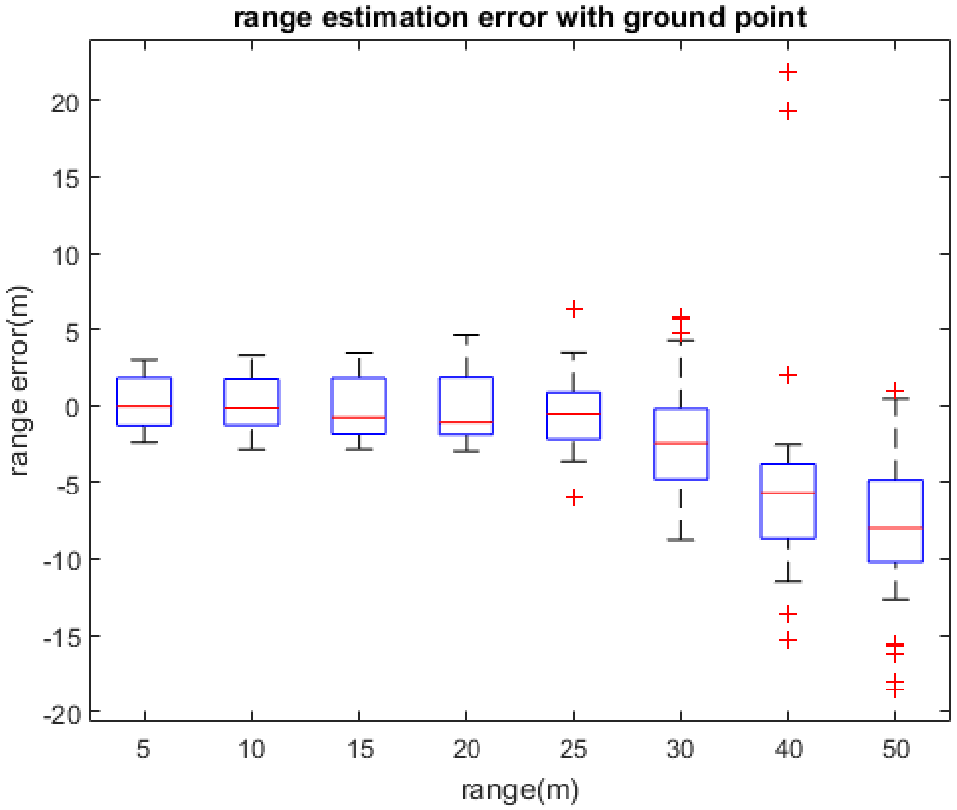


Figure 3.6 Range error distribution at different ranges

This method also allows for the calculation of the offset between two vehicles. However, the lateral offset between the two vehicles was only analyzed when the actual range is 5-15m, since the offset was not the main focus of this thesis and more accurate offset can be obtained for closer ranges. The results for the offset, reported in Table 3.4 and Figure 3.6, are good considering that the real offset in the experiment is often more than 3m.

Table 3.4 Offset error analysis for vehicle width method

range (m)	5-10	10-15
number of measurements	20	34
mean error(m)	-0.09	0.04

Standard deviation(m)	0.20	0.27
mean absolute error(m)	0.17	0.19

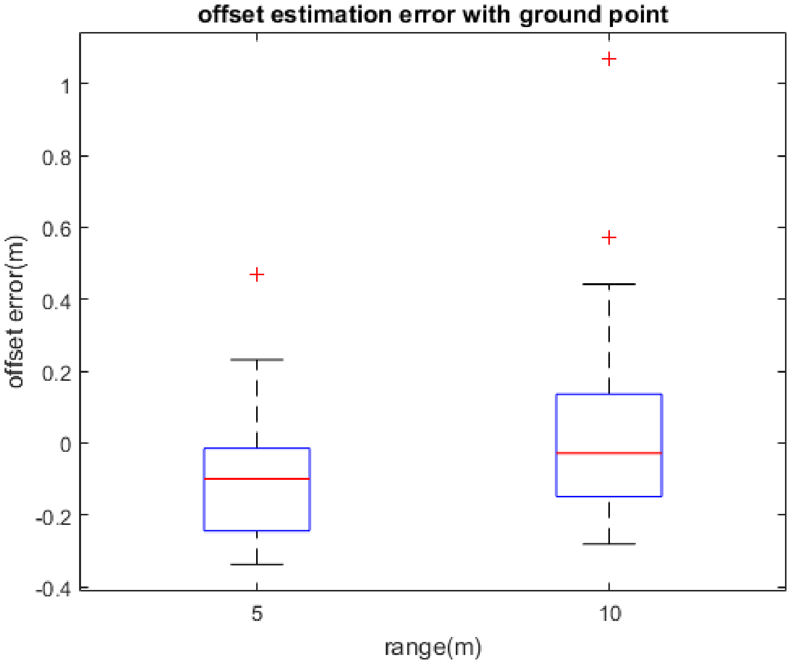


Figure 3.7 Offset error distribution at different ranges

3.3.3 Measurement result of manual optical flow method

As discussed in chapter 2.2.4 and 2.2.5, the relative speed error from vehicle width method is large and manual optical flow method is aiming to provide a more accurate estimation of speed. Applying the same match method in semi-automated tool and a stricter rule is set to match the speed. The time threshold was set as 0.1 s and distance threshold as 0.3 m. Due to the unstable manually selection and the uneven road, 65 measurements were matched with relative speed and the results can be seen in Table 3.5 and Figure 3.8.

Table 3.5 Speed error analysis for manual optical method

range (m)	5-20	＞20
number of measurements	41	24
mean error(m/s)	-1.39	0.15
Standard deviation(m/s)	3.22	4.91
mean absolute error(m/s)	2.89	3.70
mean absolute percentage error	29.25%	43.15%

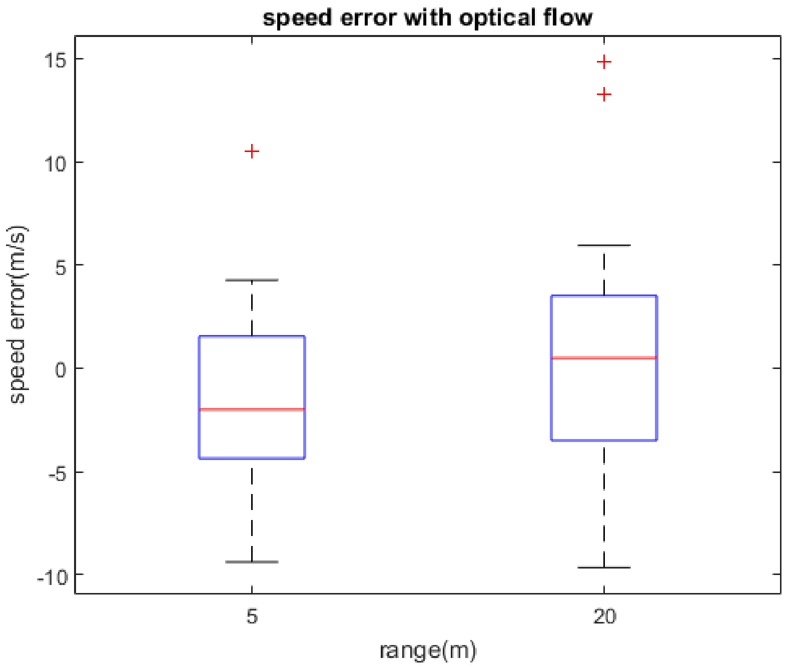


Figure 3.8 Speed error distribution at different ranges

Clearly, this result is not ideal and the mean percentage error is still around 30%. This result could be improved after applying Kalman filter.

3.3.4 Measurement result of semi-automatic method

In this section, it is described the semi-automatic method used to detect the speed and position of the oncoming vehicle. Only data from experiments, where the SV is proceeding straight forward, are used in this analysis.

In experiment 2.3, the effect of offset was investigated. The oncoming vehicle is traveling at 30 km/h and true offset (from its left headlight to the camera) is 1.9 m. We generate the benchmark with offset from 1.7 m to 2.1 m and matched the measurement optical vector length to it. The results are shown in Figure 3.9 and Figure 3.10 while the error is illustrated in Table 3.6 and Table 3.7.

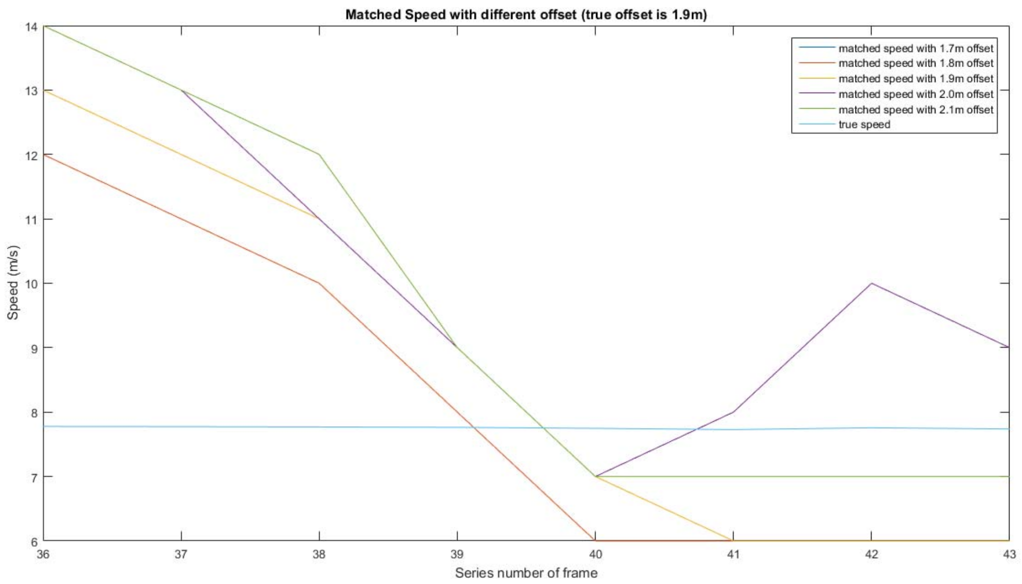


Figure 3.9 Matched speed with different offset

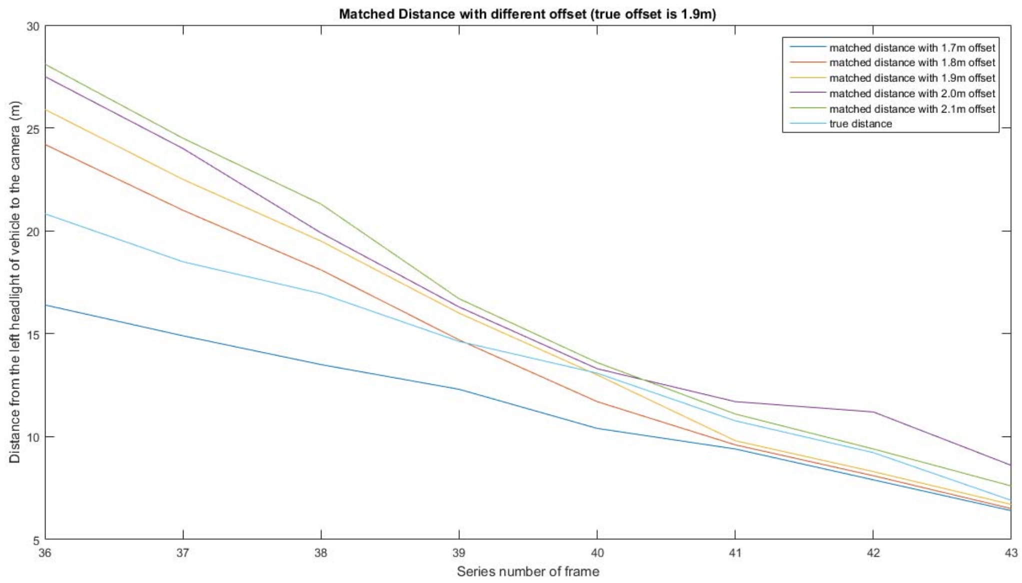


Figure 3.10 Matched Distance with different offset

Table 3.6 Mean absolute speed error of matching process (m/s)

range (m)	5-10	10-15	15-20
offset=1.7	1.7478	0.9904	3.2207
offset=1.8	1.7478	0.9904	4.2207
offset=1.9	1.2522	0.9904	4.8874
offset=2.0	0.7478	0.9904	5.2207
offset=2.1	0.7478	1.9904	5.8874

Table 3.7 Mean absolute range error of matching process (m)

range (m)	5-10	10-15	15-20
offset=1.7m	3.3408	0.7279	3.3408
offset=1.8m	0.3034	1.6443	4.8741
offset=1.9m	2.5368	1.9443	6.0408
offset=2.0m	1.4034	2.2943	6.8741

offset=2.1m	1.6034	3.5443	8.1075

The errors in Table 3.6 are under 2 m/s at small range while the errors in Table 3.7 are mostly in 3 m. Meanwhile, the results, when offset equals to 1.7 m and 1.8 m, show better accuracy than that with true offset (1.9 m). This error is discussed in next chapter. In following matching process, to simplify the process, we use the true offset, instead of a set of different offset, to generate the benchmark.

The measurement results of experiment 2.3 - 2.7 are displayed in Table 3.8, Figure 3.11 and Figure 3.12.

Table 3.8 Speed and range error in different distance range

range (m)	<15	>=15
mean absolute error of speed (m/s)	2.1573	3.4625
Mean absolute percentage error of speed	19.49%	31.03%
mean absolute error of range (m)	3.3257	5.2282
Mean absolute percentage error of range	32.75%	28.32

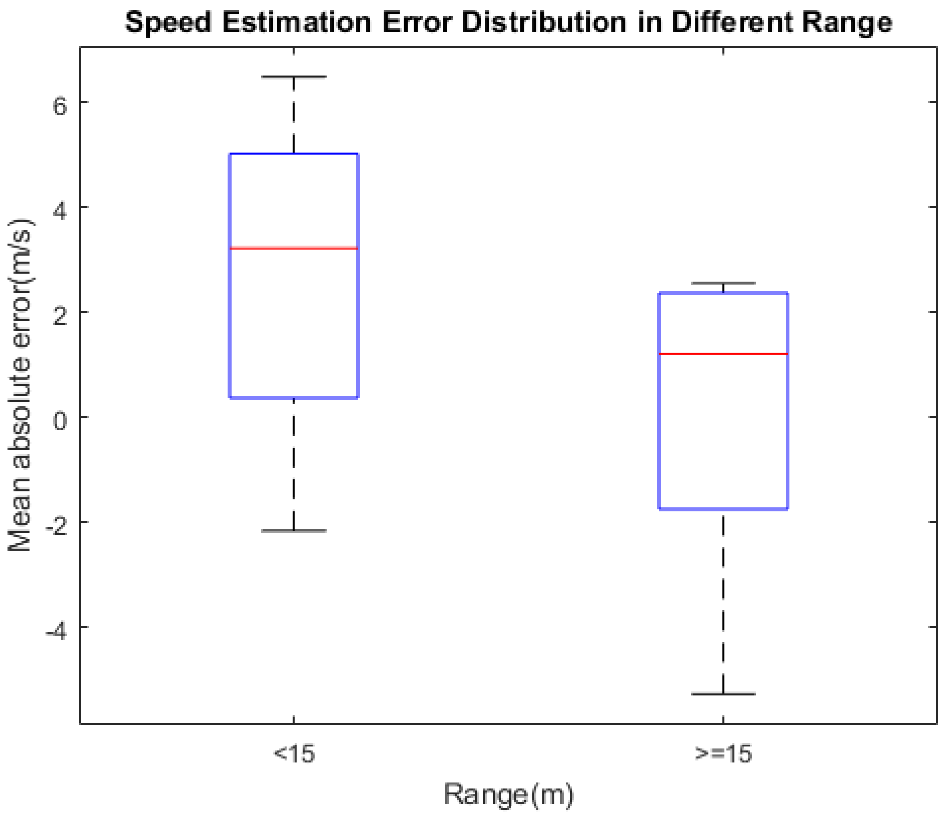


Figure 3.11 Speed estimation error distribution in different range

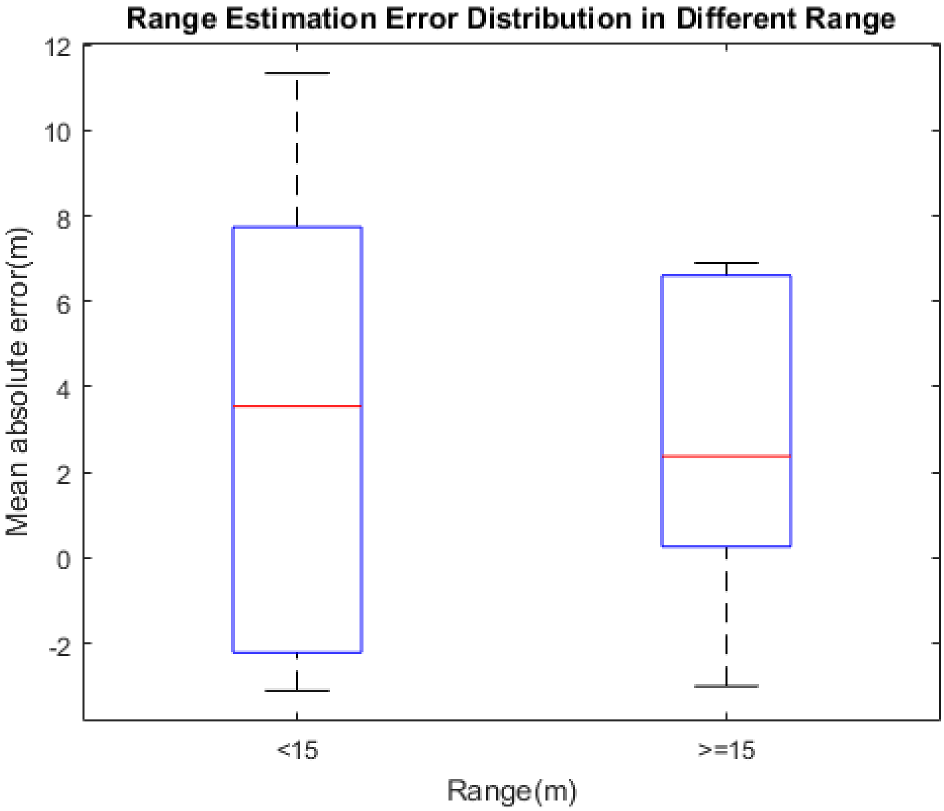


Figure 3.12 Range estimation error distribution in different range

In experiments 2.8-2.13, the road is relatively rougher. Therefore, in the video, the tracked point is floating across the theoretical line. This situation causes unreasonable direction of optical flow vectors as in figure 3.11. Hence, the measurement results show a much larger error than previous ones, such as in Table 3.9.

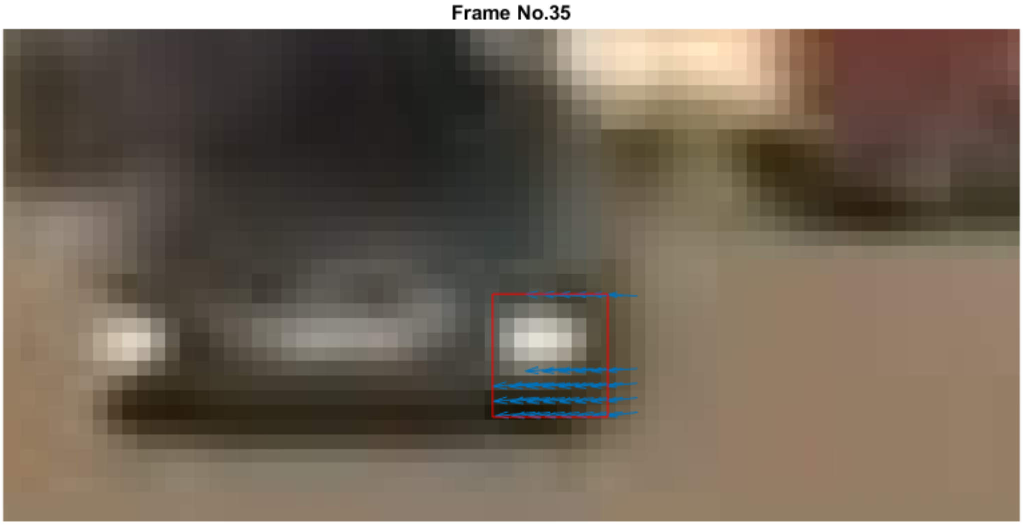


Figure 3.11 Non-reasonable vector direction caused by rough road (experiment 2.9)

Table 3.9 Speed and Distance error with rough road experiment 2.9

Range(m)	15-10	<10
Speed error	56%	32.76%
Distance error	60.95%	67.38%

In experiment 3.4, since the vehicle was braking, far away from the camera, the optical flow vector length is too small that the benchmark shows low resolution in that range. So we may get large error if we try to detect the motion information. In this case, we skip the braking situation.

3.3.5 Kalman filter setting and result

As shown in Table 3.2 and 3.3 and 3.5, the standard error is calculated at different values of the range. Thus, according to the discussion in chapter 2.3.4, the measurement variance r1, r2 and r3 - which is the square of standard error - can be set as reported in Table 3.10:

Table 3.10 Measurement variance setting in Kalman filter

range (m)	5-10	10-15	15-20	20-25	25-30	30-40	40-50	＞50
r1(m)	5.51	4.30	5.17	4.15	5.62	8.53	12.61	21.38
r2(m)	2.87	3.23	3.85	4.92	5.90	11.25	46.76	28.18
r2(m/s)	8.35	8.35	8.35	13.69	13.69	13.69	13.69	13.69

Figure 3.12 is the result of the range estimation in experiment 2.8 by utilizing different methods. The Kalman filter result is a compromise of all methods and the constant acceleration model, according to the setting in Table 3.10, at different range, Kalman filter trusts different methods to varying degrees and the variance can be regarded the measure of this trust.

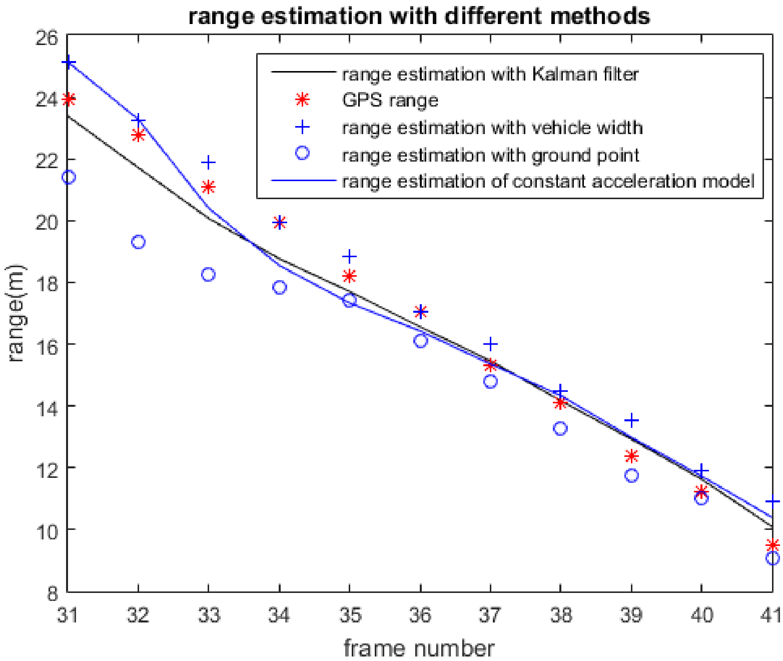


Figure 3.12 Example of range estimation with different methods.

The range estimation results after applying the Kalman filter is shown in Table 3.11 and Figure 3.13.

Table 3.11 Range error analysis for Kalman filter

range (m)	5-10	10-15	15-20	20-25	25-30	30-40	40-50	＞50
number of measurements	20	34	36	35	35	58	40	26

mean error(m)	1.42	0.58	0.72	0.57	0.31	-0.88	-1.55	-1.14
Standard deviation(m)	1.62	1.87	1.97	2.04	2.40	2.78	3.95	4.15
mean absolute error(m)	1.67	1.52	1.63	1.57	1.92	2.45	3.34	3.80

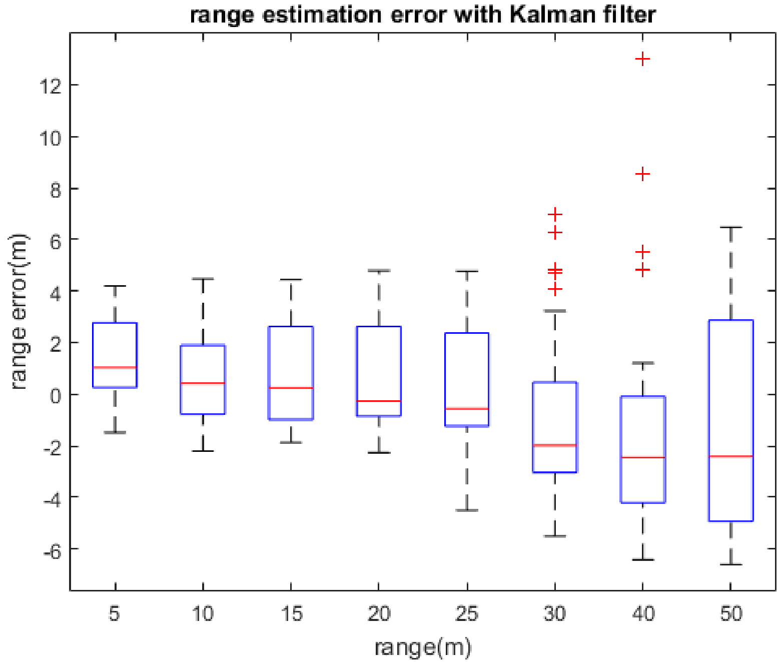


Figure 3.13 Range error distribution at different ranges

Also the relative speed is obtained from Kalman filter, since the initial relative speed is always set to 0, thus Kalman filter needs 2 or 3 frames to adjust to close the 'real' value, thus the speed estimation of first 2 frames are far from real value, thus they were deleted in the following analysis. The rest results are shown in Table 3.12 and Figure 3.11.

Table 3.12 Speed error analysis for Kalman filter

range (m)	5-10	10-15	15-20	20-25	25-30	30-40	40-50	＞50
number of measurements	20	34	36	35	32	46	30	18
mean error(m/s)	0.68	0.10	0.44	-0.25	0.59	0.74	0.85	-0.29
Standard deviation(m/s)	2.08	1.43	2.42	2.11	2.74	3.83	4.80	3.96
mean absolute error(m/s)	1.39	0.93	1.38	1.64	1.99	2.38	2.65	2.53
mean absolute percentage error	14.7%	10.5 %	15.6%	19.7%	40.6%	44.2%	46.2%	47.9%

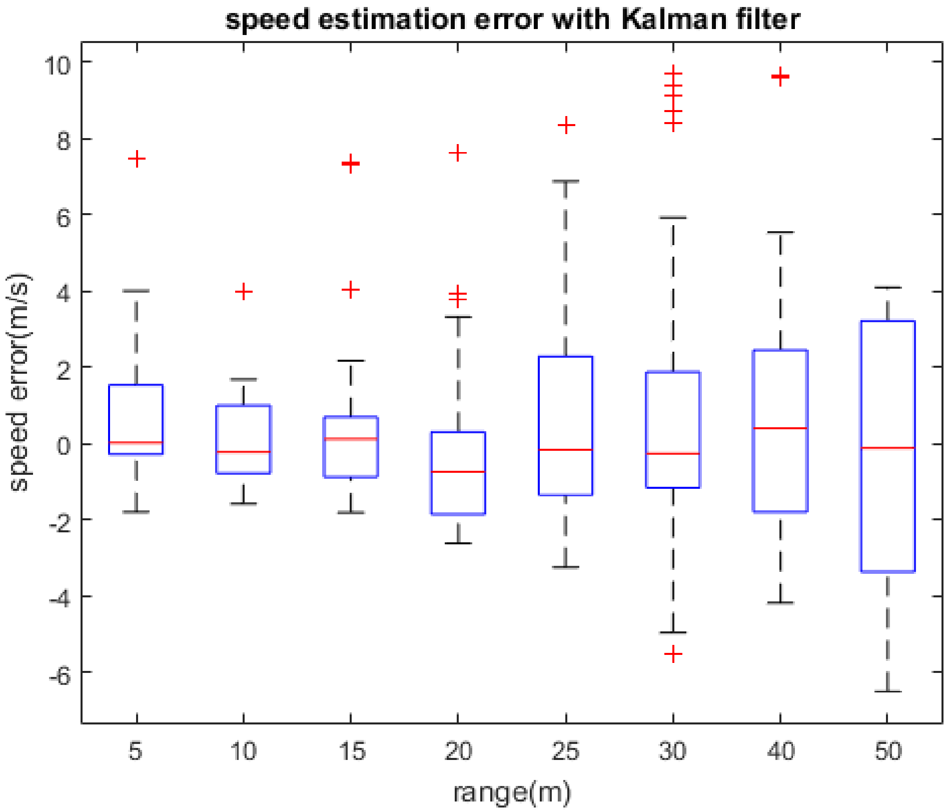


Figure 3.14 Speed error distribution at different ranges

As can be seen from Table 3.12, except the shorter range 5-10m, the mean absolute percentage error increased with the range. The speed estimation percentage error within the range 5-20m is less than 15%. Figure 3.14 shows that there are many abnormal points of speed estimation, this is because the range rate is sensitive to the range estimation, a 0.5m error in range can lead to 2m/s error in speed if the sample time is 0.25s.

4 Discussion

4.1 Manual tool discussion

Considering the range estimation for different methods, taking into account the tables for the actual range 5-15 m, the ground point method gets the best result, being the mean absolute percentage error around 7%. The Kalman filter does not improve the range estimation in the range 5-15 m but, between 15 m and 30 m, it has smaller mean absolute value and standard deviation, compared to the other methods. As for the larger range area (>30m), the vehicle width method gets the best result and the ground point method is far from accurate.

Comparing to the range estimation results in Paper [21], our results of vehicle width method are much worse in short range (less than 10m) and slightly worse in the range between 10m-40m. This is mainly because the lateral offset has an influence in the vehicle width annotation on images especially at a short range. On the other hand, in paper [21], a linear regression model is used to calibrate the actual range with measurements from 1m-10m, this increases the accuracy comparing to only calibrating the camera with Matlab toolbox.

When it comes to the speed estimation, the manual optical flow method obtains poor results although small thresholds have been set. The Kalman filter significantly improves the speed estimation result since it integrates three methods together and a smoother range is obtained. Overall, this result represents the greatest contribution of applying the Kalman filter in the present thesis. However, as discussed in 2.2.4, range rate errors increase with range and range rate and, due to the limited numbers of experiment, we failed to observe this relationship among range, range rate and range rate error. Also, due to the limited experiment, we cannot clearly find how the braking of POV influences the speed estimation.

As mentioned in chapter 3.1, we planned to find how heading angle influences the range estimation. Figure 4.1 is the result of range estimation in experiment 2.2, which provides the only data available with different heading angle. It seems that this angle has a small influence on the range estimation with vehicle width and Kalman filter tends to slightly modify this error. Using the measurements for ranges inferior to 25 m, the average speed of Kalman output is 47.6 km/h, while the speed of POV from CAN data is 48.6 km/h. From these results, the estimation seems to be good but more experimental data are needed to support the conclusion..

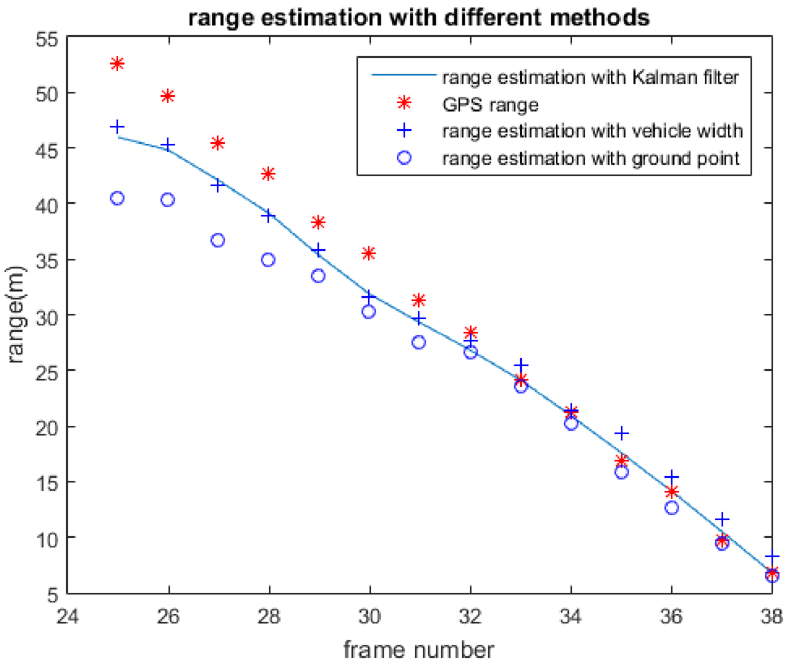
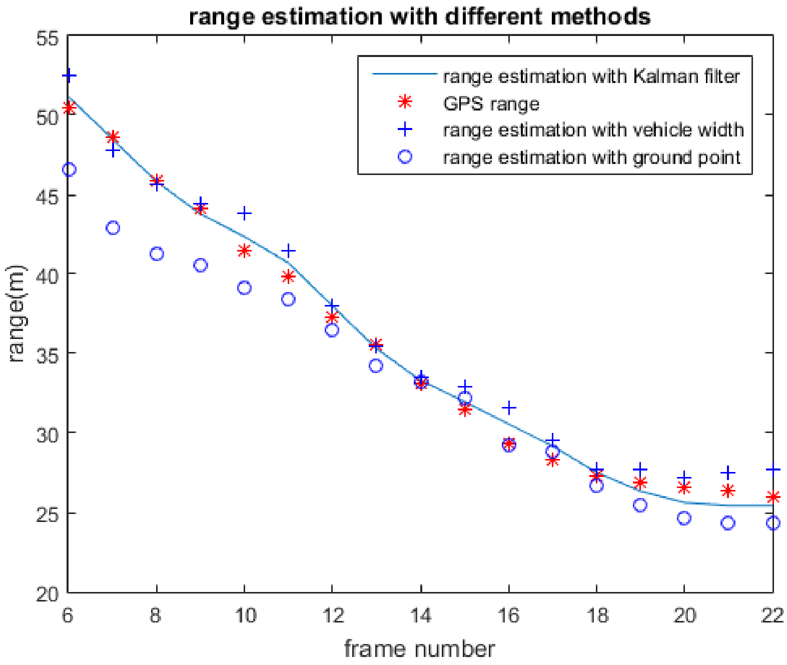


Figure 4.1 Range estimation of different methods in experiment 2.2

Regarding for the effect of braking, as illustrated in 3.3.4, optical flow matching cannot be used. Figure 4.2 and 4.3 illustrate the results for range estimation and speed estimation. The Kalman filter quickly adapts to the real speed although the initial speed of POV is set to zero and good range estimation results has been observed in this hard braking case.



,

Figure 4.2 Range estimation of different methods in experiment 3.2

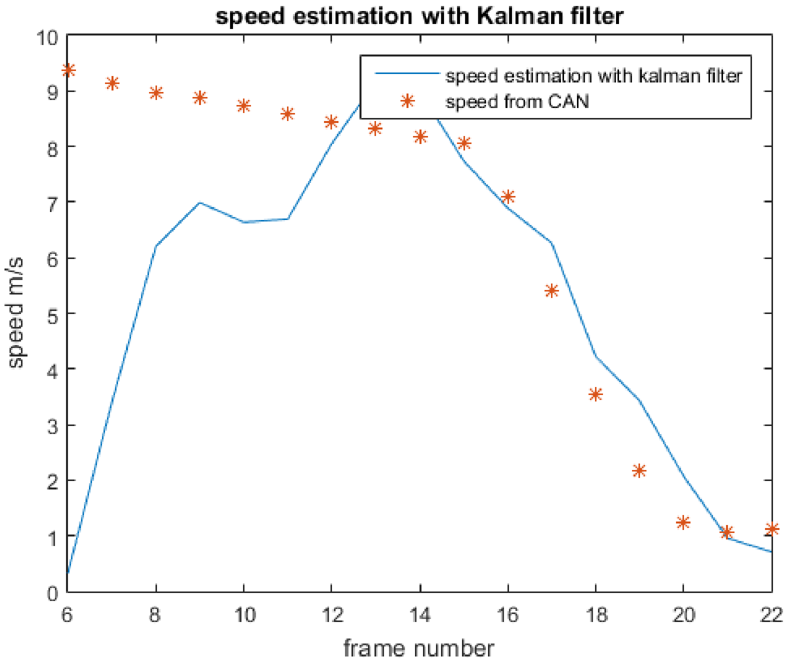


Figure 4.2 Speed estimation with Kalman filter in experiment 3.2

The influence of offset on the methods is illustrated in Figure 4.4 and 4.5 that consider the results of experiment 5.2 and 5.3 in table 3.1. The offset is around 9.5 m and 6.7 m respectively from GPS data. The results show that the offset has a larger effect on the vehicle width method: the annotator tends to estimate a smaller vehicle width in pixels when the offset is larger, thus the range estimation is larger. The ground point method is also slightly influenced by the offset: the estimation is always larger, probably due to the support lines shown in Figure 2.8 and because the annotator tends to select points on one side of these lines.

As for speed estimation, similar slopes of different range measurement are observed, and, therefore, the offset has little effect on speed estimation. The Kalman filter does not help in improving range estimation on these scenarios but it provides a smoother range line, thus the speed estimation is improved.

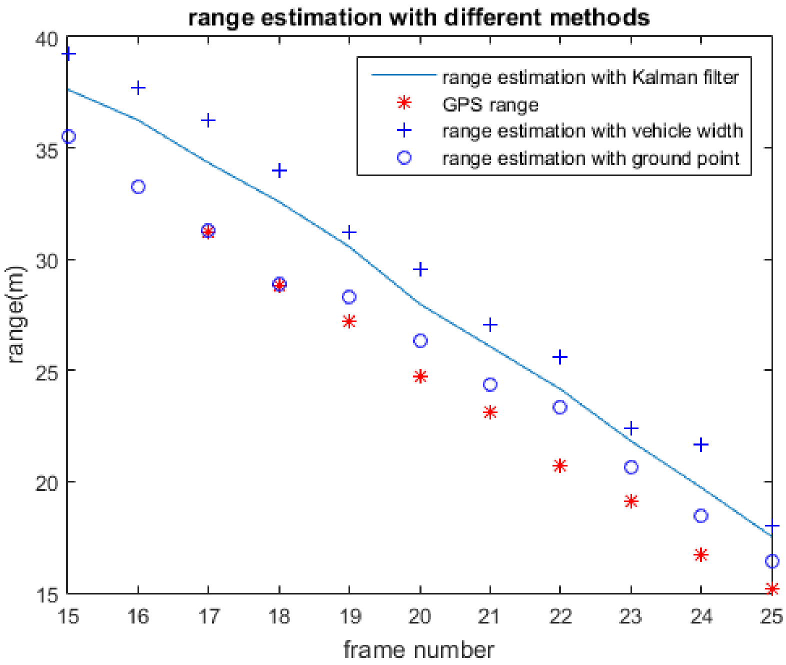


Figure 4.4 Range estimation of different methods in experiment 5.2

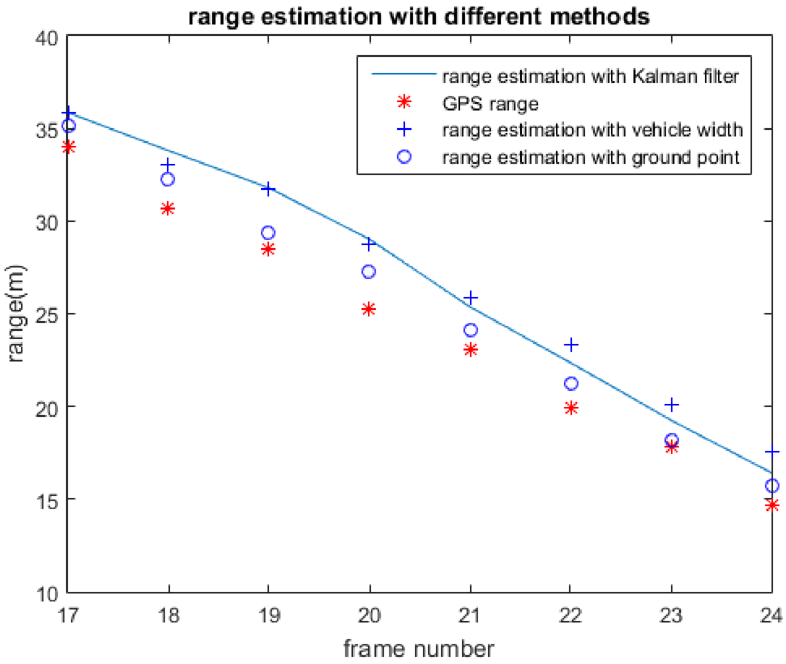


Figure 4.5 Range estimation of different methods in experiment 5.3

Overall, the Kalman filter does not provide better results in all situations and, therefore, the best results can be obtained by combining different methods, depending on external parameters (e.g. range). This means that, depending on the driving situation, the settings of the Kalman filter should be changed to obtain reliable results.

For example, in experiment 1.9, the SV is traveling on uneven road, provoking a change in the pitch angle: this change has a significant influence on the ground point method, as can be seen from Figure 4.6 in yellow circle. sometimes the closer points were estimated a larger distance which is an obvious wrong value, so in this case we should trust more on other methods.

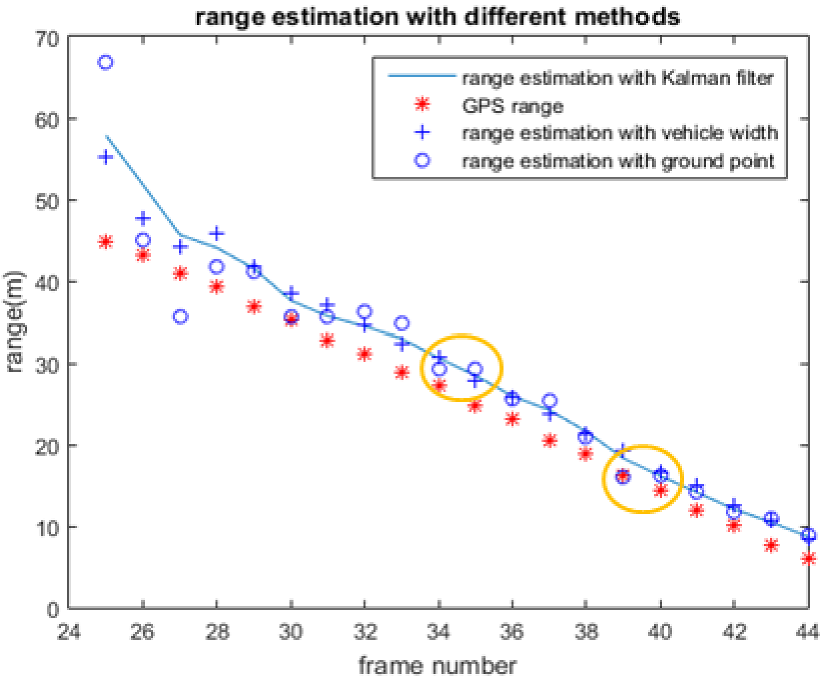


Figure 4.6 Range estimation of different methods in experiment 1.9

4.2 Semi-automatic tool discussion

One of the significant error is originated from the camera calibration. In the experiment, the GPS data, provide the correct position of the left headlight (tracked points for the calculation of optical flow vector length) of oncoming vehicle. In Figure 4.7, the correct position (from GPS data) is displayed with red star, large error appears due to the camera calibration. The error in quantified in Figure 4.8, illustrating the difference between the measurement result and the calculation result. This error effect all these four methods in this paper and it is unavoidable.

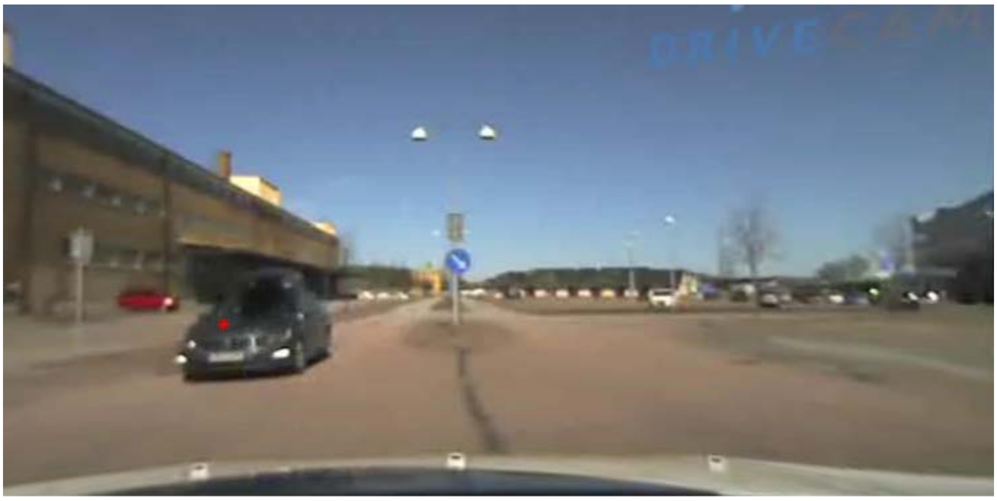


Figure 4.7 Tracked point position calculated from GPS data

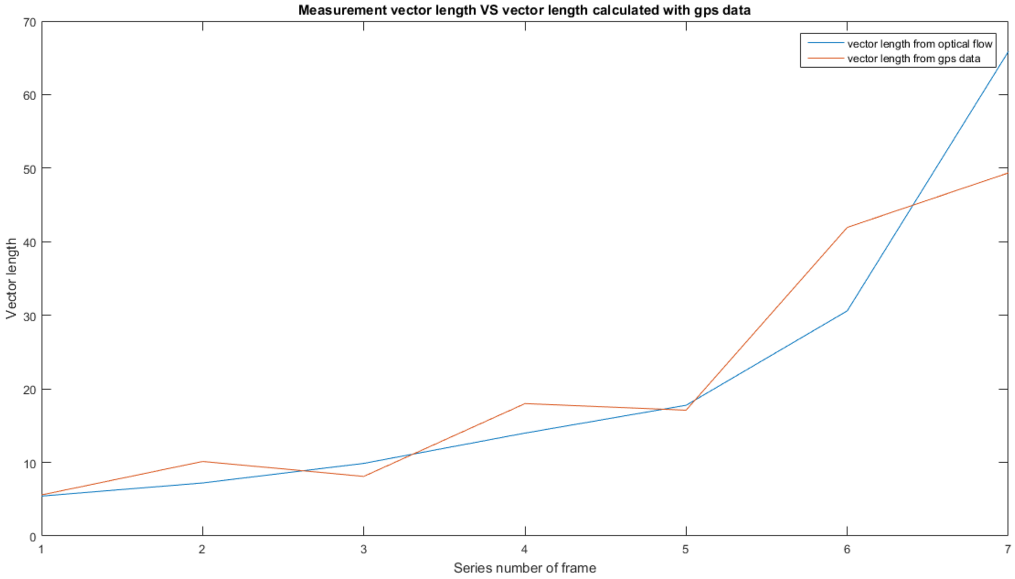


Figure 4.8 Comparison between measurement result (from optical flow) and calculation result (from GPS data)

The offset also cause error on the measurement results since they effect the benchmark. In real measuring process, the offset can be estimated with a relatively small error with the lane line.

Another error is caused by the manually selected point when the oncoming vehicle is in a small range. As the matching process starts from the vectors which are obtained in small range, the accurate manually selected point leads to a more accurate measurement result while the inaccurate one make a large error. As in Figure 4.9 (in this experiment, the oncoming vehicle is traveling with constant speed), it is clear that the inaccurate point causes an undesired change of the vector length.

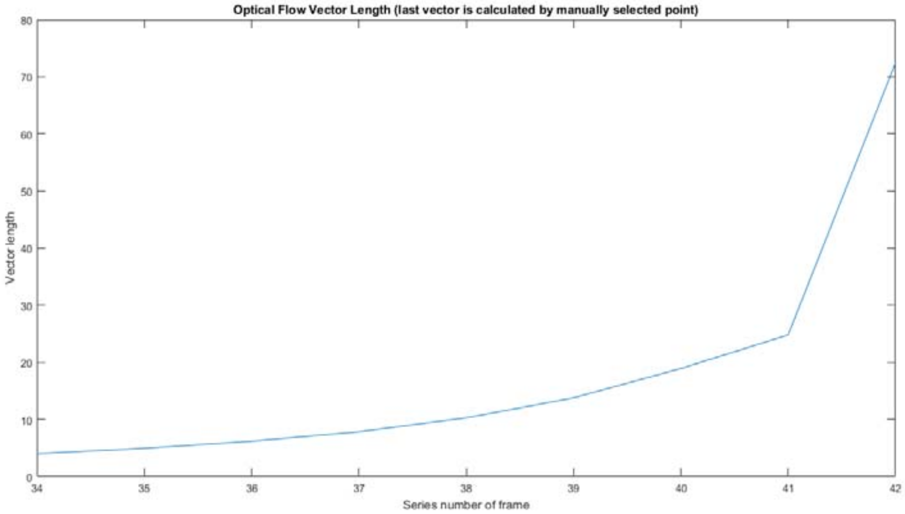


Figure 4.9 Effect of manually selected point

An improvement can be to increase the time duration between the two frames which are used to compute the optical flow vectors. A larger time duration makes larger change of the vector length and it shows a better resolution in the benchmark. Also it can fix the issue in experiment 2.9 since the direction of a longer vector in the frame suffers a less influence of the angle which is caused by floating.

5 Application

5.1 Data set formation

The data set used in this study is the EuroFOT naturalistic driving study, described in the introduction. The study collected data from 102 vehicles running for one year in the Gothenburg area. The data were collected continuously and several signals and videos were recorded during the experiment. In this thesis work, the video of the forward roadway, the speed and longitudinal and lateral acceleration were used to reconstruct the LTAP/OD scenario and analyse the driver behaviours.

In the data set, about 10 000 left turns though 19 intersections were automatically extracted and these cases were coded by trained annotators through reviewing videos of the driver and of the forward roadway. Among the information annotated, also the presence of other road users and drivers' glance behaviours were coded. From this 10000 left turn case, 331 videos were selected to meet the following criteria: 1) There were no traffic lights at the intersection. 2) At least, one oncoming vehicle was present and it went straight through the intersection. 3) The subject vehicle turned left after the oncoming vehicles passed. Due to the short time available, the analysis focused on 123 cases to apply the developed tool described in the earlier section.

5.2 Scenario reconstruction

A 2D model was built to reconstruct these LTAP/OD scenarios and obtain the PET.

5.2.1 Trajectory of subject vehicle

The trajectory of the subject vehicle is built by using the speed from CAN, and the lateral and longitudinal acceleration from inertial measurement unit (IMU) under the assumption that the heading angle γ is 0 at the beginning of the video..

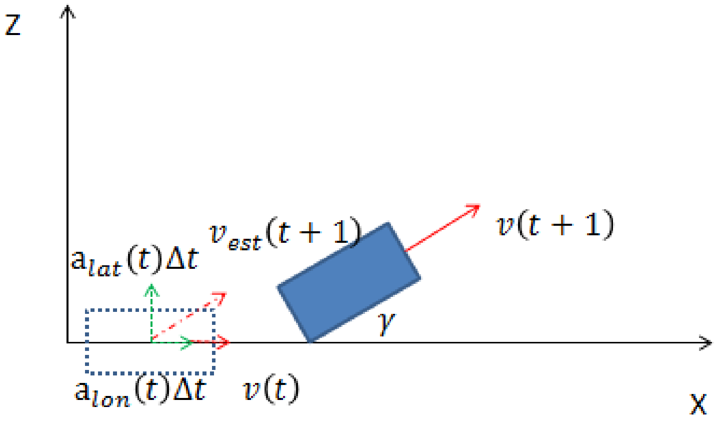


Figure 5.1 Heading angle estimation

Since the initial state is known, then the main problem is to calculate the heading angle in next sample time, according to Figure 5.1:

$$v_{lon}\,(\mathrm{t}+1) = v(t) + a_{lon}(t) * \Delta t \qquad (5.1)$$

$$v_{lat}\,(\mathrm{t}+1) = a_{lat}(t) * \Delta t \qquad (5.2)$$

$$\gamma(\mathrm{t}+1) = \gamma(\mathrm{t}) + arcsin\left(\frac{v_{lon}\,(\mathrm{t}+1)}{v(t+1)}\right) \tag{5.3}$$

In the equations, v is the resultant velocity, a_{lon} is longitudinal acceleration in the vehicle heading direction, and a_{lat} is lateral acceleration perpendicular to the vehicle heading direction.

Unfortunately, in the recording acquired during driving, the data from sensors presented some noise. Then, when the subject vehicle stopped, a small amount of velocity or acceleration fluctuated around zero. This error would influence the heading angle and cause a non-existent rotation of the subject vehicle. To address this problem, the value of speed and accelerations was set to zero when the vehicle was standing. On the other hand, the correction of speed and accelerations is difficult when the vehicle is not standing. To find out possible error, the speed was calculated by using the longitudinal and lateral components according to the formula below:

$$v_{est} = \sqrt{{v_{lon}}^2 + {v_{lat}}^2} \tag{5.4}$$

The v_{lon} and v_{lat} are obtained from equation 5.1 and 5.2. For each iteration, v_{est} was compared with v from CAN data. The case was rejected when there were five or more instances for which the absolute difference value $v_{est} - v$ was lager the 5km/h, before the subject vehicle entered the encroachment zone (about 3 seconds). Based on this procedure, 21 cases out of the 123 were rejected.

5.2.2 Trajectory of oncoming vehicle

Since the thesis focuses on how the drivers turn left after giving way to other vehicles, only the last oncoming vehicle (defined as POV) before the subject vehicle passed was annotated.

To obtain the trajectory of the POV, the speed and relative position between the SV and the POV are required. All this information was acquired from annotation with the tool developed within the thesis project.

To show an example of trajectory calculation, we can consider the scenario shown in Figure 5.2. Here, the subject vehicle is planning to perform a left turn and stops with an heading angle γ to wait for the oncoming vehicle. On the other hand, the POV is oncoming from opposite direction and planning to go straight. In this scenario, the offset is defined as the lateral distance between the two vehicles when both of them are parallel to the road. To calculate the offset, the coordinates of the ground point P in camera coordinate need to be obtained in world coordinates with the formulae below:

$$Z_p = Z_o + z_p * \sin(\gamma) - x_p * \cos(\gamma) \tag{5.5}$$

$$X_p = X_o + z_p * \cos(\gamma) - x_p * \sin(\gamma) \tag{5.6}$$

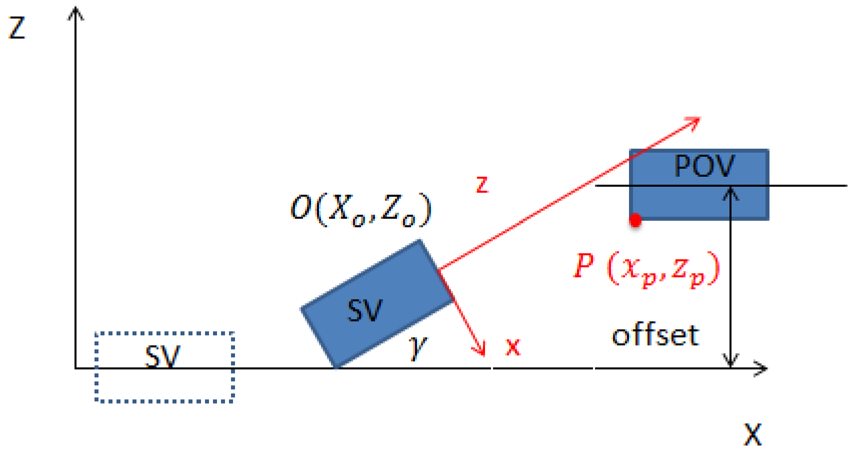


Figure 5.2 World coordinate and camera coordinate

Using the equations 5.5 and 5.6, the relative position of the two vehicles can be obtained and the offset equals X_p plus half width of the POV. During the process, special attention had to be placed on the synchronization due to the fact that the sample frequency of signals for the subject vehicle data was 10 HZ and the sample frequency for the image detection of POV was 12.5 HZ. As a consequence, the synchronization can be done exclusively every 0.4 s and the offset value can be chosen as the mean value of several synchronized points.

For the velocity of the POV, the average velocity is used in the 2D model. One reason to adopt this shortcoming is that obtaining accurate velocity for each frame is difficult since the POV only showed in the camera about 1 to 2 seconds. Another reason is that it does not influence much on PET calculation since the last synchronized point is always close to post encroachment zone. Even though POV has acceleration, the speed cannot change to a significant level in a short time since there were no hard braking for POV from videos.

5.2.3 PET calculation

In this thesis, the PET is always negative because the SV turned left after the POV (Figure 5.3).

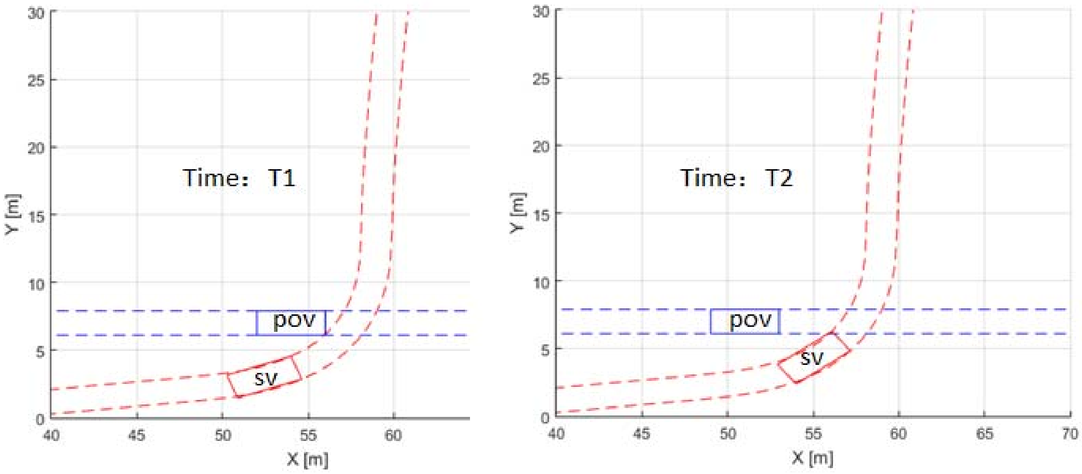


Figure 5.3 PET calculation in 2D model

The PET is calculated as T1-T2, and clearly the width and length of vehicles slightly influence the PET; these two parameters are set as the following values in the model, depending on the type of vehicle displayed in the image:

Table 5.1: estimated width and length of oncoming vehicle

Vehicle type	Estimated width (m)	Estimated length (m)
Light passenger cars	1.7	4
SUV (Sport utility vehicle)	1.9	4.5

In the EuroFOT data, the ego vehicle is always a Volvo V70 or Volvo XC70 and they are estimated as a light passenger car in Table 5.1. Buses and trucks were not present in the sample of maneuvers considered and, therefore, only light passenger vehicles and SUV were considered in Table 5.1.

5.3 Results

Overall, 102 cases were used in the following analysis, after eliminating the 21 cases with unreliable information about speed and accelerations (see section 5.2.1). The case regarded 37 drivers who had age varying between 37 to 57 years old and average age of M=47.5 years (SD=6.2 years). It is worth to note that three drivers did not consent to publish demographic information and, therefore, had to be excluded from the age calculation. All the drivers were experienced at the time of the data collection and they drove 25654.3 km per year on average (SD=6376.6 km).

The distribution of PET for the 102 cases is displayed in Figure 5.4: during their daily driving, the PET varies from -2s to -4.1s.

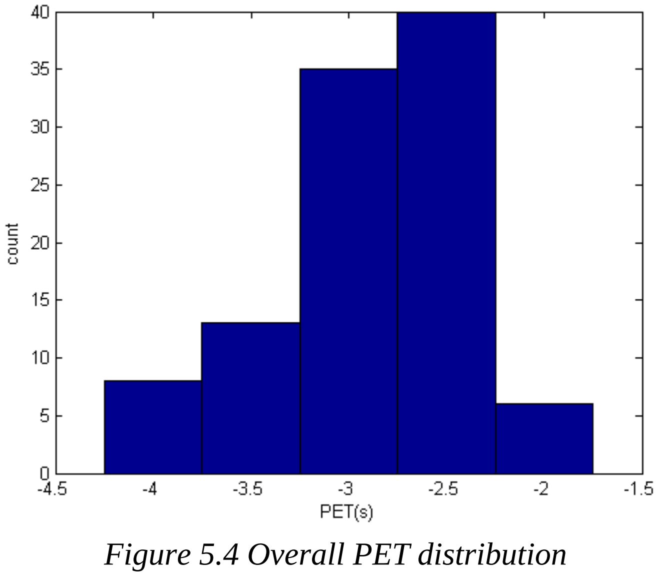


Figure 5.4 Overall PET distribution

A further analysis was performed to assess the influence of other variables on the PET distribution. In particular, two variables were considered: whether the SV fully stops and the offset between two vehicles. Based on the combination of those two variables, four sub scenarios were defined

:

1) The subject vehicle stop and the offset is more than 6m. 20 cases (19.6%) belong to this sub scenario and Figure 5.5shows the corresponding PET distribution.

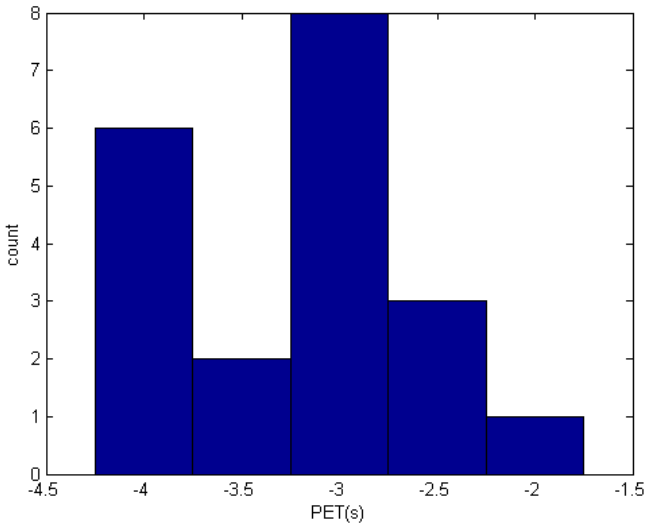


Figure 5.5 PET distribution in sub scenario 1

2) The subject vehicle stop and the offset is less than 6m. 23 cases (22.5%) belong to this sub scenario and Figure 5.6 shows the corresponding PET distribution.

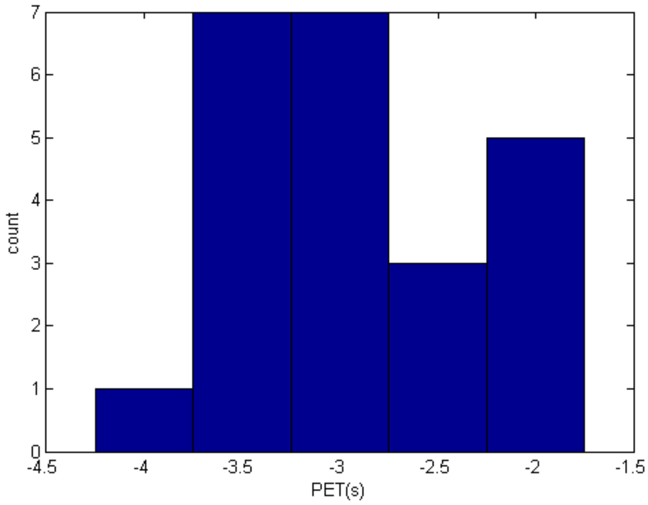
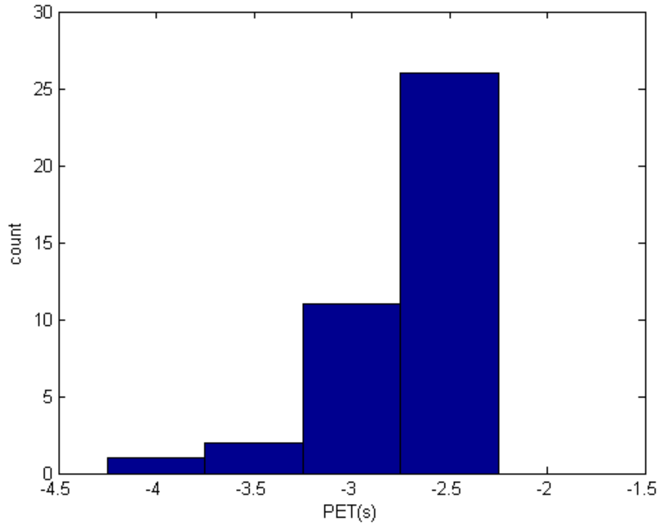


Figure 5.6 PET distribution in sub scenario 2

3) The subject vehicle decelerates but does not stop and the offset is more than 6m: 40 cases (39.2%) belong to this sub scenario and Figure 5.7 shows the corresponding PET distribution.



10

Figure 5.6 PET distribution in sub scenario 3

4) The subject vehicle decelerates but does not stop and the offset is less than 6m. 19 cases (18.6%) belong to this sub scenario and Figure 5.8 shows the corresponding PET distribution.

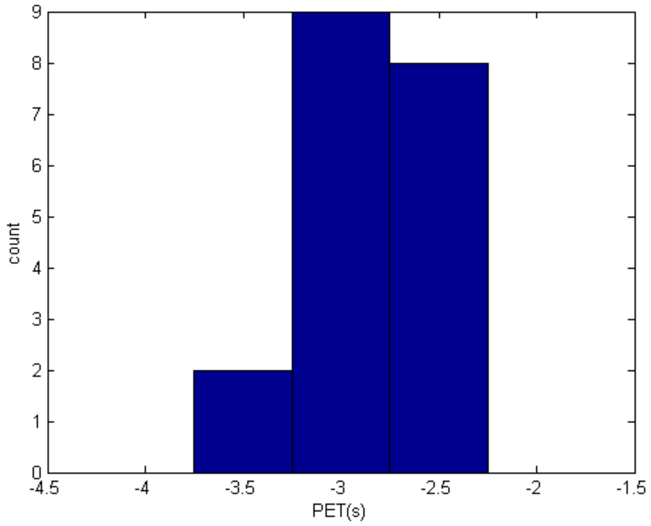


Figure 5.8 PET distribution in sub scenario 4

5.4 Implication

The EuroFOT data set shows that in 57.8% cases experienced drivers tend not to fully stop when they meet oncoming traffic at non-signal intersections. From forward videos, drivers choose to stop mostly because they meet a queue of oncoming vehicles and they have to stop and wait for longer time.

From the whole distribution of PET (Figure 5.4), in 73.5% cases drivers prefer to turn into the encroachment zone from 2.25 to 3.25 seconds after the pass of oncoming vehicle. Very few drivers chose absolute PET less than 2 seconds or more than 4 seconds.

Compare Figures from 5.5 to 5.8, most cases (6/7) where PET belong to [-4.25s, -3.75s] are present in the first sub scenario: this result is due to the fact that accelerating from fully stop takes long time for the SV driver. However few cases were observed in the first and second sub-scenarios where PET is around -2 s (Figure 5.5 and 5.6). The forward videos were checked for these events and showed that the SV drivers accelerated early before the last oncoming vehicle came into the intersection after waiting for a long time.

In Figure 5.5, 80% of PET belongs to [-4.25,-2.75] while in Figure 5.6, 73.9% PET is between -3.75 s and -2.25 s. This comparison shows that drivers have shorter absolute PET when the offset between two vehicles is smaller and the subject vehicle stops. However, when it comes to the comparison between Figure 5.7 and Figure 5.8 (sub-scenarios in which the SV does not stop), drivers do not seem to accept shorter absolute PET due to the offset: PET around -2.5 s cases account for 65% in the third sub scenario while in the fourth sub-scenario this percentage decreases to 42.1%. To explain this result, the average speed of the subject vehicle was calculated by considering the time duration for which the speed of oncoming vehicles was calculated. The average speed of subject vehicle for PET in [-3.25s,-2.75s] cases in sub-scenario 3 is 2.8 m/s while the value is 2.1 m/s in sub-scenario 4. This result illustrates that drivers tend to accept a larger velocity when they have a large offset.

From the comparison between Figure 5.5 and 5.7 (where the offset is larger than 6 m), it is possible to assert that drivers prefer a smaller absolute PET in non-stop scenarios: the PET range changed from [-4.25s,-2.75s] (80%) in Figure 5.5 to [-3.25,-2.25] (92.5%) in Figure 5.7. Similar trend can be seen from Figure 5.6 and 5.8.

The results of the quantitative analysis have implications for the design of autonomous driving: the drivers will expect autonomous vehicles to behave as human driver, especially in a mixed traffic environment [23]. Then, an autonomous vehicle turning left, after the POV has cleared the way, should adopt values of PET that make the drivers feel comfortable. Based on the results of this work,PET equals -4s and -2s can be considered as comfortable boundaries: this means that most drivers would feel comfortable when autonomous vehicles turn into the encroachment zone in a range defined between 2 and 4 s after the last oncoming vehicle leaves the zone. However, the PET can be adjusted depending on the driving situations, as indicated by the sub-scenarios considered, which shown different values of PET.

However, not only the PET should be taken into account to guarantee the comfort of drivers during left-turn manoeuvres: other parameters such as lateral and longitudinal accelerations are relevant, even more since they relate to the PET.

6 Conclusion

Three methods are discussed and implemented in the thesis: ground point methods, vehicle width method and optical flow methods. Each method has its strengths and drawbacks. In addition a Kalman filter were used to merge data from the three methods.

The ground point method is more accurate at shorter range but it relies much on the pitch angle calibration. The vehicle width method obtains the range and rang rate with an estimation of real vehicle width in meters a 0.1m error of width estimation can lead of 5% error of range estimation. This method is quite robust to pitch angles but at shorter range(when the POV is about to pass), the accurate vehicle width in pixels on image is hard to be obtained thus increasing the error of range estimation. Optical flow method provides more possibility to track the POV and match relative range and speed with benchmark in an automatic way. However, this method is sensitive to the lateral offset between the vehicles, a small angle between the orientation and the trajectory of POV can lead to large changes of offset, and thus the matching result is not ideal. Another important reason is because the variation of camera positions leads to the changes of length and direction of the optical flow.

The Kalman filter provided the possibility to integrate three methods with a constant acceleration vehicle model. Different measurement matrix were used at different ranges, and were set according to the experiment result of each method. The filter provided a more accurate and stable range estimation at range between 10 to 40 meters and improve the speed estimation much. Actually, the settings of Kalman filter parameters could be optimized in a real driving situations, for example, if the pitch angle of the camera changes due to the hard braking or uneven road, the ground point method and optical flow should not be trusted and the output of Kalman filter should rely much on vehicle width method instead.

Finally, in order to apply the manual tool at its good range estimation area(10m-30m), a LTAP/OD driving scenario was selected and how drivers turn left after yielding to the POV was analyzed. Results show that most drivers feel comfortable to turn into the encroachment zone in a range between 2s to 4s after the last oncoming vehicle leaves the zone.

7 Future work

Due to the loss of GPS signals, some scenarios (runs) were lost and thus we fail to observe the how large heading angles (e.g.30 degree) influence different methods. In addition, because the limited numbers of experiment runs, it was not possible to understand whether the speed estimation error increases with relative speed. Furthermore, different methods have their intrinsic limitations and their performance at undesired situations should be tested. Thus extra experiment can be used to obtain the measurement variance matrix which can provide more information to Kalman filter.

More advanced image analysis methods can be applied to improve the manual (semi-automatic) tool and make it an automatic tool. For example, image segmentation can be used to obtain the ground lines and the vehicle width in pixels. This can be a challenging work due to the resolution and compression of the image in this particular application.

As for the semi-auto tool, a better camera calibration, higher quality of can definitely help; however, they are out of the scope of the thesis. One possible improvement which can be implemented in this semi-automatic method could be increasing the time interval when computing optical flow vectors. First, larger time interval leads to a benchmark with high resolution, which can improve the accuracy of the matching process. Second, the direction of optical flow vectors may be more close to the theoretical ones if the vectors are generated with a larger time interval. In this case, the problem caused by rough road can be solved to some degree. The disadvantage is that the method will waste more frames since we cannot calculate the optical flow vectors of the last few frames.

Finally, when it comes to the application part, more parameters other than PET should be considered to define the comfortable zone of the left turn behavior, such as the lateral and longitudinal acceleration.

8 References

[1] WHO. (2015): Global Status Report on Road Safety.

http://www.who.int/violence_injury_prevention/road_safety_status/2015/en/

[2] ERSO. (2015), Traffic safety basic facts 2015 – Main Figures, available at: http://ec.europa.eu/transport/road_safety/pdf/statistics/dacota/bfs2015_main_figures.pdf

[3] Markoff, J. (2010). Google cars drive themselves, in traffic. The New York Times, 10(A1), 9.

[4] Li, T. H. S., Chang, S. J., & Chen, Y. X. (2003). Implementation of human-like driving skills by autonomous fuzzy behavior control on an FPGA-based car-like mobile robot. Industrial Electronics, IEEE Transactions on, 50(5), 867-880.

[5] Najm, W. G., Smith, J. D., & Smith, D. L. (2001). Analysis of crossing path crashes (No. HS-809 423,).

[6] Bärgman, J., Smith, K., & Werneke, J. (2015). Quantifying drivers' comfort-zone and dread-zone boundaries in left turn across path/opposite direction (LTAP/OD) scenarios. Transportation Research Part F: Traffic Psychology and Behaviour, 35, 170-184.

[7] Chan, C. Y. (2006). Characterization of driving behaviors based on field observation of intersection left-turn across-path scenarios. Intelligent Transportation Systems, IEEE Transactions on, 7(3), pp 322-331

[8] Nobukawa, K., Barnes, M., Goodsell, R., & Gordon, T. (2009). Reconstruction of Vehicle Trajectories for Intersection Conflict Analysis Using Vehicle-Based Sensors. International Association for Vehicle System Dynamics.

[9] Dingus, T.A., Klauer, S.G., Neale, V.L., Petersen, A., Lee, S.E., Sudweeks, J., Perez, M.A., Hankey, J.,Ramsey, D., Gupta, S., Bucher, C., Doerzaph, Z.R., Jermeland, J., & Knipling, R.R. (2006). The100-Car Naturalistic Driving Study, Phase II – Results of the 100-Car Field Experiment. Report no: DOT HS 810 593.

[10] Victor, T., Dozza, M., Bärgman, J., Boda, C. N., Engström, J., Flannagan, C., ... & Markkula, G. (2015). Analysis of naturalistic driving study data: Safer glances, driver inattention, and crash risk (No. SHRP 2 Report S2-S08A-RW-1).

[11] SAFER. (2008), Test Site Sweden Field Operational Test, available at: http://www.chalmers.se/safer/EN/projects/traffic-safety-analysis/tss-naturalistic-field/downloadFile/attachedFile_f0/Report__TSS-FOT?nocache=1212679059.62

[12] EuroFOT, available at: http://www.eurofot-ip.eu/en/about_eurofot/objectives/ .

[13] Klauer, S. G., Perez, M., & McClafferty, J. (2011). Chapter 6 - Naturalistic Driving Studies and DataCoding and Analysis Techniques. In E. P. Bryan (Ed.), Handbook of Traffic Psychology (pp. 73-85). San Diego: Academic Press.

[14] Shinar, D., & Oppenheim, I. (2011). Review of models of Driver Behaviour and development of a Unified Driver Behaviour model for driving in safety critical situations. In Human Modelling in Assisted Transportation. Springer Milan. pp. 215-223

[15] Carsten, O. (2007). From driver models to modelling the driver: what do we really need to know about the driver?. In Modelling driver behaviour in automotive environments (pp. 105-120). Springer London.

[16] Keith K, Trentacoste M, Depue L, Granada T, Huckaby E, Ibarguen B, Kantowitz B, Lum W,Wilson T (2005) Roadway human factors and behavioral safety in Europe. Federal HighwayAdministration, report no. FHWA-PL-05–005 US. Department of Transportation, Washington, DC.

[17] Bradski, G. and Kaehler, A., (2008). Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc.". 372 pp.

[18] Corke, P. (2011). Robotics, vision and control: fundamental algorithms in MATLAB (Vol. 73). Springer, pp 261-262

[19] Zhang, Z. (2000). A flexible new technique for camera calibration. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22(11), pp1330-1334.

[20] Heikkila, J., & Silvén, O. (1997). A four-step camera calibration procedure with implicit image correction. In Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on (pp. 1106-1112). IEEE.

[21] Bärgman, J., Werneke, J., Boda, C. N., Engström, J., Smith, K., & Volvo, A. B. (2013). Using manual measurements on event recorder video and image processing algorithms to extract optical parameters and range. In Proceedings of the Seventh International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design (pp. 177-183).

[22] Stein, G. P., Mano, O., & Shashua, A. (2003, June). Vision-based ACC with a single camera: bounds on range and range rate accuracy. In Intelligent vehicles symposium, 2003. Proceedings. IEEE (pp. 120-125). IEEE.

[23] Cui, J., Liu, F., Li, Z., & Jia, Z. (2010, June). Vehicle localisation using a single camera. In Intelligent Vehicles Symposium (IV), 2010 IEEE (pp. 871-876). IEEE.

[24] Hong, J. G. (1984). Gray level-gradient co-occurrence matrix texture analysis method. Acta Automatica Sinica, 10(1), 22-25.

[25] Xu Yuneng, Zhu Xichan, Ma Zhixiong, Li Lin.(2014)Three-dimensional Reconstruction of the Road Ahead of Vehicle Based on Mono-vision. ,Automobile Technology. (pp.48-52).

[26] Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. In Image analysis (pp. 363-370). Springer Berlin Heidelberg.

[27]Farnebäck, G. (2000). Fast and accurate motion estimation using orientation tensors and parametric motion models. In Pattern Recognition, 2000. Proceedings. 15th International Conference on (Vol. 1, pp. 135-139). IEEE.

[28] Gadepally, V. N. (2013). Estimation of Driver Behavior for Autonomous Vehicle Applications (Doctoral dissertation, The Ohio State University).

[29] Jaesik Choi. (2012). Realtime On-Road Vehicle Detection with Optical Flows and Haar-Like Feature Detectors (Department of Computer Science, University of Illinois at Urbana-Champaign)

[30] Andrew Burton and John Radford (1978). Thinking in Perspective: Critical Essays in the Study of Thought Processes. Routledge. ISBN 0-416-85840-6.

[31] Jianbo Shi and Carlo Tomasi.(June 1994), IEEE Conference on Computer Vision and Pattern Recognition (CVPR94) Seattle

[32] Available at http://www.dewesoft.com/

[33] Available at http://www.automobiledimension.com/

[34] B.D.Lucas and T.Kanade. An iterative image registration technique with an application to stereo vision. In Proc. Of the 7th IJCAI, pages 674-679, Vancouver, Canada, 1981.

[35] SHRP2 (2016). SHRP2 Researcher Dictionary for Video Reduction Data. Version 3.4.available at :

https://vtechworks.lib.vt.edu/bitstream/handle/10919/56719/V4.1_ResearcherDictionary_for_VideoReductionData_COMPLETE_Oct2015_10-5-15.pdf?sequence=1&isAllowed=y

[36] Available at www.lytx.com

[37] More B B, Pathade M. Moving Object Detection, Tracking, Object Counting and Speed Measurement[J], 2013