UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

REAL-TIME VEHICLE MEASUREMENT USING DIGITAL IMAGE CORRELATION

by

Devin Kent Johnson

Submitted in partial fulfilment of the requirements for the degree

Master of Engineering (Mechanical Engineering)

in the

Department of Mechanical and Aeronautical Engineering
Faculty of Engineering, Built Environment and Information Technology

at the

UNIVERSITY OF PRETORIA

November 2017

SUMMARY

REAL-TIME VEHICLE MEASUREMENT USING DIGITAL IMAGE CORRELATION

by

Devin Kent Johnson

Supervisor(s):	Dr. T.R. Botha and Prof. P.S. Els
Department:	Mechanical and Aeronautical Engineering
University:	University of Pretoria
Degree:	Master of Engineering (Mechanical Engineering)
Keywords:	Computer Vision, Side-slip angle, Rut measurements

The tyre-road interface is one of the most important research topics in the field of vehicle dynamics. This is largely due to all the vehicle excitation forces (besides aerodynamic forces) being generated at this interface. There are many parameters which govern the generation of tyre forces, of which the side-slip angle is of utmost importance.

Vehicle side-slip angle can be used as a measure of vehicle stability. Stability control schemes require side-slip angle and typically estimate this parameter instead of using a direct measurement. The relationship between tyre lateral force and tyre side-slip allows the lateral force generated by the tyre to be determined from the tyre side-slip angle. Therefore, real-time measurement of side-slip angle is important in tyre research and vehicle stability. Solutions exist to measure the side-slip angle, however, do not perform well at low speeds or over rough terrain and are prohibitively expensive.

In terramechanics, tyre soil deformation in the form of rut depth is a widely researched topic as it can be used as a measure of the vehicle's ability to traverse the terrain, estimate soil characteristics and for vehicle environmental impact studies. Currently, these measurements are labour intensive and are typically conducted by hand. Other solutions exist however they are developed for road use and are prohibitively expensive. The research field would, therefore, benefit largely from online rut depth measurements.

Digital Image Correlation is the mathematical process of tracking changes in digital images. The

development of robust algorithms and ease of implementation has allowed many fields to be adapt this non-contact based, optical technique for application-specific measurements. Previous studies (Botha, 2015) have proved DIC to be a viable candidate for measuring the side-slip angle and rut depth that overcome current measuring hurdles. However, the analysis was conducted in post-processing from pre-recorded footage due to the large computational expense of the image processing. This opens the opportunity to adapt and optimise these techniques to achieve real-time processing speeds required for these camera-based sensors.

This study builds on Botha (2015) with a real-time implementation which allows for online measurements to be made using inexpensive, off-the-shelf cameras with dedicated software. This will eventually provide systems such as ABS, stability control schemes and semi-active suspension with real time vehicle side-slip angle and rut depth with a cost-effective camera-based sensor. The aim of the present study is to develop and test two systems that can measure the side-slip angle and rut depth in real-time.

The side-slip angle is measured using a single camera pointing down on the terrain and digital image correlation. It is shown to measure accurately and in real-time. The sensor is tested on a flat surface using a rig that allows for validation.

The rut depth is measured using multiple cameras pointing at the terrain and digital image correlation to create a 3D map of the terrain. Three methods for determining the rut depth from the 3D map is investigated, with varying degree of accuracy and processing speed.

ACKNOWLEDGEMENTS

I would personally like to thank the following persons for the help and continuous support throughout this investigation:

- All my VDG colleges that created an environment where I could be myself and allowed me to grow to my full potential.
- To Dr. Theunis Botha who guided me throughout this investigation and was always willing to help when needed.
- Prof. S. Els who has given me this opportunity in VDG.
- My girlfriend, Monique Geyer for all the continuous love and support.
- My parents for giving me the opportunity and making sure I have a roof over my head and a full stomach.
- My Uncle, Stuart Martin, for his continuous support and technical advice.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
ADAS	Advanced Driver Assist Systems
ABS	Anti-Lock Braking System
ADC	Analogue to Digital Converter
BM	Block Matching
BRISK	Binary Robust Invariant Scalable Keypoints
CCD	Charge Coupled Device
CMOS	Complementary Metal Oxide Semiconductor
CG	Centre of Gravity
CI	Cone Index
CPU	Central Processing Unit
CV	Computer Vision
DAQ	Data AcQuisition device
DDR	Double Data Rate
DIC	Digital Image Correlation
DoG	Difference of Gaussian
FOV	Field of View
FPS	Frames Per Second
GPIO	General-purpose input/output
GPS	Global Position Systems
GPU	Graphic Processing Unit
HOG	Histogram of Oriented Gradients
ISO	International Standards Organisation
KITTI	Karlsruhe Institute of Technology and Toyota Technological Institute
KLT	Kanade-Lucas-Tomasi
LiDAR	Light Detection and Ranging
LK	Lucas-Kanade

Abbreviation	Description
NCC	Normalised Cross Correlation
OpenCV	Open Computer Vision
ORB	Oriented FAST and Rotated BRIEF
RANSAC	Random Sample Consensus
RMS	Root Mean Square
SAD	Sum of Absolute Differences
SGBM	Semi-Global Block Matching
SI	International System of Units
SIFT	Scale-Invariant Feature Transform
SSD	Sum of Squared Differences
STD	Standard Deviation
SURF	Speeded-Up Robust Features

NOMENCLATURE

Symbol	Description	Units
d	Disparity	$Pixel$
d_i	Distance between image plane and lens	mm
d_o	Distance between the object and the lens	mm
e	Epipolar point	
e'	Epipolar point	
f	Lens focal length	mm
f_x	Combined focal length in lateral direction	$Pixel$
f_y	Combined focal length in vertical direction	$Pixel$
h_i	Height of the object	$Pixel$
h_o	Height of the object on the lens	$Pixel$
k_i	i^{th} Radial distortion correction parameter	
p	Projection of P on the plane Π	
p'	Projection of P' on the plane Π'	
p_i	i^{th} Tangential distortion correction parameter	
t	Translational distance between cameras	mm
u	Pixel motion in x-direction	$Pixel/s$
u_o	x location of the principal point	$Pixel$
v	Pixel motion in y-direction	$Pixel/s$
v_o	y location of the principal point	$Pixel$
x	Pixel coordinate in the x-direction	$Pixel$
x'	Pixel coordinate in the x-direction on the corresponding sensor	$Pixels$
x_c	Corrected x-coordinate (from distortion)	$Pixel$
x_r	Radial distortion model in the x-direction	
x_t	Tangential distortion model in the x-direction	
y	Pixel coordinate in the y-direction	$Pixel$
y_c	Corrected y-coordinate (from distortion)	$Pixel$
y_r	Radial distortion model in the y-direction	
y_t	Tangential distortion model in the y-direction	
D	Pixel displacement	$Pixel$

Symbol	Description	Units
$H(x,y)$	Hessian matrix of Pixel Intensities	
I	Pixel intensity on image	
I'	Pixel intensity on corresponding image	
$\boldsymbol{I}_t$	Image at time t	
I_t	Image pixel gradient at time t	
I_x	Image pixel gradient in the x-direction	
I_y	Image pixel gradient in the y-direction	
N	Window size	
O	Origin of reference frame	
S_x	Sensor size in the x-direction	$Pixel/mm$
S_y	Sensor size in the y-direction	$Pixel/mm$
$\boldsymbol{V}$	Vehicle velocity vector	m/s
V_x	Pixel velocity in x-direction	$Pixel/s$
$\boldsymbol{V}_x$	Vehicle velocity in x-direction/longitudinal direction	m/s
V_y	Pixel velocity in y-direction/lateral direction	$Pixel/s$
$\boldsymbol{V}_y$	Vehicle velocity in y-direction	m/s
X	x-coordinate of the object	mm
Y	y-coordinate of the object	mm
Z	z-coordinate of the object or distance to the object	mm
α	Tyre side-slip angle	deg
α_f	Front tyre side-slip angle	deg
α_r	Rear tyre side-slip angle	deg
β	Vehicle side-slip angle	deg
$\dot{\beta}$	Vehicle side-slip angle rate	deg/s
δ	Distortion model	
Δx	Change in vehicle displacement in x-direction	m
Δy	Change in vehicle displacement in y-direction	m
Π	Left Image plane	
Π'	Right Image plane	

CHAPTER 1 INTRODUCTION

In off-road vehicle dynamics research, the interaction between the tyre and the terrain it is travelling on is one of the most studied areas. This is because it is the only region where external forces act on the vehicle in the absence of aerodynamic forces.

Modern passenger vehicles rely on complex control systems known as Advanced Driver Assist Systems (ADAS) to assist drivers in avoiding accidents. These systems, use features such as anti-lock brake systems (ABS) and electronic stability control schemes (ESC) to enhance the safety of the occupants and require a large combination of parameters. One of these parameters is the vehicle side-slip angle. Typically control strategies require vehicle side-slip angle as an input (Rajamani, 2005). Accurate real-time measurements of vehicle side-slip angle will provide invaluable information for significantly improving vehicle dynamic control. Tyre side-slip angle is also an important parameter in vehicle dynamics as there is a direct relationship between the lateral force and the tyre side-slip angle. Accurate side-slip angle measurements are therefore a necessity for the characterisation of tyres which is required for vehicle control and vehicle modelling.

Side-slip angle is notoriously difficult to measure. Commercial sensors to measure the side-slip angle do exist, but are developed for smooth, hard roads or require large dynamic excitation. These solutions prohibitively expensive and perform poorly over rough, off-road terrain and at low speeds that are typically experienced during terramechanic applications (Botha and Els, 2012). Tyre side-slip angle is not typically measured in field tests, however, are measured during tyre characterisation. The tyres are typically mounted to a rig where the side-slip angle is accurately controlled.

Another important research topic in off-road vehicle dynamics is vehicle mobility on deformable terrains. When vehicles travel over deformable terrains, such as soil or snow, they compact the terrain and form ruts. This significantly affects the vehicle‘s ability to traverse the terrain and can have a detrimental effect on the environment. The rut depth can be used as a measure of vehicle mobility on the terrain. It may also be useful to estimate soil characteristics such as soil strength and can be used as a measure of the disturbance on the terrain for environmental studies. Current methods to measure rut depth are labour intensive and generally only possible at a few discrete locations (Shoop et al., 2013). They are also difficult to synchronise with other measurements.

Digital image correlation (DIC) is the process of correlating pixels or regions of pixels between digital images. Typically contrast patterns on the image, naturally occurring or artificially generated, are used to measure changes in image sequences using non-contact optical techniques. These techniques can be tailored to the desired application making it an attractive measuring technique and has been adopted in various fields for measurement purposes. The depth of a scene can also be determined using DIC and multiple cameras, known as stereography or stereo-vision (Grimson, 1981). This enhances the capabilities of DIC as it provides the ability to track points in space as well as determine their 3D coordinates. Recent years has seen large development in DIC and has lead to robust algorithms that are relatively easy to implement and have produced better performing cameras at lower costs. These techniques can be applied to low cost cameras making the techniques inexpensive to implement. The use of DIC in vehicle dynamics is almost none-existent, especially with real-time implementations. Moser et al. (2010) used DIC to perform laboratory tyre testing to determine material properties of the tyre as well as tyre deformation under loading. DIC has also been used in vehicle crash tests, where stereography was used to perform automated deformation analysis of the vehicle (Rentschler and Uffenkamp, 1992).

Botha (2015) proved that accurate side-slip angle and terrains profiles (from which rut depth can be extracted) can be made using DIC as an alternative to current measuring techniques. DIC techniques are computationally expensive and the analysis was conducted in post-processing from pre-recorded footage. This opens an opportunity to develop camera-based sensors that will measure the side-slip angle and rut depth in real-time. This study builds on Botha (2015) by implementing these algorithms in real-time as a new camera-based sensor that will eventually provide systems such as ABS, traction control, four-wheel drive and torque vectoring with these parameters using cost-effective, off-the-shelf cameras and lenses. DIC is used to address the shortcomings of current measuring solutions by developing camera-based sensors that give improved performance on hard surfaces that are also able to measure on rough, off-road terrains.

1.1 PROBLEM STATEMENT

This study aims to answer the following question:

1. Is it possible to adapt and optimise the off-line techniques developed by Botha (2015) to enable real-time measurements of side-slip angle and rut depth?

1.2 OVERVIEW OF STUDY

Chapter 2 is a literature study to give a brief summary of the relevant literature required for this investigation. The parameters being measured, side-slip angle and rut depth, are discussed. Current measuring techniques and their short-comings are discussed. The use of DIC is discussed and proved to be a viable alternative measurement technique. DIC requires digital images and therefore the chapter further discusses the basics of obtaining digital images as well as fundamental camera models used for calibration and lens distortion removal. The side-slip angle uses an optical flow algorithm that is discussed and mathematically described. The rut depth sensor requires full DIC and uses a stereo-vision rig to recover depth from a scene. The mathematical models used to achieve this are discussed as well as the algorithm used for full DIC.

Chapters 3 discuss the development of the side-slip angle sensor, as well as the testing and validation of the results.

Chapter 4 discusses the implementation of the rut depth sensor. The sensor was validated by measuring the depth of a preformed rut and compared to manual measurements. Various methods for extracting the rut depth are evaluated.

Chapter 5 concludes the investigation and gives the recommendations on how it could be improved for future work.

CHAPTER 2 LITERATURE STUDY

2.1 CHAPTER OBJECTIVES

This chapter aims to summarise the relevant literature with the aim to develop a research question and project plan. The chapter starts by exploring the concepts of the side-slip angle and rut depth. This includes the relevance of these parameters to the vehicle dynamics field, current measuring techniques and how digital image correlation could be used to perform these measurements.

DIC requires digital images and therefore the fundamental basics of acquiring these images are discussed. These images have to be preprocessed before they can be used by the algorithms. The mathematical models required for the preprocessing are discussed. These models require parameters that are unique to the camera sensor and lens but can be determined through calibrating the cameras. The calibration process to determine these parameters is also briefly discussed.

Both side-slip angle and rut depth require their own image correlation algorithms that are discussed in detail. For side-slip angle, feature tracking is discussed and for rut depth measurements stereo-vision is discussed.

2.2 VEHICLE DYNAMICS

The following section aims to discuss two important vehicle parameters i.e. side-slip angle and rut depth that are measured during this investigation. A brief description of each parameter is provided and the possibility of obtaining it in real-time, is discussed.

2.2.1 Side-slip angle

In the field of vehicle dynamics, the side-slip angle can be separated into two variables, namely tyre side-slip angle, α, and vehicle side-slip angle, β. Figure 2.1 shows a vehicle undergoing vehicle side-slip. The vehicle is moving in the direction of the velocity vector, $\boldsymbol{V}$, however, is oriented along

the longitudinal component of the velocity vector, $\boldsymbol{V}_x$. Vehicle side-slip angle, β, is the angle between $\boldsymbol{V}$ and $\boldsymbol{V}_x$. β can therefore be calculated using Equation 2.1. If the change in vehicle displacement in

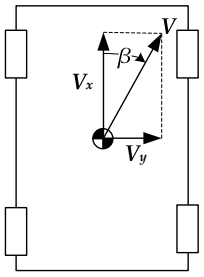


Figure 2.1. Vehicle side-slip angle

the longitudinal and lateral directions can be determined simultaneously, β can be determined using Equation 2.2.

$$\beta = \arctan(\frac{\boldsymbol{V}_y}{\boldsymbol{V}_x}) \tag{2.1}$$

$$\beta = \arctan(\frac{\Delta y}{\Delta x}) \tag{2.2}$$

Similarly, tyre side slip angle, α, is the angle between the direction the tyre is moving in and the direction it is oriented. Figure 2.2 shows a simple bicycle model representation of a vehicle undergoing side-slip where the front and rear tyre side-slip angles, α_f and α_r, as well as the vehicle side-slip angle, β, are indicated (Gillespie, 1992).

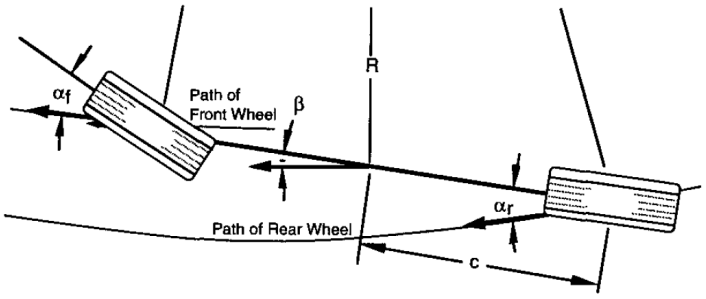


Figure 2.2. A bicycle model of a vehicle and its tyres experiencing side-slip (Gillespie, 1992)

Modern passenger vehicles rely on complex control systems to assist drivers in avoiding accidents or at least to limit damages in the event of a collision. These systems, known as Advanced Driver Assist Systems (ADAS) use features such as anti-lock brake systems (ABS) and electronic stability control schemes (ESC) to enhance the safety of the occupants. Typically these systems require a

large combination of sensors to determine the state of the vehicle. A vehicle model then uses the information to predict the behaviour of the vehicle within normal driving conditions. The predicted model is compared to the actual vehicle behaviour and then corrects any deviation using ADAS features (Rajamani, 2005). One of these parameters are vehicle side-slip angle, β. Vehicle side-slip angle can be used as a measure of the vehicle‘s handling and stability and is, therefore, a valuable parameter in vehicle dynamics. Inagaki et al. (1995) demonstrated that the vehicle side-slip angle and it's derivative, $\dot{\beta}$, offer better insight to the vehicle‘s lateral stability as compared to yaw-rate. Chung and Yi (2006) found that using a stability control scheme that was based on side-slip angle resulted in an overall improved vehicle performance on a virtual test track. Lu et al. (2016) and Barbarisi et al. (2009) used a combination control of vehicle side-slip angle and yaw rate to enhance vehicle performance.

Side-slip angle is notoriously difficult to measure. Sensors to measure the side-slip angle do exist but have been developed for smooth, hard roads and perform poorly over rough, off-road terrain and at low speeds. This is unsuitable for terramechanic applications which generally occur under these conditions. Due to prohibitive costs of such sensors, the field of vehicle dynamics has opted to instead estimate the vehicle side-slip angle rather than using a direct measurement. These methods use sensor fusion techniques that combine sensors such as accelerometers, GPS, and rate gyroscopes to estimate the side-slip angle (Bevly et al. (2006), Botha and Els (2012), Nishio et al. (2001) and Hac and Simpson (2000)). One major disadvantage of these methods is that the sensors are inherently noisy and therefore require large sensor excitations, such as high-speed dynamic manoeuvres or large side-slip angle, to provide accurate results. These methods are also prone to integration drift. As a result, these methods are unsuitable for off-road scenarios, where tests occur at low speeds and experience high levels of noise due to terrain roughness.

Tyre side-slip angle is also an important parameter in vehicle dynamics. Bakker et al., 1987 showed that there is a direct relationship between the lateral force generated by the tyre and the tyre side-slip angle. The “Magic Formula” is an empirical fit to best describe this relationship. Accurate side-slip angle measurements are therefore a necessity for the characterisation of tyres that is used for evaluating dynamic manoeuvres, performing tests on tyre test tracks or vehicle model validation. Typically these tests are conducted in a laboratory where the tyre is mounted on a rig, such as a drum test or flat tracks where the side-slip angle is accurately controlled. These setups are generally large and expensive, creating the need for a cheaper alternative. These setups also do not represent realistic friction surfaces for road and off-road tests. Another alternative setup is a tyre test trailer which is less expensive and

can conduct testing on real terrains. These test trailers however require direct measurement of the tyre side-slip angle.

The Kistler Correvit S-HR is a commercially available side-slip angle sensor, and can measure tyre side-slip angle or vehicle side-lip angle depending on where the sensor is mounted. It uses a combination of the Doppler effect and an absolute measuring method for determining the side-slip angle. It is has a range of $\pm$ 40deg slip angle, speed range of 0.5km/h-200km/h and has a sampling frequency of 250Hz (Kistler, 2016). It was developed for on-road vehicles, where the motion is predominately planar with little body motion. This makes the sensor well suited for testing high-speed dynamic manoeuvres on fairly smooth roads but it performs poorly in off-road scenarios that typically occur at low speeds, on varied terrain, at large side-slip angles (Botha and Els, 2015).

Botha and Els (2015) devised an alternative to current methods whereby the side-slip angle could accurately be measured over various terrains such as smooth concrete and Belgian paving by using inexpensive, off-the-shelf cameras and DIC. It was later proven that these methods could also be used on snow and ice (Botha, Els, Shoop, Becker and Sopher, 2016). This technique was developed to overcome the hurdles faced with measuring the side-slip angle in off-road conditions, to robustly measure at low speed over varied terrain and does not require large dynamic excitation. Various methods were proposed, either using a single camera or a calibrated stereo-vision rig, comprising of two cameras at a fixed distance apart. The sensor could be mounted on the vehicle to measure vehicle side-slip angle or be mounted on the hub of the wheel (to allow rotation around the king pin as the wheel is steered) to measure tyre side slip angle. For both cases, the sensors must mounted downwards towards the terrain at all times. One disadvantage of the using a camera based system is that it requires adequate lighting and measurements will be compromised in low lighting conditions such as at night. In such cases, additional light sources are required or potentially the use of other optical sensors, such as infra-red cameras. This however will not fall part of the current investigation. There is also a risk if the camera lens obscured by debris which will affect measurements. Three algorithms were devised by Botha and Els (2015) and will be discussed next. All the algorithms were implemented in post-processing using pre-recorded footage and measurements were not conducted in real-time due to the large computational expense of the algorithms.

Algorithm 1 uses is an efficient 2D algorithm that uses a single camera. Identifiable points on the image, known as features, are tracked between sequential images (feature tracking is discussed in

detail in Section 2.3.8). In brief, it means that the new pixel location of that point or feature is found on the next sequential image. The direction of the feature motion can be considered the directional travel of the camera. Therefore, if the camera is mounted in a known orientation with respect to the vehicle body, the direction of travel of the camera would be measured relative to the mounting orientation. If the the mounting direction is such that the feature motion is vertical on the image when the side-slip angle is zero, then simply determining the angle of the feature motion relative to the vertical (or horizontal) direction would result in direct measurement of the side-slip angle. A graphical representation will be shown in Section 3.2. It should be noted that the motion is in Pixels and not scaled to real-world coordinates. This algorithm is the least computationally expensive of the algorithms making it a suitable candidate for real-time applications. The main disadvantage of this algorithm is that it assumes that the vehicle undergoes predominately planar motion which is valid for traditional paved road conditions but does not necessarily hold when considering off-road scenarios where the terrain induces other motions on the vehicle such as vertical displacement, roll, pitch, and yaw. Figure 2.3 illustrates the effect that various vehicle motions impose on the image. Since these effects can have a significant impact on the images, algorithm 2 and 3 were developed to compensate for these motions.

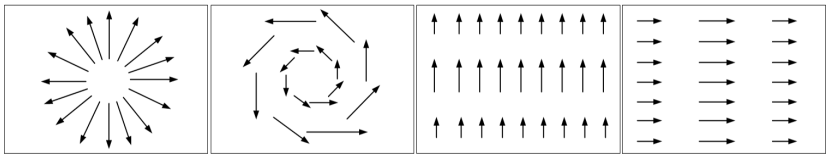


Figure 2.3. Effect of motion on the camera (a) Vertical motion (b) yaw motion (c) roll motion (d) pitch motion (Botha, 2015)

Algorithm 2 uses pose estimation, which determines the rigid body motion of the camera, using a single camera. Features are once again tracked between sequential images that are used to calculate the Essential matrix as shown by (Stewenius et al., 2005). This matrix describes the mapping of these features from one image to the other. From the essential matrix, all rotations can be determined, as well as all translations in scale. This means that they need to be multiplied by a scale factor in order to convert them to real-world coordinates. The scale is dependent on the motion and is therefore not consistent throughout the measuring process. The result is that the pure planar motion can be distinguished from the rigid body motion experienced when travelling on off-road terrain. The side-slip angle can be calculated by taking the angle between the lateral and longitudinal motion and does not require the measurement to be unscaled. The additional velocity measurements obtained from

this algorithm can be used in conjunction with other velocity measurements to evaluate the produced side-slip angle. This algorithm requires many complex iterations and error checking before converging to a suitable answer, making it computationally expensive and potentially difficult to implement in real-time.

The third algorithm uses a stereo-vision rig that obtains all measurements directly in real-world units and does not require scaling. This method uses two cameras that view the same scene at the same time which allows for 3D measurements much like human eyes determine depth. Determining 3D coordinates from a stereo-vision rig will be discussed in detail in Section 2.3.7. Once a 3D map of the scene has been generated, features on the map can be tracked between sequential images. As before, the rotation and translation matrices can be determined from the tracked points. However, unlike algorithm 2, these are unscaled and represent the motions in SI units. Similar to algorithm 2, the pure planar motion underwent by the vehicle can be distinguished from other vehicle motions and the side-slip angle calculated by taking the angle between the lateral and longitudinal movement. The 3D map generation is computationally expensive and is the bottleneck for computation time in this algorithm. Fortunately, the calculations are vastly parallel, meaning they can run in parallel to one another and the result combined. Graphics processing units or GPU's, are extremely efficient with these types of calculations and can greatly reduce computation time. The use of GPU's could make it a viable candidate for real-time measurements.

Botha and Els (2015) validated the algorithms by comparing the measurements to the Kistler Correvit S-HR sensor. The cameras and sensor were mounted on a gimbal that allowed them to be rotated in one-degree increments relative to the vehicle. This allows for the side-slip angle to be set at a fixed angle while the vehicle travels in a straight line. This forces a known artificial side-slip angle to be measured by the sensor in a controlled manner. This method was chosen as opposed to dynamic manoeuvres that would induce vehicle side-slip as it is not a repeatable and controlled process. There is also no accurate comparative measure that would validate the measurements. The gimbal system was mounted close to the vehicle's centre of gravity, or CG, to reduce the effect of other vehicle motions. These tests were conducted at various side-slip angles and over two terrains, a flat concrete surface and Belgian paving (a non-deformable cobblestone-like terrain) to simulate off-road terrains. The vehicle velocity did not exceed 20km/h during these tests. The results of the tests are summarised in Table 2.1 and show the mean values, the standard deviation (STD) and the root-mean-square (RMS) values. It is evident that all three the algorithms outperformed the Correvit S-HR sensor with lower noise levels

proving the use of cameras as a viable alternative to the sensor.

Table 2.1. Results obtained for all techniques at various side-slip angles and over two terrains

Angle	Correvit S-HR			2D Planar			2D Pose			3D		
Flat [deg]	RMS	Mean	STD	RMS	Mean	STD	RMS	Mean	STD	RMS	Mean	STD
0	2.30	0.42	2.26	0.91	-0.36	0.84	0.91	-0.31	0.85	0.83	0.06	0.83
2	2.76	2.45	1.28	1.87	1.75	0.66	1.89	1.76	0.69	1.81	1.70	0.60
5	5.86	5.48	2.09	4.98	4.90	0.89	5.02	4.92	0.98	4.93	4.85	0.87
10	10.39	10.10	2.48	9.96	9.95	0.53	9.97	9.95	0.60	9.81	9.77	0.87
Belgian [deg]	RMS	Mean	STD	RMS	Mean	STD	RMS	Mean	STD	RMS	Mean	STD
0	1.71	0.65	1.58	0.91	-0.09	0.91	0.94	-0.02	0.93	0.92	-0.09	0.92
2	2.98	2.52	1.59	2.10	1.85	0.99	2.18	1.92	1.01	2.06	1.83	0.95
5	5.60	5.42	1.42	4.92	4.85	0.83	4.97	4.90	0.86	4.89	4.79	0.96
10	10.56	10.35	2.12	10.03	10.00	0.75	10.07	10.04	0.78	9.90	9.87	0.87

Comparing the results from the flat surface to the Belgian paving it can be seen that the STD of all three algorithms increased. However, they only increased (on average) by 10% and the error never exceeded 1deg. The mean values stayed relatively constant with a low error. Even though the assumption of pure planar motion was clearly violated over the Belgian paving, the 2D Planar algorithm performed better than expected with comparable accuracies to the other algorithms. This leads to the conclusion that using a more computationally expensive algorithm to distinguish between the planar motion and the rigid body motion does not significantly improve the accuracy of the results. The computationally efficient, 2D Planar method, is therefore sufficient and the best candidate for real-time implementation and will be used during this investigation.

2.2.2 Rut depth measurements

When vehicles traverse over deformable soils, it creates a loading-unloading sequence which induces large, three-dimensional deformation of the soil (Hambleton and Drescher, 2009). Figure 2.4 shows a longitudinal cross-section of the soil under loading conditions, demonstrating how ruts are formed. The soil is typically pushed forward and to the sides of the wheel. The soil that is pushed to the side of the wheel results in the formation of two parallel berms that run along side the length of the rut. These are known are known as the rut piles. A substantial portion of the soil is also compacted resulting in vertical deformation. This permanent deformation is commonly known as the rut. This can be seen

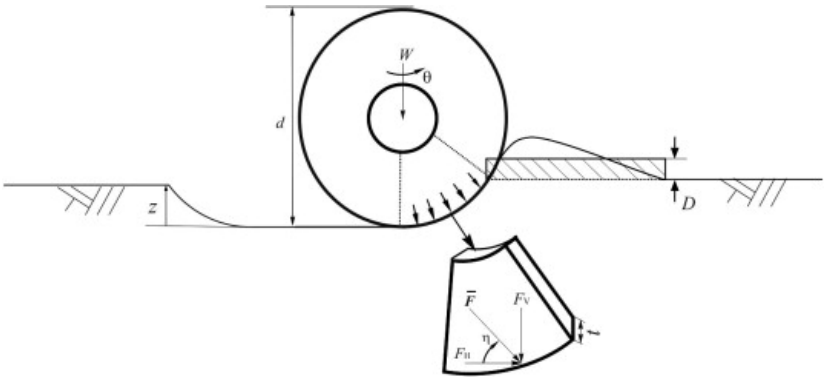


Figure 2.4. Rut formation on deformable soils (Esmaeili and Ivanovic, 2015)

in Figure 2.5 which shows a lateral cross section of a rut. The formation of ruts and the resulting compaction of the soil drastically affects the ability of the vehicle to traverse the terrain.

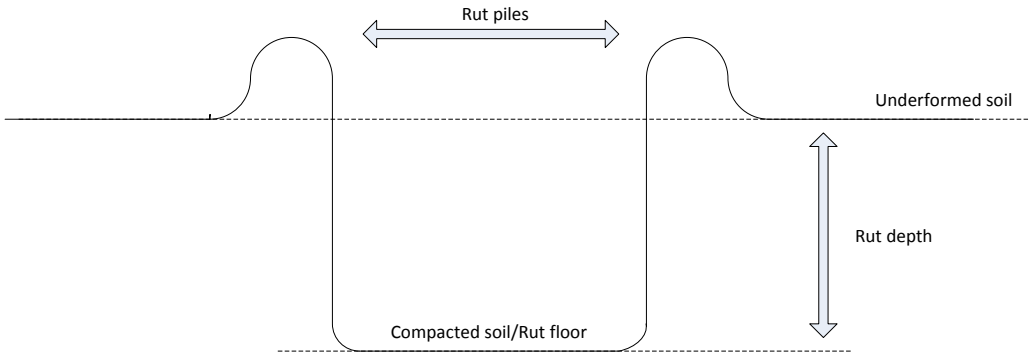


Figure 2.5. Lateral cross section of a rut

Soil compaction reduces the total soil porosity which affects the soil air, water permeability, and water infiltration (Soane and van Ouwerkerk, 2013). In agriculture, this leads to reduced crop yield. It also enhances physical, chemical and biological processes that are harmful for crop production. Soil damage occurs not only on the surface but at subsoil layers that are not visible from the surface. The exposed soil from the rut is highly susceptible to erosion. Water that has pooled in these ruts, can freeze in frigid/wintry conditions and affect vehicle traction. The pooling of water can also cause inconsistency in soil water content resulting in effected trafficability The continuously increasing size and weight of off-road vehicles and agricultural machinery threatens soil quality and ultimately leads to soil degradation. Many studies have been conducted to assess and monitor the degradation of the

soil to reduce the severity of traversing vehicles, whether it be on unpaved roads or agricultural fields (Shoop et al., 2013) (Halvorson et al., 2001).

Soil strength is a measure of how the soil reacts when applying a force to it and is an important parameter that can be used to determine whether a vehicle can traverse the terrain without becoming immobilised. Soil strength is a measure of the soil‘s mechanical resistance to penetration and is commonly used to measure soil compaction. One of the most commonly used penetrative resistance measures is the Cone Index or CI. The CI is defined as the force per unit base area that is required to press a standard shape cone into the soil. Fountas et al. (2013) used cone-shaped penetrometers with multiple tips in conjunction with GPS to characterise the soil's penetrative resistance (producing soil strength profiles) and its spatial distribution over a large area. In the past, these tests have been manually conducted resulting in an extremely labour-intensive testing procedure. This led to the development of electronic devices that increase productivity, however, the measurement remains a tedious process since measurements are still made at discrete locations. Hemmat et al. (2014) showed that there is a correlation between the rut depth and soil cone penetrative resistances that can be used as a measure of the soil's bearing capacity and trafficability. Although the correlation is dependent on the vehicle, the soil strength can be measured using only the soil sinkage or rut depth. This makes rut depth an important parameter that can be measured with relative ease and is very beneficial to estimate vehicle mobility. It also provides information for environmental studies (Anderson et al., 2005) and provides soil characteristics.

Accurate soil deformation measurement is also vital for improving soil models for simulation and future prediction (Iagnemma et al., 2003). The soil models can be used to evaluate the performance of wheels on deformable terrains which in turn aids in the development of off-road wheels. Improved off-road wheels would reduce the rutting of soils and increase overall wheel efficiency and performance. Botta et al. (2012) and Kurjenluoma et al. (2009) investigated the motion resistance experienced by off-road vehicles traversing deformable soils typically found in agricultural fields. Both investigations concluded that higher motion resistances are experienced in soils with a low bearing capacity which is the result of soil rutting. Motion resistance increases fuel consumption and therefore increases running costs.

It is apparent that many systems can benefit from real-time rut measurements. Providing this information to vehicle control systems such as traction control or tyre pressure inflation systems would improve

vehicle performance, reduce running costs and reduce the impact on the terrain. It can also be used to determine whether the vehicle can safely traverse the terrain reducing the risk of immobilization. Continual assessment of roads would also be beneficial since these systems can be used to evaluate the need for road maintenance.

Rut measurements are generally conducted manually resulting in a time consuming, labour intensive process. In Shoop et al. (2013) rut depth was measured using a rod that was placed over the rut, perpendicular to the direction in which the wheel is travelling. Using another ruler, the relative distance between the rod and the lowest point within the rut is measured and taken as the rut depth. This is the preferred method for rut measurements due to its simplicity and low cost. An example of this method is shown in Figure 2.6. In Kurjenluoma et al. (2009) the rut depth was measured by using two peiseler wheels that ran parallel to each other with one in the formed rut and the other one next to it. An ultrasonic sensor was used to measure the relative height difference between the wheels which gives the rut depth. Real-time measurements can also be done using LiDAR and laser line scanners, however, these are expensive and are mainly developed for asphalt usage.

Figure 2.6. The prevailing method used for rut depth measurements (Shoop et al., 2013)

Botha, Els, Shoop, Becker and Sopherb (2016) developed a method to perform 3D rut profile measurements in snow and mud. In their investigation, they tested the feasibility of using DIC to measure un-deformed and deformed terrain profiles. Inexpensive off-the-shelf cameras were used to create

a stereo-vision rig that enables the system to determine a 3D view of the scene, similar to the way human eyes determine depth. Determining 3D coordinates from a stereo-vision rig will be discussed in detail in Section 2.3.7. A typical result of the measured terrain profile is shown in Figure 2.7, where a vehicle had traversed soft mud, deforming the soil and creating the observed terrain profile. The shape of the rut is clearly visible. The depth was determined by first reducing the surface profile to a single lateral cross-section of the surface profile. As the vehicle travels over the terrain, the distance between the camera and the terrain continuously changes and therefore a ground plane (the reference height to the sides of the rut) needs to be determined. Points to the left and the right of the rut are taken (taking the rut piles into consideration) and a straight line is fitted through these points which represent the ground plane datum. The rut depth is determined by taking the maximum of the reduced surface profile relative to the ground plane datum. Measurements were validated using the ruler technique (as demonstrated in Figure 2.6). The exact locations of these measurements could not be obtained, making precise correlations not possible. However, the largest rut depths were measured and proved to correlate. It should be noted that measurements were conducted in post-processing from pre-recorded footage.

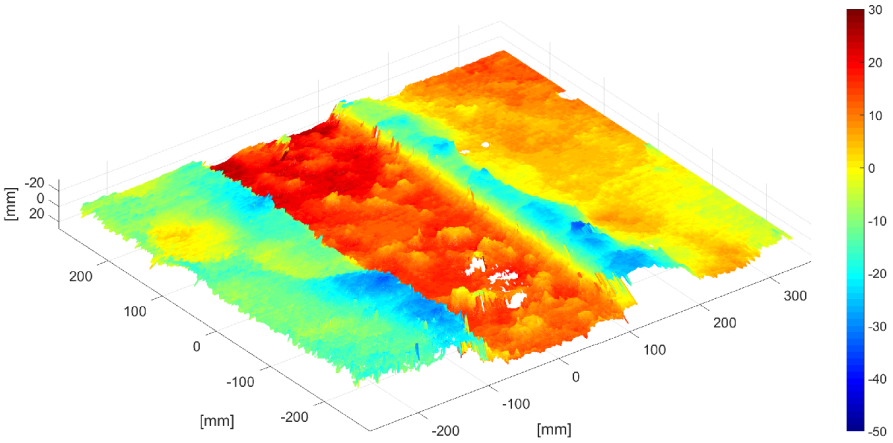


Figure 2.7. Terrain profile in mud (Botha, Els, Shoop, Becker and Sopherb, 2016)

One advantage of this system is that it is non-contact based and therefore does not influence the terrain while measurements are taken. Another advantage is that it provides length and width to the measurement when compared to conventional point or line scan techniques, providing more information that can be used to evaluate the road profile. As previously mentioned, camera based systems require

adequate lighting conditions and additional light sources are required for use in poor lighting conditions such as at night. The system is also at risk in the event the lens is obscured by debris.

Botha, Els, Shoop, Becker and Sopherb (2016) showed that this is a viable alternative to current methods during mobility tests. Implementation of this system in real-time can add benefits to vehicle mobility. For this reason, DIC was chosen as the best candidate for implementation of real-time rut depth measurements. During this investigation, only the rut depth is measured although there are many other applications for the use of DIC in terramechanics.

Chapter 2.3 aims to summarise the fundamental basics that are required to develop the rut depth sensor. This includes fundamentals of acquiring digital images, mathematical models to remove distortion from the images and recovering depth from a stereo-vision rig.

2.3 CAMERA-BASED VISION

DIC is the process of comparing and analysing sequential images by applying software algorithms to these images. In the previous sections, it was shown that DIC is a useful tool that can be used for various vehicle measurements. The following section aims to summarise the fundamental principles required for applying DIC. During this investigation, an open source library was used for the majority of the DIC algorithms. For both measurements, it is required to first obtain digital images. The images then need to be processed to remove any distortion created by the lens, which is achieved through calibration. For rut measurements, a stereo-vision setup, consisting of two cameras, is used to determine depth. These cameras also require calibration to find the relative geometry between the two cameras. DIC algorithms are applied to the whole images to obtain the 3D coordinates of the scene. Side-slip angle has different requirements. As discussed, discrete points are tracked across sequential images. Detailed implementation of the algorithms are discussed in the relevant chapters.

2.3.1 OpenCV

Open Computer Vision (OpenCV) is a free, open source computer vision library that includes thousands of computer vision algorithms (OpenCV, 2017). The goal of the project was to advance computer vision

research by providing optimized, open source code for basic computer vision. Many of the algorithms that are used during this investigation are the library implementation's of the algorithm.

2.3.2 Digital imagery sensors

Digital cameras use a matrix of electronic photodiodes that convert the light, which is focused through the lens, into an electronic signal. A photodiode is a semiconductor diode which, when exposed to light, generates a potential difference or changes its electrical resistance proportionally to the intensity of the exposed light. An analogue-to-digital converter is used to transform the electronic signals into a 3-byte digital value that represents the intensity of red, green and blue light of the colour spectrum. In a grey scale image, the light intensity is represented by a single byte of data.

Two popular digital image sensors are the Charged Coupling Device (CCD) and the Complimentary Charged Oxide Semiconductor (CMOS). Both these devices use photodiodes. Historically, the major difference between the CMOS and CCD was the shutter mechanism, where CMOS used a rolling shutter and CCD used a global shutter. With rolling shutters, the sensor is sampled from one side to the other, either vertically or horizontally. Global shutters sample the image in a global fashion where all the pixels are processed in one time instant. Global shutters are preferred for video or motion photography where the scene is not stationary and pixels are required to represent the exact same time of exposure. Rolling shutters cause a distortion effect on a moving image. Figure 2.8 demonstrates this effect, where the instantaneous shutter gave an undistorted image of the fan blades compared to the severely distorted image produced by the rolling shutter. (Andor, 2014) Both sensors have their

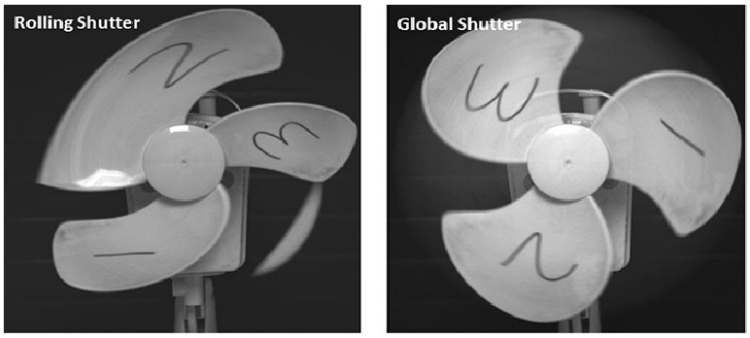


Figure 2.8. Images of moving fan blades using a Rolling Shutter vs Global Shutter (Andor, 2014)

advantages and disadvantages. CMOS sensors have higher processing rates, are cheaper and consume

less power. However, CCD sensors produce far less noise. Since most tests during this investigation are non-stationary, a global shutter is preferred. Recent developments have allowed CMOS sensors to use global shutters as opposed to rolling shutters (Sony, 2017). For this reason, a CMOS based camera with global shuttering was used for this investigation.

2.3.3 Camera model

Conceptually, there is no difference between a digital image and a photograph. Light from an observed 3D scene is projected through a lens to concentrate the rays coming from different scene elements and is captured on a 2D image plane. Figure 2.9 demonstrates the model, where light from a scene is projected onto the image plane. In Figure 2.9, f represents the lens focal length, d_i the distance from

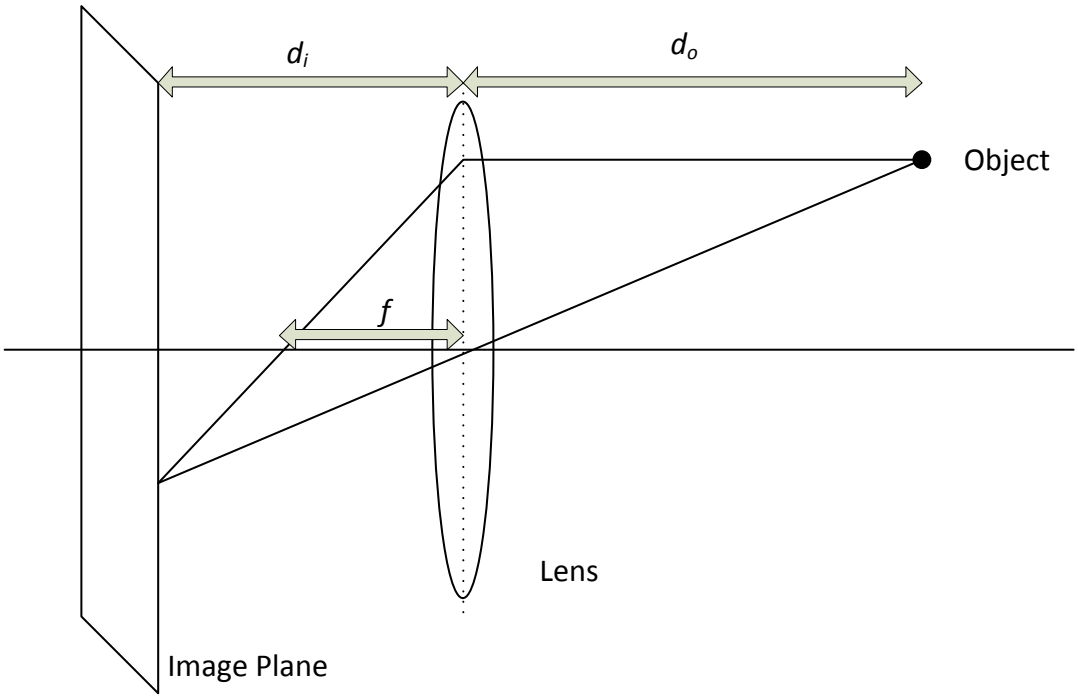


Figure 2.9. Photograph model (Laganiére, 2011)

the image plane to the lens and d_o the distance between the object and the lens. These dimensions are related through the thin lens equation (Laganiére, 2011):

$$\frac{1}{f} = \frac{1}{d_0} + \frac{1}{d_i} \tag{2.3}$$

In computer vision, the above model is simplified to produce the pin-hole camera model. The model can be simplified in three ways: (Laganiére, 2011)

1. Assume that d_o is much larger than d_i and therefore the image plane is located at the focal length. This is valid as typically the focal length ranges between 4-6mm where the distance to the object is minimum two orders of magnitude larger.
2. Neglect the effect of the lens by considering a camera with an infinitesimal aperture and therefore only the central ray is considered. This is a valid assumption since theoretically, this does not affect the image.
3. Place the image plane in front of the lens to obtain an identical but upright image which is known as the pin-hole model (Figure 2.10). This is physically not possible but is mathematically equivalent.

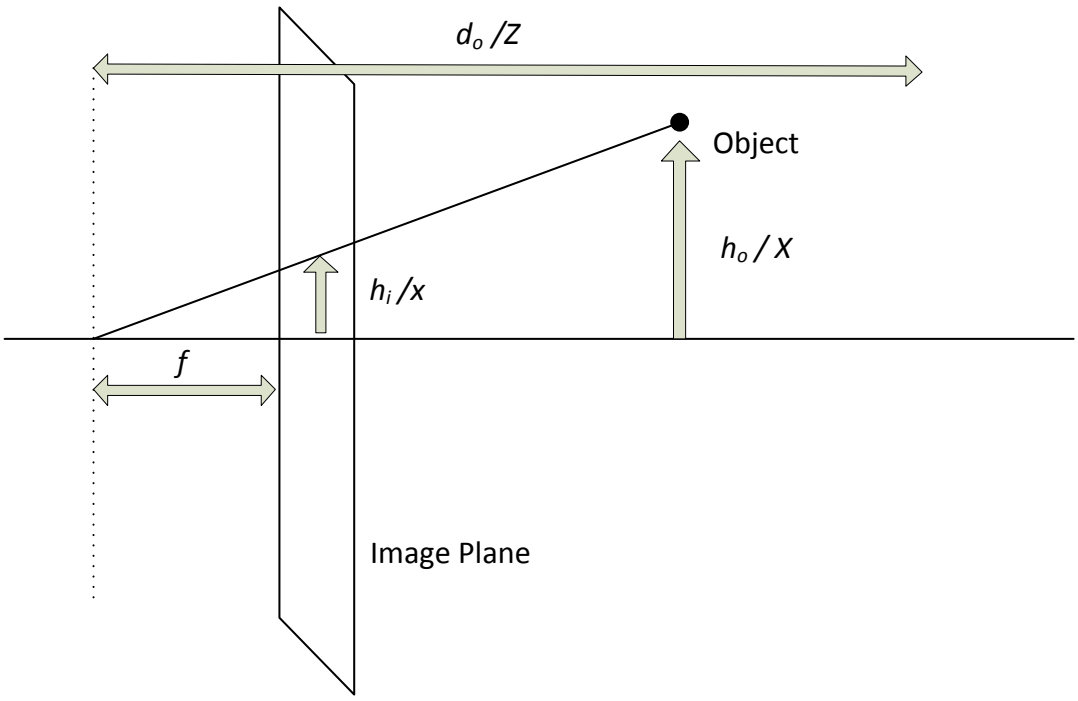


Figure 2.10. Pin-hole model (Laganiére, 2011)

Mathematically, the basic projective equation can be determined from the graphical pin-hole model using triangles:

$$h_i = f \times \frac{h_o}{d_o} \tag{2.4}$$

This can be applied in both x- and y-direction of the image plane to obtain:

$$x = f \times \frac{X}{Z} \tag{2.5}$$

$$y = f \times \frac{Y}{Z} \tag{2.6}$$

This model allows the position of a point on a 3D scene to be projected on a 2D plane. These coordinates are in *mm*, however, it is not ideal to relate the projection onto the image plane, but rather onto the image sensor. Typically the focal length is represented as a combination of the sensor and the lens focal lengths with the units in *Pixels*, in comparison to the focal length of the lens which is in *mm*. The sensor‘s focal length is redefined by multiplying the focal length of the lens in *mm* with the size of the pixels (S_x and S_y) in $pixel/mm$ to covert the units to *pixels*. The sensor focal length is not necessarily the same in the x-and y-direction because of the rectangular shape of the pixels in the sensor and is therefore split into x and y components. Noting that the sensor‘s focal length is given as the product of the lens focal length and the sensor size, the sensor focal length is given as:

$$f_x = S_x \times f \tag{2.7}$$

$$f_y = S_y \times f \tag{2.8}$$

The pixel coordinates of a point in a 3D scene that is projected on the image sensor are then obtained as (in *pixels*):

$$x = f_x \times \frac{X}{Z} + u_o \tag{2.9}$$

$$y = f_y \times \frac{Y}{Z} + v_o \tag{2.10}$$

Where u_o and v_o represent the principal point which is the distance from the centre origin to the intersection of the optical axis. Equations 2.9 and 2.10 are depicted in Figure 2.11 (Laganiére, 2011).

2.3.4 Camera and lens Settings

Various camera and lens settings can be used to alter the digital images obtained from the camera. On the camera, there are two main settings, i.e. the sensor sensitivity and shutter exposure time. The sensor sensitivity, commonly known as ISO, is a measure of amplification used on the image sensor. High ISO values make the sensor more sensitive to light. This is ideal in low lighting conditions, or when short exposure time is required to capture fast motion, or when a small aperture is required for

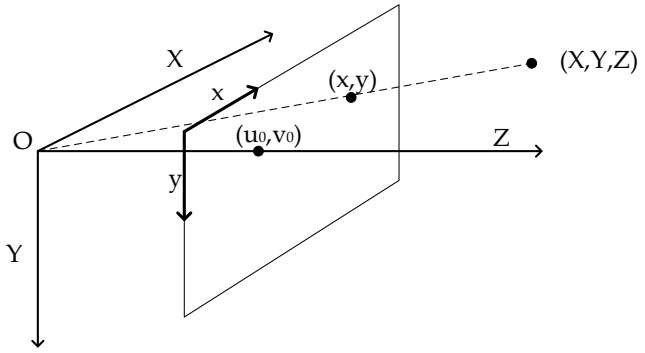


Figure 2.11. Image reference frame

an increase in depth-of-field. However, high ISO values introduce unwanted noise on the image. The duration of time that the photodiode on the image sensor is exposed to light is known as the shutter exposure time. An increase in exposure time allows a larger charge to build up in the photodiode, that represents the intensity of that pixel. This is useful when there is insufficient lighting. However, if the exposure time is too long, the image is overexposed which removes detail on the image. Large exposure times are unwanted when capturing images from moving scenes as they cause motion blur on the image.

There are two main lens parameters, i.e. aperture and focal length. The aperture is the opening or hole that the light of the scene travels through to the camera sensor. The size of the opening is defined by the f-number, which is the ratio of focal length to the aperture opening. A large f-number refers to a small opening and vice-versa. A small aperture (large f-number) increases the depth of field which improves the ability to adequately focus on an object. The focal length of a lens is the distance at which the light rays from the lens converge to form a clear image. The focal length of the image changes the magnification of the image. A large focal length increases the magnification of the image and reduces the region of interest.

In general, shutter time, sensor sensitivity and aperture should be as low as possible to reduce noise and motion blur and to increase the depth of field. This poses a problem as a reduction on these parameters results in a reduction of brightness on the image. A compromise should be made between these parameters based on the application. This can be improved by providing the scene with an abundance of light which will result in less of a compromise.

2.3.5 Single view calibration

It is evident from the pinhole camera model that parameters such as focal length, size of the image plane and principal points are required to map from 3D points in space to 2D points on the image plane, and vice-versa. These parameters are known as the intrinsic parameters and are related to the camera and lens. Calibration is the process of determining these parameters. Open source calibration algorithms were used to generate these parameters and only fundamental equations will be discussed in this section.

To start this discussion, a reference frame or coordinate system is added that intersects the image plane through the centre of projection or principal point. This is illustrated in Figure 2.11.

Homogenous coordinates are used to represent the projective geometry. They allow for a 3D point at infinity to be represented by finite coordinates. In this coordinate system, an extra dimension is added to represent 2D points as 3D vectors and 3D points as 4D vectors. In the 2D case, it can be thought of as representing a line from the origin through the point on the image plane. Applying this to Equations 2.9 and 2.10, they can be rewritten into matrix form:

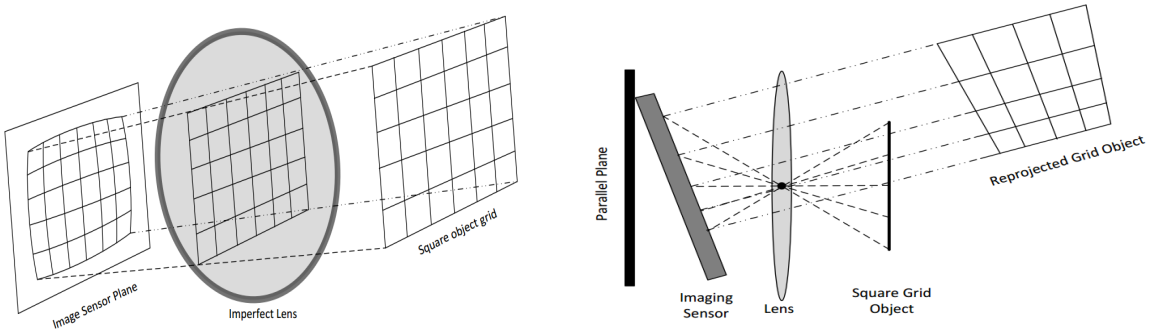
$$S \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_o \\ 0 & f_y & v_o \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.11}$$

$$\left[\text{Sensor size}\right] = \left[\text{Intrinsic matrix}\right] \times \left[\text{Projection matrix}\right] \times \left[\text{Coordinate vector}\right]$$

If the reference frame is not at the projection centre of the lens, it must be accounted for by adding a rotation matrix and a translation vector to the projection matrix of Equation 2.11. These are known as the extrinsic parameters and represent a rigid transform to ensure the reference frame is at the projection centre. This yields:

$$S \times \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_o \\ 0 & f_y & v_o \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 & r_2 & r_3 & t_1 \\ r_4 & r_5 & r_6 & t_2 \\ r_7 & r_8 & r_9 & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.12}$$

Another factor that has to be taken into account in the calibration process is lens distortion. All lenses



(a) Illustration of radial distortion **(b)** Illustration of radial distortion

Figure 2.12. Effects of lens distortion on the image plane (Botha, 2015)

distort the image in one way or another. This effect is more significant with lower quality lenses, lenses with a short focal length or when a lens is misaligned. Also, different colours distort the image unequally as their frequencies vary. Two types of distortion commonly occur, namely radial and tangential. Figure 2.12 illustrates lens distortion, where an imperfect lens results in radial distortion on the image plane sensor (a) and a misaligned lens causes tangential distortion (b).

Radial distortion is a result of rays refracting more when they are further away from the centre of the lens when compared to the rays closer to the centre. Tangential distortion occurs when the image plane is not parallel to the lens, which causes a slant on the image plane. The corrected coordinates (x_c and y_c) are calculated by adding the distortion corrections (x_r, x_t, y_r and y_t) to the coordinate, as shown in Equations 2.13 and 2.14. The distortion corrections are substituted with a suitable model, δ, to give the final corrected coordinates (Equation 2.15 and 2.16) that are a function of the radial (k_i) and tangential (p_i) distortion coefficients. Radial and tangential distortion coefficients are approximated separately and the result combined. This requires 4 parameters to be solved before the image can be corrected for distortion (Slama et al., 1980).

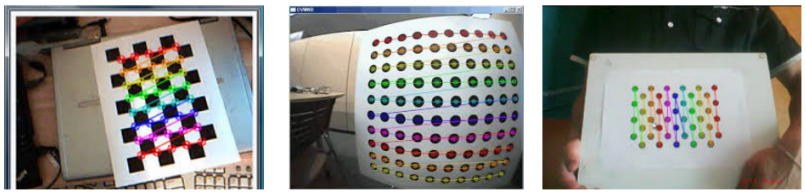
$$x_c = x + \delta x_r + \delta x_t \tag{2.13}$$

$$y_c = y + \delta y_r + \delta y_t \tag{2.14}$$

$$x_c = x + x(k_1 r^2 + k_2 r^4 + \ldots) + [2p_1 y + p_2(r^2 + 2x^2) \tag{2.15}$$

$$y_c = y + y(k_1 r^2 + k_2 r^4 + \ldots) + [2p_2 x + p_1(r^2 + 2y^2)] \tag{2.16}$$

The most popular method for calibration were developed by Zhang (2000). By supplying these



(a) Chessboard pattern (b) Symmetric circle grid (c) Asymmetric circle grid

Figure 2.13. Various patterns used for calibration (Botha, 2015)

algorithms with images of a known pattern and size, at various angles and distances from the camera, the parameters can be solved in the least squares manner. The calibration algorithms used for calculating these parameters (intrinsic matrix and distortion coefficients) are not in the scope of this study and those supplied by the OpenCV library is use without modification. Many possible calibration patterns exist such as dots, squares, or chequered patterns as shown in Figure 2.13.

2.3.6 Multiple view calibration

The previous section described the process of calibrating a single camera by determining the intrinsic parameters and distortion coefficients. This section describes the calibration process used over multiple views which requires determining the relative geometry between the cameras. Multiple views of a scene can be obtained in two ways:

1. Using two cameras that are imaging the same scene, at the same time. This is often referred to as a stereographic or stereo-vision rig.
2. Using a single camera but imaging the same scene from two different locations.

However, this investigation will focus on viewing the scene at the same time using two cameras. Figure 2.14 shows the geometric relationship between two stereo images, also known as epipolar geometry. In this illustration, point P refers to the scene in consideration, points p and p' are the projections of P in the planes Π and Π' respectively. The lines joining ep and $e'p'$ are the epipolar lines, where e is the epipolar point and represents the projection of plane Π' centre onto plane Π and e' represents the projection of plane Π centre onto the Π'. This is of great significance as the ray OP in the plane Π is projected onto plane Π' as the epipolar line. The implication of this is that a feature in one image can be found on the other image by searching along its corresponding epipolar line. This is known

as the epipolar constraint (Hartley and Zisserman, 2003). The importance of this relationship will be discussed in Section 2.3.7. Calibration of a stereo-vision rig requires the determination of the relative

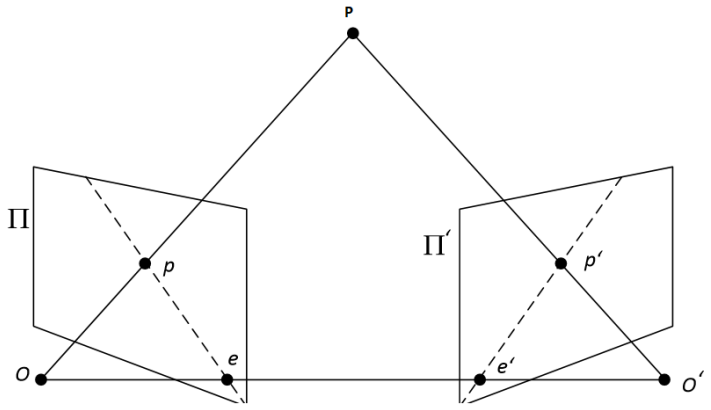


Figure 2.14. Epipolar Geometry between two views (Botha, 2015)

rotation and translation between the two cameras. The calibration process is similar to that of the single camera calibration process used to determine the camera intrinsic and distortion parameters. However, in this case, both cameras view the calibration pattern simultaneously. The calibration, therefore, produces the intrinsic parameters and distortion coefficients of each camera as well as the geometric relationship between the two cameras. Once again the algorithms that are used to calibrate the rig do not fall within the scope of this study and will not be discussed. As with calibration of a single camera, calibration is achieved using built-in algorithms supplied by OpenCV. Once the stereo-vision rig has been calibrated, it is possible to extract 3D coordinates of a scene by back-projecting the rays. This is discussed in the following section.

2.3.7 Stereo vision

The process used to determine depth from a stereographic rig is similar to one of the methods that the human eyes use to determine depth, commonly known as binocular vision. Binocular disparity is defined as the change in location that an object appears to undergo between the left and right views of a stereographic rig. This is best illustrated in Figure 2.15. In this image, it is evident that the features (moon, sun and face) appear at different locations on the left and right images. This change in location is a result of having a translational and rotational separation between the left and the right images. In

order to retrieve 3D coordinates of a scene, the disparity between the images has to be determined.

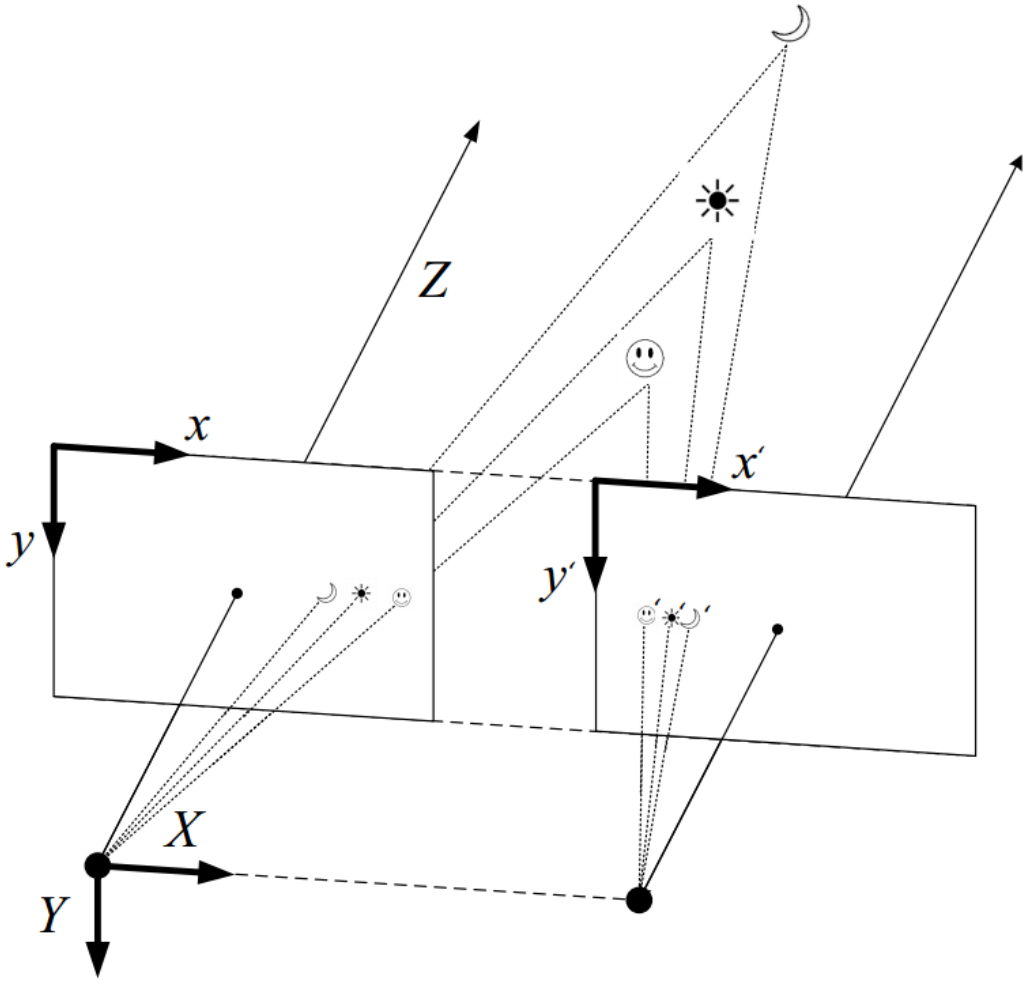


Figure 2.15. Illustration of disparity on a stereo-vision rig

Stereo-vision is applied to the whole image and obtains the depth of the whole scene7. This means that each pixel is matched between the left and right images. The disparity d is computed by tracking regions of pixels between the left and right images using an exhaustive search and suitable correlation metric and a winner-takes-all matching strategy. This means that the regions that have the best match, are most likely to be the corresponding region. The simplest and most common correlation metric used is the minimization of the sum of absolute differences or SAD given as (Konolige, 1997)

$$SAD = \sum_{i=-N/2}^{N/2} \sum_{j=-N/2}^{N/2} |I(x+i,y+j) - I'(x+i+dx,y+j+dy)| \tag{2.17}$$

where I and I' represent the brightness of the left and right images. Other metrics such as the sum of squared differences, or SSD, and normalised cross-correlation, or NCC, also exist. For a pixel in the left image, the absolute difference of a $N \times N$ window around the pixel is taken between the left and the right images. The result is then summed. The window on the right image is moved and the process repeated until the SAD of the whole second image has been determined. The region with the lowest SAD is said to have the best correlation. The next pixel on the first image is then selected and the process once again repeated, leading to an extensive computationally expensive search.

The exhaustive search can be simplified by remembering that epipolar geometry (shown in Figure 2.16) allows for a feature in one image to be mapped onto the epipolar line of the other image, which reduces the complex search down to a 1D line search. This is achieved by reprojecting the images such that the epipolar lines of both images are horizontal and correspond to the same row, so that a feature in a row on the left image, appears in the same row on the corresponding right image. This is known as image rectification. The rectification can be obtained from known geometry between cameras by the calibration process. In Figure 2.16 point P is project onto the left and right images at point p and p'. It can be seen that after the images have been rectified, both points lie on the same horizontal line. OpenCV supports two algorithms that calculate the disparity between two images,

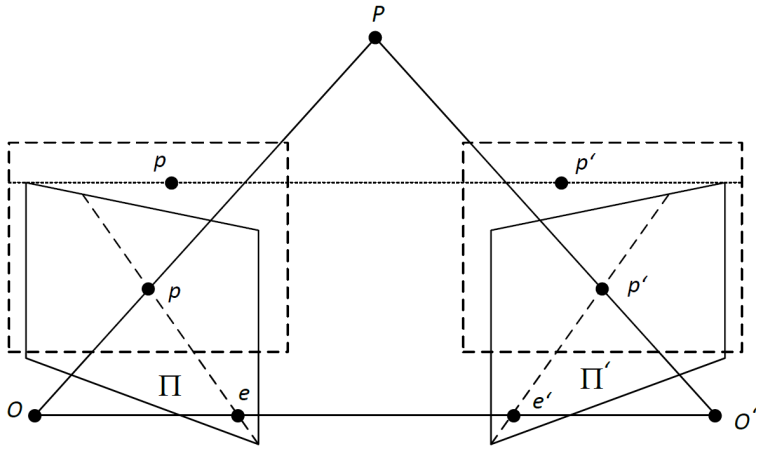


Figure 2.16. Image Rectification (Botha, 2015)

Block Matching (BM) and Semi-Global Block Matching (SGBM), although many others exist. Both these algorithms feature in the top ten of the KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago) stereo benchmark, where SGBM is ranked 3rd and BM ranked 9th (Geiger et al., 2012). According to the benchmark, BM was computationally cheaper but far less accurate than SGBM. This is because SGBM first implements BM and optimises the result. In this

study, processing time dominates the importance of accuracy and therefore BM will be implemented due to its faster processing time. Computing the disparity map was one of the major bottlenecks in computation time. Seeing as the majority of computations consist of multiple 1D line searches, these calculations can be done parallel to one another. Graphics processing units, or GPU's, are efficient with simple parallel calculations such as these and have shown significant reductions in processing time for disparity mapping. Hernandez-Juarez et al. (2016) showed that for a 640x480 image, frame rates of up to 230Hz can be achieved using high-end GPU's. The use of GPU's do not fall within the scope of this investigation but, will significantly reduce processing time making it a necessity for future work.

The algorithm determines the disparity of a pixel by starting at the same location on the other image and then searches along the row using the SAD metric to find its corresponding pixel. Generally, the disparity is computed relative to the left image, but the algorithms allow for the disparity to be computed relative to the right image. This means that the algorithm starts on the right image at the same pixel location as that of the left image and moves the window accordingly. Figure 2.17 graphically illustrates how the disparity is calculated between the left and right image ($d = x - x'$). Once the disparity between the images has been obtained, it is a trivial process to recover the depth. Using the process of similar triangles, indicated by the coloured triangles in Figure 2.17, the depth (Z) can be derived using Equation 2.18, where the left side of the equation is represented by the blue triangle and the right side represented by a red triangle. This provides the relationship in Equation 2.20.

$$\frac{t-(x-x')}{Z-f} = \frac{t}{Z} \tag{2.18}$$

$$\frac{t-d}{Z-f} = \frac{t}{Z} \tag{2.19}$$

$$Z = \frac{tf}{d} \tag{2.20}$$

The depth resolution is a crucial factor to be considered. From Equation 2.20 it can be shown that the depth resolution (ΔZ) can be derived from the focal length, translation displacement between the cameras and the minimum measurable disparity Δd (Grewe and Kak, 1994).

$$\Delta Z = \frac{-\Delta d Z^2}{tf + \Delta d Z} \tag{2.21}$$

From Equation 2.21 it is evident that it is desired to have the object as close to the camera as possible in order to improve the depth resolution.

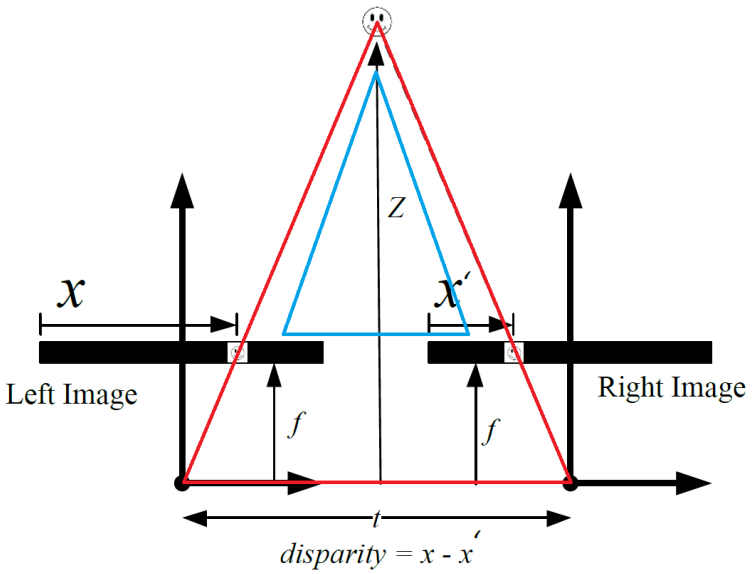


Figure 2.17. Disparity mapping between left and right images (Botha, 2015)

2.3.8 Feature tracking

Feature tracking across successive images is required for measuring the side-slip angle. Features are regions of the image that are unique and make it easy to identify and subsequently, track. In the field of computer vision, the process of tracking identifiable points is known as optical flow and is heavily used in video-based applications. Optical flow algorithms can be classified into two main groups, namely sparse and dense optical flow algorithms. Sparse optical flow techniques track the motion of a few discrete features rather than tracking the motion of the whole image as in the case of dense optical flow. In many applications the motion of the whole scene is necessary such as stereo-vision where DIC is applied to the full image. However, for side-slip angle measurements, it is only required to track relatively few discrete points. This approach is also more computationally efficient making it better suited for real-time applications.

Feature tracking faces many challenges. To start, one first has to determine which features can be tracked. This is challenging as some features may change appearance due to the motion of the camera and lighting conditions. Additionally, points may disappear and appear resulting in further ambiguities. Often, the tracking algorithms are prone to drift due to the accumulation of small errors as the appearance model is updated. Numerous sparse optical flow algorithms have been developed to

select and track features to combat these challenges. The OpenCV library has a variety of algorithms. The most popular and widely used is the Lucas-Kanade (LK) (Lucas and Kanade, 1981) algorithm, which will be discussed in detail in Section 2.3.8.1. A short comparison in performance of sparse optical flow algorithms available from the OpenCV library will be discussed in Section 3.1 that proves the LK algorithm to be the fastest algorithm on a single core CPU.

Many other feature tracking algorithms exist such as the Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Oriented FAST and Rotated BRIEF (ORB) and Binary Robust Invariant Scalable Keypoints (BRISK). A short description of each algorithm is given below. More algorithms do exist but will not be considered during this investigation.

SIFT (Lowe, 2004) address the problem of matching features that have changed in scale and rotation. In this algorithm, features are identified by finding local extremes between scales of the difference of Gaussian (DoG). Orientation is determined by selecting the dominant orientation from histogram of oriented gradients (HOG). The feature descriptor then defines the features by its scale, orientation and it's location. By calculating feature descriptor on both images, the corresponding features can be found by calculating the euclidean distances between all features descriptor. Features with the smallest distances are said to be a match. This is known as a brute-force matching strategy as each feature descriptor is compared to all other possibilities.

SURF (Bay et al., 2008) is based on the same principals as SIFT, however each operation has been optimised. The scale and DoG are approximated using box shaped filters. The features are identified by taking the second-order derivative of the DoG at each scale and selected if the determinant is maximal. The orientation of each feature is approximated through wavelet responses and the dominant orientation selected by summing the responses along a sliding window and finding the largest vector. As with SIFT, the feature descriptor then defines the feature by its scale, orientation and it's location. The feature descriptors are then matched using the brute force matching strategy, as with SIFT.

ORB (Rublee et al., 2011) uses a feature detector based on machine learning that has been trained as a classifier to detect edges. It has been modified to include the orientation of the feature using the intensity centroid. The algorithm uses a modified binary feature descriptor that is a bit string comparative description of the feature that is steered in the direction of the edge orientation. The descriptors are then matched by comparing the Hamming distance between all descriptors.

BRISK (Leutenegger et al., 2011) uses a scale-space feature detection method that detects features using a saliency criterion across the image and the scale dimensions. Local intensity gradients are obtained in concentric circles in the neighbourhood of each feature and is used to obtained the orientation of the feature. The feature descriptor consists of a comparative bit string that is formed using the descriptor. The descriptors are then matched by comparing the Hamming distance between all descriptors. The descriptors are then matched by comparing the Hamming distance between all descriptors.

2.3.8.1 Lucas Kanade

The OpenCV implementation of the LK algorithm, is actually a variant called the Kanade-Lucas-Tomasi, or KLT, algorithm. This algorithm combines the LK tracking algorithm with features from Shi and Tomasi (1994) or the so called "Good Features to Track". Shi and Tomasi (1994) proposed a feature selection criterion that is optimal by construction as it based on how the LK tracking algorithm functions. These features do not necessarily represent a physical point in space but mathematically represent the best possible point that the LK can track from the image. This greatly improves the accuracy and reliability of the algorithm.

The LK optical flow algorithm tracks these unique features across sequential images by solving the optical flow equation in a least squares approach. As mentioned before, the features for KLT are selected based on how the LK functions. Therefore it is required to discuss the LK algorithm before determining what features should be tracked. The LK algorithm makes three important assumptions. Firstly, the pixel brightness across images are consistent. This implies that the brightness of the pixel being tracked stays constant from the image to image. This would be invalid if e.g. shadows appear on the image but can be overcome by using adjustable camera parameters such as sensor gain, aperture, and contrast. The second assumption is that the pixel motion is small. The assumption of small motion is overcome by introducing a modification to the algorithm that tracks over a pyramid of smaller scaled images which reduces the motion on the image. The motion on the smallest image is then tracked and the solution used on the next largest image. This process is repeated for the larger scaled images until the solution of the motion of the original image is found (Bouguet, 2000). This is valid when lateral motion is observed from a stationary scene although may not hold if multiple objects in the scene move or if the motion is predominantly rotational. To start, consider Figure 2.18 where a point with pixel intensity I is located at (x,y) at time t. At time $t+1$, the point has moved by (u,v). The

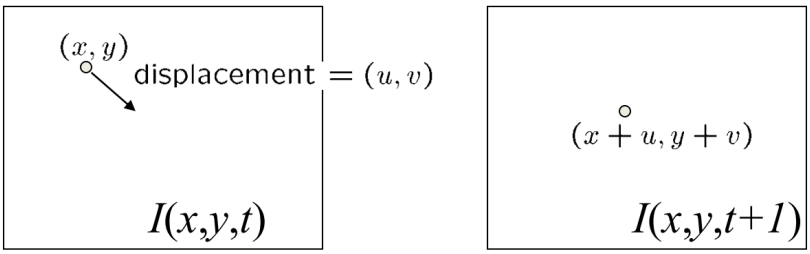


Figure 2.18. Feature motion

brightness consistency assumption therefore states

$$I(x,y,t) = I(x+u,y+v,t+1) \tag{2.22}$$

The right-hand side of Equation 2.22 can be linearise by taking a first order Taylor expansion to produce Equation 2.23. In this equation, I_x and I_y represent the partial image derivative along the x and y directions. I_t is the partial time derivative of the brightness and is determined by determining the change in brightness of the same location over a finite time difference apart. By rearranging and substituting Equation 2.22 into Equation 2.23, produces Equation 2.26. By dividing this equation by dt, Equation 2.27 is obtained.

$$I(x+u,y+v,t+1) \approx I(x,y,t) + I_x u + I_y v + I_t \tag{2.23}$$

$$I(x+u,y+v,t+1) - I(x,y,t) \approx I_x u + I_y v + I_t \tag{2.24}$$

$$0 \approx I_x u + I_y v + I_t \tag{2.25}$$

$$[I_x, I_y] \cdot [u,v]^T + I_t = 0 \tag{2.26}$$

$$\therefore [I_x, I_y] \cdot [V_x, V_y]^T + I_t = 0 \tag{2.27}$$

Equation 2.27 can be used to recover the velocity of the pixel in consideration. However this is a scalar equation with two variables. To create more equations for the pixel, the spatial coherence constraint (neighbouring pixels around the feature undergoes the same motion as the feature) is employed. If for example, a 5x5 window around the feature is considered, 25 equations are generated.

$$\begin{bmatrix} I_x(p_1) & I_y(p_1) \\ I_x(p_2) & I_y(p_2) \\ \vdots & \vdots \\ I_x(p_{25}) & I_y(p_{25}) \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix} = - \begin{bmatrix} I_t(p_1) \\ I_t(p_2) \\ \vdots \\ I_t(p_{25}) \end{bmatrix} \tag{2.28}$$

This produces an over-constrained system (for illustration, the system is donated by $Ad = b$) that can be solved using a least squares approach. This is done by multiplying the equation by A^T such that $(A^T A)d = A^T b$. The result is known as the Lucas-Kanade equation (Equation 2.29). The algorithm solves this equation to determine the velocity in the x-direction, V_x, and the velocity in the y-direction, V_y. The exact location of the feature is determined by optimising cost function in Equation 2.30 by stepping in the direction of the velocity vector, at each iteration.

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = - \begin{bmatrix} \sum I_x I_x & \sum I_y I_x \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}^{-1} \begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix} \tag{2.29}$$

$$\varepsilon = \sum_i^m \sum_j^m [I(x,y,t) - I(x+u, y+v, t+1)] \tag{2.30}$$

Shi and Tomasi (1994) stated that in order to accurately and reliably solve for the feature motion, some requirements of the LK equation need to be evaluated. Firstly, $A^t A$ should be invertible, secondly, $A^t A$ should also not be too small (which would result in small eigenvalues) and lastly $A^t A$ should be well-conditioned meaning that the ratio between the eigenvalues should not be too large. By choosing specific features to satisfy these requirements, the ability to track the features are greatly increased. The so-called "Good Features to Track" features are identified by firstly obtaining the Hessian matrix (the second derivative) of the image intensity.

$$H(x,y) = \begin{bmatrix} \frac{\delta^2 I}{\delta x^2} & \frac{\delta^2 I}{\delta x \delta y} \\ \frac{\delta^2 I}{\delta y \delta x} & \frac{\delta^2 I}{\delta y^2} \end{bmatrix} \tag{2.31}$$

The eigenvalues of the Hessian matrix are then calculated. Points of the image are then selected based on eigenvalues. If the smallest eigenvalue of the point is above a certain threshold as well as the relative ratio of the eigenvalues are below another threshold, the point is said to be unique and classed as a good to feature to be tracked. When the LK algorithm then tracks these features, it ensures that the LK equation (Equation 2.29) is fully satisfied. This results not only in pixels with good contrast but gives preference to corner points instead of edge points. Considering Figure 2.19, location A has little to no gradients which result in small eigenvalues. Location B has large gradients in some directions and small gradients in others, resulting in a high relative ratio of the eigenvalues. Location C has large gradients in all directions which result in larger eigenvalues making it the best candidate to track. It is therefore required to have texture in the observed scene that creates contrast in the image to sufficiently identify unique features. Most road surfaces offer a large variation in texture, making it easy to identify

Figure 2.19. Effect of texture variation

these unique features. Figure 2.20 depicts a typical road surface where it is clear that the road offers many unique variations in texture.

Figure 2.20. Typical asphalt surface

Botha, Els, Shoop, Becker and Sopher (2016) also studied the applicability of these algorithms on off-road surfaces such as ice and snow, which although might have large texture variation, does not

necessarily show up as illumination variation on the image. Accurate side-slip angle measurements could still be obtained on these terrains.

2.4 CONCLUSION

The focus of this chapter is split into two components. The first part of the chapter was based on the measurements being made, i.e. side-slip angle and rut depth. In both cases, the measurement was described and their applicability in vehicle dynamics discussed, as well as current measurement techniques.

For side-slip angle, the investigation showed that there is a need for an alternative measuring method on off-road terrains and at low speeds as current commercial sensors or estimation method fail to accurately measure under these conditions. For rut depth, the ability to make online measurements is almost unheard of and current manual measuring techniques are widely adopted. Botha (2015) showed a proof of concept that used DIC as a viable alternative to current measuring techniques that overcome the hurdles of measuring side-slip angle at low speeds and on off-road terrain and allowed for rut depth to be determined. However, these measurements were conducted offline from recorded footage and was not implemented in real-time. This opens an opportunity to optimise these algorithms such that they can operate in real-time.

For this reason, the remainder of the literature study was based on DIC. The basics of acquiring digital images were discussed. In order to perform DIC, intrinsic and extrinsic parameters of the camera need to be performed and therefore this was discussed for both the case of a single camera and multi-camera setups. For rut measurements, stereo-vision is required and was discussed. This includes epipolar geometry, rectification, and disparity mapping. The side-slip angle algorithm tracks change across images and therefore feature tracking was investigated.

CHAPTER 3 SIDE-SLIP ANGLE

Side-slip angle is an important, but inherently difficult parameter to measure as discussed in Section 2.2.1. This chapter describes the real-time implementation of the 2D planar algorithm developed by Botha and Els (2015) to directly measure side-slip angle. This was achieved by optimising the original algorithm and its parameters to increase performance and to allow a live video stream to be processed in real-time, instead of using pre-recorded footage. This chapter includes the detailed implementation and validation of the algorithm, the setup used for testing and the results obtained. The performance of the sensor is evaluated using two criteria, namely the maximum achievable sampling frequency and the accuracy of the measurements.

3.1 SPARSE OPTICAL FLOW ALGORITHM COMPARISON

As discussed in Section 2.2.1, feature tracking is used to estimate the direction that the vehicle is travelling in. A comparative study was conducted between the feature tracking algorithms available from the OpenCV library (these algorithms were discussed in Section 2.3.8) to determine which algorithm would be the best candidate for real-time applications. It should be emphasized that OpenCV implementations were used for all the algorithms. Table 3.1 shows the computation time required to identify and track 50 features across a 640x480 sized image on a desktop computer with an i5 quad core processor 2.8GHz, using a single core. Pre-recorded footage of a vehicle travelling over an asphalt road was used for this investigation as shown in Figure 3.1. From this, it is evident that the Lucas-Kanade (LK) algorithm is the least computationally expensive and was able to track the features in 15ms, which is less than half it's the closest competitor. The accuracy of the algorithms was not tested during the analysis as it is outweighed by the importance of computation time. Section 3.2 will discuss how outliers are rejected using random sample consensus (RANSAC) to ensure the accuracy of the motion.

3.2 ALGORITHM

The LK tracking algorithm, discussed in Section 2.3.8, is used to track features across consecutive images. The images are obtained from a single camera mounted to the vehicle, pointing downwards

Table 3.1. Various feature tracking algorithms performance comparison

Algorithm	Average Computation time [ms]
Lucas-Kanade	15
SIFT	570
SURF	95
ORB	35
BRISK	550

towards the terrain. Before features are identified, distortion is removed from the images. This is achieved by calibrating the camera to determine the intrinsic parameters, as well as distortion coefficients as discussed in Section 2.3.5. Features are identified on the image, $\boldsymbol{I}_t$ and their locations tracked to the next sequential image $\boldsymbol{I}_{t+1}$ using the LK tracking algorithm. Typically the features are corner edges or points with a large contrast in multiple directions. A summary of parameters used for the algorithm are given in Appendix B.1. It is therefore required to have texture on the testing surface that shows up as illumination variation or contrast in the image to sufficiently identify features. The input parameters for the LK algorithm are optimised for mainly reduced computational time. An example of the feature motion vectors produced are shown in Figure 3.1 where features were tracked between two consecutive images on a smooth concrete surface, where the feature motion vector starts at the red circle. The various colours represent inliers and outliers, which will be discussed later in this section.

The feature motion vectors are used to estimate the relative motion that the camera has undergone between frames. Since the camera is perpendicular (pointing downwards) to the terrain (as shown in Figure 3.3), the direction of the feature vectors can be considered the direction of motion of the camera, and therefore the vehicle. By mounting the camera such that the direction of the feature motion vectors are either horizontal or vertical when the side-slip angle is zero, then the angle that the feature motion moves in is considered the side-slip angle of the vehicle. For the feature vector motion to represent the motion of the vehicle, the assumption is made that the camera (and therefore the vehicle) undergoes predominantly planar motion. This means that the vehicle exhibits negligible motion such as yaw, pitch and roll as discussed in Section 2.2.1. An offset in the sensor can easily be corrected by measuring the side-slip angle when the side-slip angle should be zero, typically when driving in a straight line on a flat road (Botha and Els, 2015).

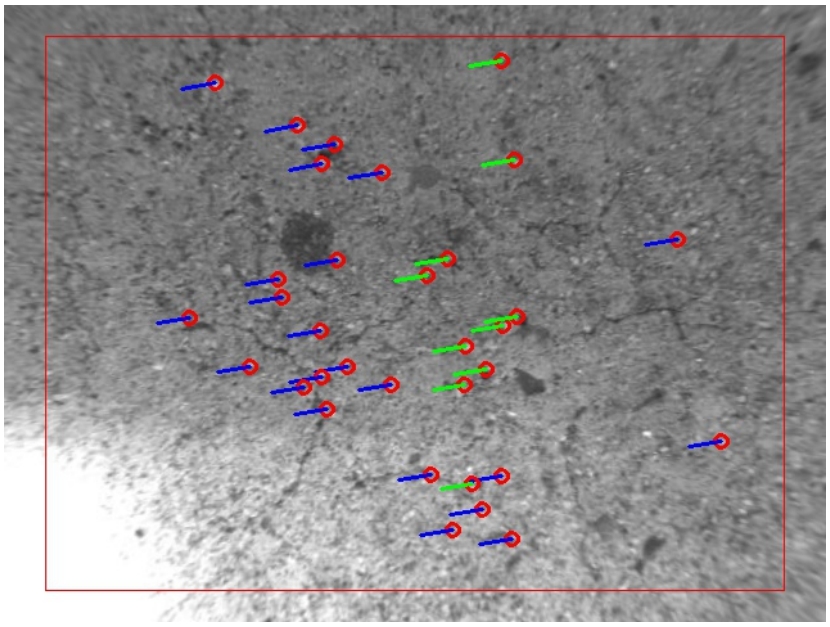


Figure 3.1. Features tracked across sequential images

Considering many features are tracked in one measurement, a method of determining outliers is required to ensure that the measurement is not obscured by outliers. The OpenCV implementation of the LK algorithm already contains a filtering technique whereby tracked features are rejected if their error is higher than a prescribed threshold. However, this fails to detect all outliers and therefore RANdom SAmple Consensus (RANSAC) is employed (Fischler and Bolles, 1981). RANSAC is an iterative algorithm that is used to estimate parameters of a model from data which contains outliers. In this case, only one parameter is required for the direction of the motion of the features and therefore only one point is required. The general idea is to randomly select the minimum samples required to build the model, which in this instance is one sample, from the data set and build a model from the selected samples. The rest of the data set is compared to this model with a suitable metric, in this case, the difference in angle. If the error of a point in the data set is more than a set threshold ($0.05 deg$), the point is rejected. The number of inliers is noted and those data points stored. The algorithm then again randomly selects a sample and builds a model based on the new sample. The rest of the data set is once again tested against this model. If the number of inliers is more than the previous model, the new points are stored. This process is repeated for a suitable number of iterations which is dependent mainly on the number of points being used and the order of the model. For this setup, only one parameter is required for the model that has a maximum of 50 points and therefore 100 iterations were used to keep computation cost to a minimum. Upon completion, the model with the most number of inliers is returned, as well as the inliers. A best-fit model is then generated with

the inliers and considered to be the correct model for the data set. This algorithm ensures an accurate model, free from the influences of outliers. Although this extra check increases computational expense, it is still minimal when compared to the feature tracking. Figure 3.1 shows features that have been rejected in blue and features that are classed as inliers as green.

The original algorithm developed by Botha and Els (2015) had a variable sampling frequency based on the displacement the vehicle had undergone between frames (D was used as the measure of displacement). If the displacement was too small, the frame was skipped until the displacement was above a set threshold of 5 *Pixels*. This ensured that motion vectors were adequately large for accurate measurements. This improved the algorithm performance at low speeds. However, it was not implemented in this investigation due to the complexity of multi-threading the algorithm. Instead, the data was filtered by checking the size of the displacement vector. If it was too small (2 *Pixels*), the measurement is rejected. Figure 3.2 shows a flow-diagram of the algorithm.

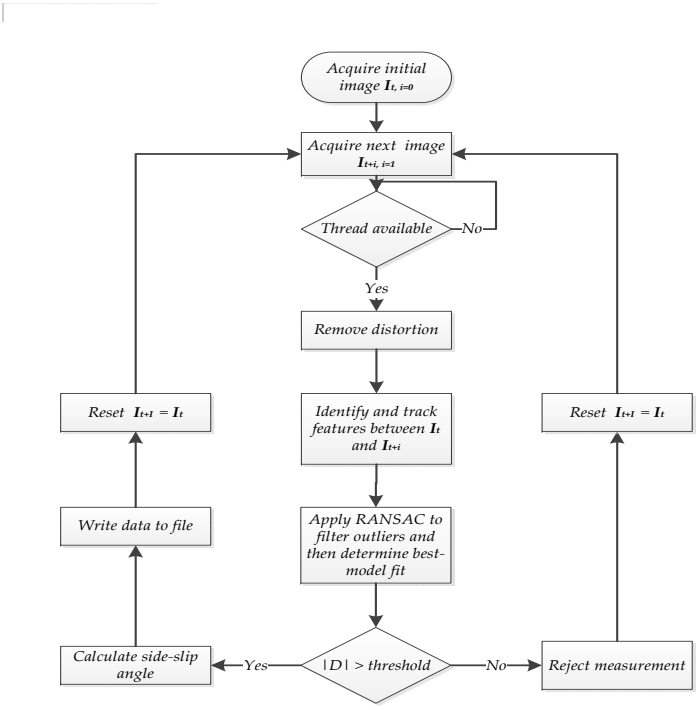


Figure 3.2. Flow diagram of side-slip algorithm

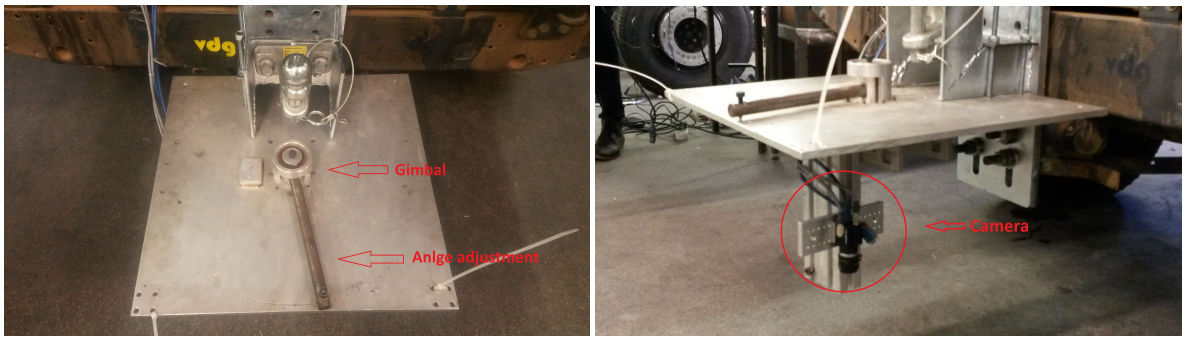
The algorithm was set to track 50 features. The number was kept low to keep computational costs at a minimum but allows enough points for redundancy. The algorithm was threaded across all cores to

boost the computational efficiency. This means that each core ran its own local version of the algorithm. This increases throughput (the overall sampling frequency of the system) but does not decrease latency (the time it takes to process a set of images). The main function would continuously check whether a thread was busy and then supply it with a set of images for processing once it became available. To further enhance computational efficiency, threads were continuously active and only destroyed at user request rather than starting and destroying a thread on each iteration, thereby keeping thread overhead at a minimum.

3.3 TEST SETUP

The side-slip angle measurement was tested by mounting the camera on a gimbal, which allowed rotation of the camera in fixed increments relative to the vehicle as shown in Figure 3.3. This forces a known artificial side-slip angle to be measured by the sensor in a controlled manner when the vehicle is driving in a straight line. This method was chosen as opposed to dynamic manoeuvres that would induce vehicle side-slip, as the measurements would not be controlled or repeatable. There is also no accurate comparative measure that would validate the measurements. The camera was mounted about 450mm from the testing surface, resulting in a field of view (FOV) of 480mm x 350mm. Ideally, the camera should be mounted on or close to the centre of gravity of the vehicle to minimise the effect of vehicle motion, however, for testing purposes, it was mounted on the rear of the vehicle. The vehicle was driven in a straight line on a flat concrete road and the side-slip angle incremented between 0, 5 and 10 deg. It should be noted however that the system can theoretically measure any angle but is limited because of the gimbal setup only allowing up to 10deg rotation.

The camera used in during this investigation is a Point Grey Flea3 USB 3.0 camera with Fujinon DF6HA-1B lenses (see Appendix A for camera and lens specifications). The camera was set to a resolution of 640x480 pixels with which a maximum frame rate of 450Hz can be achieved. Two methods for triggering the camera to take an image were investigated. In the first method, the software determined when an image should be taken. The second used a synchronizing pulse which is generated by the data acquisition device, referred to as hardware triggering. It was found that the software triggering limited the processing rate to 180Hz and therefore the decision was made to only use hardware triggering. This also enables the synchronization of other measurement devices if needed, making it the preferred triggering method. Due to the high sampling frequency, it is required for the



(a) Top view of gimbal setup **(b)** Side view of gimbal setup

Figure 3.3. Camera Gimbal Setup

camera to have a short shutter time. This reduces the exposure the camera sensor has to the scene and therefore more light is required. The aperture of the camera was set relatively large to allow high light exposure to the sensor. This decreases the reduces the depth of field. However, since the distance between the camera and the testing surface is fairly constant, a large depth of field is not required. The short shutter time also reduces motion blur on images. The processing unit used during this investigation was an Intel NUC with a 6th generation i7 quad core/8 thread processor that runs at 2.5GHz (Intel Corporation, 2016) with solid state hard drive and 16Gb DDR4 memory.

3.4 RESULTS

This section discusses the results obtained from testing the sensor. The results are split into two sections, namely the processing time and accuracy of the measurements that is followed by a discussion of the results.

3.4.1 Processing time

Initial performance evaluation was focussed on the maximum obtainable sampling frequency. This was accomplished by increasing the sampling frequency until the computation time became excessive and the algorithm started skipping frames. As a result, it was found that 250Hz was the maximum obtainable sampling frequency with no frame skipping. Setting the sampling frequency to 300Hz resulted in an average sampling frequency of 283Hz and increasing it further to 350Hz resulted in an actual sampling frequency of 302Hz. Although higher sampling frequencies were obtained, they are

Table 3.2. Side-slip angle results on flat concrete at 250Hz

Side-slip angle [deg]	Run	Mean [deg]	Standard deviation
0	1	-0.0671	0.4621
	2	0.0598	0.5030
	3	0.3620	0.4990
5	1	5.053	0.6144
	2	5.399	0.5024
	3	5.10	0.6914
10	1	10.46	0.5873
	2	10.43	0.5742
	3	10.06	0.6507

not consistent and the skipped frames create ambiguities in the measurements. This sensor matches the sampling frequency of the commercial sensor, the Correvit S-HR.

3.4.2 Accuracy

The second part of the performance evaluation was focussed on the accuracy of the sensor. This was achieved by setting the side-slip angle to a predefined value on the gimbal setup and then driving in a straight line. The actual side-slip angle is then compared to the measured side-slip angle. The sampling frequency was set to 250Hz which is the maximum obtainable sampling frequency. In these tests, the vehicle was accelerated from rest up to 60km/h. Table 3.2 shows the results where the side-slip angle was set and compared to the average side-slip angle that was measured for that run. The STD for each run is also added as an indication of the noise levels. The results show that a maximum of 0.46deg error in the side-slip angle was obtained throughout testing, although most measurements were considerably closer with a mean error of 0.25deg. Considering the STD, the mean is 0.56deg but not exceed 0.7deg. This is comparable to the accuracies of Botha (2015), which implies results are better than commercial. It should be noted that the testing surface was not completely flat and at speed, caused motions on the vehicle which could explain some of the noise experienced. Figure 3.4 shows a test where the side-slip angle was set to 10deg and (a) shows the pixel displacement or size of the motion vector (D) in *pixels*. Coincidently D correlates well with the actual vehicle velocity, in

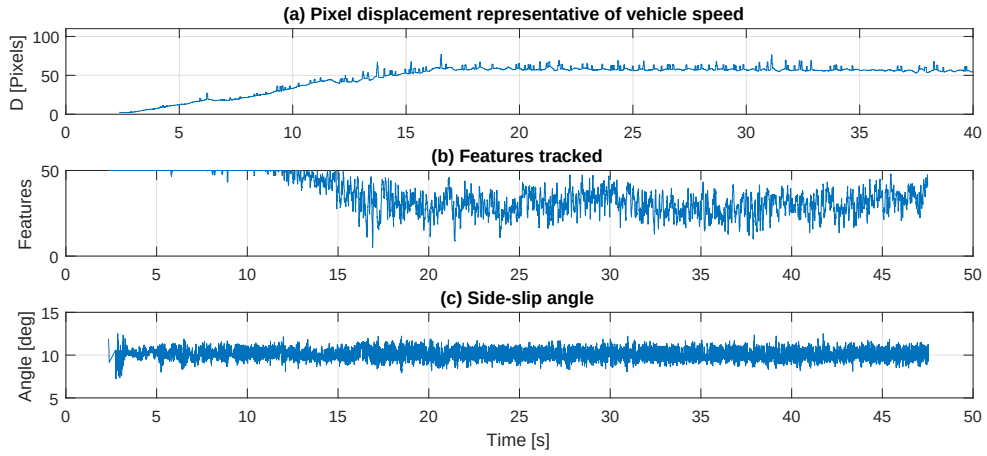


Figure 3.4. Results obtained with side-slip angle set at 10 deg

km/h, that was obtained from the vehicle‘s speedometer and was used as an approximation towards the vehicle velocity. This assumption is only valid for this specific setup and does not generally hold. For future investigations other means of vehicle velocity are required. Figure 3.4 (b) shows the number of features that were successfully tracked. The decrease in this number will be discussed in the following section. Figure 3.4 (c) shows the side-slip angle measurement which measures 10deg as expected.

3.4.3 Discussion

A commercial sensor, such as the Correvit S-HR, was not available during this investigation and no direct comparison could be made. However, these results are comparable to that of Botha and Els (2015) who used a similar experimental setup on the same test surface (refer to Table 2.1) for both the DIC methods and the Correvit S-HR sensor. Comparing the results to those measurements, it is evident that similar accuracies were obtained proving that the camera based sensor outperforms the commercial sensor. Lower noise bands are also observed.

For the accuracy measurements, the vehicle velocity was limited to 60km/h. However, to further test the capabilities of the sensor tests were conducted at higher speeds. It was found that measurements could still be made up to 100km/h which was the maximum obtainable speed for the test vehicle on the test surface. This is also far better than required for most off-road testing on rough terrain. Figure 3.5 shows a test where the side-slip angle was set to 5deg and the vehicle accelerated from rest to 100km/h.

Figure 3.5 (a) shows the pixel displacement and therefore an estimate of the velocity as explained. Figure 3.5 (b) shows the number of features tracked during the measurement. As the vehicle velocity increases, the ability to track features also decreases. At 100km/h it appears only a small portion of the 50 features are correctly tracked. Although only one point is required to make a measurement, there is zero redundancy and no room for error. Even at the lower speeds of 60km/h where the accuracy

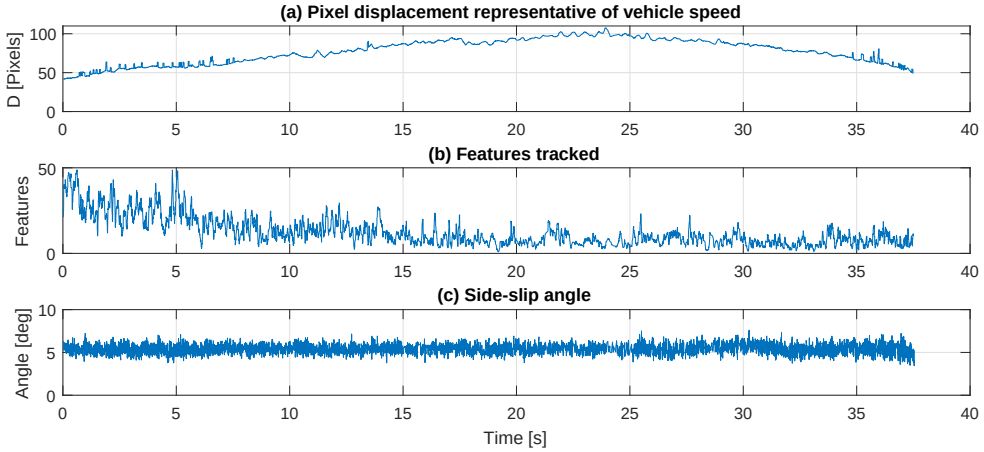


Figure 3.5. (a) Pixel displacement which represents velocity (b) the number of features that were successfully tracked (c) the measured side-slip angle which was set to 5 deg

tests were conducted, a reduction in features tracked occurs although not to the extent of 100km/h. There are three possible causes for this phenomenon. The first is blurring on the images caused by the high velocity of the vehicle. This could be reduced by decreasing the shutter time and decreasing the aperture but will require better lighting. The second is the assumption of small motion made by the LK tracking algorithm being violated. The motion can be reduced by increasing the distance between the camera and the ground. This would also reduce blurring caused by the motion, however, the accuracy at lower speeds will be reduced due to the larger field of view. Variable focal length can also be used to achieve the same effect. The focal length can be adjusted either manually or automatically based on the velocity at which the test occurs. The last possibility is the lack of overlap between frames. Figure 3.6 shows the distance the vehicle travels between samples as the speed increases. At 100km/h, there is 110mm displacement between samples. This results in a 70% overlap between images. By rotating the camera 90deg such that the features are tracked in the width rather than the length, the overlap is increased to 78%. The overlap is important as only features in this region can be tracked. If features are chosen where there is no overlap, the feature cannot be tracked and the feature is therefore lost. As mentioned before, the pixel displacement was used as an approximation towards the vehicle velocity. If a calibration is done to find the correlation between the vehicle velocity and D, the exact vehicle

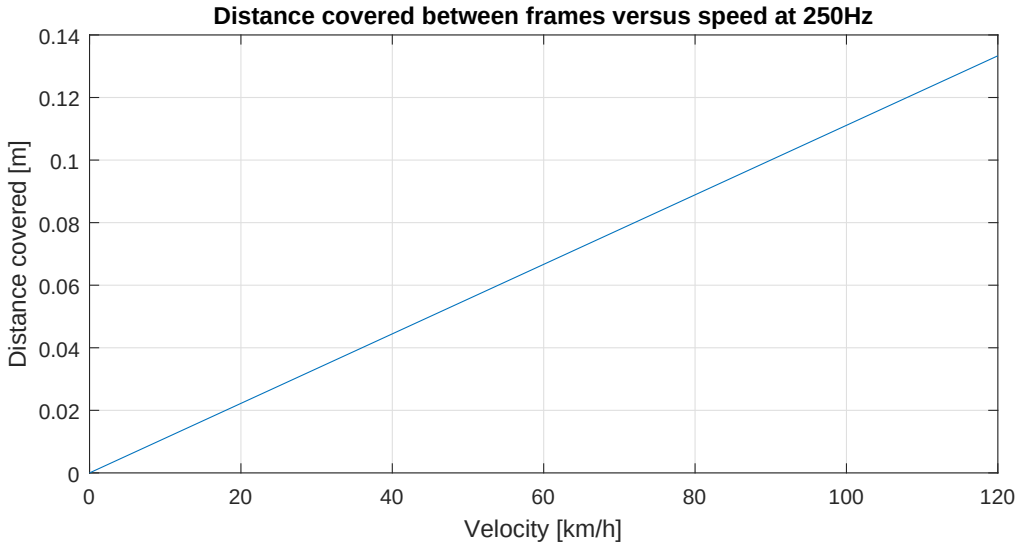


Figure 3.6. Distance travelled between frames at 250Hz

velocity can be measured by the sensor in addition to the side-slip angle. This could supply ADAS with an alternative vehicle velocity measurement during dynamic manoeuvres where the velocity determined by the wheel speed is complex and generally unreliable, especially at lower speeds.

3.5 CONCLUSION

From the results obtained it is clear that the real-time implementation of the camera-based side-slip angle sensor was successful. A maximum sampling frequency of 250Hz was obtained. The mean error across measurements was 0.25 deg with a mean STD of 0.56deg. Although no direct comparison can be made between the camera based sensor and the commercial sensor, a comparison can be made to previous results obtained that used the same experimental setup. This proved that the camera-based sensor outperforms the commercial sensor in accuracy and with lower noise bands. It also shows that, although no tests were conducted on rough off-road terrains, the sensor will still outperform the commercial sensor on these rough terrains. Supplementary tests showed that the sensor could measure up to 100km/h which is close to highway speeds. This is also far better than required for most off-road testing on rough terrain, showing excellent capabilities. In conclusion, the camera based sensor successfully provides a cheaper alternative to current methods by using inexpensive, off-the-shelf cameras with dedicated software.

3.6 RECOMMENDATIONS

Although the investigation was deemed successful, various improvements can be made. Firstly, the cause for the reduction in features tracked at higher speeds can be identified. This will improve the accuracy of the measurements at high speeds and possibly allow for even higher operating speeds. The overlap can be increased by rotating the camera as to allow the features to move along the width of the camera which has a larger field of view. Next would be to mount the sensor higher on the vehicle, increasing the distance between the camera and the ground. This would reduce the motion observed by the sensor, reducing any blurring on the image and will aid the small motion assumption made by the LK tracking algorithm. The same effect can be achieved by changing the focal length. This decreases the accuracy at low speeds. However, using a lens with a variable focal length that changed depending on the velocity the accuracy could be maintained. The next improvement would be to incorporate the variable sampling frequency in the algorithm as done by Botha (2015). By calibrating the system as to determine the exact relationship between the actual vehicle velocity and D (making the assumption that the sensor is fixed to the vehicle and its position will not change), the system could provide the velocity of the vehicle independent of other measurements. The sensor should also be tested on terrains that may not have large texture variation such as snow and ice, where it would be difficult to identify and track features. The sensor should be tested on off-road like terrains that would induce vehicle motion such as roll, pitch and yaw to establish the effect of on the measurement. Lastly, the output of the sensor should be incorporated in current ADAS to improve performance, or vehicle stability programs.

CHAPTER 4 RUT DEPTH MEASUREMENTS

DIC has proved to be a viable alternative to current rut measurement techniques as discussed in Section 2.2.2. Inexpensive, off-the-shelf cameras with dedicated DIC software can be used to measure rut depth, as opposed to current contact-based methods. The following section describes the real-time implementation and validation of the rut depth sensor. This includes the detailed implementation of the algorithms, the setup used for testing and the results obtained. A stereo-vision rig is used to recover 3D coordinates of the disturbed terrain from which the rut depth can be obtained. Three methods for extracting the rut depth are investigated. The first method averages the heights over a predefined area that represent the rut and ground plane datum. The second and third methods attempt to automatically detect the rut from the 3D map and then calculates the rut depth by fitting a horizontal plane through the rut and ground plane datums. The performance of the sensor is evaluated using two criteria, i.e. the maximum sampling frequency and the accuracy of the measurements.

4.1 ALGORITHMS

Before any measurements can be made it is first required to calibrate the stereo-vision rig in order to determine the intrinsic and distortion coefficients of each camera and the geometric relationship between one another (discussed in Section 2.3.5 and 2.3.6). The rut measurement is made by first determining the 3D map of the terrain using the calibrated stereo-vision setup. This is done by simultaneously taking a set images of the scene using the stereo-vision rig. Distortion is removed from the image set after which they are rectified. The corresponding left and right images are matched to determine the disparity map using the BM algorithm. The minimum and maximum disparities as well as the window size were optimally tuned to the scene as to reduce computational requirement. The disparity map is then converted into 3D coordinates resulting in a 3D surface profile of the rut. Figure 4.1 shows a typical output of a surface profile once it has been converted into 3D coordinates, where the vehicle was travelling along the y-axis. The z-axis represents the distance between the camera and the terrain. The rut depth is measured by taking the difference between the ground plane datum and ”floor” of the rut. It is therefore required to determine where the rut is located on the 3D surface profile. The most simplistic method for determining the ground plane and rut floor is by manually defining these locations. Before rut-depth can be measured, the location of the boundaries of the rut in the 3D image

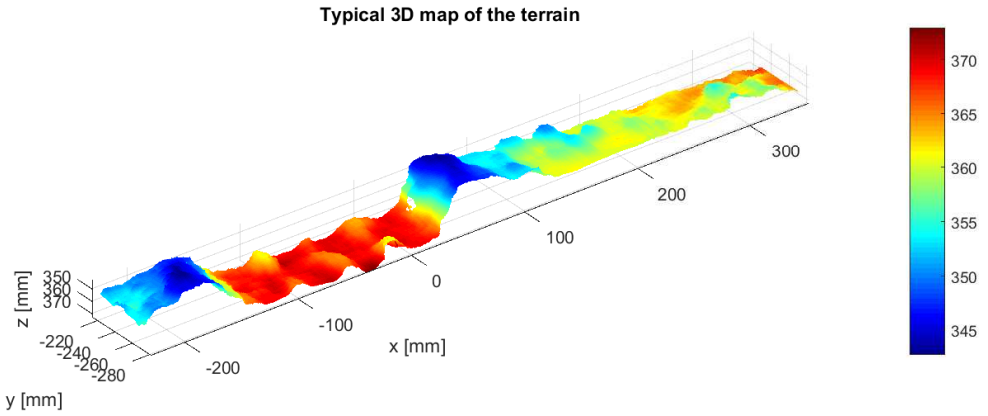


Figure 4.1. Reduced surface profile from stereo-vision rig in mud

has to be set. These locations are hard-coded into the algorithm, fixing them throughout measurements. This method does not accommodate any changes in the rut's location that might occur such as when the vehicle is turning. The rut floor and ground plane datum heights are then calculated by simply averaging over the predefined area. An example is shown in Figure 4.2 where the top view of a surface profile is shown with the red rectangles demarcating the areas for the ground plane and rut floor and the green rectangle demarcating the pile. The colour bar shows the distance between the camera and the terrain. Large outliers can skew mean values and therefore averaging was calculated by only selecting values in the predefined area that fell within a set range height range.

Figure 4.2. Top view of surface profile showing averaged areas

In order to accommodate changes in the rut's location, two algorithms were devised that automatically detected the boundary locations of the rut. Due to the shape of the rut (refer to Figure 2.5 in Section 2.2.2), the pile peaks can be used to find the boundaries as they typically represent the maximum heights of the rut. Therefore if these locations can be determined, points in between can be identified as the rut and points to the outer sides of the piles can be used as the ground plane. To start, the algorithm reduces the surface profile to a single lateral cross-section of the surface profile. This is done by simply averaging the height values along the y-axis (direction of travel of the vehicle).

Two methods of identifying the piles were used during this investigation. The gradient detection method identified turning points on the profile by finding where the first derivative of the rut profile

is equal to zero. A rolling average was applied to the derivative to filter the data. Alternatively, the polynomial detection method used a windowed 2nd order model (parabola) fit to aid the search for the turning points. The model was fit across a window of size N and the turning point of the model calculated. If the turning point of the model was within the window, it represents a turning point on the profile. The window is then shifted to the location of the turning point. If it did not contain a turning point, the window was incrementally shifted across the whole profile and checked if it contains a turning point at each iteration.

The two turning points with minimum heights represent the peak pile heights. Although the pile peaks are actually the highest points on the rut profile, the term minimum is used as they are represented by the minimum distance from the camera. It should, however, be noted that the rut profiles are not always ideal as rocks and debris cause ambiguity in the actual location of the piles. To combat this, the algorithm checked that the candidate pile locations were between 200mm and 300mm apart (width of a typical passenger vehicle tyre) and that the average height between the piles was more than the height to the outer sides of the pile.

Once the rut locations have been determined, a 0th order model fit (a horizontal line) with RANSAC is applied to the points in between the piles to determine the rut floor. Similarly, a 0th order model fit with RANSAC is applied to the points to the right of the right-hand side pile, to determine the ground plane. Using RANSAC increases the accuracy of the model if the exact location of the piles has not been found. This is because the rut and the ground plane will be dominated by a horizontal line and any points that deviate from this will be rejected. The rut depth is then taken as the difference between the two planes. Appendix B.2 summarises the main parameters used for these algorithms. Figure 4.3 shows the typical output of the polynomial fit detection algorithm where the rut has been located and planes fitted to the rut floor and ground datum. The y-axis on the figure represents the distance from the camera/sensor and the x-axis is the image width in Pixels. The rut depth was calculated to be 8.5mm.

Figure 4.4 shows the output of the gradient detection algorithm from the same map where the rut has been located and the planes fitted to the rut floor and ground datum. As before, the y-axis (left) represents the distance between the sensor and the x-axis is the image width in Pixels. The y-axis (right) represents the filtered gradient of the rut profile and it is evident points where the gradient is zero corresponds to turning points on the rut profile. The rut depth was calculated to be 8.5mm which

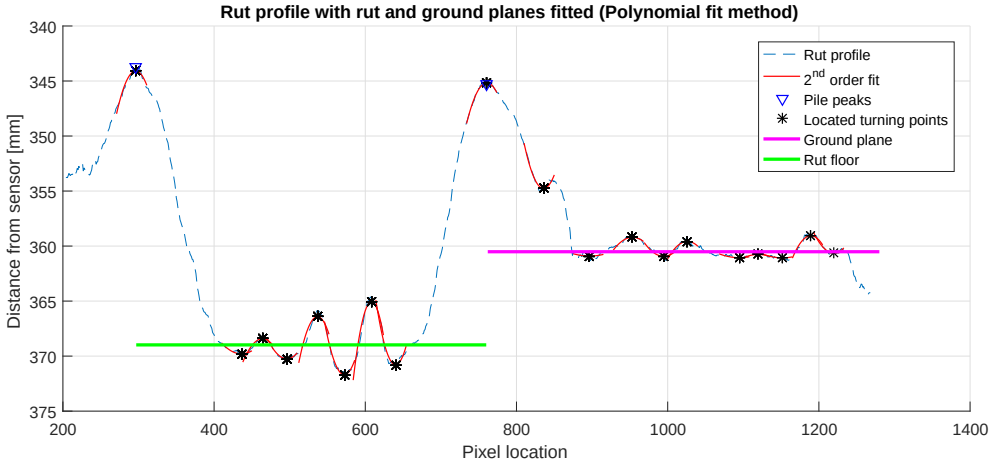


Figure 4.3. Typical output of the polynomial detection algorithm

is the same as the polynomial fit detection method. In comparison, the manual method resulted in a rut depth of 8.6mm which corresponds well with the gradient detection method and the polynomial detection method.

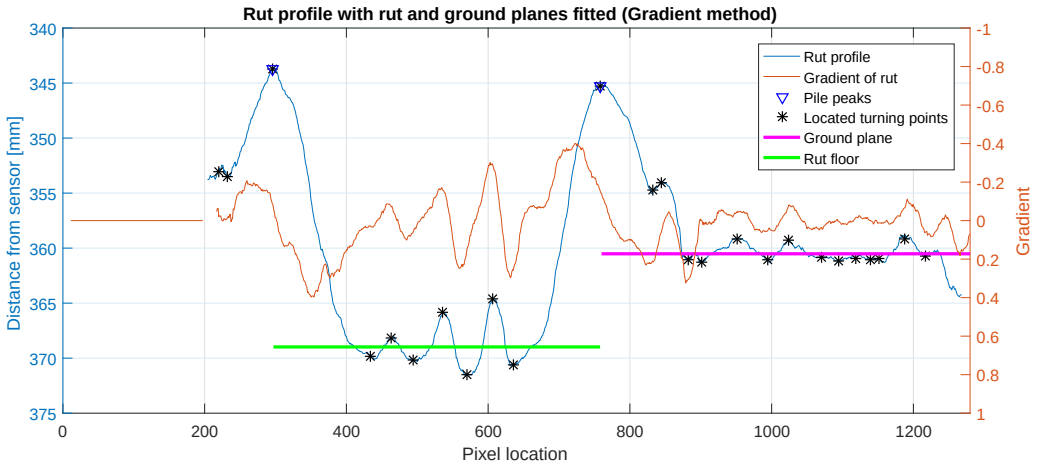


Figure 4.4. Typical output of the gradient detection algorithm

A basic filter was added to the algorithm to reject a rut depth measurement if it exceeded a threshold. It was found that in the event that the rut locations where not correctly identified, the rut location detection algorithms mistakenly identified the rut and ground planes as the same plane or calculates one of the planes to have no height. The resulting rut depth is then either very small (typically zero) or the distance between the camera and the terrain, which is typically two orders of magnitude larger than the actual rut depth. These extreme values are easily identified and simply neglected by the system.

4.2 TEST SETUP

Ideally, the rut depth sensor should be tested by mounting the stereo-vision rig to the vehicle, behind one of the rear wheels as to measure the depth of the deformed terrain as the vehicle is driving. However, it is then difficult to correlate manual measurements with the sensor‘s measurements. Instead, a man-made rut was used with three height levels that can be validated using manual measurements. The rut used is shown in Figure 4.5(b). The stereo-vision rig was mounted on an x-y table and positioned over the rut at approximately the same height that it would be mounted to the vehicle. The x-y table allowed the cameras to move along the length of the rut to simulate the vehicle driving forward. The test rig is shown in Figure 4.5(a). Each algorithm was tested by moving the camera along the rut as to

(a) Camera mounted on a x-y table **(b)** The pre-formed rut used for testing

Figure 4.5. Rut measurement test setup

measure the three rut depths. The sensor outputs not only the measured rut depth but also computation times for various sections of the algorithm to gain a better understanding of the possible bottlenecks in sampling frequency. Each test was repeated for a minimum of three times to ensure repeatability. Figure 4.6 shows a 3D map that was generated from the setup. The colour on the image represents the height of that location, donated by the colour bar on the right. The figure shows regions A, B, and C that are the different depths of the rut. Region D (to the left and right of the right), indicate the rut piles.

The cameras used during this investigation were Point Grey Flea3 USB 3.0 cameras with Fujinon DF6HA-1B lenses (see Appendix A for further details). The cameras were set to a resolution of

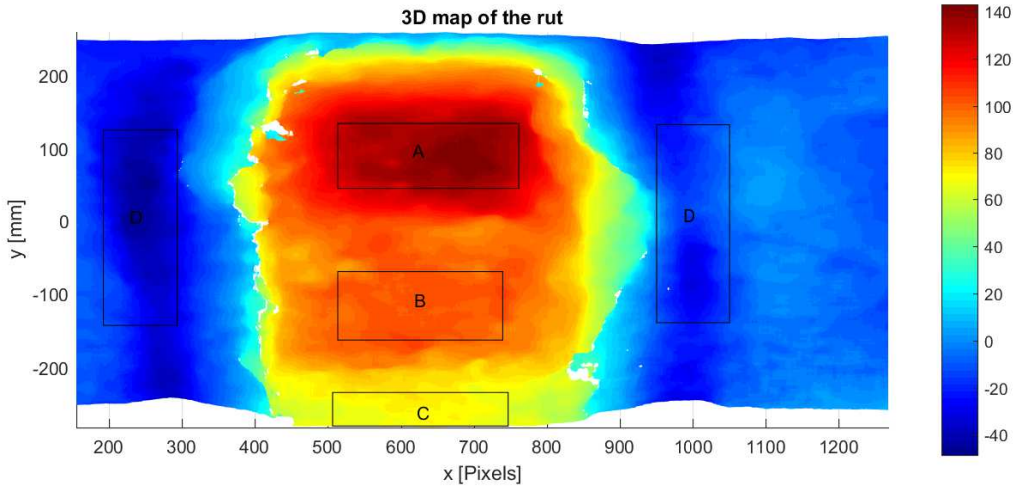


Figure 4.6. 3D Map of the rut

1280x960 pixels. An average of 1100x900 points are in view of both cameras for which a grid of 3D points can be obtained. It should be noted that this high resolution was used to ensure the whole width of the rut is in the field of view of the cameras. A higher resolution increases computation time as more pixels need to be matched between the left and right images. However, due to the fact that only a small longitudinal section of the rut it required, the matching was reduced to a 1280x90 window. This decreases computation time but still allows the required field of view. The theoretical depth resolution was calculated to be 0.25mm using Equation 2.21 for this setup. An Acer laptop with a core i7 dual-core processor that runs at 2GHz with 8Gb DDR3 ram was used for processing during this investigation.

In this setup, the software triggered the cameras to capture images. In Section 3.3 it was stated that software triggering limited the camera frame rate to 180Hz, however, due to the much lower sampling frequency of the system, it is deemed sufficient. As with the side-slip angle measurements, the shutter time was set as low as possible with a small aperture to reduce any blurring effects caused by the vehicle movements.

4.3 RESULTS

The performance of the sensor is split into two categories, namely processing time and accuracy of the measurements. The overall sampling frequency of the sensor is discussed with the various algorithms, however, to better understand what the bottlenecks in the processing times are, the computation time of various sections of the algorithms are studied. The accuracy of the depth measurement is discussed by comparing the measurements to manual measurements of the rut depth.

4.3.1 Processing time

Table 4.1 shows the average sampling frequencies that were obtained from the sensor for each test, as well as the standard deviation of the sampling frequency.

Table 4.1. Rut depth measurements results: Sample frequencies

Method	Run	Sample Freq [Hz]	STD [Hz]
Averaging	1	7.7	0.09
Averaging	2	7.8	0.09
Averaging	3	7.7	0.06
Polynomial fit	1	2.6	0.15
Polynomial fit	2	2.6	0.19
Polynomial fit	3	2.6	0.18
Gradient	1	4.2	0.03
Gradient	2	4.2	0.04
Gradient	3	4.2	0.03

Each algorithm showed a relatively constant sampling frequency, with low noise bands. The simple averaging method, substantially outperforms the other methods with a sampling frequency of 7.7Hz. The gradient method had an average sampling frequency of 4.2Hz and the polynomial fit method had a sampling frequency 2.6Hz. These sampling frequencies are relatively low and might seem insignificant, however, to current manual measurements these are major improvement in productivity. Figure 4.7 shows the distance a vehicle would travel at various speeds and sampling frequencies. From this, it can be seen that with the sampling frequencies obtained, at a speed of 15 km/h which is a typical speed

when traversing deformable terrains, a sample would be taken between every 0.5-1.5m which is less than one revolution of the wheel.

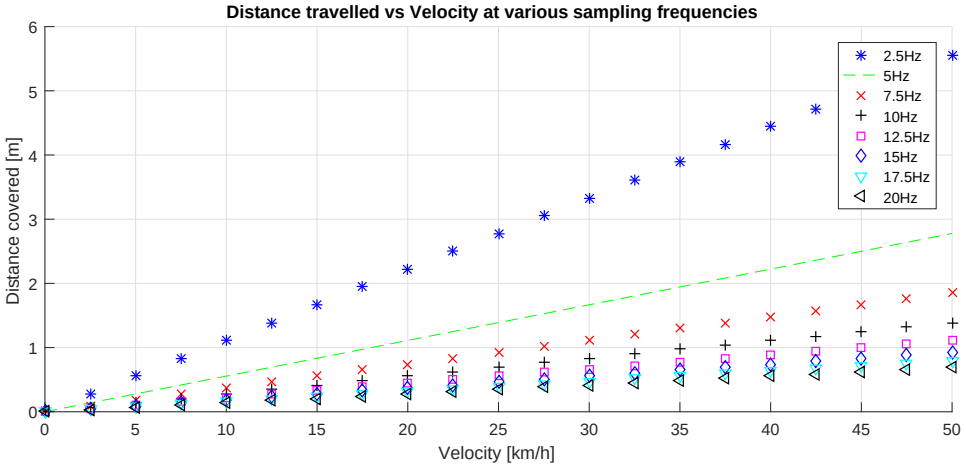


Figure 4.7. Distance travelled for various speeds at different sampling frequencies

To fully understand the computational bottleneck within algorithms, the computation time of each section of each algorithm was recorded and is summarised in Table 4.2. The simple averaging method consists of two major sections, obtaining the disparity map between the left and right images, and averaging the pre-defined regions. From Table 4.2 it is clear that the disparity mapping dominates the computation time, taking 0.125s. The more complex methods that attempt to locate the rut automatically consist of four major sections i.e., disparity mapping, finding peaks in the rut profile, locating the rut from the peaks and fitting planes to rut floor and ground datum. For the polynomial fit method, the disparity mapping takes 0.125s, finding the peaks take 0.15s, locating the rut is essentially negligible and fitting the planes takes 0.11s. For the gradient method, the computation time is dominated by the disparity mapping (0.125s) and fitting the planes (0.11s) and the rest is negligible.

Table 4.2. Rut depth measurements results: Processing times

Method	Run	Disparity [s]	Averaging [s]	Finding Peak [s]	Locating rut [s]	Fitting planes [s]
Averaging	1	0.125	2.70E-04	NA	NA	NA
Averaging	2	0.125	2.70E-04	NA	NA	NA
Averaging	3	0.125	2.70E-04	NA	NA	NA
Polynomial fit	1	0.125	NA	0.150	7.69E-05	0.108
Polynomial fit	2	0.125	NA	0.148	8.93E-05	0.109
Polynomial fit	3	0.125	NA	0.146	9.09E-05	0.110
Gradient	1	0.125	NA	7.60E-05	7.40E-05	0.109
Gradient	2	0.125	NA	8.79E-05	6.59E-05	0.108
Gradient	3	0.125	NA	1.12E-04	5.75E-05	0.108

One common bottleneck between all methods is the disparity mapping. However, as stated before, this can be significantly reduced if implemented on a GPU. Finding the rut's location using the polynomial detection method proved to be computationally expensive when compared to the gradient detection method. Only a single core was used of the CPU. This could potentially sped up by multi-threading these algorithms. Both detection methods fitted planes to these locations using RANSAC to improve their accuracy. Each plane (the rut floor and the ground datum) contains many points (typically hundreds) and therefore many iterations are required (typically thousands) to successfully reject outliers. From Table 4.2 it is evident that this is computationally expensive.

In order to achieve higher sampling frequencies, a combination of these methods could be used. Using a GPU to obtain the disparity mapping would substantially decrease the computation time to determine the disparity. If the gradient method is used to locate the rut, and the planes found by averaging the heights rather than fitting a plane through the points, the total computation time would be reduced, increasing sampling frequency.

4.3.2 Accuracy

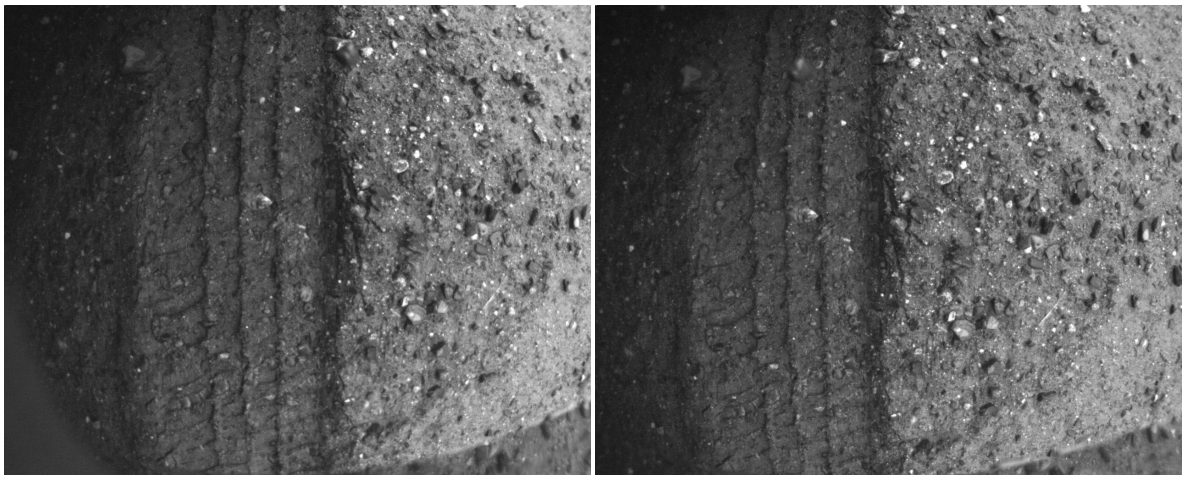
The accuracy of the measurements was tested against manual measurements. The manual measurements were conducted as described in Section 2.2.2. The results are summarised in Table 4.3. Each method

shows fairly consistent measurements, although all methods under estimated the rut depth. From the table is evident that the simple averaging method was the most accurate and had a mean error of 1.5mm (1.8%), the gradient method was second most accurate and had a mean error of 3.1mm (3.5%) and the polynomial fit method had an error of 10.6mm (12.2%). The measurements between the simple averaging method and the gradient method correlate well with each other, typically within 1mm. However, the polynomial fit method consistently measured an offset between 8-9mm lower than the other methods. This is possibly due to the method failing to detect the correct peaks although further investigation is required. All parameters relating to locating the rut from the peaks and fitting the planes to the rut floor and ground datum between the gradient method and the polynomial fit method were kept constant leading to the possibility that there is a mistake in the implementation of the algorithm.

Table 4.3. Rut depth measurements results: Depth

Method	Run	Rut depth 1 [mm[	Rut depth 2 [mm[	Rut depth 3 [mm[
Manual	N/A	64	103	139
Averaging	1	62.0	102.0	137.1
Averaging	2	62.0	102.0	137.0
Averaging	3	62.0	102.0	137.9
Mean error	N/A	3.1%	0.9%	1.4%
Polynomial fit	1	51.1	94.7	128.0
Polynomial fit	2	50.9	94.3	128.9
Polynomial fit	3	51.1	94.8	128.6
Mean error	N/A	20.4%	8.4%	7.9%
Gradient	1	60.8	100.8	135.4
Gradient	2	61.6	100.5	134.7
Gradient	3	60.8	100.4	134.7
Mean error	N/A	5.0%	2.5%	3.1%

Various factors contribute to the accuracy of the measurement. Firstly, considering that all measurements made from the sensor were lower than the manual measurements leads to the possibility that the manual measurements were inaccurate. The size of the area of the averaging method, as well as



(a) Left camera view **(b)** Right camera view

Figure 4.8. Typical view of a formed rut

where the area is located, could either increase or decrease the accuracy of the measurement. A larger area would be less prone to outliers (e.g. small stones etc.) and would better represent the height of that area. The window size used for the polynomial fit did not seem to greatly affect the algorithm, as long as it was large enough to capture the curvature. The number of iterations used for the RANSAC algorithm also plays an important part in accurately determining the height of the plane although this comes at the cost of computation time.

4.3.3 Recorded footage

The field tests prove that the algorithms are capable of measuring the rut depth, with varying degree of accuracy. For clarification, the algorithms were tested on pre-recorded footage to demonstrate their capability under real-world conditions over longer periods. The footage was recorded from 1977 Jeep Cherokee that traversing soft, muddy soil. The stereo-vision cameras were mounted behind the left front wheel beneath the chassis, to have full view of the permanent deformation in the soil caused by the wheel. Figure 4.8 shows the typical view obtained from the stereo-vision cameras, where the left and right views are shown. The formed rut is clearly visible in both views.

Figure 4.9 shows the calculated rut depth from the pre-recorded footage, where the x-axis represent the frame number and the y-axis the depth of the rut in *mm*. It can be seen that all three algorithms, (the averaging algorithm, the polynomial detection algorithm and the gradient detection algorithm)

correspond to the one another throughout the video. The variation in rut depth is likely due to change in soil characteristics. No manual hand measurements were conducted for this footage as it difficult to accurately synchronise the hand measurements with the system's measurements. However, the algorithms have been validated in the previous section and therefore no further validation is required. It should however be noted that the rut depths and pile heights are relatively small in sections of the footage. It was found that the rut location detection algorithms perform poorly in these conditions and from the 760 frames of the footage, almost 100 measurements are rejected due to the filter implemented for both the polynomial detection algorithm and gradient detection algorithm. No further investigation was conducted to determine whether this was due to the small rut depth or whether it was due to the minimal pile heights. Interestingly there is no offset in the polynomial detection algorithm as found in the validation tests in Section 4.3.2. This further points to the fact that there is a mistake in the algorithm.

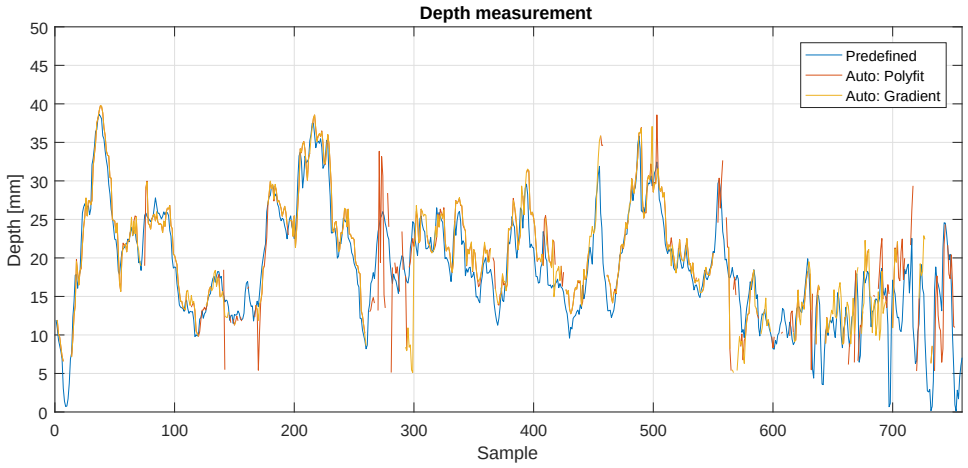


Figure 4.9. Rut depth results from pre-recorded footage

No investigation was conducted on the processing times of the algorithms on the footage. This is due to the total processing time largely being dominated by reading and decoding the images from the hard drives and does not represent real-time conditions.

4.4 CONCLUSION

From the results, it is clear that the real-time implementation of the camera-based rut-depth sensor was successful. The simplest method proved to have the highest sampling frequency of 7.7Hz and

also was the most accurate with a mean error of 1.8%. Although the area over which the algorithm averages are fixed, the use of other vehicle parameters can be used to estimate movement in the rut's location on the image. The polynomial fit method proved to be computationally expensive and had a mean sampling frequency of 2.6Hz. It also was deemed the least accurate out of all the methods with a mean error of 12.2%. The gradient method had a mean sampling frequency of 4.2Hz and a mean error of 3.5%. The methods that attempted to automatically detect the rut's location were slow and were not as accurate as the averaging method. However slow, they are still an improvement over current methods whereby the rut depth is measured by hand. Even without the accuracy of the averaging method, the measurements could still be used as a good approximation towards the rut depth. The computation times of the algorithms were critically evaluated and found that one common bottleneck is the disparity mapping. Pre-recorded footage showed the capabilities of the algorithm in real-world scenarios and compared well within one another. This leads to the possibility that there is a mistake in the implementation of the polynomial detection method that caused the offset in the validation of the algorithms.

4.5 RECOMMENDATIONS

Although the study was successful, a few specific aspects need to be addressed to improve the results. The first would be to investigate how much the rut location would move during dynamic manoeuvres. This might eliminate the need for automatically detecting the rut. If the sensor had other information available, such as side-slip angle and steering angle, the location of the rut could easily be determined based on the geometry of the vehicle eliminating the need to automatically detect the location. The study can be further improved by considering both the left and right ground datums and not only one side, as in this study. This would require a larger field of view which could be achieved by either increasing the height between the sensor and the ground or using wide-angle lenses. It should be noted that the automatic rut detection algorithms make the assumption that the rut has piles along its side. No investigation was conducted to test the performance in the absence of piles or when the rut depth is small and should be investigated. The implementation of the polynomial detection algorithm needs to be critically evaluated to determine the cause for the offset in measurements in the validation testing. One common bottleneck in computation time between the algorithms was the disparity mapping. The use of GPU's would greatly reduce this computation time and increase the sampling frequency. All the algorithms could be multi-threaded to potentially increase throughput. Further, a combination of

the algorithms could be used to exploit the best properties of each. If, the gradient method is used to locate the rut and the ground plane and rut floor calculated by averaging, the computation time would be reduced substantially. Often rut depth measurements are conducted on snow and ice for environmental impact studies. Snow and ice pose the problem that although the terrain has texture, it does not necessarily show as illumination variation on an image making it difficult to track. The applicability of using this sensor on snow would be beneficial.

CHAPTER 5 CONCLUSION AND RECOMMENDATIONS

During this investigation, Digital Image Correlation, or DIC, was successfully used to measure vehicle dynamic properties in real time. These techniques provided better results and proved to be viable alternatives to current measuring techniques. The use of DIC in vehicle dynamics is still vastly unknown with limitless possibilities yet to be discovered.

Two parameters in vehicle dynamics were measured using DIC, namely side-slip angle and rut depth. The basic definition of these parameters were discussed, as well as current measuring techniques and their shortcomings. For side-slip angle, the investigation shows that there is a need for an alternative measuring method on off-road terrains and at low speeds as current commercial sensors or estimation methods fail to perform under these conditions and are prohibitively expensive. The ability to measure rut depth in real-time is none-existent and current manual measuring techniques are widely adopted. Botha (2015) showed a proof of concept that used DIC as a viable alternative to current measuring techniques. However, these measurements were conducted offline from recorded footage. This opened the opportunity to optimise these algorithms such that they could operate in real-time.

DIC requires digital images therefore the basic concepts for acquiring digital images were discussed. A compromise between camera settings are required to obtain clear and sharp digital images, which is discussed. Mathematical camera models such as camera intrinsic and extrinsic parameters and lens distortion are discussed, where these parameters are determined through calibration.

The DIC algorithms and techniques used during this investigation are discussed. For the side-slip angle measurements, features need to be identified and tracked using sparse optical flow techniques. A short study was conducted to determine which optical flow algorithm (from the OpenCV library) had the lowest computation time. Rut depth measurements required full DIC to obtain a disparity map from a stereo-vision rig which is used to map pixels to 3D coordinates. Disparity mapping requires the image pair to be rectified, where after regions of pixels are matched using a correlation metric and a winner-takes-all matching strategy.

SIDE-SLIP ANGLE

The real-time side-slip angle sensor was successfully implemented during this investigation. The sensor consists of a single camera that was mounted to the vehicle with dedicated software. The software identified and tracks features across consecutive images. Considering the vehicle and thus sensor, undergoes predominantly horizontal planar motion, all motion feature vectors point in the same direction. By mounting the sensor such that the angle of the feature motion vectors are zero when the side-slip angle is zero, any change in angle is said to be the side-slip angle. The sensor was tested by mounting it to a gimbal that induced a known artificial side-slip angle to be measured in fixed increments. The testing surface consisted of a flat concrete surface.

The results showed that accurate measurements can be made with relatively low noise bands. The sensor was able to successfully sample up to 250Hz, which is comparable to current commercial sensors. Although no direct comparison can be made between the camera based sensor and the commercial sensor, a comparison can be made to previous results obtained that used the same experimental setup. The results proved that the camera-based sensor outperforms the commercial sensor in accuracy and with lower noise bands. Supplementary tests showed that the sensor could measure at 100km/h which is close to highway speeds and more than sufficient for off-road testing applications. This is also far better than required for most off-road testing on rough terrain. In conclusion, the camera based sensor successfully provides a cheaper alternative to current methods by using inexpensive, off-the-shelf cameras with dedicated software.

The investigation could be improved by first identifying the cause of the reduction of features tracked at higher speeds. This will possibly allow the system to measure at higher speeds as well as increase the accuracy at lower speeds. The camera should be rotated such that the feature motion vectors move across the width of the image, instead of the height. This increases the overlap available for the features. Another improvement would be to mount the sensor higher on the vehicle, increasing the distance between the camera and the ground. This would reduce the motion observed by the sensor, reducing any blurring on the image and will aid the small motion assumption made by the LK tracking algorithm. The same effect can be achieved by changing the focal length. This decreases the accuracy at low speeds. However, using a lens with a variable focal length that changed depending on the velocity the accuracy could be maintained. An independent vehicle velocity measurement can be supplied by calibrating the system as to determine the exact relationship between the actual vehicle velocity and

pixel displacement. The sensor should be tested on various surfaces such as snow and ice where it is inherently difficult to identify and track features. Lastly, the sensor should be tested on off-road like terrains that would induce vehicle motion such as roll, pitch and yaw to establish the effect of on the measurement.

RUT MEASUREMENTS

The sensor consisted of a stereo-vision rig which has two cameras pointing towards the scene with a fixed relative geometry. The camera-based rut depth sensor was tested on a preformed rut that had three different heights in the rut. The measurements were compared to manual measurements made on the rut. Three methods of determining the rut depth from the 3D coordinates were investigated. The first method simply averaged the height over a fixed, predefined area that represented the rut floor and the ground datums. This method proved to not only be the fastest between the methods, but also the most accurate when compared to the manual measurements. It had a mean sampling frequency of 7.7Hz and a mean error of 1.5mm. The second method attempted to locate the rut's location by finding the maximum points on the rut profile, which are the piles. This was achieved by fitting a polynomial on the rut profile over a small window. If the turning point of the polynomial is within the window, it was said that it represented the turning point on the rut profile. The window is incrementally shifted until the whole rut profile was covered. The turning points with the largest heights represented the pile peaks. Additional checks were added to ensure that the peaks were correctly identified. This method proved to be computationally expensive and had a mean sampling frequency of 2.6Hz. It also was deemed the least accurate out of all the methods with a mean error of 10.6mm. The last method also attempted to locate the rut's location by finding the maximum points on the rut profile. However, this method checked where the gradient of the rut profile was equal to zero to find turning points on the rut profile. The same process as the polynomial fit method was used to determine the rut depth from the turning points. This proved to be a better approach than the polynomial fit method and showed a reduction in computation time to locate the rut. The mean sampling frequency was 4.2Hz and had a mean error of 3.1mm.

The study can be improved by investigating how much the rut‘s location would move during dynamic manoeuvres. This might eliminate the need for automatically detecting the rut's location. If the sensor had other information available, such as side-slip angle and steering angle, the location of the rut

could easily be determined based on the geometry of the vehicle eliminating the need to automatically detect the location which will increase the sampling frequency. The study can be further improved by considering both the left and right ground datums and not only one side, as in this study. This would require a larger field of view which could be achieved by either increasing the height between the sensor and the ground or using wide-angle lenses. The automatic rut detection algorithms made the assumption that the rut has piles along its side. No investigation was conducted to test the performance in the absence of piles or when the rut depth is small and therefore should be investigated. The implementation of the polynomial detection algorithm needs to be critically evaluated to determine the cause for the offset in measurements in the validation testing. One common bottleneck in computation time between the algorithms was the disparity mapping. The use of GPU's would greatly reduce this computation time and increase the sampling frequency. All the algorithms could be multi-threaded to potentially increase throughput and further increase the sampling frequency. In order to achieve higher sampling frequencies, a combination of these methods could be used. If the gradient method is used to locate the rut, and the planes found by averaging the heights rather than fitting a plane through the points, the total computation time would be reduced, increasing sampling frequency. Often rut depth measurements are conducted on snow and ice for environmental impact studies. The sensor should be tested on various surfaces such as snow and ice where it is inherently difficult to correlate regions to determine the disparity.

BIBLIOGRAPHY

Anderson, A. B., Palazzo, A. J., Ayers, P. D., Fehmi, J. S., Shoop, S. and Sullivan, P. (2005). "Assessing the impacts of military vehicle traffic on natural areas. Introduction to the special issue and review of the relevant military vehicle impact literature". In: *Journal of Terramechanics* 42.3, pp. 143–158.

Andor (2014). *What is Rolling and Global Shutter?* [Online]. Available at: `http://www.andor.com/learning-academy/rolling-and-global-shutter-exposure-flexibility` [Accessed]: 07/06/2017.

Bakker, E., Nyborg, L. and Pacejka, H. (1987). "Tyre modelling for use in vehicle dynamics." In: *SAE Technical Paper*. Detroit, MI, USA: SAE International.

Barbarisi, O., Palmieri, G., Scala, S. and Glielmo, L. (2009). "LTV-MPC for yaw rate control and side slip control with dynamically constrained differential braking". In: *2009 European Control Conference (ECC)*, pp. 4810–4815.

Bay, H., Ess, A., Tuytelaars, T. and Gool, L. V. (2008). "Speeded-Up Robust Features (SURF)". In: *Computer Vision and Image Understanding* 110.3, pp. 346–359.

Bevly, D. M., Ryu, J. and Gerdes, J. C. (2006). "Integrating INS Sensors With GPS Measurements for Continuous Estimation of Vehicle Sideslip, Roll, and Tire Cornering Stiffness". In: *IEEE Transactions on Intelligent Transportation Systems* 7.4, pp. 483–493.

Botha, T. R., Els, P. S., Shoop, S. A., Becker, C. M. and Sopherb, A. (2016). "Three Dimensional Rut Profile Measurement in Snow and Mud". In: *Proceedings of the 8th ISTVS Americas Regional Conference*. ISTVS '16. Troy, MI, USA: IEEE Computer Society.

Botha, T. R. (2015). "Digital Image Correlation: Application in Vehicle Dynamics". PhD dissertation. University of Pretoria. [Online]. Available at: `http://hdl.handle.net/2263/45956`.

Botha, T. R., Els, P. S., Shoop, S. A., Becker, C. M. and Sopher, A. (2016). "Tire Slip and Slip Angle Measurement in Snow". In: *Proceedings of the 8th ISTVS Americas Regional Conference*. Troy, MI.

Botha, T. R. and Els, P. S. (2012). "Vehicle Sideslip Estimation Using Unscented Kalman Filter, AHRS and GPS". In: *ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. 651–659.

Botha, T. R. and Els, P. S. (2015). “Digital image correlation techniques for measuring tyre-road interface parameters: Part 1 - Side-slip angle measurement on rough terrain”. In: *Journal of Terramechanics* 61, pp. 87–100.

Botta, G. F., Tolon-Becerra, A., Tourn, M., Lastra-Bravo, X. and Rivero, D. (2012). “Agricultural traffic: Motion resistance and soil compaction in relation to tractor design and different soil conditions”. In: *Soil and Tillage Research* 120.Supplement C, pp. 92–98.

Bouguet, J. (2000). “Pyramidal implementation of the Lucas Kanade feature tracker”. In: *Intel Corporation, Microprocessor Research Labs*.

Chung, T. and Yi, K. (2006). “Design and Evaluation of Side Slip Angle-Based Vehicle Stability Control Scheme on a Virtual Test Track”. In: *IEEE Transactions on Control Systems Technology* 14.2, pp. 224–234.

Esmaeili, M. and Ivanovic, A. (2015). “Analytical and numerical modelling of non-driven disc on friction material”. In: *Computers and Geotechnics* 68, pp. 208–219.

Fischler, M. A. and Bolles, R. C. (1981). “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Communications of the ACM* 24.6, pp. 381–395.

Fountas, S., Paraforos, D., Cavalaris, C., Karamoutis, C., Gemtos, T. A., Abu-Khalaf, N. and Tagarakis, A. (2013). “A Five-point Penetrometer with GPS for Measuring Soil Compaction Variability”. In: *Comput. Electron. Agric.* 96, pp. 109–116.

Geiger, A., Lenz, P. and Urtasun, R. (2012). “Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gillespie, T. D. (1992). *Fundamentals of Vehicle Dynamics*. Warrendale: Society of Automotive Engineers, pp. 206–208.

Grewe, L. L. and Kak, A. C. (1994). “Handbook of Pattern Recognition and Image Processing (Vol. 2)”. In: Orlando, FL, USA: Academic Press Inc. Chap. Stereo Vision, pp. 239–317.

Grimson, W. E. L. (1981). “A computer implementation of a theory of human stereo vision”. In: *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, pp. 217–253.

Hac, A. and Simpson, M. D. (2000). “Estimation of Vehicle Side Slip Angle and Yaw Rate”. In: *SAE Technical Paper*. SAE International.

Halvorson, J. J., McCool, D. K., King, L. G. and Gatto, L. W. (2001). “Soil compaction and over-winter changes to tracked-vehicle ruts, Yakima Training Center, Washington”. In: *Journal of terramechanics* 38.3, pp. 133–151.

Hambleton, J. P. and Drescher, A. (2009). “Modeling wheel-induced rutting in soils: Rolling”. In: *Journal of Terramechanics* 46.2, pp. 35–47.

Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge University Press.

Hemmat, A., Yaghoubi-Taskoh, M., Masoumi, A. and Mosaddeghi, M. (2014). “Relationships between rut depth and soil mechanical properties in a calcareous soil with unstable structure”. In: *Biosystems Engineering* 118, pp. 147–155.

Hernandez-Juarez, D., Chacón, A., Espinosa, A., Vázquez, D., Moure, J. C. and López, A. M. (2016). “Embedded Real-time Stereo Estimation via Semi-global Matching on the GPU”. In: *Procedia Computer Science* 143 - 153.

Iagnemma, K., Kang, S., Brooks, C. and Dubowsky, S. (2003). “Multi-sensor terrain estimation for planetary rovers”. In: *Proceedings of the 8th international symposium on artificial intelligence, robotics, and automation in space. IEEE Press, New York*.

Inagaki, S., Kushiro, I. and Yamamoto, M. (1995). “Analysis on vehicle stability in critical cornering using phase-plane method”. In: *JSAE Review* 16.2, p. 216.

Intel Corporation (2016). *Intel NUC Kit NUC6i7KYK Features and Configurations*. [Accessed: 2016-09-10]. [Online]. Available at: `https://www.intel.com/content/www/us/en/nuc/nuc-kit-nuc6i7kyk-features-configurations.html` [Accessed]: 10/09/2016.

Kistler (2016). *Correvit©S-HR Sensors*. [Online]. Available at: `https://www.kistler.com/?type=669%5C&fid=67238` [Accessed]: 10/09/2016.

Konolige, K. (1997). “Small vision system: Hardware and implementation”. In: *Proceedings of the International Symposium of Robotics Research*, pp. 111–116.

Kurjenluoma, J., Alakukku, L. and Ahokas, J. (2009). “Rolling resistance and rut formation by implement tyres on tilled clay soil”. In: *Journal of Terramechanics* 46.6, pp. 267–275.

Laganiére, R. (2011). *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Pub Limited.

Leutenegger, S., Chli, M. and Siegwart, R. Y. (2011). “BRISK: Binary Robust Invariant Scalable Keypoints”. In: *Proceedings of the 2011 International Conference on Computer Vision*. ICCV ’11. Washington, DC, USA: IEEE Computer Society, pp. 2548–2555.

Lowe, D. G. (2004). “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal for Computer Vision* 60.2, pp. 91–110.

Lu, Q., Gentile, P., Tota, A., Sorniotti, A., Gruber, P., Costamagna, F. and Smet, J. D. (2016). "Enhancing vehicle cornering limit through sideslip and yaw rate control". In: *Mechanical Systems and Signal Processing* 75.Supplement C, pp. 455–472.

Lucas, B. D. and Kanade, T. (1981). "An Iterative Image Registration Technique with an Application to Stereo Vision (DARPA)". In: *Proceedings of the 1981 DARPA Image Understanding Workshop*, pp. 121–130.

Moser, R. A., Sube, H. J., Turner, J. L. and Zakelji, P. (2010). "3D Digital image correlation: Applications to tire testing". In: *Tire Science and Technology* 38.2, pp. 100–118.

Nishio, A., Tozu, K., Yamaguchi, H., Asano, K. and Amano, Y. (2001). "Development of Vehicle Stability Control System Based on Vehicle Sideslip Angle Estimation". In: *SAE Technical Paper*. SAE International.

OpenCV (2017). *OpenCV library*. [Online]. Available at: `https://opencv.org/about.html` [Accessed]: 01/09/2017.

Rajamani, R. (2005). *Vehicle Dynamics and Control*. Springer.

Rentschler, W. and Uffenkamp, V. (1992). "Digital Photogrammetry in analysis of crash tests". In: *SAE Technical Paper*. SAE International.

Rublee, E., Rabaud, V., Konolige, K. and Bradski, G. (2011). "ORB: An Efficient Alternative to SIFT or SURF". In: *Proceedings of the 2011 International Conference on Computer Vision*. ICCV '11. Washington, DC, USA: IEEE Computer Society, pp. 2564–2571.

Shi, J. and Tomasi, C. (1994). "Good features to track". In: *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600.

Shoop, S. A., Knuth, M. and Wieder, W. (2013). "Measuring vehicle impacts on snow roads". In: *Journal of Terramechanics* 50.1, pp. 63–71.

Slama, C. C., Theurer, C. and Hendrikson, S. W. (1980). *Manual of Photogrammetry*. 4th ed. Falls Church, VA: American Society Of Photogrammetry.

Soane, B. D. and van Ouwerkerk, C. (2013). *Soil Compaction in Crop Production*. Developments in Agricultural Engineering. Elsevier Science.

Sony (2017). *Pregius image sensor for industry*. [Online]. Available at: `http://www.sony-semicon.co.jp/producten/IS/sensor0/technology/pregius.html` [Accessed]: 30/09/2017.

Stewenius, H., Nister, D., Kahl, F. and Schaffalitzky, F. (June 2005). "A minimal solution for relative pose with unknown focal length". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 2, pp. 789–794.

Zhang, Z. (2000). “A Flexible New Technique for Camera Calibration”. In: *IEEE Transactions On Pattern Aanlysis And Machine Intelligence* 22.11, pp. 1330–1334.

APPENDIX A LENS AND CAMERA SPECIFICATIONS

A.1 CAMERA SPECIFICATIONS: POINT GREY FLEA3 (FL3-U3-13Y3M-C)

Specification	Value
Max Resolution	1280x1024
Frame Rate (at max resolution)	150 FPS
Megapixels	1.3 MP
Chroma	Mono
Sensor Name	On Semi VITA1300
Sensor Type	CMOS
Readout Method	Global Shutter
Sensor Format	1/2"
Pixel Size	4.8 μm
Lens mount	C-Mount
ADC	10-bit
Interface	USB 3.0
Power Requirements	5-24V via GPIO or 5V via USB3.0
Power Consumption	3W
Dimension	29mm x 29mm x 30 mm
Mass	41g

A.2 LENS SPECIFICATIONS: FUJINON DF6HA-1B

Specification	Value
Main sensor size	1/2"
Focal length	6mm
Lens mount	C-Mount
Iris control	Manual

Specification	**Value**
Aperture range	f/1.2 - f/16
Focusing range	0.1m - ∞
Angle-of-view	56 deg x 43 deg
Focus control	Manual
Minimum object distance	4"
Dimensions	29.5mm x 29.5mm x 36.7mm
Weight	55g

APPENDIX B ALGORITHM PARAMETERS

B.1 SIDE-SLIP ANGLE MEASUREMENTS

Parameter	**Value**	**Units**
Lucas-Kanade optical flow		
Quality Level	0.06	*%*
Distance between corners	10	*Pixels*
Number of features to identify	50	
Hessian block size	3x3	*Pixels*
Window search size (per pyramid level)	21x21	*Pixels*
Minimum Eigen Threshold	7	
Pixel displacement threshold	2	*Pixels*
RANSAC		
Sample iterations	100	
Error threshold	0.05	*deg*

B.2 RUT DEPTH MEASUREMENTS

Parameter	**Value**	Units
RANSAC threshold for Ground Plane fit	1	*mm*
RANSAC Iterations for Ground Plane fit	2000	
RANSAC threshold for Rut Plane fit	2	*mm*
RANSAC Iterations for Rut Plane fit	2000	
Stereo-vision		
SAD Window size	25x25	*Pixels*
Minimum Disparity	9	*Pixels*
Maximum Disparity	128	*Pixels*
Gradient rut detection		
Rolling average filter	20	

Parameter	**Value**	Units
Polynomial rut detection		
Polynomial window size	80	*Pixels*
RANSAC Iterations for model fit	160	
RANSAC threshold for model fit	1	*mm*