Smart Agricultural Vehicle by Integrating Motion Model with Machine Vision Data

（運動モデルとマシンビジョンのデータ融合による
スマート農用車両の開発）

2018

北海道大学 大学院農学院

環境資源学専攻 博士後期課程

Ospina Alarcon Ricardo

Smart Agricultural Vehicle by Integrating Motion Model with Machine Vision Data

BY

Ospina Alarcon Ricardo

Dissertation

Submitted to Department of Environmental Resources in the Graduate School of Agriculture
Hokkaido University, Sapporo, Japan, 060-8589 in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

2018

Table of Contents

Acknowledgements

First of all, I want to thank my wife Yasuko, who has done nothing but support me throughout this whole process. I want to thank her for her patience and the sacrifices she made in order to allow me to pursue this degree. Without a doubt, none of this would have been possible without her.

I would like to extend my sincere gratitude to Dr. Noboru Noguchi for his support during the development of this work. I also would like to express my thanks to Dr. Kazunobu Ishii and Dr. Hiroshi Okamoto for sharing their knowledge and ideas during my research. Their patience and guidance when the project proved to be the most difficult, allowed me to find my way and accomplish my goals. For that I am grateful.

I would like acknowledge the cooperation and contribution of all my colleagues and staff in the Laboratory of Vehicle Robotics. Deep appreciation is given to student affairs section of the Hokkaido University Faculty of Agriculture, for all their help concerning paperwork and general support. Also, I would like to thank the support obtained through the JASSO scholarship.

I would like to acknowledge my family for the tremendous amount of support that they have given me throughout my life. Thank you for the sacrifices you have made to ensure that I always had what I needed and the guidance you have given to me.

Lastly, this work is dedicated to my daughter Nanako; only God knows how much I love you baby.

List of Figures

List of Tables

Notations

Variable	Description	Units
a	Distance between frontal axle and center of gravity	m
α_f	Frontal axle tire slip angle	deg
α_r	Rear axle tire slip angle	deg
b	Distance between rear axle and center of gravity	m
β	Body sideslip angle	deg
CI	Soil cone index	kPa
$C_{\alpha f}^{axle}$	Frontal axle cornering stiffness	N/deg
$C_{\alpha r}^{axle}$	Rear axle cornering stiffness	N/deg
$C_{\alpha f}^{tire}$	Frontal tire cornering stiffness	N/deg
$C_{\alpha r}^{tire}$	Rear tire cornering stiffness	N/deg
δ	Front wheel steering angle	deg
Fy	Vehicle's lateral force at the center of gravity	N
Fyf	Frontal axle lateral force	N
Fyr	Rear axle lateral force	N
Fz	Vehicle's normal load	N
φ_{GPS}	direction of travel given by the GPS	deg
φ_{FOG}	heading given by the FOG	deg
Izz	Polar moment of inertia	kg*m^2
m	Vehicle mass	kg
ρ	Turning radius	m
V	Vehicle velocity at the center of gravity	m/s
Vx	Longitudinal velocity	m/s
Vy	Lateral velocity	m/s
$\dot{V}_x$	Longitudinal acceleration	m/s^2
$\dot{V}_y$	Lateral acceleration	m/s^2
ω	Yaw rate	deg/s

Acronyms and abbreviations

Legend	Description
AHRS	Attitude Heading Reference System
ARP	Active Rollover Protection
CG	Center of gravity
DGPS	Differential Global Positioning System
ECU	Electronic Control Unit
EKF	Extended Kalman Filter
ESP	Electronic Stability Program
FAO	Food and Agriculture Organization
FOG	Fiber Optic Gyroscope
GIS	Geographical Information Systems
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
IMU	Inertial Measurement Unit
IRNSS	Indian Regional Navigational Satellite System
ISO	International Organization for Standardization
LSM	Least Squares Method
MAFF	Ministry of Agriculture, Forestry and Fisheries
NDGPS	Nationwide Differential GPS System
NMEA	National Marine Electronics Association
PA	Precision Agriculture
PC	Personal Computer
PID	Proportion-Integration-Differentiation
QZSS	Quasi-Zenith Satellite System
RAM	Random-Access Memory
RMS	Root Mean Square
RTK-GPS	Real Time Kinematic -Global Positioning System
SUV	Sports Utility Vehicle
UAV	Unmanned Aerial Vehicles
USB	Universal Serial Bus

UTM	Universal Transverse Mercator
VRS	Virtual Reference Stations
WADGPS	Wide-Area DGPS
Wi-Fi	Fixed Wireless IEEE 802.11b
WLAN	Wireless Local Area Network

Chapter 1 Introduction

This chapter presents concepts concerning agricultural automation. It briefly describes the autonomous navigation systems for agricultural vehicles as a solution for current agricultural problems in Japan. It also introduces the motivation of this work and the research objectives pursued.

1.1 Research background

Agriculture is mankind's oldest economic activity, providing necessary food and fuel for our survival. With the global population expected to keep increasing in the future, agricultural production must also increase in order to meet the new demands for food and bioenergy. In Japanese society, aging and population decline have been progressing; and this tendency is more significantly seen in rural areas than in urban areas (Hashimoto et al., 2001).

In order to protect food production in Japan, encouraging the development of technologies in the field of agriculture automation such as autonomous guidance systems for agricultural vehicles seems to be an effective strategy to deal with the dwindling farming labor force, in addition to increase production efficiency and safer operation (Kondo et al., 2011).

1.1.1 Agriculture automation

Trends in the evolution and development of agricultural field machinery are often shaped by the technological development in other sectors of the world's economy such as defense and transportation. Robots have played a fundamental role in increasing the efficiency and reducing the cost of industrial production and products. In the past twenty years, a similar trend has started to take place in agriculture, with GPS- and vision-based self-guided tractors and harvesters already being available commercially. Robotics and automation can play a significant role in society by meeting the future agricultural production needs. To cope with the aging population and labor force reduction problems in agriculture, the agriculture automation field is devoted to increase productivity, reduce production cost and improve the quality of agricultural products.

Looking towards the future to a point in time when humans are removed from field machinery, there are several emerging technologies that will be essential for autonomous operation. In some cases infrastructure development such as densification of Real-Time Kinematic (RTK) GPS networks to generate Virtual Reference Stations (VRS) correction data along with the development of Internet connectivity via Wi-Fi and WLAN to support data transfer. What follows is a brief overview of the status of many of the allied technologies that are essential for totally agriculture automation.

Space-Based Positioning Systems: Advancements in sensing, communication and control technologies coupled with Global Navigation Satellite Systems (GNSS) and Geographical Information Systems (GIS) are aiding the progression of agricultural machines from the simple, mechanical machines of yesterday to the intelligent, autonomous vehicles of the future.

Wireless Communications: For large scale high-tech agricultural operations, establishing vehicle to vehicle and vehicle to office communication is becoming imperative to manage the logistics of the tasks and to ensure the safety of the machines working in the field. The capability to wirelessly transfer data can help monitor the working statuses of these machines and allow dynamic reallocation of tasks in the event of malfunctions. Point to point and point to multi-point communication can specifically be used for leader-follower systems.

On-Vehicle Communications: With the introduction of microcontrollers to agricultural filed machinery it was not long until equipment designers realized the need to share and manage information between controllers. Following the lead of the truck, bus and automotive industries, equipment designers began looking for bus configurations and data structures to support continuing machinery development. Quickly, most designers realized the need for standardization to facilitate interoperability and interchangeability the industry came to grips with for hitching (ISO 730, 2009) and hydraulic systems (ISO 5675, 2008).

Data Structures: While on-vehicle communication has relatively well defined data structures (ISO 11783), standards for transfer of data between the farm office and field machinery continue to evolve. The latter is being driven for the most part by software developers who recognize the need to reconcile data transfer from a farm office to a field machinery and back again. Today, the need to reconcile data is being driven by map-based application. “Prescription maps” direct where and how inputs will be

applied to crop production systems. Data regarding input metering and placement is further complicated by the nature of field equipment apply inputs. Crop production managers and suppliers have multifaceted data transfer needs that range from moving prescription maps form the farm office to field equipment and then returning plans field operations verification files along sensor data for summarizing crop health and performance to the field office.

By implementing these technologies, agriculture automation has achieved a rapid development in the execution of all the agricultural tasks. More recently, farmers have started feasible tests with autonomous systems that automate or augment operations such as pruning, thinning, and harvesting, as well as mowing, spraying, and weed removal.

Advances in sensors and control systems allow for optimal resource and integrated pest and disease management. This is just the beginning of what will be a revolution in the way that food is grown, tended, and harvested.

1.1.2 Automatic navigation systems for agricultural vehicles

Systems designed to accomplish automated guidance on agricultural vehicles can be seen back as far as the 1920s when furrows were used to guide tractors across fields with reduced effort from the operator. Since that time, as technology has improved, automated guidance has evolved from mechanical sensing to electronic sensors, machine vision, and global positioning systems (GPS) to successfully navigate equipment across the field (Reid et al., 2000). In most cases, operators utilize automatic guidance to follow parallel paths through the field. At the beginning of field operations, an A-B line is input into the control console, and the GPS coordinates are stored. As the operator continues to cover the field, the automatic guidance system can be engaged and the equipment will attempt to follow parallel paths to cover the field based on steering sensor feedback and GPS data. Many systems also provide the ability to follow curved paths which are input in much the same way.

Agricultural vehicles, such as tractors, are designed to provide some drivability over fields and off-road surfaces. The counterpart is their propensity to roll over the ground. As a result, the development of on-board systems preventing agricultural machinery from rollover situations is encouraged. Several solutions (Anderson et al., 2005) have already been proposed for road vehicles: steering and braking

control or Electronic Stability Program (ESP) systems are some examples. In modern agriculture, tractors and other agricultural machinery are essential. The Laboratory of Vehicle Robotics (VeBots) of Hokkaido University School of Agriculture has succeeded conducting researches on unmanned tractors and developing new technologies. Sensors are installed on a tractor, and its position is measured by satellite (GPS). A computer commands its movement, replacing manual operation and thus giving birth to robot tractors.

Two basic types of automated guidance systems are typically used today by producers. The first system consists of a steering actuator which is mounted to the tractor's steering wheel. The second system is integrated into the tractor's steering system and utilizes a control valve to actuate the hydraulic steering cylinder directly. The overall accuracy of these systems relies heavily on the type of GPS technology used (RTK GPS provides the highest accuracy) as well as proper installation and setup. Ultimately, these systems benefit producers by reducing operator effort and pass-to-pass overlap during field applications. In the field of agricultural vehicles, the implementation of advanced sensor technologies and GPS has given as a result autonomous agricultural machines with high navigation accuracy. Since the early 1990s, the GPS receivers have been widely used as space-based satellite navigation system to provide longitude and latitude position coordinates anywhere on the earth.

For some years now, governments, researchers and farmers have become increasingly conscious of the important role Precision Agriculture (PA) will play in the near future (Burgos-Artizzu et al., 2011). With the current projections of expected world population growth and the subsequent decrease of available land and natural resources, there will be a pressing need for a cheaper, more efficient and environmentally friendly agriculture (Srinivasan, 2006). PA seeks to avoid applying the same management practices to a crop regardless of site conditions and may be used to improve field management from several perspectives; for example, it can help to minimize the wastage of pesticides required for the effective control of weeds, diseases and pests and to ensure that crops receive adequate nutrients, leading to more efficient and greener agriculture (Kropff et al., 1997; Earl et al., 1996).

1.1.3 Terramechanics: Applications to Terrain-Vehicle Systems

Information on the state of an agricultural vehicle such as its location, and tire parameters like the cornering stiffness can be estimated using a Global Positioning System (GPS). Methods using a GPS

and an Inertial Measurement Unit (IMU) integration have been developed (Bevly et al., 2006) to predict critical tire parameters in the limits of handling. Previously GPS/IMU solutions (Sienel, 1997) have been shown to estimate a vehicle's sideslip angle and tire slip angle. Using these estimates, the tire cornering stiffness can be estimated for the linear region of the tire (Bouton et al., 2007). In this regard, some researches involve the study of a wheeled vehicle running on various surfaces and the soil-wheel interaction (Ospina and Noguchi, 2018). This area of engineering is known as *Terramechanics*. The tire's operating range and the soil properties are topics of particular interest.

The tire's operating range is divided into three regions: linear, transitional and sliding. A linear tire model can be used to predict the properties in the linear region, but generally cannot be employed in transitional and sliding regions since they are nonlinear (Baffet et al., 2006).

Linear tire models that consider only the linear region of the tire's operating range have been successfully implemented in road vehicles. However, since road vehicles are supposed to move on high grip ground, such systems consider only pseudo-sliding phenomenon with constant parameters (Pepy et al., 2006). Most of active devices focused on vehicle stability concerns road cars and cannot be applied satisfactorily in an off-road context, as in path planning of autonomous agricultural vehicles like tractors and utility vehicles (Noguchi et al., 1998); since the variability and nonlinearities of the tire/ground contact are often neglected.

For off road vehicles, the nonlinearities of the interaction between the tires and the road surface gives as result a hysteresis loop in the linear region of the tire's operating range. For this reason, it is very complicated to model the lateral tire forces due to the nonlinear dependence of the lateral forces on several parameters, such as longitudinal slip, sideslip angle, normal load, camber angle, tire pressure, wear, and road surface characteristics (Koo et al., 2004). Even thought, the nonlinearities of the interaction between the tires and the road surface should not be ignored for off road vehicles. Therefore, there is a necessity of a precise vehicle model that does not depend on the parameters mentioned above; namely a "non-parametric" vehicle model, that fills the gaps existing in current models applied to autonomous vehicle development.

Some works consider the nonlinear behavior of the tire by estimating the lateral force and tire slip angle over the tire's entire operating range (linear, transitional and sliding) obtaining a non-parametric vehicle model using mathematical expressions that account for the sliding nonlinear behavior in the

vehicle dynamics (Ospina and Noguchi, 2016). This thesis aims to use this tire information in order to improve the navigation model of an automatic steering controller based on machine vision applied to an agricultural test vehicle.

1.1.4 Machine vision systems for agricultural vehicles

Machine vision systems onboard robots have become increasingly important in Precision Agriculture (PA) in order to fully automate some in-field agricultural tasks; like automatic navigation of an agricultural vehicle. In addition, machine vision systems in precision agriculture are also used to gather data from the field in an automated manner at minimal cost; like crop mapping. Several researches (Pajares et al., 2016) have achieved good results performing either crop mapping or crop navigation separately.

Research work in this area is difficult to classify and compare due to the variations among different crops species and to the different approaches taken to collect field data (Thorp and Tian, 2004). However, almost all existing crop row detection methods process the image in two steps: 1) segmentation of vegetation against the background (soil and/or harvest residues) and 2) detection of the vegetation pixels.

The procedures for the segmentation of vegetation usually assume that all pixels belonging to vegetation can be easily extracted by some combination of the color planes on the RGB model (Woebbecke et al., 1995; Andreasen et al., 1997). Other approaches propose the use of the HIS color model combined with classification methods such as Bayes networks and clustering (Lee et al., 1996, Zheng et al., 2009). Segmentation can also be performed by selecting texture features based on their similarities with previous models encountered, stored in a database (Bosch et al., 2007). Moreover, segmentation can be performed by combining different cameras, such as conventional and NIR cameras (Gerhards and Christensen, 2003).

In addition, weed detection by computer vision methods is also usually performed by combining information on differences in color, position, shape, texture, size or spectrum of weeds and crop. The use of only one or many of these characteristics depends on the way the photographic images are taken, the crop type, and the weed species involved. Some works present statistical studies of the features

involved (Van Evert et al., 2006). Others choose to distinguish between species by their different spectra, combining the information obtained from a conventional camera with a NIR camera, (Gerhards and Oebel, 2006). In lettuce, plants and weeds can be clearly distinguished by differences in their sizes and position. This is similar to the case of cauliflowers, where weeds can be located by their position and some shape characteristics (Onyango and Marchant, 2003), or in carrots and cabbages, where weeds have distinct differences in color and size (Hemming and Rath, 2001). Other studies make use of classification or feature extraction methods, including as many characteristics as necessary. This is used, for example, in Bayes networks (Granitto et al., 2005), or neuronal networks (Burks et al., 2005). Additionally, some studies process the images in the frequency domain (Vioix et al., 2002) or even using Fuzzy Logic.

Crop row location in real time is often an important goal in the autonomous guidance of agricultural vehicles (Gottschalk et al., 2008). In this context, the crop rows are roughly approximated by lines. Some other works use the Hough transform (Gonzalez and Woods, 2003), to fully locate the crop rows. The drawback of this approach is the high computational complexity of the Hough transform, which makes it unsuitable for applications in which there is a need to process images in real-time, i.e. at 25 fps (frames per second), the standard video camera frame rate. Finally, some other studies deal with simpler images, taken closer to the ground and in such a way that perspective is eliminated, so that crop rows can be more easily located and the processing adapted to real-time (Sogaard and Olsen, 2003).

In the other hand, accurate maps showing both weed location and crops density have numerous uses including monitoring the effectiveness of weed management strategies, understanding weed population dynamics and verifying model predictions. In particular, they can be the data source for sprayers, which can determine their location using a GPS receiver and apply treatments where data recommends it. These spatial information systems have the potential to allow farmers to fine-tune the locations and rates of herbicide application, thereby achieving sustainability and reducing treatment costs.

Data gathering using a tractor or vehicle as a mobile platform requires only one operator and enables continuous field sampling. In continuous sampling, data are collected over the entire sample area, whereas with discrete sampling, data are collected only from pre-defined points throughout an area. Interpolation methods are then used to estimate the crop densities in the intervening areas. Continuous data can provide a qualitative description of abundance (i.e., presence or absence, or zero, low,

medium, or high) rather than the quantitative plant counts usually generated from discrete sampling. Moreover, acquiring video from a mobile platform may become a good opportunity to obtain accurate weed and crop maps, and also crop row location in real time has often been an important goal in the autonomous guidance of agricultural vehicles (Billingsley and Schoenfisch, 1997), which increments the advantages of the ground level approach. Mounting cameras on top of tractors or mobile platforms presents problems because the roughness of the terrain transfers to the camera mounting system and causes it to acquire images that are difficult to process, even to the human eye. Image sequence stabilization is the process of removing the effects of this unwanted motion from an input video sequence. It is a key pre-processing step in any serious application of computer vision, especially when images are acquired from a mobile platform. Based on the particular roughness of the terrain, motion in Precision Agriculture video sequences can include vibration, sway, roll and pitch.

In conclusion, machine vision has been widely utilized for crop mapping because it offers the ability to instantly assess the target crops, and it does so in a non-destructive manner. Also, significant research has already been carried out in the machine vision-based guidance systems and field sensing systems since field images acquired from vehicle-mounted cameras provided baseline data for both navigation and mapping processes.

However, current machine vision systems methods have limitations trying to perform both navigation and mapping processes at the same time. Methods for automatic navigation of an agricultural vehicle based in crop row detection (Romeo et al., 2013) focus into detecting crop rows as accurate as possible; typically shooting the central three crop rows at the same time. Detecting three crop rows might provide enough accuracy for navigation; however, only three crop rows do not seem to contain enough information to build a field map of practical use. On the other hand, a different approach (Slaughter et al., 2008) is to use several cameras on zenithal position, each camera shooting one crop row at the time giving as a result a high quality map containing several crop rows. This mapping method can be used in practical applications like spraying, but navigation from crop row detection becomes a more complex problem. In some cases (Bengochea-Guevara et al., 2016), precise detection of the central crop row might provide enough accuracy for navigation; but this method requires additional image processing steps and is not suitable to build a crop map.

1.2 Research objectives

This thesis' purpose is to enhance the navigation performance of an agricultural vehicle by integrating a nonlinear vehicle motion model with a unique machine vision system. To achieve this purpose, this thesis has two objectives. The first one is to use previous researches results concerning the vehicle's tire dynamic properties in order to improve the motion model of an automatic steering controller system applied to a test vehicle. The second one is to integrate this improved navigation model with a machine vision method capable of provide automatic navigation with increased accuracy. To integrate the best aspects of the vehicle motion model estimations with the machine vision measurements, data fusion technique was used.

To fulfill the first objective of this thesis, the measurements of the tire's lateral forces as a function of the tire's slip angle of a test vehicle traveling on a concrete surface are reported. A description about how these tire parameters can be used to improve the motion model of the test vehicle is given as well. To fulfill the second objective of this thesis, an explanation of the new machine vision method proposed is given in further chapters.

The machine vision method proposed is capable of mapping several crop rows while performing simultaneous crop row detection with high accuracy. As aforementioned, the method is intended for use in automatic guidance systems of agricultural machinery. To achieve this goal, this research implemented a new camera developed by Fujifilm Corporation. This camera can shoot high definition wide-angle and telephoto images simultaneously. The camera was mounted in the top of a test vehicle, focused on the field surface from an inclined angle in order to obtain wide-angle images that cover up to eleven crop rows. At the same time, the camera provides accurate detection of the central crop row using telephoto images. The test vehicle was also equipped with an RTK-GPS and an FOG. An on-board computer process the data from the camera, RTK-GPS and FOG. For accurate crop row detection this research used an image analysis method without segmentation instead of the Hough transformation method (Ji and Qi, 2011), in order to reduce the computational burden of the image processing software.

Results show that the new machine vision method introduced in this research displays increased accuracy and noise reduction for crop row detection used in automatic navigation. In addition, the

resulting map covers up to eleven crop rows, compared to other mapping methods that cover up to five crop rows. These results imply that the method is ideal for practical applications like spraying, avoiding to travel additional paths along the field in order to build a crop map; while providing crop row detection with increased accuracy. Results also show the benefit of determining the test vehicle's tire dynamic properties in order to improve the navigation model.

1.3 Organization of thesis

This thesis presents a navigation method applied to an agricultural vehicle based on an improved navigation model and a new machine vision method consisting in simultaneous mapping and crop row detection by fusing data from Wide-angle and Telephoto images. This thesis is organized into five chapters.

In chapter 1, the research background and the importance of autonomous navigation systems for precision agriculture has been introduced. The objectives of this research were also stated.

Chapter 2 presents the research platform used during the experiments. The test vehicle and the sensors implemented in the experiments are described.

Chapter 3 describes the vehicle modeling strategy. An explanation of the tire dynamic parameters employed in this study and how they improve the vehicle navigation model is given.

Chapter 4 describes the image processing algorithms employed in this study. It includes a description of the programming language and the libraries used.

In chapter 5, experimental results for navigation are discussed. An approach to combine the vehicle navigation model and the machine vision method is described.

In chapter 6, experimental results for crop row mapping are discussed. It includes the map accuracy calculation.

Chapter 7 summarizes the achievement of this study and proposes future work.

Chapter 2 Research platform

This chapter describes the equipment used during this research. The research platform implemented in the Kubota MD77 tractor used for the field experiments is introduced. Both the sensors and the test vehicle's technical specifications are included.

2.1 Kubota MD77

The Kubota MD series is a family of utility tractors produced by the Kubota Corporation. These mid-size tractors provide versatility in different agricultural tasks (lift, move, push, pull, and grade) with an environmentally responsible diesel engine. Among all the models of the MD series (77, 87, 97, 107, 117) the MD77 was used.

2.1.1 Vehicle overview

The Kubota MD77 most important specifications are list on Table 2.1. The PTO specifications have been omitted. For a copy of the vehicle's data sheet, please refer to Appendix A.

Table 2.1. Kubota MD77 tractor specifications.

Part	Description	
Engine	Type	Four Cylinder Four Stroke Water Cooled Diesel
	Power output	56.6 kW / 2400 rpm
	Total stroke volume	4.329 l
	Chamber configuration	Direct injection type
	Supercharger	No
Travel Speeds	Typical travel speed-forward	1.2 ~ 29.1 km/h
	Typical travel speed-reverse	1.2 ~29.4 km/h
Transmission gear	Main clutch-type	Dry type, single plate
	Main speed	8 gears
	Sub-speed	2 gears
	Differential lock device	Yes
Dimensions	Overall Width	1.96 m

	Overall Length	3.88 m
	Overall Height	2.60 m
	Front Tread Centers	1.43 m
	Rear Tread Centers	1.52 m
	Wheelbase	2.30 m
	Vehicle Weight	3265 kg
Tires	Front 9.5-24-6 PR	
	Rear 16.9-30-6 PR	

The test vehicle was equipped with an on-board computer that logs the data from all the sensors. Fig. 2.1 shows the vehicle and the sensors equipped for the experiments.

Fig. 2.1. Experimental vehicle and equipped sensors.

Both the GPS and the FOG have a direct serial port connection to the on-board computer. The potentiometer was connected to a microcontroller in order to process its analog signal. The microcontroller communicates with the on-board computer by serial port connection. The speed of all serial connections was 115200 bps. The 2-in-1 camera was connected using an HD-SDI to USB 3.0

converter (The Imaging Source, DFG/HDSDI) providing 30 frames per second at a resolution of 1280 x 480 pixels. However, since the GPS NMEA data frames, the FOG data frames, the microcontroller data frames and the 2-in-1 camera video frames have different lengths, all the data was synchronized using the computer's time stamp.

The result was a measurement update rate of 10 Hz for all the sensors, in order to make the FOG, the 2-in-1 camera and the potentiometer measurements coincide with the GPS measurement. Next sections provide a brief description of the sensors implemented in the platform shown in Fig.2.1.

2.2. GPS

The U.S. Global Positioning System (GPS) is maintained by the U.S. government and has been in operation since the late 1970s. The benefits of GPS, specifically in the agricultural industry, have been well documented as they have progressed from point location mapping (soil sampling or yield monitoring) to real-time equipment control (auto-steer or map-based automatic section control). To increase the accuracy of the existing GPS network, additional technologies have been developed by both public and private institutions. The Nationwide Differential GPS System (NDGPS) was developed for use in the U.S. and included beacons maintained by the U.S. Coast Guard and the Department of Transportation. There are other positioning systems, like the Global Navigation Satellite System (GLONASS) which is a Russian-operated satellite network that was developed in the late 1970s and was extended to non-military use in 2007. GLONASS is comparable to the U.S. GPS system and was created to provide real-time positioning data to compatible receivers. The GLONASS system is continually upgraded as existing satellites exceed their service life and new series replace them. The GLONASS-M series is currently in operation, and the GLONASS-K1 series became operational since 2011 (FSA-IAC, 2010).

Also, the Galileo global navigation satellite system is currently being developed by the European Union (EU) to provide a separate network of satellites from the Russian and U.S. systems that are now in use. The Galileo system has been developed by the European Space Agency primarily to provide real-time positioning data for civilian use and was designed to be compatible with the Russian and U.S. systems. Two experimental satellites have been successfully launched and four additional satellites have been launched in 2011 to validate system operation (ESA, 2010). As of December 2017, 22 of the planned 30 active satellites are in orbit. The next launch, which will bring the system to operational

completion, is planned in mid-2018.Galileo started offering Early Operational Capability (EOC) on 15 December 2016, providing initial services with a weak signal, and is expected to reach Full Operational Capability (FOC) in 2019. Table 2.2 lists a comparison of the GNSS systems.

Table 2.2. Comparison of GNSS systems.

	Political entity	**Number of satellites**	**Orbital height**	**Period**
GPS	United States	At least 24satellites	20,180 km	11.97 hours
GLONASS	Russian Federation	31 (24 operational,1 in preparation, 2 on maintenance, 3 reserve, 1 on tests)	19,130 km	11.26 hours
Compass	China	5 geostationary earth orbit (GEO) satellites, 30 medium earth orbit (MEO) satellites	21,150 km	12.63 hours
Galileo	European Union	22 satellites in orbit; 30 operational satellites budgeted	23,220 km	14.08 hours
IRNSS	India	7 geostationary earth orbit (GEO) satellites	36,000 km	N/A
QZSS	Japan	1 satellite in orbit, 2 satellite in budget	32,618km	23.96hours

The accuracy of differential global position systems (DGPS) degrade with increasing distance to the reference station. For DGPS systems, an inter-receiver distance of a few hundred kilometers will yield a sub-meter level accuracy, whereas for Real Time Kinetic (RTK) systems a centimeter level accuracy is obtained for distances of less than 10 km. To service larger areas without compromising on the accuracy, several reference stations have to be deployed. Instead of increasing the number of real reference stations, Virtual Reference Stations (VRS) are created from the observations of the closest reference stations. The locations of the VRS can be selected freely but should not exceed a few kilometers from the rover stations. Typically one VRS is computed for a local area and working day.

The observations from the real reference stations are used to generate models of the distance dependent biases. Individual corrections for the network of VRS are predicted from the model parameters and the

user's position. This kind of network applied to DGPS and RTK systems is known as wide-area DGPS (WADGPS) and network RTK respectively. An example of a commercially available network RTK is Trimble's VRS that provides high-accuracy RTK positioning for wider areas. A typical VRS network set up consists of GNSS hardware, communications interfacing and, modeling and networking software. Most of the existing network RTK systems have been installed in the densely populated areas of central Europe.

RTK-GPS:

The RTK-GPS (Topcon, Legacy-E) provides the position, direction of travel and speed of the vehicle. The low latency configuration (update rate: 10 Hz, latency: 0.02 seconds, data link: 9600 Baud) was chosen for the RTK mode. This configuration provides a horizontal position accuracy of 5mm + 0.5 ppm, a vertical position accuracy of 3 cm + 2 ppm and a speed accuracy of 0.16 km/h. The RTK correction signal was obtained using a Virtual Reference System via an Internet Service Provider connected to the on-board computer that logs the data from the GPS receiver.

Real-Time Kinematic (RTK) positioning is based on at least two GPS receivers—a reference receiver and one or more rover receivers. The reference receiver takes measurements from satellites in view and then broadcasts them, with its location, to the rover receiver(s). The rover receiver also collects measurements to the satellites in view and processes them with the reference station data. The rover then estimates its location relative to the reference. Typically, reference and rover receivers take measurements at regular 1 second epochs (events in time) and produce position solutions at the same rate.

The key to achieving centimeter-level positioning accuracy with RTK is the use of the GPS carrier phase signals. Carrier phase measurements are like precise tape measures from the reference and rover antennas to the satellites. In the Legacy-E Series, carrier phase measurements are made with millimeter-precision. Although carrier phase measurements are highly precise, they contain an unknown bias, termed the integer cycle ambiguity, or phase ambiguity. The Legacy-E Series rover has to resolve, or initialize, the carrier phase ambiguities at power-up and every time the satellite signals are interrupted.

The Legacy-E Series can automatically initialize the carrier phase ambiguities as long as at least 5 common satellites are being tracked at reference and rover sites. Automatic initialization is sometimes

termed On-The-Fly (OTF) or On-The-Move, to reflect that no restriction is placed on the motion of the rover receiver throughout the initialization process.

The Legacy-E Series uses L1 and L2 carrier measurements plus precise code range measurements to the satellites to automatically initialize the ambiguities. The initialization process takes between 25 seconds to a few minutes. While the receiver is initializing the ambiguities it generates a float solution with meter-level accuracy. When the initialization process is complete, the solution mode switches from float to fix, and the precision changes from meter-level to centimeter-level accuracy. As long as at least 4 common satellites are continuously tracked after a successful initialization, the ambiguity initialization process does not have to be repeated. The number of position fixes delivered by an RTK system per second also defines how closely the trajectory of the rover can be represented and the ease with which position navigation can be accomplished.

The number of RTK position fixes generated per second defines the update rate. Update rate is quoted in Hertz. For the Legacy-E receiver, the maximum update rate is 20 Hertz. Solution latency refers to the lag in time between when the position was valid and when it was displayed. For precise navigation, it is important to have prompt position estimates, not values from 2 seconds ago. Solution latency is particularly important when guiding a moving vehicle. For example, a vehicle traveling at 25 kilometers/hour, moves approximately 7 meters per second. Thus, to navigate to within 1 meter, the solution latency must be less than 1/7 (= 0.14) seconds. For a copy of the GPS receiver data sheet, please refer to Appendix B.

2.3 FOG

A Fiber Optic Gyroscope (FOG) senses changes in orientation using the Sagnac effect, thus performing the function of a mechanical gyroscope. However its principle of operation is instead based on the interference of light which has passed through a coil of optical fiber, which can be as long as 5 km. Two beams from a laser are injected into the same fiber but in opposite directions. Due to the Sagnac effect, the beam travelling against the rotation experiences a slightly shorter path delay than the other beam. The resulting differential phase shift is measured through interferometry, thus translating one component of the angular velocity into a shift of the interference pattern which is measured photometrically.

Beam splitting optics launches light from a laser diode into two waves propagating in the clockwise and anticlockwise directions through a coil consisting of many turns of optical fiber. The strength of the Sagnac effect is dependent on the effective area of the closed optical path: this is not simply the geometric area of the loop but is enhanced by the number of turns in the coil. The FOG was first proposed by Vali and Shorthill (Vali and Shorthill, 1976). Development of both the passive interferometer type of FOG, or IFOG, and a newer concept, the passive ring resonator FOG, or RFOG, is proceeding in many companies and establishments worldwide (Lefèvre, 1993).

The FOG (Japan Aviation Electronics Industry Ltd., JCS7402-A) provides the vehicle's angular rates (accuracy $\leq \pm 5$ deg. /s), attitude (accuracy $\leq \pm 0.15$ deg.) and lateral acceleration (accuracy $\leq \pm 1.5$ m/s^2) readings. The advantage of using an FOG ever an IMU is that the drift effects in the FOG are smaller and therefore can be neglected. Both the FOG and the GPS antenna are placed parallel to the test vehicle's center of gravity. Fig. 2.2 shows the coordinate system of the JCS7402-A.

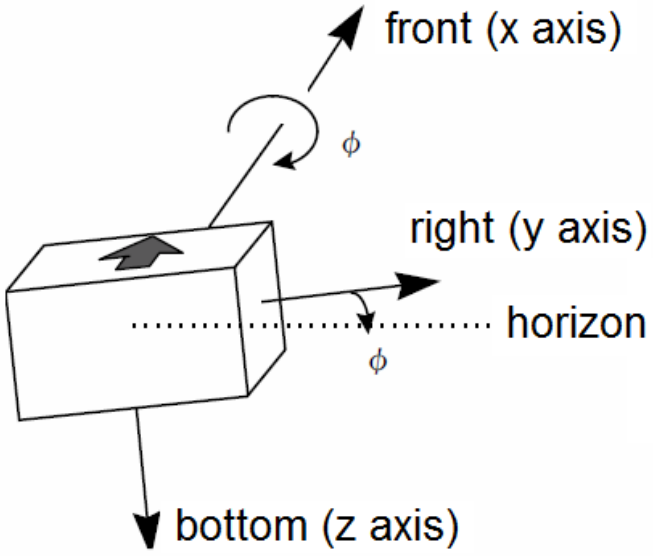


Fig. 2.2. JCS7402-A coordinate system.

Table 2.3 lists the most important specifications of the FOG. For a copy of the FOG data sheet, please refer to Appendix C.

Table 2.3. JCS7402-A FOG specifications.

Specification	Value
Attitude & Heading	
Range: Heading, Roll:	±180 deg.
Range: Pitch :	±90 deg.
Resolution:	≤ 0.1 deg.
Accuracy:	≤ ± 0.15 deg.
Baud rate:	19.2 kbps ~ 115.2 kbps.
Output Rate :	50 Hz ~ 200 Hz
Azimuth	
Range:	±180 deg.
Resolution:	≤ 0.1 deg.
Accuracy:	≤ ± (1% of input+(2 deg./min))
Acceleration	
Range:	±19.6 m/s^2
Resolution:	≤ 0.1 m/s^2
Accuracy:	≤ ± 1.5 m/s^2
Rate	
Range:	≤ ± 100 $deg./s^2$
Resolution:	≤ 0.1 $deg./s^2$
Accuracy:	≤ ± 5 $deg./s^2$
Input Voltage:	20 V ~ 30 V
Current Draw:	≤ 1 A
Digital Interface:	Serial TTL, RS-232

The FOG provides extremely precise rotational rate information, in part because of its lack of cross-axis sensitivity to vibration, acceleration, and shock. Unlike the classic spinning-mass gyroscope, the FOG has no moving parts and doesn't rely on inertial resistance to movement. Hence, this is one alternative to the mechanical gyroscope. Because of their intrinsic reliability, FOGs are used for high performance space applications. The FOG typically shows a higher resolution than a ring laser gyroscope, but suffered from greater drift and worse scale factor performance until the end of the 1990s. However, among other disadvantages, FOGs requires calibration (determining which indication

corresponds to zero angular velocity) while ring laser gyroscopes do not (zero beat frequency always means zero angular velocity). Also, FOG are somewhat sensitive to vibrations.

2.4 Potentiometer

A potentiometer is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider. If only two terminals are used, one end and the wiper, it acts as a variable resistor or rheostat.

A potentiometer working as a measuring instrument is essentially a voltage divider used for measuring electric potential (voltage); the component is an implementation of the same principle, hence its name. Potentiometers are commonly used to control electrical devices such as volume controls on audio equipment. Potentiometers operated by a mechanism can be used as position transducers, for example, in a joystick. Potentiometers are rarely used to directly control significant power (more than a watt), since the power dissipated in the potentiometer would be comparable to the power in the controlled load.

Potentiometers comprise a resistive element, a sliding contact (wiper) that moves along the element, making good electrical contact with one part of it, electrical terminals at each end of the element, a mechanism that moves the wiper from one end to the other, and a housing containing the element and wiper.

A 10 k Ω Potentiometer (Midori Precisions, CPP-60, linearity ±0.05%) attached to the kingpin of one of the steering wheels provides the steering angle (alignment error: ±3.2 deg.).

By 1970, CPP Series had been developed as Angle Sensor of the first precision conductive potentiometer in Japan. Today this model has typically 3 kinds of diameters; CPP-35 series (Φ 36.5mm), CPP-45 series (Φ45mm), CPP-60 series (Φ62mm). These models are available with many special specifications; for example, electrical & mechanical angle range, shaft shape, center tap, multi gangs, water proof type, and so on. Applications can be found in many areas in construction machinery, measuring instruments, factory automation, process automation, so on.
Fig. 2.3 shows the layout of the potentiometer.

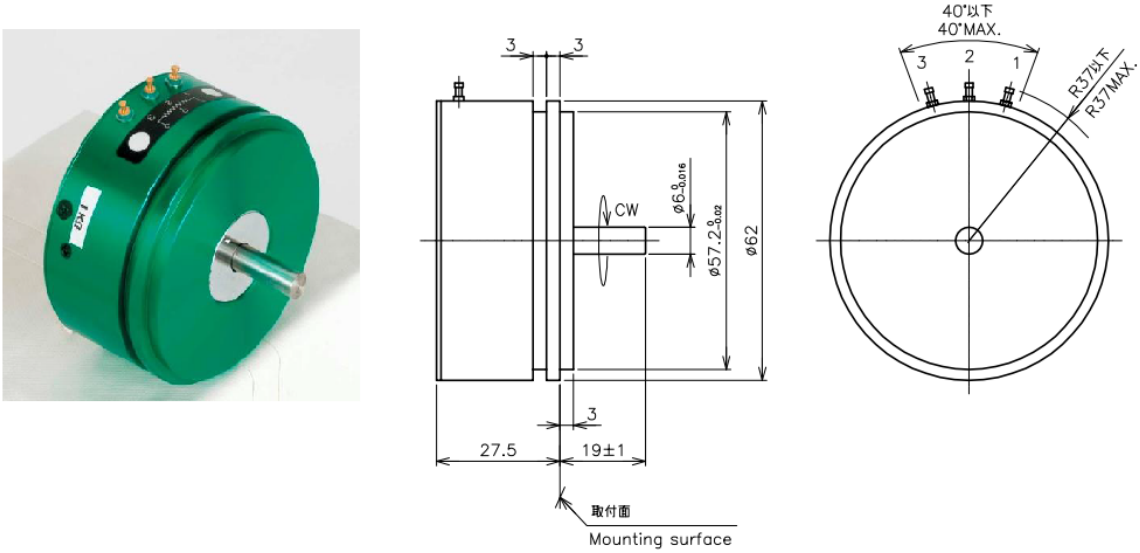


Fig. 2.3. Midori Precisions, CPP-60 potentiometer layout.

Table 2.4 lists the electrical specifications of the CPP-60 potentiometer.

Table 2.4. Electrical specifications of the CPP-60 potentiometer.

Specification	Value	Units
Effective Electrical Travel	355 +1, -2	°
Total Resistance	0.5, 1, 2, 5, 10, 20	kΩ
Total Resistance Tolerance	±15	%
Independent Linearity	±0.05	%
Rated Dissipation	3/70℃	W
Output Smoothness	MAX. 0.1	%
Insulation Resistance	MIN. 100/DC1000V	MΩ
Dielectric Strength	AC1000/1 Minute	V
Temperature Coefficient of Resistance	±400	ppm/K

For a copy of the CPP-60 potentiometer data sheet, please refer to Appendix D.

2.5 Bifocal imaging device (Fujifilm 2-in-1 Camera)

The bifocal imaging device is a non-commercial camera developed by Fujifilm Corporation. The innovation of the device consist of simultaneous shooting a wide-angle image and a telephoto image. Due to its capability of shooting two images at the same time, the developer refers to it as the Fujifilm 2-in-1 camera. The 2-in-1 camera has a composite shooting lens with a maximum diameter of Φ70mm. The camera's focal lengths are 25mm (35mm format equivalent) for the wide-angle image, and 330mm (35mm format equivalent) for the telephoto image. The camera's image sensor is a 2/3 inch CMOS sensor, with a pixel size of 7.1μm and an actual number of pixels of 1280 x 480.

It provides a digital video output of 30 frames per second under the HD-SDI format. For this research purposes, the Fujifilm 2-in-1 camera has two main advantages over the use of two separate conventional cameras. First, thanks to its composite shooting lens the wide-angle and telephoto video frames are synchronized; whereas two separate conventional cameras provide image frames with slight time differences. Second, the position of the telephoto image inside the wide-angle image is fixed and know for all video frames despite mechanical vibration of the test vehicle; whereas two separate conventional cameras will vibrate at different paces causing that the position of the telephoto image inside the wide-angle image changes randomly in each frame. Fig. 2.4 shows the layout of the camera.

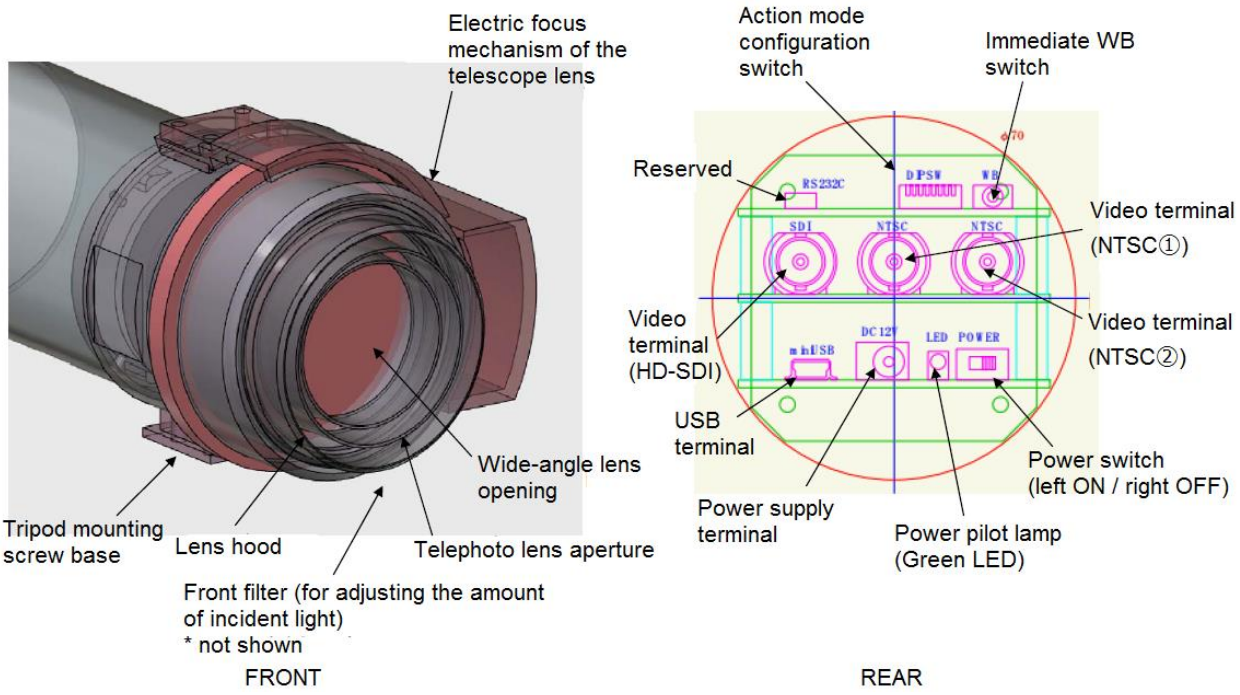


Fig. 2.4. Fujifilm 2-in-1camera layout.

Since the camera is a non-commercial prototype it has some hardware limitations; for example it does not have a diaphragm mechanism. The amount of incident light is adjusted by the selection and replacement of an ND filter in the front surface of the lens. Due to its automatic exposure, for outdoor shooting it is necessary to equip a 1% transmittance filter and for indoor shooting it is necessary to equip a 10% transmittance filter. However, both filters are generally suitable for a wide range of exposures. If the output image is too dark due to insufficient exposure amount, it is necessary to use the device selecting either the high transmittance filter or no filter at all. If the output image is in uniform white or gray this is because the exposure amount is excessive, making it necessary to use a low transmittance filter. The filter mount is a screw thread mounting.

The focus adjustment of the telephoto lens uses an electric focusing mechanism attached to the camera body, as shown in Fig. 2.4. By the rotation of the internal shaft of an electric motor, the frontal part of the lens barrel is rotated along with the gear. Table 2.5 lists the specifications of the Fujifilm 2-in-1 camera. For a copy of the Fujifilm 2-in-1 layout, please refer to Appendix E.

Table 2.5. Specifications of the Fujifilm 2-in-1 camera

Specification	Value
Shooting lens	
Diameter	Φ70mm (max)
Focal length	Wide-angle: 25 mm
	Telephoto: 330 mm
Image sensor	2/3 inch CMOS
	pixel size 7.1μ □
Video output	30fps color video
Digital video	HD-SDI x 1ch, 1280x720
Power	12 V DC/ 3 W
Weight	1.9 kg

Fig. 2.5 shows the Fujifilm 2-in-1 camera field of view, the full size of the side by side image is 1280 x 480 pixels. The left side corresponds to the wide-angle image and the right side corresponds to the telephoto image; both images have a size of 640 x 480 pixels respectively. The blue rectangle represents the Region of Interest (ROI) used for crop row detection, which has a size of 420 x 256

pixels. The red rectangle has a size of 53 x 40 pixels and it represents the position of the telephoto image inside the wide-angle image.

Fig. 2.5. Fujifilm 2-in-1camera side by side images.

The telephoto image was set to a zoom factor of twelve; magnifying the image area of 53 x 40 pixels represented by the red rectangle in Fig. 4 into an image area of 640 x 480 pixels. This shows an advantage in the use of the 2-in-1 camera, because an increased image resolution can be achieved for a specific area with fixed position inside the wide-angle image. Therefore, the wide-angle image was used for crop mapping and crop row detection while the telephoto image was used for correction of the central row crop row detection. Chapter 4 and 5 include a more detailed description concerning the application of the device.

2.6 Conclusions

This chapter described the equipment used during this research. The research platform used for the field experiments was described part by part. The test vehicle's main technical specifications were introduced. Each one of the sensors used in the experiments were described, highlighting their most relevant aspects.

Chapter 3 Estimation of vehicle status using nonlinear vehicle motion model

The increased performance of today's agricultural vehicles can be attributed to the improved understanding of how vehicles behave throughout a wide range of operating conditions. This understanding has been achieved through the use of advanced analysis tools, which are able to model the most complex nonlinear behavior to a high level of fidelity. The change in development methodology has allowed manufacturers to introduce new agricultural vehicles to the market faster while at the same time reducing development costs. The extra development time and money can be used to optimize agricultural vehicle systems which ultimately results in the higher levels of quality and refinement that farmers have come to expect.

This chapter presents the vehicle modeling framework used in this research. It presents a brief description of the most common vehicle control strategies. The vehicle dynamics are explained thru the description of the tire mechanics. Measurements of the tire's lateral forces as a function of the tire's slip angle of the test vehicle traveling on a concrete surface are reported. A description about how these tire parameters can be used to improve the navigation model of the test vehicle is given as well.

3.1 Vehicle control strategies

Control strategies applied to vehicles can be classified in many different ways. For example they can be grouped based on the design method used in their development (optimal, robust, neural networks) or by the type of actuation used (active, semi-active, etc.). However, in the area of vehicle robotics it is much more useful to group the various control approaches according to the type of vehicle behavior that they are designed to stabilize. By classifying control methods in this way, three primary areas are observed within vehicle handling: yaw rate control, roll mitigation, and lateral positioning control. These three areas have been the focus of most of the work over the last decade regarding vehicle handling, whereas roll mitigation has become more popular in recent years due to the popularity of larger vehicles such as SUV's and trucks.

3.1.1 Yaw rate control

The majority of control approaches that deal with vehicle handling revolve around the use of yaw rate feedback in calculating control functions. This is because most of the disturbances such as road irregularities and actuator nonlinearities can be by-passed through the use of the yaw rate measurement. The majority of commercially available chassis control technologies utilize yaw rate feedback in conjunction with other subsystems, such as anti-lock brakes (ABS), or traction control (TCS). The most popular term used by manufacturers who implement these technologies is electronic stability control (ESC).

First developed by Bosch in 1995, ESC methods typically monitor yaw rate, lateral velocity and body accelerations in order to determine the state of the vehicle. These states are then compared to vehicle models in order to determine the vehicle's stability state. Inputs from the driver, such as, steer angle and throttle position are used to determine the driver's intent when calculating the response. The controller can then use the ABS or TCS systems to carry out the control function it deems necessary in order to maintain yaw stability. This system has been shown to be very successful at maintaining stability in low friction situations, such as in wet weather or icy conditions.

3.1.2 Roll mitigation

This control approach is also known as active rollover protection (ARP). Basically, it is a system that recognizes rollover and avoids it by selectively applying braking sequences. Excessive lateral force, generated by excessive speed in a turn, may result in a rollover. ARP systems automatically responds whenever it detects a potential rollover by rapidly applying the brakes with a high burst of pressure to the appropriate wheels and decreasing the engine torque to interrupt the rollover before it occurs.

Additionally, rollover stability systems have begun to incorporate an active suspension systems in rollover protection. To accomplish this, the onboard computer uses data from an Inertial Measurement Unit (IMU) to determine when a vehicle is in a rollover condition independent of yaw rate and vehicle speed. When the computer determines that the vehicle is at risk of rollover, it calculates the direction of roll and activates the active suspension system. The force produced in the suspension creates a torque opposite to that created by the lateral force, and keeps the vehicle safe. Roll mitigation is

becoming increasingly popular on Sports Utility Vehicles (SUV) and trucks, since their high center of gravity makes them prone to rollovers.

3.1.3 Lateral positioning control

The purpose of lateral positioning control is to maintain the vehicle's position on the road surface. In particular, this kind of steering controller is designed to track the center of the present lane on both curved and straight pathway sections without knowledge of the radius of curvature of the path. Also, a lane change maneuver can be completed on a curved road section using this scheme of control.

For the most part the driver is not considered in this control method and therefore these control approaches are regulated to autonomous vehicle applications. The approaches are also broad in terms of feedback and actuation methods. For example GPS, radar, and vision systems are some of the common methods used to measure the vehicle's state.

3.2 Vehicle dynamics

Modern agricultural vehicles have dramatically improved in key areas such as safety, performance, durability and efficiency. In order to realize these goals, manufacturers have relied heavily on advanced vehicle modeling and computer control.

The implementation of computer control has had a profound impact on vehicle safety, with present control systems having the ability to monitor the state of a vehicle thousands of times per second. Whatever the implementation, there is no doubt that computer control has had a dramatic effect on how vehicles of today function and most importantly, on what customers have come to expect from manufacturers. No single area of vehicle development is this more evident than in vehicle dynamics, where these technologies have become common place. Any discussion on the topic of vehicle dynamics and control can be broken down into three fundamental areas; longitudinal dynamics, handling (planar dynamics) and ride (vertical dynamics). This thesis focuses on the handling dynamics of the test vehicle and therefore a brief development of important concepts associated with handling behavior will be presented.

3.2.1 Vehicle kinematic model

The kinematic model gives an estimation of the vehicle's position and heading, described by Eq. (3.1) (Zakaria et al., 2013 and Zhao et al., 2011):

$$\varphi = \varphi_0 + \int \omega \, dt \tag{3.1}$$

$$x = x_0 + \int V \sin(\varphi + \beta) \, dt$$

$$y = y_0 + \int V \cos(\varphi + \beta) \, dt$$

Where φ is the vehicle's heading, obtained by integration of the yaw rate ω. Also, x and y are the position coordinates (x, y) of the vehicle's center of mass in an inertial frame (X, Y). As described in Eq. (3.1) the position coordinates (x, y) are obtained by integration of the vehicle's body sideslip β and the yaw rate ω. The inertial frame (X, Y) describes the global position of the vehicle as shown in Fig. 3.1.

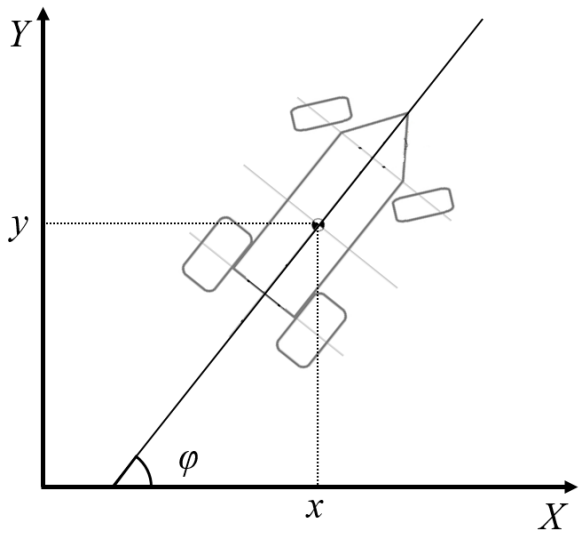


Fig. 3.1. Vehicle kinematic model.

Several variations of the vehicle kinematic model shown in Fig. 3.1 have been used in numerous researches (Bevly et al., 2002); but it should be noted that the general form is the same and its accuracy depends on the vehicle's body sideslip β and the yaw rate ω.

3.2.2 Tire dynamic properties

The pneumatic tire might be the simplest part of a vehicle, but at the same time is one of the most complicated devices to model. Tire response is dependent on dozens of variables, some of which include: internal pressure, temperature, chemical compound and tread design. In terms of vehicle dynamics, the tire is one of the most important factors in determining how a vehicle responds to driver inputs. This is due to the fact that the only portion that connects a vehicle to the road is the contact patch of each tire. This situation is more complicated for tractors and other agricultural vehicles, since loose soil conditions differ from on road conditions. Therefore, an extensive amount of work has been conducted over the years in order to investigate how a tire behaves under various conditions. This section will highlight some of the basic concepts involved in their analysis.

Slip angle:

In order for a tire to produce the traction forces that are required to accelerate a vehicle in a straight line or around a corner, it is necessary for the tire to develop a slip angle. The slip angle can be explained by the fact that the elastic characteristics of a tire allow it to be pointed in a direction different from which the vehicle is heading. The angle between these two headings is known as the lateral slip angle. Fig. 3.2 (Wong, 1993) shows a bottom view of a tire contact patch which is being subjected to a slip angle.

Due to elastic distortion within the contact patch, the wheel travels along a different heading than that which it is pointed. The angle between these two headings is known as the lateral slip angle. As tread enters the contact patch it builds up at the leading edge causing the lateral distortion shown above. In this region, the normal load is the highest allowing the tread to adhere to the road. As the tread moves rearward through the contact patch, the normal load decreases causing the tread to start slipping before it is allowed to recover back into the main tread, assuming its original undistorted shape. The lateral traction is a result of the elastic reaction forces created by the distortion in the contact patch, which means that tires cannot produce traction forces without first assuming a slip angle.

When a rolling tire is subjected to a slip angle, the leading edge of the tread contacts the ground slightly to one side of the rest of the contact patch. As the tire continues to roll, each small increment of tread rubber coming into the contact patch contacts the road a small distance toward the direction the tire is

pointed. When all of the small deviations in the tire's contact patch are added up, it results in the vehicle moving along a curved path.

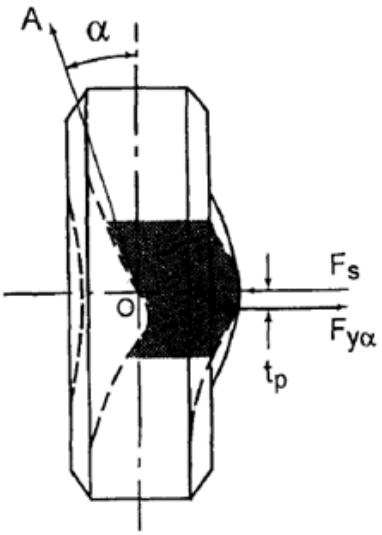


Fig. 3.2. Bottom view of a tire contact patch subjected to a lateral slip angle.

Traction forces:

A tire does not generate traction forces until it is steered away from its current course and develops a slip angle. Fig. 3.3 (Liburdi, 2010) shows a typical lateral force curve of a radial tire. The shape of this curve is not the same for all tires, but rather is a specific characteristic of the design of the tire.

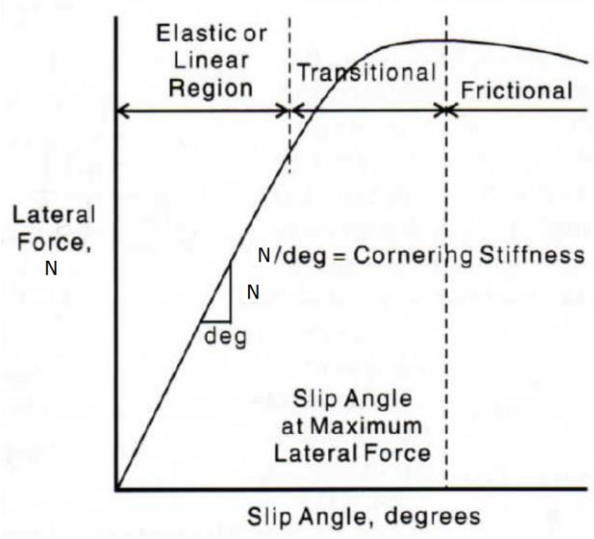


Fig. 3.3. Lateral traction force curve of a pneumatic tire.

Notice that this curve has three distinct regions. First there is a straight section at small slip angles, where an increase in slip angle gives a proportional increase in lateral force. The slope of this section of the curve is the cornering stiffness of the tire. In this region of the curve, the tread is not sliding on the road at any point in its contact patch. At higher slip angles, portions of the tire patch are sliding and there is less of an increase in lateral force with respect to slip angle. This is called the transitional region. As the curve reaches its peak, more of the contact patch is sliding and the tire produces less lateral force. After the peak of the curve, lateral force can reduce a lot within a few degrees of extra slip angle. At these high slip angles, most of the contact patch is sliding; producing an increased amount of heat and wear. This high slip angle region is also known as tire saturation or frictional region, since the tire is no longer able to generate any extra lateral force. The force curve decreases with increases to slip angle which is known as tire saturation. A similar situation exists for longitudinal traction forces.

Normal load dependency:

One of the unique properties of the pneumatic tire is known as normal load dependency. Referring to Fig. 3.4 (Liburdi, 2010), it is seen that as the normal load on a tire increases, the maximum traction force that the tire can produce also increases; however, this occurs in a nonlinear way.

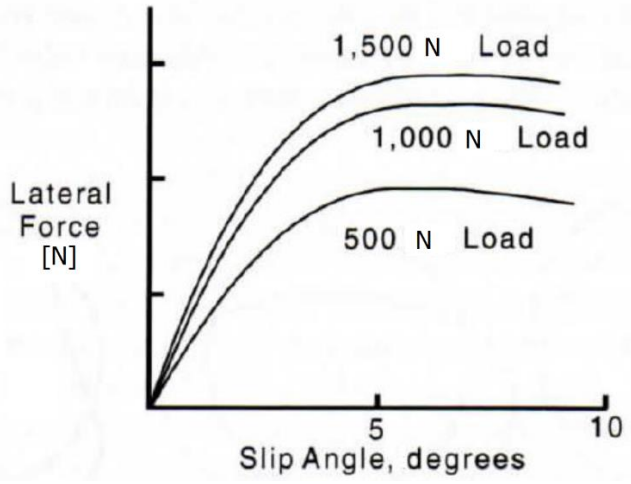


Fig. 3.4. Normal load dependency of a pneumatic tire.

This concept is critical in the design of an agricultural vehicle chassis for specific handling behavior. The reasoning behind this is that as a vehicle corners and starts generating lateral acceleration, a portion of the weight that was once on the inside tires is transferred to the outside tires. Due to the normal load dependency of the tire, load transferred across an axle results in a larger reduction of traction forces on

the inside tires than that which is gained by the outside tires. The result is a net decrease in the traction capability of the two tire combined.

3.2.3 Vehicle dynamic model

The area of vehicle handling primarily deals with motions within the plane of the road surface; which include lateral and longitudinal speeds, and the yaw rate. However, handling also encompasses out of plane dynamics such as roll and pitch motions, which generally need to be included when load transfer is an important factor in dictating tire performance.

Vehicle handling models can be grouped into two main areas: linear and nonlinear. Linear models have the benefit of using frequency response methods, such as transfer function representations and eigenvalue analysis. These methods can be used to investigate the stability of the system and are very valuable in demonstrating any fundamental dynamic modes that may exist. Linear models are generally a result of a system simplification, which usually limits their application to narrow operating ranges.

Nonlinear modeling is generally more accurate over a wider operating range, but requires a more complicated system representation. Usually nonlinearities introduced into handling models are due to the method in which tire traction forces are calculated or how the suspension system is modeled. Modern software tools use nonlinear models of suspension elements, such as springs, bushings, dampers and tires in order to predict behavior over the vehicle's entire operating range. Although these software tools are very powerful, the vast majority of vehicle models used within controller design are linear. This is due to the fact that they do not require large computational resources to calculate vehicle states and are generally accurate enough to capture the important dynamics involved.

Fig. 3.5 shows the typical configuration of the bicycle model for a four wheeled vehicle (Wong, 1993). The input of the system is given by the vehicle's velocity V and the steering angle δ. The output of the system is given by the vehicle's body sideslip angle β and the yaw rate ω.

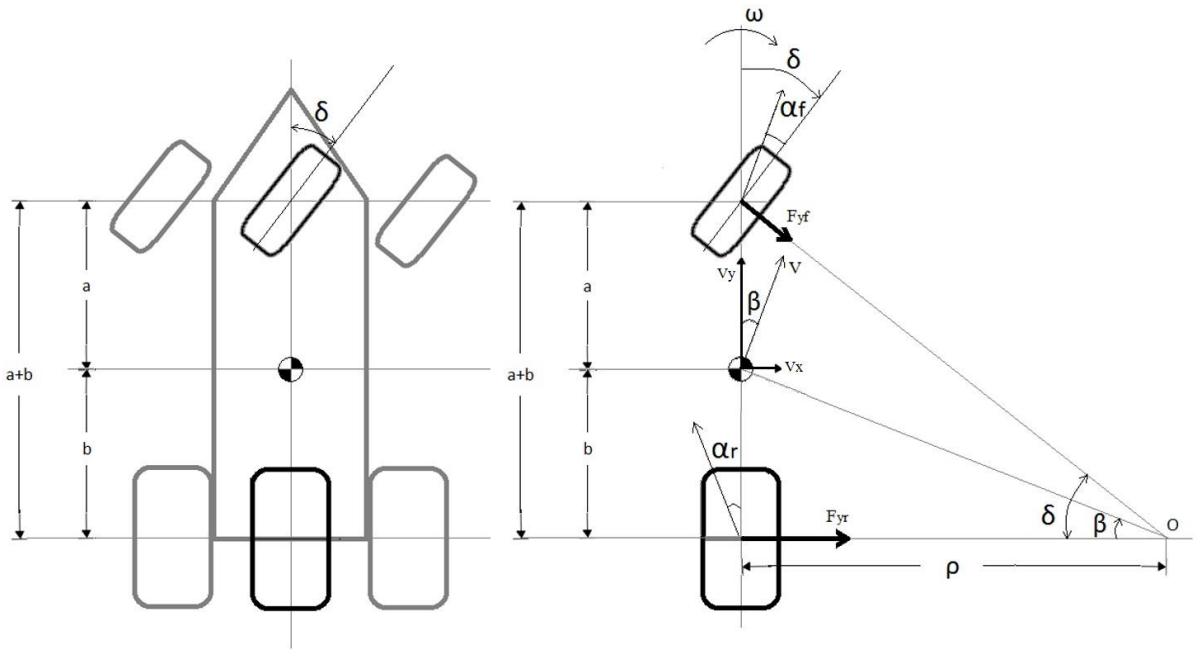


Fig. 3.5. Schematic of the bicycle model.

In addition the Fig. 3.6 shows the relation between the movement of the vehicle's center of gravity CG and the yaw rate ω. In this thesis, the most common model associated to the bicycle model is presented as follows.

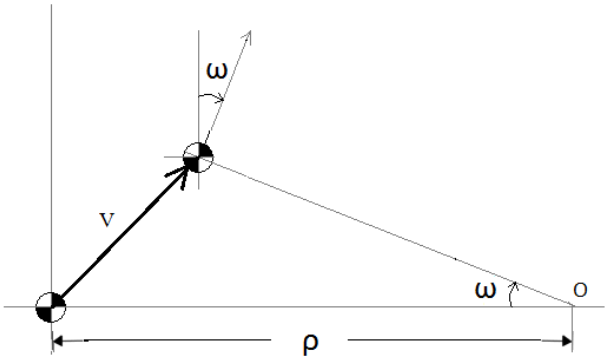


Fig. 3.6. Geometric relation of the yaw rate.

Bicycle dynamic model:

The vehicle motion is derived from a kinematic representation based on the assumption of rolling without sliding at the tire/ground contact point. If such a hypothesis is relevant in the urban vehicle

context, its direct transposition to all-terrain vehicles leads to inaccurate results. Consequently, low-grip conditions have to be properly modeled. The bicycle dynamic model incorporates variables representative of sliding phenomena, namely tire sideslip angles (Liljedahl et al., 1989). These variables denote the difference between wheel plane orientation and actual speed vector direction. In other words, we need to consider the sliding in the vehicle's lateral dynamics.

Since we are describing a dynamic framework, agricultural vehicles (and thus, agricultural mobile robots) dynamics can also be described using the bicycle model shown in Fig. 3.5. As depicted in Fig. 3.5, the angles α_f and α_r are the slip angles on the front and rear wheel respectively and are defined as the difference between a wheel's direction of heading and its direction of travel. Since this model lumps the wheels on each axis together, equal slip angles and road wheel angles are assumed on the left and right wheels on both axis. The tires are assumed to generate lateral forces that are directly proportional to the tire slip angle. Hence, the lateral forces are modeled as shown in Eq. (3.2):

$$F_{yf} = 2C_{\alpha f}^{tire}\alpha_f = C_{\alpha f}^{axle}\alpha_f, \qquad F_{yr} = 2C_{\alpha r}^{tire}\alpha_r = C_{\alpha r}^{axle}\alpha_r \tag{3.2}$$

Where $C_{\alpha f}^{axle}$ and $C_{\alpha r}^{axle}$ represent the cornering stiffnesses on each axis, which here equals two times the cornering stiffness on each wheel, and is defined as the slope of the curve seen in Fig. 3.3 for the elastic region. The bicycle model consists of two degrees of freedom; namely the vehicle's body sideslip angle β, and the vehicle yaw rate ω. By analyzing the free body diagram shown in Fig. 3.5, the governing equations of motion can be derived from Newton's second law yielding Eq. (3.3):

$$\sum F_y = m\left(V_x\omega + \dot{V}_y\right) = F_{yf}\cos\delta + F_{yr}, \qquad \sum M_z = I_{zz}\dot{\omega} = aF_{yf}\cos\delta - bF_{yr} \tag{3.3}$$

Where m represents the vehicle mass and I_{ZZ} represents the vehicle moment of inertia around the z-axis. The distances between the vehicle's center of gravity (CG) and the front and rear axis are labeled as a and b respectively. From Fig. 3.5, the relationship $\tan\beta = {V_y}/{V_x}$ can be found at the center of gravity. By assuming small angles ($\tan\beta \approx \beta$), $\dot{V}_y$ can be written as $\dot{V}_y = \frac{d}{dt}(\beta V_x) = V_x\dot{\beta} + \dot{V}_x\beta$. Inserting this relationship together with the expressions for the lateral forces given by Eq. (3.2) into Eq. (3.3), we obtain Eq. (3.4):

$$m\left(V_x\omega + V_x\dot{\beta} + \dot{V}_x\beta\right) = C_{\alpha f}^{axle}\alpha_f \cos\delta + C_{\alpha r}^{axle}\alpha_r$$
$$I_{zz}\dot{\omega} = aC_{\alpha f}^{axle}\alpha_f \cos\delta - bC_{\alpha r}^{axle}\alpha_r \quad (3.4)$$

The slip angles can be defined in terms of the vehicle motion variables, β and ω. The expression for the front wheel side slip angle can be formulated by considering the velocity of the center point of the front wheel being $(V_x, V_y + a\omega)$, and then:

$$\tan(\delta - \alpha_f) = \frac{V_y + a\omega}{V_x} \quad (3.5)$$

And similarly for the rear wheel:

$$\tan\alpha_r = \frac{-V_y + b\omega}{V_x} \quad (3.6)$$

Assuming small angles, and again recognizing $V_y/V_x \approx \beta$ yields the final slip angle expressions used by the bicycle dynamic model as given in Eq. (3.7):

$$\alpha_f = -\beta - a\frac{\omega}{V_x} + \delta\,, \quad \alpha_r = -\beta + b\frac{\omega}{V_x} \quad (3.7)$$

Inserting the expressions given by Eq. (3.7) into Eq. (3.4), and assuming small steering angles we obtain the following equations of lateral and yaw motion, summarized as Eq. (3.8):

$$mV_x\omega + mV_x\dot{\beta} + m\dot{V}_x\beta = C_{\alpha f}^{axle}(-\beta - a\frac{\omega}{V_x} + \delta) + C_{\alpha r}^{axle}(-\beta + b\frac{\omega}{V_x})$$
$$I_{zz}\dot{\omega} = aC_{\alpha f}^{axle}(-\beta - a\frac{\omega}{V_x} + \delta) - bC_{\alpha r}^{axle}(-\beta + b\frac{\omega}{V_x}) \quad (3.8)$$

Which can be rewritten into Eq. (3.9) as:

$$mV_x\dot{\beta} = -\beta\left(C_{\alpha f}^{axle} + C_{\alpha r}^{axle} + m\dot{V}_x\right) - \omega\left(mV_x + \frac{aC_{\alpha f}^{axle}}{V_x} - \frac{bC_{\alpha r}^{axle}}{V_x}\right) + \delta C_{\alpha f}^{axle}$$
$$I_{zz}\dot{\omega} = -\beta(aC_{\alpha f}^{axle} - bC_{\alpha r}^{axle}) - \omega\left(\frac{a^2C_{\alpha f}^{axle} + b^2C_{\alpha r}^{axle}}{V_x}\right) + \delta aC_{\alpha f}^{axle} \quad (3.9)$$

The vehicle's longitudinal velocity V_x is supposed to be slow varying, so that longitudinal forces can be disregarded. Moreover, this longitudinal velocity can be considered as constant, since this is the

typical case for tractors and agricultural vehicles, were the vehicle's operation speed do not change and thus the longitudinal acceleration $\dot{V}_x$ can be considered as cero. We can then rewrite Eq. (3.9) on state space form, as shown in Eq. (3.10):

$$\begin{bmatrix}\dot{\beta}\\ \dot{\omega}\end{bmatrix} = \begin{bmatrix}\frac{-C_{\alpha f}^{axle}-C_{\alpha r}^{axle}}{mV_x} & \frac{-aC_{\alpha f}^{axle}+bC_{\alpha r}^{axle}}{mV_x^2}-1\\ \frac{-aC_{\alpha f}^{axle}+bC_{\alpha r}^{axle}}{I_{zz}} & \frac{-a^2C_{\alpha f}^{axle}-b^2C_{\alpha r}^{axle}}{I_{zz}V_x}\end{bmatrix} * \begin{bmatrix}\beta\\ \omega\end{bmatrix} + \begin{bmatrix}\frac{C_{\alpha f}^{axle}}{mV_x}\\ \frac{aC_{\alpha f}^{axle}}{I_{zz}}\end{bmatrix} * \delta \tag{3.10}$$

For this model, it is assumed that the cornering stiffnesses $C_{\alpha f}^{axle}$ and $C_{\alpha r}^{axle}$ are entirely representative of sliding effects. Table 3.1 gives a summary of the parameters used in this model.

Table 3.1. Parameters of the bicycle dynamic model.

Variable	Description	Units
Vx	longitudinal velocity	m/s
β	sideslip angle	deg
ω	yaw rate	deg/s
m	vehicle mass	kg
Izz	yaw moment of inertia	kg*m^2
$C_{\alpha f}^{axle}$	frontal axle cornering stiffness	N/deg
$C_{\alpha r}^{axle}$	rear axle cornering stiffness	N/deg
a	distance between front axle and CG	m
b	distance between rear axle and CG	m
δ	front wheel steering angle	deg

Parameters like *m*, *a* and *b* where taken from the vehicle's technical specifications, as described in section 2.1.1. Taking into account this model, the next section describes some of the most common control strategies.

3.3 Tire dynamic properties determination method

As mentioned in section 3.1.1, the majority of control approaches that deal with vehicle handling revolve around the use of yaw rate feedback in calculating control functions. Therefore this section describes the procedure to determine the tire's dynamic properties and the vehicle motion variables β and ω (Ospina and Noguchi, 2016).

The experimental runs were taken in a flat, dry concrete surface performing extreme cornering maneuvers. For this tests, the tire–road friction dynamics can be ignored because the road surface conditions did not change during the experiments. The cornering maneuvers consisted of the vehicle being driven in a sinusoidal trajectory; performing a hard left turn followed by a hard right turn while the velocity is held fairly constant. The experimental path was outlined by using reference landmarks so the driver knew how to steer the vehicle. A total of three runs following the same trajectory were performed. The results of all the three runs were quite similar, and the overall behavior of the outputs in response to the input is roughly the same. Therefore, this thesis summarizes the results from one experiment.

The parameters listed in Table 3.1 were used in combination with Eqns. (3.2) to (3.10) to simulate the vehicle behavior for comparison with the experimentally measured states. The vehicle's yaw moment of inertia was obtained using the approximation $I_{zz}=m_f a^2+m_r b^2$; where m_f=1260 kg is the vehicle's mass in the front axle and m_r=2005 kg is the vehicle's mass in the rear axle. The yaw moment of inertia I_{zz} plays a key role in the dynamic model shown in Eq. (3.10) since it is a divisor factor that modifies the former values of both the sideslip β and the yaw rate ω into the current value of the yaw rate ω. The yaw moment of inertia I_{zz} is an important parameter, due to the fact that it reflects the vehicle's resistance to change its direction; this means that a big yaw moment of inertia I_{zz} makes the vehicle slower to swerve or go into a tight curve, and it also makes it slower to turn straight again.

The vehicle's longitudinal velocity V_x was intended to be constant during the maneuvers because this is the case for most agricultural vehicles during actual operation. Moreover, the main input of the models is the steering angle δ; the vehicle has to be turning in order to produce a yaw rate ω and a sideslip angle β. The vehicle's longitudinal velocity V has an effect over the outputs ω and β, but the vehicle's longitudinal velocity V_x operates as a divisor in Eq. (3.10). Changing the magnitude of the vehicle's longitudinal velocity V_x will change the magnitude of the outputs ω and β but not their way of oscillation; this is a reflection of the input δ. Although the vehicle's top speed is 8 m/s (29 km/h), the test were performed at an average speed of 2 m/s (7.5 km/h) for safety reasons.

3.3.1 Improvement of the estimations of the vehicle dynamic model

The model input all over the trajectory is the steering angle δ, shown in Fig. 3.7.

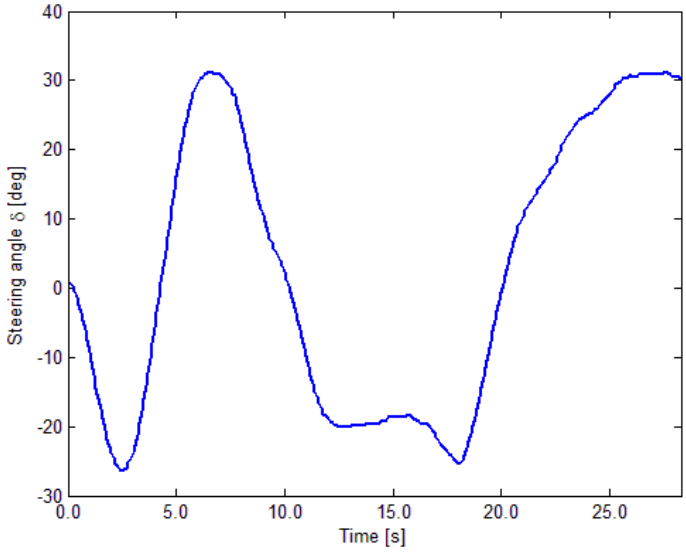


Fig. 3.7. Time series of steering angle δ input.

The body sideslip β was calculated as the difference of the direction of travel given by the GPS and the heading given by the FOG (Bevly, 2004), as shown in Eq. (3.11).

$$\beta = \varphi_{GPS} - \varphi_{IMU} \tag{3.11}$$

The yaw rate ω was obtained directly from the FOG. The measured sideslip β and yaw rate ω will be considered as the true data, and it will be compared with the data obtained from the model. The tire's slip angle can be calculated for both the front axle and the rear axle as shown in Eq. (3.12):

$$\alpha_f = \beta + a\frac{\omega}{V} - \delta\,, \quad \alpha_r = \beta - b\frac{\omega}{V} \tag{3.12}$$

Using the measurement given by Eq. (3.11) and the steering angle δ shown in Fig. 3.7; it is possible to obtain measurement estimations for the tire slip angles (Baffet et al., 2006). We can calculate the lateral forces (Bevly et al., 2006) by analyzing the bicycle model shown in Fig. 3.5 with the parameters listed in Table 3.1 and using Newton's equations as shown in Eq. (3.13). The lateral acceleration $\dot{y}$ was obtained directly from the FOG and it was numerically differentiated to obtain $\ddot{y}$.

$$\sum F_y = m\ddot{y} = F_{yf}\cos\delta + F_{yr}\,, \qquad \sum M_z = I_z\dot{\omega} = aF_{yf}\cos\delta - bF_{yr} \tag{3.13}$$

Solving the equations simultaneously, it is possible to obtain relationships for the frontal and rear lateral forces as shown in Eq. (3.14):

$$F_{yf} = \frac{I_z\dot{\omega} + bm\ddot{y}}{(a+b)\cos\delta}\,, \qquad F_{yr} = m\ddot{y} - F_{yf}\cos\delta \tag{3.14}$$

The lateral tire forces are a function of the cornering stiffness (Baffet et al., 2009), as shown in Eq. (3.15):

$$F_{yf} = 2C_{\alpha f}^{tire}\alpha_f = C_{\alpha f}^{axle}\alpha_f, \qquad F_{yr} = 2C_{\alpha r}^{tire}\alpha_r = C_{\alpha r}^{axle}\alpha_r \tag{3.15}$$

It is necessary to consider a multiplier factor of 2 in Eq. (3.15) because the bicycle model considers both the left and right tires as one single tire for each axle. It is possible to estimate the cornering stiffness for the rear and frontal axles using the lateral forces calculated in Eq. (3.14) and the tire slip angles obtained from the sensors in Eq. (3.12) and computing them in Eq. (3.15).

It possible to estimate a constant cornering stiffness from Eq. (3.15), which is the typical approach for on-road vehicles; the estimated front and rear tire cornering stiffness are Cf = 1.342 x10^3 N/deg. and Cr = 4.065 x10^3 N/deg. respectively. These values were used into the dynamic model described in Eq. (3.10). However, contrary to on-road vehicles for off-road vehicles it is difficult to determine a constant cornering stiffness from the experimental data due to the differences between the lateral force F_y and the tire sideslip angle α. Moreover, due to the transitory nature of the pneumatic tire, for a given value of α there are multiple values of F_y and thus it is not possible to obtain the tire lateral force as a linear function of the tire sideslip angle from the measured data.

Several methodologies (Burhaumudin et al., 2012; Pacejka et al., 1992; and Pacejka, 2006) have been implemented in order to obtain a linear F_y/α relation from the experimental data. However, in agricultural applications a linear model is not the right approach since it considers ideal conditions that might not apply to muddy and slippery soils. We are interested into consider the hysteresis behavior of the F_y/α relation experimental data. Since no simple parametric models can effectively describe the tire force generating functions $F_{yf}(\alpha_f)$ and $F_{yr}(\alpha_r)$ (Ospina and Noguchi, 2016), "non-parametric" tire functions (this means functions that do not depend on several parameters, such as longitudinal slip, sideslip angle, normal load, camber angle, tire pressure, wear, and road surface characteristics) are proposed in this thesis to represent the non-linear characteristic curves. The methodology is quite simple (Ospina and Noguchi, 2018); by applying the method of the least squares to the experimental data, we find that a polynomial of degree 4 will fit the behavior of the cornering maneuver. The results of this regression are shown in Fig. 3.8 and Fig. 3.9.

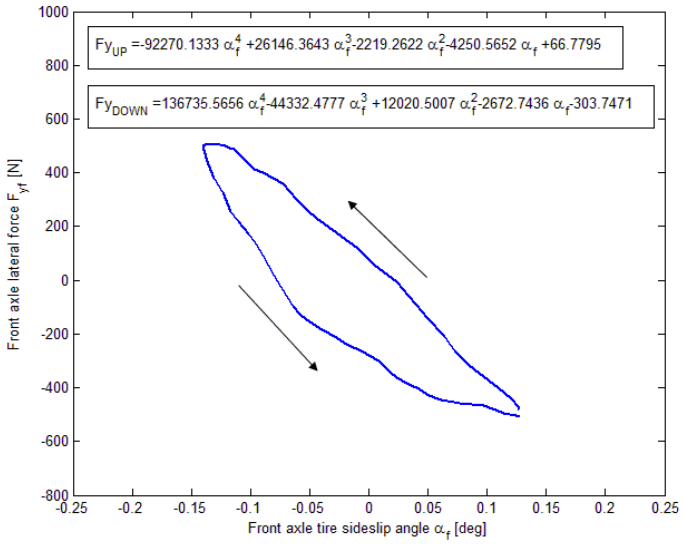


Fig. 3.8. Frontal lateral force vs rear tire sideslip.

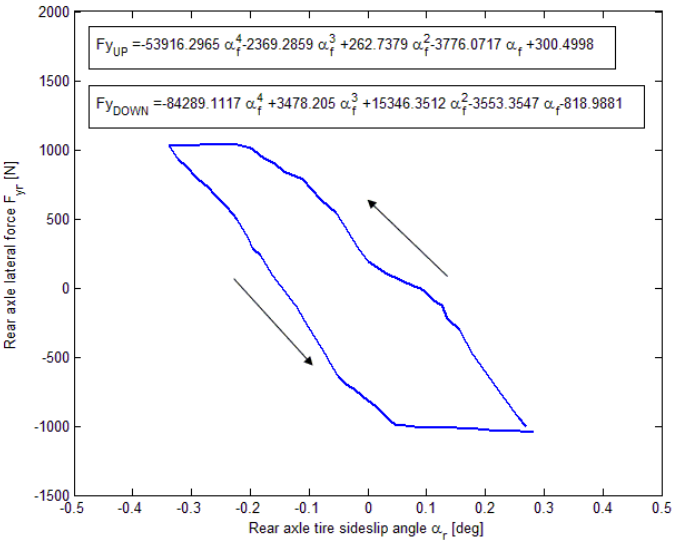


Fig. 3.9. Rear lateral force vs rear tire sideslip.

The equations found using the regression are included in Fig. 3.8 and Fig. 3.9 for the increasing (F_{yUP}) and decreasing (F_{yDOWN}) portion of the F_y/α relation. The increasing (F_{yUP}) portion corresponds to the top part of the loop, marked with the increasing arrow. The decreasing (F_{yDOWN}) portion corresponds to the bottom part of the loop, marked with the decreasing arrow.

In order to validate the functionality of the proposed regression model, it is necessary to perform a simulation using the equations included in Fig. 3.8 and Fig. 3.9 evaluating them for the values of the tire sideslip angles α_f and α_r all over the trajectory. By recombining Eq. (3.12) and Eq. (3.15), it is possible to obtain the body sideslip β and the yaw rate ω in terms of the lateral forces F_{yf} and F_{yr}, as shown in Eqns. (3.16) and (3.17):

$$\omega = \left(\frac{V}{a+b}\right)\left[\frac{F_{yf}}{C_{\alpha f}^{axle}} - \frac{F_{yr}}{C_{\alpha r}^{axle}} + \delta\right] \tag{3.16}$$

$$\beta = \frac{F_{yr}}{C_{\alpha r}^{axle}} + \left(\frac{b}{a+b}\right)\left[\frac{F_{yf}}{C_{\alpha f}^{axle}} - \frac{F_{yr}}{C_{\alpha r}^{axle}} + \delta\right] \tag{3.17}$$

The resulting force values obtained from the equations included in Fig. 3.8 and Fig. 3.9 are then evaluated into Eq. (3.16) and Eq. (3.17) to obtain the regression modeled body sideslip β_{REG} and yaw rate ω_{REG}.

The constant cornering stiffness Cf = 1.342 x10^3 N/deg. and Cr = 4.065 x10^3 N/deg. were used in Eq. (3.10) in order to obtain the dynamic modeled body sideslip β_{DYN} and yaw rate ω_{DYN}. As stated before, constant cornering stiffness are a typical approach for on-road vehicles but might not the right approach in agricultural applications.

Fig. 3.10 shows in purple the body sideslip β_{REG} and yaw rate ω_{REG} obtained from the modeled regression equations in comparison to the measured body sideslip β and the measured yaw rate ω shown in blue. The dynamic modeled body sideslip β_{DYN} and yaw rate ω_{DYN} are shown in red. It can be seen how the behavior of each data is consistent to the measured data.

It is necessary to calculate the error in order to evaluate the accuracy improvement of the body sideslip β_{REG} and yaw rate ω_{REG} obtained from the regression model. The RMS error for both the body sideslip β_{REG} and yaw rate ω_{REG} is also shown in Fig. 3.10. The RMS error for both the body sideslip β_{DYN} and yaw rate ω_{DYN} is included as well.

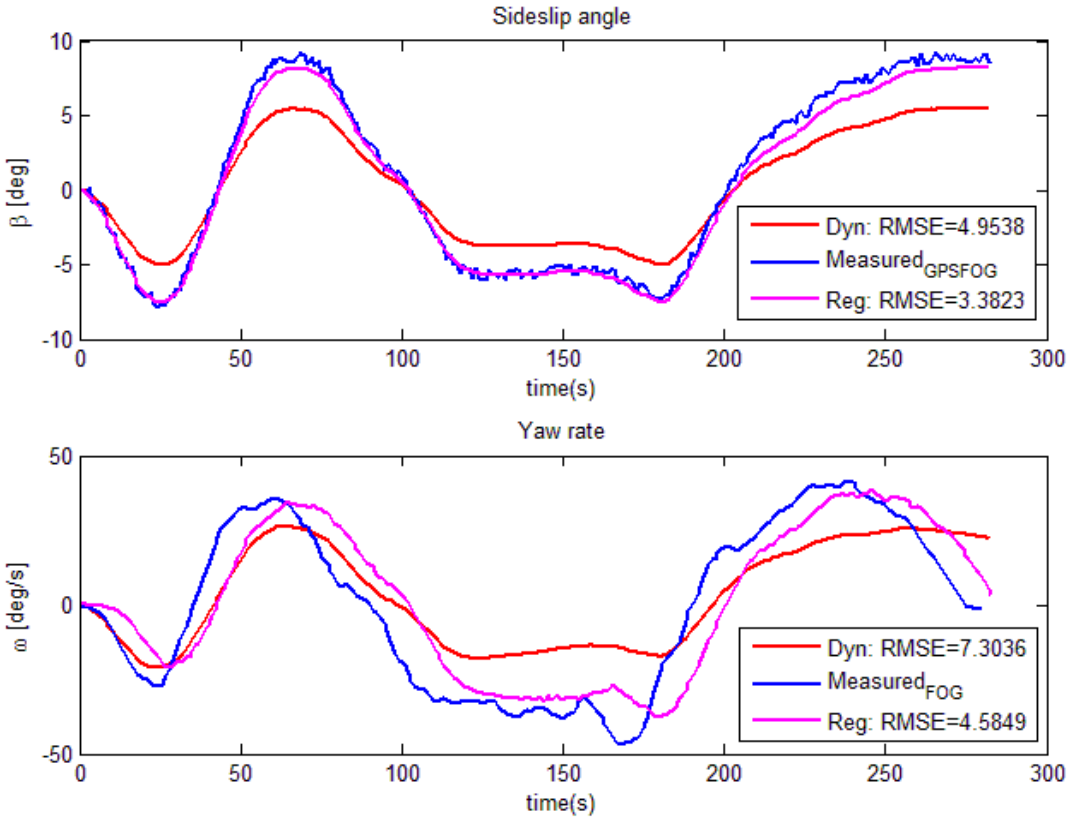


Fig. 3.10. Body sideslip β and yaw rate ω comparison.

Table 3.2 summarizes the calculated RMS errors of the body sideslip β and the yaw rate ω obtained from the dynamic model and the regression model.

Table 3.2. Summary of RMS errors.

Model type	ω RMSE (deg./s)	β RMSE (deg.)
Dynamic	7.30	4.95
Regression	4.58	3.38

From Table 3.2, it can be noted how using the values obtained from the regression model, the accuracy improved for both the sideslip β and the yaw ω rate in contrast with the dynamic model. Smaller error values might be obtained applying more sophisticated filtering techniques to the measured data; especially to the sideslip β data set, due to its characteristic noise inherited from to subtraction of the GPS and FOG data sets. Moreover, measured data accuracies might be affected by the inherent error caused by the mechanical installation of the sensors. Despite all, this methodology has proven to be a

correct approach to show that the measurements performed can be used to estimate the tires parameters, and also has proven to account for the sliding nonlinear behavior in the vehicle dynamics.

3.3.2 Improvement of the estimations of the vehicle kinematic model

Following section 3.3.1, it is possible to estimate both the vehicle's body sideslip β_{REG} and the yaw rate ω_{REG} with improved accuracy thanks to the regression model. The vehicle's body sideslip β_{REG} and the yaw rate ω_{REG} can in turn be used to improve the navigation model. Several researches (Pepy et al., 2006) have compared the performance of a pure kinematic model and a dynamic kinematic model, concluding that a pure kinematic model has an overall poor performance for most conditions. Despite having its own restrictions; like an inherent error caused by numerical integration, the dynamic kinematic model provides better position estimations for most experimental conditions, even in experiments with big tractors (Bevly et al., 2002).

It is possible to estimate the dynamic kinematic model improvement by using the data summarized in Fig. 3.10; this means to compute the body sideslip β_{REG} and yaw rate ω_{REG} into Eq. (3.1) to obtain the position and heading of the vehicle estimated from the regression model. Similarly, it is possible to compute the body sideslip β_{DYN} and yaw rate ω_{DYN} into Eq. (3.1) to obtain the position and heading of the vehicle estimated from the dynamic model with constant cornering stiffness.

Fig. 3.11 shows in purple the heading φ_{REG} obtained by computing the body sideslip β_{REG} and yaw rate ω_{REG} into Eq. (3.1). The measured heading φ is shown in blue for comparison purposes. The dynamic modeled heading φ_{DYN} is shown in red. Again, it is evident how the behavior of each data set is consistent to the measured data.

It is necessary to calculate the error in order to evaluate the accuracy improvement of the navigation model. The RMS error for both headings φ_{REG} and φ_{DYN} respect to the measured data are also shown in Fig. 3.11. It can be noted how using the values obtained from the regression model, the accuracy improved for the heading φ_{REG} in contrast with the dynamic model φ_{DYN}.

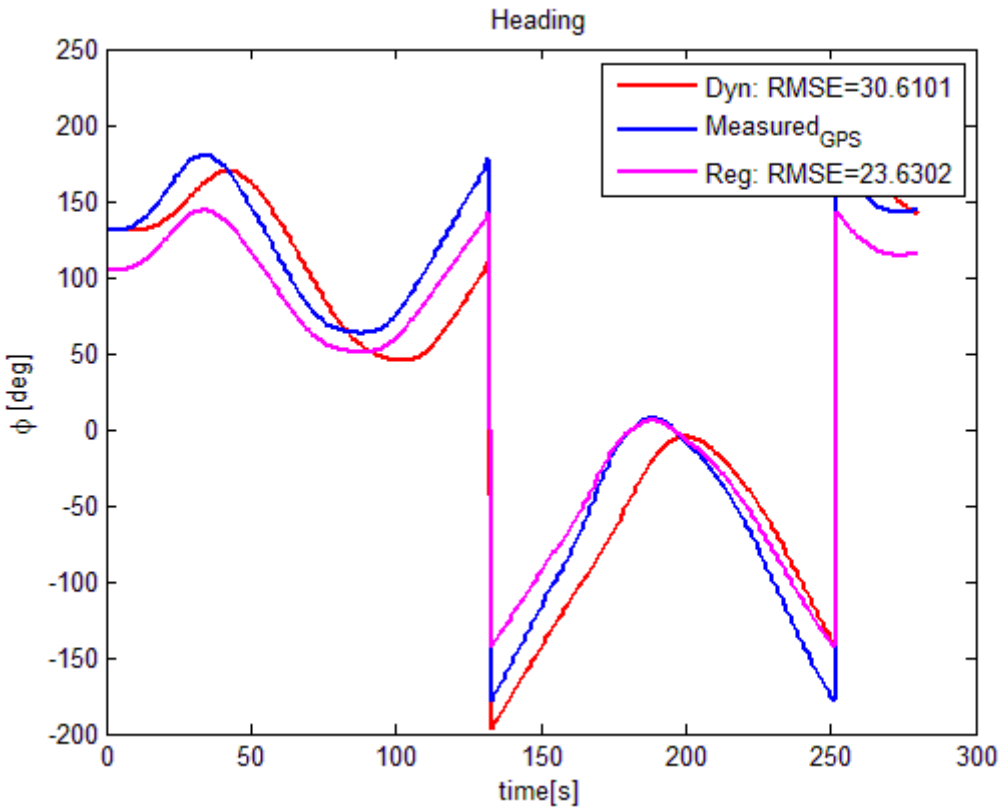


Fig. 3.11. Heading comparison.

Fig. 3.12 shows in purple the position of the vehicle obtained by computing the body sideslip β_{REG} and yaw rate ω_{REG} into Eq. (3.1). Note that Fig. 3.12 is plotted in Easting and Northing coordinates. The measured position is shown in blue for comparison purposes. The dynamic modeled position is shown in red. Again, it can be observed how the behavior of each data set is consistent to the measured data. The RMS error for both the regression modeled position and the dynamic modeled position respect to the measured data are also shown in Fig. 3.12. It can be noted how using the values obtained from the regression model, the accuracy improved for the regression modeled position in contrast with the dynamic modeled position.

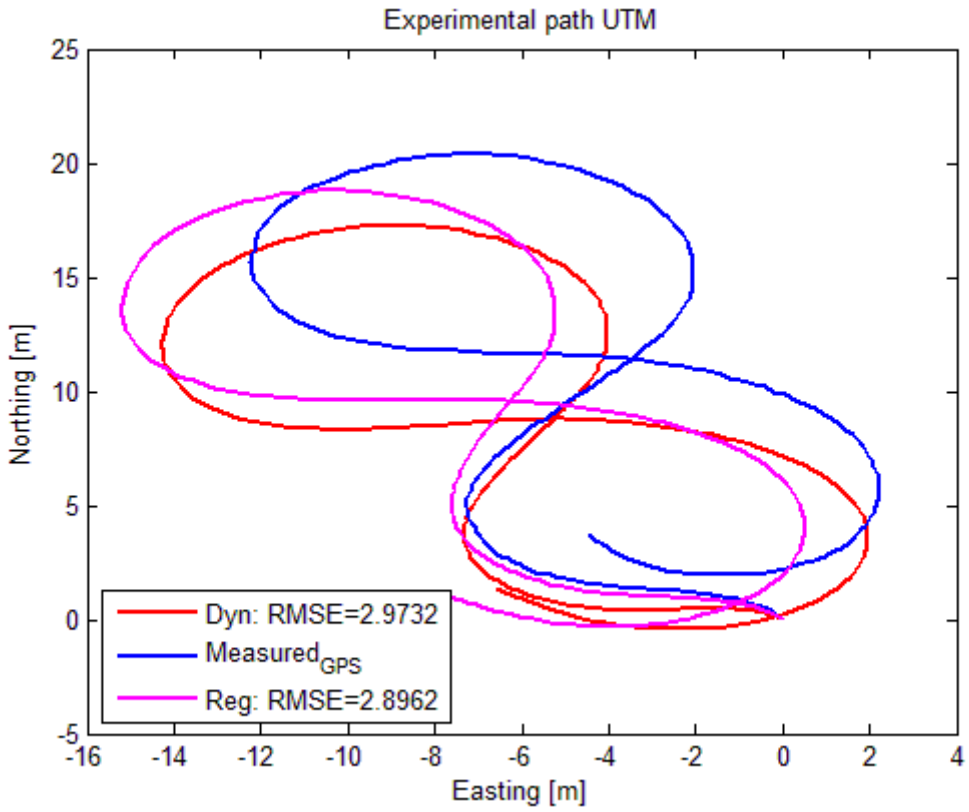


Fig. 3.12. Navigation path comparison.

From Figs. 3.11 and 3.12, it can be observed how by using the data obtained from the regression model the accuracy improved for both the heading and the position of the vehicle. This result was expected due to the fact that the accuracy improved for both the sideslip β and the yaw ω rate in contrast with the dynamic model as described in section 3.3.1. As stated before, one important drawback of the dynamic navigation model described by Eq. (3.1) is its inherent integration error; however, this does not interfere with the purpose of comparing the results obtained by computing the body sideslip β_{REG} and yaw rate ω_{REG} with the results obtained by computing the body sideslip β_{DYN} and yaw rate ω_{DYN} because the same Eq. (3.1) is being used; only the input data sets are different.

In conclusion, smaller RMS errors mean that tire parameters can be used to obtain better estimations of the body sideslip β_{REG} and yaw rate ω_{REG}; which in turn can be used to improve the navigation model of the test vehicle.

3.4 Conclusions

This chapter has described the theory framework used in this research. It also presented a brief description of the most common vehicle control strategies. The description of the tire mechanics was used to explain the vehicle dynamics. The vehicle handling was described using the bicycle dynamic model. These experimental results have shown the effectiveness of a measurement methodology for estimating a vehicle's dynamic parameters β and ω using an RTK-GPS, and an FOG. The comparison between the data measured experimentally and the data yield by the models validates the effectiveness of this measurement methodology. The data measured experimentally have verified that this measurement methodology can be applied to off-road vehicles used in agriculture to estimate the tire dynamic properties like the tire slip angle, the lateral forces and the tire cornering stiffness.

A vehicle tire model that does not depend on several parameters, such as longitudinal slip, sideslip angle, normal load, camber angle, tire pressure, wear, and road surface characteristics was development for the experimental vehicle. Such model was obtained using a regression model technique consisting on applying the method of least squares to the data obtained experimentally; in order to account for the nonlinear relationship between the tire slip angle and the lateral force observed as a hysteresis loop.

The vehicle's dynamic parameters β and ω were computed into the dynamic navigation model to obtain estimations of the heading and position of the vehicle. The tire dynamic properties described by a regression model technique allowed to obtain better estimations of the vehicle motion model, reducing its RMS error by 30%. However, these estimations alone are not accurate enough to guide the vehicle through a field because there is a wide range of environment factors present in the tire-soil interaction; like soil moisture and cone index, which change from field to field and cannot be predicted by the vehicle motion model. Therefore, it is necessary to integrate the vehicle motion model estimations with some sensing method such as machine vision.

Chapter 4 Image processing algorithm development of a machine vision with both wide-angle and telephoto images

Computer technologies have been shown to improve agricultural productivity in a number of ways. Image processing has emerged as a powerful tool among those computer technologies. Several image processing techniques have been used to assist researchers and farmers to improve agricultural practices (Saxena and Armstrong, 2014). As a result, image processing has been used in different agricultural industry contexts to assist with precision agriculture practices like weed and herbicide technologies, plant growth monitoring and plant nutrition management, and agricultural machinery autonomous navigation. This optimize agricultural vehicle systems which ultimately results in the higher levels of quality and refinement that farmers have come to expect.

This chapter presents the image processing algorithms used in this research. The algorithms are briefly described with examples.

4.1 Image processing in the agricultural context

Image processing techniques can be used to enhance agricultural practices, by improving accuracy and consistency of processes while reducing farmers' manual monitoring. Often, it offers flexibility and effectively substitutes the farmers' visual decision making. Table 4.1 summarizes some of the image processing terminologies applicable in agricultural practices (Saxena and Armstrong, 2014).

Table 4.1. Image processing terminology applicable in agriculture.

Image processing term	Description
Image acquisition	Process of retrieving a digital image from a physical source capture using image sensors
Gray scale conversion	Process of converting a color or multi-channel digital image to a single channel where image pixel possess a single intensity value
Image background extraction	Separation of image background, retrieving foreground objects
Image enhancement	Improvement in perception of image details for human and machine analysis
Image histogram analysis	Pixel plot analysis in terms of peaks and valleys formed by pixel frequency vs pixel intensities
Binary image segmentation	Foreground objects separation from background in a binary (black-and-white) image

Color image segmentation	Image objects separation in a color image, regions of interests
Image filtering	Process of distorting an image in a desired way using a filter
Feature extraction	Process of defining a set of features, or image characteristics that efficiently or meaningfully represent the information important for analysis and classification
Image registration	Process of transforming different sets of data into one coordinate system
Image transition	Process of changing state or defining a condition between two or more images
Image object detection	Process of finding instances of real-world objects such as weeds, plants, and insects in images or video sequences
Image object analysis	Process extracting reliable and meaningful information from images

Computer-vision applied to Precision Agriculture by using Image Processing techniques typically involves five basic processes such as image acquisition, preprocessing, segmentation, object detection and classification. At first, these terms and all the terminology summarized in Table 4.1 might sound confusing for the inexpert. This chapter, however does not pretend to be a detailed Image Processing programming guide; its scope is to describe the Image Processing algorithms applied in this research.

4.2 Open CV

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision (Brahmbhatt, 2013). Originally developed by Intel, it was later supported by Willow Garage and is now maintained by Itseez. The library is cross-platform and free for use under the open-source BSD license. OpenCV supports the deep learning frameworks TensorFlow, Torch/PyTorch and Caffe.

Officially launched in 1999, the OpenCV project was initially an Intel Research initiative to advance CPU-intensive applications, part of a series of projects including real-time ray tracing and 3D display walls. The main contributors to the project included a number of optimization experts in Intel Russia, as well as Intel's Performance Library Team. In the early days of OpenCV, the goals of the project were described as:

- Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel.

- Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.

- Advance vision-based commercial applications by making portable, performance-optimized code available for free – with a license that did not require code to be open or free itself.

The first alpha version of OpenCV was released to the public at the IEEE Conference on Computer Vision and Pattern Recognition in 2000, and five betas were released between 2001 and 2005. The first 1.0 version was released in 2006. A version 1.1 "pre-release" was released in October 2008. The second major release of the OpenCV was in October 2009. OpenCV 2 includes major changes to the C++ interface, aiming at easier, more type-safe patterns, new functions, and better implementations for existing ones in terms of performance (especially on multi-core systems). Official releases now occur every six months and development is now done by an independent Russian team supported by commercial corporations. In August 2012, support for OpenCV was taken over by a non-profit foundation OpenCV.org, which maintains a developer and user site. OpenCV's application areas include:

- 2D and 3D feature toolkits.
- Egomotion estimation.
- Facial recognition system.
- Gesture recognition.
- Human–computer interaction (HCI).
- Mobile robotics.
- Motion understanding.
- Object identification.
- Segmentation and recognition.
- Stereopsis stereo vision: depth perception from 2 cameras.
- Structure from motion (SFM).
- Motion tracking.
- Augmented reality.

To support some of the above areas, OpenCV includes a statistical machine learning library that contains:

- Boosting.
- Decision tree learning.
- Gradient boosting trees.
- Expectation-maximization algorithm.
- k-nearest neighbor algorithm.
- Naive Bayes classifier.
- Artificial neural networks.
- Random forest.
- Support vector machine (SVM).
- Deep neural networks (DNN).

OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. There are bindings in Python, Java and MATLAB/OCTAVE. The API for these interfaces can be found in the online documentation. Wrappers in other languages such as C#, Perl, Ch, Haskell and Ruby have been developed to encourage adoption by a wider audience. All of the new developments and algorithms in OpenCV are now developed in the C++ interface.

OpenCV runs on the following desktop operating systems: Windows, Linux, macOS, FreeBSD, NetBSD, OpenBSD. OpenCV can also run on the following mobile operating systems: Android, iOS, Maemo, BlackBerry 10. The user can get official releases from SourceForge or take the latest sources from GitHub. OpenCV uses CMake for managing the build process of software using a compiler-independent method. This research implemented the OpenCV library in Microsoft Visual Studio 2013 Community running on Windows 7 and Windows 8.

4.3 Image acquisition and RGB color space

There are multiple ways to acquire digital images from the real world: digital cameras, scanners, computed tomography, and magnetic resonance imaging to name a few. In every case what we humans

see are images. However, when transforming this to our digital devices what we record are numerical values for each of the points of the image.

A digital image is nothing more than a 2D matrix containing all the intensity values of its pixel points. Each pixel point position in the image matrix is identified by a pair of Cartesian coordinates *p(x,y)*. How to get and store the pixels values may vary according to the needs, but in the end all images inside a computer world may be reduced to numerical matrices and other information describing the matrix itself. OpenCV is a computer vision library whose main focus is to process and manipulate this matrix information. Fig. 4.1 illustrates this scheme.

Fig. 4.1. Image acquisition scheme.

To acquire digital images is equivalent to store the pixel values; i.e. store the value of a pixel in the position *p(x,y)*. It is possible to select the color space and the data type used. The color space refers to the combination of color components in order to code a given color. The simplest one is the gray scale where the only colors at available are black and white. The combination of these two make it possible to create many shades of gray.

For colorful ways there are a lot more methods to choose from. Each of them breaks it down to three or four basic components and it is possible to use the combination of these to create the others components. The most popular color space is RGB, mainly because this is also how the human eye builds up colors. Its base colors are Red, Green and Blue. Each of the color components has its own valid domains; which makes it necessary to define the data type used. The smallest data type possible is char, which means one byte or 8 bits. This may be unsigned (containing values from 0 to 255) or

signed (values from -127 to +127). In the case of three components such as RGB, 8 bits data type gives as a result 16 million representable colors. It is possible to acquire an even finer control by using the float (4 byte = 32 bits) or double (8 byte = 64 bits) data types for each one of the RGB components. However, increasing the size of a color component also increases the size of the whole picture in the memory. In summary, to represent color images in the RGB color space, separate Red, Green and Blue components must be specified for each pixel *p(x,y)*, and so the *pixel value* is actually a vector of three numbers. This means that each pixel at the image position *p(x,y)* has a pixel value (R, G, B). Often the three different components are stored as three separate *grayscale* images known as color planes (one for each of Red, Green and Blue), which have to be recombined when displaying or processing. Due to its 3 components, the RGB color space can be represented as a three-dimensional volume described by treating the component values as ordinary Cartesian coordinates in a Euclidean space. This representation is shown in Fig. 4.2.

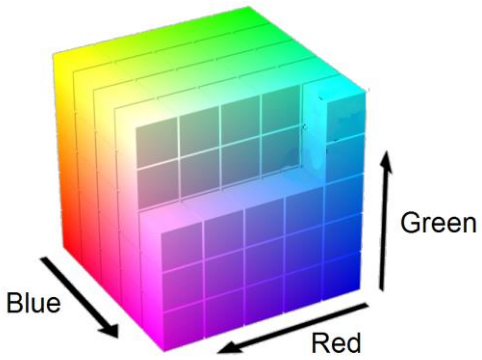


Fig. 4.2. RGB color space mapped to a cube.

For the RGB color space, the cube shown in Fig. 4.2 uses non-negative values within a 0-255 range, assigning black to the origin at the vertex (0, 0, 0), and with increasing intensity values running along the three axes up to white at the vertex (255, 255, 255), diagonally opposite black. An RGB triplet (R, G, B) represents the three-dimensional coordinate of the given color within the cube or its faces or along its edges. This approach allows computations of the color similarity of two given RGB colors by simply calculating the distance between them: the shorter the distance, the higher the similarity.

4.4 HSV color space

The Hue, Saturation, Value (HSV) color space is an alternative representation of the RGB color space, designed in the 1970s by computer graphics researchers to more closely align with the way human vision perceives color-making attributes (Agoston, 2005). In this color space, colors of each hue are arranged in a radial slice, around a central axis of neutral colors which ranges from black at the bottom to white at the top. This representation is shown in Fig. 4.3. The HSV color space models the way liquid paints of different colors mix together, with the saturation dimension resembling various shades of brightly colored paint, and the value dimension resembling the mixture of those paints with varying amounts of black or white paint.

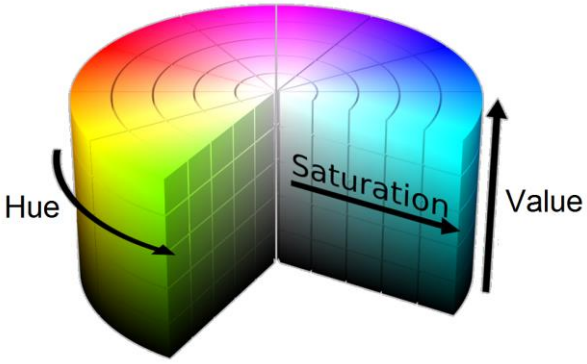


Fig. 4.3. HSV color space mapped to a cylinder.

In this research, the HSV color space was chosen for color filtering instead of the RGB color space because it provided a better contrast for different illumination conditions. Unlike the RGB color space, the HSV color space separates luma, or the image intensity, from chroma or the color information. Therefore, the HSV color space filtering provided a better contrast because the HSV color space uses only one channel to describe color (H), making the filtering very intuitive to specify the crops color in contrast to the soil.

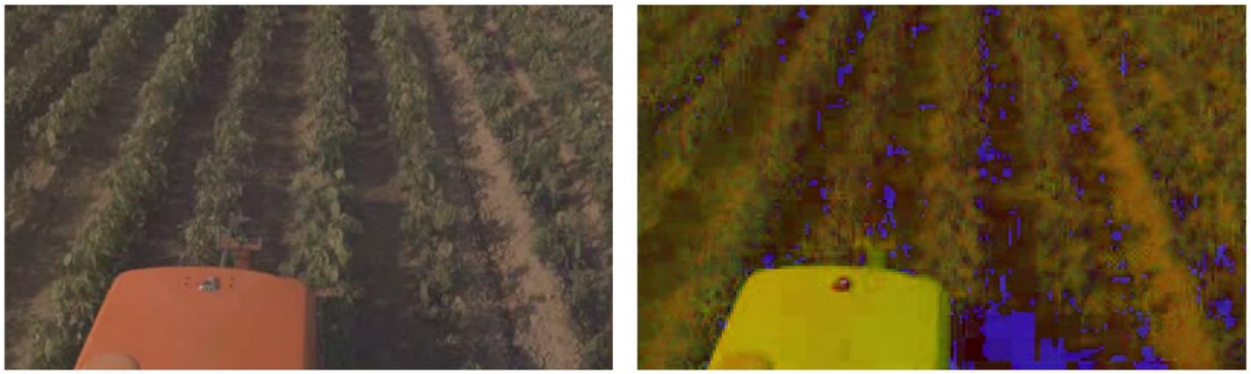
To convert an RGB image into an HSV image let *MAX* be the maximum value of the components RGB and *MIN* the minimum value of those same values; then, the components of the color HSV space can be calculated (Agoston, 2005) according to Eq. (4.1):

$$H = \begin{cases} undefined, & if\ MAX = MIN \\ 60 \times \frac{G-B}{MAX-MIN} + 0, & if\ MAX = R\ and\ G \geq B \\ 60 \times \frac{G-B}{MAX-MIN} + 360, & if\ MAX = R\ and\ G < B \\ 60 \times \frac{B-R}{MAX-MIN} + 120, & if\ MAX = G \\ 60 \times \frac{R-G}{MAX-MIN} + 240, & if\ MAX = B \end{cases} \tag{4.1}$$

$$S = \begin{cases} 0, & if\ MAX = 0 \\ 1 - \frac{MIN}{MAX}, & otherwise \end{cases}$$

$$V = MAX$$

Fig. 4.4 illustrates the result of the RGB to HSV conversion of a field sample image. Fig. 4.4 a) shows the input image in the RGB color space. The image shows crop rows vegetation and inter crop row soil from the point of view of the tractor's cabin. A portion of the tractor's capot is also shown. Several crops shadows are also visible. The output image in the HSV color space is shown in Fig. 4.4 b). Note that the resulting HSV color space output image shown in Fig. 4.4 b) does not show useful information by itself since the crop rows are not visible emphasized in contrast to the inter-row space; but it does provides the HSV values necessary to separate the crop rows from the rest of the image.



a) b)

Fig. 4.4. RGB to HSV color space conversion. a) RGB input image. b) HSV output image.

The HSV color space was developed in the late 1970s, for the hardware that was available at the time, and was used in most mid-end computers in the 1990s. Its design sacrificed perceptual relevance for computation speed which made it less demanding on systems. Therefore, it also has some general disadvantages; for example, saturation and lightness are confounded, so a saturation scale may also contain a wide range of lightness's. This means that it may progress from white to green which is a

combination of both lightness and saturation. Likewise, Hue and lightness are confounded so a saturated yellow and a saturated blue may be designated as the same *lightness;* but in reality they have wide differences in perceived lightness.

4.5 Binary threshold

The Binary threshold is performed by means of a simple segmentation method. Segmentation methods are useful to separate out regions of an image corresponding to objects of interest; in the case of this research, the objects of interest are the crop rows. This separation is based on the variation of intensity between the object pixels and the background pixels. To differentiate the pixels corresponding to the objects of interest from the rest, it is possible to perform a comparison of each pixel intensity value with respect to a threshold (Bradski, and Kaehler, 2008). This threshold value is determined according to the problem to solve. In the case of this research, a threshold value is required for each one of the HSV color space components. Once the important pixels have been identified and separated properly, it is possible to set them with a determined value to identify them. This means, assign them a value of (0, 0, 0) corresponding to black, or (255, 255, 255) corresponding to white.

To illustrate how thresholding process work consider a source image in the HSV color space; i.e., each pixel from the source image *src(x,y)* has intensity values (H, S, V). The HSV intensity values can be obtained from the RGB values by using Eq. (4.1).For explanation purposes, consider only one of the three HSV components. For example, choose the Hue value and plot its distribution for all the pixels in the source image *Hsrc(x,y)*, as shown in Fig 4.5 a). The horizontal red line represents the threshold value fixed for the Hue. The binary thresholding operation can be expressed by Eq. (4.2) as:

$$Hdst(x,y) = \begin{cases} 255, & if\ Hsrc(x,y) > threshold \\ 0, & otherwise \end{cases} \tag{4.2}$$

Therefore, if the Hue intensity of the pixel *Hsrc (x,y)* in the source image is higher than the threshold, then the new pixel intensity is set to 255 (white). Otherwise, the pixels are set to 0 (black). This means that the resulting image; the destiny image, is a binary image because it contains only 2 colors, black or white. The plot of the Hue value distribution for all the pixels in the destiny image *Hdst(x,y)*, is shown in Fig 4.5 b).

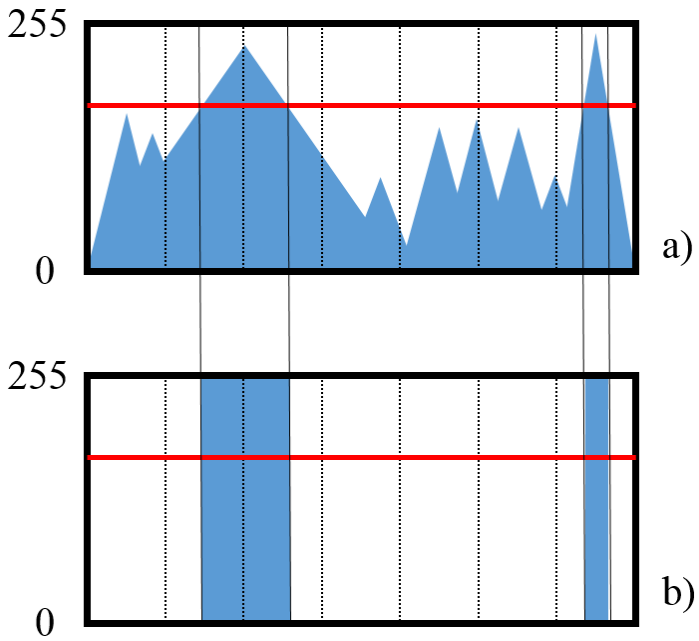


Fig. 4.5. Binary threshold representation. a) Hue input distribution *Hsrc*. b) Hue output distribution *Hdst*.

Note that the binary threshold operation has to be performed on each one of the three components of the HSV color space; Eq. (4.2) and Fig. 4.5 show the situation for only one component for explanation purposes. In the case of this research, the threshold value is determined by means of fine tuning according to the HSV values present in the crop rows. Fig 4.6 shows the result of the Binary threshold operation.

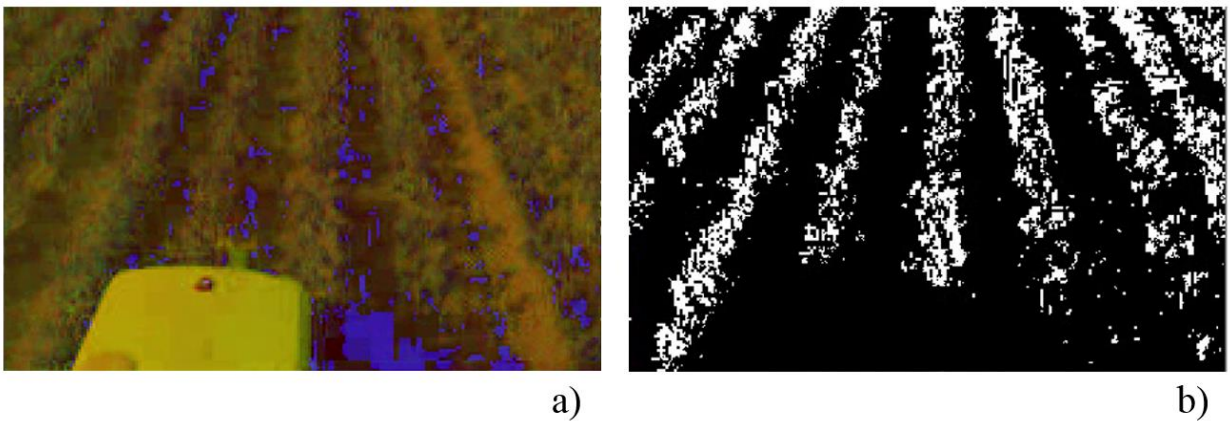


Fig. 4.6. HSV to binary conversion. a) HSV input image. b) Binary output image.

Fig. 4.6 a) shows the input image in the HSV color space. As mentioned before, in the HSV color space image the crop rows are not visible emphasized in contrast to the inter-row space; but it does provides the HSV values necessary to estimate their respective thresholds and separate the crop rows from the rest of the image. Fig. 4.6 b) shows the resulting binary image; it is evident how the crops rows have an overall good contrast to other elements like shadows and soil. However, due to the shape of each leaf of the plants, the vegetation appears as disparate elements in the binary image making the crops rows look noisy. This noise makes the center line from the crop row difficult to detect (Ji and Qi, 2011).

4.6 Morphological operations

Morphological operations are defined as a set of operations that process images based on shapes; a structuring element is applied to an input image to generate a smoothed output image. The most basic morphological operations are Erosion and Dilation. These two operations can be combined to obtain a wide range of applications in the image, like:

- Removing noise.
- Isolation of individual elements.
- Fusion of disparate elements.
- Detection of intensity bumps or holes.

A brief description of each operation is given in the lines below. A more detailed explanation can be found at (Gonzalez and Woods, 2002).

4.6.1 Dilation

The Dilation operator takes two pieces of data as inputs (Fisher et al., 2003). The first is the image which is to be dilated. The second is a usually small set of coordinate points, known as a structuring element. This structuring element that determines the precise effect of the Dilation on the input image. As an example of Binary Dilation, suppose that the structuring element is a 3×3 square, with the origin at its center, as shown in Fig. 4.7. The 3×3 square is probably the most common structuring element used in Morphological operations, but others can be used. Also, as shown in Fig. 4.6 b) in the input image the foreground pixels are represented by 1 (white) and background pixels by 0 (black). To

compute the Dilation of a Binary input image by a 3×3 square structuring element, consider each of the background pixels in the input image. For each background pixel of the input image (the input pixel), superimpose the structuring element on top of the input image so that the origin of the structuring element coincides with the input pixel position. If at least one pixel in the structuring element coincides with a foreground pixel in the image underneath, then the input pixel is set to the foreground value. If all the corresponding pixels in the image are background, however, the input pixel is left at the background value. The effect of the Dilation operation using a 3×3 square structuring element on a binary image is shown in Figure 4.7.

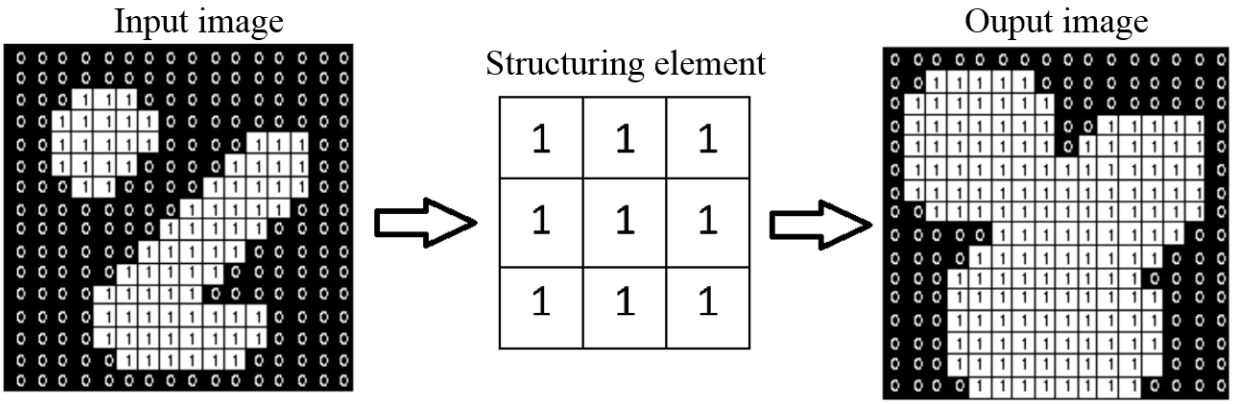


Fig. 4.7. Effect of Dilation using a 3×3 square structuring element.

4.6.2 Erosion

The Erosion operator also takes an input image which is to be eroded and a structuring element as inputs (Fisher et al., 2003). Similar to the Dilation operation, consider a 3×3 square structuring element and each of the foreground pixels in the input image. For each foreground pixel; the input pixel, superimpose the structuring element on top of the input image so that the origin of the structuring element coincides with the input pixel coordinates. If for every pixel in the structuring element, the corresponding pixel in the image underneath is a foreground pixel, then the input pixel is left as it is. If any of the corresponding pixels in the image are background, however, the input pixel is also set to background value. The effect of this operation is to remove any foreground pixel that is not completely surrounded by other white pixels. Such pixels must lie at the edges of white regions, and so the practical upshot is that foreground regions shrink and holes inside a region grow. The effect of the Erosion operation using a 3×3 square structuring element on a binary image is shown in Figure 4.8.

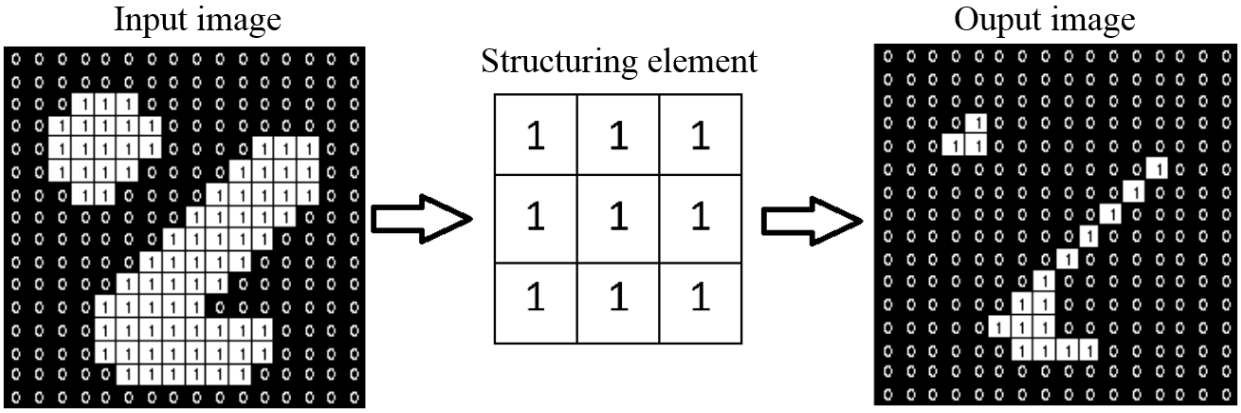


Fig. 4.8. Effect of Erosion using a 3×3 square structuring element.

Therefore, by combining both Erosion and Dilation operations it is possible to smooth the noise of the crop rows shown in Fig. 4.6 b). The erosion operation eliminates small isolated white pixel areas. The dilation operation merges the areas where large groups of white pixels are found next to each other. The erosion operation uses a rectangular shape structuring element of 11 x 22 pixels. The dilation operation makes white crop rows denser and eliminates breaks using a rectangular shape structuring element of 7 x 14 pixels. These morphological operations transform the binary image shown in Fig. 4.9 a) into the smoothed result shown in Fig. 4.9 b). It can be seen how the small white pixel areas within the inter-row space have been eliminated and the black pixels in the crops rows have been replaced by white pixels producing a continuous white area. Note that the central five rows were selected to remain in Fig 4.9 b), discarding the four outside rows. This is because the central five rows are present in every frame and they can be seen with good resolution after binary conversion.

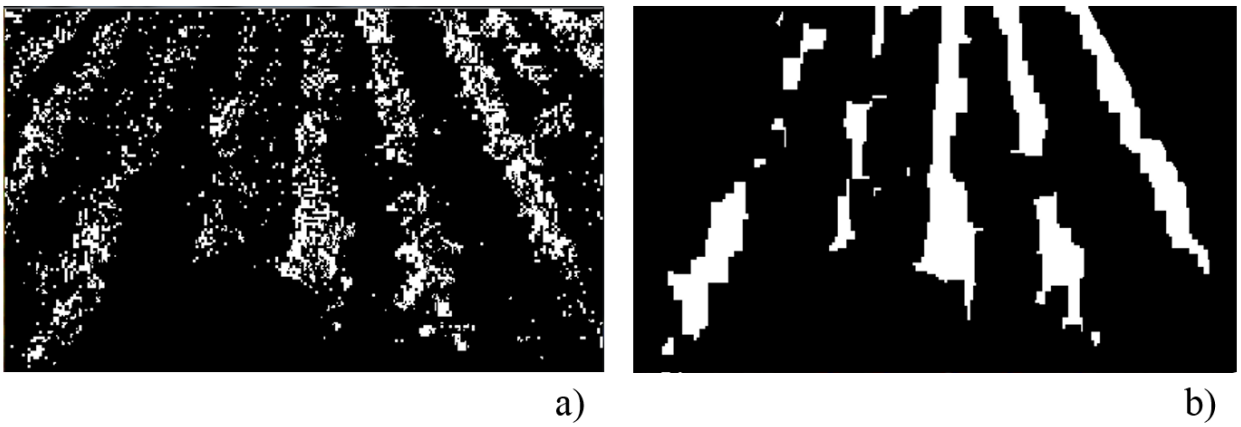


Fig. 4.9. Morphological operations result. a) Binary input image. b) Smoothed output image.

A common approach used to detect the crop rows is to thin down the white pixels area using Erosion operations. The crop rows represented by the white areas in the smoothed binary image shown in Fig. 4.9 b) could be thinned down until they look like a line, in order to detect the contours of the thinned crop rows. Then it is possible to apply the Hough transform (Rovira-Más et al., 2005) to obtain a straight line representing the crop row. However, since the computational load of the Hough transformation method is quite high (Søgaard and Olsen, 2003), this research used a least squares method based on the geometric centers of the crop rows instead of the Hough transformation.

4.7 Contours and geometric centers

The geometric centers of the crop rows can be detected thanks to the fact that dispersed white pixels shown in Fig. 4.9 a) have been connected in the output image giving as a result the wider crop row areas shown in Fig. 4.9 b). To find the geometric centers of the crop rows, it is necessary to find the horizontal crop segments. The smoothed binary image shown in Fig.4.9 b) has a size of 420 x 256 pixels; it is divided into 16 horizontal strips of 420x16 pixels, where maximum white values indicate the presence of a candidate row. The 16 horizontal strips are separated by the green lines, as shown in Fig. 4.10 a); it can be seen how each horizontal strip contains a segment of each crop row represented as a white square.

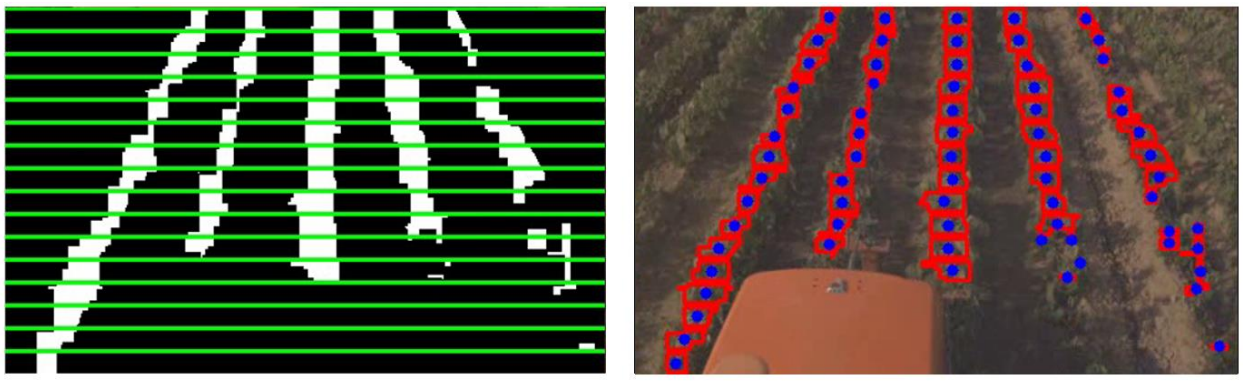


Fig. 4.10. Morphological operations result. a) Segments of each crop row. b) Geometric centers.

It is possible to retrieve the contour of each white square from the binary image shown in Fig. 4.10 a) using the algorithm of topological structural analysis by border following (Suzuki, and Abe, 1985).

The resulting contours are shown in red in Fig. 4.10 b). These red contours represent the white squares obtained from each one of the 16 horizontal strips; although the white areas are not perfect squares it is guaranteed that they are inside the crop row region. The mass centers of each one of these red contours were calculated using an algorithm included in the OpenCV library; which is based on the Green's formula (Green, 1828). Fig. 4.10 b) also shows the result of each block mass center; the blue points represent their respective geometric centers. Fig. 4.10 b) also shows that the size of the red contours might vary; however, since their geometric centers are contained inside the crop row they can be used to calculate the crop's center line anyway.

These centers are distributed over the five central crop rows, so the next step is to fit each line to each set of the point's coordinates by using the least squares method. Since least squares method is a well-known procedure its explanation is not included in this thesis. The least squares method results in the estimates of the slope and intercept of the center line of the guiding row as shown in Fig. 4.11.

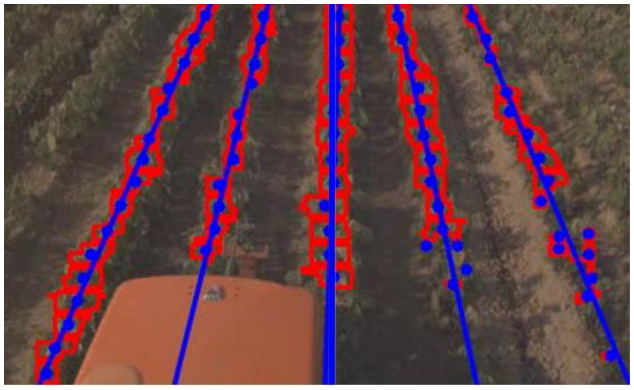


Fig. 4.11. Lines from least squares regression.

Some factors might affect the accuracy of this method (Søgaard and Olsen, 2003). For example, if weeds grow beside the crop row, the points in the image strip containing the weed will be offset from their correct position. Moreover, due to the natural variation in the plant growth, the width and position of the canopy make it difficult to determine the center line of a crop row. Therefore, accuracy of the method can be improved by obtaining correction data from the Telephoto image as described in the next section.

4.8 Centers of gravity

In order to improve the accuracy of the crop row detection method it is possible to obtain correction data from the Telephoto image of the 2-in-1 camera. Fig. 4.12 a) shows the Telephoto image capture of the central crop row; thanks to its increased resolution there is a bigger contrast between crops and the soil, to the point that some of the leaves veins can be identified in the image.

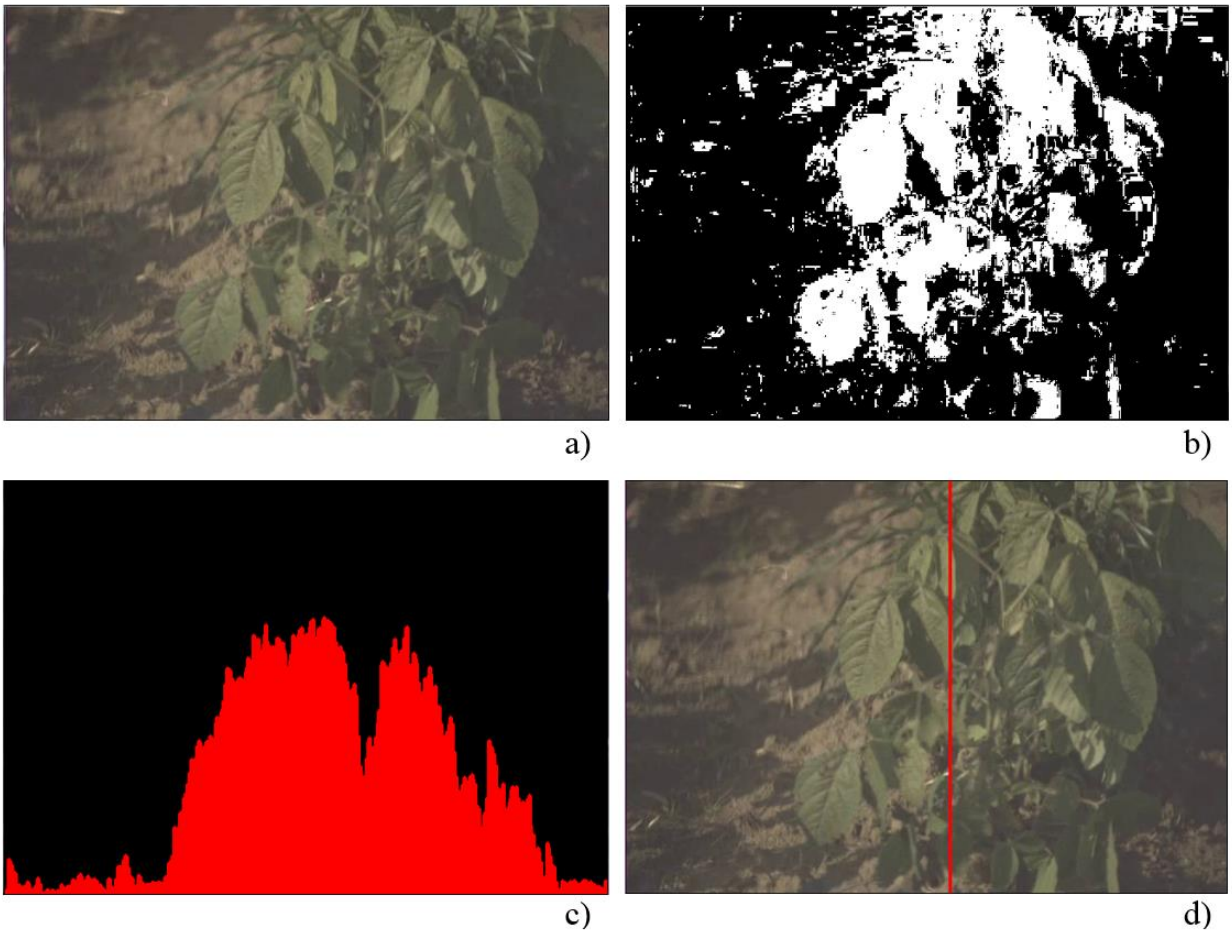


Fig. 4.12. Telephoto image crop row detection. a) Input image. b) Binary image. c) Vertical sum of white values. d) Line from the center of gravity.

The same HSV color space conversion used in the Wide-angle image was used for color filtering of the Telephoto image as well, giving as result the binary image shown in Fig. 4.12 b). The advantage of using the Telephoto image becomes obvious by looking at Fig. 4.12 b); where the leaves' silhouettes are easily differentiated as white pixels from the rest of the elements in the image. Since the central crop row does not look noisy and the leaves are well defined, it is not necessary to apply the morphological operations of dilation and erosion described in sections 4.6.1 and 4.6.2 respectively.

The Telephoto binary image shown in Fig. 4.12 b) is then processed in order to find the crop's center line. Since the plants are represented as white pixels, the white intensities of the image are calculated by vertical integration (Facciolo et al., 2014). This means to calculate the vertical sum of the white pixel values in the image column by column. Since the Telephoto image has a size of 640 x 480 pixels, it is necessary to sum the 480 rows of each one of the 640 columns of the image. This gives as a result the white pixels intensity distribution with the form of a histogram shown by the red plot in Fig. 4.12 c). This white pixels intensity distribution contains a total of 640 elements, which can be stored in a vector. The crop's center line could be found from the distribution maximum value or by fitting a sinusoidal curve by means of a least squares fit (Olsen, 1995). However, the center of the crop row was obtained by calculating the center of gravity of the white pixels intensity distribution because it gave as a result a more stable value. To calculate the center of gravity, the histogram shown by the red plot in Fig. 4.12 is arranged on a unit circle evenly distributed, as illustrated by the cut cylinder in Fig. 4.13.

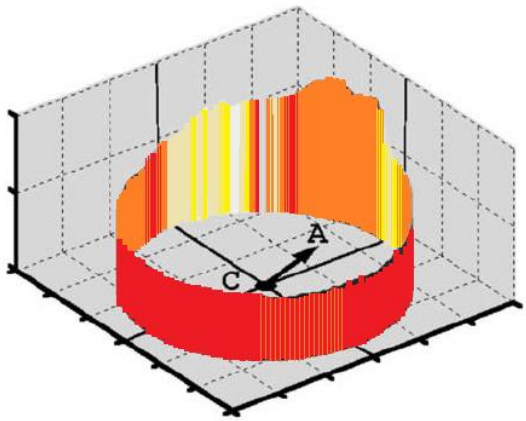


Fig. 4.13. Circular representation of the sum curve and determination of the center of gravity.

As the number of elements in the histogram is m=640, the angular distance between two consecutive elements on the circle will be $2\pi/m$, and in a plane rectangular co-ordinate system with origin in the circle's center, the position, (x_l, y_l), of each element l (l=1, 2, … , m) can be defined by Eq. (4.3) as:

$$x_l = \cos\left(2\pi \frac{l-1/2}{m}\right) \quad y_l = \sin\left(2\pi \frac{l-1/2}{m}\right) \quad l = 1, \dots, m \tag{4.3}$$

Now, regarding the elements of the distribution as point masses with masses W_l , (i.e. W_l is the numerical value of each one of the elements of the distribution) the center of gravity A (x_A, y_A) of the circular arrangement shown in Fig. 4.13 will be given by Eq. (4.4) as:

$$x_A = \frac{\sum_{l=1}^{m} W_l x_l}{\sum_{l=1}^{m} W_l} \quad y_A = \frac{\sum_{l=1}^{m} W_l y_l}{\sum_{l=1}^{m} W_l} \tag{4.4}$$

After obtaining the center of gravity A (x_A, y_A) it is possible to calculate the direction angle φ ($0 \leq \varphi < 2\pi$) of the vector CA from the circle center C to the center of gravity A shown in Fig. 4.13 by using Eq. (4.5) as:

$$\varphi = \begin{cases} \tan^{-1}\frac{y_A}{x_A} & if\ x_A > 0\ and\ y_A \geq 0 \\ 2\pi + \tan^{-1}\frac{y_A}{x_A} & if\ x_A > 0\ and\ y_A < 0 \\ \pi + \tan^{-1}\frac{y_A}{x_A} & if\ x_A < 0 \\ \frac{1}{2}\pi & if\ x_A = 0\ and\ y_A > 0 \\ \frac{3}{2}\pi & if\ x_A = 0\ and\ y_A < 0 \\ undefined & if\ x_A = 0\ and\ y_A = 0 \end{cases} \tag{4.5}$$

From the angle φ the index l^* corresponding to the center of the rows can be computed by Eq. (4.6) as:

$$l^* = round\left(\frac{\varphi}{2\pi} m\right) \tag{4.6}$$

In Eq. (4.6) the round operation indicates rounding to the nearest integer in the interval from 1 to m=640. Fig. 4.12 d) shows the crop row central line obtained from the center of gravity, represented by the red line. Thanks to the shape of the white pixels intensity distribution, it is guaranteed that this red line will be contained inside the crop row.

4.9 Downscaling

The final step is to take the detected crop row central line obtained from the Telephoto image and downscale it into the Wide-angle image. This allows to correct the position of the crop row central line obtained from the Wide-angle image shown in Fig 4.11. This situation is shown Fig. 4.14; the blue lines represent the crop rows detected inside the Wide-angle image ROI by following the procedure

described in section 4.7. The red line represent the central crop row detection obtained from the Telephoto image by following the procedure described in section 4.8. The red line can be extended along the Wide-angle ROI due to the fact that most of the crop rows are seldom curved; they are usually straight and parallel due to the geometry of agriculture implements like plows and planters, to a point that it is even possible to build perspective models to guide a row crop navigation vehicle (Pla et al., 1997).

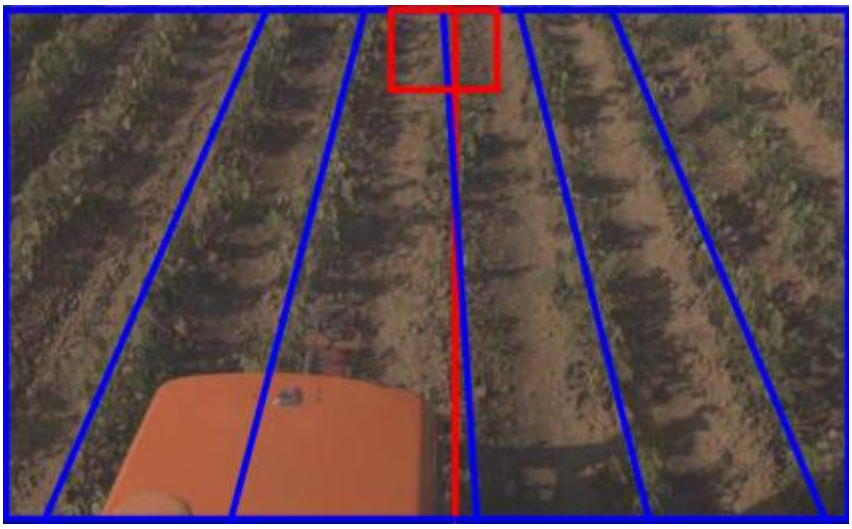


Fig. 4.14. Wide-angle and Telephoto image crop row detection comparison.

Fig. 4.14 also shows that the central blue line corresponding to the central crop row detected inside the Wide-angle image ROI has some slope inclination, although it is contained within the limits of the central crop row. This is the result of the natural variation in the plants, which can create some line offset from their ideal position. The method used to improve this situation is explained in the following section.

4.10 Camera calibration and image recording

Accuracy of detection is an important issue to be addressed in image processing (Romeo et al., 2013). There are two main types of parameters affecting the accuracy of the images, namely: extrinsic, related to the camera's positioning in the test vehicle; and intrinsic, related to the camera's specifications, such as CCD resolution, focal length or iris aperture, among others. Therefore, correct camera sensor arrangement is required to adjust the extrinsic parameters by means of appropriate camera calibration.

4.10.1 Camera calibration

Before recording the video frames in the field, the Fujifilm 2-in-1 camera was calibrated by shooting an image of a rectangular whiteboard. The whiteboard had distance points marked on it, forming a 10 cm by 10 cm square grid. The camera was placed in the top of the test vehicle, 2.75m aboveground at a pitch angle of 23.2 deg. from a horizontal orientation. Fig. 4.15 shows the Fujifilm 2-in-1 camera calibration setup. The whiteboard was placed on the ground, in the center of the image field of both wide-angle and telephoto images with the purpose of relating the image pixel position with the whiteboard distance points, which correspond to the real world distances in cm.

The top right portion of Fig. 4.15 shows the relation between the image pixel position and the real world distance points drawn on the white board obtained from the telephoto image. The 10 cm by 10 cm square grid looks like a trapezoid due to perspective from the top of the test vehicle.

Fig. 4.15. Fujifilm 2-in-1 camera calibration setup.

The resulting pixel-distance relation allowed transformation of coordinates between the image plane and the ground plane. As mentioned in section 4.2, in this research the camera calibration and all the image processing was developed in C++ language using the OpenCV library (Bradski and Kaehler, 2008).

4.10.2 Image recording

Series of video frames were acquired in a soybean test field, located in the experimental farm of Hokkaido University. The test field was 150 m long and it had 40 crop rows with 0.70 m inter row spacing. The test vehicle was driven as parallel to the crop rows as possible so that, ideally, the calculated lateral offset, i.e. the lateral distance between the row and the image vertical center line, should have a constant value (Søgaard and Olsen, 2003).

The test vehicle traveled four paths along the test field at an average speed of 0.75 m/s. When the test vehicle is traveling, mechanical vibration and unevenness of the soil might affect the image stability. However, the video frames recorded during the experiments were smooth enough so neither image stabilization algorithms nor gimbal support was necessary. This statement can be verified by observing the experimental vehicle's sensors measurements, depicted in Fig. 4.16. The experimental path recorded by the RTK-GPS in Easing and Northing coordinates is shown in Fig. 4.16 a); this path data was used to build the crop map describe in Chapter 5 and to verify that the vehicle traveled with a small lateral offset. The vehicle's attitude recorded by the FOG is shown in Fig. 4.16 b). Note that since the FOG range of measurement is 0 deg. to 360 deg. the glitches in the plots actually represent the same direction; i.e. 0 deg. is equal to 360 deg. in the FOG reference frame of measurement. The yaw is equivalent to the heading of the vehicle, and it is evident how changes 180 deg. for each one of the four experimental paths. The attitude data recorded by the FOG was originally intended to stabilize the images required in the mapping process, in case there were abrupt changes in the vehicle's attitude; particularly in the pitch. However, it can be seen in Fig. 4.16 b) that the yaw, pitch and roll values have an overall stable behavior.

The experiments were conducted on two different days; at different hours of the day, in order to account for different illumination conditions. Fig. 4.17 shows the Fujifilm 2-in-1 camera field of view, as shown in Chapter 2, section 2.5. The full size of the side by side image is 1280 x 480 pixels. The left side corresponds to the Wide-angle image and the right side corresponds to the Telephoto image; both images have a size of 640 x 480 pixels respectively. The blue rectangle represents the Region of Interest (ROI) used for crop row detection, which has a size of 420 x 256 pixels. The red rectangle has a size of 53 x 40 pixels and it represents the position of the Telephoto image inside the wide-angle image.

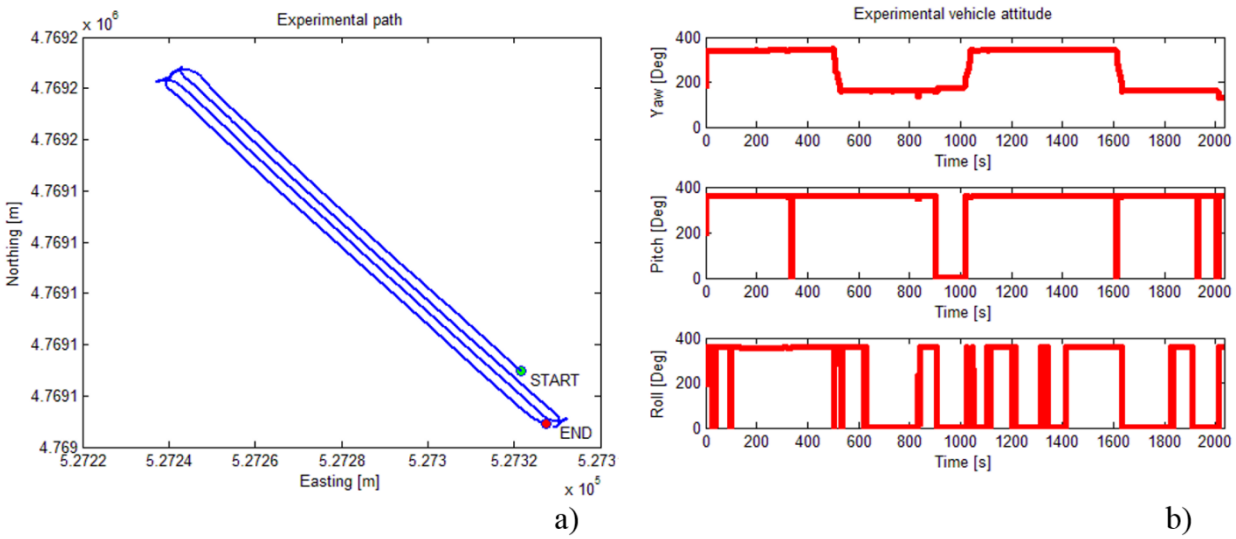


Fig. 4.16. Sensors measurements. a) Experimental path from RTK-GPS. b) Experimental vehicle attitude from FOG.

Fig. 4.17. Fujifilm 2-in-1 camera field of view; wide-angle image (left) and telephoto image (right).

As described in section 2.5 the telephoto image was set to a zoom factor of twelve; magnifying the image area of 53 x 40 pixels represented by the red rectangle in Fig. 4.17 into an image area of 640 x 480 pixels. This shows an advantage in the use of the 2-in-1 camera, because an increased image resolution can be achieved for a specific area with fixed position inside the wide-angle image. Therefore, the wide-angle image was used for crop row detection while the telephoto image was used for correction of the central row crop row detection. The wide-angle image was used for crop mapping as well, but this method will be introduced in Chapter 5. It is possible to perform crop row correction

(Guerrero et al., 2013) by using the Theil-Sen estimator or the Pearson product–moment correlation coefficient considering the dispersion of plants; however, these methods result in an increment of the image processing time.

4.11 Crop row detection

To analyze the crop row detection accuracy, the lateral offset from the central crop row real position was recorder for both wide-angle and telephoto images. To understand better the lateral offset estimation Fig. 4.18 shows the crop row detection results for one of the four experimental paths. The yellow line represent the ideal center of the central crop row; which correspond to the zero position of the left and right lanes. This means that any deviation to the left of the ideal center will be considered as a negative value and any deviation to the right will be considered as a positive value.

The blue lines represent the detected crop rows obtained from the Wide-angle image using the procedure described in section 4.7. Similarly, the red line represent the detected central crop row obtained from the telephoto image using the procedure described in section 4.8. The white line represent the maximum value of the distribution also described in section 4.8; as mentioned before this maximum value quite unstable, therefore is actually not used in the correction process. The position of the telephoto image inside the wide-angle image is represented by the white square. The white circle represent the center of the image frame. Note that lines have been downscaled using the procedure explained in section 4.9.

Fig. 4.18. Crop row detection results.

The pixel position of each one of the lines is then recorded for each one of the video frames; again, consider that any deviation to the left of the ideal center will be considered as a negative value and any deviation to the right will be considered as a positive value.

To visualize better this situation, Fig. 4.19 shows a detail of the crop row detection with a plot of the lanes pixel position shown in the top left portion of the figure. The plot vertical axis represents the image frame horizontal position in pixels. This means the detected crop rows are found within the range 280-280 pixels. This 100 pixels window is used to show the lateral offset of the detected crop rows. As mentioned in Fig. 4.18 the yellow line represents the central crop row ideal position. The green line represents the central crop row real position measured manually frame by frame based on an expert criterion (Bengochea-Guevara et al., 2016). The blue line represents the wide-angle image central crop row detection and the red line represents the telephoto image central crop row detection.

Fig. 4.19. Crop row detection detail.

The next step is to take these pixel values and convert them to real world units, thanks to the calibration process described in section 4.10.1. The plot of the crop rows center is shown in Fig. 4.20. The lateral offset is shown in cm. The yellow line represents the central crop row ideal position. The blue line represents the wide-angle image central crop row detection and the red line represents the telephoto image central crop row detection.

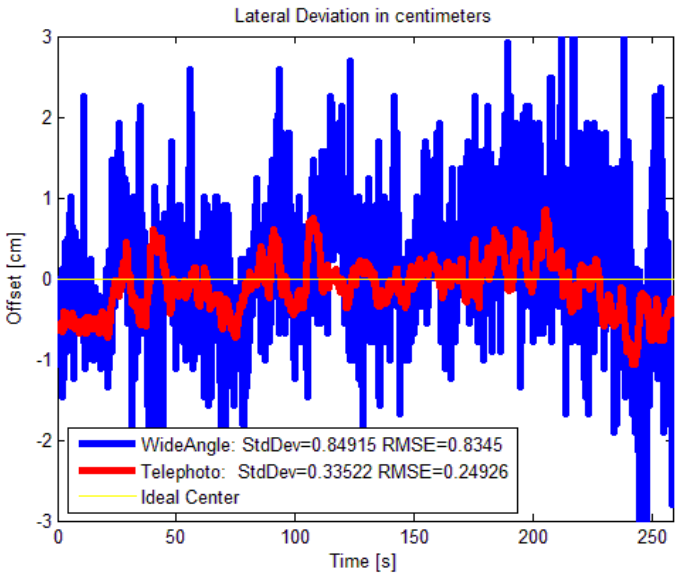


Fig. 4.20. Crop row detection plot.

To analyze the crop row detection accuracy, the Root Mean Square Error and the Standard Deviation respect to the central crop row ideal position were calculated. From Fig. 4.20 it can be seen how the wide-angle image central crop row detection represented by the blue line shows a standard deviation of 0.8491 and Root Mean Square Error of 0.8345; which are comparable to traditional crop row detection methods (Romeo et al., 2012). Also from Fig. 4.20, it is evident how the telephoto image central crop row detection represented by the red line shows a smaller standard deviation of 0.3352 and Root Mean Square Error of 0.2492 compared to the crop row detection from the wide-angle image (Kise et al., 2005). This can be interpreted as an improvement to traditional crop row detection methods. This research took advantage of the telephoto image data and applied data fusion to correct the wide-angle image data as described in next section.

4.12 Wide angle and telephoto images data fusion

Data fusion is the process of integrating multiple data sources to produce more consistent, accurate, and useful information than that provided by any individual data source (Hall and Llinas, 1997). Data fusion processes are often categorized as low, intermediate, or high, depending on the processing stage at which fusion takes place (Blasch et al., 2006). Low-level data fusion combines several sources of raw data to produce new raw data. The expectation is that fused data is more informative and synthetic than the original inputs.

There has been an ever-increasing interest in multi-disciplinary research on multisensor data fusion technology, driven by its versatility and diverse areas of application. Data fusion is a wide ranging subject and many terminologies have been used interchangeably. These terminologies and ad hoc methods in a variety of scientific, engineering, management, and many other publications, show the fact that the same concept has been studied repeatedly. As a result, the terms data fusion and multisensor data fusion are used interchangeably (Khaleghi et al., 2013). The data fusion research community have achieved substantial advances, especially in recent years. Nevertheless, realizing a perfect emulation of the data fusion capacity of the human brain is still far from accomplished.

In conclusion, several data fusion methodologies have been developed with their respective advantages and disadvantages according to their field of application. Table 4.2 summarizes some of the most common data fusion techniques (Castanedo, 2013) as quick reference of how wide this field of knowledge has become.

Table 4.2. Data fusion methods summary.

Data Association Techniques	State Estimation Methods	Decision Fusion Methods
1. Nearest Neighbors and K-Means 2. Probabilistic Data Association 3. Joint Probabilistic Data Association 4. Multiple Hypothesis Test 5. Distributed Joint Probabilistic Data Association 6. Distributed Multiple Hypothesis Test 7. Graphical Models	1. Maximum Likelihood and Maximum Posterior 2. The Kalman Filter 3. Particle Filter 4. The Distributed Kalman Filter 5. Distributed Particle Filter 6. Covariance Consistency Methods: Covariance Intersection/Union	1. The Bayesian Methods 2. The Dempster-Shafer Inference 3. Abductive Reasoning 4. Semantic Methods

After reviewing some if these data fusion techniques, the simplest solution found was a complementary filter (Higgins, 1975). Deep explanation of the complementary filter is beyond the scope of this research. For this research purposes in particular, the complementary filter has a considerable advantage over other data fusion techniques; like the Kalman filter, because the Riccati equation and Kalman gains are not computed.

Therefore, the update rate of the complementary filter can be higher than the update rate of the Kalman filter. The basic complementary filter is shown in Fig. 4.21; where *x* and *y* are noisy measurements of some signal *z*, and $\hat{z}$ is the estimate of *z* produced by the filter. Assume that the noise n_2 in *y* is mostly high frequency, and the noise n_1 in *x* is mostly low frequency. Then *G(s)* can be made a low-pass filter to filter out the high-frequency noise in *y*. If *G(s)* is low-pass, *[1 - G(s)]* is the complement, i.e., a high-pass filter which filters out the low-frequency noise in *x*. No detailed description of the noise processes are considered in complementary filtering (Stewart and Parks, 1957).

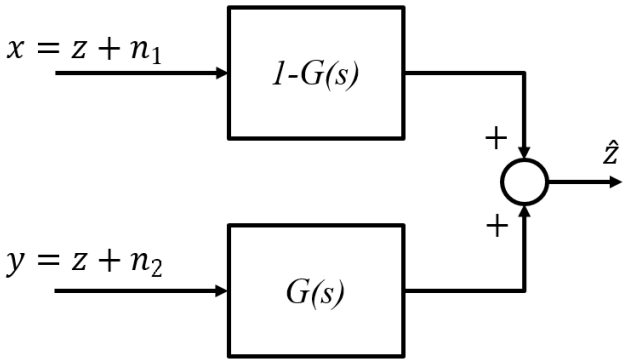


Fig. 4.21. Basic complementary filter.

The inputs of complementary filter are defined as follows. Consider the data shown in Fig. 4.22, the input *y* corresponds to the wide-angle image data shown in blue and the input *x* corresponds to the telephoto image data shown in red. Therefore, the estimate output of the complementary filter is given by Eq. (4.7):

$$\hat{z} = x[1 - G(s)] + y[G(s)] \tag{4.7}$$

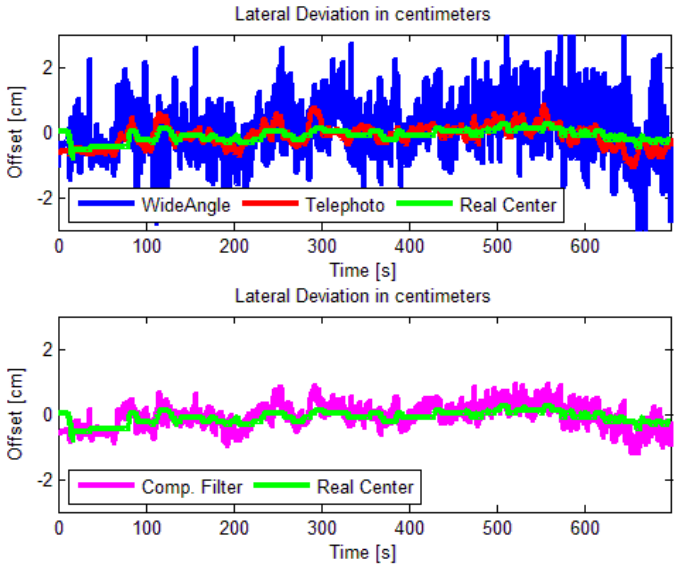


Fig. 4.22. Complementary filter result.

The purple line in Fig. 4.22 represents the complementary filter fusion result $\hat{z}$. The gain of the complementary filter was found by means of fine tuning. It can be seen how the complementary filter fusion result combines the best qualities of the two input signals x and y, evident as an overall noise reduction. The comparison of this overall noise reduction is shown in detail in Fig. 4.23 a); again, the yellow line represents the central crop row ideal position; i.e. the center of the image frame. The green line represents the central crop row real position measured frame by frame based on an expert criterion. It can be seen how the complementary filter result has a reduced Root Mean Square Error and Standard Deviation compared to the wide-angle image data.

It is also important to consider the stability of the error. This result is shown in detail in Fig. 4.23 b) which compares the central crop row real position, represented by the green line, with the complementary filter fusion result represented by the purple line. The error of the complementary filter fusion result respect to the central crop row real position is represented by the black line, which is within a margin of ±1cm. The Mean of the error is -0.0623 and the Variance is 0.0855; these values show a meaningful performance improvement in comparison to other approaches (Åstrand and Baerveldt, 2005).

Although correction is performed only in the central crop row, the accuracy of the method can be used to build perspective models to guide a row crop navigation vehicle (Pla et al., 1997). This is possible

based on extraction of the scene structure from perspective information that a set of coplanar parallel lines, representing the crop rows, generate in the image. This means that correction of the central crop row can be extrapolated to other crop rows; however, that is part of future work.

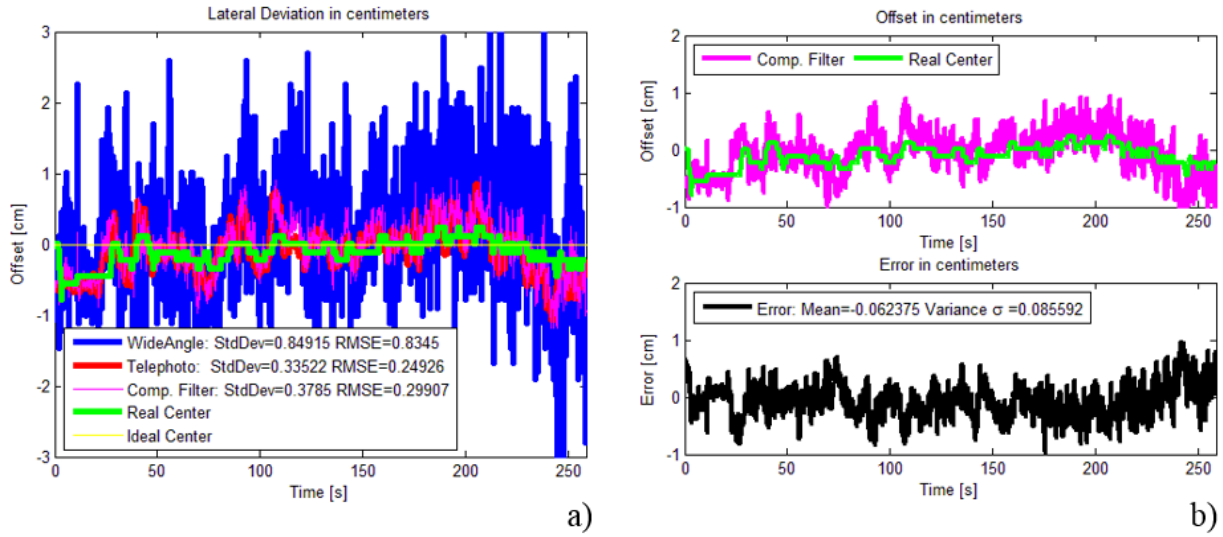


Fig. 4.23. Crop row detection results comparison. a) Wide-angle and Telephoto RMSE. b) Mean and variance of the complementary filter error.

4.13 Conclusions

This chapter has described the image processing methods used in this research. The context of image processing or precision agriculture was explained as an introduction. The software used was briefly explained. The image processing methods used to detect the crop rows from the experimental field where explained step by step including their mathematical models. The differences between the RGB color space and the HSV color space were described. Morphological operations and their relevance were examined. The contours and geometric centers calculation lead to the conclusion that the crop rows position can be estimated by using a combination of point estimation and weighted linear regression.

The camera calibration method was introduced; giving as a result the pixel-distance relation that allowed transformation of coordinates between the camera's image plane and the ground plane. The image recording section explained how series of video frames were acquired in the soybean test field, located in the experimental farm of Hokkaido University.

From the figures included it was found that the traditional method for detecting the crop rows from a wide-angle image can be improved by the use of correction data from a telephoto image. Thanks to the telephoto image's increased resolution, accuracy of the image recognition algorithm can be improved by fusing the wide-angle image data with the telephoto image data using a complementary filter, reducing the lateral position deviation range from 0.061 m to 0.028 m. Although results display increased accuracy for the lateral position calculated from crop row detection, the machine vision measurements still have some inherent noise that can affect the navigation performance of the vehicle. Next chapter discusses how to integrate the best aspects of the vehicle motion model estimations described in chapter 3 with the machine vision measurements in order to clean this inherent noise.

Chapter 5 Application to automatic navigation and crop mapping

This chapter describes how to apply the machine vision system integrated with the vehicle motion model to a smart agricultural vehicle capable of performing automatic navigation and crop mapping. To achieve automatic navigation, it is explained how to measure the vehicle's heading φ from the detected crop rows. A complementary filter was used in order to integrate the best aspects of the vehicle motion model estimations with the machine vision measurements. Some sections of this Chapter refer to the equations models described in Chapter 3 in order to account to the vehicle's motion model improvement in combination with the machine vison method.

In addition; this research goes a step beyond the automatic navigation from crop row detection adding a crop mapping method. This chapter describes the methodology to build a crop map taking advantage of the Fujifilm 2-in-1 camera wide-angle lens. Limitations of the current crop mapping methods are explained in order to justify the purpose of this method. Results of the proposed approach are discussed according to the theoretical error found between the processed images and the RTK-GPS path data. It is concluded that the mapping method has a reasonable error and also has a clear advantage over mapping methods that employ conventional cameras.

5.1 Heading detection

In order to apply the machine vision method described in chapter 4 to an automatic navigation system, it is necessary to estimate the vehicle's lateral position y and the heading φ from the detected crop rows (Pinto et al., 2000). The method to measure the vehicle's lateral position y (also called offset y) from the detected crop rows was described in section 4.11, concluding that the machine vision measurements still have some inherent noise that can affect the navigation performance of the vehicle.

Despite the noise each combination of the vehicle guidance parameters; lateral position y and heading angle φ, can be treated as a pose of the crop rows. An image sensing technique using image plane analysis to measure the heading angle φ of the vehicle based on the crop rows viewed from the camera was used (Yeh et al., 1994).

To measure the heading angle φ from the straight crop rows the geometric relationship of the test vehicle relative to the crop rows is shown in Fig. 5.1 a). The vehicle's lateral position *y* is the same as the offset shown in Fig. 5.1 a). The nearest distance that the camera can capture is located at baseline l_1 with distance *d* ahead of the Center of Gravity *C.G.* of the vehicle. The crop rows shown in Fig. 5.1 a) and the center line between them are parallel. Therefore, from the perspective of the top of the test vehicle they converge to a vanishing point *P* located in the horizon line P_1P_2. This vanishing point *P* is shown in Fig. 5.1 b). According to the analysis illustrated in Fig. 5.1, the heading angle can be obtained from the position of vanishing point *P* as described by Eq. (5.1):

$$\frac{\tan\varphi}{\tan\theta} = \frac{\overline{OP}}{\overline{OP_2}} \cong \frac{\varphi}{\theta}$$

$$\varphi = \tan^{-1}\left(\overline{OP} \times \tan\theta / \overline{OP_2}\right) \tag{5.1}$$

In Eq. (5.1), the angle θ is the field of view of the camera, also known as the visual angle. Only the calculation of the heading angle φ from the wide-angle image was considered. This is because the visual angle of the telephoto image is too narrow and fits only the central crop row; it does not contain the vanishing point *P* shown in Fig. 5.1 b).

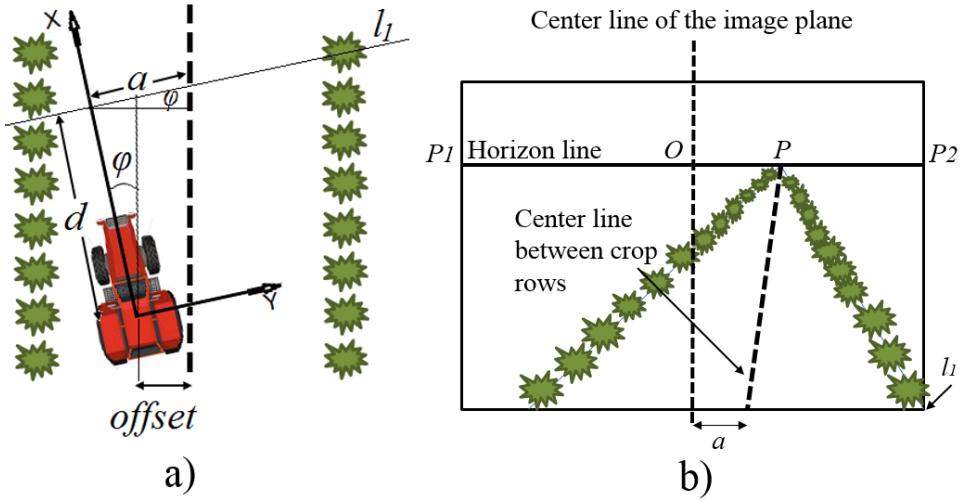


Fig. 5.1. Heading detection. a) Geometric relation of the vehicle relative to the crop rows. b) Corresponding view in image plane.

The plot of the heading angle φ_{WA} from the wide-angle image is shown in Fig. 5.2 as the green line. The blue line represents the measured heading φ from the FOG. It can be seen in Fig. 5.2 that although the measurement from the wide-angle image is noisy, it is within a range of 2 deg. and both the Standard Deviation and the Root Mean Square Error have reasonable values.

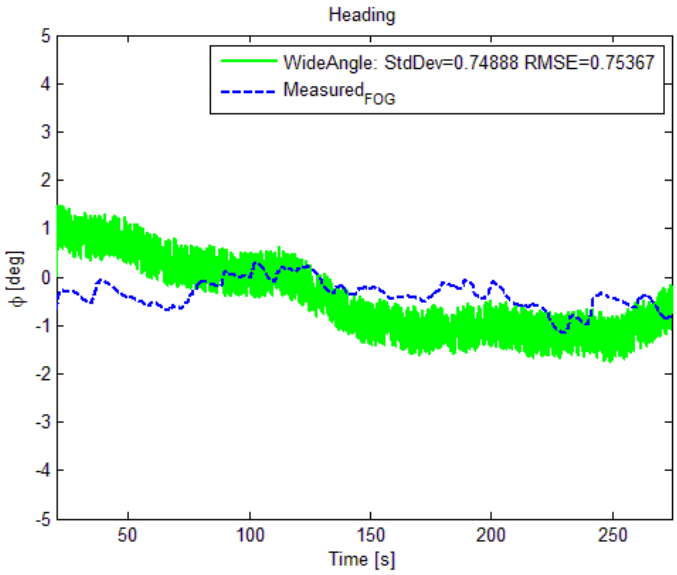


Fig. 5.2. Heading measurement comparison.

As described in section 2.3 the FOG (Japan Aviation Electronics Industry Ltd., JCS7402-A) provides both the vehicle's angular rates (accuracy $\leq \pm$ 5 deg. /s) and the attitude (accuracy $\leq \pm$ 0.15 deg.). This means that a high precision sensor like the FOG has an accuracy that is difficult (if not impossible) to match with the heading angle φ estimation calculation from Eq. (5.1). Even so, a range of 2 deg. seems reasonable for this kind of approach (Pinto et al., 2000). Next section discusses how to clean the inherent noise of the machine vision measurements.

5.2 Navigation system by sensor fusion integration

Following the results from section 5.1, it is possible to clean the machine vision inherent noise using a complementary filter that integrates the machine vision measurements with the vehicle motion model estimations. Remember that the vehicle motion model estimations were described in sections 3.3.1 and 3.3.2. The vehicle's heading φ and lateral position y are the critical parameters for automatic navigation systems (Zhang et al., 1999). However, there are many problems in developing a robust automatic navigation system based only on machine vision, mainly because the environment in which the vehicle

navigates is an outdoor space with many disturbances resulting from variable soil and weather conditions. Moreover, this environment also include various disturbances such as shadows, weed infestation and different soil colors and types.

Even thought, machine vision has been widely investigated and implemented in crop row detection systems (Burgos-Artizzu et al., 2011) in combination with an RTK-GPS for the vehicle's automatic navigation system. This has become possible thanks to the decreasing costs of RTK-GPS. Nevertheless, RTK-GPS also has low reliability issues caused by the interference of the GPS signals by trees and buildings. Therefore, to improve stability of the automatic navigation system, redundant sensing systems are required. The typical approach (Noguchi et al., 1998) is to develop the navigation system by sensor fusion integration with machine vision, RTK-GPS and some attitude sensor like a geometric a direction sensor (GDS) or an IMU. A data fusion methodology like the Kalman Filter or a Probability Density Function is then applied to combine data from the sensors.

In this research however, the RTK-GPS position and the FOG attitude data was used in the measurement methodology described in section 3.3.1 for estimating the vehicle's dynamic parameters β and ω. These dynamic parameters were used to describe the tire dynamic properties by means of a regression model technique, resulting in a model that does not depend on several parameters, such as longitudinal slip, sideslip angle, normal load, camber angle, tire pressure, wear, and road surface characteristics. The tire dynamic properties described by this regression model technique can be applied to obtain better model estimations of the vehicle's dynamic parameters β and ω. In turn, this vehicle's improved estimations were used to improve the motion model estimations of the test vehicle given by the kinematic model described by Eq. (3.1) giving as a result more precise estimations of the heading φ and position (x, y) of the vehicle, as described in section 3.3.2.

In other words, the purpose of this research is not to fuse de data from the machine vison system with the RTK-GPS data as typical approaches have achieved. The purpose of this research is to fuse the improved motion model estimations with measurements from the machine vison system and then compare them to the real position of the vehicle measured by the RTK-GPS and the real heading measured by the FOG.

5.2.1 Vehicle dynamic model estimations for a straight path

The experimental image recording was introduced in section 4.10.2, showing in Fig. 4.16 a) the four experimental paths recorded by the RTK-GPS in Easing and Northing coordinates. For simplicity, section 4.11 shows the crop row detection results for one of the four experimental paths. Similarly, this section shows the vehicle navigation model improvement results for one of the four experimental paths, following the method described in section 3.3.1. Fig. 5.3 shows in purple the body sideslip β_{REG} and yaw rate ω_{REG} obtained from the modeled regression equations in comparison to the measured body sideslip β and the measured yaw rate ω shown in blue.

The dynamic modeled body sideslip β_{DYN} and yaw rate ω_{DYN} are shown in red. It can be seen how the behavior of each data is consistent to the measured data. It is necessary to calculate the error in order to evaluate the accuracy improvement of the body sideslip β_{REG} and yaw rate ω_{REG} obtained from the regression model. The RMS error for both the body sideslip β_{REG} and yaw rate ω_{REG} is also shown in Fig. 5.3. The RMS error for both the body sideslip β_{DYN} and yaw rate ω_{DYN} is included as well.

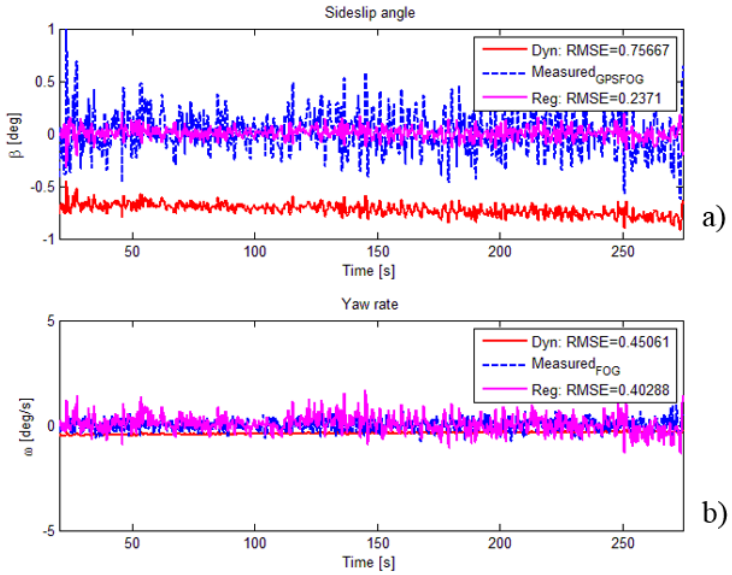


Fig. 5.3. Straight path dynamic parameters. a) Body sideslip β. b) Yaw rate ω.

Although the body sideslip β and the yaw rate ω are more notorious for a sinusoidal path described in section 3.3.1 (where the RMS error of the modeled body sideslip angle was reduced from 5.0 deg. to 3.4 deg.; and the modeled yaw rate RMS error was reduced from 7.3 deg/s to 4.6 deg/s.); determination

of both the body sideslip β and the yaw rate ω is also necessary for a straight path. This is because in farm machinery; like tractors, constant lateral velocities (known as vehicle ‘‘crab’’) can occur from traveling on sloped terrain, tires getting caught in a furrow, or pulling a heavy implement. Additionally, an apparent crab angle can occur from misalignment of the heading sensor (Bevly et al., 2002).

5.2.2 Heading integration results

Again, as described in section 3.3.2 it is possible to estimate the navigation model improvement by using the data summarized in Fig. 5.3; this means to compute the body sideslip β_{REG} and yaw rate ω_{REG} into Eq. (3.1) to obtain the position $(x, y)_{REG}$ and heading φ_{REG} of the vehicle estimated from the regression model. Similarly, it is possible to compute the body sideslip β_{DYN} and yaw rate ω_{DYN} into Eq. (3.1) to obtain the position $(x, y)_{DYN}$ and heading φ_{DYN} of the vehicle estimated from the dynamic model with constant cornering stiffness.

Fig. 5.4 shows in purple the heading φ_{REG} obtained by computing the body sideslip β_{REG} and yaw rate ω_{REG} into Eq. (3.1). The measured heading φ obtained from the FOG is shown in blue for comparison purposes. The dynamic modeled heading φ_{DYN} is shown in red. It is evident how the dynamic modeled heading φ_{DYN} has an overall bigger deviation in comparison with the other data sets. Similarly to the plot shown in Fig. 5.2, the heading angle φ_{WA} from the wide-angle image is shown in Fig. 5.4 as the green line.

It is necessary to calculate the error in order to evaluate the accuracy improvement of the navigation model. The RMS error for both headings φ_{REG} and φ_{DYN} respect to the measured data are also shown in Fig. 5.4. It can be noted how using the values obtained from the regression model, the accuracy improved for the heading φ_{REG} in contrast with the dynamic model φ_{DYN}.

The next step is to combine the heading φ_{REG} from the regression model with the heading angle φ_{WA} from the wide-angle image. The black line in Fig. 5.4 represents the complementary filter fusion result $\hat{z}$. The gain of the complementary filter was found by means of fine tuning. It can be seen how the complementary filter fusion result combines the best qualities of the two input signals φ_{REG} and φ_{WA}, evident as an overall noise reduction. It can be seen how the complementary filter result has a reduced Root Mean Square Error and Standard Deviation compared to the other data sets.

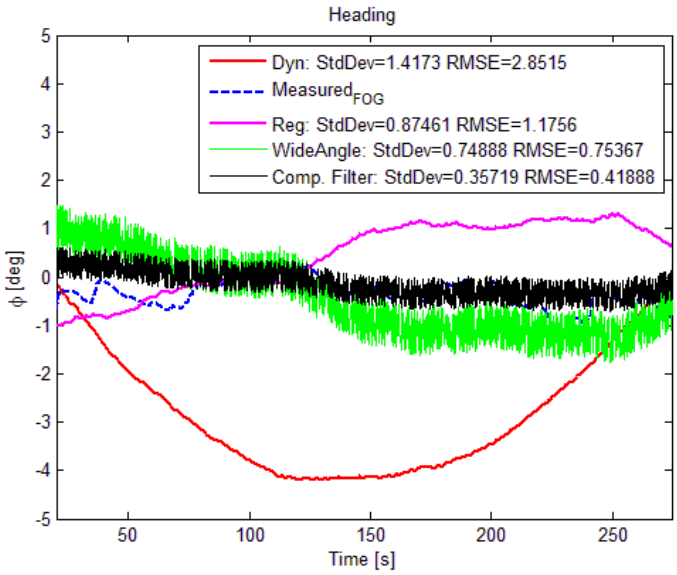


Fig. 5.4. Heading comparison.

In conclusion, the RMS error of the heading angle φ_{WA} from the wide-angle image was reduced from 0.75 deg. to 0.42 deg., i.e. the RMS error was reduced over 30%.

5.2.3 Position integration results

Fig. 5.5 shows in purple the position of the vehicle obtained by computing the body sideslip β_{REG} and yaw rate ω_{REG} into Eq. (3.1). Note that Fig. 5.5 is plotted in Easting and Northing coordinates. The measured position from the RTK-GPS is shown in blue for comparison purposes. The dynamic modeled position is shown in red. Again, it can be observed how the behavior of each data set is consistent to the measured data. However, in order to compare the path with the lateral offset measured with the machine vision system it is necessary to rotate it respect to the horizontal axis.

The rotated path is plotted in Fig. 5.6; which shows in purple the position obtained by computing the body sideslip β_{REG} and yaw rate ω_{REG} into Eq. (3.1). Same as Fig. 5.5, the measured position from the RTK-GPS is shown in blue for comparison purposes. The dynamic modeled position is shown in red. It is evident how the dynamic modeled position has an overall bigger deviation in comparison with the other data sets. The complementary filter result obtained from fusing the wide-angle and telephoto images shown as the purple line in Fig. 4.23 is represented in Fig. 5.6 as the green line. The RMS error

for both the regression modeled lateral position and the dynamic modeled position respect to the measured data are also shown in Fig. 5.6. It can be noted how using the values obtained from the regression model, the accuracy improved for the regression modeled lateral position in contrast with the dynamic modeled lateral position.

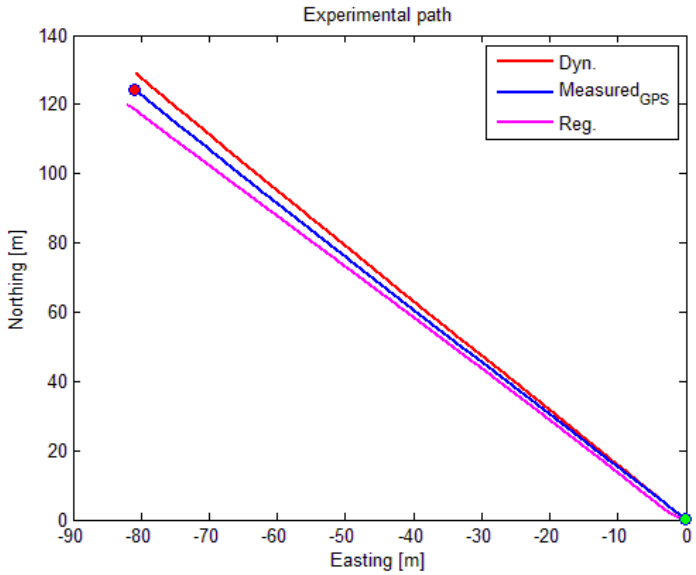


Fig. 5.5. Navigation path comparison.

The next step from Fig. 5.6 is to integrate the position from the regression model shown in purple with the position from machine vision shown in green. The black line in Fig. 5.6 represents the complementary filter fusion result $\hat{z}$. The gain of the complementary filter was found by means of fine tuning. It can be seen how the complementary filter fusion result integrates the best qualities of the two input data sets of position, evident as an overall noise reduction. It can be seen how the complementary filter result has a reduced Root Mean Square Error and Standard Deviation compared to the other data sets. In other words, the machine vision lateral position RMS error was reduced from 0.028 m to 0.024 m; therefore reducing the RMS error around 20%.

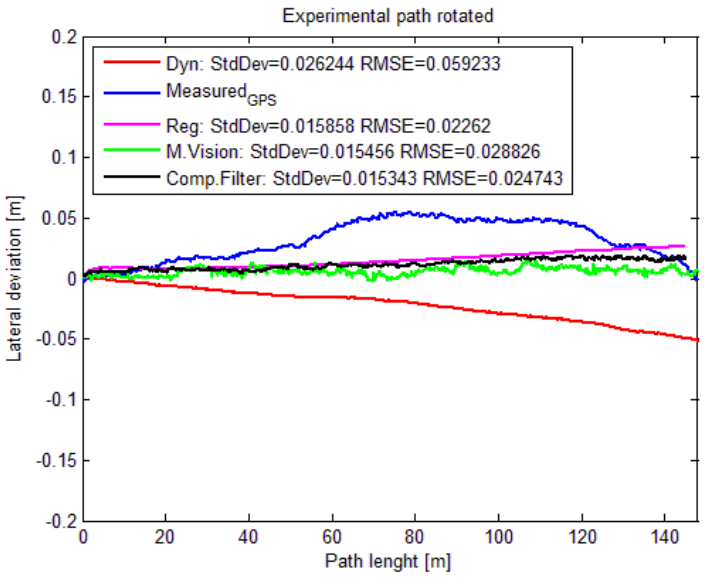


Fig. 5.6. Navigation path rotated comparison.

From Figs. 5.4 and 5.6, it can be observed how by using the data obtained from the regression model the accuracy improved for both the heading and the lateral position of the vehicle. This result was expected due to the fact that the accuracy improved for both the sideslip β and the yaw rate ω in contrast with the dynamic model as shown in Fig. 5.3. As stated before, one important drawback of the dynamic navigation model described by Eq. (3.1) is its inherent integration error; however, this does not interfere with the purpose of comparing the results obtained by computing the body sideslip β_{REG} and yaw rate ω_{REG} with the results obtained by computing the body sideslip β_{DYN} and yaw rate ω_{DYN} because the same Eq. (3.1) is being used; only the input data sets are different.

As concluded before, smaller RMS errors mean that tire parameters can be used to obtain better estimations of the body sideslip β_{REG} and yaw rate ω_{REG}; which in turn can be used to improve the vehicle motion model estimations. It can also be noted how to reduce Root Mean Square Error of the machine vision lateral position measurement; by fusing the machine vision result shown in green with the improved navigation model result shown in purple. As explained in section 2.2, the RTK-GPS has an inherent error of approximately 3cm; and it is evident that the position plot shown blue in Fig. 5.6 is between 0 and 5 cm. Therefore, RMSE will be very small because lateral deviations are really small (~3 cm) in contrast to the path length of the test field (~150 m).

5.3 Crop mapping background

Machine vision systems onboard robots have become increasingly important in precision agriculture in order to fully automate some in-field agricultural tasks; like automatic navigation of an agricultural vehicle. In addition, machine vision systems in precision agriculture are also used to gather data from the field in an automated manner at minimal cost; like crop mapping. Several researches (Pajares et al., 2016) have achieved good results performing either crop mapping or crop navigation separately.

However, current machine vision systems methods have limitations trying to perform both navigation and mapping processes at the same time. Methods for automatic navigation of an agricultural vehicle based in crop row detection (Romeo et al., 2013) focus into detecting crop rows as accurate as possible; typically shooting the central three crop rows at the same time. Detecting three crop rows might provide enough accuracy for navigation; however, only three crop rows do not seem to contain enough information to build a field map of practical use. This situation is illustrated in Fig. 5.7 a) showing a top view of the tractor in the field. The vertical green lines represent the soybean crop rows and the triangle represents the angle of view of the camera. It is shown that the main limitation of this method comes from the fact that conventional camera lens have an angle of view less than 65 deg. On the other hand, a different approach (Slaughter et al., 2008) is to use several cameras on zenithal position, each camera shooting one crop row at the time giving as a result a high quality map containing several crop rows. This mapping method can be used in practical applications like spraying, but navigation from crop row detection becomes a more complex problem.

Considering these limitations, this research proposes a machine vision method capable of mapping several crop rows while simultaneously performing the crop row detection describe in section 4.7. As described in section 4.10.1 the camera was mounted in the top of the test vehicle, focused on the field surface from an inclined angle in order to obtain wide-angle images that cover up to eleven crop rows. This situation is illustrated in Fig. 5.7 b); the main advantage of the 2-in-1 camera and is the wide-angle lens increased angle of view of 105 deg. The camera parameters were obtained from the manufacturer's user guide and were confirmed during the calibration process described in section 4.10.1.

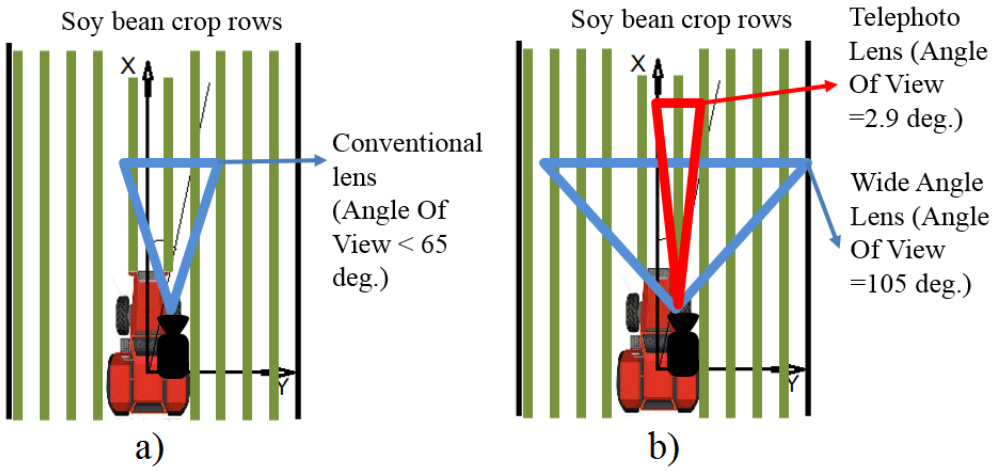


Fig. 5.7. Mapping method comparison. a) Conventional lens. b) 2-in-1 camera wide-angle lens.

Fig. 5.8 shows an overview of the wide-angle image. It can be seen how eleven crop rows fit in the image with a reasonable good resolution since the wide-angle image has a size of 640 x 480 pixels.

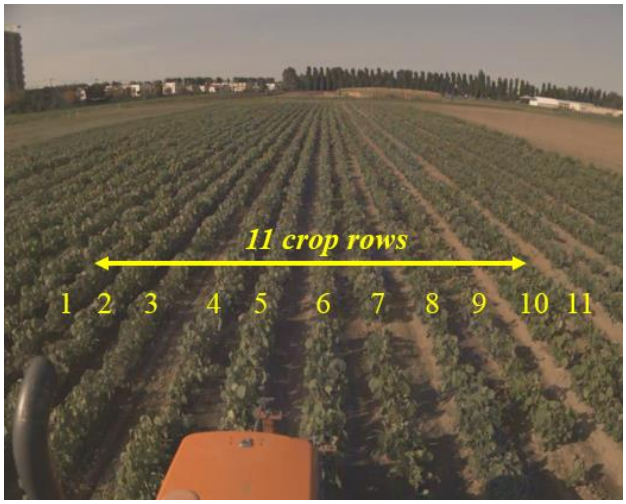


Fig. 5.8. Overview of the Wide-angle image.

5.4 Crop mapping method

As described in section 4.10.2, series of video frames were acquired in a soybean test field, located in the experimental farm of Hokkaido University. The test field was 150 m long and it had 40 crop rows with 0.70 m inter row spacing. The test vehicle traveled four paths along the test field at an average

speed of 0.75 m/s. As mentioned in section 4.10.2, the attitude data recorded by the FOG was originally intended to stabilize the images required in the mapping process, in case there were abrupt changes in the vehicle's attitude; particularly in the pitch. However, it was shown in Fig. 4.16 b) that the yaw, pitch and roll values have an overall stable behavior.

The experimental path recorded by the RTK-GPS in Easing and Northing coordinates was shown in Fig. 5.5; this path data was used to build the crop map and to verify that the vehicle traveled with a small lateral offset. Since the purpose of this section is to build a map of the crop field using recorded video sequences, it is necessary to integrate the information contained in each video frame into a complete map of the entire field length (Sainz-Costa et al., 2011). This map consists of a grid of specific dimensions, with each of its elements corresponding to a cell of a given size in the real field as represented in Fig. 5.9. Since the 2-in-1 camera is mounted on the top of test vehicle; which is traveling forwards, each new video frame covers a slightly different field area, and the map's frame of reference must be updated according to the RTK-GPS position coordinates. Remember that section 4.10.1 described the 2-in-1 camera calibration process; by using a whiteboard it is possible to relate the image pixel position with the real world position.

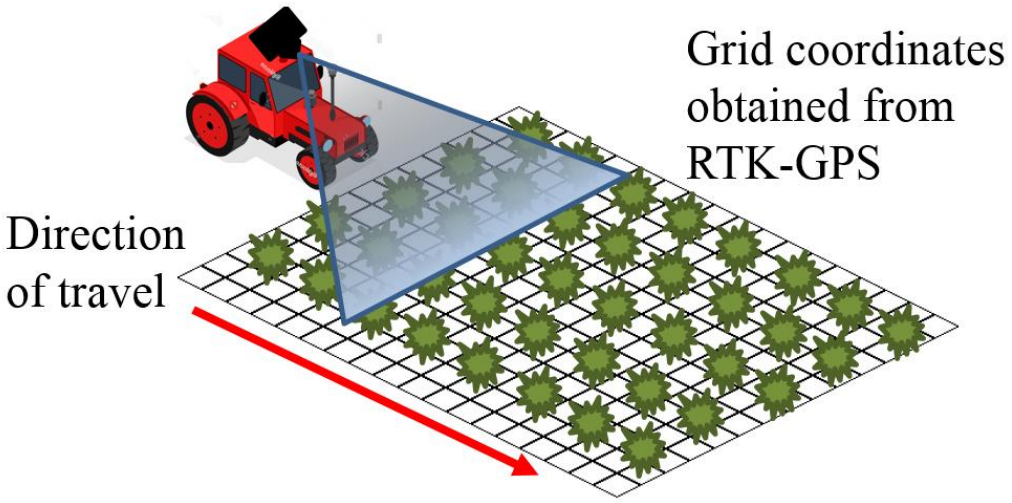


Fig. 5.9. Crop map grid scheme.

The distance in pixels in which the crops move between frames depends on the speed of the test vehicle, which was also obtained from the RTK-GPS. Mapping was not performed by using the raw series of video frames acquired in the test field because the images taken with cameras are 2D projections of the 3D world, and the recovery of 3D information such as depth, length or area requires a model of the projection transformation (Sonka et al., 2008). Therefore, the original images taken in the camera's

image plane coordinate system were transformed into the ground coordinate system, corresponding to the image view from above (Okamoto et al., 2002) or bird's-eye view.

This coordinate system transformation requires to understand the camera perspective projection in the ground plane, illustrated in Fig. 5.10. Consider a point *p* in the ground plane, perspective projection means that we draw a line through this point *p* and the center of projection of the camera and intersect it with the image plane to find the corresponding image point *p'* (Mallot et al., 1991).

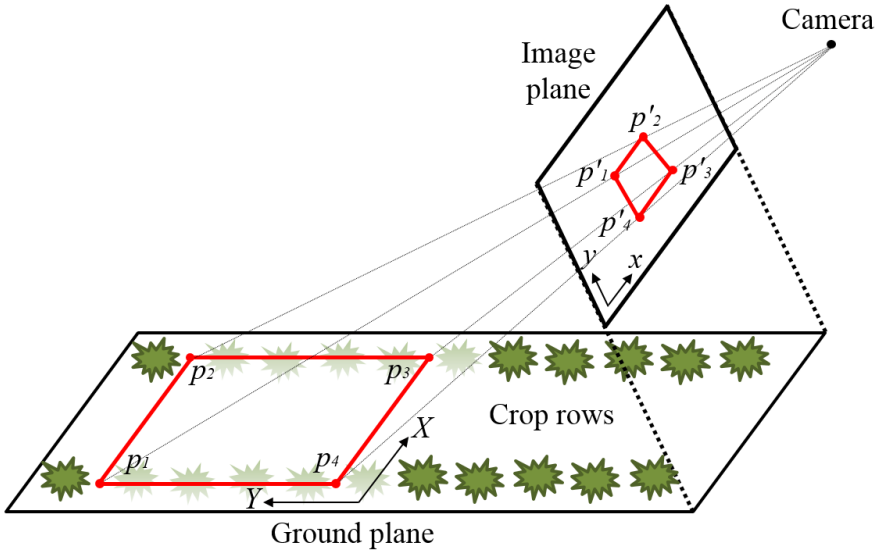


Fig. 5.10. Fujifilm 2-in-1 camera perspective projection in the ground plane.

Therefore, any point in the ground plane *p(X, Y, Z)* shown in Fig. 5.10 can be projected into the image plane as a point *p'(x, y, z)* using Eq. (5.2) (Barnard, 1983):

$$p' = Mp \tag{5.2}$$

In Eq. (5.2) the variable *M* is the 3×3 perspective transformation matrix. The calculation of *M* can be achieved by establishing direct correspondences between the ground plane points and the image plane points (Szeliski, 2010). For example; in Fig. 5.10 the point p_1 in the ground plane corresponds to the point p_1' in the image plane, the point p_2 in the ground plane corresponds to the point p_2' in the image plane and so on. The calculation of *M* requires at least four point correspondences of which no more than two are collinear (Brahmbhatt, 2013). The set of four points (p_1, p_2, p_3, p_4) shown in Fig. 5.10 are the coordinates of the quadrangle vertices from the input image frame that we want to relate to the

coordinates of the corresponding quadrangle vertices *(p_1', p_2', p_3', p_4')* in the transformed image frame. Therefore, Eq. (5.2) can be rewritten as Eq. (5.3) considering that the coordinate *z* can be set to one because the transformation is a 2D projection:

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ 1 \end{bmatrix}, (i = 1, 2, 3, 4) \tag{5.3}$$

Fig 5.11 shows the Region of Interest (ROI) of 617 x 72 pixels inside the Wide-angle image that was selected to cover eleven crop rows. Inside the ROI the set of four points *(p1, p2, p3, p4)* were selected as the vertices of the green trapezoid shown in the top right portion of Fig. 5.11. The corresponding points *(p_1', p_2', p_3', p_4')* in the transformed image were chosen to be the vertices of a rectangle of selected dimensions, as shown in the bottom right portion of Fig. 5.11. The dimensions of this rectangle; which corresponds to approximately 7.0 m width and 0.95m height, were obtained from calibration by the measurement of a known object as described in sections 4.10.1.

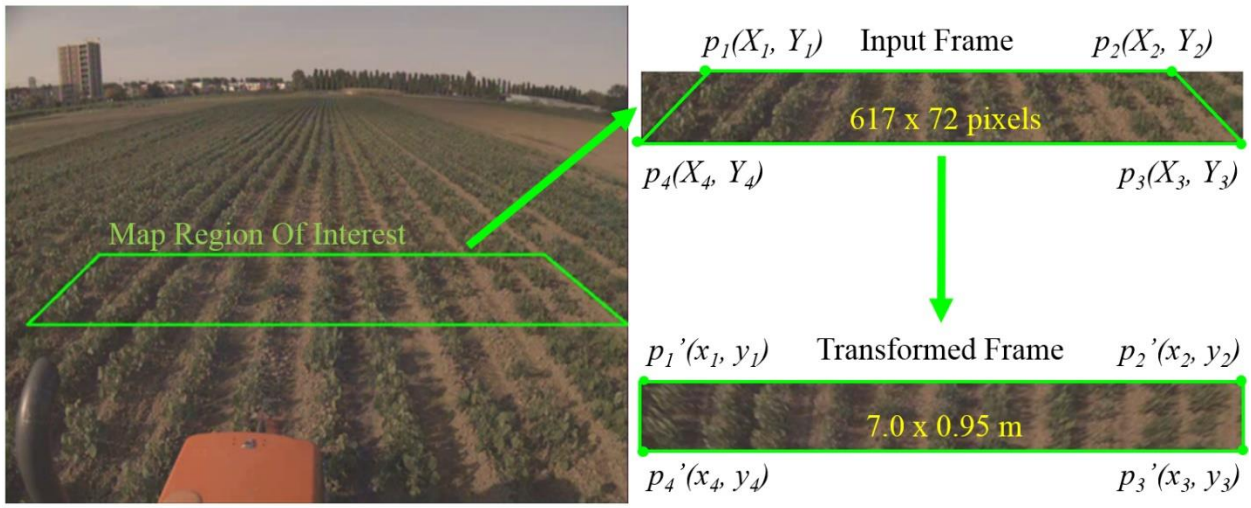


Fig. 5.11. Coordinates of quadrangle vertices in the source image.

After using Eq. (5.3) with the sets of four points *(p_1, p_2, p_3, p_4)* and *(p_1', p_2', p_3', p_4')* to calculate the 3×3 perspective transformation matrix *M*, all the remaining *n* points *$p_i(X_i, Y_i)$* of the image input frame can be computed to obtain the *n* points *$p_i'(x_i, y_i)$* of transformed frame by using Eq. (5.4):

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} \frac{M_{11}X_i + M_{12}Y_i + M_{13}}{M_{31}X_i + M_{32}Y_i + M_{33}} \\ \frac{M_{21}X_i + M_{22}Y_i + M_{23}}{M_{31}X_i + M_{32}Y_i + M_{33}} \end{bmatrix}, (i = 1, \dots, n) \quad (5.4)$$

Each transformed frame is then updated with the coordinates obtained from the RTK-GPS while the vehicle is traveling forward; as represented in Fig. 5.9, in order to build the mosaic map.

5.5 Crop mapping results

Concerning crop mapping results, traditional approaches map either four crop rows per traveled path (Kise et al., 2008) or three crop rows per traveled path (Sainz-Costa et al., 2011). Conceptually, this approaches work well but in real applications like spraying an agricultural vehicle typically covers more than four crop rows per traveled path (Hassen et al., 2014). Traveling additional paths in order to build a crop map might have undesirable effects in the field like soil compaction. Using a traditional approach mapping only three crop rows per path (Tillett et al., 2001.) means that the machine has to travel three times to cover nine crop rows. In the other hand, the method proposed in this research implementing the Fujifilm 2-in-1 camera can cover eleven crop rows travelling only one path.

Fig. 5.12 shows a segment of the resulting map. Since the test field was 150 m long and it had 40 crop rows with 0.70 m inter row spacing, the complete map is too big to show in the figure. Instead, a segment of the resulting map corresponding to one of the four experimental paths is shown. The equivalent size of the map segment in meters is shown as well; remember that the pixels-meters size relation was obtained from calibration. The eleven mapped rows are numbered from left to right. Fig. 5.12 also shows that the map has a resolution good enough to detect the absence of crops in a specific line. However, due to the inverse perspective transformation the map has obvious limitations because it does not offer a real top view of the field; therefore, detection of pests or weeds hidden from the camera's point of view is not possible.

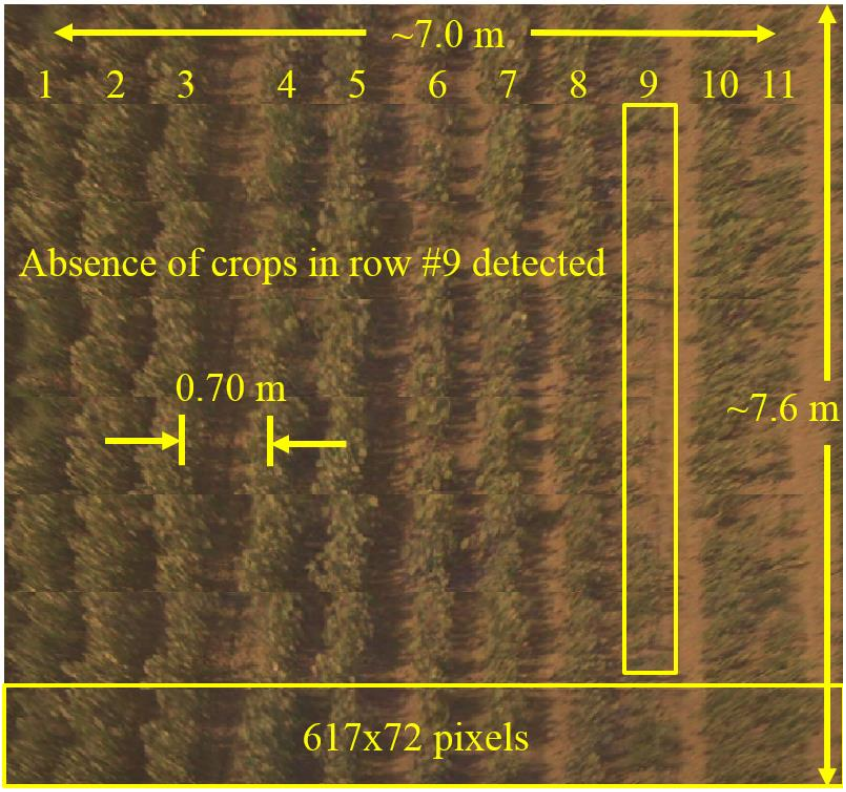


Fig. 5.12. Resulting crop map segment.

The map precision was also calculated from the camera calibration parameters in order to obtain the theoretical error. However, testing of the map's accuracy is not very straight-forward because there exists no true position and direction for the center line of a crop row due to the natural variation in the plant growth (Søgaard and Olsen, 2003). A row may locally develop more on one side than the other (Olsen, 1995) and the apparent row position at one particular point of the map may differ from the overall row position. Therefore, to evaluate the map theoretical error, the crop row values of the map image pixels are summed in the top-down direction of the map image. This gave as a result a crop row intensity distribution data set, where the mean position of each crop row can be compared with the map grid position obtained from the camera calibration parameters in combination with the RTK-GPS position data.

A map segment corresponding to one of the four experimental paths was used to estimate the map precision. The segment was 150 m length equivalent, containing eleven mapped rows. The RGB map segment was converted to a binary image so the crop rows values were easier to identify and sum.

Then again, the 150m length map segment is too big to fit in a figure; so instead, a sub-segment of 7 m length equivalent is shown in Fig. 5.13 a).

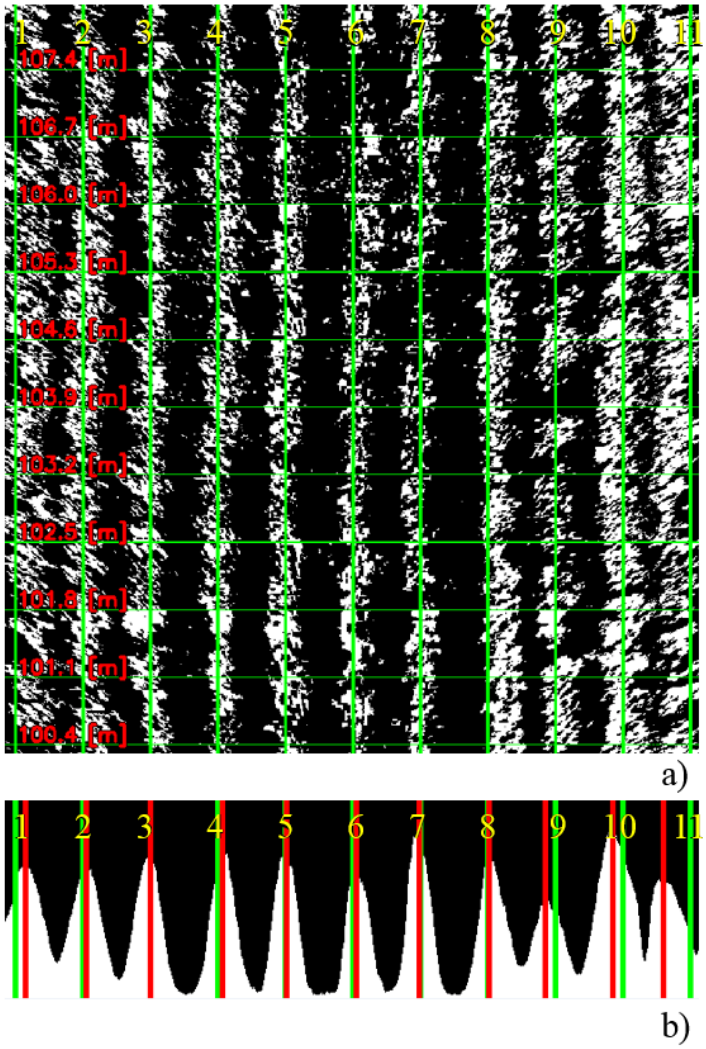


Fig. 5.13. Calculation of the map crop row intensity distribution. a) Binary map image with position grid. b) Crop row intensity distributions.

The eleven mapped crop rows are numbered from left to right in Fig 5.13 a). The map grid in green color represents the real world distances, the grid size is equivalent to 0.7 m by 0.7 m. This is the same grid represented in Fig. 5.9. The numbers in red color show the distance traveled by the experimental vehicle recorded by the RTK-GPS, represented as the horizontal lines of the grid. It can be observed how the vertical lines of the grid coincide at some degree with the crop rows. The difference between the vertical lines of the grid and the mean position of the crop rows make it possible to evaluate the

theoretical error of the map; i.e. to evaluate the lateral offset of each one of the mapped crop rows from the ideal map grid. Fig. 5.13 b) shows the crop row intensity distribution calculated for the 150m length map segment; remember that a sub-segment of only 7 m is shown in Fig. 5.13 a). The vertical green lines represent the map grid row position, and are on the same line as the vertical grid lines shown in Fig. 5.13 a). The red vertical lines represent the mean calculated for each one of the eleven distributions, which are equivalent to the mean position each one of the eleven mapped crop rows. It can be seen how the crest and valleys in the distribution have a clear difference for the central crop rows, represented by rows 4 to 8. However, the valleys are less deep for the outside crop rows, represented by rows 1 to 3 and 9 to 11. It is not possible to asset if this caused by the inverse perspective transformation, or if it is a mere coincidence produced by the random position of the crops leaves all along the map. It is possible to make the valleys more differentiated by setting a threshold along the distribution (Jiang et al., 2015), creating more space between the crests. Even thought, the crests do not seem to have a significant height difference; therefore, the crests give a good estimation of the mean position of each one of the eleven mapped crop rows. The images shown in Fig.5.13 are 617 pixels wide, and the pixel positions of the map grid represented by the green vertical lines and the row mean represented by the red vertical lines are known. This situation is more evident in Fig. 5.14, which shows in detail the data from Fig. 5.13 b). The plot of the distribution is represented by the blue line. Again, the map grid is represented by the green vertical lines, which have a constant inter-space of 60 pixels equivalent to 0.7 m. The crop rows mean positions are represented by the red vertical lines, whose inter-space pixel position is not constant.

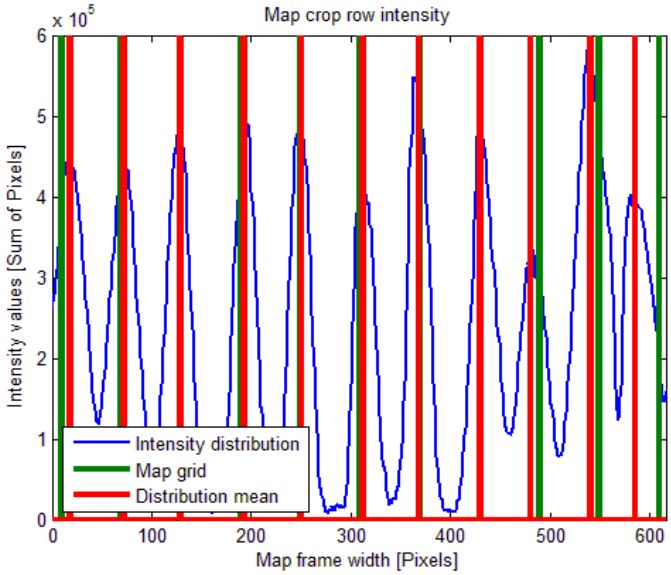


Fig. 5.14. Mean of the map crop row intensity distributions.

A summary of the theoretical error of the map, expressed as the error of the mean position for each crop row is shown in Table 5.1. The ideal map grid position in pixels for each one of the eleven crop rows is shown in the second column. In comparison, the third column shows the crop row mean position in pixels calculated from the distribution; as depicted in Fig.5.14. The error of the mean position in pixels was obtained as the difference between the crop row mean position and ideal map grid position, and it is summarized in the fourth column. The negative values represent a deviation towards the left side of the ideal map grid position, whereas positive values represent a deviation towards the right side. The same error of the mean position converted to cm is shown in the fifth column. The average error of all eleven mean position values is -2 pixels; equivalent to -2.3 cm. An average of the error of 2 pixels corresponds to 0.3 % of the 617 pixels frame width; which is the same as stating that an average of the error of 2.3 cm corresponds to 0.3 % of the 7 m frame width. Under the experimental conditions, a 0.3 % deviation margin seems to be a reasonable amount of theoretical error for the resulting map. Note that the biggest error comes from the eleventh crop row; again, it is not possible to asset if this caused by the inverse perspective transformation, or if it is a mere coincidence produced by the random position of the crops leaves all along the map. In addition, mapping accuracy is affected by environmental factors (Nguyen et al., 2015) like illumination changes and even wind; which interfere with accurate inverse perspective transformation (Bevilacqua et al., 2008).

Table 5.1. Map theoretical error.

Crop row number	Map Grid Position [Pixels]	Crop row mean position from distribution [Pixels]	Error of the mean position [Pixels]	Error of the mean position [cm]
1	9	18	9	10.5
2	69	72	3	3.5
3	129	129	0	0.0
4	189	193	4	4.7
5	249	250	1	1.2
6	309	312	3	3.5
7	369	368	-1	-1.2
8	429	430	1	1.2
9	489	480	-9	-10.5
10	549	540	-9	-10.5
11	609	585	-24	-28.0

In some cases (Sainz-Costa et al., 2011) sideslip of the experimental vehicle caused by the natural unevenness of the terrain gives as result a map that requires lateral sway stabilization. Particularly if the lateral deviation of the central crop row respect to center of the image frames is on average 20 pixels or 20 cm equivalent. However, crop row detection accuracy results discussed in section 4.12 showed that the lateral deviation of the central crop row respect to center of the image frames is within 2 cm equivalent; which is not significant enough to make the stabilization process meaningful.

5.6 Conclusions

The heading detection section showed that a high precision sensor like the FOG has an accuracy that is almost impossible to match with the heading angle φ obtained from machine vision, but the results have an error that seems reasonable for this kind of approach. It is possible to clean the machine vision inherent noise using a complementary filter that integrates the vehicle motion model estimations with the machine vision measurements. These integration results were verified by calculating their RMS error from the RTK-GPS position and the FOG heading. As a result, the RMS error of the heading was reduced from 0.75 deg. to 0.42 deg.; and the lateral position RMS error was reduced from 0.028 m to 0.024 m. Thanks to this improvement, these integration results can be applied to a smart agricultural vehicle; producing a method capable of performing automatic navigation from crop row detection with increased accuracy.

At the same time; thanks to the unique capabilities of the 2-in-1 camera, it is possible to build a field map that covers more crop rows than other mapping methods. An inverse perspective transformation in combination with the RTK-GPS coordinates gave as a result a map that covers up to eleven crop rows with a resolution good enough to detect the absence of plants in a specific crop row. The map precision was calculated from the camera calibration parameters in order to obtain the theoretical error. The calculated average lateral deviation of the mapped crop rows was 0.023 m. Thanks to the Fujifilm 2-in-1 camera wide-angle lens; and since the resulting mosaic map covers up to eleven crop rows, this method is ideal to use in practical applications like spraying, avoiding to travel additional paths.

Chapter 6 Research summary

This chapter gives a summary of this research, highlighting the most important aspects of this thesis chapter by chapter.

6.1 Introduction

This chapter presents concepts concerning agricultural automation. It briefly describes the autonomous navigation systems for agricultural vehicles as a solution for current agricultural problems in Japan. It also introduces the motivation of this work and the research objectives pursued. The research background introduces the fact that in Japanese society, aging and population decline have been progressing. In order to protect food production in Japan, encouraging the development of technologies in the field of agriculture automation such as autonomous guidance systems for agricultural vehicles seems to be an effective strategy to deal with the dwindling farming labor force, in addition to increase production efficiency and safer operation.

Several technologies used in agriculture are described. By implementing these technologies, agriculture automation has achieved a rapid development in the execution of all the agricultural tasks. For example, some autonomous navigation systems for agricultural vehicles are needed to prevent agricultural machinery from rollover situations. The Laboratory of Vehicle Robotics (VeBots) of Hokkaido University School of Agriculture has succeeded conducting researches on unmanned tractors and developing new technologies. Sensors are installed on a tractor, and its position is measured by satellite (GPS). A computer commands its movement, replacing manual operation and thus giving birth to robot tractors.

To improve autonomous navigation systems for agricultural vehicles it is possible to apply an area of engineering known as *Terramechanics*; which describes the vehicle's tire-soil interactions. Therefore, a precise vehicle model that does not depend on several parameters can fill the gaps existing in current models applied to autonomous vehicle development. It is also possible to combine this model with machine vision.

Therefore, this thesis purpose is to enhance the navigation performance of an agricultural vehicle by integrating a nonlinear vehicle motion model with a unique machine vision system. To achieve this purpose, this thesis has two objectives. The first one is to use previous researches results concerning the vehicle's tire dynamic properties in order to improve the vehicle motion model estimations. The second one is to use this improved motion model in combination with a machine vision method capable of provide crop row detection with increased accuracy. To integrate the best aspects of the vehicle motion model estimations with the machine vision measurements, data fusion technique was used.

To fulfill the first objective of this thesis, the measurements of the tire's lateral forces as a function of the tire's slip angle of a test vehicle traveling on a concrete surface are reported. A description about how these tire parameters can be used to improve the motion model of the test vehicle is given as well. To fulfill the second objective of this thesis, an explanation of the new machine vision method proposed is given.

It is introduced how experiments were conducted using a test vehicle consisting of a conventional tractor equipped with a Real-Time Kinematic Global Positioning System (RTK-GPS), a Fiber Optic Gyroscope (FOG), a Potentiometer and a new type of camera developed by Fujifilm Corporation.

6.2 Research platform

This chapter describes the equipment used in this research. The research platform implemented in the Kubota MD77 tractor used for the field experiments is introduced. Both the sensors and the test vehicle's technical specifications are included. The test vehicle was also equipped with an RTK-GPS and an FOG. An on-board computer process the data from the camera, RTK-GPS and FOG.

The RTK-GPS (Topcon, Legacy-E) provides the position, direction of travel and speed of the vehicle. The low latency configuration (update rate: 10 Hz, latency: 0.02 seconds, data link: 9600 Baud) was chosen for the RTK mode. The FOG (Japan Aviation Electronics Industry Ltd., JCS7402-A) provides the vehicle's angular rates (accuracy $\leq \pm$ 5 deg. /s), attitude (accuracy $\leq \pm$ 0.15 deg.) and lateral acceleration (accuracy $\leq \pm$ 1.5 m/s^2) readings. A 10 k Ω Potentiometer (Midori Precisions, CPP-60, linearity $\pm$0.05%) attached to the kingpin of one of the steering wheels provides the steering angle (alignment error: $\pm$3.2 deg.).

The bifocal imaging device is a non-commercial camera developed by Fujifilm Corporation. The innovation of the device consist of simultaneous shooting a wide-angle image and a telephoto image. Due to its capability of shooting two images at the same time, the developer refers to it as the Fujifilm 2-in-1 camera. In conclusion, this chapter described the equipment used during this research. The research platform used for the field experiments was described part by part. The test vehicle's main technical specifications were introduced. Each one of the sensors used in the experiments were described, highlighting their most relevant aspects. A reference for each one of the sensors' data sheets was also included.

6.3 Estimation of vehicle status using nonlinear vehicle motion model

This chapter presents the vehicle modeling framework used in this research. The handling dynamics of the test vehicle and therefore a brief development of important concepts associated with handling behavior are presented. The tire basics section explains that in terms of vehicle dynamics, the tire is one of the most important factors in determining how a vehicle responds to driver inputs. It is explained how in order for a tire to produce the traction forces that are required to accelerate a vehicle in a straight line or around a corner, it is necessary for the tire to develop a slip angle. It is shown how the traction forces and the normal load dependency are concepts critical in the design of an agricultural vehicle for specific handling behavior. It is shown that the area of vehicle handling primarily deals with motions within the plane of the road surface; which include lateral and longitudinal speeds, and the yaw rate. Therefore, in this thesis, the most common model associated to the bicycle model is introduced.

The vehicle dynamics are explained thru the description of the tire mechanics. This chapter also presents a brief description of the most common vehicle control strategies introducing that understanding how vehicles behave in the field can be achieved thanks to the vehicle motion model, which gives an estimation of the vehicle's position and heading (described by the kinematic model) taking into account the lateral forces acting on the vehicle (described by the dynamic model). It is possible to measure the vehicle's tire dynamic properties in field tests; namely the tire's lateral forces F_y as a function of the tire's slip angle α and describe them using a regression model in order to account for the vehicle's lateral offset caused by the tire-soil interaction. This gives as a result a regression

model that accounts for the hysteresis behavior of the F_y/α relation. A description about how these tire parameters can be used to improve the motion model estimations of the test vehicle is given as well.

As a result, the RMS error of the modeled body sideslip angle β_{REG} was reduced from 5.0 deg. to 3.4 deg.; and the modeled yaw rate ω_{REG} RMS error was reduced from 7.3 deg/s to 4.6 deg/s. Then, the vehicle's dynamic parameters β_{REG} and ω_{REG} were computed into the kinematic model to obtain estimations of the heading φ_{REG} and position $(x, y)_{REG}$ of the vehicle. It was verified that the tire dynamic properties described by this regression model technique can be applied to obtain better estimations of the vehicle's position and heading over time; verified by calculating the RMS error from the real position measured by the RTK-GPS and the real heading measured by the FOG. The model estimations without using the tire dynamic properties had an RMS error of 0.059 m for lateral position y_{DYN} and 2.8 deg. for heading φ_{DYN}; whereas the model estimations using the tire dynamic properties had an RMS error of 0.022 m for lateral position y_{REG} and 1.2 deg. for heading φ_{REG}. However, these estimations alone are not accurate enough to guide the vehicle through a field because there is a wide range of environment factors present in the tire-soil interaction; like soil moisture and cone index, which change from field to field and cannot be predicted by the vehicle motion model. Therefore, it is necessary to integrate the vehicle motion model estimations with some sensing method such as machine vision.

In summary, the tire dynamic properties described by a regression model technique allowed to obtain better estimations of the vehicle motion model, reducing its RMS error by 30%.

6.4 Image processing algorithm development of a machine vision with both wide-angle and telephoto images

This chapter presents the image processing algorithms used in this research. The algorithms are briefly described with examples. It is briefly explained how image processing techniques can be used to enhance agricultural practices, by improving accuracy and consistency of processes while reducing farmers' manual monitoring.

The machine vision method implemented a new type of camera developed by Fujifilm Corporation. This 2-in-1 camera can shoot high definition wide-angle and telephoto images simultaneously. The

camera was mounted on the top of the test vehicle, focused on the field surface from an inclined angle in order to calculate the vehicle's heading φ and lateral position y from the crop rows covered by the wide-angle images by using an image recognition algorithm.

The OpenCV (Open Source Computer Vision) library is introduced; due to the fact that all the algorithms used in this research were developed in this library. Image acquisition is explained establishing that a digital image is nothing more than a 2D matrix containing all the intensity values of its pixel points. Each pixel point position in the image matrix is then identified by a pair of Cartesian coordinates $p(x,y)$. The RGB color space is introduced as the most popular, because it resembles how the human eye builds up colors. Its base colors are Red, Green and Blue. Each of the color components has its own valid domains; which makes it necessary to define the data type used. It is explained that for this research the HSV color space was chosen for color filtering instead of the RGB color space because it provided a better contrast for different illumination conditions.

The Binary threshold is explained in terms of a simple segmentation method. Segmentation methods are useful to separate out regions of an image corresponding to objects of interest; in the case of this research, the objects of interest are the crop rows. Morphological operations are defined as a set of operations that process images based on shapes; a structuring element is applied to an input image to generate a smoothed output image. The Erosion and Dilation operations are explained with examples. The contours and geometric centers calculation is briefly described, and the line fitting to each set of the point's coordinates by using the least squares method is explained as well.

The center of gravity section describes how in order to improve the accuracy of the crop row detection method it is possible to obtain correction data from the telephoto image of the 2-in-1 camera. The downscaling section shows how to take the detected crop row central line obtained from the telephoto image and downscale it into the wide-angle image. The camera calibration method was introduced; giving as a result the pixel-distance relation that allowed transformation of coordinates between the camera's image plane and the ground plane. The image recording section explained how series of video frames were acquired in the soybean test field, located in the experimental farm of Hokkaido University.

From the figures included it was found that the traditional method for detecting the crop rows from a wide-angle image can be improved by the use of correction data from a telephoto image; mainly

because weeds growing beside the crop rows and natural variation in the plant growth affect the accuracy of this image recognition algorithm.. Thanks to the telephoto image's increased resolution, accuracy of the image recognition algorithm can be improved by fusing the wide-angle image data with the telephoto image data using a complementary filter, reducing the lateral position deviation range from 0.061 m to 0.028 m. In summary, thanks to the unique capabilities of the 2-in-1 camera, the crop row detection lateral offset deviation was reduced over 40% by fusing the wide-angle image data with the telephoto image data. Although results display increased accuracy for the lateral position calculated from crop row detection, the machine vision measurements still have some inherent noise that can affect the navigation performance of the vehicle.

6.5 Application to automatic navigation and crop mapping

This chapter describes how to apply the machine vision system integrated with the vehicle motion model to a smart agricultural vehicle capable of performing automatic navigation and crop mapping. To achieve automatic navigation, it is explained how to measure the vehicle's heading φ from the detected crop rows. A complementary filter was used in order to integrate the best aspects of the vehicle motion model estimations with the machine vision measurements. In other words, this chapter combines models described in chapter 3 *Estimation of vehicle status using nonlinear vehicle motion model* with the machine vison method described in chapter 4 *Image processing algorithm development of a machine vision with both wide-angle and telephoto images*.

The heading detection section showed that a high precision sensor like the FOG has an accuracy that is almost impossible to match with the heading angle φ obtained from machine vision, but the results have an error that seems reasonable for this kind of approach. It is possible to clean the machine vision inherent noise using a complementary filter that integrates the vehicle motion model estimations with the machine vision measurements. These integration results were verified by calculating their RMS error from the RTK-GPS position and the FOG heading. As a result, the RMS error of the heading φ was reduced from 0.75 deg. to 0.42 deg.; and the lateral position y RMS error was reduced from 0.028 m to 0.024 m. Thanks to this improvement, these integration results can be applied to a smart agricultural vehicle; producing a method capable of performing automatic navigation from crop row detection with increased accuracy. In summary, in order to clean the inherent noise from the machine

vision measurements, the improved estimations of the vehicle motion model were integrated with the machine vision data; reducing the RMS error around 20%.

In addition; this research goes a step beyond the automatic navigation from crop row detection adding a crop mapping method. This chapter describes the methodology to build a crop map taking advantage of the Fujifilm 2-in-1 camera wide-angle lens. Limitations of the current crop mapping methods are explained in order to justify the purpose of this method. Thanks to the unique capabilities of the 2-in-1 camera, it is possible to build a field map that covers more crop rows than other mapping methods. An inverse perspective transformation in combination with the RTK-GPS coordinates gave as a result a map that covers up to eleven crop rows with a resolution good enough to detect the absence of plants in a specific crop row.

Results of the proposed approach are discussed according to the theoretical error found between the processed images and the RTK-GPS path data. The map precision was calculated from the camera calibration parameters in order to obtain the theoretical error. The calculated average lateral deviation of the mapped crop rows was 0.023 m. Thanks to the Fujifilm 2-in-1 camera wide-angle lens; and since the resulting mosaic map covers up to eleven crop rows, this method is ideal to use in practical applications like spraying, avoiding to travel additional paths. It is concluded that the mapping method has a reasonable error and also has a clear advantage over mapping methods that employ conventional cameras.

6.6 Conclusions

In order to protect food production in Japan, encouraging the development of technologies in the field of agriculture automation such as autonomous navigation systems for agricultural vehicles has proven to be an effective strategy to deal with the dwindling farming labor force. The purpose of the research was to enhance the navigation performance of an agricultural vehicle by integrating a nonlinear vehicle motion model with a unique machine vision system. To integrate the best aspects of the vehicle motion model estimations with the machine vision measurements, data fusion technique was used. Experiments were conducted using a test vehicle consisting of a conventional tractor equipped with a Real-Time Kinematic Global Positioning System (RTK-GPS), a Fiber Optic Gyroscope (FOG), a Potentiometer and a new type of camera developed by Fujifilm Corporation.

The tire dynamic properties described by a regression model technique allowed to obtain better estimations of the vehicle motion model, reducing its RMS error by 30%. Thanks to the unique capabilities of the 2-in-1 camera, the crop row detection lateral offset deviation was reduced over 40% by fusing the wide-angle image data with the telephoto image data. In order to clean the inherent noise from the machine vision measurements, the improved estimations of the vehicle motion model were integrated with the machine vision data; reducing the RMS error around 20%. These integration results can be applied to a smart agricultural vehicle to build a field map with an average lateral deviation of 0.023 m for the mapped crop rows.

REFERENCES

Anderson R., Bevly D. M., Estimation of Tire Cornering Stiffness Using GPS to Improve Model Based Estimation of Vehicle States, Proc. IEEE Intell. Vehicles Symposium (2005) 801-806.

Andreasen, C., Rudemo, M., Sevestre, S., 1997. Assessment of weed density at an early stage by use of image processing. Weed Res. 37, 5–18.

Agoston, M. K., 2005. Computer Graphics and Geometric Modelling. Springer Science & Business Media, New York.

Åstrand, B., Baerveldt, A. J., 2005. A vision based row-following system for agricultural field machinery. Mechatronics 15 (2005), 251-269. doi:10.1016/j.mechatronics.2004.05.005.

Baffet G., Charara A., Stephant J., Sideslip angle, lateral tire force and road friction estimation in simulations and experiments, Proc. of the 2006 IEEE International Conference on Control Applications (2006) 903-908.

Baffet G., Charara A., Lechner D., Estimation of vehicle sideslip, tire force and wheel cornering stiffness, Control Eng. Pract., Vol. 17, Issue 11, (2009) 1255-1264.

Barnard, S.T., 1983. Interpreting perspective images. Artif. Intell. 1983, 21, 435-462.

Bengochea-Guevara, J. M., Conesa-Muñoz, J., Andujar, D., Ribeiro, A., 2016. Merge fuzzy visual servoing and GPS-based planning to obtain a proper navigation behavior for a small crop-inspection robot. Sensors 2016, 16, 276-299. doi:10.3390/s16030276.

Bevilacqua, A., Gherardi, A., Carozza, L., 2008. Automatic perspective camera calibration based on an incomplete set of chessboard markers. Sixth Indian Conference on Computer Vision, Graphics & Image Processing. IEEE Computer Society. 978-0-7695-3476-3/08. DOI 10.1109/ICVGIP.2008.10.

Bevly D. M., 2004. Global Positioning System (GPS): A Low-Cost Velocity Sensor for Correcting Inertial Sensor Errors on Ground Vehicles, J. of Dyn. Syst., Measurement, and Control, Vol. 126, (2004) 255-264.

Bevly, D. M., Gerdes, J. C., Parkinson, B. W., 2002. A new yaw dynamic model for improved high speed control of a farm tractor. J. of Dyn. Syst., Measurement, and Control, Vol. 124, (2002) 659-667.

Bevly, D., Daily, R., Travis, W., 2006. Estimation of Critical Tire Parameters Using GPS Based Sideslip Measurements, SAE Technical Pap. (2006).

Billingsley, J.; Schoenfisch, M., 1997. The successful development of a vision guidance system for agriculture. Comput. Electron. Agr. 1997, 16, 147-163.

Blasch, E., Kadar, I., Salerno, J., Kokar, M. M., Das, S., Powell, G. M., Corkill, D.D., Ruspini, E. H., 2006. Issues and Challenges in Situation Assessment (Level 2 Fusion). Journal of Advances in Information Fusion. Vol. 1 No. 2. December 2006, 122-139.

Bosch, A., Mu˜noz, X., Freixenet, J., 2007. Segmentation and description of natural outdoor scenes. Image Vision Comput. 25, 727–740.

Bradski, G., Kaehler, A., 2008. Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media, USA, 2008.

Brahmbhatt, S., 2013. Practical OpenCV. Apress, New York.

Burgos-Artizzu, X. P., Ribeiro, A., Guijarro, M., Pajares, G., 2011. Real-time image processing for crop/weed discrimination in maize fields. Comput. Electron. Agr. 75, 337-346.

Burhaumudin M. S., M. S. Pakharuddin, J. Hishamuddin, Rahman R. A., S. Syabillah, Modeling and Validation of Magic Formula Tire Model, International Conference on Automot., Mechanical and Mater. Eng. (2012) 113-117.

Burks, T., Shearer, S., Heath, J., Donouhue, K., 2005. Evaluation of neural network classifiers for weed species discrimination. Biosyst. Eng. 91 (3), 293–304.

Castanedo, F., 2013. A review of data fusion techniques. The Scientific World J., Volume 2013, Article ID 704504, 19 pages. http://dx.doi.org/10.1155/2013/704504.

Earl, R., Wheeler, P., Blackmore, B., Godwin, R., 1996. Precision farming: the management of variability. Landwards 51 (4), 18–23.

ESA, 2010. What is Galileo? Available at: http://www.esa.int/esaNA/GGGMX650NDC_galileo_0.html.

FSA-IAC, 2010. GLONASS. Available at: http://www.glonass-ianc.rsa.ru/pls/htmldb/f?p=202:1:2936089491394849.

Facciolo, G., Limare, N., Meinhardt, E., 2014. Integral Images for Block Matching. Image Processing On Line 4 (2014), 344-369. http://dx.doi.org/10.5201/ipol.2014.57.

Fisher, R., Perkins, S., Walker, A., Wolfart, E., 2004. HYPERMEDIA IMAGE PROCESSING REFERENCE. (https://homepages.inf.ed.ac.uk/rbf/HIPR2/hipr_top.htm).

Gerhards, R., Christensen, S., 2003. Real-time weed detection, decision making and patch spraying in maize, sugarbeet, winter wheat and winter barley. Weed Res. 43, 385–392.

Gerhards, R., Oebel, H., 2006. Practical experiences with a system for site-specific weed control in arable crops using real-time image analysis and gps-controlled patch spraying. Weed Res. 46, 185–193.

Gonzalez, R. C., Woods, R. E., 2002. Digital image processing, second ed. Prentice Hall, Upper Saddle River, N.J.

Gottschalk, R., Burgos-Artizzu, X.P., Ribeiro, A., Pajares, G., Sánchez-Miralles, A., December 2008. Real-time image processing for the guidance of a small agricultural field inspection vehicle. In: 15th International Conference on Mechatronics and Machine Vision in Practice (M2VIP'08), Auckland, New Zealand.

Granitto, P., Verdes, P., Ceccatto, H., 2005. Large-scale investigation of weed seed identification by machine vision. Comput. Electron. Agr. 47, 15–24.

Green, G., 1828. An Essay on the Application of Mathematical Analysis to the Theories of Electricity and Magnetism. Nottingham, England: T. Wheelhouse, 1828.

Guerrero, J. M., Guijarro, M., Montalvo, M., Romeo, J., Emmi, L., Ribeiro, A., Pajares, G., 2013. Automatic expert system based on images for accuracy crop row detection in maize fields. Expert Systems with Applications 40, 656–664.

Hall, D. L., Llinas, J., 1997. Introduction to Multisensor Data Fusion. Proceedings of the IEEE. 85 (1): 6-23.

Hashimoto Y., Murase H., Morimoto T., Torii T., Intelligent systems for agriculture in Japan, IEEE Control Syst., Vol.21, Issue 5, (2001) 71-85.

Hassen, N. S., Sidik, N. A. C., Sheriff, J. M., 2014. Advanced Techniques for Reducing Spray Losses in Agrochemical Application System. Life Science Journal, 11(3), 56-66.

Hemming, J., Rath, T., 2001. Computer-vision based weed identification under field contions using controlled lighting. J. Agr. Eng. Res. 78 (3), 233–243.

Higgins, W. T., 1975. A comparison of complementary and kalman filtering. IEEE Trans. Aerospace and Electronic Systems, Vol. AES-11, no. 3, May 1975.

Ji, R., Qi, L., 2011. Crop-row detection algorithm based on Random Hough Transformation. Mathematical and Computer Modelling 54 (2011), 1016–1020.

Jiang, G., Wang, Z., Liu H., 2015. Automatic detection of crop rows based on multi-ROIs. Expert Systems with Applications 42, 2429-2441.

Khaleghi, B., Khamis, A., Karray, F.O., Razavi, S.N., 2013. Multisensor data fusion: a review of the state-of-the-art. Inf. Fusion, 14 (4) (2013), pp. 28-44.

Kise, M., Zhang Q., 2008. Development of a stereovision sensing system for 3D crop row structure mapping and tractor guidance. Biosystems Eng. 101 (2008), 191-198.

Kise, M., Zhang, Q., Rovira-Más, F., 2005. A stereovision-based crop row detection method for tractor-automated guidance. Biosystems Eng. (2005) 90 (4), 357–367.

Kondo N., Monta M., Noguchi N., Agricultural Robots: Mechanisms and Practice, Kyoto University Press and Trans Pacific Press, 2011.

Kropff, M., Wallinga, J., Lotz, L., 1997. Modelling for precision weed management. In: Precision Agriculture: Spatial and Temporal Variability of Environmental Quality. Wiley, Chester, pp. 182–204.

Lee, W., Slaughter, D., Giles, D., 1996. Development of a machine vision system for weed control using precision chemical application. In: Proceedings of International Conference on Agricultural Machinery Engineering ‘96 , Seoul, Korea, pp. 802–811.

Lefèvre, H., 1993. The Fiber-Optic Gyroscope, ARTECH HOUSE, INC. 1993, ISBN 0-89006-537-3.

Liburdi, A., 2010. Development of a scale vehicle dynamic test bed. Electronic Theses and Dissertations. 195. https://scholar.uwindsor.ca/etd/195

Liljedahl J.B., Turnquist P.K., Smith D.W., Hoki M., Tractors and their Power Units, fourth ed., Van Nostrand Reinhold, New York, 1989.

Mallot, H. A., Bülthoff, H. H., Little, J.J., Bohrer, S., 1991. Inverse perspective mapping simplifies optical flow computation and obstacle detection. Biol. Cybern. 64, 177-185.

Noguchi, N., Reid, J. F., Will, J., Benson, E.R., Stombaugh, T.S., 1998. Vehicle Automation System Based on Multi-sensor Integration. An ASAE Meeting Presentation, Paper No. 983111, UILU-ENG-98-7021.

Nguyen, T. T., Slaughter, D. C., Hanson, B. D., Barber, A., Freitas, A., Robles, D., Whelan E., 2015. Automated Mobile System for Accurate Outdoor Tree Crop Enumeration Using an Uncalibrated Camera. Sensors 15, 18427-18442. doi:10.3390/s150818427.

Okamoto, H., Hamada, K., Kataoka, T., Terawaki, M., Hata, S., 2002. Automatic guidance system with crop row sensor. Proceedings of Automation Technology for Off-Road Equipment, 307-316. Chicago, USA. 26-27 July 2002.

Olsen, H. J., 1995. Determination of row position in small-grain crops by analysis of video images. Comput. Electron. Agr. 12, 147-162.

Onyango, C., Marchant, J., 2003. Segmentation of row crop plants from weeds using colour and morphology. Comput. Electron. Agr. 39, 141–155.

Ospina, R., Noguchi, N., 2016. Determination of tire dynamic properties: Application to an agricultural vehicle. Engineering in Agriculture, Environment and Food 9, 123-130.

Ospina, R., Noguchi, N., 2018. Alternative method to model an agricultural vehicle's tire parameters. Engineering in Agriculture, Environment and Food 11, 9-18.

Pacejka H.B., Bakkerb E., The Magic Formula Tyre Model, Vehicle Syst. Dyn.: International J. of Vehicle Mechanics and Mobil., Vol. 21, Suppl. 001, (1992) 1-18.

Pacejka H.B., Tyre and Vehicle Dynamics, second ed., Butterworth-Heinemann, Oxford, 2006.

Pajares, G., García-Santillán, I., Campos, Y., Montalvo, M., Guerrero, J. M., Emmi, L., Romeo, J., Guijarro, M., Gonzalez-de-Santos, P., 2016. Machine-vision systems selection for agricultural vehicles: a guide. J. Imaging 2016, 2, 34-65. doi:10.3390/jimaging2040034.

Pepy, R., Lambert, A., Mounier, H., 2006. Path planning using a dynamic vehicle model. Inf. and Commun. Technol. 1, 781-786.

Pinto, F. A. C., Reid, J. F., Zhang, Q., Noguchi, N., 2000. Vehicle guidance parameter determination from crop row images using principal component analysis. J. of Agric. Eng. Res. 75, 3, 257-264.

Pla, F., Sanchiz, J. M., Marchant, J. A., Brivot, R., 1997. Building perspective models to guide a row crop navigation vehicle. Image and Vision Computing 15 (1997), 465-473.

Romeo, J., Guerrero, J. M., Montalvo, M., Emmi, L., Guijarro, M., Gonzalez-de-Santos, P., Pajares, G., 2013. Camera sensor arrangement for crop/weed detection accuracy in agronomic images. Sensors 13, 4348-4366. doi:10.3390/s130404348.

Romeo, J., Pajares, G., Montalvo, M., Guerrero, J. M., Guijarro, M., Ribeiro, A., 2012. Crop Row Detection in Maize Fields Inspired on the Human Visual Perception. The Scientific World Journal, Volume 2012, Article ID 484390, 10 pages. doi:10.1100/2012/484390.

Rovira-Más, F., Zhang, Q., Reid, J. F., Will, J. D., 2005. Hough-transform-based vision algorithm for crop row detection of an automated agricultural vehicle. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, Volume: 219 issue: 8, 999-1010.

Sainz-Costa, N., Ribeiro, A., Burgos-Artizzu, X. P., Guijarro, M., Pajares, G., 2011. Mapping wide row crops with video sequences acquired from a tractor moving at treatment speed. Sensors 11, 7095-7109. doi:10.3390/s110707095.

Saxena, L., Armstrong, L., (2014). A survey of image processing techniques for agriculture. Proceedings of Asian Federation for Information Technology in Agriculture, 401-413.

Slaughter, D. C., Giles, D. K., Downey, D., 2008. Autonomous robotic weed control systems: a review. Comput. Electron. Agr. 61, 63-78.

Søgaard, H. T., Olsen, H. J., 2003. Determination of crop rows by image analysis without segmentation. Comput. Electron. Agr. 38, 141-158.

Sonka, M., Glavac, V., Boyle, R., 2008. Image Processing, Analysis, and Machine Vision, third ed. Thomson Learning, Toronto.

Srinivasan, A., 2006. Handbook of Precision Agriculture. Principles and Applications. The Haworth Press, New York.

Stewart, R.M., Parks, R.J., 1957. Degenerate solutions and algebraic approach to the multiple input linear filter design problems. IRE Trans. Circuit Theory, vol. CT4, pp. 10-14.

Suzuki, S., Abe, K., 1985. Topological Structural Analysis of Digitized Binary Images by Border Following. CVGIP 30 1, pp 32-46 (1985).

Szeliski, R., 2010. Computer Vision: Algorithms and Applications. Springer Science & Business Media, New York.

Thorp, K., Tian, L., 2004. A review on remote sensing of weeds in agriculture. Precis. Agr. 5, 477–508.

Tillett, N. D., Hague, T., Miles, S. J., 2001. A field assessment of a potential method for weed and crop mapping on the basis of crop planting geometry. Comput. Electron. Agr. 32, 229-246.

Vali, V., Shorthill, R.W., 1976. Fiber Ring Interferometer, Appl Optics 15, 1976.

Van Evert, F.K., Van Der Heijden, G., Lotz, L., Polder, G., Lamaker, A., De Jong, A., Kuyper, M., Groendijk, E., Neeteson, J., Van Der Zalm, T., 2006. A mobile field robot with vision-based detection of volunteer potato plants in a corn crop. Weed Technol. 20, 853–861.

Vioix, J., Douzals, J., Truchetet, F., Assémat, L., Guillemin, J., 2002. Spatial and spectral methods for weed detection and localization. EURASIP JASP 7, 679–685.

Woebbecke, D., Meyer, G., VonBargen, K., Mortensen, D., 1995. Color indices for weed identification under various soil, residue, and lighting conditions. Trans. ASAE 38 (1), 271–281.

Wong J.Y., Theory of Ground Vehicles, second ed., John Wiley & Sons, Inc., New York, 1993.

Yeh, E. C., Hsu, J. C., Lin, R. H., 1994. Image-Based Dynamic Measurement for Vehicle Steering Control. IEEE Proceedings of 94' Intelligent Vehicle Symposium, Paris, France, pp 326-332.

Zakaria, M. A., Zamzuri, H., Mamat, R., Mazlan, S. A., 2013. A path tracking algorithm using future prediction control with spike detection for an autonomous vehicle robot. Int. j. adv. Robot. Syst., 2013, Vol. 10, 309:2013. DOI: 10.5772/56658.

Zhao, X., Hong, T., Chen, H., 2011. Model-based vehicle trajectory and its properties for automatic steering systems. Electrical Power Systems and Computers, LNEE99, 845-851.

Zhang, Q., Reid, J, F., Noguchi, N., 1999. Agricultural vehicle navigation using multiple guidance sensors. UILU-ENG-99-7013, 1999.

Zheng, L., Zhang, J., Wang, Q., 2009. Mean-shift-based color segmentation of images containing green vegetation. Comput. Electron. Agr. 65, 93–98.

農用トラクター（乗用型）検査成績表

合格番号	95056	型式名	クボタMD-77	平成8年1月9日
		依頼者名	株式会社 クボタ	生物系特定産業技術
		住所	大阪府大阪市浪速区敷津東1丁目2番47号	研究推進機構

- 4輪駆動
- 大型特殊自動車
- 機関呼称出力：56.6kW {77PS}／2400rpm
- 走行速度段数：前進16段　後進16段（シャトル変速）
- PTO変速段数：2段
- グランドPTO付き
- 電子油圧制御装置付き
- 旋回時前輪増速機構付き
- パワーステアリング付き
- 装着安全キャブ：型式名　クボタIC87（合格番号：95065）

I 主要諸元・構造

1．機体の大きさ

- 全長：3.88m（前輪タイヤ～下部リンク後端）
- 全幅：1.96m（後輪最外側）
  試験時の前輪輪距1.43m
  試験時の後輪輪距1.52m
- 全高：2.60m（安全キャブ上端）
- 質量：3265kg（付加重錘なし）
  前輪1260kg　後輪2005kg
- 軸距：2.30m
- 輪距：前輪1.33，1.42，1.43，1.52m（4段階）
  後輪1.42，1.52，1.62，1.72，1.82
  1.92m　（6段階）
- タイヤ：前輪9.5-24-6 PR
  後輪16.9-30-6 PR
- PTO軸高さ：0.68m
- PTO軸端～下部リンク後端：0.65m

2．機　関

- 種類：水冷4サイクル4気筒ディーゼル
- 呼称出力：56.6kW {77PS}／2400rpm
- 総行程容積：4.329 l
- 燃焼室形式：直接噴射式
- 過給機：なし

3．伝動装置

- 主クラッチ形式：乾式単板
- 主変速：8段
- 副変速：2段
- 前進：16段（1.2～29.1km／h）
- 後進：16段（1.2～29.4km／h）
- デフロック装置：あり（前後輪）

4．PTO

- クラッチ形式：独立型
- 規　格：6スプライン軸径35mm
- 回転数（機関定格回転数のとき）：
  637，1085rpm

5．けん引装置

- 形式：スウィングドローバ

6．作業機昇降装置

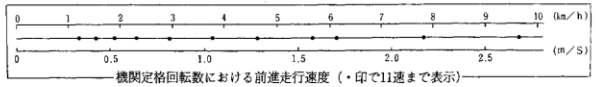
- 制御方式：ポジション及び
  ドラフトコントロール
- 油圧外部取出口：PT1/2，4個
- 作業機装着装置：3点リンク2形

7．かじ取装置

- 形式：油圧式（全油圧式）

8．装着安全キャブ

- 型式名：クボタIC87　（合格番号：95065）



機関定格回転数における前進走行速度（・印で11速まで表示）

II 検 査 成 績

1．PTO性能

- トルクバックアップ比：1.21
- 弾 性 値：2.04
- 最大出力時：

出力	燃料消費率	PTO軸回転数	排気煙濃度
53.7kW {73.0PS}	274g／kW・h {202g／PSh}	1072rpm	27%

2．けん引性能（コンクリート上）

条件	けん引力	けん引出力	速度
最大けん引力	25.02kN {2551kgf}	9.51kW {12.9PS}	1.37km／h（2速）
7.5km／hに最も近い速度段	21.48kN {2190kgf}	41.8kW {56.8PS}	7.01km／h（10速）
最大けん引出力	14.43kN {1471kgf}	44.3kW {60.2PS}	11.64km／h（12速）

3．作業機昇降装置性能

- リリーフ弁設定圧力：19.6MPa {200kgf／cm²}
- 油圧ポンプ性能：

条件	流量	圧力	出力
最大出力	53.9 l／min	15.7MPa {160kgf／cm²}	14.1kW {19.2PS}
リリーフ弁設定圧力の90%	30.7 l／min	17.7MPa {180kgf／cm²}	9.05kW {12.3PS}

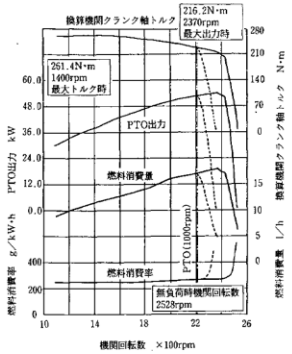
- 揚力：
  アシストシリンダなしの時

条件	揚力
下部リンクヒッチ点	23.05kN {2350kgf}
フレーム上	19.96kN {2035kgf}

アシストシリンダ1個の時（オプション）

条件	揚力
下部リンクヒッチ点	28.88kN {2945kgf}
フレーム上	25.35kN {2585kgf}

4．騒音（運転者の耳もと）：78dB(A)



※ 調速レバー位置
実線：全開位置
点線：標準PTO回転数で最大出力の得られる位置

図 PTOの性能曲線

5．静的横転倒角：38°

6．最小旋回半径（コンクリート上）

ブレーキ使用時：3.35m

7．安全装備

可動部・高温部の防護カバー，昇降部の落下防止装置，その他の装置

8．その他

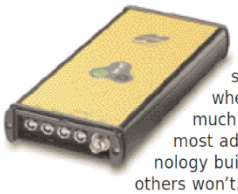
旋回時前輪増速機構及び電子油圧制御装置は円滑に作動した。

トルクバックアップ比：最大トルクと最大出力時トルクの比。大きい方が良好。
弾性値：（最大出力時と最大トルク時の機関回転数の比）×（トルクバックアップ比）。大きい方が良好。
排気煙濃度：排気ガスの黒煙の濃度を0～100%の範囲で示す。小さい方が黒煙が少ない。
最大けん引力：コンクリート上でスリップ率15%時のけん引力又はその速度段の最大けん引出力時のけん引力のどちらか小さい方の値を最大けん引力としている。
揚力：最大揚力を実際のリリーフ弁設定圧力の90%の圧力に換算した実用的揚力値。
騒音：無負荷時走行速度が7.5km／hに近い速度段で，けん引負荷をかけた時の最大騒音レベル値。

III 付　　記

本機は，高性能農業機械等の試験研究，実用化の促進及び導入に関する基本方針（平成5年8月30日農林水産大臣公表）に定められたトラクターの第III類に属するものである。

Appendix B



With a Legacy-E GPS+ receiver, fast and accurate solutions don't rely on when or where you're working, or how much time have. It has the world's most advanced GPS+ receiving technology built-in, ready to work where others won't.

At its core is our **Paradigm** chip featuring 40 universal super channels that can each track all signals of either L1 or L2 GPS and GLONASS frequencies —and up to 20 GPS+ satellites at once, the maximum available at any one time! It incorporates our new innovations in signal processing, **multipath mitigation** and **co-op tracking**, making Topcon GPS+ the best in the field for under-canopy and low signal strength reception.

Our unique Cinderella feature activates GPS L1+L2 and GLONASS reception every other Tuesday for 24 hours so you can experience the real power of Topcon GPS+. Permanently activating these frequencies and other options is easy with simple password commands entered via a PC. Options can even be added on a pay-per-use basis—only when you need it.

The MINimum INTERface keeps operation very simple. Just two function keys and two 3-color LED's are all that's needed to record data and check status. Up to four serial ports lets Legacy-E work with a variety of accessories in all types of configurations.

Topcon offers a full line of antennas, radios, post-processing software and all the other accessories and hardware that will help ensure maximum productivity and accuracy.

Powerful, simple, versatile. Topcon Legacy-E—no ordinary GPS receiver.

TOPCON

TOPCON POSITIONING SYSTEMS, INC.
800-443-4567
www.topcon.com

Topcon sells GPS products into the precision markets only. Go to www.topcongps.com for details.

Legacy-E Technical Data[1]

Description	40 channel integrated GPS+ receiver with MINTER interface.
Tracking Specifications	
Tracking Channels, standard	40 L1 GPS (20 GPS L1+L2+GLONASS on Cinderella[2] days)
Tracking Channels, optional	20 GPS L1+L2 (GD), 20 GPS L1 + GLONASS (GG), 20 GPS L1+L2+GLONASS (GGD)
Signals Tracked	L1/L2 C/A and P Code & Carrier
Performance Specifications (1 sigma)	
Baseline Accuracy	5mm + 0.5ppm
RTK (OTF) Accuracy	10mm + 1ppm
Cold Start	<60 seconds
Warm Start	<10 seconds
Reacquisition	<1 second
Power Specifications	
Battery	External (maximum of 2 ports)
Power input / consumption	6 to 28 volts DC / less than 3.3 watts
Continuous Operating Time	7 hours (typical w/2.3AH rechargeable battery)
GPS+ Antenna Specifications	
GPS / GLONASS Antenna	External
Antenna Type	Microstrip
Ground Plane	Antenna on a flat ground plane or Choke Ring
Radio Specifications	
Type	External, UHF/VHF radio modem
Base Power Output	0.5W/2.0W/35W
I/O	
Communication Ports	Serial (RS232), 4 maximum, 2 standard
Other I/O Signals	1pps, event marker, frequency input, frequency output
Status Indicator	2x3-color LED's, two-function keys (MINTER)
Control & Display Unit	External: Husky FS/2, FS/3, Ranger, 3rd Party
Memory & Recording	
Internal Memory	Up to 1 Gbyte
Raw Data Recording	Up to 20 times per second (20Hz)
Data Type	Code and Carrier fromL1 and L2, GPS and GLONASS
Data Output	
Real time data outputs	RTCM 104 version 2.3 and/or CMR2/CMR+
ASCII Output	NMEA 0183 version 2.3
Other Outputs	TPS format
Output Rate	Up to 20 times per second (20Hz)
Environmental Specifications	
Enclosure	Waterproof
Operating Temperature	-40°C to 55°C / -40°F to 130° F
Dimensions	W:240 x H:110 x D:35 mm / 9.45 x 4.33 x 1.38 in
Weight	0.6 kg / 1.32 lbs

Standard Configuration

- Legacy-E Receiver (0Mb)
- Cinderella GPS L2 + GLONASS activation
- 1 Hz Update Rate
- Co-op Tracking
- NMEA 0183 output
- User Defined Outputs
- MINTER Interface
- 2x RS232 Serial Ports
- 1x External Power Port
- Power Cables
- RS232 Cable

Optional Features

- GPS L2 and GLONASS
- Update rate 5Hz,10Hz, 20Hz
- RTK @ 1Hz, 5Hz, 10Hz, 20Hz
- Data Recording 4Mb to 1Gb
- CMR/RTCM input/output
- Advanced Multipath Reduction
- Frequency I/O
- Event Marker
- Two additional serial ports
- Additional power port
- WAAS
- RAIM

Common Accessories

- Topcon Antennas
  PG-A1 flat ground plane
  CR-3 choke-ring
  CR-4 choke-ring
- TopSURV software
- Topcon Tools software
- FC-1000 controller
- UHF Base or Rover radio kit
- 2.3AH rechargeable battery
- LitePole
- Tripod, tribrach & adapter
- Pinnacle software
- Carlson GPS software
- Survey Pro software
- Backpack, carrying case

1 Specifications are subject to change without notice. Performance specifications assume a minimum of 6 GPS or 7 GPS/GLONASS satellites above 15 degrees in elevation and adherence to procedures recommended by TPS in the appropriate manuals. In areas of high multipath, during periods of high PDOP and during periods of high Ionospheric activity performance may be degraded. Robust checking procedures are highly recommended in areas of extreme multipath or under dense foliage.

2 Cinderella feature activates GPS L2 and GLONASS reception at GPS midnight every other Tuesday for 24 hours.

P/N:7010-0425 Rev C Printed in USA 1/04

Appendix C

仕様 Specifications　外形寸法 Dimensions

性能 Performance		JCS7402-A	
		Analog Output	Digital Output(RS232C,RS422)
姿勢角 ロール角 ピッチ角 Each Angle	最大計測範囲/Range	±90°,±45°,±20°,±10°	Roll：±180°、Pitch：±90°
	出力電圧/Output	±10V	–
	分解能/Resolution	–	≦0.1°
	精度/Accuracy※1	≦±(0.2° + 1% of input) @ input≦±45°	≦±0.15° @ input ≦±10° ≦±(0.2° + 1% of input) @ input = ±10°～45°
相対方位角 Relative Azimuth Angle	最大計測範囲/Range	±180°,±90°,±45°,±20°	±180°
	出力電圧/Output	±10V	–
	分解能/Resolution	–	≦0.1°
	精度/Accuracy	≦±(1% of input + (2°/minute)) + 0.2°	≦±(1% of input + (2°/minute))
加速度 Acceleration (X,Y,Z)	最大計測範囲/Range	±19.6m/s²	
	分解能/Resolution	–	≦0.1m/s²
	精度/Accuracy※2	≦±1.5m/s²	≦±0.5m/s²
角速度 Rate (X,Y,Z)	最大計測範囲/Range	≦±100°/s	
	分解能/Resolution	–	≦0.1°/s
	精度/Accuracy※3	≦±11°/s	≦±5°/s

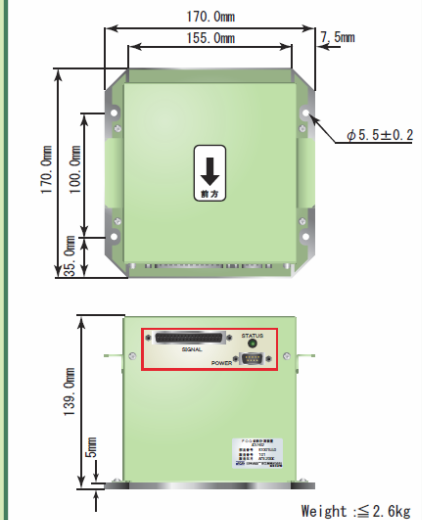
※1 水平方向の加速度がない、または正しい速度データが GPS より入力されている場合の精度です。 / This is the accuracy in case the sensors is not accelerated in horizontal direction or given precise data from GPS.
※2 10msec 間の平均加速度です。 /This is the mean acceleration for 10msec.
※3 10msec 間の平均角速度です。 /This is the mean angular rate for 10msec.

入出力 In/Out		JCS7402-A	
		Analog Output	Digital Output (RS232C,RS422)
信号形態 Signal	ボーレート/Baud Rate	–	19.2kbps／115.2kbps
	送信周期/Frequency	–	50Hz／200Hz
電源電圧/Supply Voltage		DC+24V(20～30V)	
消費電流/Current		≦1A	

耐環境性 Environment		JCS7402-A	
		Analog Output	Digital Output (RS232C,RS422)
温度環境 Temperature	作動/Operation	-10～+50℃	
	保存/Preservation	-20～+60℃	
振動環境 Vibration	作動/Operation	5～200Hz 19.6m/s²[2G]	
	保存/Preservation	5～200Hz 49m/s²[6G]	
衝撃環境/Shock		Non operation: 147m/s² [15G]	
湿度環境/Humidity		≦95%RH (Over the dew point)	

仕様 Specifications　外形寸法 Dimensions

JCS7402-A



Weight :≦2.6kg

Pin Assignment(SIGNAL) JCS7402-A

PIN	NAME	PIN	NAME
1	Roll Angle	20	Roll Angle GND
2	Pitch Angle	21	Pitch Angle GND
3	X Rate	22	Y Rate
4	Z Rate	23	Rate GND
5	X Accel	24	Y Accel
6	Z Accel	25	Accel GND
7	VALID(H)	26	VALID(L)
8	Relative Azimuth	27	Relative Azimuth GND
9	RX+ (RS422)	28	RX- (RS422)
10	TX+ (RS422)	29	TX- (RS422)
11	TEST	30	SG (RS422)
12	Realignment	31	TEST・Realignment GND
13	FAIL	32	FAIL GND
14	N/A	33	N/A
15	N/A	34	GPS RXD(RS232C)
16	GPS SG (RS232C)	35	GPS TXD(RS232C)
17	RXD (RS232C)	36	SG (RS232C)
18	TXD(RS232C)	37	1PPS
19	1PPS GND		

Pin Assignment(POWER) JCS7402-A

PIN	NAME	PIN	NAME
1	NC	6	+24V
2	+24V	7	+24V
3	+24V RTN	8	+24V RTN
4	+24V RTN	9	NC
5	NC		

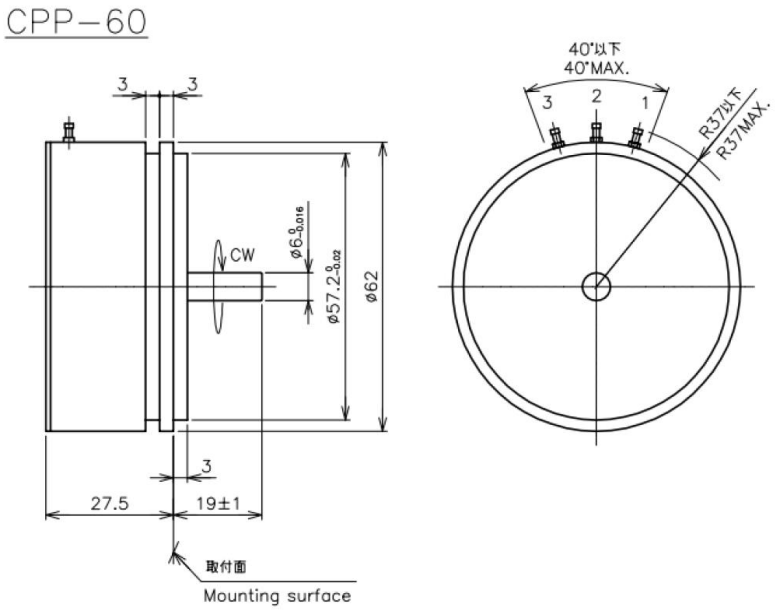
Appendix D

Conductive Plastic Angle Sensor

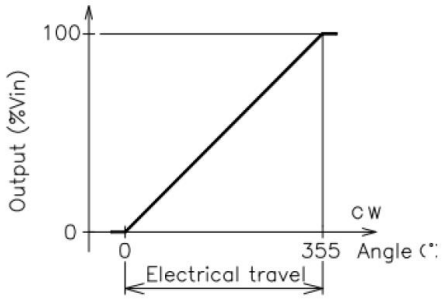
CPP-60 Series

・Conductive Plastic Angle Sensor
・Effective Electrical Angle：355°
・Independent Linearity ：±0.05%
・Servo Mount
【Option】
・Independent Linearity ： ±0.03%
【Material】
・Housing ：Aluminum
・Shaft ：Stainless Steel
・Bearing ：Stainless steel

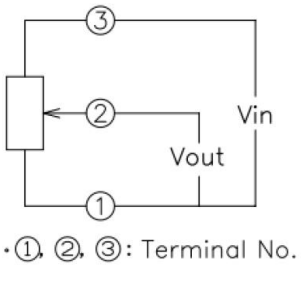
■Dimension [mm]



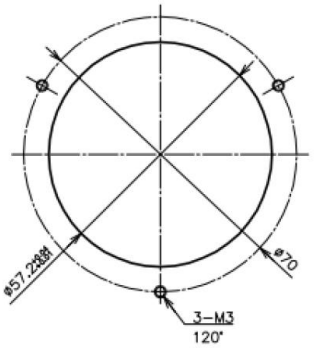
■Output Characteristics



■Schematic



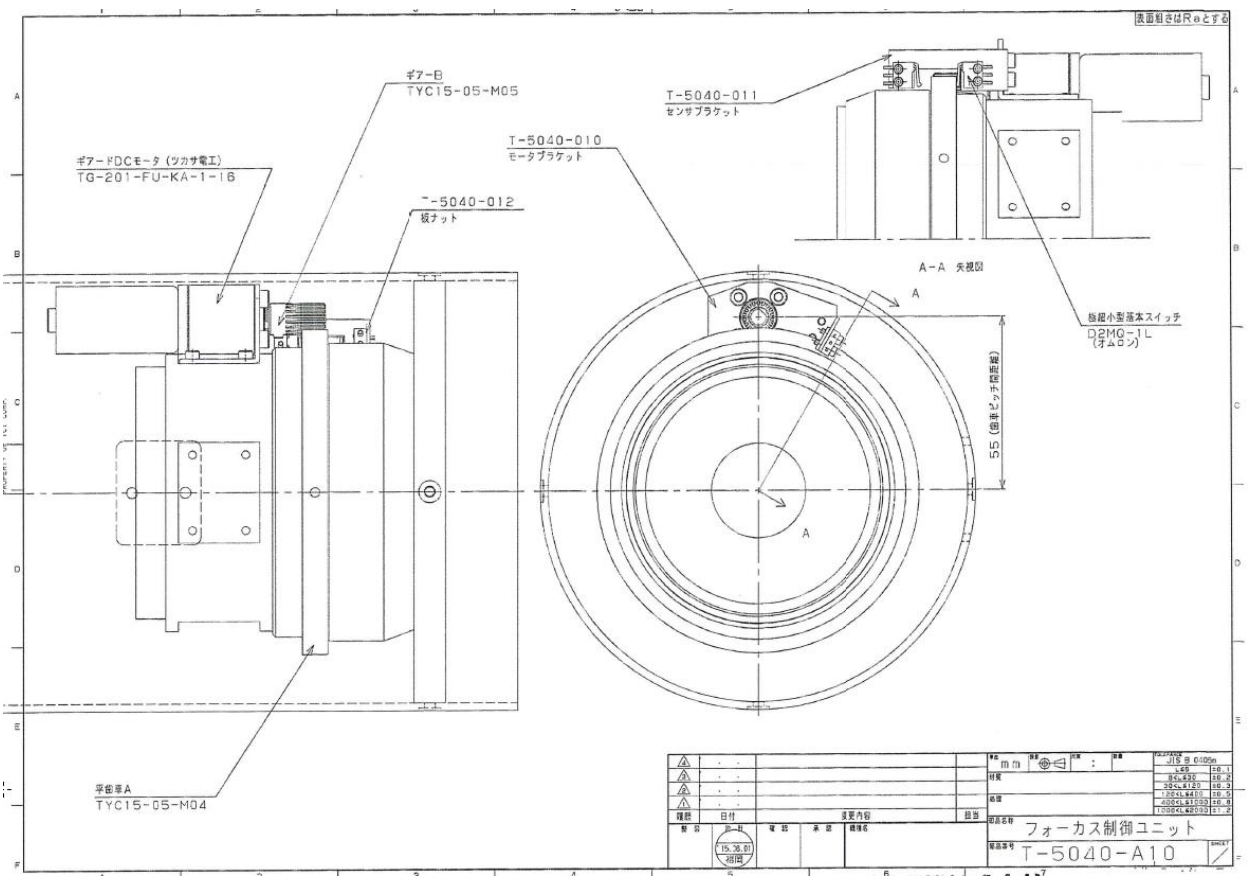
■Mounting



Servo Mount

MIDORI PRECISIONS
www.midori.co.jp
2018/02

Appendix E



Appendix F

This is the c++ source code used to analyze the video recorded in the field. It requires the OpenCV library. It was executed on Visual Studio 2013 Community on Windows 7.

```
#include <iostream> // for standard I/O
#include <string>   // for strings
#include <fstream>  // for .csv
#include <sstream>  // for .csv
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <ctime>     // for time
#include <windows.h>
#include <math.h>  // for atan
#include <iomanip>
#include <algorithm> // std::min_element,std::max_element
#include <numeric>
```

```
#include <opencv2/core/core.hpp>        // Basic OpenCV structures (cv::Mat)
#include <opencv2/highgui/highgui.hpp>  // Video write
#include "opencv2/imgproc/imgproc.hpp"
#include <opencv\highgui.h>
#include <opencv\cv.h>

using namespace std;
using namespace cv;

# define M_PI 3.14159265358979323846  /* pi */

void tiempo(SYSTEMTIME &stime){

	//structure to store system time (in usual time format)
	FILETIME ltime;
	//structure to store local time (local time in 64 bits)
	FILETIME ftTimeStamp;

	GetSystemTimeAsFileTime(&ftTimeStamp); //Gets the current system time

	FileTimeToLocalFileTime(&ftTimeStamp, &ltime);//convert in local time and store in
ltime
	FileTimeToSystemTime(&ltime, &stime);//convert in system time and store in stime
}

void findcenter(Mat &threshld, int &lindex){

	// Define Vector used to store the thresholded image white intensity values o
	vector<double> binint(threshld.cols);

	// Calculate sumatory of weights
	// See "Determination of crop rows by image analysis without segmentation",
H.T.Søgaard, H.J.Olsen
	double sumx = 0;
	double sumy = 0;
	double sum_of_elems = 0;

	// Add values of each column of TP threshold image
	for (int j = 0; j < threshld.cols; j++){
		for (int i = 0; i < threshld.rows; i++){
			binint[j] = binint[j] + (int)threshld.at<uchar>(i, j);
		}
		// Calculate sumatory of weights
		sumx = sumx + binint[j] * cos(2 * M_PI*((j - 0.5) / (binint.size())));
		sumy = sumy + binint[j] * sin(2 * M_PI*((j - 0.5) / (binint.size())));
		sum_of_elems = sum_of_elems + binint[j];
	}

	// Calculate center of gravity
	double xa = sumx / sum_of_elems;
	double ya = sumy / sum_of_elems;

	// Calculate direction angle phi (0 <= phi < 2pi)
	double phi = 0;
	if ((xa > 0) && (ya >= 0)){
		phi = atan(ya / xa);
	}
```

```
	else if ((xa > 0) && (ya < 0)){
		phi = 2 * M_PI + atan(ya / xa);
	}
	else if (xa < 0){
		phi = M_PI + atan(ya / xa);
	}
	else if ((xa == 0) && (ya > 0)){
		phi = M_PI / 2;
	}
	else if ((xa == 0) && (ya < 0)){
		phi = (3 * M_PI) / 2;
	}
	else if ((xa == 0) && (ya == 0)){
		phi = 0; // in reality is undefined, but is set to zero fro practical
purposes
	}

	//Calulate index l from vector binint corresponding to the centre of the row
	lindex = (phi / 2 * M_PI)*binint.size()*0.1;
}

void morphOps(Mat &thresh){
	//Opening Morphology Transformation: It is obtained by the erosion of an image
followed by a dilation.
	//create structuring element that will be used to "dilate" and "erode" image.
	//the element chosen here is a 11px by 22px rectangle
	Mat erodeElement = getStructuringElement(MORPH_RECT, Size(11, 22)); //Size(11,
11));
	//dilate with larger element so make sure object is nicely visible
	Mat dilateElement = getStructuringElement(MORPH_RECT, Size(7, 14)); //Size(7, 7));
	// in this particular case, better results were obtained dilating first and then
eroding
	dilate(thresh, thresh, dilateElement);
	erode(thresh, thresh, erodeElement);
}


// Find lane centers into the ROI's subregion
vector<Point2f> trackFilteredObject(Mat threshold, Mat &cameraFeed, Point offs){

	Mat temp;
	threshold.copyTo(temp);
	//these two vectors are needed for the output of findContours
	vector< vector<Point> > contours;
	vector<Vec4i> hierarchy;
	//find contours of filtered image using openCV findContours function
	findContours(temp, contours, hierarchy, CV_RETR_CCOMP, CV_CHAIN_APPROX_SIMPLE);

	// Draw contours
	//for (int i = 0; i< contours.size(); i++)
	//{
	//	drawContours(cameraFeed, contours, i, Scalar(0, 0, 255), 2, 8, hierarchy,
0, offs);
	//}

	// Get the moments
	vector<Moments> mu(contours.size());
	for (int i = 0; i < contours.size(); i++)
```

```
	{
		mu[i] = moments(contours[i], false);
	}

	//  Get the mass centers:
	vector<Point2f> mc(contours.size());
	vector<Point2f> myvec;
	for (int i = 0; i < contours.size(); i++)
	{   // Offset to Point(160, 480)
		mc[i] = Point2f((mu[i].m10 / mu[i].m00) + offs.x, (mu[i].m01 / mu[i].m00) +
offs.y);
		// Check that the point is inside the ROI
		if ((mc[i].x > offs.x) && (mc[i].x < offs.x + 420) && (mc[i].y > offs.y) &&
(mc[i].y < offs.y + 256)){
			myvec.push_back(mc[i]);
		}
		else if ((mc[i].x > offs.x) && (mc[i].x < offs.x + 640) && (mc[i].y >
offs.y) && (mc[i].y < offs.y + 439)){
			myvec.push_back(mc[i]);
		}
	}

	//for (int i = 0; i < myvec.size(); i++)
	//{
	//	circle(cameraFeed, myvec[i], 4, Scalar(255, 0, 0), -1, 8, 0);
	//}

	return myvec;
}


// Separate centers in left, center and right lanes
void separador(vector<vector<Point2f>> &centers, vector<vector<Point2f>> &lineas){

	for (int i = 0; i < centers.size(); i++){
		for (int j = 0; j < centers[i].size(); j++){
			// Evaluate if a point P(x,y) is on the left side or right side of
the line AB; A(x1,y1) B(x2,y2)
			// d = (x-x1)*(y2-y1)-(y-y1)*(x2-x1)
			// if d<0 ->left side; if d>0 ->right side; if d=0 ->on the line
			// Keep the x=110 y=360 (shift of ROI) addition terms for future
reference. The commmented equations are equivalent to the uncommented ones
			// i.e. The uncommmented equations are the simplifications of the
commented ones
			//double L1 = (centers[i][j].x - (150 + 110))*((256 + 360) - (0 +
360)) - (centers[i][j].y - (0 + 360))*((70 + 110) - (150 + 110));
			//double L2 = (centers[i][j].x - (200 + 110))*((256 + 360) - (0 +
360)) - (centers[i][j].y - (0 + 360))*((155 + 110) - (200 + 110));
			//double R1 = (centers[i][j].x - (240 + 110))*((256 + 360) - (0 +
360)) - (centers[i][j].y - (0 + 360))*((285 + 110) - (240 + 110));
			//double R2 = (centers[i][j].x - (290 + 110))*((256 + 360) - (0 +
360)) - (centers[i][j].y - (0 + 360))*((370 + 110) - (290 + 110));

			double L1 = (centers[i][j].x - 260)*(256) - (centers[i][j].y -
360)*(-80);
			double L2 = (centers[i][j].x - 310)*(256) - (centers[i][j].y -
360)*(-45);
```

```
                        double R1 = (centers[i][j].x - 350)*(256) - (centers[i][j].y -
360)*(45);
                        double R2 = (centers[i][j].x - 400)*(256) - (centers[i][j].y -
360)*(80);

                        if (L1 < 0){
                                lineas[0].push_back(centers[i][j]); //lineas[0]=left 1
                        }
                        else if ((L1 > 0) && (L2 < 0)){
                                lineas[1].push_back(centers[i][j]); //lineas[1]=left 2
                        }
                        else if ((L2 > 0) && (R1 < 0)){
                                lineas[2].push_back(centers[i][j]); //lineas[2]=center
                        }
                        else if ((R1 > 0) && (R2 < 0)){
                                lineas[3].push_back(centers[i][j]); //lineas[3]=right 1
                        }
                        else if (R2 > 0){
                                lineas[4].push_back(centers[i][j]); //lineas[0]=right 2
                        }
                }
        }
}

// Extend the calculated lines to the ROI limits y=360 to y=616
void linearizer(Point2f &A, Point2f &B, Point2f &A0, Point2f &B0){

        // Calculate line slope
        float m = (B.y - A.y) / (B.x - A.x);
        // Calculate line intercept
        float b = A.y - m*A.x;
        // Calculate line's end points
        A0.x = (360 - b) / m;
        A0.y = 360;
        B0.x = (616 - b) / m;
        B0.y = 616;

}

int main(int argc, char *argv[])
{
        // Matrices
        // Store input frame
        Mat dsp;
        Mat dsp2;
        // Separate Wide-angle from Telephoto image
        Mat croppedWA;
        Mat croppedTP;
        // Convert frame from BGR to HSV colorspace
        Mat HSV;
        Mat HSVTP;
        // Filter HSV image between values and store filtered image to threshold matrix
        Mat threshld;
        Mat threshldTP;

        // Range min and max HSV filter values for WA image
        int H_MIN = 21;
        int H_MAX = 117;
```

```
	int S_MIN = 51;
	int S_MAX = 256;
	int V_MIN = 72;
	int V_MAX = 256;

	//  0 - based index of the frame to be decoded / captured next
	double Findex = 0;

	// Windows names
	char* image_window = "VIDEO INPUT";

	// Time stamp variable
	SYSTEMTIME stime;

	// Input video
	VideoCapture invid;

	// Open Input video
	invid.open("C:/Users/vebots/Desktop/SYNCDATA/OUT_2017-10-16_01-09-34_PM.avi");

	if (!invid.isOpened())
	{
		cout << "Could not open the input from file" << endl;
		return -1;
	}

	// Define output size
	Size S = Size((int)1280, (int)720);
	Size So = Size((int)1260, (int)512);

	// Output video
	VideoWriter outputVideo;

	// Open Output video
	outputVideo.open("C:/Users/vebots/Desktop/SYNCDATA/OUT/2017-10-16_01-09-
34_PM_LNMP.avi", CV_FOURCC('m', 'p', '4', 'v'), 24, S, true);

	if (!outputVideo.isOpened())
	{
		cout << "Could not open the output video for write " << endl;
		return -1;
	}

	// Timer
	clock_t start;
	double duration;

	// Split ROI into 16 subregions of 420x16 pixels
	// Vector of Matrices for the threshold segments
	vector <Mat> strips(16);
	// Vector of Vectors of Points for areas centers
	vector<vector<Point2f>> centers(16);

	//Masking points for filled triangle
	Point tr_left[1][3];
	tr_left[0][0] = Point(0, 0);
	tr_left[0][1] = Point(0, 192);
	tr_left[0][2] = Point(105, 0);
```

```
        const Point* ppt_tr_left[1] = { tr_left[0] };
        int npt_tr_left[] = { 3 };

        Point tr_right[1][3];
        tr_right[0][0] = Point(420, 0);
        tr_right[0][1] = Point(420, 192);
        tr_right[0][2] = Point(315, 0);

        const Point* ppt_tr_right[1] = { tr_right[0] };
        int npt_tr_right[] = { 3 };

        // Declare input .csv file
        ifstream myfile;
        string values;
        string Frindex, Line, GPGGA, Time, Latitude, LatDec, Northing, N, Longitude,
LonDec, Easting, E, RTK, Nsat, HDoP, Altitude, M, Geoid, M2, lastT, Chksum, GPVTG, TTMG,
T, MTMG, M3, GSk, N2, Kph, K, mps, Chksum2, Roll, Pitch, Heading, XAccel, YAccel, ZAccel,
XRate, YRate, ZRate, hh, mm, ss, ms, PCtimer, timerfr, durtion;
        //Findex, Line, $GPGGA, Time, Latitude, LatDec, Northing, N, Longitude, LonDec,
Easting, E, RTK, #sat, HDoP, Altitude, M, Geoid, M, lastT, Chksum, $GPVTG, TTMG, T, MTMG,
M, GSk, N, Kph, K, m / s, Chksum, Roll, Pitch, Heading, XAccel, YAccel, ZAccel, XRate,
YRate, ZRate, hh, mm, ss, ms, PCtimer, timerfr, duration

        double east = 0;
        double north = 0;
        double velocity = 0;
        double Yaw = 0;

        // Initial values to set to zero
        double east0 = 0;
        double north0 = 0;
        double velocity0 = 0;
        double distance = 0;

        // Input file
        myfile.open("C:/Users/vebots/Desktop/SYNCDATA/OUT_2017-10-16_01-09-34_PM.csv");
        // Read the whole 1st line to jump headers
        getline(myfile, values, '\n');
        // Read the whole 2nd line to get initial values
        getline(myfile, values, '\n');
        // Get initial values of Easting, Northing and Velocity
        std::size_t pos = values.find(",N,");      // position of ",N," in values
        north0 = stod(values.substr(pos - 11, 11)); // get from ",N,"-11 to ",N,"
        pos = values.find(",E,");      // position of ",E," in values
        east0 = stod(values.substr(pos - 10, 10)); // get from ",E,"-11 to ",E,"
        pos = values.find(",K,");      // position of ",K," in values
        velocity0 = stod(values.substr(pos + 3, 5)); // get from ",K,"+3 to ",Chksum"
        // Reset begining of file
        myfile.clear();
        myfile.seekg(0, myfile.beg);
        // Read the whole 1st line to jump headers
        getline(myfile, values, '\n');

        // Open output .csv file
        ofstream framet;
        framet.open("C:/Users/vebots/Desktop/SYNCDATA/OUT/2017-10-16_01-09-
34_PM_LNMP.csv");
```

```
	// Write to output.csv file
	framet << "FRAME, Easting, E, Northing, N, velocity, m/s, distance, m, Heading,
Deg,TIMESTAMP, PCtimer, CenterWA, CenterTP, CenterMan" << endl;

	// Counter of mapped frames
	int mpcutwA = 1;

	// Set the temporal matrix type and size as input
	// 148 frames -> 439*148=64972 (maximum matrix size is 640x65400)
	// This temporal matrix stores the stitched image
	Mat temporalWA = Mat::zeros(72 * 67, 617, CV_8UC3);

	// Output image declaration
	string out_mapWA = "C:/Users/vebots/Desktop/SYNCDATA/OUT/2017-10-16_01-09-
34_PM_LNMP_mapWA.png";

	int pos_lane = 332;

	// Data vector to plot

	vector<Point2i> centerWA(64);
	vector<Point2i> centerTP(64);
	vector<Point2i> centerMAN(64);

	// Set Data vector to zero
	for (int i = 0; i < centerWA.size(); i++)
	{

		centerWA[i].x = i * 10 + 10;
		centerWA[i].y = 310;
		centerTP[i].x = i * 10 + 10;
		centerTP[i].y = 310;
		centerMAN[i].x = i * 10 + 10;
		centerMAN[i].y = 310;
	}

	// Start Timer, execute loop
	start = clock();
	for (;;){

		// Read a new frame
		bool bok = invid.read(dsp);
		if (!bok) {
			cout << "Cannot read frame from file" << endl;
			break;
		}

		dsp.copyTo(dsp2);

		// Input .csv variables
		//Read the whole line
		getline(myfile, values, '\n');
		// Asign the whole line to a new stringstream
		stringstream linestream(values);
		// Decompose the stringstream on each one of the values
		getline(linestream, Frindex, ',');
		getline(linestream, Line, ',');
```

```
getline(linestream, GPGGA, ',');
getline(linestream, Time, ',');
getline(linestream, Latitude, ',');
getline(linestream, LatDec, ',');
getline(linestream, Northing, ',');
getline(linestream, N, ',');
getline(linestream, Longitude, ',');
getline(linestream, LonDec, ',');
getline(linestream, Easting, ',');
getline(linestream, E, ',');
getline(linestream, RTK, ',');
getline(linestream, Nsat, ',');
getline(linestream, HDoP, ',');
getline(linestream, Altitude, ',');
getline(linestream, M, ',');
getline(linestream, Geoid, ',');
getline(linestream, M2, ',');
getline(linestream, lastT, ',');
getline(linestream, Chksum, ',');

getline(linestream, GPVTG, ',');
getline(linestream, TTMG, ',');
getline(linestream, T, ',');
getline(linestream, MTMG, ',');
getline(linestream, M3, ',');
getline(linestream, GSk, ',');
getline(linestream, N2, ',');
getline(linestream, Kph, ',');
getline(linestream, K, ',');
getline(linestream, mps, ',');
getline(linestream, Chksum2, ',');
getline(linestream, Roll, ',');
getline(linestream, Pitch, ',');
getline(linestream, Heading, ',');
getline(linestream, XAccel, ',');
getline(linestream, YAccel, ',');
getline(linestream, ZAccel, ',');
getline(linestream, XRate, ',');
getline(linestream, YRate, ',');
getline(linestream, ZRate, ',');
getline(linestream, hh, ',');
getline(linestream, mm, ',');
getline(linestream, ss, ',');
getline(linestream, ms, ',');
getline(linestream, PCtimer, ',');
getline(linestream, timerfr, ',');
getline(linestream, durtion, '\n');

// Convert string to double. Set to zero (initial values)
east = stod(Easting) - east0;
north = stod(Northing) - north0;
velocity = stod(mps) - velocity0;
Yaw = stod(Heading);

// Calculate linear distance
distance = sqrt(pow(east, 2) + pow(north, 2));

// Capture frame each 2.20 meters
```

```
		if (distance >= 2.20*mpcutwA){
			dsp(Rect(23, 360, 617, 72)).copyTo(temporalWA(Rect(0, 4824 - (72 *
mpcutwA), 617, 72)));
			mpcutwA++;
		}

		//  0 - based index of the frame to be decoded / captured next
		Findex = invid.get(CV_CAP_PROP_POS_FRAMES);

		// Select region of interest, separate Wide-angle from Telephoto image
		croppedWA = dsp(Rect(110, 360, 420, 256));
		croppedTP = dsp(Rect(640, 118, 640, 439));

		// Convert frame from BGR to HSV colorspace
		cvtColor(croppedWA, HSV, COLOR_BGR2HSV);
		cvtColor(croppedTP, HSVTP, COLOR_BGR2HSV);

		// Filter HSV image between values and store filtered image to threshold
matrix
		inRange(HSV, Scalar(H_MIN, S_MIN, V_MIN), Scalar(H_MAX, S_MAX, V_MAX),
threshld);
		inRange(HSVTP, Scalar(H_MIN, S_MIN, V_MIN), Scalar(H_MAX, S_MAX, V_MAX),
threshldTP);

		// Lateral index from TP image
		int lindex = 0;
		findcenter(threshldTP, lindex);

		// Plot index l
		line(dsp, Point(640 + lindex, 105),
			Point(640 + lindex, 615), Scalar(0, 0, 255), 2, CV_AA);

		line(dsp, Point((lindex / 12) + 302, 360),
			Point((lindex / 12) + 302, 616), Scalar(0, 0, 255), 2, CV_AA);

		// Draw solid triangles and lines to fit ROI
		fillPoly(threshld, ppt_tr_left, npt_tr_left, 1, Scalar(0, 0, 0), 8, 0);
		fillPoly(threshld, ppt_tr_right, npt_tr_right, 1, Scalar(0, 0, 0), 8, 0);
		line(threshld, Point(150, 0), Point(70, 256), Scalar(0, 0, 0), 4);
		line(threshld, Point(200, 0), Point(155, 256), Scalar(0, 0, 0), 4);
		line(threshld, Point(240, 0), Point(285, 256), Scalar(0, 0, 0), 4);
		line(threshld, Point(290, 0), Point(370, 256), Scalar(0, 0, 0), 4);

		// Go to "dilate" and "erode" image function.
		morphOps(threshld);

		// Split ROI into 16 subregions of 420x16 pixels
		for (int i = 0; i < strips.size(); i++){
			strips[i] = threshld(Rect(0, 16 * i, 420, 16));
			// Find lane centers into the ROI's 16 subregions of 420x16 pixels
			centers[i] = trackFilteredObject(strips[i], dsp, Point(110, 360 + 16
* i));
		}

		// Separate centers in left, center and right lanes
		// lineas[0]=left 1; lineas[1]=left 2; lineas[2]=center; lineas[3]=right 1;
lineas[4]=right 2
		vector<vector<Point2f>> lineas(5);
```

```
            separador(centers, lineas);

            // Find the line that fits each lane's detected points
            // Vec4f = (vx, vy, x0, y0); (vx, vy) is a normalized vector collinear to
the line and (x0, y0) is a point on the line
            // linerizado[0]=left 1; linerizado[1]=left 2; linerizado[2]=center;
linerizado[3]=right 1; linerizado[4]=right 2
            vector<Vec4f> linerizado(5);
            Point2f A0;
            Point2f B0;
            Point2f auxcenterWA;
            for (int i = 0; i < linerizado.size(); i++){
                  if (lineas[i].size() >= 2){
                        fitLine(lineas[i], linerizado[i], CV_DIST_L2, 0, 1, 0.01);
                        linearizer(Point2f(linerizado[i][2] - 60 * linerizado[i][0],
linerizado[i][3] - 60 * linerizado[i][1]), Point2f(linerizado[i][2] + 60 *
linerizado[i][0], linerizado[i][3] + 60 * linerizado[i][1]), A0, B0);
                        line(dsp, A0, B0, Scalar(255, 0, 0), 2, CV_AA);
                        if (i == 2){ auxcenterWA = B0; } // center lane coordinate
                  }
            }
            line(dsp, Point((auxcenterWA.x - 302) * 12 + 640, 105),
Point((auxcenterWA.x - 302) * 12 + 640, 615), Scalar(255, 0, 0), 2, CV_AA); // Show in TP

            // Central row
            //line(dsp, Point(332, 105), Point(332, 615), Scalar(0, 255, 255), 1,
CV_AA);   // WA
            //line(dsp, Point(1002, 105), Point(1002, 615), Scalar(0, 255, 255), 2,
CV_AA); // TP

            // Location of TP inside WA
            //rectangle(dsp, Point(302, 360), Point(355, 400), Scalar(255, 255, 255),
1, CV_AA);
            //circle(dsp, Point(327, 380), 1, Scalar(255, 255, 255), 1, CV_AA);
            //circle(dsp, Point(960, 348), 4, Scalar(255, 255, 255), 4, CV_AA);

            int tecla = waitKey(1);

            if (tecla == 97) {
                  pos_lane = pos_lane - 1;
            }
            else if (tecla == 115){
                  pos_lane = pos_lane + 1;
            }

            //line(dsp, Point(pos_lane, 345), Point(pos_lane, 615), Scalar(0, 255,
255), 1, CV_AA);
            //line(dsp, Point(pos_lane+12, 345), Point(pos_lane+12, 615), Scalar(0,
255, 255), 1, CV_AA);
            //line(dsp, Point(pos_lane-12, 345), Point(pos_lane-12, 615), Scalar(0,
255, 255), 1, CV_AA);

            // Display result image
            //imshow(image_window, dsp);

            // Save output video
            //outputVideo.write(dsp);
```

```
		// Time stamp
		tiempo(stime);
		// Compute program execution time
		duration = (clock() - start) / (double)CLOCKS_PER_SEC;

		// Write to .csv file
		framet <<
			std::fixed << std::setprecision(0) << Findex << "," <<
			std::fixed << std::setprecision(3) << east << ",E," <<
			std::fixed << std::setprecision(3) << north << ",N," <<
			std::fixed << std::setprecision(3) << velocity << ",m/s," <<
			std::fixed << std::setprecision(3) << distance << ",m," <<
			Heading << ",Deg," <<
			hh << ":" << mm << ":" << ss << ":" << ms << "," << duration << ","
<<
			(int)auxcenterWA.x << "," <<
			lindex << "," <<
			pos_lane << endl;

		// Show in cmd window
		system("CLS");
		cout << " Processing! " << endl << endl;
		cout << " FRAME: " << Findex << endl;
		cout << " TIMER: " << duration << endl;
		cout << " FPS: " << Findex / duration << endl;
		cout << " lindex: " << lindex << endl;
		cout << " auxcenterWA: " << (int)auxcenterWA.x << endl;
		cout << " POS: " << pos_lane << endl;
		cout << " Press ESC to close" << endl;

		// Matrix to show plot, refresh on each loop
		cv::Mat plot_centers = Mat::zeros(320, 640, CV_8UC3);

		// Update data
		for (int i = 0; i < centerWA.size() - 1; i++)
		{
			centerWA[i].y = centerWA[i + 1].y;
			centerTP[i].y = centerTP[i + 1].y;
			centerMAN[i].y = centerMAN[i + 1].y;
		}
		// Scale to fit in plot range (plot range 0~320) (input range 280~380)
		// y = ((x - xmin)/(xmax - xmin))*(ymax - ymin) + ymin
		centerWA[centerWA.size() - 1].y = 320 - ((auxcenterWA.x - 280.0) / 100.0 *
320.0);
		centerTP[centerTP.size() - 1].y = 320 - ((((lindex / 12) + 302) - 280.0) /
100.0 * 320.0);
		centerMAN[centerMAN.size() - 1].y = 320 - ((pos_lane - 280.0) / 100.0 *
320.0);

		// Plot lines
		for (int i = 0; i < centerWA.size() - 1; i++)
		{
			line(plot_centers, centerMAN[i], centerMAN[i + 1], Scalar(0, 255,
0), 4); // Plot centerTP
			line(plot_centers, centerTP[i], centerTP[i + 1], Scalar(0, 0, 255),
4); // Plot centerTP
			line(plot_centers, centerWA[i], centerWA[i + 1], Scalar(255, 0, 0),
4); // Plot centerWA
```

```
            }
            line(plot_centers, Point(10, 154), Point(630, 154), Scalar(0, 255, 255),
2); // Plot center of frame
            arrowedLine(plot_centers, Point(10, 310), Point(630, 310), Scalar(255, 255,
255), 2, 8, 0, 0.01); // x axis
            arrowedLine(plot_centers, Point(10, 310), Point(10, 10), Scalar(255, 255,
255), 2, 8, 0, 0.01); // y axis
            putText(plot_centers, "380", Point(15, 20), FONT_HERSHEY_SIMPLEX, .5,
Scalar(255, 255, 255), 1, 8, false); // Plot y max
            putText(plot_centers, "280", Point(15, 305), FONT_HERSHEY_SIMPLEX, .5,
Scalar(255, 255, 255), 1, 8, false); // Plot y min
            putText(plot_centers, "Center WA = " + to_string((int)auxcenterWA.x) + "
[Pixels]", Point(200, 20), FONT_HERSHEY_SIMPLEX, .5, Scalar(255, 0, 0), 1, 8, false); //
Plot centerWA
            putText(plot_centers, "Center TP = " + to_string((lindex / 12) + 302) + "
[Pixels]", Point(200, 40), FONT_HERSHEY_SIMPLEX, .5, Scalar(0, 0, 255), 1, 8, false); //
Plot centerTP
            putText(plot_centers, "Center MAN = " + to_string(pos_lane) + " [Pixels]",
Point(200, 60), FONT_HERSHEY_SIMPLEX, .5, Scalar(0, 255, 0), 1, 8, false); // Plot
centerTP
            putText(plot_centers, "Ideal Center = 332 [Pixels]", Point(200, 80),
FONT_HERSHEY_SIMPLEX, .5, Scalar(0, 255, 0), 1, 8, false); // Plot centerTP
            // Plot centers WA & TP
            imshow("Centers comparison", plot_centers);

            rectangle(dsp, Point(110, 360), Point(530, 615), Scalar(255, 0, 0), 2,
CV_AA);
            rectangle(dsp, Point(302, 360), Point(355, 400), Scalar(0, 0, 255), 2,
CV_AA);

            plot_centers.copyTo(dsp(Rect(0, 0, 640, 320)));

            // Display result image
            imshow(image_window, dsp);

            // Draw markers
            //  Outline ROI WA
            rectangle(dsp2, Point(110, 360), Point(530, 615), Scalar(255, 0, 0), 2,
CV_AA);
            // Location of TP inside WA
            rectangle(dsp2, Point(302, 360), Point(355, 400), Scalar(0, 0, 255), 2,
CV_AA);
            //// Show trapezoid ROI WA
            //line(dsp2, Point(554, 360), Point(640, 432), Scalar(0, 255, 0), 2);
            //line(dsp2, Point(23, 432), Point(640, 432), Scalar(0, 255, 0), 2);
            //line(dsp2, Point(23, 432), Point(97, 360), Scalar(0, 255, 0), 2);
            //line(dsp2, Point(554, 360), Point(97, 360), Scalar(0, 255, 0), 2);
            imshow("input", dsp2);

            // Save output video
            outputVideo.write(dsp);


            // Wait for 'esc' key press for 30ms. If 'esc' key is pressed, break loop
            if (waitKey(1) == 27)
            {
                  cout << "ESC key pressed by user" << endl;
                  break;
```

```
            }
        }

        // Close .csv input file
        myfile.close();
        // Close .csv output file
        framet.close();

        // Save stitched image
        cout << " Saving, please wait... " << endl;

        // Input Quadilateral or Image plane coordinates
        Point2f inputQuad[4];
        // Output Quadilateral or World plane coordinates
        Point2f outputQuad[4];

        // These four pts are the sides of the trapezoid used as input
        inputQuad[0] = Point2f(74, 0);
        inputQuad[1] = Point2f(531, 0);
        inputQuad[2] = Point2f(617, 72);
        inputQuad[3] = Point2f(0, 72);
        // The 4 points where the mapping is to be done , from top-left in clockwise order
        outputQuad[0] = Point2f(-10, 0);
        outputQuad[1] = Point2f(622, 0);
        outputQuad[2] = Point2f(627, 72);
        outputQuad[3] = Point2f(0, 72);

        // Lambda Matrix
        Mat lambda(2, 4, CV_32FC1);

        // Output Image
        Mat TWA = Mat::zeros(72 * 67, 617, CV_8UC3);

        // Set the lambda matrix the same type and size as input
        lambda = Mat::zeros(72, 617, CV_8UC3);

        // Get the Perspective Transform Matrix i.e. lambda
        lambda = getPerspectiveTransform(inputQuad, outputQuad);

        // Reset Counter of mapped frames
        mpcutwA = 0;

        // Transform Temporal Mosaic using warpPerspective into Output Image Mosaic
        for (int i = 0; i< 67; i++){
                warpPerspective(temporalWA(Rect(0, 72 * mpcutwA, 617, 72)), TWA(Rect(0, (72
* mpcutwA), 617, 72)), lambda, temporalWA(Rect(0, 72 * mpcutwA, 617, 72)).size());
                mpcutwA++;
        }

        // Save stitched-warped image
        cv::imwrite(out_mapWA, TWA);
        cout << " Press Enter  to exit " << endl;
        std::cin.get();

        return 0;
}
```