University of Southern Queensland

Faculty of Health, Engineering & Sciences

Orchard mapping and mobile robot localisation using on-board camera and laser scanner data fusion

A thesis submitted by

Nagham Jamil Dawood Shalal

MSc Mechatronics Engineering

For the award of

Doctor of Philosophy

2016

Abstract

Agricultural mobile robots have great potential to effectively implement different agricultural tasks. They can save human labour costs, avoid the need for people having to perform risky operations and increase productivity. Automation and advanced sensing technologies can provide up-to-date information that helps farmers in orchard management. Data collected from on-board sensors on a mobile robot provide information that can help the farmer detect tree or fruit diseases or damage, measure tree canopy volume and monitor fruit development. In orchards, trees are natural landmarks providing suitable cues for mobile robot localisation and navigation as trees are nominally planted in straight and parallel rows.

This thesis presents a novel tree trunk detection algorithm that detects trees and discriminates between trees and non-tree objects in the orchard using a camera and 2D laser scanner data fusion. A local orchard map of the individual trees was developed allowing the mobile robot to navigate to a specific tree in the orchard to perform a specific task such as tree inspection. Furthermore, this thesis presents a localisation algorithm that does not rely on GPS positions and depends only on the on-board sensors of the mobile robot without adding any artificial landmarks, reflective tapes or tags to the trees.

The novel tree trunk detection algorithm combined the features extracted from a low cost camera's images and 2D laser scanner data to increase the robustness of the detection. The developed algorithm used a new method to detect the edge points and determine the width of the tree trunks and non-tree objects from the laser scan data. Then a projection of the edge points from the laser scanner coordinates to the image plane was implemented to construct a region of interest with the required features for tree trunk colour and edge detection. The camera images were used to verify the colour and the parallel edges of the tree trunks and non-tree objects. The algorithm automatically adjusted the colour detection parameters after each test which was shown to increase the detection accuracy. The orchard map was constructed based on tree trunk detection and consisted of the 2D positions of the individual trees and non-tree objects. The map of the individual trees was used as an *a priori* map for mobile robot localisation. A data fusion algorithm based on an Extended Kalman filter was used for pose estimation of the mobile robot in different paths (midway between rows, close to the rows and moving around trees in the row) and different turns (semi-circle and right angle turns) required for tree inspection tasks. The 2D positions of the individual trees were used in the correction step of the Extended Kalman filter to enhance localisation accuracy.

Experimental tests were conducted in a simulated environment and a real orchard to evaluate the performance of the developed algorithms. The tree trunk detection algorithm was evaluated under two broad illumination conditions (sunny and cloudy). The algorithm was able to detect the tree trunks (regular and thin tree trunks) and discriminate between trees and non-tree objects with a detection accuracy of 97% showing that the fusion of both vision and 2D laser scanner technologies produced robust tree trunk detection. The mapping method successfully localised all the trees and non-tree objects of the tested tree rows in the orchard environment. The mapping results indicated that the constructed map can be reliably used for mobile robot localisation and navigation. The localisation algorithm was evaluated against the logged RTK-GPS positions for different paths and headland turns. The average of the RMS of the position error in x, y coordinates and Euclidean distance were 0.08 m, 0.07 m and 0.103 m respectively, whilst the average of the RMS of the heading error was 3.32°. These results were considered acceptable while driving along the rows and when executing headland turns for the target application of autonomous mobile robot navigation and tree inspection tasks in orchards.

Certification of Thesis

I certify that the ideas, designs, experimental work, results, analyses, software and conclusions reported in this thesis are entirely my own effort, except where otherwise acknowledged. I also certify that the work is original and has not been previously submitted for any other award, except where otherwise acknowledged.

08 / 07 / 2016

Nagham Shalal, Student — Date

08 / 07/ 2016

Dr Tobias Low, Principal supervisor — Date

08 / 07 / 2016

Dr Cheryl McCarthy, Associate supervisor — Date

N H Hancock

08 / 07 / 2016

A/Prof. Nigel Hancock, Associate supervisor — Date

Acknowledgments

First of all, I would like to thank God for helping me writing this thesis.

I would like to extend my sincere appreciation and gratitude to my principal supervisor Dr. Tobias Low for his concern, support and guidance throughout my PhD study. I would also like to express my grateful thanks to my associate supervisors Dr Cheryl McCarthy and Assoc Prof Nigel Hancock for their guidance and support.

I would acknowledge the technical staff of USQ for their support, especially Mr Chris Galligan, Mr Brian Aston and Mr Terry Byrne. I would like to thank Associated Dean (Research) Assoc Prof Thiru Aravinthan and his assistant Mrs Juanita Ryan for their support. Also my deep appreciation goes to Ms Sandra Cochrane for English proof reading of this thesis. I would also like to thank all my friends and the Iraqi students at USQ for their encouragement and support.

I am very grateful to my mother, sisters and brother for all their prayers, unconditional support, love and encouragement. Thanks to my husband Hilal for his encouragement, support and helping me through all my outdoor experiments. Thanks to my children Mariam, Fadi and Merna for being such wonderful children and for their endless love and continuous encouragement during this research.

Finally, I would like to thank the Ministry of Higher Education and the University of Technology in Baghdad, Iraq for funding the PhD scholarship.

Nagham Jamil Dawood Shalal

Associated Publications

The following publications were produced during the period of candidature:

Shalal, N., Low, T., McCarthy, C. and Hancock, N. “Orchard mapping and mobile robot localisation using on-board camera and laser scanner data fusion - Part A: Tree detection”, *Computers and Electronics in Agriculture*, Vol. 119, 2015, pp 254-266.
(The work in this paper is presented in Chapter 4).

Shalal, N., Low, T., McCarthy, C. and Hancock, N. “Orchard mapping and mobile robot localisation using on-board camera and laser scanner data fusion - Part B: Mapping and localisation”, *Computers and Electronics in Agriculture*, Vol. 119, 2015, pp 267-278.
(The work in this paper is presented in Chapter 5 and Chapter 6).

Shalal, N., Low, T., McCarthy, C. and Hancock, N. “A preliminary evaluation of vision and laser sensing for tree trunk detection and orchard mapping”, *Proc of the 2013 Australasian Conference on Robotics and Automation (ACRA2013)*, Sydney, Australia, December 2-4, 2013.
(The work in this paper is presented in Chapter 4 and Chapter 5) .

Shalal, N., Low, T., McCarthy, C. and Hancock, N. “A review of autonomous navigation systems in agricultural environments”, *2013 Society for Engineering in*

Agriculture Conference: Innovative Agricultural Technologies for a Sustainable Future, Barton, ACT, Australia, September 22-25, 2013, pp. 1025.
(The work in this paper is presented in Chapter 2).

Shalal, N., Low, T., McCarthy, C. and Hancock, N. "Navigation systems of mobile robots in orchards: a review", Intended to be submitted to *Sensors Journal.*
(The work in this paper is presented in Chapter 2).

Contents

List of Figures

List of Tables

Notation

$A_{x,y,z}$	Acceleration in x, y, z axes
A_t	Motion model Jacobian matrix for states
$B_{1,2,3,4}$	Simulated object number
c_c	Principle point coordinates of the camera
d	Tree trunk width
D_E	Euclidean distance between ground truth and the calculated positions
$D_{E_{rms}}$	Root mean square of the Euclidean distance
D_f	Forward distance travelled by the mobile robot
D_l, D_r	Distance travelled by the left and right sides of the robot
D_m	Actual distance travelled by the robot
d_x	Tangential distortion vector of the camera
E_p	Measure of edge parallelism
$E_{x_{rms},y_{rms}}$	Root mean square of x and y position errors
$E_{x(t),y(t)}$	Position error between ground truth and estimated positions in x, y
f	Mobile robot motion model
f_c	Focal length of the camera
h	Measurement model
H_d	Most dominant value of Hue
H_t	Measurement model Jacobian matrix for states
K	Camera matrix
k_c	Distortion coefficients of the camera
K_t	Kalman gain
m	Orchard local map

N	Number of the image-laser scan pairs
$P_{1,2,3}$	Projection points of start, end and centre of object on image plane
P_c	Camera Cartesian coordinates
$pdf(d)$	Probability of the object width being a tree trunk width
$pdf(\mu_d)$	Probability of the width distribution around the mean
$pdf(H_d)$	Probability of the object Hue being a tree trunk Hue
$pdf(\mu_{H_d})$	Probability of the dominant value of Hue distribution around the mean
P_l	Laser Cartesian coordinates
P_t^-	Error covariance matrix
P_t	Corrected error covariance matrix
Q_t	Process noise covariance matrix
$r_{1,2}$	Laser range from the start and end edges of the tree trunk
R^2	Measure of the goodness of the line fitting
r_c	Laser range from the centre of the tree trunk
$ROC_{C,E,L}$	Rate of confidence from tree trunk colour, edges, laser
ROC_{Tree}	Final rate of confidence of the tree trunk
R_t	Measurement noise covariance matrix
$SE_{x,y}$	Standard errors in x and y positions
$T_{1,2,\ldots,12}$	Simulated tree trunk number
$TH_{C,E,L}$	Threshold value of the rate of confidence from colour, edges, laser
TH_{Tree}	Threshold value of the rate of confidence of simulated tree trunk
u_t	Control input vector
v	Linear velocity of the mobile robot
V_t	Measurement model Jacobian matrix for noise
W_t	Motion model Jacobian matrix for noise
x_d	Normalised point coordinates after lens distortion
X_{gt}	Ground truth position in x-coordinate
x_n	Normalised pinhole image projection
x_p	Pixel x-coordinate on the image plane
$\hat{x}_t^-$	Estimated pose at time t

$\hat{x}_t$	Corrected pose estimate
$\hat{x}_{t-1}$	Estimated pose at time $t-1$
X_{tree}	Tree trunk x-coordinate
Y_{gt}	Ground truth position in y-coordinate
y_p	Pixel y-coordinate on the image plane
Y_{tree}	Tree trunk y-coordinate
z_t	Actual measurement

Greek symbols

α_c	Skew coefficient of the camera
$\beta_{1,2}$	Angle of the right and left edge lines of the object
Δ	Translation offset vector between the laser and camera frames
Δ_t	Time difference between time steps t and $(t-1)$
$\Delta_{x,y}$	Position change in x and y directions
$\Delta\phi$	Difference between bearing angles of the tree trunk
$\delta_{x,y,z}$	Translation in x, y, z axes
θ	Heading angle of the mobile robot
μ_d	Mean of tree trunk width distribution
μ_{H_d}	Mean of the most dominant value of the Hue distribution
σ_d	Standard deviation of tree trunk width distribution
σ_{H_d}	Standard deviation of the most dominant value of the Hue distribution
$\sigma_{x,y}$	Standard deviation of the position data in x, y
$\sigma_{v,\omega}$	Standard deviation of the linear and angular velocities
$\sigma_{r,\phi}$	Standard deviation of the range and bearing angle
Φ	Rotation matrix between the laser and camera frames
$\phi_{1,2}$	Bearing angle of the start and end edges of the tree trunk
ϕ_c	Bearing angle of the centre of the tree trunk
$\psi_{x,y,z}$	Rotation angle about the x, y, z axes
ω	Angular velocity of the mobile robot

Acronyms & Abbreviations

2D	Two dimensional
3D	Three dimensional
CPR	Clocks per revolution
DGPS	Differential Global Positioning System
EIF	Extended information filter
EKF	Extended Kalman filter
FIR	Finite impulse response
FN	False negative
FOG	Fiber optic gyroscope
FP	False positive
GPS	Global Positioning System
HSV	Hue, Saturation, Value
IMU	Inertial measurement unit
KF	Kalman filter
LIDAR	Light detection and Ranging
NIR	Near-infrared
PF	Particle filter
RANSAC	RANdom SAmple Consensus
RFID	Radio frequency identification
RGB	Red, Green, Blue

RMS	Root mean square
ROI	Region of interest
ROS	Robot Operating System
RTK-GPS	Real time kinematic Global Positioning System
SLAM	Simultaneous localisation and mapping
TN	True negative
TP	True positive
UAV	Unmanned aerial vehicle
UTM	Universal Transverse Mercator
UWB	Ultra-Wide-Band

Chapter 1

Introduction

1.1 Robotics in Agriculture

Agriculture is a most promising sector that provides human with the basic life requirements as well as supporting the economic growth of most industrialised and developed countries. The advancement in automation is highly required in the agriculture sector to help with reducing the costs of production and increasing the quality of crops for greater income (Thamrin et al., 2013).

The cost of manual labour and the availability of skilled workers doing hard physical, repetitive and intensive agricultural tasks have become concerns for many farmers. With the development of technology, these agricultural tasks have been subject to automation. Advanced technology has resulted in the introduction of vehicle automation at a consumer level and has made it feasible to build systems that are more economical, efficient and reliable than human labour (Reske-Nielsen et al., 2006; Rovira-Más, 2009).

Mobile robots operating in real world environment have been a significant subject for many researchers because of their efficiency and flexibility. In agriculture, research on unmanned agricultural robots started in the early 1960s (Fountas et al., 2007). These agricultural robots have been introduced in different indoor and outdoor agricultural environments. Agricultural robots play an important role in implementing different agricultural tasks. One mobile robot can perform the work of two or more people. This can save a significant amount of labour costs for growers. Furthermore, robots reduce the risk of human errors and can be used in a variety of applications and locations (Griepentrog et al., 2009). The autonomous performance of these vehicles provides continuous supervision of the agricultural environments as information regarding the environment can be autonomously acquired. This provides the farmer with up-to-date and precise information to assist with management decisions (Auat Cheein and Carelli, 2013).

In recent years, advanced technology has encouraged many researchers to develop more intelligent and adaptable vehicles. These vehicles should have a sufficient amount of intelligence in order to sense the environment for long periods of time, whilst achieving specific tasks (Blackmore et al., 2005; Tabile et al., 2011). The need for autonomous navigation systems of agricultural robots has been recognised as a key component to achieve autonomous farming. This is because the autonomous navigation system is an important component in achieving different agricultural tasks such as planting, spraying, fertilizing, cultivating, harvesting, thinning, weeding, and inspection. The rapid advancement in communication, sensors, data acquisition, processing methods and computing technologies has provided important progress in the field of agricultural autonomous robot systems.

The design of mobile robots operating in outdoor environments such as agricultural applications is still a challenging subject. The autonomy of the mobile robot is obtained by means of sensing the environment and employing appropriate control algorithms for specified tasks. These tasks are complex and many repetitive tasks are not exactly the same for each repetition (Christiansen, 2011). The design of efficient and robust sensing and control systems for agricultural mobile robots needs to be planned and optimised before the robot can execute the task. It is also required that difficulties due to the weather conditions, dynamic environments, unexpected obstacles, terrain nature variations and vegetation be overcome (Ayala et al., 2008). A comprehensive knowledge of the robot's facilities and limitations, the environment characteristics and the task requirements are necessary to obtain good outcomes (Vougioukas et al., 2005).

1.2 Autonomous mobile robot navigation in agricultural environments

Autonomous navigation is one of the important issues in mobile robot applications. The task of navigation is to guide the mobile robot safely and autonomously within different environments. The mobile robot's navigation ability relies on complex sensor systems and intelligent control algorithms. The mobile robot must be capable of sensing and detecting the environment firstly, then analysing and modeling it. A navigation algorithm uses the sensed information to allow the mobile robot to determine a suitable trajectory, make a decision and move correctly within its environment (Parhi and Singh, 2009).

Navigation of the mobile robot in agricultural environments requires the consideration of the position of the mobile robot and the detection of the surrounding area and obstacles. The mobile robot should be able to avoid occasional unexpected obstacles such as animals (dead or alive), fallen tree branches and fence posts.

1.2.1 Autonomous mobile robot navigation components

The successful navigation of mobile robot in agricultural environments consists of feature extraction, mapping, localisation, path planning and obstacle avoidance as shown in Figure 1.1.

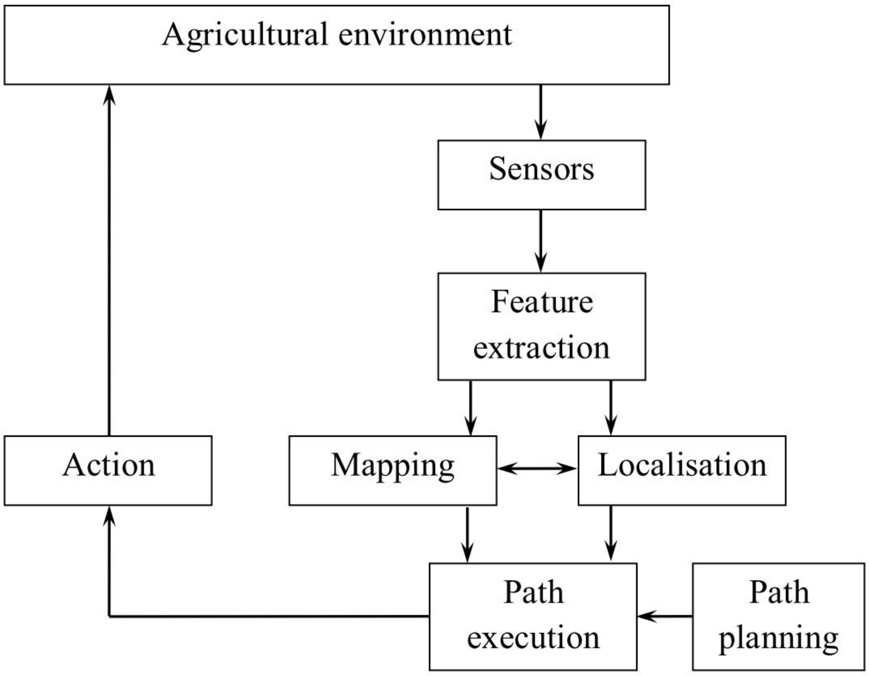


Figure 1.1: The block diagram of the autonomous navigation in agricultural environments.

.

The task of detection and feature extraction is more challenging in outdoor environments. The major problem is finding the suitable natural features that are stable under different environmental conditions. The detection of these agricultural features is directly related to the task of the agricultural robot and the sensors incorporated on it. The robot's on-board sensors acquire the environmental information to extract the useful features to be used for mapping, localisation and navigation (Auat Cheein and Carelli, 2013).

The agricultural robot, with its on-board sensors, can sense the surroundings and build a map or a model of the environment with the most relevant features. The combination of different sensors has been employed to increase the robustness and reliability of the map (Bonin-Font et al., 2008). If the map of the environment is perfect, the robot can easily determine its position and orientation at each time. An agricultural robot may not be able to perform its task successfully if the elements from the environment are not properly located within the map. Moreover, an incomplete map cannot be used for autonomous navigation due to the risk of collision (Auat Cheein and Carelli, 2013).

Localisation is the process of accurately determining the mobile robot's pose (position and orientation) relative to a given map of the environment using the data acquired from the mobile robot sensors (Dudek and Jenkin, 2010). The core of the localisation problem is the reliable acquisition or extraction of sensor information and the automatic correlation or correspondence of this information with the environment map (Zhang et al., 2008). Localisation is a critical issue in mobile robot navigation and particularly in an agricultural environment where, unlike a factory environment, a level ground surface cannot be presumed. The mobile robot cannot effectively plan its path, locate objects and navigate to the target unless it knows its position in the environment (Bloss, 2008).

Simultaneous localisation and mapping (SLAM) is the technique used by mobile robots and autonomous vehicles to build up a map within an unknown environment, while at the same time using this map to estimate their current pose. In SLAM both the trajectory of the mobile robot and the location of all landmarks are estimated without the need for any *a priori* knowledge of location (Durrant-Whyte and Bailey, 2006; Williams et al., 2002).

Path planning is needed for the automatic operation of autonomous field robots. Typically, a path planning algorithm has to find an optimal path from starting point to the goal position so that no collisions with obstacles occur. The path is also required to be optimal with respect to a certain measure, for example traveling in minimum time or using minimum energy (Oksanen et al., 2005).

1.2.2 Limitations and benefits of agricultural environments

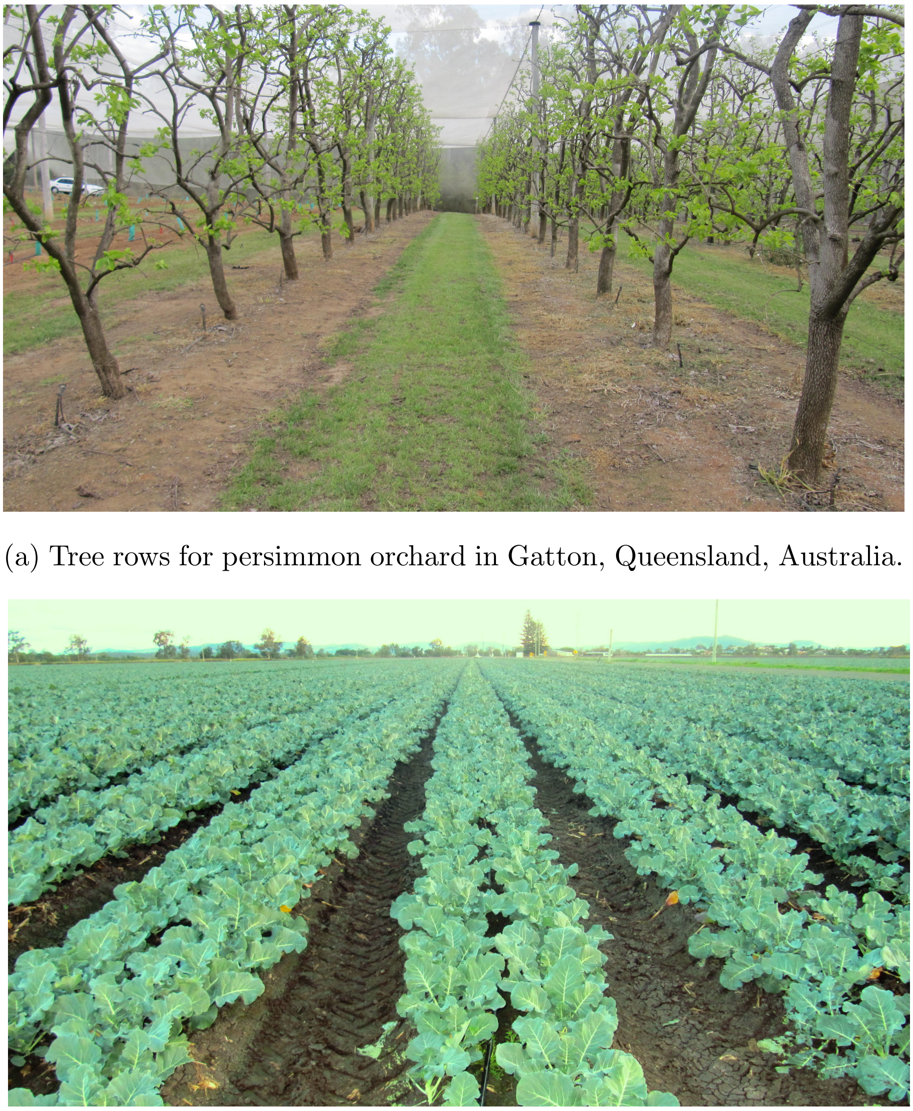
Agricultural environments consist of various elements such as plants, trees, weed, soil, objects, and landmarks (e.g. posts, fences). This diversity creates some difficulties and complications in the mobile robot navigation process. For example, the cultivated areas are large and the ground surfaces are usually uneven. Weather conditions such as rain, dust, fog, and sunlight may affect the data acquired by mobile robot sensors. Plant colour may change during different growth stages and the quality of soil may vary from one place to another (Li et al., 2009; Mousazadeh, 2013).

However, agricultural environments also provide some simplicity. For example, most crops or trees from the same kind are planted in straight rows and the intervals between the rows are almost equal as shown in Figure 1.2. The landmarks that already exist in the field can be used as stationary landmarks for localization and navigation algorithms.

1.2.3 Mobile robot sensors

The use of sensors in agricultural vehicles has increased rapidly in recent years. There have been substantial advances in the development of new and more powerful perception systems for agriculture, providing the automation of agricultural applications with important stimulus. These autonomous agricultural systems incorporate and integrate perception systems to acquire information from the surroundings, decision-making systems for interpreting and analysing such information, and actuation systems to perform various agricultural tasks. These systems consist of different sensors, actuators and computers that work together to perform a specific agricultural task (Emmi et al., 2014).

Navigation sensors provide information about the vehicle states (position, orientation, speed, etc.) and the objects in the surrounding environment. Some guidance sensors provide information for absolute positioning and others only provide relative



(a) Tree rows for persimmon orchard in Gatton, Queensland, Australia.

(b) Crop rows for cauliflower farm in Gatton, Queensland, Australia.

Figure 1.2: Examples of tree rows and crop rows.

positioning. Selection of the optimal sensors to provide the basic information for mobile robot navigation is a critical process. Different sensors such as vision, laser range scanner and Global Positioning System (GPS) are used in autonomous agricultural robot systems as primary sensing systems. Other sensors such as odometer and inertial measurement unit (IMU) are typically used as secondary sensors to complement the primary sensing systems. Table 1.1 illustrates the sensors and the information derived from these sensors using the required algorithms and manipulations for mapping, localisation and navigation in agricultural environment.

Table 1.1: Common mobile robot sensors and the information derived from them in agricultural environment using the required algorithms and manipulations.

Sensor type	Derived information
Camera	-Provide colour and intensity images of the agricultural environments. -Extract different features from trees and crops (e.g. colour, edge, texture, shapes, etc.). -Measure the relative position and heading of the mobile robot.
Laser scanner	-Provide precise range and angle measurements. -Detect objects in the surroundings and extract features such as edges, width, etc.
GPS	-Provide position measurements and guidance information.
IMU	-Measure the acceleration, angular velocity and orientation of the robot.
Odometer	-Measure the position and velocity of the robot.

1.3 Orchard environment

Orchards are usually semi-structured environments, since trees of the same type are planted in nominally straight and parallel rows and the distances between the rows are almost equal as shown in Figure 1.2a. In addition, the headland is available to drive the mobile robot to consecutive rows. The organised layout of the orchard makes it a suitable and promising environment for autonomous mobile robot navigation. However, the environment is only 'semi-structured' because the regularity of tree and non-tree object placement is neither precise nor reliable as the orchard grows, branches fall, etc.

1.3.1 Tree detection using mobile robots

Trees in orchards and groves are common natural landmarks providing suitable cues for mobile robot localisation and navigation. To navigate the mobile robot between the tree rows, the mobile robot must first detect the trees then determine its position related to the trees' positions. Therefore, individual tree detection is crucial in facilitating the localisation and navigation of the agricultural robots between the tree rows in the orchard.

Detecting trees robustly from a ground-level perspective is a challenging problem for several mobile robot applications. Different parts of the trees can be detected using the mobile robot on-board sensors such as trunk, stem and canopy. This detection depends on the type of the trees and the specific agricultural task. Several features can be extracted from the tree trunk and canopy such as colour, texture, edges, height and width to be used for mapping, localisation, navigation and tree inspection tasks.

1.3.2 Row detection and following using mobile robots

Since most of the trees in commercial farms, orchards and groves are planted systematically with a uniform row arrangement, many paths or inter-row spaces with a uniform width between the trees facilitate the movement of farmers between the rows. Hence, one of the main tasks of a mobile robot in an agricultural structured field, is to keep track of the rows so it can perform the plantation activities autonomously (Thamrin et al., 2012). For this, an important step is to develop a row detection system, which allows the mobile robot to accurately navigate along the row.

In the row recognition process, the problem is identifying the accurate features that are stable under different environmental conditions. In addition, the row detection process is accompanied with some difficulties such as incomplete rows, missing plants, and the irregular shape and size of the plants along the row. In addition, the presence of weeds along the row might distort the process of row recognition by adding some noise to the row structure which makes the row recognition task more difficult (Åstrand and Baerveldt, 2005).

1.3.3 Tree inspection tasks

Recent sensor developments allow growers to closely monitor and control many aspects of crop production. Robots are widely used for inspection tasks using diverse types of sensors. The robot with on-board sensing technologies can provide up-to-date information that helps the farmers with orchard management. Mobile robots with vision systems can be used to locate and differentiate between ripe and unripe fruits. Other applications include the development of robotic vehicles which can perform a range of tasks such as collecting information on plant health, monitoring plants growth and crop damage (Yang et al., 2011). With the advance of optical sensing technologies in agriculture, researchers have attempted different approaches to detect crop diseases and infections at an early stage (Grift et al., 2008). The inspection task is also needed for precision crop management including crop yield, soil properties, crop nutrients, crop canopy volume, and pest conditions (disease, weeds, and insects) (Lee et al., 2010).

1.4 Research problem

Studies in the literature have typically developed autonomous navigation systems for tractors or large agricultural vehicles (Barawid et al., 2007; Andersen et al., 2010; Hansen et al., 2011). In this study, a small robot platform is used rather than traditional (manually-driven) agricultural vehicles. A robot of small size operating autonomously has the potential to meet the major requirements of routine orchard inspection tasks which are labour intensive, especially in large orchards. The small size and light weight are important as they imply easy accessibility between tree rows and higher maneuverability when moving close to the tree rows and between trees in the row. Similar to traditional agricultural vehicles, small robot platforms are expected to have the potential to work in most weather conditions and behave sensibly in a semi-natural environments. They may have less energy requirements, are more cost effective, create less soil compaction and produce potentially less damage for trees and other objects in the orchard (such as irrigation infrastructure) compared to traditional agricultural vehicles (Pedersen et al., 2005; English et al., 2013; Blackmore and Griepentrog, 2002). In recent years, unmanned aerial vehicles (UAVs) have developed obvious potential for orchard aerial mapping and surveying. However, a small mobile robot is more convenient for collecting mapping and inspection information of the lower parts of the trees (such as trunks) which typically cannot be accessed easily by a UAV. Furthermore, UAVs require full manual control in such situations.

An orchard environment was selected in this study because of its semi-structured nature. However, the orchard environment presents some challenges especially in the robust detection of trees. This is mainly because of the non-tree objects that might be present between the trees in the tree rows such as posts and tree supports and animals. Thus, the development of a robust algorithm to detect trees and discriminate between trees and non-tree objects is necessary. It is hard to achieve this using single sensor such as vision only or laser scanner only since trees and non-tree objects might have some common features such as width, colour and parallel edges. Fusion of data from a camera and laser scanner was found to improve the tree detection accuracy because the laser scanner can provide reliable ranges, angles and widths of the tree trunks and non-tree objects, whilst the vision system can distinguish between tree trunks and other non-tree objects from different features (e.g. colour, edges, texture, etc.).

In some orchards, GPS cannot be used effectively for localisation and navigation as the agricultural robots frequently move under the tree canopy blocking the satellite signals from the GPS receiver (Li et al., 2009). In addition, using a precise GPS system such as Real Time Kinematic RTK-GPS is an expensive solution for position estimation. For these reasons, this study aims to develop a localisation system for mobile robot in an orchard without using GPS as the primary sensor for localisation. Furthermore, odometer is a very common sensor for position measurement that has been widely used due to its simplicity and low cost. However, this assumption is not always correct because of the wheel slippage on different surfaces which generate errors that accumulate over distance especially in outdoor environments (Mousazadeh, 2013). As a result, the mobile robot cannot rely only on the odometer to determine its position. All these challenges have motivated the development of a localisation system based on data fusion from different on-board sensors to provide accurate pose estimation of the mobile robot using the natural landmarks (i.e., trees) rather than using artificial landmarks.

The tree inspection and individual tree growth monitoring tasks require the mobile robot to have a map of the individual trees in the orchard. This facilitates the mobile robot's navigation to a specific tree to undertake these tasks. Hence, this arises the need for constructing a local map of the individual trees in the orchard rather than using the map of the tree rows as lines. The constructed map is essential for robust mobile robot localisation since the mobile robot needs to know its position relative to these trees in the row.

The mobile robot needs to execute different paths such as moving midway between tree rows, close to the row and between trees in the row to implement different tree inspection tasks. In addition, the movement of the mobile robot from one row to another requires executing either semi-circle turns or right angle turns. Therefore, the developed localisation algorithm is required to be capable of determining the mobile robot position for all these paths and turns.

The above challenges give rise to the following research questions:

1. *Can a camera and laser scanner combination be used in a commercial orchard environment and consistently detect trees with natural variation present in the orchard?*
2. *Is a small mobile robot with its on-board sensors capable of achieving sufficient maneuverability and localisation accuracy to move between tree rows*

and around individual trees?

1.5 Research aim and objectives

The aim of this research is to develop an orchard mapping and localisation system for a small mobile robot platform. This system is required to be capable of localising the mobile robot in different paths and turns in the orchard using the constructed orchard map of the individual trees based on tree trunk detection using camera and laser scanner data fusion.

The objectives of this study are:

1. To develop a tree trunk detection algorithm that can detect trees and discriminate between trees and non-tree objects using camera and laser scanner data fusion.
2. To develop a method for constructing a local orchard map of the individual trees and non-tree objects using the on-board mobile robot sensors to localise the mobile robot in the orchard and to enable the individual tree monitoring and inspection.
3. To develop a data fusion algorithm to estimate the pose (position and orientation) of the mobile robot for different paths and turns in the orchard.
4. To evaluate the performance of the developed algorithms through extensive experimental tests using a small mobile robot platform under different illumination conditions.

A low cost camera and 2D laser scanner are used as primary sensors in this study. Vision systems and laser scanners are becoming more common in outdoor agricultural applications such as tree detection, map construction, mobile robot localisation and navigation because of their ability to provide instantaneous information that can be used for feature extraction and object detection. Vision systems are low cost solutions for extracting different features (e.g. colour, edge, texture). Laser scanners are popular sensors in outdoor applications as they provide precise range and angle measurements in large angular fields under different illumination conditions. Fusing images from cameras with range data from laser scanners enables mobile robots and vehicles to more confidently perform a variety of tasks in outdoor environments (Garcia-Alegre et al., 2011).

1.6 Research contributions

The contributions of this research are as follows:

1. Development of a novel tree trunk detection algorithm that detects the tree trunks and discriminates between trees and non-tree objects using the fusion of low cost camera and 2D laser scanner data to enhance detection capability.

2. Development of a new method to detect and measure the width of the tree trunks and non-tree objects from laser scan data depending on the derivative data of the scan ranges.

3. The sequence of the projection of the edge points of the tree trunk and non-tree objects from the laser scanner coordinates to the image plane to construct a region of interest (ROI) with the required features for colour and edge detection. This was effective because the ROI is a small part of the image and required less processing time than processing the whole image. In addition it minimises the effect of the noise in the other parts of the image.

4. Automatic adjustment of the colour detection parameters after each test which was shown to increase the detection accuracy.

5. A local orchard map of the individual trees and non-tree objects of the rows was constructed rather than the whole row as line in the orchard. This allows

the mobile robot to navigate to a specific tree in the orchard to perform specific tasks such as tree inspection, pruning and harvesting.

6. The developed mobile robot localisation algorithm uses the positions of the individual trees in the correction step rather than correcting with the whole row as line. This will potentially contribute to the enhancement of the precision of in-row localisation.
7. The localisation algorithm was capable of localising the mobile robot in different paths (midway between rows, close to the rows and moving around trees in the row) and different turns (semi-circle turns and right angle turns).
8. The developed algorithms depend only on the on-board sensors of the mobile robot without adding any artificial landmarks, reflective tapes or tags to the trees.

1.7 Thesis outline

This dissertation is organised as follows:

- Chapter 2 presents the literature review of the related studies published in the field of this work. The key elements of the related work are tree detection and row following, orchard mapping, mobile robot localisation and tree inspection.
- The robot platform architecture used in this study is described in Chapter 3. This chapter also provides details of the sensors (laser scanner, camera, odometers, IMU and RTK-GPS) and the role of each sensor used in this study. In addition, a detailed description of the simulated environment used for preliminary tests is presented. Chapter 3 also describes the layout of the real orchard used for experimental tests.
- The design of the tree trunk detection algorithm using camera and laser scanner data fusion is explained in Chapter 4. This chapter describes in details

the main detection steps of the developed algorithm. Chapter 4 also presents the experimental tests and the evaluation of the performance of the tree trunk detection algorithm in the simulated environment and the real orchard.

- Chapter 5 demonstrates the method used for orchard map construction. The constructed map consists of the position of the individual trees and non-tree objects. Experimental tests and results for mapping the trees and non-tree objects in the simulated environment and the real orchard are also presented in this chapter.
- Chapter 6 deals with the mobile robot localisation algorithm using extended Kalman filter (EKF) to estimate the pose of the mobile robot in the orchard. This chapter also illustrates the experimental tests used to verify the performance of the developed localisation algorithm in the simulated environment and real orchard for different paths (midway between rows, close to the rows and moving around trees in the row) and different turns (semi-circle turns and right angle turns).
- Chapter 7 reports the overall conclusions with respect to the achievement of the objectives of this study. It also presents recommendations for potential applications of this research and future work.

Chapter 2

Literature review

Mobile robots in agricultural environments use perception sensors such as laser scanners and cameras to acquire information from the environment and detect potential trees, crops and objects. The data acquired by these sensors are primarily used to create maps of the environment, which allow localisation, path planning and navigation. Tree detection, feature extraction, mapping the robot's environment and mobile robot localisation have been central research topics in the field of developing robust and reliable navigation approaches for autonomous mobile robots in orchards and outdoor agricultural environments.

This chapter provides an overview of the studies reported in the literature related to the field of this PhD research. The main key topics are tree detection, row detection and following, orchard map construction and mobile robot localisation and navigation in orchards and other agricultural environments as shown in Figure 2.1. A summary of the significant conclusions is also presented along with the formulation of this research work.

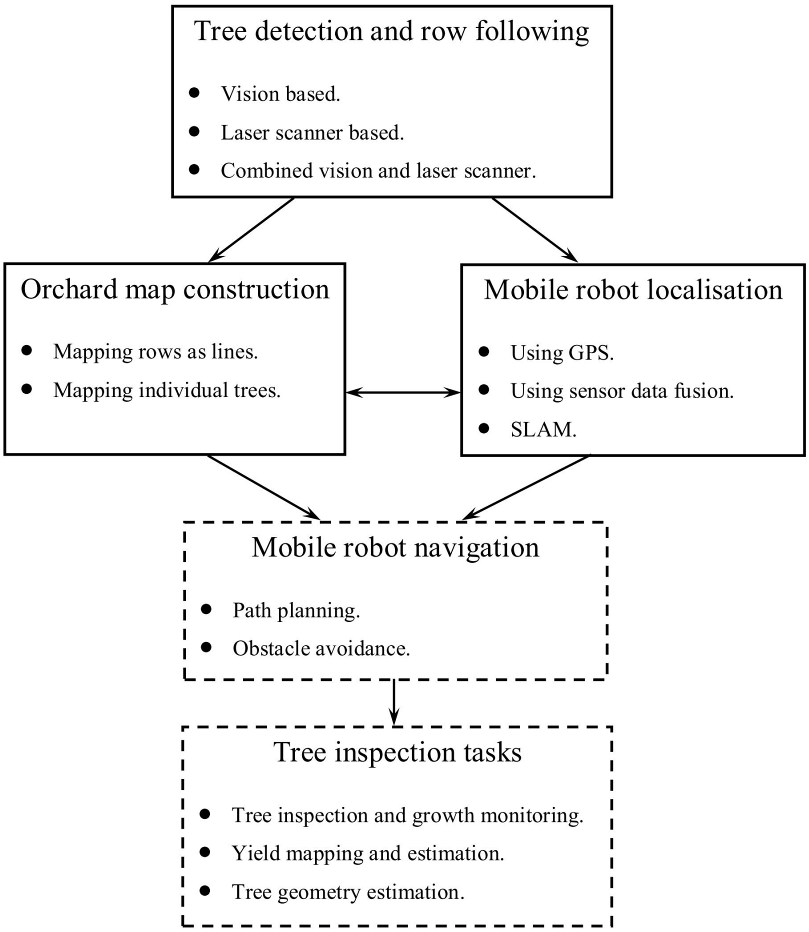

Tree detection and row following
Vision based.
Laser scanner based.
Combined vision and laser scanner.
Orchard map construction
Mapping rows as lines.
Mapping individual trees.
Mobile robot localisation
Using GPS.
Using sensor data fusion.
SLAM.
Mobile robot navigation
Path planning.
Obstacle avoidance.
Tree inspection tasks
Tree inspection and growth monitoring.
Yield mapping and estimation.
Tree geometry estimation.

Figure 2.1: The main topics of the literature review (the dashed blocks represent the future work and the potential applications of this research work).

.

2.1 Tree detection and row following using mobile robots

Natural landmarks such as trees are important in providing adequate information to localise the mobile robots in structured agricultural environments. In orchards and groves, one of the important tasks for an autonomous mobile robot is to detect the trees or tree rows and keep tracking between them in order to perform different agricultural tasks. Therefore, several methods have been implemented by researchers for tree detection, row detection and following in orchards and other similar agricultural settings. Laser scanner and machine vision have been used by many authors as the main sensors to detect various parts of the trees such as trunk, stem and canopy as described in the following sections.

2.1.1 Vision based tree detection and row following

Vision sensors have been widely used in mobile robot applications because of their cost effectiveness and their capability to provide information that can be utilised to generate steering control signals for the agricultural mobile robots. Vision systems are becoming more common in outdoor agricultural applications. The limiting disadvantages of using vision sensors are the influence of varying ambient lightening conditions (especially in outdoor environments) and the amount of computation required to extract useful features.

The use of vision only sensors allows the extraction of different features from the environment such as colour, texture, shape, and edges of the trees. For example, Ali et al. (2008) presented a classification based tree trunk detection method for the autonomous navigation of a forest vehicle using the integration of colour and texture cues to segment the images into tree trunks and background objects. They also proposed a distance estimation method that estimates the distance between the vehicle and the base of the detected trees. The tree detection performed well with an accuracy of 94.7%. However, their segmentation and distance estimation procedure assumed that the ground was relatively flat and there were no overhanging other objects in scene image.

Another related algorithm proposed by He et al. (2011) was designed to generate a navigation path in an orchard for a harvesting robot based on machine vision. A horizontal projection method was adopted to dynamically recognise the main tree trunk's area from orchard images. Border crossing points between the tree trunks and the ground were detected by scanning the trunks' areas, and these points were divided into two clusters on both sides. Their algorithm achieved a correct recognition rate of 91.7%. The authors concluded that this algorithm was extremely affected by the presence of the weeds in the orchard.

Autonomous navigation of mobile robots using vision sensors for tree row detection and following in orchards and groves is seldom used. It is difficult to find a possible path between rows because of the discrete planting of trees which breaks the visual continuity of a row and adds complexity to the segmentation methods. A few vision-based autonomous navigation systems have been developed for tree row following (Ayala et al., 2008; Gao et al., 2010; Torres-Sospedra and Nebot, 2011; Zhang et al., 2012). These systems used different image segmentation and classification methods to extract the useful information for navigation and focused on optimising the classification methods. They typically used line detection methods to detect the tree rows as lines to locate guidance paths.

In the work developed by Ayala et al. (2008), a robust system based on artificial vision for navigation of an autonomous mobile robot in olive groves, vineyards, and fruit plantations was proposed. The typical structure of olive groves cultivations provided visual information that can be used for navigation. This information consisted of perspective lines defined by the tree rows which motivated the use of visual sensors in these kinds of environments and image processing based on a statistical segmentation and Hough transform. Similarly, Hough transform was used by Gao et al. (2010), Torres-Sospedra and Nebot (2011) and Sharifi and Chen (2015) to detect tree rows as lines to determine the desired path between the tree rows. Figure 2.2 shows the tree rows where the borders between the land and the tree trunks are presented in blue lines and the centre of the path is presented by the green line (Torres-Sospedra and Nebot, 2011). Detecting tree rows as lines using a vision sensor was also implemented by Zhang et al. (2012). However, Zhang et al. (2012) used the RANdom SAmple Consensus (RANSAC) method to fit the 3D points corresponding to the trees into straight lines.

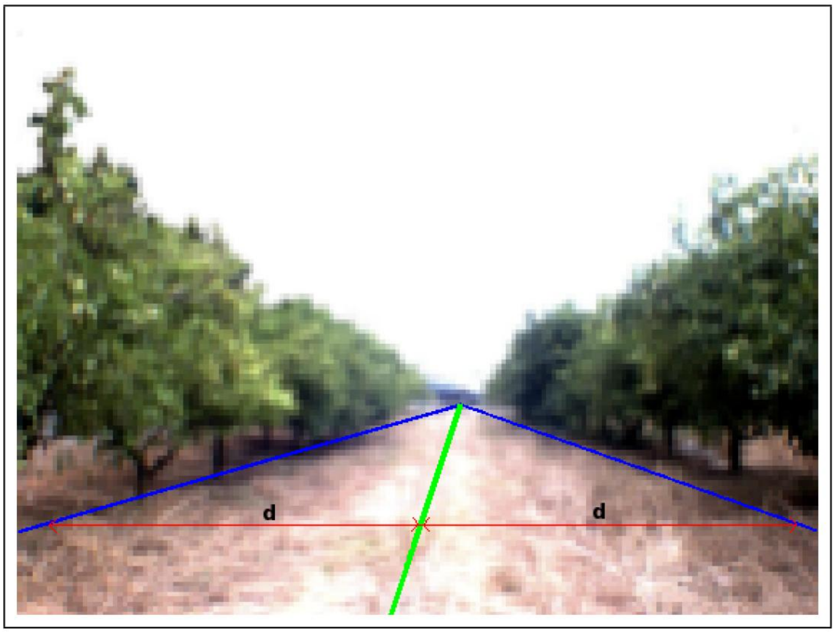


Figure 2.2: Detecting tree rows as lines using Hough transform, (Torres-Sospedra and Nebot, 2011)

.

2.1.2 Laser scanner based tree detection and row following

Tree detection and row following using laser scanners has been the source of many research contributions. This is mainly because laser scanners provide the benefits of high resolution and a large field of view. The laser scanner is one of the most popular devices in outdoor applications. It determines the relative distance of objects in the surrounding area by measuring the time of flight of laser pulses. One important advantage of laser sensors over visual systems is its ability to provide more robust ranging data for object detection under different weather and ambient illumination conditions.

2.1.2.1 Tree detection using 2D laser scanner

Studies have been reported on the utilisation of 2D laser scanners for tree detection and row following in orchards. Hansen et al. (2011) and Libby and Kantor (2011) used a 2D laser scanner to detect the dense canopy of tree rows. In this situation, the tree trunks were narrow and often occluded by leaves, and the canopies were closed such that the canopies of individual trees could not be distinguished. In addition, the dense vegetation of the tree rows provided a surface that is suitable for laser scanner measurements. Therefore, lines were fitted to the detected canopies in the tree rows using Hough transform and individual trees were not detected. Libby and Kantor (2011) enhanced the tree row detection by using reflective tapes to mark the start and the end of each row as shown in Figure 2.3. In a study presented by Bergerman et al. (2015), the authors developed a method to fit the laser points into two parallel straight lines representing the tree rows on both sides of the agricultural vehicle using RANSAC and least-square fitting methods.

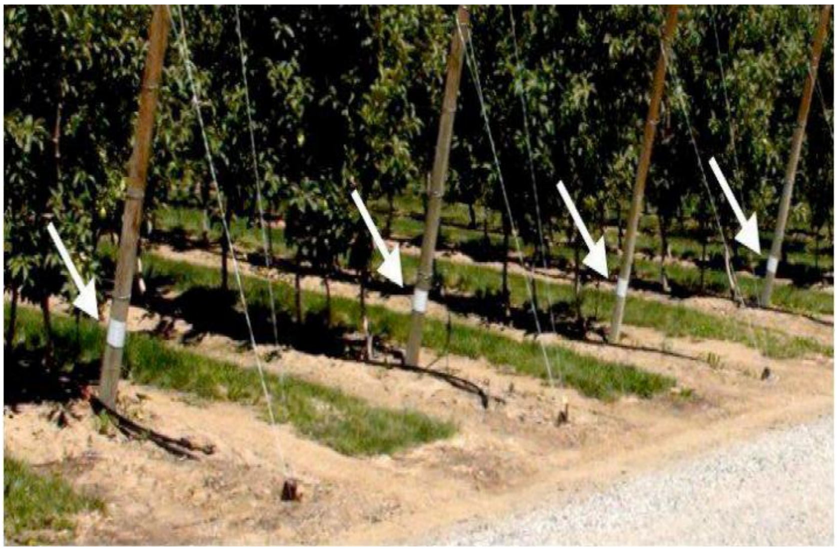


Figure 2.3: Reflective tape placed around posts at the ends of the rows (Libby and Kantor, 2011)

Hamner et al. (2010) suggested a method to detect the trunk and/or canopy of the trees for tree row recognition using a 2D laser scanner. Hough transform was then applied to extract point and line features to navigate the agricultural vehicle between the rows as shown in Figure 2.4. The 2D laser scanner was also used in the study reported by Guivant et al. (2002) to detect the trunks of the trees as point features, and the trees were used as landmarks to implement a SLAM algorithm.

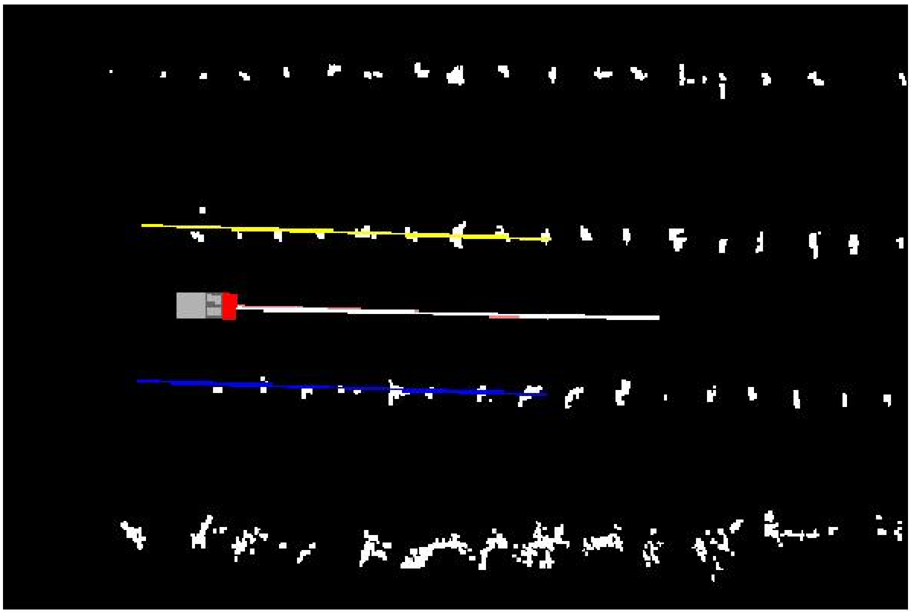


Figure 2.4: Hough transform for detecting tree rows using 2D laser scanner (Hamner et al., 2010)

Some studies focused on the use of terrestrial 2D laser scanners for the detection of trees and the estimation of tree trunk diameter in forests. Such laser scanners provide scan lines in a typically horizontal plane, and have been used to estimate the diameter and also the location of the centrepoint of trees (Jutila et al., 2007; Zheng et al., 2012; Ringdahl et al., 2013; Brunner and Gizachew, 2014; Kong et al., 2015). Jutila et al. (2007) estimated tree trunk diameters using the angle between the edges and the shortest distance to a laser point cluster. They found that the circle fit method is significantly less accurate. Usually, 2D laser scanners have large beam divergence for long distances. Therefore, Ringdahl et al. (2013) investigated the possibility of enhancing existing algorithms for tree detection and trunk diameter estimation to compensate for laser beam width. They also improved the algorithms with the fusion of several consecutive scans. They concluded that the estimation error grows with increasing distance, in particular for trees with a small diameter. A 2D laser scanner was used by Zheng et al. (2012) and Kong et al. (2015) to collect point clouds for standing trees, then a cluster extracting algorithm and filtering algorithm were used to classify each trunk from the point cloud. Ringdahl et al. (2013) suggested combining the laser scanner with a camera in future work to improve the detection and estimation capability; whilst Kong et al. (2015) suggested further work for dynamic diameter estimation combining a laser scanner with a camera to achieve real-time and rapid measuring results in the forest environment.

2.1.2.2 Tree detection using 3D laser scanner

Researchers have also investigated the use of 3D laser scanners for tree detection and row following. Zhang et al. (2013) developed a 3D perception method to extract tree row and trunk information from a 3D LIDAR point cloud that can be used by an autonomous vehicle to follow the tree rows in an orchard. They concluded that 3D laser scanners provided much better row information than 2D laser scanners in environments characterised by canopies with large volumes, branches sticking out into the row and objects such as pipes and rocks that make 2D sensing infeasible for row following. However, 3D laser scanners are expensive and require intense computing compared to 2D scanners.

Some recent studies developed algorithms using 3D terrestrial laser scanners to detect and estimate tree trunk geometry in orchards and forests (Olofsson et al., 2014; Lindberg et al., 2012; Raumonen et al., 2013; Méndez et al., 2014). These studies are based on different techniques such as the RANSAC algorithm (Olofsson et al., 2014), Hough transform (Schilling et al., 2011; Lindberg et al., 2012) or least square adjustments (Raumonen et al., 2013) to fit circles or cylinders to the data points belonging to the tree trunks in the scan data. They used algorithms which often require a high resolution 3D point cloud and focused on tree detection and measurement, but not specifically for mobile robot localisation and navigation. Range imagery (Bienert and Schneider, 2013) and full-waveform (Yang et al., 2013) data were also used in tree stem detection. However, these data need special instruments that are not widely used nowadays and more pre-processing is required
.

Alternatively, techniques such as LIDAR can provide more comprehensive canopy geometrical information and Rosell-Polo et al. (2009) have demonstrated its use for the measurement of tree-row structure in orchards, but not specifically for local mapping.

2.1.3 Combined vision and laser scanner based tree detection

Some research studies in the literature used both camera and laser scanner for tree detection in orchards. Their objective was to improve tree detection capability over the individual use of either a camera or a laser scanner. In the study presented by Auat Cheein et al. (2011), a camera and a laser scanner were used together for tree stem detection in an olive grove in which individual tree stems could be distinguished. The algorithm first detected the stems from the acquired image and returned the angle of the stem with respect to the mobile robot. The laser scanner was then used to confirm the presence of the stem in the neighborhood of the detected angle. Moorehead et al. (2012) and Subramanian et al. (2006) developed autonomous vehicle guidance in orchards and groves. They used a combination of visual and laser sensors to detect trees and map the environment. However, they did not target individual trees.

2.2 Crop row detection and following using mobile robots

Crops in farms are usually planted in approximately straight lines and parallel rows. The uniform layout of the crop rows and the ability to discriminate the crops from other objects or features (such as soil colour and row tilts) enable robot implementation in agriculture fields. These crop row features are used as the baseline for row detection and following, mobile robot tracking and navigation (Zhao and Jiang, 2010).

Researchers have explored the use of different vision sensors to find guidance paths on crop rows; for instance, detecting the position and orientation of crop rows relative to the vehicle and detecting the edges along harvested crops. These methods typically identified crop rows based on the crop colour and intensity difference to the soil. They mainly focused on the development of different image segmentation techniques and line fitting methods to extract the guidance information for crop rows applications (Billingsley and Schoenfisch, 1997; Okamoto et al., 2002; Benson et al., 2003; Han et al., 2004; Ortiz and Olivares, 2006; Gottschalk et al., 2010; Jinlin and Weiping, 2010; Ericson and Astrand, 2010; Ding et al., 2011; Lulio et al., 2012; English et al., 2013; Tian et al., 2014).

Various methods of image segmentation have been investigated to extract different features for crop row detection and following. Studies often combined two or more image segmentation techniques to detect the crop rows accurately. For example, Benson et al. (2003) studied the use of histogram based segmentation and edge detection to detect the crop rows, whilst Ericson and Astrand (2010) described a method of detecting parallel rows using a combination of an edge-based method and a Hough transform. In the paper presented by Gottschalk et al. (2010), a combination of histogram-based method, threshold function, and morphological imaging functions was used for extracting geometrical lines corresponding to the crop rows.

As most crops are cultivated in nominally straight rows, most of the image processing algorithms for crop row detection are based on line fitting methods such as Hough transform (Leemans and Destain, 2006; Bakker et al., 2008; Jiang et al., 2010; Ji and Qi, 2011; Behfar et al., 2014; Choi et al., 2015), linear regression method (Billingsley and Schoenfisch, 1997; Han et al., 2004; Montalvo et al., 2012; Jiang et al., 2015) or fixed template matching (Tillett et al., 2002).

Researchers reported the use of spectral filters to enhance the detection of vision-based guidance systems for row crop following application (Åstrand and Baerveldt, 2005; Kaizu and Imou, 2008). These systems achieved good performance in detecting plants with near-infrared (NIR) images. Stereo vision systems have also been researched for crop row detection systems and 3D field maps (Kise et al., 2005; Hanawa et al., 2012). These methods require a distinct height difference between the crop and ground and cannot be used on very young crops.

The common denominator among the crop row detection and following studies is the use of a fixed forward field-of-view camera arrangement, which works adequately in the case of tall, mature plants, but has limitations when the plants are small and when the robot turns at the end of the row (Xue et al., 2012). This problem was considered in the work presented by Xue et al. (2012) by developing a novel variable field-of-view machine vision method allowing an agricultural robot to navigate accurately between rows in cornfields. The machine vision hardware consisted of a camera with pitch and yaw motion control.

Studies have also reported crop row detection using a laser scanner (Satow et al., 2004; Weiss and Biber, 2011; Hiremath et al., 2014). However, laser scanners are less convenient for crop row detection and following because the laser scanner cannot effectively detect the crops, especially low or short crops.

As crops mature, more canopies grow together and rows overlaps. This can narrow the inter-row spaces and cause difficulties in detecting and discriminating the crop rows in view. Under these circumstances, the crops could not be detected accurately (Thamrin et al., 2012).

2.3 Orchard map construction

Tree rows in well-organised agriculture fields such as orchards provide sufficient information for the mobile robot to navigate and track its location. Therefore, it is essential for the mobile robot to detect the trees and tree rows in order to construct a substantive map that is useful for mobile robot localisation and navigation (Thamrin et al., 2014).

The literature offers several methods for the construction of a map of the orchard to be used later as an *a priori* map for mobile robot localisation and navigation. In some orchards, the trees are closely spaced in rows with branches spanning the area between the trees in the row, or the tree trunks have small diameters with branches hanging low to the ground. Therefore, the best option is to estimate the position of the tree rows rather than the individual trees (Andersen et al., 2010). Hansen et al. (2011) and Andersen et al. (2010) developed a simple map of the orchard containing the starting and ending points of the rows in the orchard. The map was formed in the Universal Transverse Mercator (UTM) coordinate system. From this map, straight line representation of the orchard rows can be obtained for localisation and navigation.

In the work reported by Libby and Kantor (2011), an *a priori* map was developed consisting of line features formed by rows of trees in the orchard and point features consisting of reflective tape placed at the ends of the rows. This map was used for localisation by detecting both the row line and the row ends. Similarly, Zhang et al. (2014) presented a method to create a local map for autonomous orchard navigation that relies on the placement of strips of retro-reflective tape on the posts at the end of the rows, which are easily detected by the LIDAR on the vehicle. These artificial landmarks increased the probability of finding the next row and entering it successfully.

In some orchard applications, it is important to map the individual trees to implement tree inspection tasks or yield maps. Creating a map of individual trees in the orchard using GPS introduce challenges since accurate tree position data cannot be obtained by the use of the GPS due to outages under the tree canopy. Tree recognition and counting alone are not sufficient for obtaining tree position because of the occurrence of missing trees in the row or false positive sensing caused by fallen branches (Heidman and Rosa, 2008).

The use of radio frequency identification (RFID) tags is another method to localise the trees in the orchard. Potentially, one tag per tree could help localise the tree. Such tags enable RFID readers to detect trees, even without the requirement of line-of-sight. Moreover, a mobile robot with an on-board RFID reader is able to detect and localise trees and objects without a computationally expensive vision system (Rohweder et al., 2009). In the paper presented by Ampatzidis et al. (2009), a novel location technology based on RFID has been proposed to establish each tree's position and the associations between trees and the bins for a yield mapping system. Special passive RFID tags attached to the trees and bins and a RFID reader located on the harvesting platform were utilised to identify the trees and the bins. However, there are concerns from farmers in implementing this technology due to tag acquisition costs and high labour maintenance issues which involve replacing missing tags and tracking of the tags. In addition, passive tags have a limited reading range and active tags have the issue of battery replacement after few years of use (Heidman and Rosa, 2008).

In the study presented by Heidman and Rosa (2008), tree localisation under the canopy was obtained from the combined information of the partially known a priori tree plot characteristics and real-time sensed odometric information. The method developed based on the interpoint distance method which successfully localised the trees in the orchard. However, their method assumed that the orchard tree grid spacing was nearly perfect, i.e., the trees were accurately placed with constant distances between them in the row. This method might produce some errors in orchards with tree placement errors (deviations from the planted positions) and missing trees. Estimation of tree positions in an orchard to create a graph map that can be used for mission planning and localisation for an autonomous robot was also presented by Jæger-Hansen et al. (2012). The authors used LIDAR to estimate the positions and foliage radius of the trees using ellipse fitting on point cloud.

2.4 Mobile robot localisation in orchards

Mobile robot localisation in orchard environments has motivated several works due to its importance to the autonomous navigation of mobile robots in orchards. Position data collected from the mobile robot can be used to relate its position to the orchard map. This allows the same robot to return to specific locations and perform tasks such as tree inspection, thereby saving valuable resources. Localisation is also critical for automated or semi-automated robots that can improve productivity for agricultural applications and fulfill the growing demand for labour.

2.4.1 Mobile robot localisation using GPS

The localisation system using GPS guidance technology has been widely used in many agricultural tasks. GPS navigation systems provide absolute position measurements which are used to navigate the mobile robot to perform many field applications. To increase the accuracy of the conventional GPS navigation system, additional technologies have been developed by many institutions. The more recently developed systems are the Differential Global Positioning System (DGPS) and the Real-Time Kinematic Global Positioning System (RTK-GPS).

Research has been reported using the RTK-GPS as the only positioning sensor for the automatic steering system of agricultural vehicles (Stoll and Dieter Kutzbach, 2000; Thuilot et al., 2001). Regardless of the type of GPS, this navigational technology has some limitations when the GPS system is used as the single position sensor for the localisation of mobile robots. Therefore, a GPS is often combined with other sensors to provide more accurate navigation information (Hellström, 2002). Examples of combining RTK-GPS with IMU can be found in Kise et al. (2002) and Eaton et al. (2010). Research has also been developed using RTK-GPS and fibre optic gyroscope (FOG) sensors for autonomous agricultural vehicles (Noguchi et al., 2002; Nagasaka et al., 2004). The work of Nagasaka et al. (2004) proposed the use of RTK-GPS with tilt sensor to develop an autonomous system for intra-row weed control.

The most common problems associated with the use of GPS for localisation and navigation involve the obstruction of line-of-sight to satellites, multi-path issues and interference from other RF sources (Hellström, 2002). In some orchards, GPS cannot be effectively used for localisation and navigation since the agricultural robots frequently move under the tree canopy, which blocks the satellite signals to the GPS receiver (Li et al., 2009). For this, many researchers are developing autonomous navigation systems for mobile robots in agricultural environments without using GPS as a primary sensor for navigation.

2.4.2 Mobile robot localisation using sensor data fusion methods

The literature demonstrates the effective use of sensor data fusion methods to estimate the pose of agricultural robots in orchards. For example, EKF provides a theoretical framework for multi-sensor data fusion to estimate the pose of agricultural robots once appropriate models of the robot and the sensors are defined. EKF is commonly used to combine laser scanner data with the data from other sensors for mobile robot localisation in orchards (Hansen et al., 2009; Mogensen et al., 2009; Subramanian et al., 2009; Libby and Kantor, 2011; Bergerman et al., 2015). In the study presented by Libby and Kantor (2011), the EKF used two types of laser-based correction steps. The first used point features (ends of rows), whilst the second used line features (tree rows). Similarly, Bergerman et al. (2015) developed navigation system that used the vehicle's current location and guidance commands as inputs for tree row following and turning in orchards. The localisation system was based on range data acquired by a 2D laser scanner to premapped landmarks at the beginning and the end of each tree row. The landmarks consisted of posts with reflective tape installed on each post. EKF with one prediction step and two update steps (row update and landmark update) was used to estimate the pose of the agricultural vehicle as shown in Figure 2.5. The row update step used the tree rows detected as lines, whilst the landmark update step used the position of the mapped landmarks.

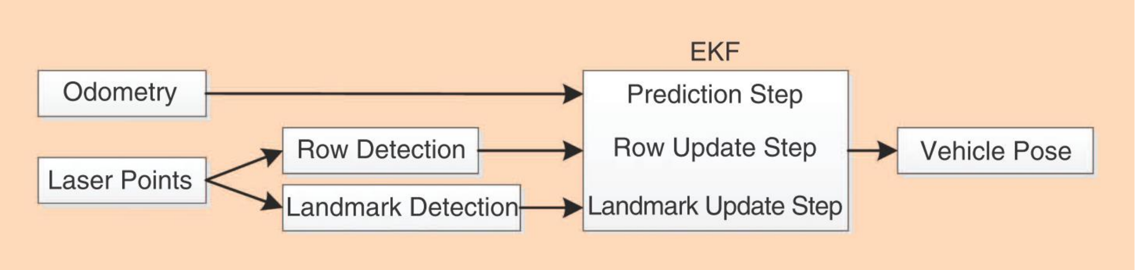


Figure 2.5: A block diagram of the localisation system using EKF using two update steps (Bergerman et al., 2015)

To improve the performance of EKF, it is often combined with different optimisation techniques and control strategies. Subramanian et al. (2009) developed a fuzzy logic enhanced KF for sensor data fusion to guide an autonomous vehicle in the orchard. The guidance system was then tested in citrus grove alleyways, and average errors of 7.6 cm at 3.1 m/s speed and 9.1 cm at 1.8 m/s speed were observed. In the work reported by Hansen et al. (2011), the performance of EKF was compared with derivative free filters for mobile robot localisation and navigation in an orchard using a 2D laser scanner. The localisation solution used the tree rows as measurements to correct the pose estimated by the filters. The authors concluded that EKF performed better; however, the difference between EKF and derivative free solutions was within an acceptable boundary.

Another localisation algorithm was developed by Andersen et al. (2010) which utilised an *a priori* map with rows mapped as lines. The localisation algorithm detected tree rows as well as the row ends based on laser scanner measurements. The localisation error, laterally, when driving between rows had a standard deviation of 10.3 cm, and at the row ends a standard deviation of 16.7 cm longitudinally. A 2D laser scanner was also used by Barawid et al. (2007) to develop an automatic guidance system capable of navigating an autonomous vehicle between tree rows. Hough transform was used to recognise the tree rows. The auto-regression method was used to obtain the minimum lateral and heading mean error. Figure 2.6 shows an outline of the orchard rows recognition using Hough transform (Barawid et al., 2007).

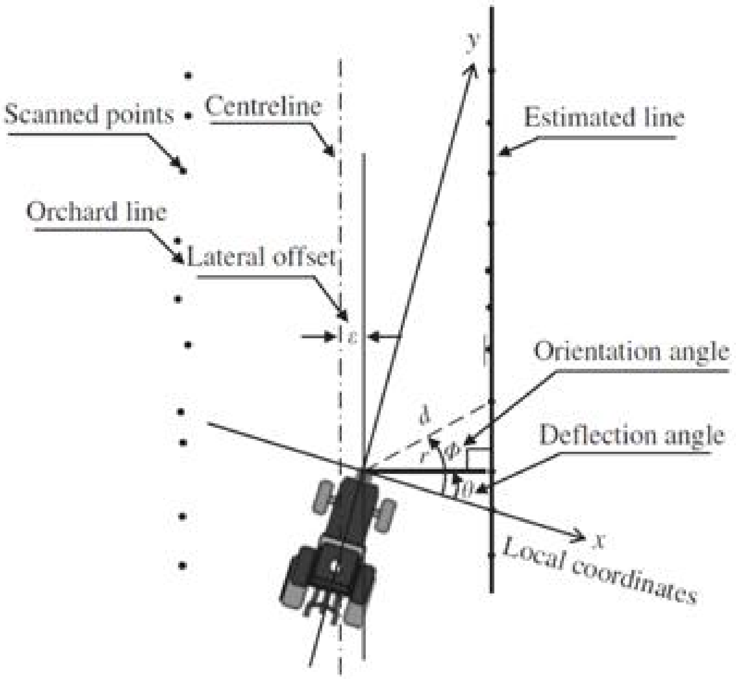


Figure 2.6: Outline of the orchard rows recognition using Hough transform (Barawid et al., 2007)

Particle filters (PF) are also used to solve the localisation problem. Unlike the EKF, the PF is not restricted to Gaussian processes and it better manages the non-linearities associated with the estimation process (Auat Cheein et al., 2011). González et al. (2009) developed a mobile robot localisation using Ultra-Wide-Band (UWB) range measurements and PF. The position of a mobile transceiver was determined from the distances to a set of fixed, well-localised beacons. The overall positioning errors (x, y) were 0.2 m, while the mean heading error was 10°. However, this method was only tested in an indoor environment. In the study presented by Kurashiki et al. (2010), a self-localisation algorithm consisting of a 2D laser sensor and a PF to handle the sensing uncertainties caused by unexpected objects was developed. Experimental results were obtained to travel through a real orchard and the standard deviation of the control error in the lateral direction was less than 15 cm. Hiremath et al. (2014) presented a PF based navigation algorithm for autonomous navigation in a maize field using a 2D rangefinder. The algorithm was designed and tested in various field conditions with crop rows, but not specifically with tree rows.

2.4.3 Mobile robot simultaneous localisation and mapping (SLAM)

SLAM algorithm is used to build up a map within an unknown environment, while at the same time using this map to localise the mobile robot's current location. SLAM can be implemented using different types of filters reported in the literature. For example, the Extended Information filter EIF-SLAM implemented by Auat Cheein et al. (2011) using vision system and laser sensor and based on detecting the tree stems. The error in the x or y coordinates of the robot's position did not exceed 0.5 m.

EKF can be considered as the most widely used algorithm to solve the SLAM problem and has been introduced in some studies. Guivant et al. (2002) and Christiansen (2011) implemented EKF-SLAM using a 2D laser scanner. EKF-SLAM has disadvantages in processing time and computational requirements. The complexity of EKF-SLAM increases with the number of landmarks and features in the map (Bailey and Durrant-Whyte, 2006). However, recent studies are seeking to address these issues (Christiansen, 2011). Christiansen (2011) suggested a method for controlling the size of the map in the SLAM algorithm by adding the new landmarks and removing the old landmarks to keep a fixed number of landmarks to be logged.

2.5 Agricultural inspection tasks

Agricultural robots for orchard applications have been introduced for inspection and supervision tasks. With the advance of electronic and information technologies, various sensing systems have been developed to provide increased information to farmers. Accurate information concerning the spatial variability within fields is very important for precision farming of specialty crops and has great potential to improve overall operational efficiency (Lee et al., 2010).

Canopy height, width and volume of trees have a key role in assessing the efficiency and effectiveness of the main operations performed in orchards such as precise fertilizers, irrigation, chemical applications and health assessment. Several studies have shown the existence of a relationship between the canopy geometrical parameters and yield (Pascual et al., 2011) and pruning and harvesting (Sanz et al., 2011; Rosell-Polo and Sanz, 2012).

There have been several attempts to achieve canopy detection, using different sensors such as vision, ultrasonic, and laser sensors. Wei and Salyani (2005) applied a laser scanning system to measure the foliage density of a citrus canopy, whilst Campoy et al. (2010) presented a canopy sensing system for modeling the tree canopy. They used a laser range finder to generate a 3D grid map of the canopy. Tumbo et al. (2002) reported a comparison between a laser scanner and ultrasonic transducers to measure the canopy volume of citrus trees while Schumann and Zaman (2005) developed a real-time software system to map citrus tree canopy volume and height using ultrasonic sensors. In the study presented by Han and Burks (2009), a 3D reconstruction of a citrus canopy was studied using vision system, whilst Grocholsky et al. (2011) demonstrated that laser sensing and computer vision can provide high resolution automated canopy volume and crop yield estimates for vineyard management. Moreover, multi-spectral imaging was used to detect disease in citrus trees (Gonzalez-Mora et al., 2010).

In the study reported by Changyi et al. (2015), a robust vision-based algorithm for detecting and counting apples based on their color and shape modes was proposed. The work presented by Cubero et al. (2015) shows the development of an efficient computer vision system to be mounted on agricultural vehicles with the aim of monitoring the yield and inspecting individually the quality of the production during harvesting. The use of 3D cameras have been investigated in agricultural autonomous inspection such as 3D reconstruction of apple trees (Karkee et al., 2014) and localisation of fruit in trees for crop-load estimation and harvesting (Gongal et al., 2016).

2.6 Conclusion

The autonomous navigation of mobile robots is having an increasing presence in agriculture. However, more research must be undertaken to improve technology, overcome limitations and decrease cost. The selection of sensors for row detection and following, mapping, localisation and navigation depends on the specific agriculture applications, the existing natural landmarks and objects in the selected environment, and the features required for these applications. From the reviewed literature, the use of sensors for mobile robot applications in orchards and crop rows can be summarised as follows:

- Camera and laser scanner can be considered as the most promising sensors used as primary sensors for feature extraction, mapping, localisation and navigation of mobile robots. Each one can be used individually or they can be used together with other secondary sensors such as odometer and IMU (Hansen et al., 2011).
- The majority of the research using vision sensors has been implemented in crop rows rather than tree rows. Vision sensors are more suitable for detecting navigation paths in crop rows because of the continuous plantation of crops

which provides simplicity in detecting a line though the rows using suitable image segmentation methods (Jinlin and Weiping, 2010; Jiang et al., 2015). On the other hand, the discrete planting of trees provides difficulty in finding a possible path between rows using vision sensors and adds complexity to the image segmentation methods.

- Laser scanners are more convenient for tree row detection and following than crop rows because the laser scanner cannot effectively detect the crops especially the low or short crops, while they can effectively detect trees and provide accurate ranges and bearing angles of the surrounding trees in orchards (Libby and Kantor, 2011; Christiansen, 2011).

Trees in orchards and groves are usually planted in nominally straight and parallel lines. Tree and tree row detection depend on the type of the trees in the orchards, the spacing between the trees in the row, the available features that can be extracted from the trees and the other objects existing between the trees in the rows. The main aspects regarding the studies in the literature for tree detection, row detection and following and orchard mapping can be summarised as follows:

- Most of the studies in the literature focused on detecting tree rows and mapping them as lines rather than detecting and mapping the individual trees in the row (Hansen et al., 2011). For example, when the trees are closely spaced and the trunks are thin and often occluded by leaves, or the tree rows are very noisy because of the irregular shape of the tree canopies. This makes the detection and mapping of the individual trees a challenge (Andersen et al., 2010). However, some algorithm detected and mapped the end of the rows as point features as well as the tree rows as lines between the row ends (Libby and Kantor, 2011). Other studies detected the individual trees based on tree trunk, stem or canopy detection then line fitting methods such as Hough transform, RANSAC method and least square method were often used to fit lines through these detected trees (Hamner et al., 2010).

- Few studies in the literature detected and mapped the individual trees in the orchards based on tree trunk, stem or canopy (Guivant et al., 2002; Auat Cheein et al., 2011). However, the prior work (cited in this chapter) paid considerably less attention to the need to discriminate between trees and non-tree objects in the tree rows of the orchard.

- Some of the studies of tree row detection and mapping used artificial landmarks such as tags or reflective tapes placed on the trees or at the end of the tree rows to detect and map the tree rows (Libby and Kantor, 2011).

- Most of the studies reported for tree trunk detection using a 2D or 3D laser scanner were based on circle or cylinder fitting methods (Olofsson et al., 2014; Lindberg et al., 2012). However, algorithms based on circle fitting are sensitive to noise in the laser scans and often provide large estimation errors (Ringdahl et al., 2013).

Localisation systems of mobile robots in orchards are mainly based on sensor data fusion methods, the natural landmarks (i.e., trees) and other landmarks in the orchard to associate the position of the mobile robot to these trees or landmarks. Localisation of the mobile robot in orchards reported in the literature can be summarised as follows:

- GPS is more reliable for localisation and navigation of the mobile robot in crop rows rather than tree rows in orchards. This is mainly because of the closed canopy which frequently blocks the satellite signal to the GPS receiver when the robot moves under the tree canopies (Li et al., 2009). However, GPS can be used in the headland area where there are no trees.

- The majority of the studies reported in the literature focused on developing localisation and navigation systems to localise and navigate the mobile robot in midway along tree rows and perform a semi-circle turn at the end of the row to enter the next row (Christiansen, 2011; Hansen et al., 2011; Mogensen et al., 2009).

- Most of the studies in the literature paid considerably less attention to handling the problems with headland turning efficiently. In addition, localisation and navigation might often fail when a clear row signal is missing, such as an incomplete row or missing trees in the rows. This often happens at the headland area where the agricultural vehicle cannot detect the trees.

- From the literature, it can be seen that EKF can be considered as the most promising techniques for localisation and SLAM (Mogensen et al., 2009; Subramanian et al., 2009; Bergerman et al., 2015). It is an effective tool that is applicable to non-linear robotic systems and allows the integration of measurements from the complementary sensors to estimate the pose of the mobile robot.

- From the studies presented in the literature, it is obvious that most of these studies update the position estimation of the mobile robot with the tree rows as lines in the correction step of the EKF. Some studies used two correction steps such as tree rows update step as line features and end of rows update step as point features (Libby and Kantor, 2011; Bergerman et al., 2015).

2.7 Formulation of research work

From reviewing the related studies in the literature, the following points identify and highlight the research gap and the formulation of the research work as the starting points for further analysis:

- The focus of this research is to detect the individual trees in the orchard rather than detecting the tree rows as lines.

- Combining the feature extracted from the camera images and 2D laser scanner data may enhance the detection capability and facilitate the discrimination between trees and non-tree objects in the orchard.

- It is important to map the position of the individual trees in the row rather than the whole row as a line in the orchard. This helps the mobile robot to potentially navigate to a certain tree in the orchard to perform tree inspection and monitoring tasks.

- The focus on using the natural landmarks (i.e., trees) and the objects that already exist in the orchard to develop the tree detection, mapping and localisation algorithms rather than using artificial landmarks which add cost to the farmers.

- The information gained from the recent literature indicated that EKF method could provide good promises and results for localisation of mobile robot in orchards. However, the EKF correction step in this study will depend on the positions of the individual trees rather than correcting with the whole rows as line features and the end of rows as point features. This will potentially contribute to enhance the precision of in-row localisation.

- Multiple paths will be investigated as the studies in the literature did not investigate the localisation of the mobile robot in different paths. For example, in some harvesting, pruning, thinning and tree inspection tasks, the mobile robot is required to move close to the tree row and follow a line with a constant distance to one side. In addition, turning around the trees in the row to inspect the trees from different angles is required. Moreover, the right angle turn in the headland has not been explored in the literature. The right angle turn is attractive as it requires less space than the semi-circle turn and can be executed by a small mobile robot with less soil damage compared to large agricultural vehicles.

Chapter 3

Orchard robot equipment and environment

In recent years, the use of sensors with agricultural robots has increased as their costs decline. The mobile robot, with its on-board sensors, can accomplish different tasks in outdoor agricultural environments. The selection, coordination and combination of sensors to provide the information required for specific tasks is a critical process. The placement of the sensors on the mobile robot platform plays a vital role in sensors' data fusion and relies on the type of the agricultural task and the feature of interest from the environment. Data fusion from different sensors provides a good estimate of the mobile robot states.

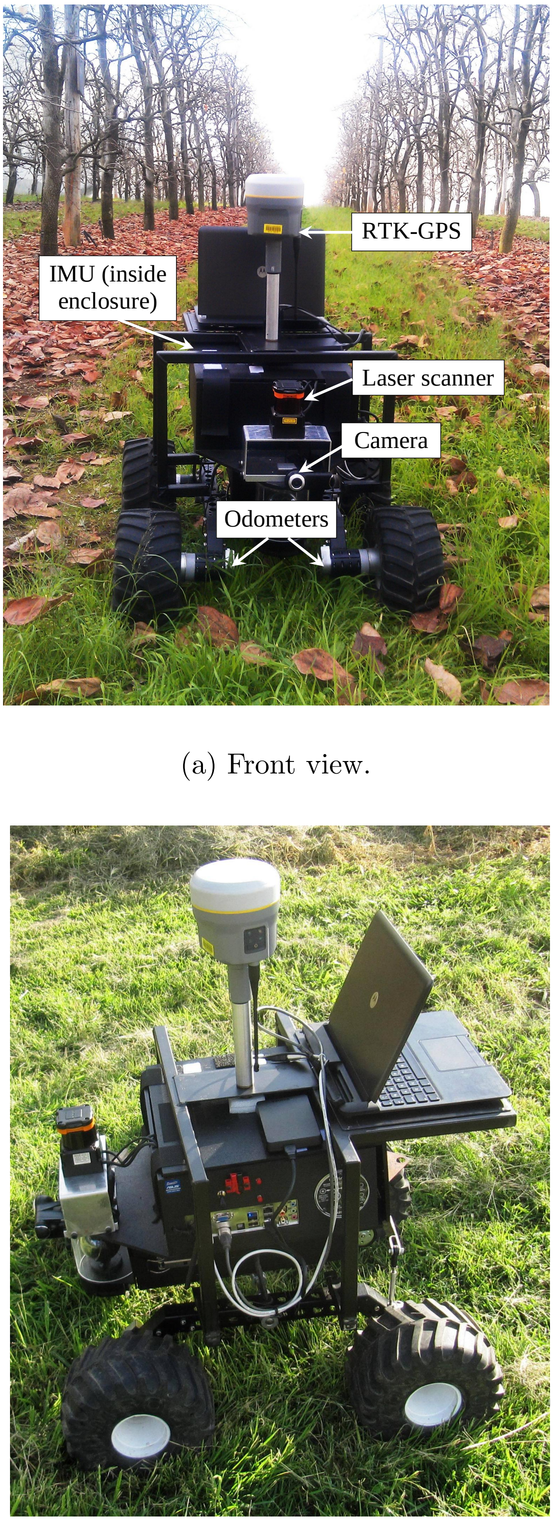
This chapter illustrates the mobile robot architecture and demonstrates the role of each sensor in this study. It also explains the main errors in the sensor measurements and their influence on the accuracy and performance of the designed algorithms. This chapter also focus on methods of pre-processing the data by means of filtering and rectifying before sensor data fusion. In addition, the simulated environment constructed from artificial trees and used for preliminary evaluation of the developed algorithms is presented. The orchard layout and the major components of the orchard that might affect this study, and their influence on the sensor's measurements, are also discussed.

3.1 Mobile robot architecture

The platform used in this research is a CoroWare Explorer (CoroWare Inc., USA) which is designed to operate in outdoor environment conditions. It can withstand environmental elements such as dirt, dust, leaf trash, sand, gravel, and shallow puddles. It has a rugged 4 wheel drive chassis with skid steering. The robot has the ability to climb over 150 mm obstacles allowing it to move over places difficult for other robot platforms with lower clearance to access. It is equipped with an on-board computer running the Ubuntu Linux operating system with the Robot Operating System (ROS) for building and running generated C++ and Matlab codes. Whilst this robot was used principally as a convenient sensor platform, a robot of this size, operating autonomously, could meet a major requirement in orchard management, namely simple inspection, e.g. to observe the state of flowering, crop development, or damage following a storm event carrying little more than a camera mounted on a suitable height and communications systems to capture the required images for tree inspection task. Currently this is a labour intensive task, especially in large orchards. A small robot with such size can also maneuver around trees with ease.

Figure 3.1 shows the CoroWare Explorer platform with the on-board sensors. A camera-laser scanner combination shown in Figure 3.2 is mounted on the robot platform. The laser scanner is mounted on top of the camera and the camera-laser scanner combination is positioned at the front of the robot. The robot platform is also equipped with an IMU, two odometers attached to the right and left front wheels and the RTK-GPS used for measuring the ground truth.

The on-board computer of the mobile robot is equipped with an external hard drive used for on-board sensor data logging. The data from each sensor is logged with



(a) Front view.

(b) Side view.

Figure 3.1: CoroWare Explorer platform with on-board sensors.

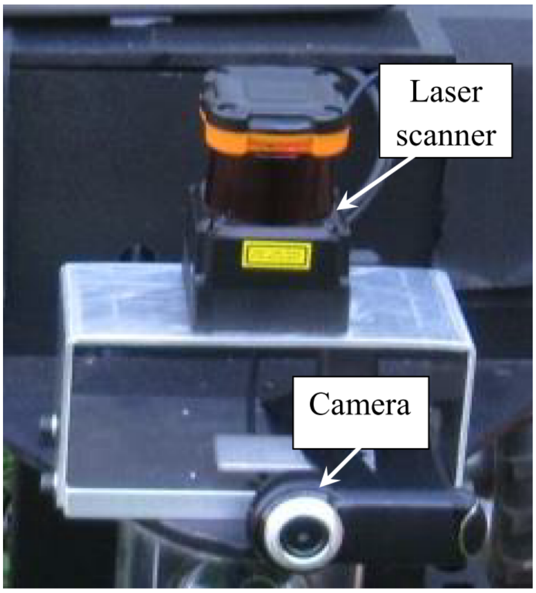


Figure 3.2: The camera-laser combination.

a time-stamp which represents the moment the data was received and stored. The acquired data is then easily used for later sensor data fusion implementation.

3.2 Mobile robot sensors

In this study, vision and laser scanner are used as the primary sensing systems for feature extraction, tree detection, orchard mapping and mobile robot localisation. Other sensors, such as odometer and IMU, are used as secondary sensors to complement the primary sensing systems. The RTK-GPS is used to measure the ground truth of the mobile robot positions that are used as reference positions to evaluate the estimated positions.

3.2.1 Laser scanner

The 2D laser scanner generates a single horizontal scan of the environment. Each scan produces a polar map of the surrounding by combining the distances with the angle of each object. This polar map can be converted to a 2D Cartesian map of the environment.

The 2D laser scanner is used in this study as the primary sensor because of its benefits of high resolution and large field of view. In addition, it provides accurate relative distances and bearing angles of objects in the surrounding area by measuring the time of flight of the laser pulses. One important advantage of laser sensors over visual systems is the ability to provide robust ranging data for object detection and localisation under ambient illumination conditions in outdoor agricultural environments (Weiss and Biber, 2011). In this study, the laser scanner is used to detect the surrounding trees in the orchard. The trees planted in rows provide nearly continuous data for the mobile robot traveling between rows to be used for mapping, localisation and navigation.

The laser scanner used in this research is a Hokuyo UTM-30LX/LN scanning laser range finder as shown in Figure 3.2. It is a small, affordable and accurate laser scanner that is perfect for small robotic applications. It operates with a scanning angle of 270°, angular resolution of 0.25° and maximum detectable range equal to 30 m as shown in Figure 3.3. The supply voltage of the laser scanner is 12V and the current consumption is 700mA which allows it to be used on battery operated platforms. More information about the laser scanner specifications can be found in Appendix A.

The height of the laser scanner from the ground is 450 mm. The laser scanner is placed at this height because it provides the best view of the tree trunks. Any lower position will be affected by weeds, high grass, piles of fallen leaves and irrigation fittings. Any higher position will detect the tree branches which might affect the laser scanner measurements.

Some external parameters are expected to produce errors on both angle and distance measurements of the laser scanner:

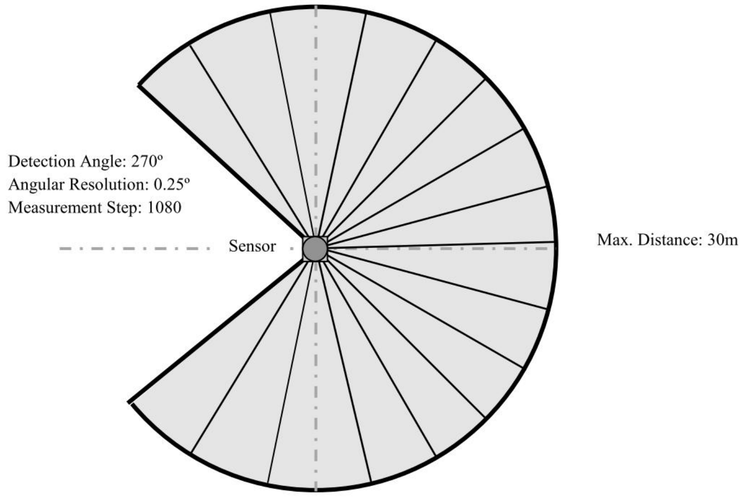


Figure 3.3: The Hokuyo UTM-30LX/LN laser scanner detection angle and distance (Hokuyo, 2008).

- The velocity of the mobile robot affects the number of the scans of each tree or object.
- The vibration produced by the driving motors of the platform affects the projection position of the laser scan on the trees and objects.
- The terrain roughness might introduce more vibration that affects the laser scanner measurements.

3.2.2 Camera

Cameras are low cost sensors providing spatial and spectral information about the scene making it a good candidate for the main detection sensors (Subramanian et al., 2009). The acquired images allow the extraction of different features from the environments such as colour, texture, shape, and edges of the objects. The camera was used in this study to verify the colour and the edges of tree trunks and the non-tree objects in the tree rows of the orchard.

The camera used in this study is a Logitech webcam Pro 9000 (see Figure 3.2) with 75° view angle. It has a high performance CMOS sensors with resolution of 2 megapixel. The image resolution has been set to 640 × 480 pixels, as higher resolutions imply greater computing time. The height of the camera from the ground is 330 mm. To achieve the laser and vision data fusion, camera calibration and camera laser calibration are required. These are described in Sections 3.2.2.2 and 3.2.2.3 respectively.

3.2.2.1 Pinhole camera model

The pinhole camera model defines the geometric relationship between a 3D point in the world (P) and its 2D corresponding projection (p) onto the image plane. When using a pinhole camera model, this geometric mapping from 3D to 2D is called a perspective projection. The center of the perspective projection (the point at which all the rays intersect) is known as the optical center or camera center, and the line perpendicular to the image plane passing through the optical center as the optical axis. The intersection point of the image plane with the optical axis is called the principal point (c_c). The focal length (f_c) is defined as the distance between the optical centre and the image plane (Cyganek and Siebert, 2011). The pinhole camera that models a perspective projection of 3D points onto the image plane is shown in Figure 3.4

3.2.2.2 Camera calibration

Camera calibration is one of the most fundamental problems in robot applications that deal with vision systems. It is the process of determining the intrinsic and/or extrinsic parameters of the camera with respect to known world coordinates. Intrinsic parameters are the internal parameters of the camera which include the focal length f_c, the principal point coordinates c_c, skew coefficient α_c, and the distortion coefficients k_c. Extrinsic parameters represent the camera position and orientation relative to the world coordinates. Calibration is also an essential first step for

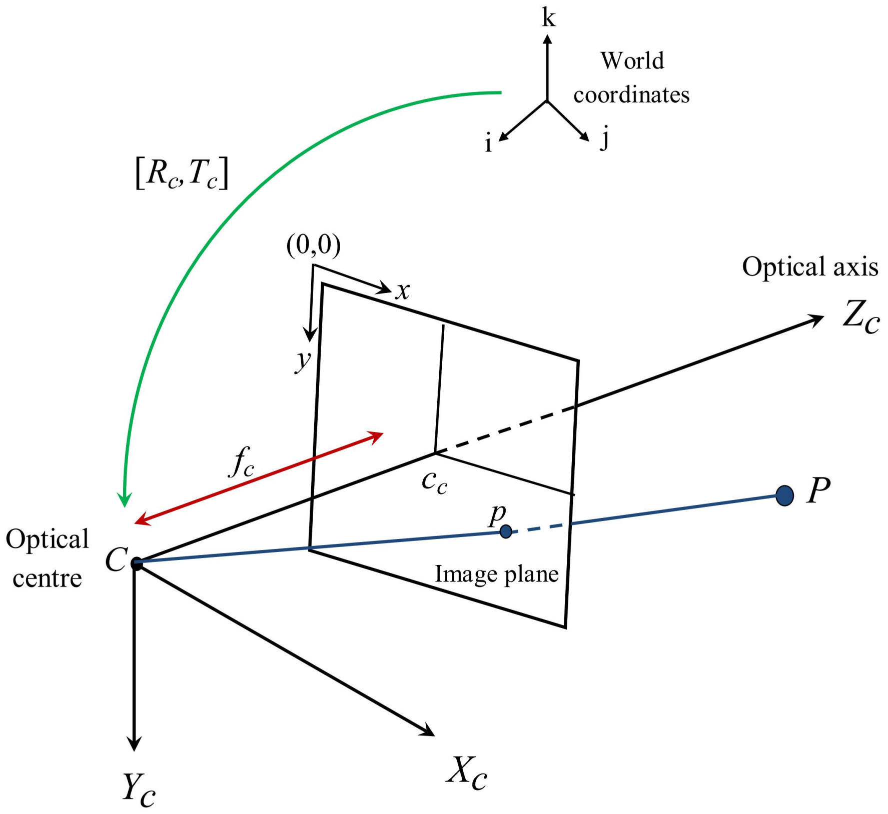


Figure 3.4: The pinhole camera model.

minimizing lens distortion and improving the accuracy of object detection.

The intrinsic parameters are:

- Focal length: The focal length in pixels is stored in the 2×1 vector f_c.
- Principle point: The principle point coordinates are stored in the 2×1 vector c_c.
- Skew coefficient: The skew coefficient defining the angle between the x and y pixel axes is stored in the scalar α_c.
- Distortion coefficients: The image radial and tangential distortion coefficients are stored in the 5×1 vector k_c.

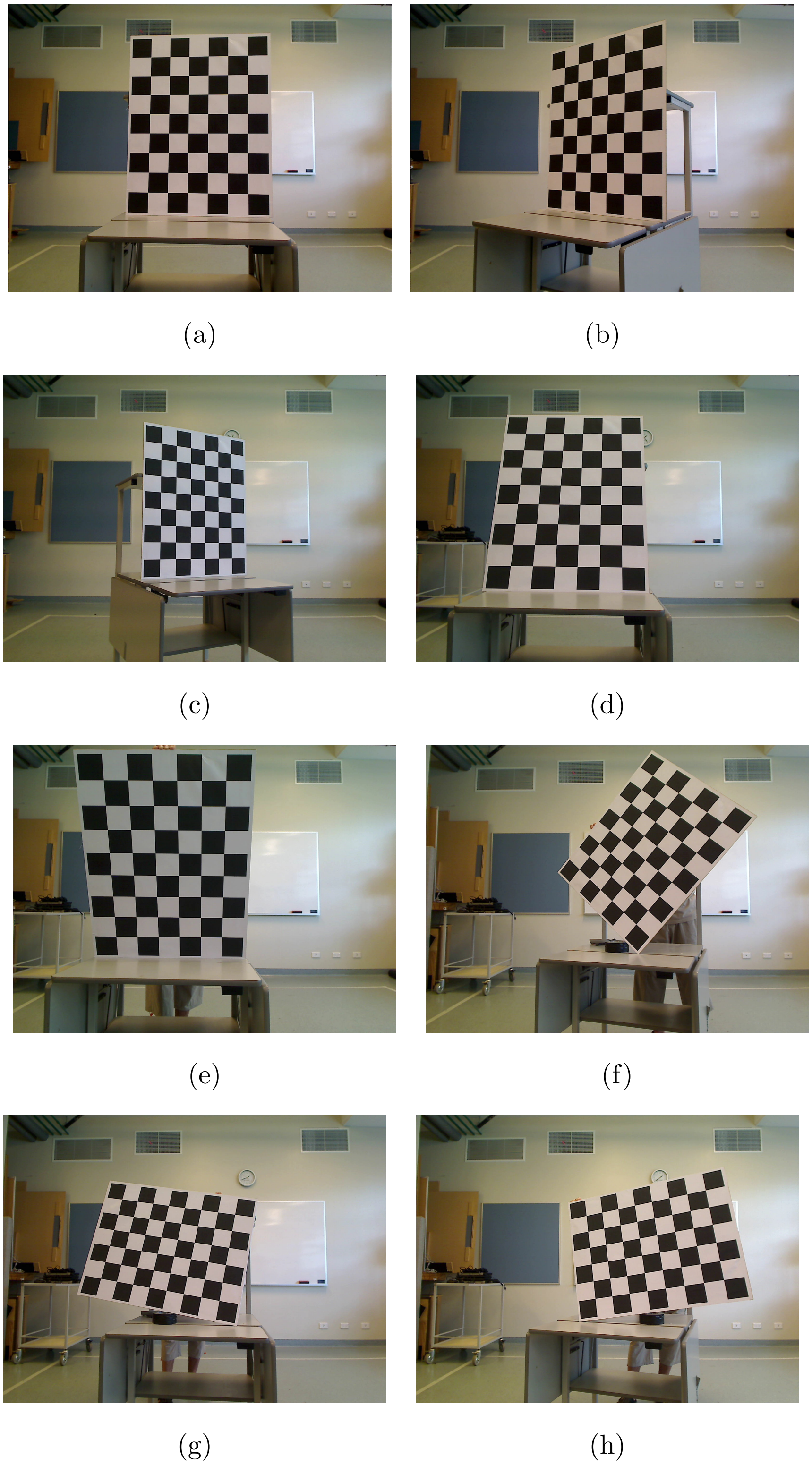
The extrinsic parameters are:

- Rotations R_c: A set of n (3×3) rotation matrices R_{c1}, R_{c2}, ..., R_{cn} (n=number of images used in the calibration).
- Translations T_c: A set of n (3×1) vectors T_{c1}, T_{c2}, ..., T_{cn} (n=number of images used in the calibration).

The camera calibration process is achieved by taking a diverse set of sharp images for the calibration target. The chessboard pattern is recommended as a calibration target as it is easy to detect its corners and it produces more accurate results. In this study, the Matlab Camera Calibration Toolbox developed by Bouguet (2009) was used to perform the camera calibration process. Camera calibration was performed using 50 images with 640 × 480 pixels image resolution taken for the chessboard. The chessboard was viewed from various distances, locations, and orientations in the images as shown in Figure 3.5. The whole calibration target needs to be visible in each image. The size of the squares in the chessboard must be measured correctly to accurately extract the corners of the chessboard squares from the images using the Matlab Camera Calibration Toolbox.

The results of the camera calibration were saved in a Matlab file to be used later for camera-laser calibration and projection of the points from the 3D world reference frame to the image plane. The intrinsic parameters resulted from the camera calibration were as follows:

- f_c = [532.66196 ; 533.22203]
- c_c = [310.06442 ; 234.86502]
- α_c = 0.00000;
- k_c = [0.06036 ; -0.17309 ; -0.00015 ; -0.00122 ; 0.00000]



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Figure 3.5: The chessboard calibration target viewed in images for camera calibration process.

The uncertainty in the intrinsic parameters were as follows:

- f_c_error = [1.32793 ; 1.31631]
- c_c_error = [0.99522 ; 0.82969]
- α_c_error = 0.00000
- k_c_error = [0.00598 ; 0.03041 ; 0.00053 ; 0.00065 ; 0.00000];

3.2.2.3 Camera-laser calibration

Mobile robots are typically equipped with multi sensors, which require a robust calibration process to represent the sensed data in a common coordinate system. In many mobile robot applications, the data from the camera and laser sensors are fused for better and more accurate detection and perception systems. It is critical to estimate the precise homogeneous transformation between the coordinate systems of the camera and the laser scanner to fuse the data acquired from these two sensors. Therefore, camera-laser calibration was implemented to determine the position and orientation of the camera relative to the laser scanner and to obtain a rigid transformation between the camera and the laser scanner under the assumption of known intrinsic parameters of the laser scanner (Meng et al., 2010; Kassir and Peynot, 2010).

The camera-laser calibration was performed in this study using the Automatic Camera-Laser Calibration Matlab Toolbox developed by Kassir and Peynot (2010). The calibration method relies on the chessboard calibration target to act as a common dataset between the laser scanner and the camera. This toolbox assumes that camera calibration has already been achieved. The output of the camera calibration was a requirement in this step since it provides the orientation and position of the calibration plane. This toolbox automatically extracted the points in the laser scans corresponding to the chessboard. The camera-laser calibration utilised the points originating from chessboard which were viewed in the laser scans to determine the camera to laser rigid transformation (Kassir and Peynot, 2010).

A single 2D laser scan provides little information about the location of the chessboard line. Therefore, chessboard extraction was achieved by using the integration of information from the entire dataset to estimate the length of the chessboard in the laser scans. The flowchart of the chessboard extraction algorithm is shown in Figure 3.6.

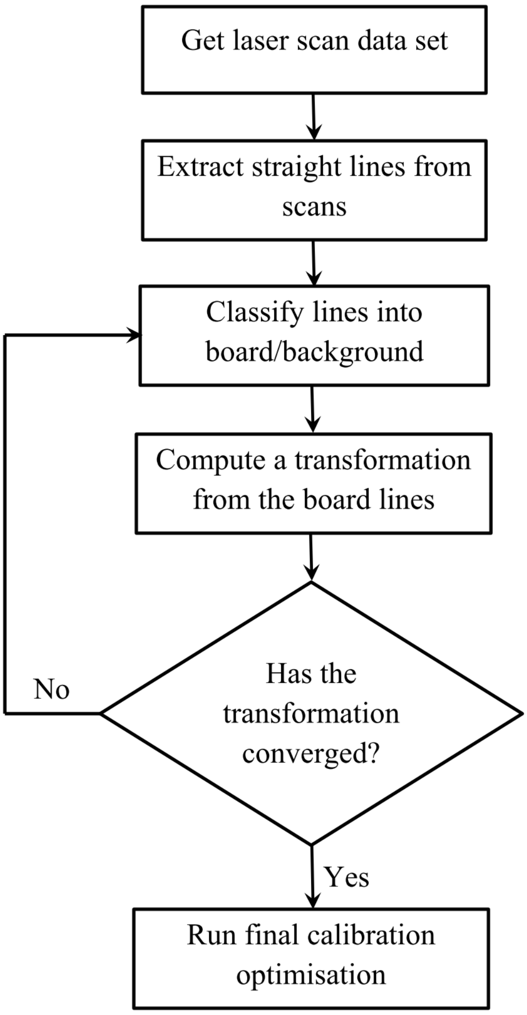

Get laser scan data set
Extract straight lines from scans
Classify lines into board/background
Compute a transformation from the board lines
Has the transformation converged?
No
Yes
Run final calibration optimisation

Figure 3.6: Automatic chessboard extraction from laser scans (Kassir and Peynot, 2010).

The algorithm begins with straight line extraction. These extracted straight lines are then classified into chessboard or background lines. From the classified lines, an estimate of the camera-laser rigid transformation was obtained. This transformation was then further used to aid in the classification. Iteratively, the transformation was refined until the same lines were reselected indicating convergence of the classification. At this point, the final calculation of the camera-laser transformation was performed following Kassir and Peynot (2010). The Automatic Camera-Laser Calibration Matlab Toolbox provided calibration results with uncertainty values which represented the estimated standard deviation of the errors. The results of this toolbox were as follows:

- Translation offset $\Delta = [\delta_x, \delta_y, \delta_z]$, which represents the translation in x, y, and z axes respectively.
- Euler angles $R = [\psi_x,\ \psi_y,\ \psi_z]$, which represents the rotation angles about the x, y, and z axes respectively.

In this study, a set of 50 synchronised pairs of images and laser scans was taken for the chessboard from different poses. The images used were the same as those used for camera calibration. The results from the camera calibration were used as input to the toolbox. The secure mounting of the camera and laser scanner ensured that their relative positions were maintained during and after the camera-laser calibration. The results obtained from the camera-laser calibration were as follows:

- $\Delta \pm error$ (in m) $= [-0.00487; 0.191; -0.0483] \pm [0.0204; 0.0688; 0.00967]$
- $R \pm error$ (in degree) $= [10.3; 3.19; -179] \pm [2.87; 0.379; 0.46]$

3.2.3 Odometer

Odometer is an instrument that indicates distance travelled by a vehicle or a mobile robot. Odometers are usually rotary encoders attached to the driving wheels to measure the vehicle motion in straight lines and turns. They are widely used to measure the position, orientation and velocity of agricultural robots. In this study, two odometers were attached to the front wheels and used to measure the linear velocity of the mobile robot.

External factors expected to influence the measurements obtained from the odometers were as follows:

- Wheel slippage which produces significant errors in the odometer measurements that are accumulated over time in real-world applications.
- The nature of the agricultural environments such as rough soil surface and uneven terrain might produce large errors in the mobile robot position measurements.

The CoroWare Explorer platform used in this study is driven by 4 DC motors attached to each wheel. The DC motors of the front wheels (GM9236S025 DC Servo Gearmotors from Pittman Express Inc.) have a built-in encoder with a resolution of 500 clocks per revolution (CPR) and a gearbox with a reduction ratio of 65.5. A bi-directional motor controller is used to control the DC motors for the front and rear wheels for each side together. Further information regarding the DC motors, encoders and motor controller can be found in Appendix A.

3.2.3.1 Odometer determination of travelled distance and position of the mobile robot

The odometer system is based on the assumption that the rotation of the wheel provides a corresponding linear motion. In principle, odometers can be used to measure the travelled distance and position of the mobile robot relative to its start position. This is achieved using known constants such as the circumference of the wheels and distance between the left and right wheels to measure the x and y positions as the robot moves. The Coroware Explorer wheel diameter was 203 mm, whilst the distance between the left and right wheels was 489 mm. The distance travelled by the wheels of each side was calculated using the following formula:

$$D = \frac{\Delta_M \times C}{G \times n} \tag{3.1}$$

where:

D is the distance travelled by each side.

Δ_M is the change in the odometer count between the current and previous readings

C is the circumference of the wheel.

G is the gear ratio.

n is the number of odometer ticks per wheel revolution.

The forward distance D_f travelled by the mobile robot was determined using the following equation:

$$D_f = \frac{D_r + D_l}{2} \tag{3.2}$$

where D_r and D_l are the distances travelled by the right and left side respectively and are determined using Equation 3.1.

Figure 3.7 illustrates the robot movement from previous point A to a new point B. The change in x and y positions from point A to point B were calculated from Equation 3.3 and Equation 3.4 respectively.

$$\Delta x = D_f \cos(\theta) \tag{3.3}$$

$$\Delta y = D_f \sin(\theta) \tag{3.4}$$

where θ is the heading angle of the mobile robot. These position changes were added to the previous positions x_A and y_A to determine the new positions x_B and y_B as follows:

$$x_B = x_A + \Delta x \tag{3.5}$$

$$y_B = y_A + \Delta y \tag{3.6}$$

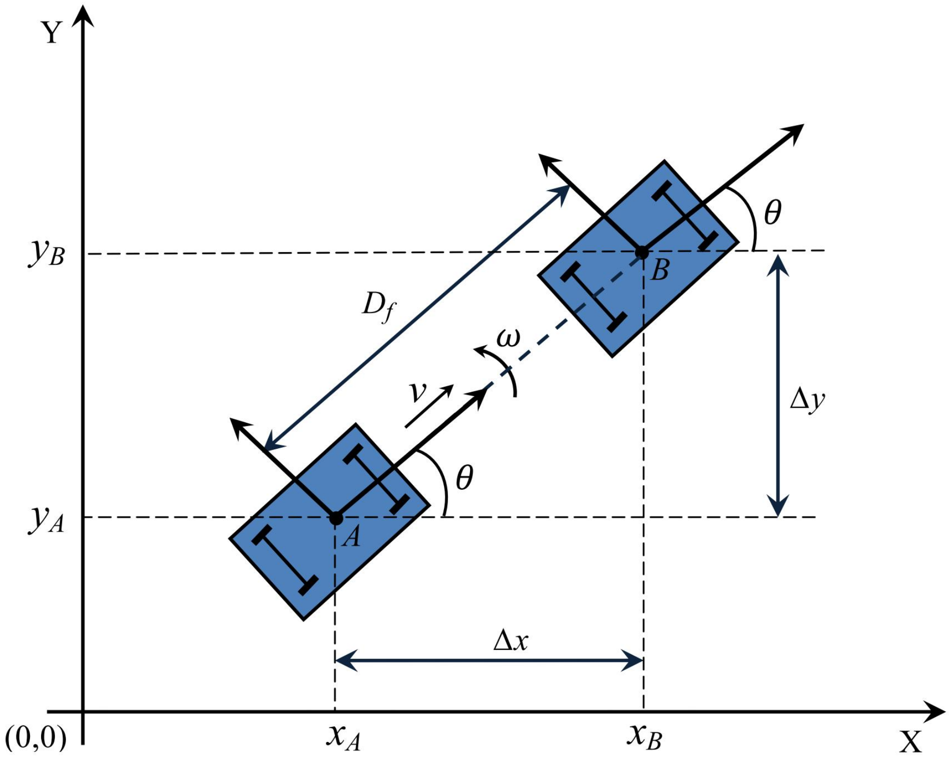


Figure 3.7: Determination of the new position of the mobile robot.

Different tests were conducted to evaluate the efficacy of the travelled distance data obtained from the odometers. The mobile robot was moved forward from the same starting position in straight lines for different velocities (0.3 m/s, 0.5 m/s and 0.8 m/s) on the grass to test the data acquired from the odometer. This was achieved by setting the same velocity for the left and right wheels. These tests were repeated three times for each velocity. In each test, the actual distance travelled by the mobile robot (D_m) was measured manually and compared with the travelled distance measured by the odometer (D_f) to determine the error in the travelled distance data. The percentage error in the forward travelled distance ($E_d\%$) was calculated using the following equation:

$$E_d\% = \frac{|D_m - D_f|}{D_m} \times 100 \tag{3.7}$$

Table 3.1 shows the percentage error in the forward travelled distance for a straight line movement for different velocities. From Table 3.1, it can be observed that the increase in mobile robot velocity caused greater wheel slippage and increased the error in the odometer measurements. The average of the percentage error in the travelled distance on the grass for all tests was 10.32%. This error is expected to be higher when the mobile robot executes different turns (semi-circle turn and right-angle turn) due to the high wheel slippage caused by the mobile robot turns. Hence, the mobile robot cannot rely only on the odometer measurements to determine its position.

Table 3.1: The results of the percentage error of the forward travelled distance in the mobile robot on the grass for different velocities.

Linear velocity (m/s)	Replicate	D_m (m)	D_f (m)	Percentage distance error (%)
0.3	I	10	10.853	8.53
	II	10	10.934	9.34
	III	10	10.889	8.89
0.5	I	10	10.934	9.34
	II	10	10.97	9.70
	III	10	11.113	11.13
0.8	I	15	16.878	12.52
	II	15	16.761	11.74
	III	15	16.748	11.65
			Average	10.32

3.2.3.2 Linear velocity determination from the odometer

The odometer was used in this study to determine the linear velocity v of the mobile robot. The linear velocity is determined as follows:

$$v = \frac{D_f}{\Delta t} \tag{3.8}$$

where Δt is the time difference between time steps t and $(t-1)$.

Figure 3.8 shows the linear velocity measured by the odometer for the straight line movement.

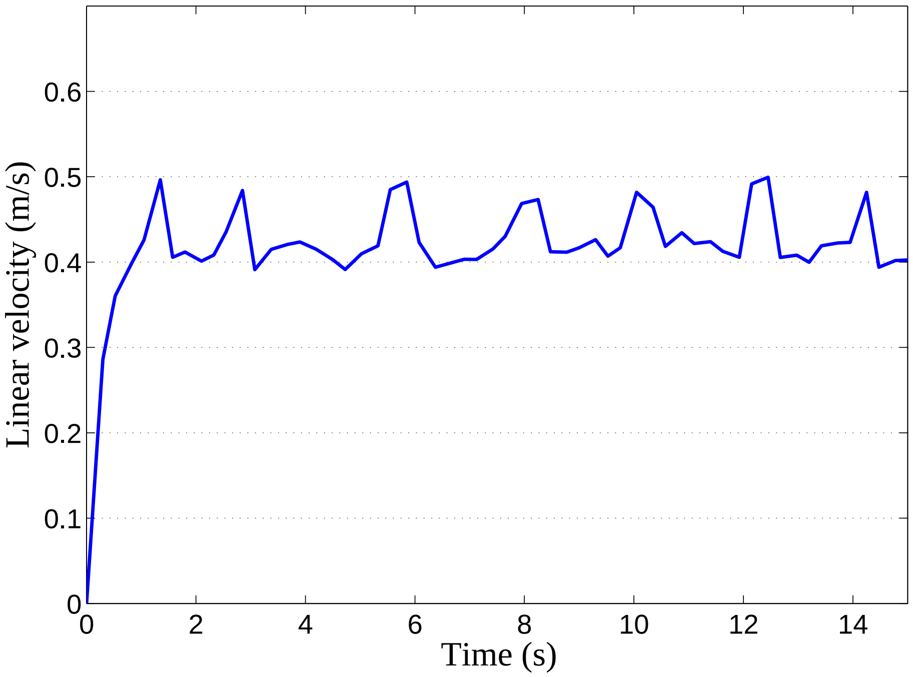


Figure 3.8: Linear velocity of the mobile robot measured using the odometer.

3.2.4 Inertial Measurement Unit (IMU)

The IMU comprises an orthogonal configuration of three accelerometers, three gyroscopes and three magnetometers permitting direct measurements of the acceleration, the angular velocity and the strength of the magnetic field respectively in x, y, and z axes of the mobile robot. These sensors are rigidly mounted to a common base to maintain the same relative orientation. The IMU sensor used by the CoroWare Explorer platform is 1044_0 - PhidgetSpatial Precision 3/3/3, which combines the functionality of a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer all in one convenient package. Further information regarding the IMU can be found in Appendix A.

If the starting position and orientation of the mobile robot are known, the IMU measurements could be used to estimate the pose (position and orientation) of the mobile robot by using numeric integration. However, the IMU has the disadvantage of accumulated errors when used for position measurements. Since the IMU measurements are integrated to obtain position values, the measurement errors are accumulated and the accumulated errors can produce a drift or inaccurate estimate of the actual position. The IMU measurements might be affected by some external parameters in the orchard. The ground roughness introduces tilt in the IMU affecting its measurement accuracy in yaw as it is assumed that the angular velocity from the IMU in pitch and roll are equal to zero.

3.2.4.1 Accelerometer data evaluation

The accelerometer data was tested thoroughly to figure out the consistency of its readings. Figure 3.9 illustrates the inherent bias of the accelerometer data acquired when the robot was stationary. A_x, A_y and A_z represent the acceleration in the x, y and z axes respectively. The reading of the acceleration A_z in Figure 3.9 is approximately ($9.8\ m/s^2$) which is due to Earth's gravity. The accelerometer data was tested for different straight forward and turns movement of the mobile robot. For all these tests the accelerometer data was prone to noise. For this, the accelerometer was not used in this study. Figure 3.10 shows the inherent bias and noise of the accelerometer for straight line movement of the mobile robot on the grass.

3.2.4.2 Gyroscope data evaluation

In this study, the gyroscope was used to measure the angular velocity of the mobile robot. It was observed that the gyroscope readings produced bias values when the mobile robot was stationary, which require compensation. An experimental test was carried out to measure the bias value of the angular velocity obtained from the gyroscope. This was achieved by keeping the mobile robot stationary and determining the mean value of the angular velocity for specific data samples. This mean value was used as a constant value to be subtracted from the measured angular velocity of the mobile robot.

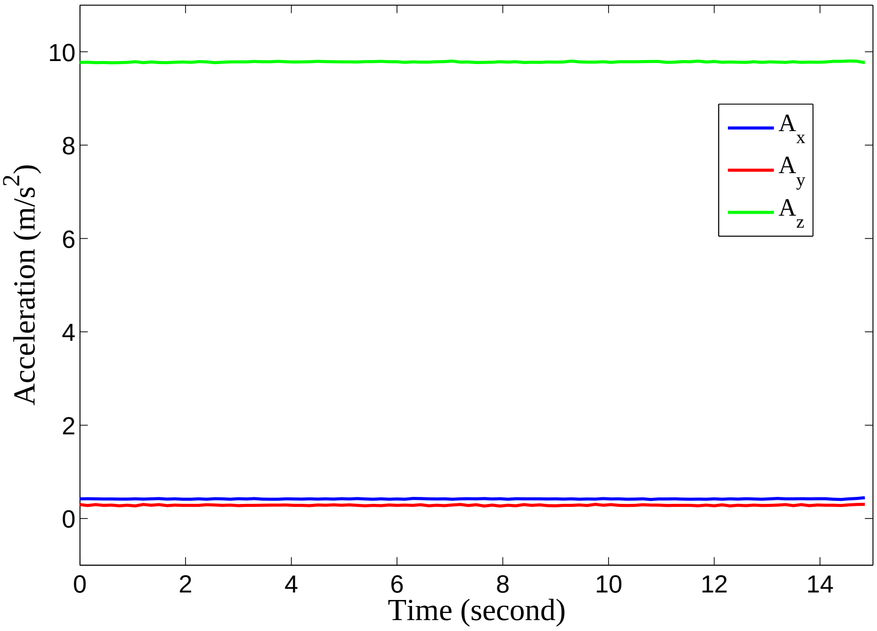


Figure 3.9: Acceleration data from the accelerometer for stationary test.

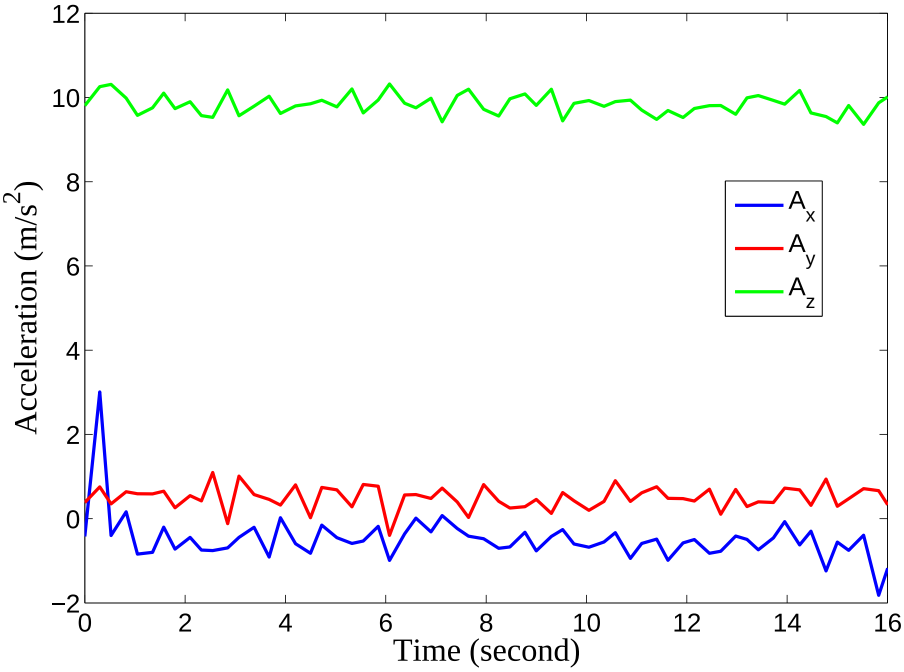


Figure 3.10: Acceleration data from the accelerometer for straight line movement on the grass.

The angular velocity from the gyroscope was measured and tested. Figure 3.11 shows the angular velocity obtained from the gyroscope when the robot is moving in a straight line. The deviations of the movement of the mobile robot from the theoretical straight line were estimated to be not greater than (± 20) mm.

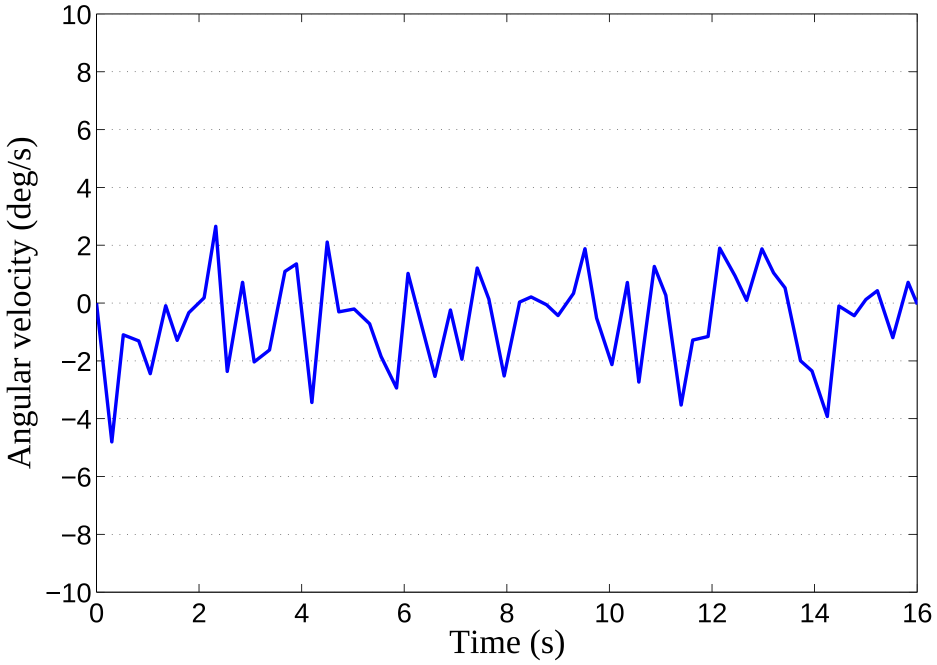


Figure 3.11: The angular velocity from the gyroscope for straight line movement on the grass.

3.2.5 Real Time Kinematic Global Positioning System (RTK-GPS)

The RTK-GPS is widely used to determine the position of agriculture vehicles performing different tasks. It provides x and y position data in UTM coordinates from the latitude and longitude positioning data. The RTK-GPS uses a base station for position correction. However, the RTK-GPS is not expected to provide perfect data all the time as data dropout caused by tree canopies might block the signal from the satellites. In addition the signal from the base station could also be lost and thereby the RTK-GPS would provide less accurate positioning data (Christiansen, 2011).

The RTK-GPS used in this study is Trimble R10 GNSS system. It is used to measure the ground truth to evaluate the estimated position from the developed localisation algorithm using EKF. The RTK-GPS antenna is placed on the top of the mobile robot as shown in Figure 3.1. The distance from the RTK-GPS antenna to the ground is about (780 mm). The offset between the RTK-GPS antenna position and the laser scanner position on the mobile robot platform in x and y coordinates is determined when evaluating the position measurements from the EKF localisation algorithm. The accuracy of the position data from this RTK-GPS is (8 mm-15 mm). Further information regarding the RTK-GPS can be found in Appendix A.

3.3 Simulated environment

A simulated environment was constructed for preliminary data collection and evaluation of the developed tree trunk detection, orchard mapping and mobile robot localisation algorithms. In this study, the simulated environment is not computer simulation but it is real environment with simulated objects. The tree trunks were the only simulated objects, whilst the rest are real (e.g. uneven terrain, illumination conditions, trunk inclinations, mobile robot with on-board sensors, etc). This simulated environment can be considered as a small scale model of the ideal orchard and consists of simulated tree trunks constructed from mailing tubes (vertical cylinders) of 900 mm height and 90 mm diameter. The simulated tree trunks were placed in rows with approximately equal distances between the simulated trees in the row, and approximately equal distances between the rows as would be expected in an orchard as shown in Figure 3.12. In this simulated environment, it was assumed that there was no tall grass and 500 mm of each simulated tree trunk was exposed above ground level.

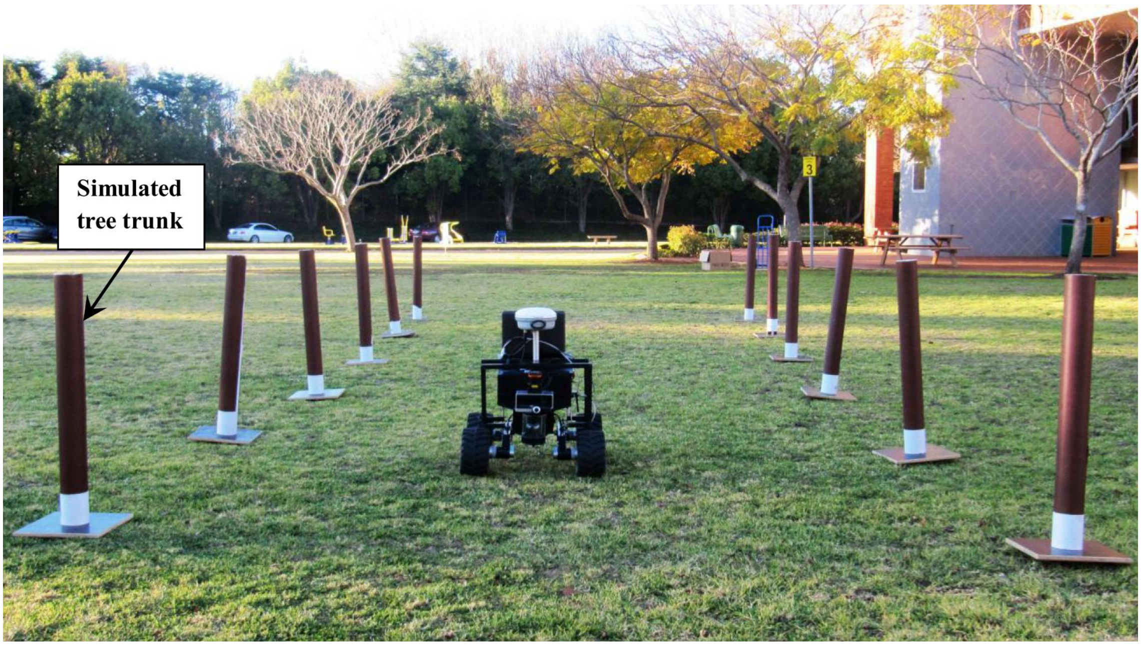
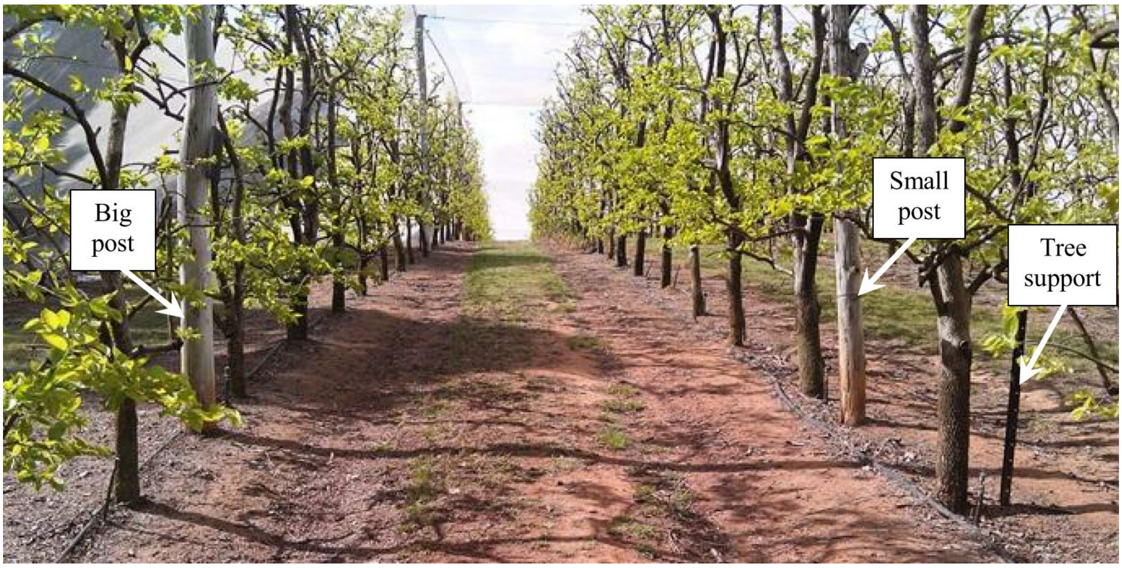


Figure 3.12: The simulated environment with simulated tree trunks.

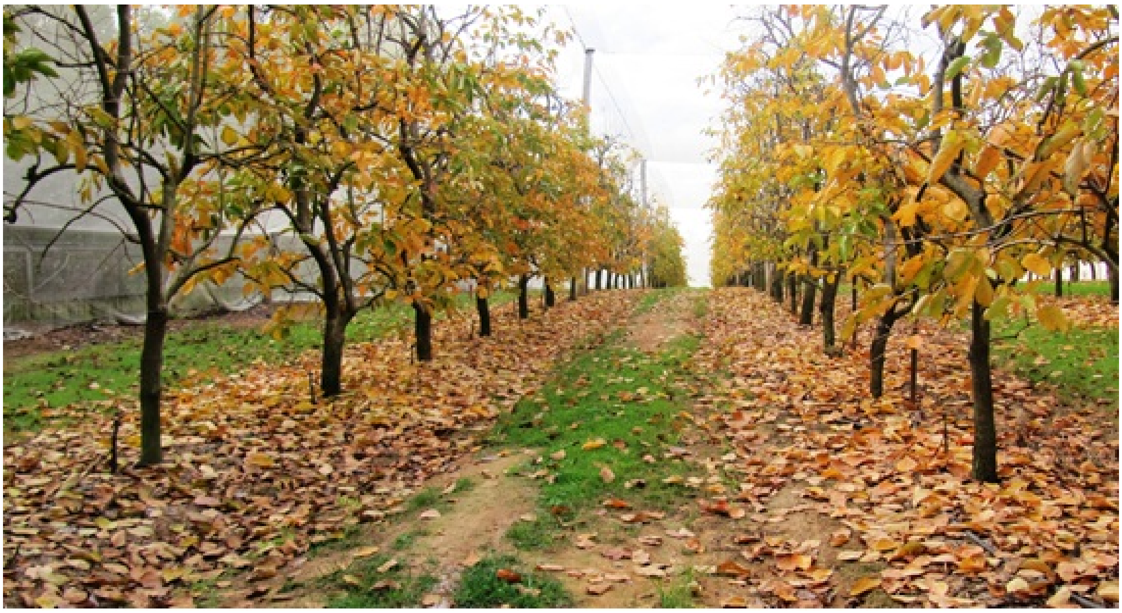
3.4 Orchard environment layout

In this study, a persimmon orchard, shown in Figure 3.13, was chosen for the experimental tests. This orchard is located in Gatton, Queensland, Australia. The distance between the tree rows is approximately 3.5 m, while the trees are planted at intervals of approximately 2.5 m in each row. In this orchard, 500 – 700 mm of each tree trunk is exposed above ground level and, for most trees, the branches start above this level. The age of the trees was 12 years, whilst the height of the trees was approximately 2.5 – 3 m. It was observed that this orchard had 'regular tree trunks' with trunk width ranging from 88 to 145 mm and 'thin tree trunks' with trunk width ranging from 54 to 67 mm. Figure 3.14 shows a sample of a thin tree trunk and a regular tree trunk.

In this orchard, there were also some posts and tree supports placed in different positions between the trees. These posts and tree supports have a different colour than the tree trunks. The width of some posts in this orchard was within the range of the width of the regular tree trunks. These posts were considered in this study as ‘small posts’, whilst the width of other posts was greater than the range of the width of the regular tree trunks (i.e. greater than 145 mm) and these posts were considered as ‘big posts’. The width of the ‘tree supports’ is less than the range of the width of the regular tree trunks. Figure 3.13a shows the non-tree object samples in the tree rows. The mobile robot and the orchard were used in developing the tree trunk detection, orchard mapping and localisation algorithms.



(a) Late September 2014.



(b) May 2014.

Figure 3.13: Tree rows in the orchard and typical non-tree objects

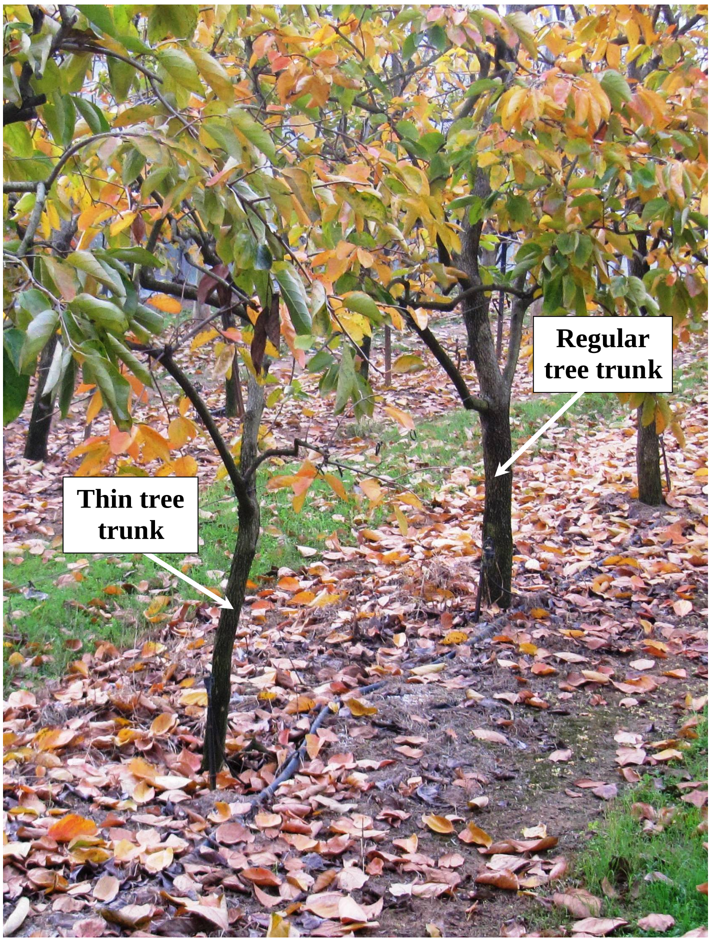


Figure 3.14: Thin and regular tree trunks (May 2014).

Chapter 4

Tree trunk detection

This chapter presents the tree trunk detection algorithm using camera and 2D laser scanner data fusion. A preliminary tree trunk detection algorithm, denoted here as ‘Detection Algorithm A’ was implemented to detect simulated tree trunks of similar width as well as non-tree objects. This algorithm was enhanced to become ‘Detection Algorithm B’ which is capable of detecting tree trunks with different widths as well as non-tree objects. The three main detection stages of each algorithm are set out in Sections 4.2.1, 4.2.2 and 4.2.3, whilst Detection Algorithm A and Detection algorithm B are presented in Sections 4.3.1 and 4.3.2 respectively.

Preliminary tests and results of Detection Algorithm A in the simulated environment are presented in Section 4.4, whilst the experimental tests and results of Detection Algorithm B in the real orchard are presented in Section 4.5.

4.1 Tree trunk feature extraction

Feature extraction is the process of defining a set of features which will most efficiently or meaningfully represent the information that is important for analysis and classification. The detection of the agricultural features is directly related to the purpose of the mobile robot design and the sensors incorporated on it. The on-board sensors of the agricultural mobile robot acquire the environmental information and the processing of this information allows the extraction of the required biological features. The detection process raises the following questions (Auat Cheein and Carelli, 2013):

- What is the biological feature of interest?
- How is such a feature extracted/detected?

In this study, trees already existing in the orchard were selected as the natural landmarks for mobile robot localisation as they were planted in straight and parallel rows. In the selected orchard, described in Sections 3.4, it was observed that the tree trunks could be distinguished from the leaves and other objects. Therefore, the tree trunk was selected as the part of the tree suited to detection.

There are differences between the data acquired from the 2D laser scanner and the camera images. The 2D laser scanner generates a single horizontal scan of the environment, whereas the camera provides an instantaneous image of the local environment with precise depth information. A laser scanner provides range and bearing data, while the camera primarily provides intensity and colour information. There are some common features in both types of data. For example, many corners and edges correspond to a sudden change in the range along the laser scan data and a sudden variation in image intensity (Peynot and Kassir, 2010) which enable the fusion of the data from these sensors.

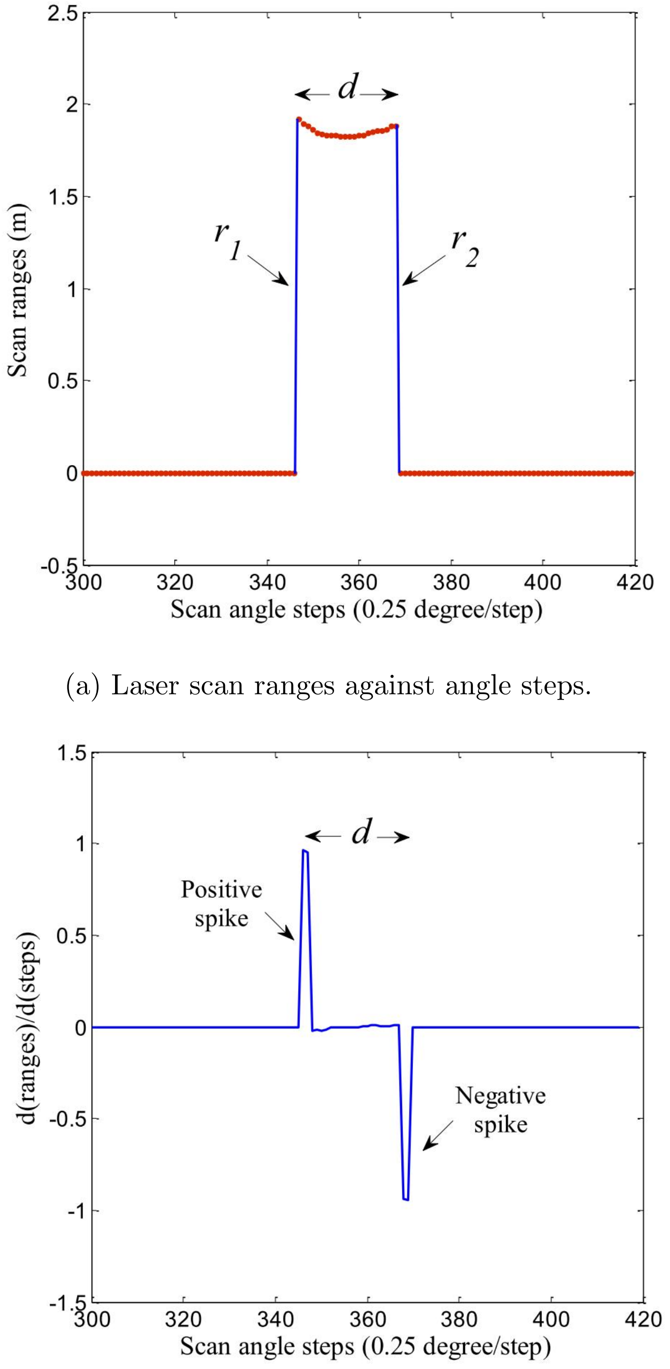
4.2 Tree trunk detection algorithm

This work presents a new tree trunk detection algorithm using a low cost colour camera and 2D laser scanning technologies as a component of fully automated operation for such small mobile robot. The three main detection stages of the developed algorithm are:

- Laser-based tree trunk detection.
- Projection of laser points on camera image plane.
- Vision-based tree trunk detection.

4.2.1 Laser-based tree trunk detection

The laser-based tree trunk detection stage starts by reading the laser scan data and detecting the objects from the laser scan to determine their width (d). The objects found in the scan represent the trees and non-tree objects. The width d is determined by detecting the start and the end edges of each object at ranges r_1 and r_2 respectively. These edges are detected from the derivative data $d(ranges)/d(steps)$ since the ranges r_1 and r_2 generate positive and negative spikes respectively in the derivative data. Figure 4.1a shows the scan ranges plotted against the scan angle steps for a detected tree trunk with width d. The laser scanner returns a value of zero when no object is detected. Figure 4.1b shows the $d(ranges)/d(steps)$ data plotted against the scan angle steps. The algorithm detects each object by searching for two consecutive positive and negative spikes in the derivative data. The amplitude of the spike depends on the distance between the laser scanner and the object. The tree trunk width d is represented by the number of angle steps between the positive and negative spikes in the derivative data.



(a) Laser scan ranges against angle steps.

(b) Derivative data against angle steps.

Figure 4.1: Tree trunk representation in the laser scan data and the derivative data.

The width d of the object is calculated from the polar representation of a single object as shown in Figure 4.2 using the following equation:

$$d = \sqrt{r_1^2 + r_2^2 - 2r_1r_2\cos(\Delta\phi)} \tag{4.1}$$

where ϕ_1 and ϕ_2 are the object angles at r_1 and r_2 respectively, $\Delta\phi$ represents the difference between ϕ_1 and ϕ_2.

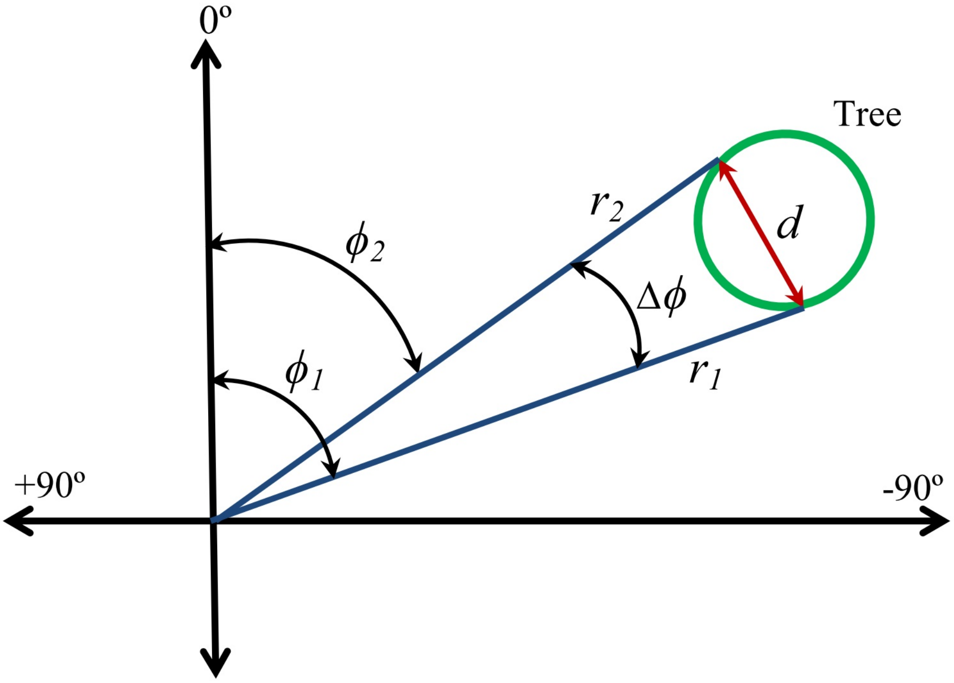


Figure 4.2: Object width determination from laser scan data.

Prior to implementing the algorithm, the width of multiple tree trunks selected randomly from the orchard are measured from the laser scan data using Equation 4.1 to set out the initial tree trunk width distribution. Subsequently, the mean (μ_d) and the standard deviation (σ_d) of the tree trunk width distribution are determined. These values would vary from one orchard to another depending on the type of the trees and need to be determined before starting the algorithm. The calculated μ_d and σ_d are used to determine the probability density function of the normal distribution $pdf(\mu_d)$ which represents the peak value of the probability around the mean. To implement the tree trunk detection algorithm, the algorithm determines the width of each object in each laser scan and calculates its probability density function $pdf(d)$ which represents the probability of the object width being a tree trunk width. The $pdf(d)$ is compared with $pdf(\mu_d)$ to determine the rate of confidence of the object from the laser scanner (ROC_L). The $pdf(d)$ and ROC_L are calculated using Equation 4.2 and Equation 4.3 respectively:

$$pdf(d) = \frac{1}{\sigma_d\sqrt{2\pi}} e^{-\frac{(d-\mu_d)^2}{2\sigma_d^2}} \tag{4.2}$$

$$ROC_L = \frac{pdf(d)}{pdf(\mu_d)} \tag{4.3}$$

Figure 4.3 illustrates the procedure of the laser-based tree trunk detection stage.

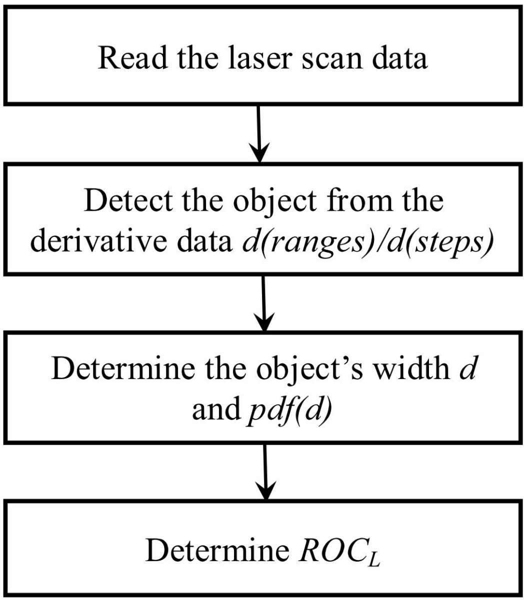

Read the laser scan data
Detect the object from the derivative data *d(ranges)/d(steps)*
Determine the object's width *d* and *pdf(d)*
Determine ROC_L

Figure 4.3: Laser-based tree trunk detection stage.

4.2.2 Projection of laser points on camera image plane

The projection of any laser point on the image plane of the camera is achieved in two steps:

- Transformation of the Cartesian coordinates of a point in a 3D space from laser frame to camera frame.
- Projection from camera frame to the image plane.

Laser-camera transformation requires *a priori* camera calibration to determine the intrinsic parameters of the camera, which include the focal length f_c, the principal point coordinates c_c, skew coefficient α_c, and the distortion coefficients k_c. Camera calibration was achieved using the Matlab Camera Calibration Toolbox developed by Bouguet (2009). Extrinsic camera-laser calibration is also required to estimate the parameters of the transformation between the laser frame and the camera frame. The camera-laser calibration was performed using the Automatic Matlab Camera-Laser Calibration Toolbox developed by Kassir and Peynot (2010). To achieve the laser-camera transformation, the 3D coordinates of any point with respect to the laser frame is first calculated from the range and bearing angle of the point. Then the transformation from laser frame $P_l = [X_l; Y_l; Z_l]$ to the camera frame $P_c = [X_c; Y_c; Z_c]$ is performed using Equation 4.4:

$$P_c = \frac{P_l}{\Phi} + \Delta \tag{4.4}$$

where $\Delta = [\delta_x, \delta_y, \delta_z]$ is the translation offset vector and Φ is the rotation matrix between the laser frame and camera frame about x, y and z axes and is defined by a set of three Euler angles $R = [\psi_x,\ \psi_y,\ \psi_z]$ which were obtained from camera-laser calibration. The rotation matrix is determined using the following equation (Peynot and Kassir, 2010):

$$\Phi = \begin{bmatrix} c_y c_z & c_x s_z + s_x c_z s_y & s_x s_z - c_x s_y c_z \\ -c_y s_z & c_x c_z - s_x s_y s_z & s_x c_z + c_x s_y s_z \\ s_y & -s_x c_y & c_x c_y \end{bmatrix} \tag{4.5}$$

where s_i and c_i stand for $sin(\psi_i)$ and $cos(\psi_i)$ respectively. After that, the point in the camera frame P_c is projected to the image plane according to the method implemented in the Matlab Camera Calibration Toolbox by Bouguet (2009) as described in Equations 4.6 to 4.7 below.

To project a point in space P of coordinate vector $[X_c; Y_c; Z_c]$ in the camera reference frame onto the image plane, the normalised pinhole image projection is given by:

$$x_n = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{4.6}$$

Let $r^2 = x^2 + y^2$. After including lens distortion k_c, the new normalised point coordinates x_d is defined as follows:

$$x_d = \begin{bmatrix} x_d(1) \\ x_d(2) \end{bmatrix} = (1 + k_c(1)r^2 + k_c(2)r^4 + k_c(5)r^6)x_n + dx \tag{4.7}$$

where dx is the tangential distortion vector which is due to imperfect centering of the lens components and other manufacturing defects in a compound lens:

$$dx = \begin{bmatrix} 2k_c(3)xy + k_c(4)(r^2 + 2x^2) \\ k_c(3)(r^2 + 2y^2) + 2k_c(4)xy \end{bmatrix} \tag{4.8}$$

Once the distortion is applied and using the intrinsic parameter of the camera (f_c, c_c, α_c, k_c), the final pixel coordinates $[x_p; y_p]$ of the projection of point P on the image plane is determined as follows:

$$\begin{cases} x_p = f_c(1)(x_d(1) + \alpha_c * x_d(2)) + c_c(1) \\ y_p = f_c(2)x_d(2) + c_c(2) \end{cases} \tag{4.9}$$

Therefore, the pixel coordinate vector $[x_p; y_p]$ and the normalised (distorted) coordinate vector x_d are related to each other by:

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = K \begin{bmatrix} x_d(1) \\ x_d(2) \\ 1 \end{bmatrix} \tag{4.10}$$

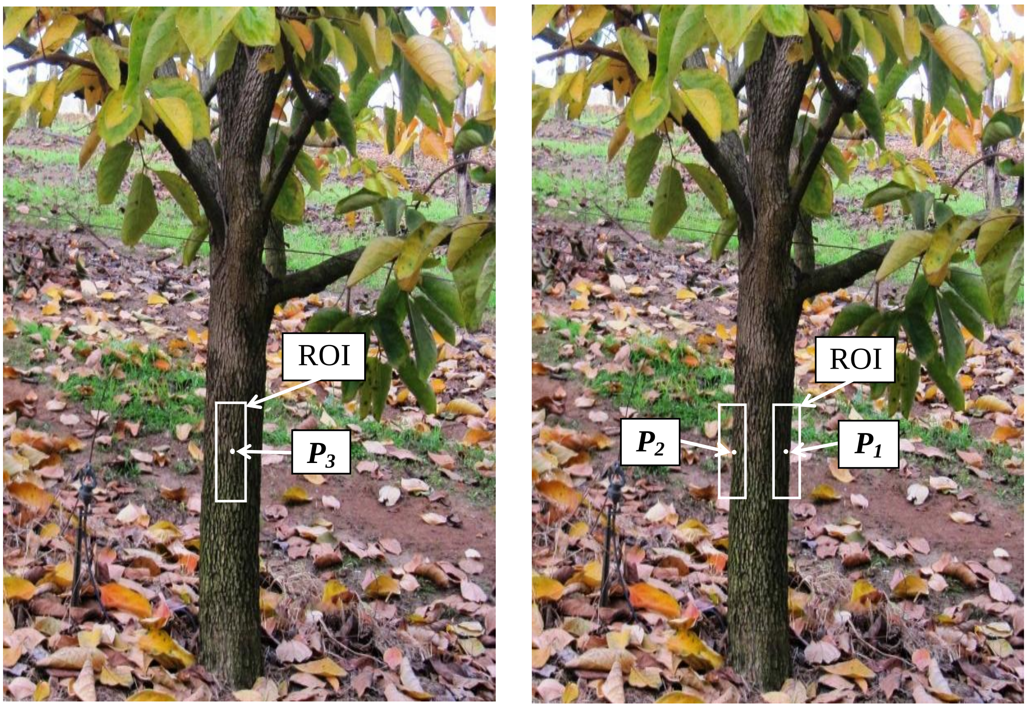
where K is the camera matrix and is defined as follows :

$$K = \begin{bmatrix} f_c(1) & \alpha_c * f_c(1) & c_c(1) \\ 0 & f_c(2) & c_c(2) \\ 0 & 0 & 1 \end{bmatrix} \tag{4.11}$$

The above procedure is used for camera and laser scanner data fusion. To perform the fusion, the ϕ_1, r_1 and ϕ_2, r_2 are converted from laser polar coordinates to laser Cartesian coordinates P_l then transformed to camera Cartesian coordinates P_c and projected into the image plane as P_1 and P_2 respectively to be used for tree trunk edge detection. The centre point P_3 of the object is determined for tree trunk colour detection as follows:

$$P_3 = P_2 + \frac{P_1 - P_2}{2} \tag{4.12}$$

A rectangular region of interest (ROI) window is selected around P_1, P_2 and P_3 as it is assumed that the required features are located in the ROI as shown in Figure 4.4. This will reduce the processing time and minimise the effect of unused information and noise in the other parts of the image. The size of the ROI is directly proportional to the object's width and inversely proportional to the range of the object determined by the laser scanner. The width of the ROI is approximately half the width of the tree trunk measured by the laser scanner. Figure 4.4a demonstrates the P_3 of the tree trunk and the ROI selected around this point, while Figure 4.4b shows the projected points P_1 and P_2 and the selected ROI around each tree trunk edge.



(a) ROI for colour detection. (b) ROI for edge detection.

Figure 4.4: Samples of the ROI for colour and edge detection respectively.

4.2.3 Vision-based tree trunk detection

The vision-based tree trunk detection stage consists of both the colour and edge detection methods, since colour only can not work well in different illumination conditions. Edges alone also can not provide enough information for use in tree trunk detection because there might be other vertical edges in the image such as posts and tree supports.

4.2.3.1 Tree trunk colour detection

The vision-based tree trunk detection algorithm first implements tree trunk colour detection for the selected ROI since the tree trunks were observed to have visually discernible colour from other scene elements (e.g. grass, sky, foliage). The HSV (Hue, Saturation, Value) colour space was chosen for colour detection where the Hue dimension represents the colour, the Saturation dimension represents the dominance of that colour and the Value dimension represents the brightness. This colour space is less affected by the illumination conditions compared to the RGB colour space. The ROI pixels are converted from RGB space to HSV space. The most dominant value of the Hue (H_d) in the ROI was used for colour detection because it provides the information about the colour, whilst Saturation and Value focus on illumination conditions.

Before starting the algorithm, a study was performed to determine the H_d of multiple tree trunks selected randomly in the orchard to set out the initial tree trunk H_d distribution under different illumination conditions. Subsequently, the mean μ_{H_d}, the standard deviation σ_{H_d} and the probability density function $pdf(\mu_{H_d})$ of the H_d distribution are calculated. The $pdf(\mu_{H_d})$ represents the peak value of the probability around the mean. To implement colour detection, the algorithm determines the H_d in the ROI selected around P_3 and calculates its probability density function $pdf(H_d)$ which represents the probability of the object H_d being a tree trunk H_d and is determined from Equation 4.13. The rate of confidence of the tree trunk colour (ROC_C) is then determined from Equation 4.14:

$$pdf(H_d) = \frac{1}{\sigma_{H_d}\sqrt{2\pi}} e^{-\frac{(H_d-\mu_{H_d})^2}{2\sigma^2_{H_d}}} \tag{4.13}$$

$$ROC_C = \frac{pdf(H_d)}{pdf(\mu_{H_d})} \tag{4.14}$$

Figure 4.5 illustrates the procedure for tree trunk colour detection from images.

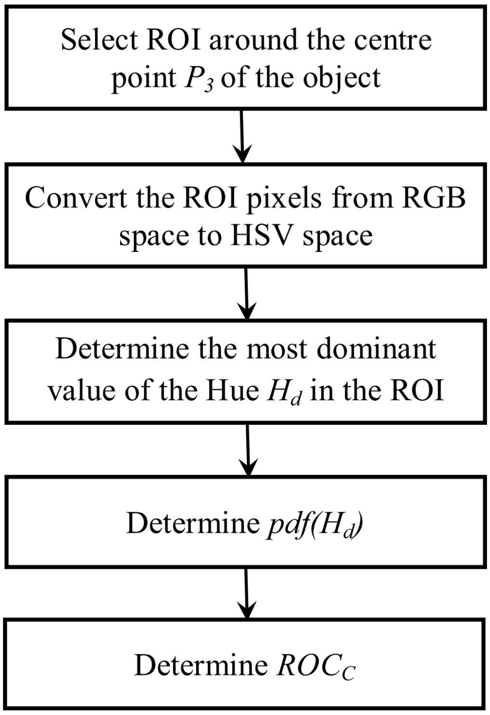

Select ROI around the centre point P_3 of the object
Convert the ROI pixels from RGB space to HSV space
Determine the most dominant value of the Hue H_d in the ROI
Determine $pdf(H_d)$
Determine ROC_C

Figure 4.5: Tree trunk colour detection.

4.2.3.2 Tree trunk edge detection

Tree trunk edges are the other features used (in this study) to detect the tree trunk from images. Tree trunk edge detection is implemented for each ROI window around P_1 and P_2. The ROIs constructed at these points are converted from RGB images to gray-scale images, where the gray-scale value is calculated as the weighted sum of the R, G, and B components. The gray-scale images are filtered to remove the noise prior to the edge detection using a Median filter because of its capability to simultaneously reduce noise and preserve edges. Edge detection is then implemented for each ROI window using the Canny method (Canny, 1986) yielding binary windows. The algorithm searches for a possible straight edge in each window using the least-squares linear regression method. This is achieved by fitting a linear model to the edge data in each ROI and calculating the measure of the goodness of the fit R^2 for each edge which has a value ranging from 0 to 1. The angle of each line (β) for each edge is determined to test if the two edges are parallel (E_p) as follows:

$$E_p = \begin{cases} 1 & \text{if} \quad 0 \leq |\beta_1 - \beta_2| \leq \Delta\beta_{max} \\ 0 & \text{if} \quad |\beta_1 - \beta_2| > \Delta\beta_{max} \end{cases} \tag{4.15}$$

where, β_1 and β_2 are the angles of the right and left edge lines respectively and $\Delta\beta_{max}$ is the maximum allowable angle difference for the two lines being parallel which was estimated from a set of tree trunk images. The rate of confidence from edge detection for each tree trunk (ROC_E) is determined from Equation 4.16.

$$ROC_E = w_1 R_1^2 + w_2 R_2^2 + w_3 E_p \tag{4.16}$$

where R_1^2 and R_2^2 are the measure of the goodness of the fit for the right and left edge lines respectively. The wights w_1, w_2 and w_3 are the weights for R_1^2, R_2^2 and E_p respectively and were determined empirically. Figure 4.6 explains the procedure for tree edge detection.

Figure 4.7 shows an example of the ROI images in RGB, gray-scale and Canny edge detection, whilst Figure 4.8 depicts the fitting of a linear model to the edge data in the same ROI of Figure 4.7 using least-squares linear regression method.

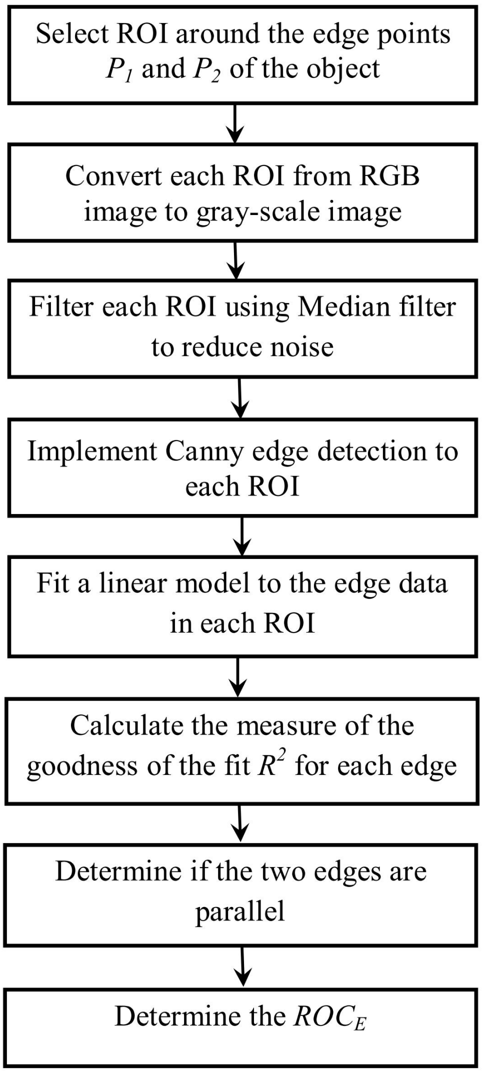
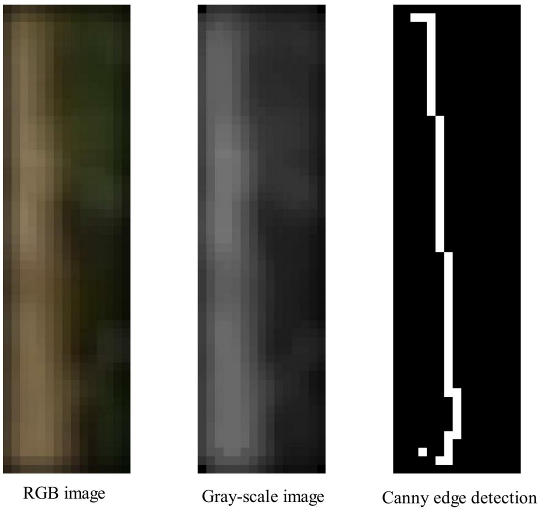

Select ROI around the edge points P_1 and P_2 of the object
Convert each ROI from RGB image to gray-scale image
Filter each ROI using Median filter to reduce noise
Implement Canny edge detection to each ROI
Fit a linear model to the edge data in each ROI
Calculate the measure of the goodness of the fit R^2 for each edge
Determine if the two edges are parallel
Determine the ROC_E

Figure 4.6: Tree trunk edges detection procedure.

4.3 Combination of laser and vision based tree trunk detection

In this study, a combination of laser and vision based tree trunk detection was achieved to improve the tree trunk detection method. Early work, Detection Algorithm A (Shalal et al., 2013), was implemented to detect simulated tree trunks and trees of similar trunk size using vision and laser scanner data fusion. This algorithm was tested in the persimmon orchard considered in this study. This algorithm faced


RGB image

Gray-scale image

Canny edge detection

Figure 4.7: The ROI images in RGB, gray-scale and Canny edge detection.

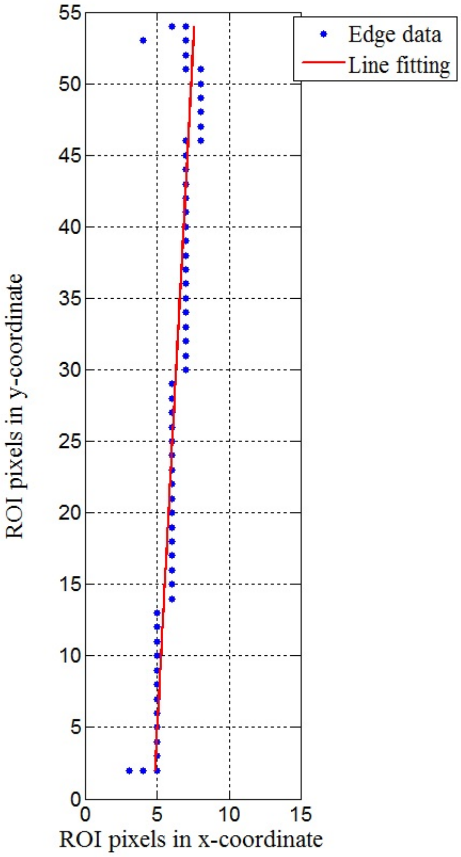


Figure 4.8: Least-squares linear regression method of tree trunk edge detection.

some difficulties, especially in detecting the thin tree trunks and discriminating between trees and different non-tree objects found in the tree rows. As a result, Detection Algorithm A was revised and modified to become Detection Algorithm B to overcome these limitations.

4.3.1 Tree trunk detection for trees of similar trunk size – Detection Algorithm A

Detection Algorithm A developed in this study calculates the 'rate of confidence' for each object in the scene and determines whether it is a tree trunk or non-tree object. The rate of confidence is a value between 0 (definite non-tree) and 1 (definite tree) that is assigned to each tree trunk and non-tree object. In the first stage of the Detection Algorithm A, laser scanner data was used to distinguish between the candidate tree trunk and the non-tree objects based on width only. The candidate tree trunk was further tested by the vision to decide if it was a tree trunk or a non-tree object as illustrated in Figure 4.10a.

The ROC_L for each object was determined and used to decide whether the object is a candidate tree trunk or non-tree object depending on its value. A threshold value TH_L for the ROC_L that distinguishes between candidate tree trunks and non-tree objects was determined empirically. The ROC_L of each object was compared with the preset TH_L. If ROC_L of the object is greater or equal to TH_L, then the algorithm considers this object as a candidate tree trunk and projects its edge points and centre point onto the image plane, otherwise the object is considered as a non-tree object. For the candidate tree trunk, vision-based tree trunk detection was implemented and the final rate of confidence for the tree trunk (ROC_{Tree}) was obtained from laser, colour and edge detection stages as follows:

$$ROC_{Tree} = ROC_L W_L + ROC_C W_C + ROC_E W_E \tag{4.17}$$

where W_L, W_C and W_E represent the weight of the laser scanner, colour and edge detection respectively which need to be determined empirically prior to the algorithm implementation. The ROC_{Tree} was compared with a preset threshold value (TH_{Tree}) to decide whether the candidate tree trunk is a tree trunk or non-tree object. Figure 4.9 shows the flowchart of Detection Algorithm A.

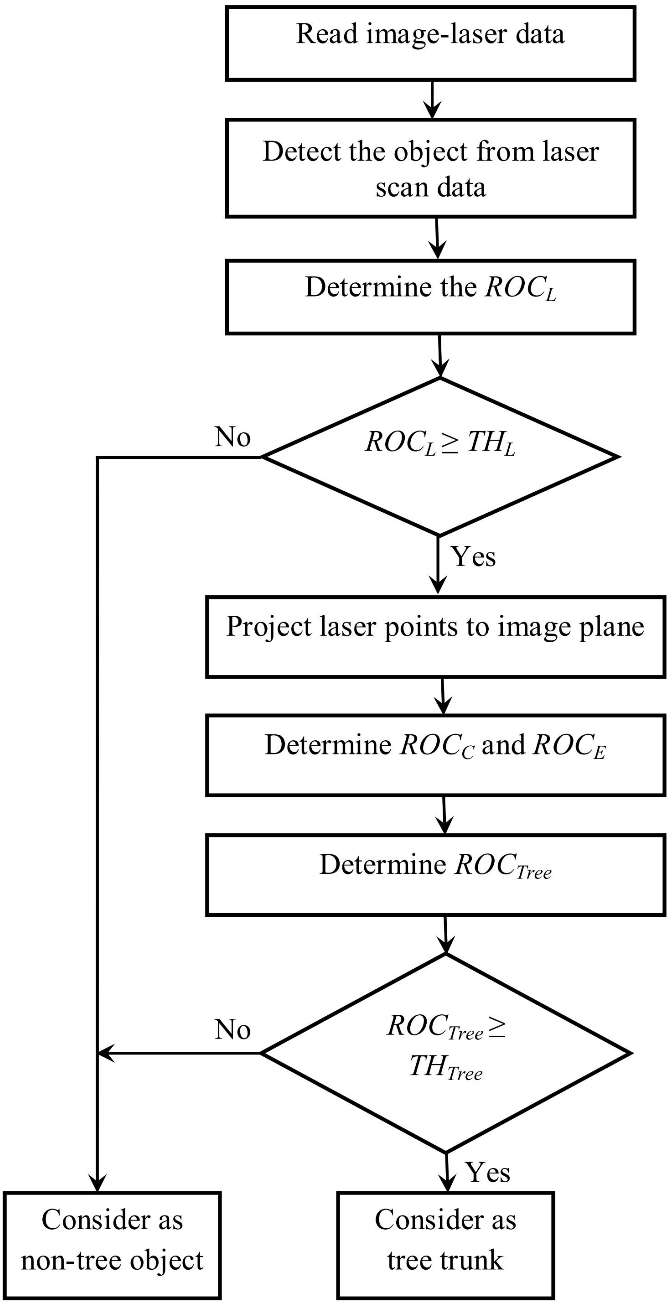
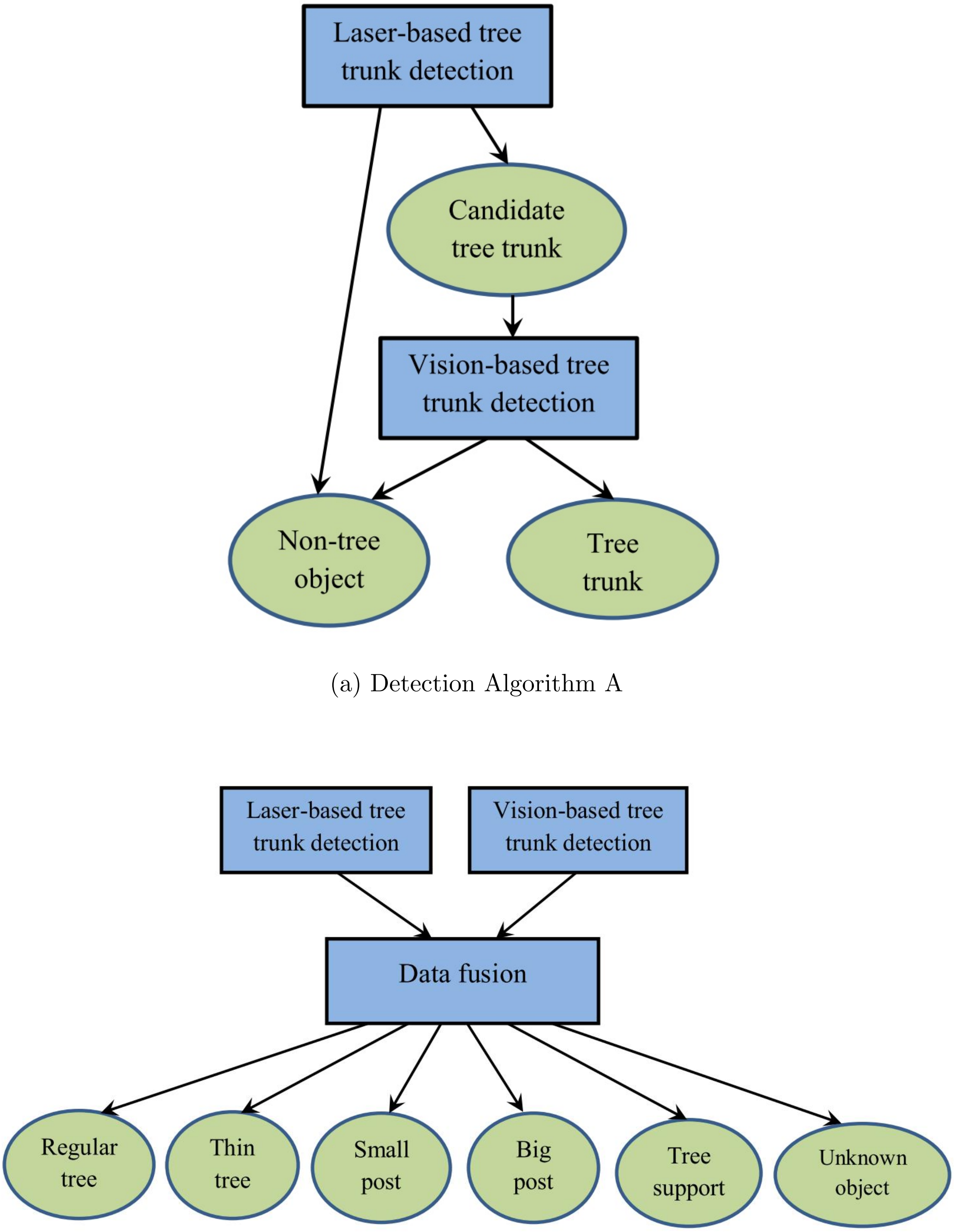

Read image-laser data
Detect the object from laser scan data
Determine the ROC_L
$ROC_L \geq TH_L$
No
Yes
Project laser points to image plane
Determine ROC_C and ROC_E
Determine ROC_{Tree}
$ROC_{Tree} \geq TH_{Tree}$
No
Yes
Consider as non-tree object
Consider as tree trunk

Figure 4.9: Detection Algorithm A for trees of similar trunk size.


Laser-based tree trunk detection
Candidate tree trunk
Vision-based tree trunk detection
Non-tree object
Tree trunk
Laser-based tree trunk detection
Vision-based tree trunk detection
Data fusion
Regular tree
Thin tree
Small post
Big post
Tree support
Unknown object

(a) Detection Algorithm A

(b) Detection Algorithm B.

Figure 4.10: The block diagram of the two tree trunk detection algorithms.

4.3.2 Non-uniformly sized tree trunk detection – Detection Algorithm B

The Detection Algorithm B was designed to detect trees with non-uniformly sized tree trunks using vision and laser scanner data fusion. This algorithm used the same laser scanner and vision detection routines, but in a different sequence to discriminate between different types of trees and non-tree objects as shown in Figure 4.10b. The Detection Algorithm B evaluates object width and colour concurrently instead of using width as the primary discriminator of trees and non-tree objects. It was observed that the tree trunks, posts and tree supports have vertical edges. Therefore, the vision-based edge detection was only used to distinguish between the big posts and the unknown objects that might have non vertical edges (e.g. animals, big rocks).

In Detection Algorithm B, the ROC_L, ROC_C and ROC_E are determined for each tree and non-tree object. Threshold values TH_L, TH_C and TH_E for ROC_L, ROC_C and ROC_E respectively were estimated prior to the algorithm implementation. Detailed information regarding the setting of these thresholds can be found in Section 4.5.1. The ROC_L, ROC_C and ROC_E are compared with these threshold values as shown in Figure 4.11 to distinguish between all the types of trees and non-tree objects in the orchard.

The algorithm's colour classification of the tree trunks using the H_d distribution, was made adjustable. This was achieved by accumulating the new tree trunk H_d values from subsequent data collection events that were not included in the previous H_d distribution into the previous H_d distribution. This process provides a recalibration of the colour parameters which will enhance Detection Algorithm B accuracy.

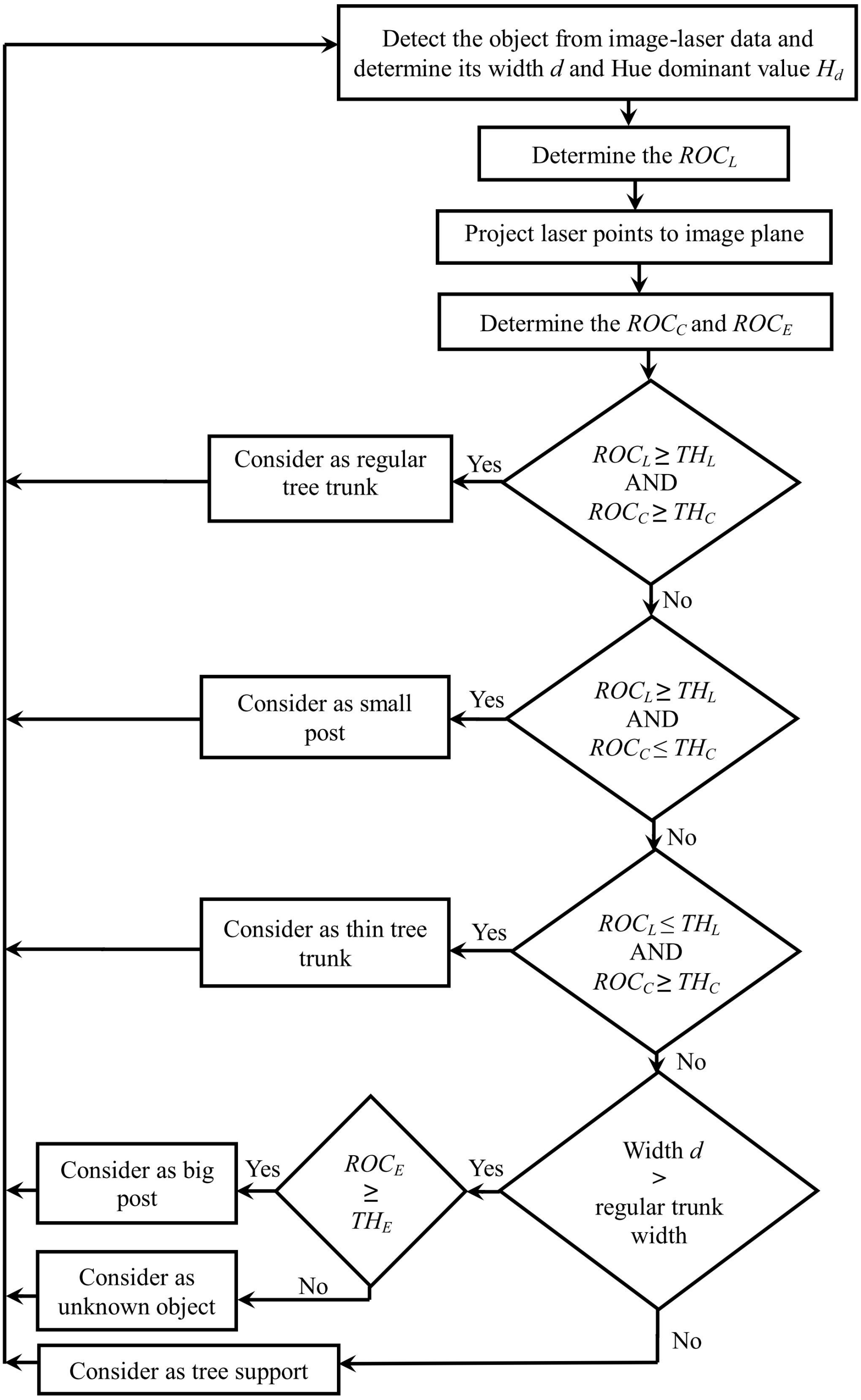

Detect the object from image-laser data and determine its width d and Hue dominant value H_d
Determine the ROC_L
Project laser points to image plane
Determine the ROC_C and ROC_E
$ROC_L \geq TH_L$ AND $ROC_C \geq TH_C$
Yes
Consider as regular tree trunk
No
$ROC_L \geq TH_L$ AND $ROC_C \leq TH_C$
Yes
Consider as small post
No
$ROC_L \leq TH_L$ AND $ROC_C \geq TH_C$
Yes
Consider as thin tree trunk
No
Width d > regular trunk width
Yes
$ROC_E \geq TH_E$
Yes
Consider as big post
No
Consider as unknown object
No
Consider as tree support

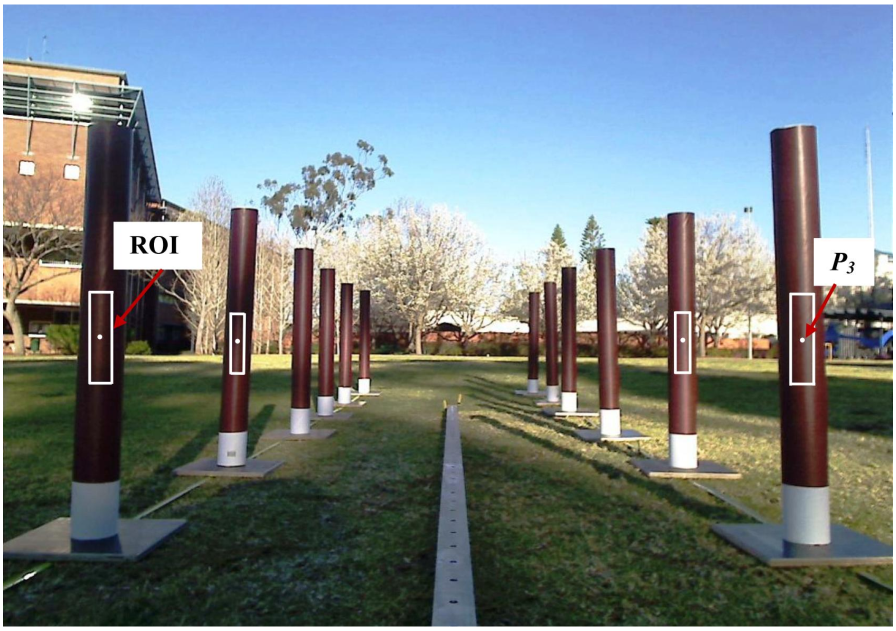
Figure 4.11: Detection Algorithm B to detect non-uniformly sized tree trunks in the orchard.

4.4 Preliminary test of Detection Algorithm A in the simulated environment

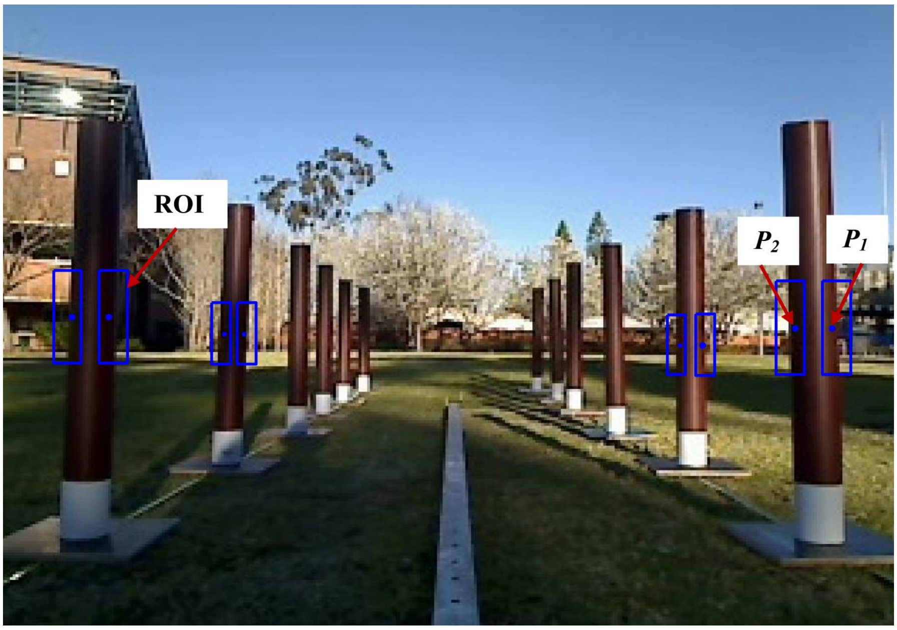
Detection Algorithm A was tested using two rows of simulated tree trunks (T_1-T_{12}). Each row consists of 6 simulated tree trunks as shown in Figure 4.12. The mobile robot was moved from a known starting position midway between the two rows to collect the image-laser scan pairs. The mobile robot movement was remotely controlled to follow midway between the simulated rows. For each image-laser scan pair, Detection Algorithm A was implemented. Figure 4.12 shows the simulated tree trunks with the selected ROI around the centre and edges of the simulated tree trunks.

Four objects (B_1-B_4) were inserted in the simulated environment at different locations between the rows and outside the rows to test Detection Algorithm A ability to distinguish between the simulated tree trunks and the objects. The objects were additional mailing tubes that were modified to be either different in geometry or different in colour. Three of these objects (B_1,B_3, and B_4) had a width of 170mm. The fourth object (B_2) had the same dimensions of the simulated tree trunks but was a different colour.

Table 4.1 shows the results of the mean of the ROC_{Tree} of each individual simulated tree trunk for N number of image-laser scan pairs. From the results, it can be seen that the range of the mean of the ROC_{Tree} for the simulated tree trunks was between 0.786 to 0.903 which is acceptable for identifying tree trunks.



(a) ROI for colour detection.



(b) ROI for edge detection.

Figure 4.12: The simulated tree trunks with the ROI around the centre and edges.

Table 4.2 shows the results of each non-tree object. The objects B_1, B_3, and B_4 have low ROC_{Tree} because they have a different width than the simulated tree trunks. The algorithm was capable of distinguishing them from the laser scanner

Table 4.1: The results of the Detection Algorithm A using the simulated tree trunks.

Simulated tree trunk	N	Mean of ROC_{Tree}
T_1	5	0.903
T_2	5	0.888
T_3	6	0.867
T_4	8	0.890
T_5	6	0.883
T_6	8	0.854
T_7	7	0.885
T_8	6	0.875
T_9	6	0.860
T_{10}	6	0.877
T_{11}	5	0.786
T_{12}	5	0.878

data by determining their width and ROC_L. These objects had ROC_L less than the TH_L and were considered as non-tree objects. The ROC_{Tree} of B_2 was higher than the other objects since B_2 had the same width as the simulated trees but a different colour. This object was detected by the laser scanner as candidate tree trunk and the colour of this object was compared with the range of simulated tree trunk colour. The algorithm detected it as a non-tree object because its colour was not within the simulated tree trunk colour range.

Table 4.2: Simulated environment results for non-tree objects.

Non-tree object	N	Mean of ROC_{Tree}
B_1	5	0.0009
B_2	5	0.461
B_3	5	0.0012
B_4	5	0.0008

4.5 Experimental results and discussion for real orchard

Orchard tests were conducted using the CoroWare Explorer platform to extract the tree trunk detection features and to evaluate the performance of the Detection Algorithm B.

4.5.1 Tree trunk features extraction

The tree trunk feature extraction was achieved by moving the mobile robot between multiple tree rows in the orchard to collect the required image-laser scan pairs for each tree trunk. A set of 100 regular tree trunks selected randomly from the orchard was used to extract the initial distribution of the regular tree trunk width. The sampling frequency of the data collection from the on-board sensors was 7 Hz. It was observed that each tree trunk was detected in a number of image-laser scan pairs (N) ranging from 22 to 26. The following parameters were determined from the laser scanner data set prior to implementing the Detection Algorithm B:

- The mean of the width d of each tree trunk for N image-laser scan pairs to generate the initial tree trunk width distribution.
- The μ_d and σ_d for the initial tree trunk distribution.
- The ROC_L using Equation 4.3 for all the tree trunks and the TH_L.

Figure 4.13 shows the histogram of the width distribution of the 100 selected regular tree trunks. The resulting values of the μ_d and σ_d were 117.7 mm and 15.6 mm respectively.

Figure 4.14 shows the histogram of the ROC_L distribution of the same 100 regular tree trunks. The lowest value of ROC_L of the regular tree trunks was selected as TH_L.

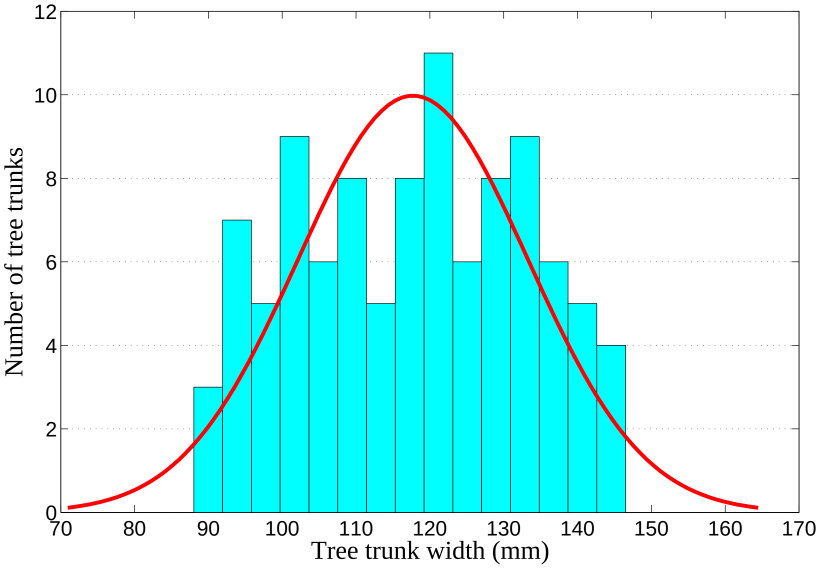


Figure 4.13: Histogram of the regular tree trunk width distribution.

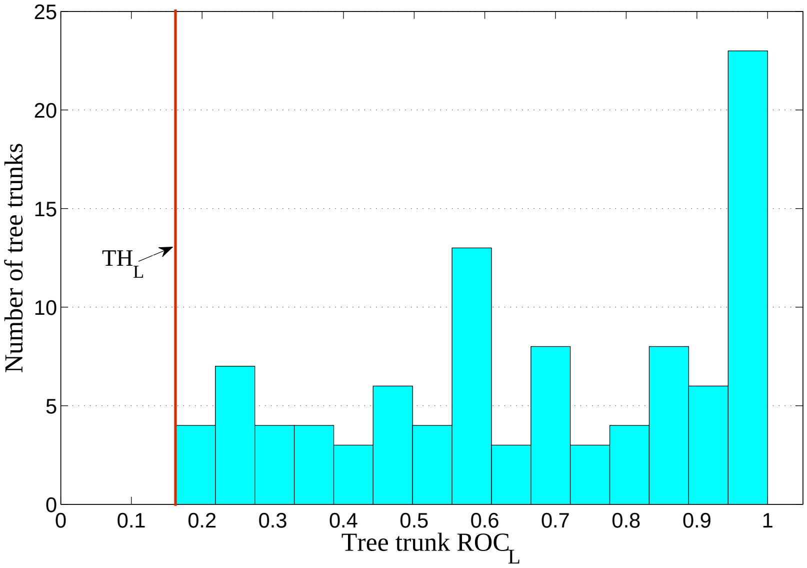
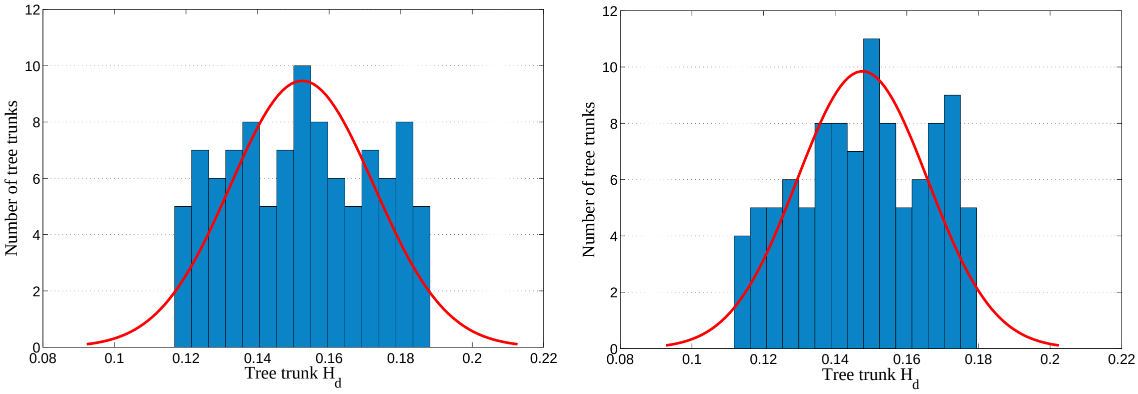


Figure 4.14: Histogram of the ROC_L distribution of the regular tree trunks.

Two sets of images were collected for 100 tree trunks selected randomly from the orchard under different illumination conditions to study the effect on H_d values. These were labelled either ‘sunny’ in which there was either clear sky (zero oktas) or less than 1 okta of patchy cloud cover; or ‘cloudy’ in which the cloud cover was either completely overcast (8 oktas) or very extensive (at least 6 oktas). These were chosen because they represent the extremes of illumination and are also the two main illumination conditions observed at this orchard location. The normalised values of H_d, described as values between 0 and 1 are used in this study. The following parameters were determined from each image data set prior to implementing the Detection Algorithm B:

- The mean of the H_d of each tree trunk for N image-laser scan pairs to generate the initial tree trunk H_d distribution.
- The μ_{H_d} and σ_{H_d} for the initial tree trunk H_d distribution.
- The ROC_C using Equation 4.14 for all the tree trunks and the TH_C.

Figure 4.15 shows the histogram of the initial H_d distribution of the 100 tree trunks in typical sunny and cloudy days obtained in May 2014. A two-sample Kolmogorov−Smirnov test was conducted to see if the two distributions of H_d of the tree trunks in sunny condition versus cloudy condition are significantly different. The null hypothesis, that the two distributions of H_d are not significantly different from each other, was accepted at a significance level of 5%.



(a) Typical sunny day distribution.

(b) Typical cloudy day distribution.

Figure 4.15: The histogram of the initial H_d distribution of the tree trunks.

The H_d distributions for both sunny and cloudy illumination conditions were combined by adding the sunny and cloudy measurements together to include the full range of the tree trunk H_d. Figure 4.16 shows the histogram of the H_d distribution for the combined sunny and cloudy illumination conditions. Table 4.3 illustrates the tree trunk H_d range, μ_{H_d} and σ_{H_d} for the initial sunny, cloudy and combined distribution collected in May 2014.

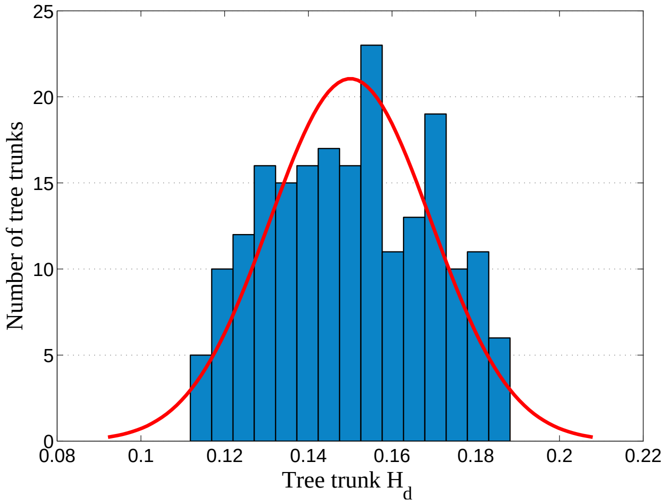


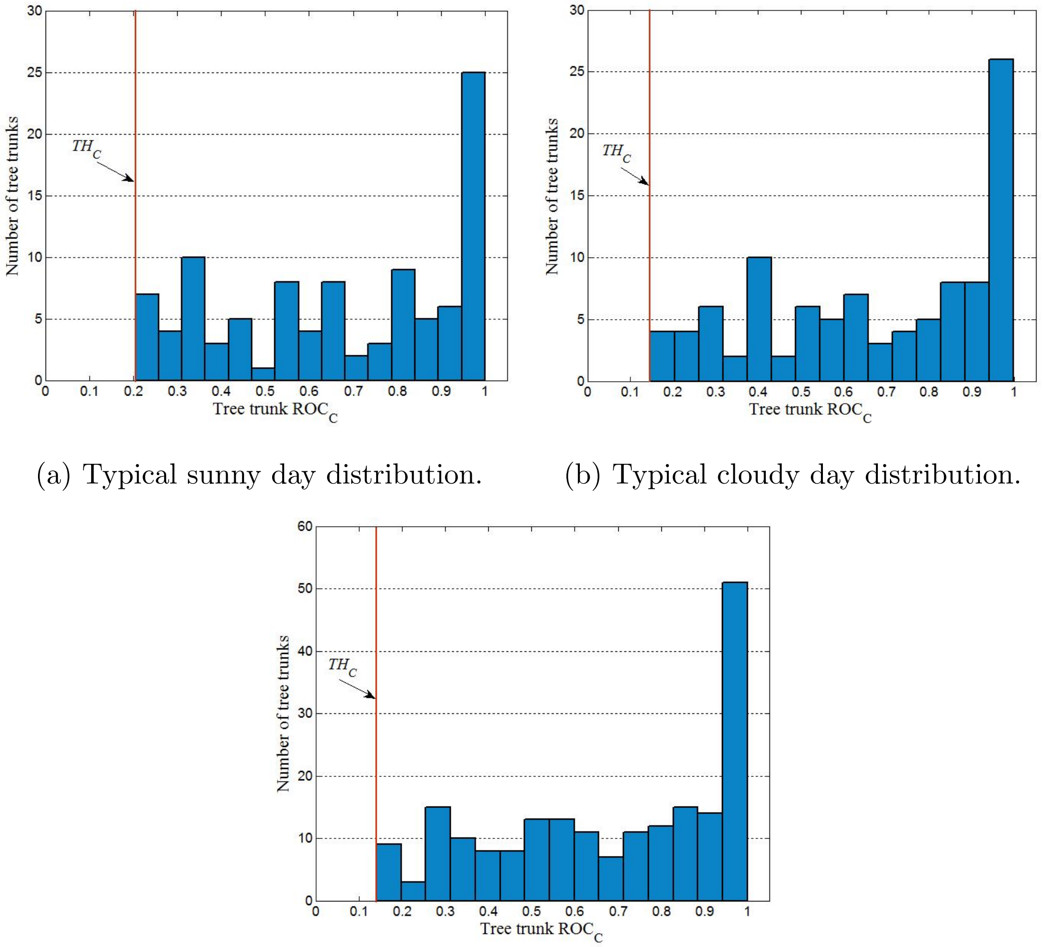
Figure 4.16: Histogram of the tree trunk H_d distribution for the combined sunny and cloudy illumination conditions.

Table 4.3: The results of the initial sunny, cloudy and combined tree trunk H_d distributions.

Distribution type	H_d range	μ_{H_d}	σ_{H_d}
Sunny (May 10, 10-11 am)	0.1168-0.1882	0.1525	0.0201
Cloudy (May 18, 9-10 am)	0.1118-0.1795	0.1476	0.0183
Combined sunny and cloudy	0.1118-0.1882	0.15	0.0193

The tree trunk width distribution in Figure 4.13 and the H_d distributions in Figure 4.15 and Figure 4.16 were tested using z-test which is a statistical test used to determine whether a sample data set comes from a population with a normal distribution. The null hypothesis that these distributions were normally distributed was accepted at a significant level of 5%. Therefore, these distributions were considered as approximately normally distributed.

Figure 4.17 shows the histogram of the ROC_C distribution of the tree trunks for the sunny, cloudy and combined distributions. The lowest value of the ROC_C for each distribution was chosen as TH_C to distinguish between tree trunks and non-tree objects. The values of TH_C of the sunny, cloudy and combined distributions were 0.2048, 0.1469 and 0.1406 respectively.



(a) Typical sunny day distribution.

(b) Typical cloudy day distribution.

(c) Combined distribution.

Figure 4.17: The histogram of the ROC_C distribution of the tree trunks.

Figure 4.18 shows the histogram of the ROC_E distribution of the same 100 tree trunks. The lowest value of ROC_E was chosen as TH_E which is used in the Detection Algorithm B to distinguish between big posts and unknown objects (e.g. animals, big stones) since the big posts were observed to have similar edges as the tree trunks.

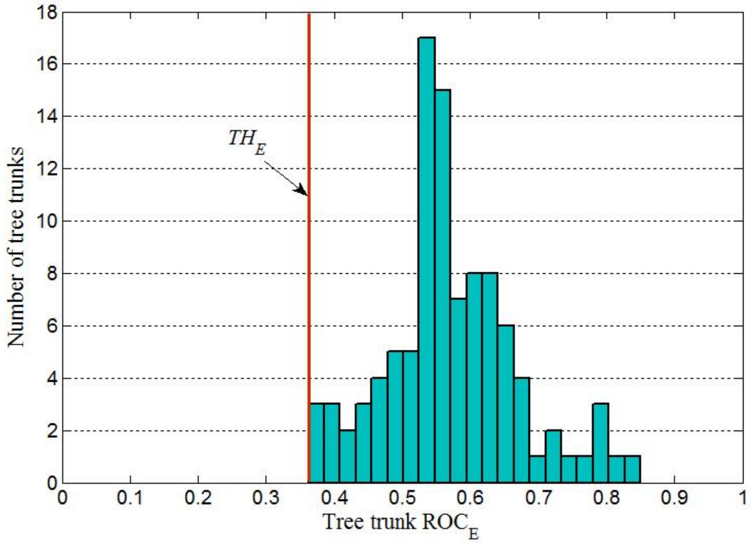


Figure 4.18: Histogram of the ROC_E distribution of the tree trunks.

4.5.2 Detection Algorithm B test results and discussion

An area of approximately 50 m by 20 m in length and width from the persimmon orchard, which contained five tree rows, was selected for experimental tests and evaluation of the developed tree trunk detection algorithm. These rows contain 96 trees (90 regular trees and 6 thin trees) and 23 non-tree objects (10 big posts, 5 small posts and 8 tree supports). The experimental tests were carried out on four days in 2014 as follows:

1. Sunny day on May 20 from 1:30 pm to 2 pm.
2. Cloudy day on June 28 from 10 am to 10:30 am.
3. Cloudy day on July 26 from 2:30 pm to 3 pm.
4. Sunny day on September 27 from 9 am to 9:30 am.

These four tests covered the autumn, winter and spring seasons, where the leaf trash on the ground varied in amount but was not observed to affect laser scanner readings and image projection. The growth of the tree trunk width over the experimental test periods was very small with a median variation of 0.28% from the initial tree trunk width distribution. From the colour data collected, it was observed that the tree trunks were separable from non-tree objects based on H_d values for all days.

4.5.2.1 Results using the initial H_d distributions

Four tests (Test 1—Test 4) were carried out to evaluate the performance of the Detection Algorithm B using the initial sunny, cloudy and combined tree trunk H_d distributions from Table 4.3. The algorithm was able to detect the tree trunks and distinguish between tree trunks and non-tree objects in the orchard. Each tree trunk or non-tree object may considered as one of the following definitions:

- True positive (TP): Tree trunks correctly identified as tree trunks.
- False positive (FP): Non-tree objects incorrectly identified as tree trunks.
- True negative (TN): Non-tree objects correctly identified as non-tree objects.
- False negative (FN): Tree trunks incorrectly identified as non-tree objects.

For each test the sensitivity (recall), precision and accuracy were calculated using Equation 4.18, Equation 4.19 and Equation 4.20 respectively. Table 4.4 summarises the results of the Detection Algorithm B.

$$Sensitivity = \frac{TP}{TP + FN} \tag{4.18}$$

$$Precision = \frac{TP}{TP + FP} \tag{4.19}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.20}$$

Table 4.4: Detection Algorithm B results for Test 1-Test 4 using the initial sunny, cloudy and combined distributions from Table 4.3.

Test	Illumination condition	H_d distribution	TP	FN	TN	FP	Sensitivity (%)	Precision (%)	Accuracy (%)
Test 1	Sunny	Sunny	87	9	21	2	91	98	91
(May 20,		Cloudy	86	10	21	2	90	98	90
1:30-2 pm)		Combined	90	6	21	2	94	98	93
Test 2	Cloudy	Sunny	86	10	21	2	90	98	90
(June 28,		Cloudy	86	10	20	3	90	97	89
10-10:30 am)		Combined	88	8	20	3	92	97	91
Test 3	Cloudy	Sunny	85	11	21	2	89	98	89
(July 26,		Cloudy	87	9	21	2	91	98	91
2:30-3 pm)		Combined	88	8	21	2	92	98	92
Test 4	Sunny	Sunny	85	11	22	1	89	99	90
(September 27,		Cloudy	84	12	22	1	88	99	90
9-9:30 am)		Combined	87	9	22	1	91	99	92

From Table 4.4, the results indicate that there is no significant difference between the detection accuracy of the algorithm using the initial sunny and cloudy distributions over the four tests. However, Detection Algorithm B produced better results when using the combined distribution. Therefore, the combined distribution was used for later tests, since it covers the range of both sunny and cloudy distributions.

4.5.2.2 Results using the adjusted H_d distributions

To enhance Detection Algorithm B accuracy, the tree trunk H_d distribution was automatically recalibrated after each test. This was achieved by adding the H_d values of the tree trunks that were incorrectly identified in the current test to be included in the new H_d distribution used in the next test. Table 4.5 shows the tree trunk H_d range, μ_{H_d}, σ_{H_d} and TH_C for the initial combined distribution and the adjusted distribution after each test. From Table 4.5, there is no significant logical seasonal shift in μ_{H_d}.

Table 4.6 illustrates the results of Detection Algorithm B for the same four tests using the adjusted H_d distributions. These results indicate that the adjusted H_d

Table 4.5: The parameters of the initial combined distribution and the adjusted combined distribution after each test.

H_d distribution	H_d range	μ_{H_d}	σ_{H_d}	TH_C
Initial distribution	0.1118-0.1882	0.15	0.0193	0.1406
Distribution after Test 1	0.1086-0.1908	0.1497	0.0202	0.1253
Distribution after Test 2	0.1063-0.1925	0.1504	0.0211	0.1176
Distribution after Test 3	0.1042-0.1925	0.1492	0.0218	0.1186

distributions significantly improved the detection accuracy for all tests. For example, the accuracy in Test 4 was increased from 92% using the initial distributions to 97% using the adjusted distribution after Test 3.

Table 4.6: Detection Algorithm B results using the adjusted H_d distribution.

Test	TP	FN	TN	FP	Sensitivity (%)	Precision (%)	Accuracy (%)
Test 1	90	6	21	2	94	98	93
Test 2	90	6	20	3	94	97	92
Test 3	92	4	21	2	96	98	95
Test 4	93	3	22	1	97	99	97

4.5.2.3 Test 4 results and discussion

Test 4 has been used in this discussion because it had the full benefit of adjusted colour parameters and achieved greatest detection accuracy because the distribution of H_d used in Test 4 was adjusted three times (after Test 1, Test 2 and Test 3). This means it includes wider range of H_d values. Figure 4.19 shows the width of the tested tree trunks determined from the laser data of Test 4. In this figure, there are six thin tree trunks. The algorithm assigns these thin tree trunks a very low ROC_L value less than the TH_L as shown in Figure 4.20.

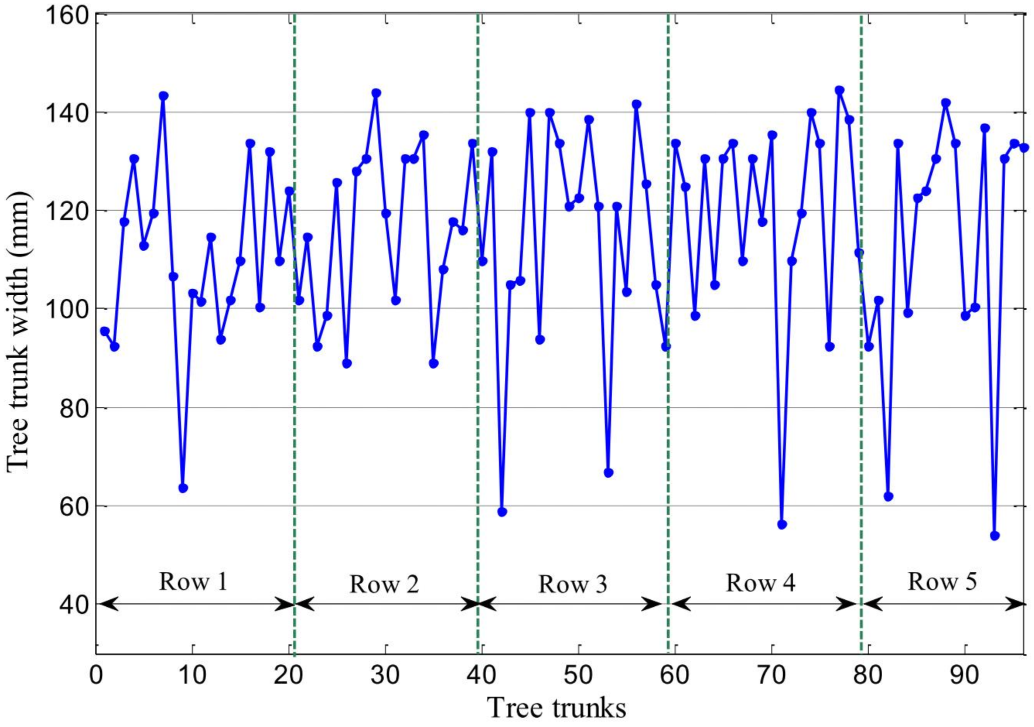


Figure 4.19: The distribution of the width of the tested tree trunks for Test 4.

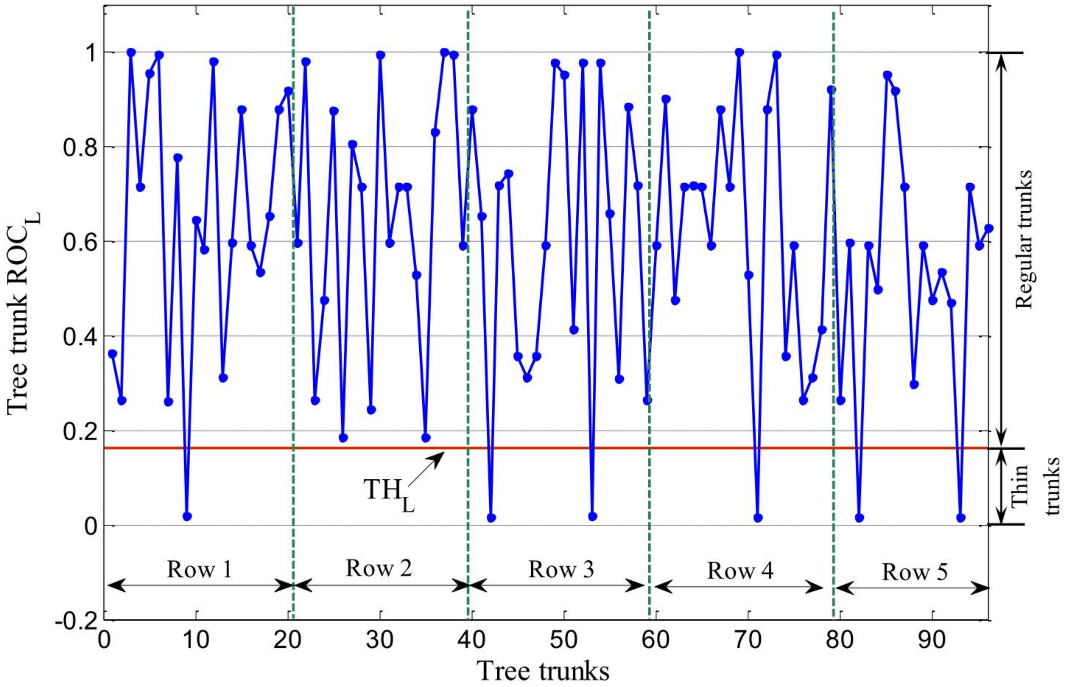


Figure 4.20: The distribution of the ROC_L of the tested tree trunks for Test 4.

Figure 4.21 illustrates the H_d values of the same tree trunks in Test 4, whilst Figure 4.22 shows the ROC_C of the same tree trunks using the adjusted H_d distribution after Test 3. Three tree trunks (two regular tree trunks and one thin tree trunk) were incorrectly identified as they had H_d values out of the adjusted H_d range by $2-3\%$ as shown in Figure 4.21. The algorithm assigns a low ROC_C for these three tree trunks which is less than the TH_C as depicted in Figure 4.22.

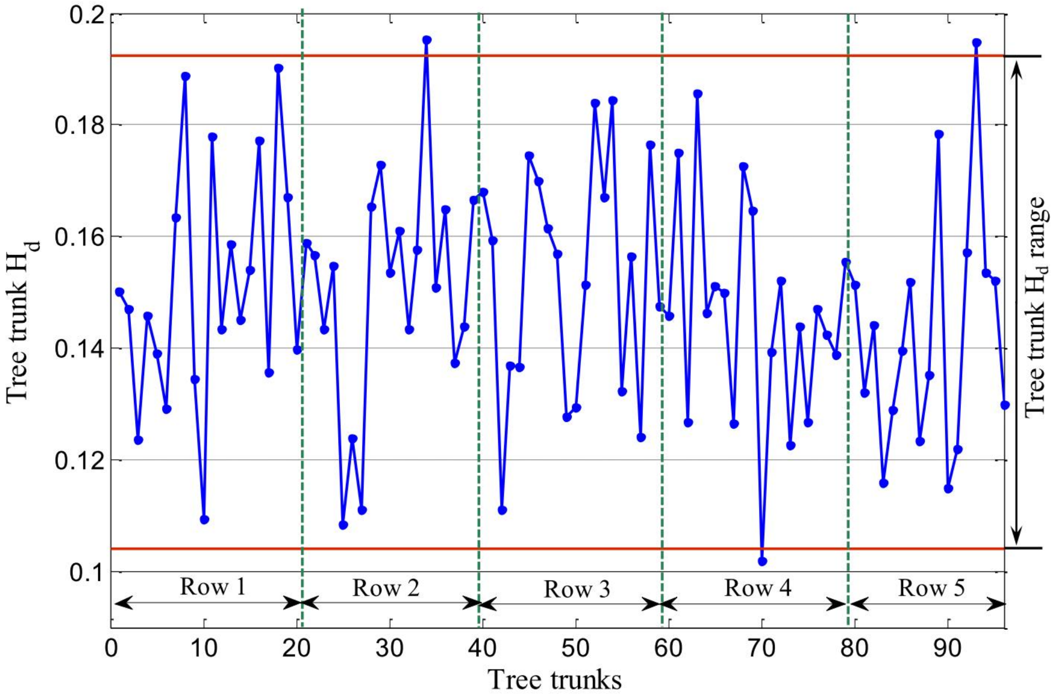


Figure 4.21: The H_d distribution of the tested tree trunks for Test 4.

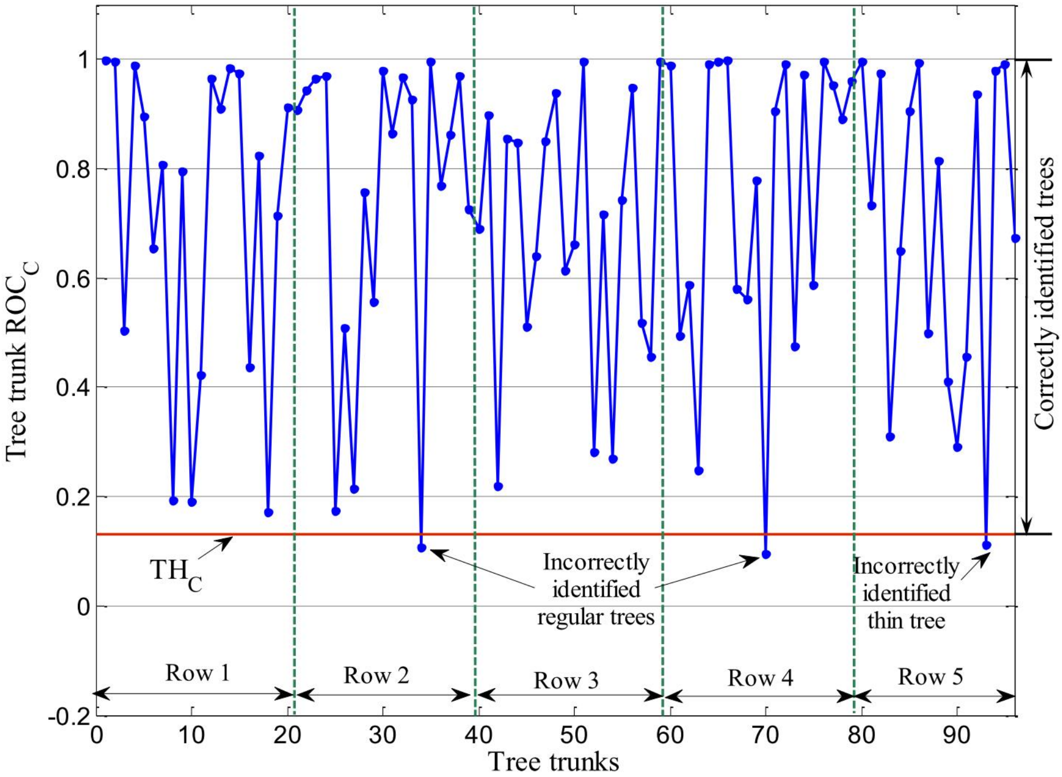


Figure 4.22: The distribution of the ROC_C of the tested tree trunks for Test 4.

Table 4.7 illustrates the detailed results of the tested tree trunks and the non-tree objects of Test 4 using Detection Algorithm B. From Table 4.7, Detection Algorithm B failed in detecting two regular tree trunks and considered them as small posts. These two regular tree trunks passed the laser detection but failed the colour detection because their H_d values were 0.1962 and 0.1015 which are out of the tree trunk H_d range presented in Table 4.5. The algorithm also failed in detecting one thin tree trunk and considered it as tree support since its H_d value was 0.1954 which is out of the tree trunk H_d range. In addition, one small post was considered as a regular tree trunk because its H_d value was 0.1125 which is within the tree trunk H_d range. However, the algorithm detected all the big posts and the tree supports correctly.

Table 4.7: The detailed results of the tree trunks and the non-tree objects of Test 4.

Object type	Tested objects	Correctly identified	Incorrectly identified
Regular tree trunks	90	88	2
Thin tree trunks	6	5	1
Big posts	10	10	0
Small posts	5	4	1
Tree supports	8	8	0

4.6 General discussion

The developed tree trunk detection algorithm performed successfully in a real orchard environment. The variety of the trees and non-tree objects in the selected orchard helped to make the algorithm more general and potentially applicable to any well-maintained orchard in which the tree trunks and the non-tree objects are separable in colour. The Detection Algorithm B can be used in another orchard with the same type of trees without repeating the procedure for the initial distributions of H_d and tree trunk width. The algorithm can use the same distributions and thresholds that were already determined from previous tests as initial distributions and thresholds in the new orchard. Then the algorithm will automatically adjust the distributions and the thresholds after the first test in the new orchard. However, when implementing this algorithm in another orchard with different type of trees, an initial tree trunk width and H_d distributions would need to be determined since they vary from one orchard to another depending on the type of the trees. Subsequently, the system will perform automatic recalibration after each test. The developed algorithm can be modified to be used in other orchards which have more classes of trees and non-tree objects by including a separate distribution for each type of the trees and non-tree objects and adding extra features to distinguish between them.

If the tree trunks and non-tree objects cannot be separated by colour the algorithm would need to be modified to extract another feature from the images such as texture. Another situation that might affect the performance of the algorithm is when the tree trunk is occluded by many leaves e.g. on a broken branch hanging low to the ground, such that the individual tree trunks cannot be recognised. This could be a problem for other types of trees that their trunks are occluded by leaves.

The algorithm used the fusion of the laser scanner and camera data to enhance the detection capability in different illumination conditions and more object types. It was observed that the tree trunk width measured by the laser scanner was not affected by the illumination conditions. In addition, the use of HSV colour space was effective since the Hue dimension of the colour is less affected by the illumination conditions and it is a robust indicator for identifying the same colour and separating objects with different colours.

The processing time of the image depends on the number of pixels of the image. As the selected ROIs for colour and edge detection are small parts of the whole image, then these ROIs required less processing time than processing the whole image. This helped reducing the processing time and minimising the effect of unused information in the image. The tree detection algorithm suggested in this study was successfully used for mapping the individual trees in the orchard and mobile robot localisation.

The accuracy achieved by the developed tree trunk detection algorithm was 97% which is higher than the accuracy of 94.7% obtained by Ali et al. (2008) and the accuracy of 91.7% achieved by He et al. (2011). The developed algorithm requires less image processing time than the algorithm presented by Auat Cheein et al. (2011). This is mainly because the projection sequence from the laser scanner coordinates to the image plane and constructing the ROI reduced the size of the processed images, whilst Auat Cheein et al. (2011) processed the whole images to detect the angle of the tree stem with respect to the mobile robot which required longer processing time. The tree trunk detection algorithm developed in this study was capable of detecting the tree trunks and discriminating between different tree trunks and non tree objects. However, the prior work (cited in Sections 2.1) paid considerably less attention to the discrimination between trees and non-tree objects in the rows of the orchard.

4.7 Conclusion

This chapter demonstrated a new method for detecting trees and non-tree objects such as posts and tree supports using a camera and laser scanner data fusion. The utilisation of both camera and laser scanner data enhanced the tree trunk detection. Projection from the laser scanner to the image plane and selecting the region of interest with the required features was effective since it reduced the processing time and minimised the effect of the noise in the other parts of the image. The developed algorithm relies only on the on-board sensors of the mobile robot without adding any artificial landmarks such as tags or reflective tapes on the trees in the orchard. The algorithm automatically adjusts the colour detection parameters after each test which was observed to improve the detection accuracy.

A small robot platform was used to collect the data required to implement the developed tree trunk detection algorithm. The algorithm was tested and evaluated through extensive experimental tests in a real orchard environment under different illumination conditions. At the end of Test 4, the algorithm was successful in detecting the tree trunks and discriminating between different tree trunks and non-tree objects with a detection accuracy of 97%.

Chapter 5

Orchard Map Construction

5.1 Introduction

Autonomous mobile robots are required to explore their environment without colliding with any stationary or moving obstacles to successfully perform the desired task. To efficiently carry out complex missions, autonomous robots need to maintain a map of their environment. Robotic mapping addresses the problem of acquiring spatial models of physical environments through the mobile robot's sensors. The availability of efficient mapping systems to produce accurate representations of the environment is one of the main requirements for successful autonomous mobile robot systems (Habib, 2007).

In agricultural environments, mapping is the process of constructing a map of the agricultural field with its most relevant features (e.g. trees, crops and objects). The knowledge of the location of the elements and the agricultural robots in agricultural environments plays a crucial role in the design of automatic agricultural robot (Auat Cheein and Carelli, 2013).

During the mapping process, a map of the surrounding environment is built and maintained to aid the navigation process or to perform a given task. Bad detection can lead to an incomplete or unreliable map which affects the navigation process (Auat Cheein and Carelli, 2013). A mobile robot can continuously update the map to incorporate, for example, fallen branches or other new obstacles. Precise orchard maps are essential for agricultural robots' path planning, localisation and navigation when detailed inspection (e.g. flowers, fruit, diseases) is to be undertaken.

This chapter starts with an explanation of the orchard map construction procedure using the mobile robot's on-board sensors in Section 5.2. Map construction of the simulated environment is presented in Section 5.3, whilst real orchard map construction is presented in Section 5.4.

5.2 Orchard map construction procedure

The map construction process is achieved by moving the mobile robot from a known starting position midway between each two tree rows in the orchard. The mobile robot is manually driven between the tree rows to collect the required data. The image-laser scan pairs are acquired at different predetermined positions measured by the RTK-GPS along the way between the two rows to implement tree trunk detection and mapping algorithm.

The mapping algorithm typically requires accurate detection of the tree trunks and estimation of trunk centre positions O_c in laser data. The tree trunks are detected using the combination of laser and vision based tree trunk detection algorithm presented in Section 4.3. Subsequently, the range r_c and angle ϕ_c, which represent the laser range and bearing angle respectively between the laser position on the mobile robot and the tree trunk centres, are determined from Figure 5.1 using the following equations:

$$r_c = \frac{r_1}{\cos(\frac{\Delta\phi}{2})} \tag{5.1}$$

$$\phi_c = \phi_1 - \frac{\Delta\phi}{2} \tag{5.2}$$

where r_1 and r_2 are the ranges at the edge points of the object, ϕ_1 and ϕ_2 are the object angles at r_1 and r_2 respectively and $\Delta\phi$ represents the difference between ϕ_1 and ϕ_2.

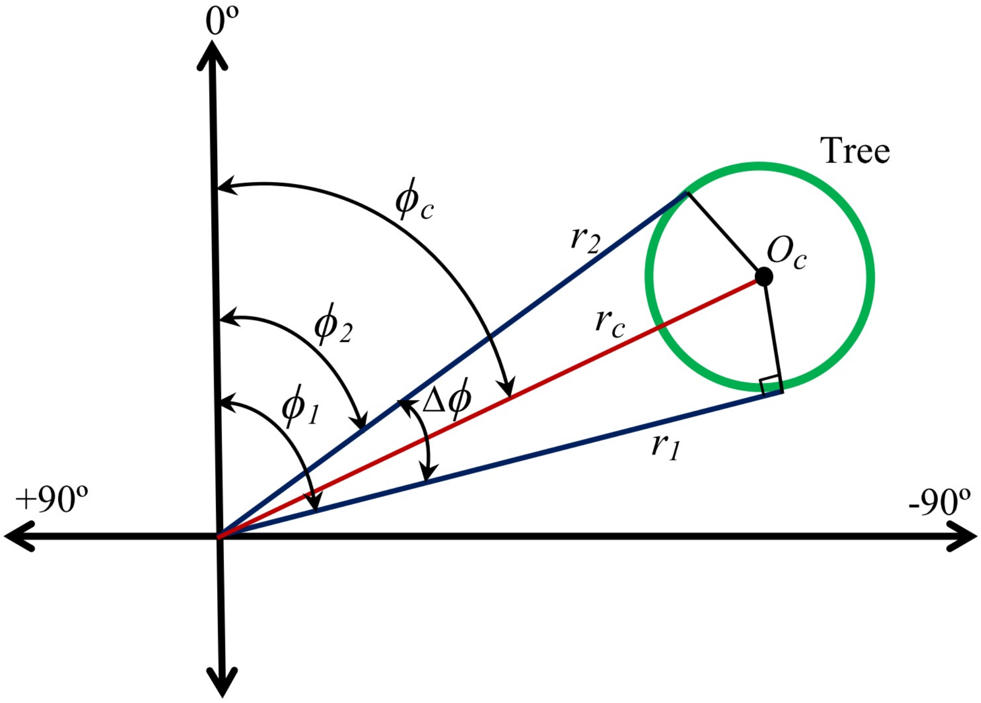


Figure 5.1: Principle of the determination of the range and the angle of the tree's centre position.

The tree trunk coordinates X_{tree}, Y_{tree} for each image-laser scan pair are determined from Figure 5.2 as follows:

$$X_{tree} = x + r_c \cos(\phi_c + \theta) \tag{5.3}$$

$$Y_{tree} = y + r_c \sin(\phi_c + \theta) \tag{5.4}$$

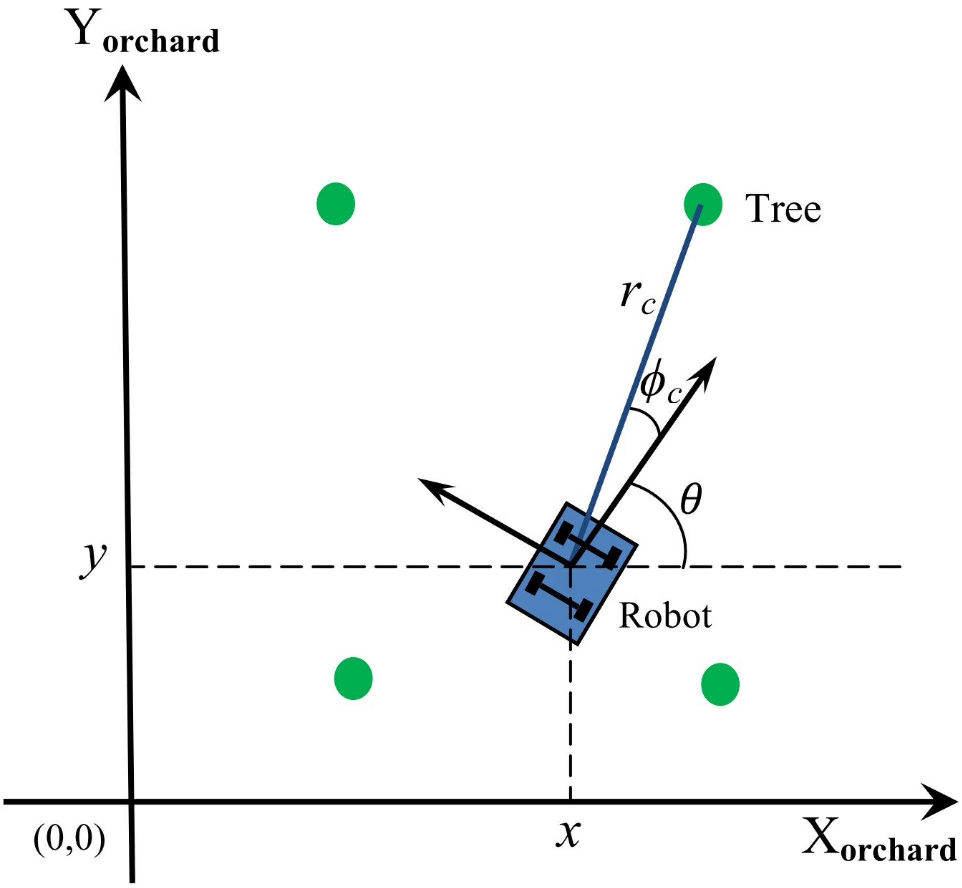


Figure 5.2: Graphical representation of the positions of the trees with respect to the mobile robot in the orchard environment.

where θ is the heading angle of the mobile robot platform and (x, y) represent the mobile robot coordinates in the orchard environment. The position of the mobile robot in the orchard is determined with respect to a selected reference point $(X_{orchard}, Y_{orchard})$ = (0,0) as shown in Figure 5.2. The position of the mobile robot in the orchard is obtained from the RTK-GPS data. The RTK-GPS provides x and y position data in UTM coordinates which must be converted to the orchard map reference coordinates $(X_{orchard}, Y_{orchard})$. Therefore, the x and y position data in UTM coordinates have been compensated for the starting position of the mobile robot to match the orchard map reference coordinates. The heading angle of the mobile robot is obtained from the IMU. Figure 5.3 shows the flowchart of the orchard mapping procedure.

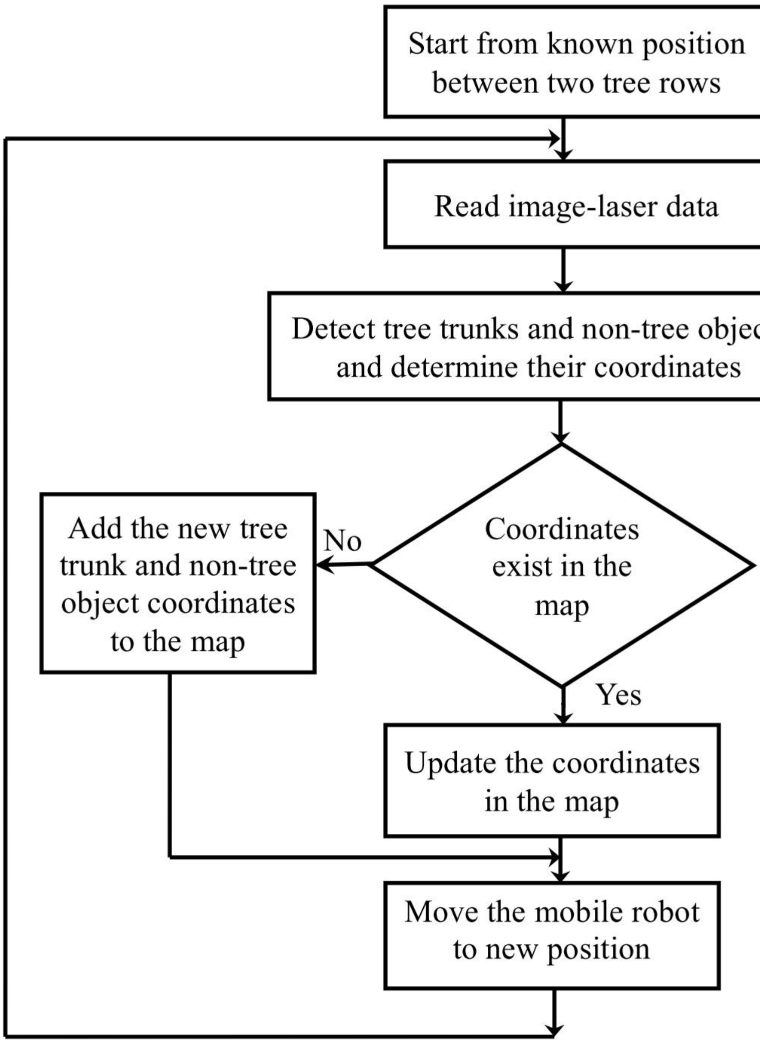

Start from known position between two tree rows
Read image-laser data
Detect tree trunks and non-tree objects and determine their coordinates
Coordinates exist in the map
No
Add the new tree trunk and non-tree object coordinates to the map
Yes
Update the coordinates in the map
Move the mobile robot to new position

Figure 5.3: The developed orchard mapping algorithm.

The coordinates of each individual tree trunk (X_{mean}, Y_{mean}) are determined by calculating the mean of X_{tree} and Y_{tree} for all the image-laser scan pairs of the same tree trunk. The same procedure is used to determine the position of the non-tree objects found in the tree rows.

The final map consists of the 2D coordinates of the trees and non-tree objects in the orchard. Another map m, which contains the 2D coordinates of the individual trees only, was extracted from the final map and stored in the on-board computer of the mobile robot to be used for mobile robot localisation and navigation.

The estimated map accuracy required in this research was considered to be 2-3% of the distance between the trees in the row. This accuracy was considered to be sufficient for mobile robot localisation and navigation in the orchard for tree inspection task.

5.3 Map construction of the simulated environment

The map construction algorithm was preliminary tested in the simulated environment described in Section 4.4. This was achieved by moving the mobile robot with its on-board sensors from a known starting position in straight line in the mid-way between the two simulated tree rows in equal steps. The position of the mobile robot in the simulated environment was determined manually using measuring tape with respect to the reference point $(X_{environment}, Y_{environment}) = (0,0)$ shown in Figure 5.4. This was considered acceptable for a small scale simulated environment (8m × 3m). The image-laser scan pair was acquired at each step to implement tree trunk detection and mapping algorithms to detect the simulated tree trunks and non-tree objects and determine their x and y positions.

Figure 5.4 shows the final map of the simulated environment which contains the simulated tree trunks (T_1-T_{12}) and the non-tree objects (B_1-B_4). Each tree trunk in this map is represented by a green circle, while each non-tree object is represented by a red circle.

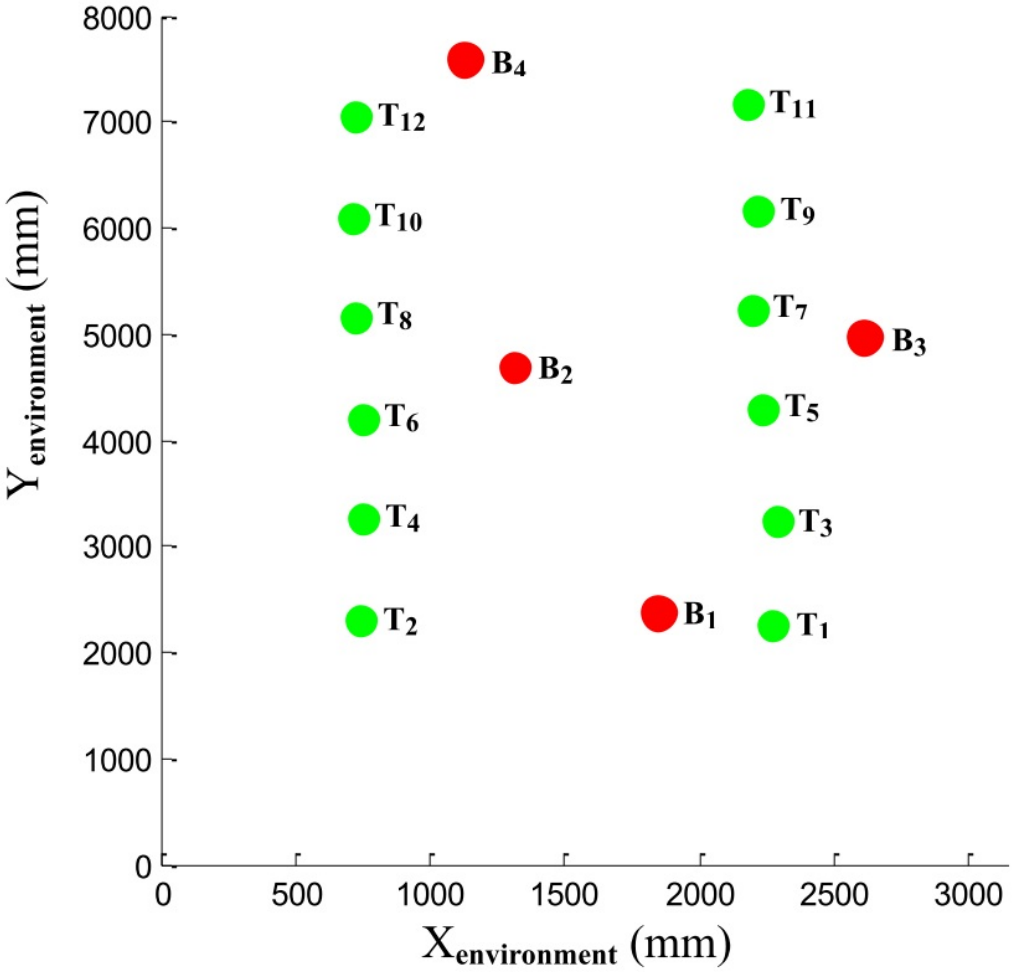


Figure 5.4: The map of the simulated environment.

The standard errors in x and y positions (SE_x, SE_y) of the simulated tree trunk and non-tree objects were calculated using Equation 5.5 and Equation 5.6 respectively:

$$SE_x = \frac{\sigma_x}{\sqrt{N}} \tag{5.5}$$

$$SE_y = \frac{\sigma_y}{\sqrt{N}} \tag{5.6}$$

where σ_x and σ_y are the standard deviation of the position data in x and y coordinates respectively. N is the number of image-laser scan pairs for each simulated tree trunk or non-tree object.

Table 5.1 shows the standard error results of each individual simulated tree trunk depicted in Figure 5.4. From the results, it can be seen that there is small variation in SE_x and SE_y results for the simulated tree trunks. This variation is due to the range measurement errors determined by the laser scanner. The angular resolution of the laser scanner also affects the standard error because the number of laser point data decreases with the range for the same object. This affects the bearing angle measurements and errors in SE_x and SE_y. Table 5.2 shows the standard error results of each non-tree object presented in Figure 5.4.

Table 5.1: The results of the standard errors for the simulated tree trunks (T_1-T_{12}) in the simulated environment depicted in Figure 5.4.

Simulated tree trunk	N	σ_x(mm)	SE_x(mm)	σ_y(mm)	SE_y(mm)
T_1	5	14.32	6.40	18.63	8.33
T_2	5	13.63	6.09	21.84	9.76
T_3	6	8.15	3.32	18.49	7.54
T_4	8	16.34	5.77	24.24	8.57
T_5	6	20.11	8.20	24.27	9.91
T_6	8	24.20	8.55	23.12	8.17
T_7	7	13.32	5.03	18.77	7.09
T_8	6	10.56	4.31	19.35	7.90
T_9	6	9.12	3.72	9.21	3.75
T_{10}	6	10.55	4.31	19.35	7.90
T_{11}	5	18.30	8.18	11.37	5.08
T_{12}	5	19.46	8.70	6.38	2.85

Table 5.2: The results of the standard errors for the non-tree objects (B_1-B_4) in the simulated environment shown in Figure 5.4.

Non-tree object	N	σ_x(mm)	SE_x(mm)	σ_y(mm)	SE_y(mm)
B_1	5	6.61	2.95	12.11	5.41
B_2	5	7.60	3.40	12.67	5.66
B_3	5	12.59	5.63	9.42	4.21
B_4	5	12.33	5.51	13.22	5.91

To verify the performance of the mapping algorithm, the accuracy of the constructed map of the simulated environment was determined using the root mean square (RMS) of the position error of the simulated tree trunks and non-tree objects. The RMS of the x and y position errors are calculated using Equation 5.7 and Equation 5.8 respectively:

$$E_{x_{rms}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_{gt}(i) - X_{mean}(i))^2} \tag{5.7}$$

$$E_{y_{rms}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_{gt}(i) - Y_{mean}(i))^2} \tag{5.8}$$

where $X_{gt}(i)$ and $Y_{gt}(i)$ are the ground truth position values in x and y coordinates of the $i-th$ simulated tree trunk and non-tree object which were measured manually using measuring tape. n is the number of tree trunks and non-tree objects of the simulated environment.

The accuracy of the constructed map of the simulated environment was also calculated using the RMS of the Euclidean distance (D_E) between the ground truth positions (X_{gt},Y_{gt}) and the calculated positions (X_{mean},Y_{mean}). The D_E of the $i-th$

simulated tree trunk and non-tree object is calculated from Equation 5.9, whilst the RMS of D_E is determined using Equation 5.10:

$$D_E(i) = \sqrt{(X_{gt}(i) - X_{mean}(i))^2 + (Y_{gt}(i) - Y_{mean}(i))^2} \tag{5.9}$$

$$D_{E_{rms}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(D_E(i))^2} \tag{5.10}$$

Table 5.3 summarises the calculated position, the ground truth measurement and the position error of each simulated tree trunk and non-tree object of the simulated environment. The $E_{x_{rms}}$, $E_{y_{rms}}$ and $D_{E_{rms}}$ were also determined from Table 5.3 and their values were 20 mm, 18 mm and 27 mm respectively. These values indicate that the developed mapping algorithm was capable of providing the necessary map accuracy for both localisation and navigation of the mobile robot in the orchard presented in Section 5.2.

Table 5.3: The results of the calculated position (X_{mean}, Y_{mean}), ground truth measurement (X_{gt},Y_{gt}), absolute value of the position error between the ground truth and the calculated position and D_E of each simulated tree trunk and non-tree object of the simulated environment shown in Figure 5.4.

Simulated trunk and object	X_{mean} (mm)	Y_{mean} (mm)	X_{gt} (mm)	Y_{gt} (mm)	$\lvert X_{gt} - X_{mean}\rvert$ (mm)	$\lvert Y_{gt} - Y_{mean}\rvert$ (mm)	D_E (mm)
T_1	2272	2271	2290	2285	18	14	23
T_2	750	2296	730	2280	20	19	28
T_3	2282	3275	2265	3260	17	15	23
T_4	754	3262	770	3280	16	18	24
T_5	2231	4297	2250	4275	19	22	29
T_6	742	4209	760	4195	18	14	23
T_7	2202	5236	2225	5220	23	16	28
T_8	725	5163	747	5180	22	17	28
T_9	2212	6160	2195	6140	17	20	26
T_{10}	715	6096	735	6115	20	19	28
T_{11}	2204	7171	2220	7155	16	16	23
T_{12}	720	7042	745	7065	25	23	34
B_1	1857	2383	1875	2400	18	17	25
B_2	1309	4680	1330	4695	21	15	26
B_3	2614	4950	2630	4970	16	20	26
B_4	1124	7578	1100	7560	24	18	30

5.4 Map construction of the real orchard

Experimental test was conducted on September 27, 2014 at 9 am to construct the map of the selected area of the orchard. The orchard map construction was achieved using the procedure described in Section 5.2. The tree trunks and non-tree objects in the selected area of the orchard were detected using Detection Algorithm B based on camera and laser scanner data fusion presented in Section 4.3.2. It was observed that each tree trunk was detected in a number of image-laser scan pairs ranging from 22 to 26. The detailed results of the correctly identified and incorrectly identified tree trunks and non-tree objects are summarised in Table 5.4 and their

2D coordinates are calculated.

Figure 5.5 shows the constructed map of the positions of the trees and non-tree objects in the selected area of the orchard. From this map, it can be seen that the tree supports were always located on the right side of the tree trunk. In addition, the posts in each row were located after every eight trees and were lined up for the five rows. There were small deviations in $x-$direction between the tree positions along the row, whilst the deviations in $y-$direction were larger. The constructed map was shown to be consistent with the visual verification of the selected area of the real orchard environment. The 2D coordinates of the tree trunks in the orchard map were extracted and saved in a map file to be used by the localisation algorithm using EKF.

Table 5.4: The detailed results of the correctly identified and incorrectly identified tree trunks and non-tree objects in the selected area of the orchard depicted in Figure 5.5.

Object type	Tested objects	Correctly identified	Incorrectly identified
Regular tree trunks	90	88	2
Thin tree trunks	6	5	1
Big posts	10	10	0
Small posts	5	4	1
Tree supports	8	8	0

5.5 General discussion

For the selected area of the orchard, it was not feasible to measure the ground truth of the tree trunk and non-tree object positions manually or by using RTK-GPS. The manual measurement is impractical in such a large area with uneven ground. In addition, RTK-GPS is not an absolute measure of tree trunk position because the RTK-GPS cannot be held over the centre of the tree trunk, and the tree canopy might block the signal from the satellites. Therefore, RTK-GPS was not

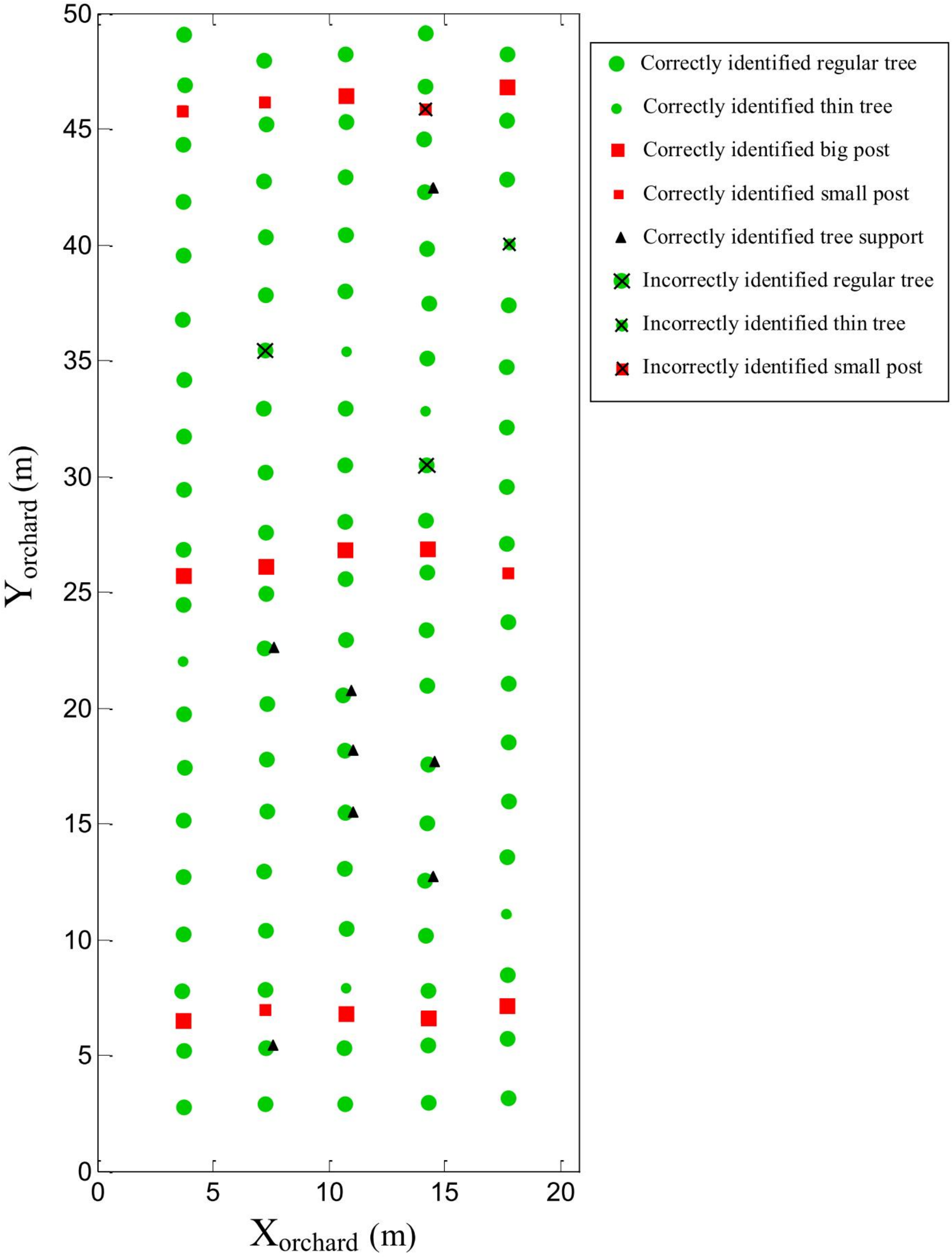

50
45
40
35
30
25
20
15
10
5
0
0
5
10
15
20
$Y_{orchard}$ (m)
$X_{orchard}$ (m)
Correctly identified regular tree
Correctly identified thin tree
Correctly identified big post
Correctly identified small post
Correctly identified tree support
Incorrectly identified regular tree
Incorrectly identified thin tree
Incorrectly identified small post

Figure 5.5: The map of the selected area of the orchard.

used to measure the ground truth position of the tree trunks and non-tree objects. Moreover, expensive surveying equipment was not available for this study.

The RTK-GPS and on-board sensors (camera, laser scanner and IMU) have been used to generate the map; and the map is subsequently used as a ground truth to evaluate the accuracy of the mobile robot localisation. In the absence of other ground truth methods (e.g. measuring of tree position using surveying methods), the map generated by RTK-GPS and on-board sensors is assumed to be an accurate representation of the orchard and suitable as a ground truth.

The uneven ground of the orchard produces some errors in the laser scanner measurements. This is because the laser scanner is not always perfectly perpendicular to each individual tree trunk when the robot is moving on uneven ground. The estimated error of tree trunk centre position using laser scanner data grows with increasing distance to the tree trunk. However, the developed algorithm considered only the closest tree trunks and objects to the mobile robot. In addition, not all the trees have perfect circular cross-section. This might produce some error in estimating the tree trunk centre position in the orchard.

An algorithm to construct a local orchard map of the individual trees and non-tree objects of the rows was presented in this study rather than mapping the whole tree row as a line (Hansen et al., 2011; Andersen et al., 2010). This allows the mobile robot to navigate to a specific tree in the orchard to implement different tree inspection tasks. The developed mapping algorithm relied only on the natural landmarks (i.e., trees) and the objects that already exist in the orchard rather than using artificial landmarks and reflective tapes (Libby and Kantor, 2011; Bergerman et al., 2015) which add extra cost to the farmers.

5.6 Conclusion

In this chapter, a method to generate a local map of the orchard was presented. This method mainly detects the tree trunks and the non-tree objects using camera and laser scanner data fusion and determines their 2D positions. The orchard map was constructed using the RTK-GPS, IMU, camera and laser scanner. This map was constructed only once and was saved on the mobile robot's on-board computer to be used as an *a priori* map for localisation and navigation. The integration of different sensors for feature extraction and orchard mapping improved the robustness of the map.

The constructed orchard map has shown to be consistent with the real orchard environment and can be reliably used for mobile robot localisation and navigation. Mapping each individual tree in the row, rather than the whole tree row as a line, allows the mobile robot to navigate to a specific tree in the orchard to perform tree inspection, pruning and harvesting tasks.

Chapter 6

Mobile Robot Localisation

6.1 Introduction

Localisation refers to the determination of the position and orientation of the mobile robot with respect to a fixed coordinate system. The problem of robot localisation consists of answering the question Where am I? from a robot's point of view (Negenborn, 2003). An accurate and reliable positioning system is an important component in autonomous mobile robots. For safe and successful navigation, the mobile robot has to know its pose within the field and the elements from the surrounding environment (Auat Cheein and Carelli, 2013).

The robot's knowledge of its pose within the environment is not reliable if the mobile robot has a bad localisation system. Thus, it would not be able to perform path following and tracking activities. Autonomous navigation without a precise knowledge of the actual pose in the agricultural field is dangerous for the robot's integrity and, more importantly, could represent a risk for field workers and the possibility of the failure of an agricultural task. Additionally, the agricultural mobile robot will not be able to perform any action associated with the agricultural task. For example, it will not be able to supervise a specific portion of an orchard or grove (Auat Cheein and Carelli, 2013).

The mobile robot needs to acquire relative and absolute measurements from on-board sensors giving the robot feedback about its driving actions and the situation of the environment around the robot in order to localise itself. Given this information, the mobile robot has to determine its pose as accurately as possible. What makes this difficult, is the existence of noise and errors in both the sensing and driving of the mobile robot. The uncertain information needs to be combined in an optimal way (Negenborn, 2003). In addition, mobile robots need to recognise structures like objects and natural landmarks from on-board sensor data. Once these natural landmarks are detected, they are matched with *a priori* known information of the environment to determine the pose of the mobile robots in order to perform their tasks.

A mobile robot localisation system based on EKF data fusion to localise the mobile robot in the orchard is presented in Section 6.5. Preliminary tests to verify the performance of the localisation algorithm in the simulated environment are presented in Section 6.7. In addition, Section 6.8 presents the experimental results of the localisation algorithm evaluation in the real orchard.

6.2 Mobile robot localisation types

According to Christiansen (2011), localisation problem can be identified in different cases as following:

- Local localisation: The case when the initial position of the mobile robot is known. The task is to keep tracking of the mobile robot position while the mobile robot is moving in the environment.
- Global positioning: The case when the initial position of the mobile robot is unknown. The mobile robot position is determined without any initial

reference. Hence, there can be multiple beliefs about the current position.

- Kidnapped robot: The case when the mobile robot is stolen from its current position and placed in other position rather than its last known position. In this case, the robot needs to recognise that it has been kidnapped and determine its new position.

In this study, local localisation will be implemented since the initial position of the mobile robot is known.

6.3 Sensor data fusion using Kalman filter

A key issue arises from the fact that a single sensor is usually insufficient to measure the robot pose. Therefore, the robot has to gather data from different sensors over time to determine its position and orientation. Sensor fusion methods are used to combine data measured from different sensors such that the resulting estimated position is more accurate and dependable than it would be when using each sensor individually. In practice, many sensors are used simultaneously to compute the best estimate of the robot's pose (Hellström, 2002).

The Kalman filter (KF) provides a robust mathematical method for multi-sensor data fusion in real time. It fuses the data from multiple sensors to generate the system's state estimates. This method not only estimates the past or the present state, but also can predict the future status. The standard version of the KF is designed to be used on processes that can be described by linear stochastic differential equations (Welch and Bishop, 2006). In most cases, complicated systems have nonlinear characteristics. To account for this problem, extensions to the filter have been developed. The extended Kalman filter (EKF) has been adopted to settle the problem of nonlinear filtering. EKF is a linear estimate of nonlinear systems.

There are many techniques for linearizing nonlinear functions. EKF utilises a method called the (first order) Taylor expansion. The Taylor expansion constructs a linear approximation to a nonlinear function from its value and slope. The slope is given by the partial derivative. EKF linearizes the current mean and covariance. The Taylor series is used to linearize the estimation around the current estimate using the partial derivatives of the process and measurement functions (Thrun et al., 2005).

6.4 Localisation using EKF

Along with the progress of digital computing technology, EKF has become the subject of much research and associated applications, especially in autonomous navigation research. In the mobile robot application, EKF is used to moderate the sensor noise and estimate the pose of nonlinear mobile robot systems. It can be used with different kinds of sensors under different environments as well as the combinations (Yan et al., 2009).

The implementation of EKF for mobile robot pose estimation requires the identification of two important models:

- Mobile robot motion model, f.
- Measurement model, h.

These two models are described in the following sections.

6.4.1 Mobile robot motion model

The mobile robot motion model describes the general movement of the mobile robot platform to predict the next pose of the robot using the current pose and control inputs. The motion model plays an essential role in the prediction step of the EKF.

The pose of the mobile robot operating in planner environments such as orchard environment comprises its x and y coordinates relative to the environment coordinates, along with its heading angle θ. Therefore, the pose estimate vector is represented by:

$$x_t = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{6.1}$$

The motion model used in this study assumes that the robot can be controlled through two variables, a linear velocity v and a angular velocity ω. The positive angular velocities ω induce a counterclockwise rotation (left turns) while positive linear velocities v correspond to forward motion as shown in Figure 6.1. Hence, the control input vector is represented by:

$$u_t = \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{6.2}$$

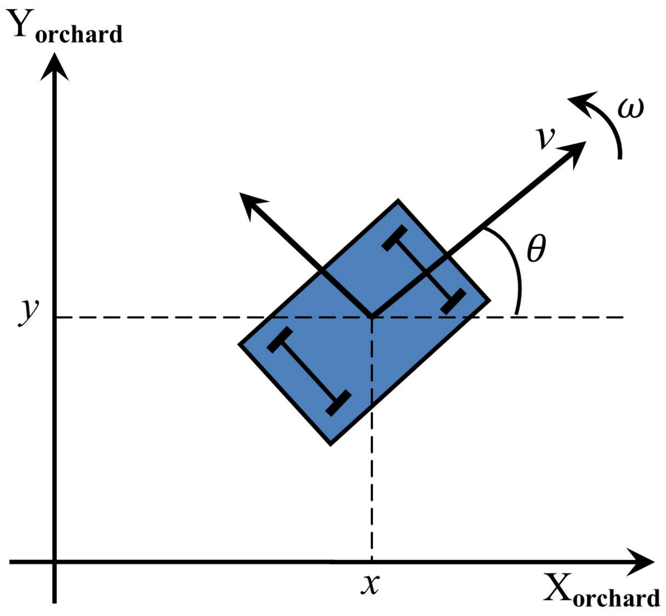


Figure 6.1: Graphic representation of the mobile robot used in this work.

The discrete time motion model used in this study is based on Dudek and Jenkin (2010) and it is represented in Equation 6.3:

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} v_t \cos(\theta_{t-1})\Delta t \\ v_t \sin(\theta_{t-1})\Delta t \\ \omega_t \Delta t \end{bmatrix} \tag{6.3}$$

where Δt is the time difference between time steps t and $(t-1)$.

6.4.2 Measurement model

The measurement model describes the formation process by which sensor measurements are generated in the physical world. The specifics of the model depends on the sensor measurements. In many robotics applications, features extracted from range sensors correspond to distinct objects in the environment. For example, in outdoor applications, they may correspond to tree trunks or other non-tree objects. In robotics, these physical objects are called landmarks to indicate that they are being used for robot navigation. The most common model for processing landmarks assumes that the sensor can measure the range r and the bearing angle ϕ of the landmark relative to the mobile robot's local coordinate frame (Thrun et al., 2005).

The measurement model h used in this study is based on Thrun et al. (2005) and it is described in Equation 6.4:

$$h(x_t, y_t, \theta_t, m_{j,x}, m_{j,y}) = \begin{bmatrix} r_t^i \\ \phi_t^i \end{bmatrix} = \begin{bmatrix} \sqrt{(m_{j,x} - x_t)^2 + (m_{j,y} - y_t)^2} \\ atan2(m_{j,y} - y_t, m_{j,x} - x_t) - \theta_t \end{bmatrix} \tag{6.4}$$

where $m_{j,x}$ and $m_{j,y}$ are the coordinates of the j-th tree in the local map m and (x_t, y_t, θ_t) represent the estimated position of the mobile robot. This equation matches the i-th detected tree from the laser scan to the j-th tree in the map at time t (Thrun et al., 2005). The measurement model is essential for the correction step in the EKF.

6.5 The developed mobile robot localisation algorithm

This study presents a mobile robot localisation solution that does not rely on GPS. The movement of the mobile robot in the orchard includes moving in straight lines along the tree rows and turning from one row to another in the headland. As the feature based EKF provides an effective method for mobile robot pose estimation, it was adopted in this study for mobile robot localisation in the orchard. The multiple measurements gathered from the robot's sensors, containing random noise are combined mathematically to generate the robot's pose estimates at that time instant.

The localisation solution uses the trees, which naturally exist in the orchard, as landmarks to correct the pose estimated by the EKF. This localisation algorithm will be used later as an input for the autonomous navigation of the mobile robot in the orchard. The two main steps of EKF used in this study are described in detail in Sections 6.5.1 and 6.5.2

6.5.1 Prediction step of EKF

This step starts by placing the mobile robot in a known starting pose in the orchard which is used as an initial pose in the prediction step of EKF. This step relies on the motion model of the mobile robot and the control inputs (linear velocity and angular velocity). In this study, the linear velocity is computed using the odometer data, while the angular velocity is obtained from the IMU measurements. Only the angular velocity in yaw from IMU was used in implementing the localisation algorithm.

The prediction step is responsible for projecting the current pose and error covariance estimates ahead in time from time step $t-1$ to step t. It predicts the pose forward in time ($\hat{x}_t^-$) using the estimated pose $\hat{x}_{t-1}$ and the control input u_{t-1} in the motion model f. The pose prediction step is presented in Equation 6.5. Equations (6.5) and (6.6) are based on Welch and Bishop (2006):

$$\hat{x}_t^- = f(\hat{x}_{t-1}, u_{t-1}) \tag{6.5}$$

The error covariance matrix P_t^- in the prediction step is calculated using Equation 6.6:

$$P_t^- = A_t P_{t-1} A_t^T + W_t Q_t W_t^T \tag{6.6}$$

where A_t is the motion model Jacobian matrix for states and it is derived by partial differentiation of the process model with respect to the state vector as follows:

$$A_t = \frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} & \frac{\partial f_x}{\partial \theta} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} & \frac{\partial f_y}{\partial \theta} \\ \frac{\partial f_\theta}{\partial x} & \frac{\partial f_\theta}{\partial y} & \frac{\partial f_\theta}{\partial \theta} \end{bmatrix} \tag{6.7}$$

Based on the mobile robot motion model presented in Equation 6.3, the motion model Jacobian matrix for states can be calculated from Equation 6.8. The Jacobian matrix A_t was evaluated using the predicted values of the state variables.

$$A_t = \begin{bmatrix} 1 & 0 & -v_t \sin(\theta_{t-1})\Delta t \\ 0 & 1 & v_t \cos(\theta_{t-1})\Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{6.8}$$

The W_t in Equation 6.6 represents the motion model Jacobian matrix for noise, and it is obtained by computing the partial derivatives of the process model with respect to the control inputs as in the following equation:

$$W_t = \frac{\partial f}{\partial u} = \begin{bmatrix} \frac{\partial f_x}{\partial v} & \frac{\partial f_x}{\partial \omega} \\ \frac{\partial f_y}{\partial v} & \frac{\partial f_y}{\partial \omega} \\ \frac{\partial f_\theta}{\partial v} & \frac{\partial f_\theta}{\partial \omega} \end{bmatrix} \tag{6.9}$$

The result of determining the derivatives of W_t in Equation 6.9 is shown in Equation 6.10. The Jacobian matrix W_t was evaluated using the predicted values of the state variables.

$$W_t = \begin{bmatrix} \cos(\theta_{t-1})\Delta t & 0 \\ \sin(\theta_{t-1})\Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \tag{6.10}$$

The Q_t is the process noise covariance matrix that describes the noise in the linear velocity and angular velocity. Q_t was determined prior to the implementation of the algorithm and kept the same for all tests. Q_t is determined as follows:

$$Q_t = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\omega^2 \end{bmatrix} \tag{6.11}$$

where σ_v and σ_ω are the standard deviation of the linear and angular velocities respectively.

6.5.2 Correction step of EKF

This step depends on an *a priori* map, measurement model and the measurement inputs (range and bearing angle) measured by the laser scanner. The correction step is responsible for correcting the projected pose and error covariance estimates. The localisation algorithm uses the trees, which naturally exist in the orchard, as landmarks to correct the estimated pose. The Kalman gain K_t is first calculated from Equation 6.12. Equations (6.12), (6.17) and (6.18) are based on Welch and Bishop (2006).

$$K_t = P_t^- H_t^T (H_t P_t^- H_t^T + V_t R_t V_t^T)^{-1} \tag{6.12}$$

where H_t is the measurement model Jacobian matrix for states. It relates the states to the measurements and it represents the partial derivatives of the measurement vector with respect to state vector as in Equation 6.13:

$$H_t = \frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial r_t^i}{\partial x} & \frac{\partial r_t^i}{\partial y} & \frac{\partial r_t^i}{\partial \theta} \\ \frac{\partial \phi_t^i}{\partial x} & \frac{\partial \phi_t^i}{\partial y} & \frac{\partial \phi_t^i}{\partial \theta} \end{bmatrix} \tag{6.13}$$

The result of determining the derivatives of H_t in Equation 6.13 is shown in Equation 6.14. The Jacobian matrix H_t was evaluated using the predicted values of the state variables.

$$H_t = \begin{bmatrix} -\frac{m_{j,x}-x_t}{\sqrt{q}} & -\frac{m_{j,y}-y_t}{\sqrt{q}} & 0 \\ \frac{m_{j,y}-y_t}{q} & -\frac{m_{j,x}-x_t}{q} & -1 \end{bmatrix} \tag{6.14}$$

where $q = (m_{j,x} - x_t)^2 + (m_{j,y} - y_t)^2$. The states x_t and y_t represent the mobile robot position coordinates where the laser scanner was located and installed.

The V_t in Equation 6.12 is the measurement model Jacobian matrix for noise, and it is obtained by computing the partial derivatives of the measurement model with respect to the measurement inputs as in the following equation:

$$V_t = \frac{\partial h}{\partial u} \begin{bmatrix} \frac{\partial r_t^i}{\partial r} & \frac{\partial r_t^i}{\partial \phi} \\ \frac{\partial \phi_t^i}{\partial r} & \frac{\partial \phi_t^i}{\partial \phi} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{6.15}$$

R_t is the measurement noise covariance matrix that describes the noise in the ranges and the bearing angles measured by the laser scanner. R_t was determined prior to the implementation of the algorithm and kept the same for all tests. R_t is described in Equation 6.16:

$$R_t = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\phi^2 \end{bmatrix} \tag{6.16}$$

where σ_r and σ_ϕ are the standard deviation of the range and bearing angle respectively. In this study, the values used for σ_r and σ_ϕ were 0.03 m and 0.25° respectively.

The next step is to detect the tree trunks in the mobile robot's path and search for the closest tree trunk detected by both camera and laser scanner. The range r and the bearing angle ϕ of this tree represent the actual measurements $z_t = [r, \phi]^T$ that will be used in the correction step. In order to relate these measurements to an actual tree in the orchard, the map m of the tree coordinates was used from which the robot can look up tree positions when this is needed in the localisation procedure. The algorithm will then associate the detected tree to the closest tree in the orchard map and return its coordinates to be used in the measurement model h to determine the estimated range and bearing angle between the tree and the mobile robot.

The corrected pose estimate $\hat{x}_t$ is determined based on the difference between the collected actual measurement z_t and the estimated measurements that are computed through the measurement model h as follows:

$$\hat{x}_t = \hat{x}_t^- + K_t(z_t - h(\hat{x}_t^-)) \tag{6.17}$$

The final step is to obtain the corrected error covariance estimate P_t matrix from Equation 6.18:

$$P_t = (I - K_t H_t)P_t^- \tag{6.18}$$

The EKF is a recursive estimation process. The updated pose and error covariance matrix are used to predict the new estimates in the next time step. This recursive nature is one of the very appealing features of the EKF. Figure 6.2 represents the flowchart of the localisation algorithm developed in this study.

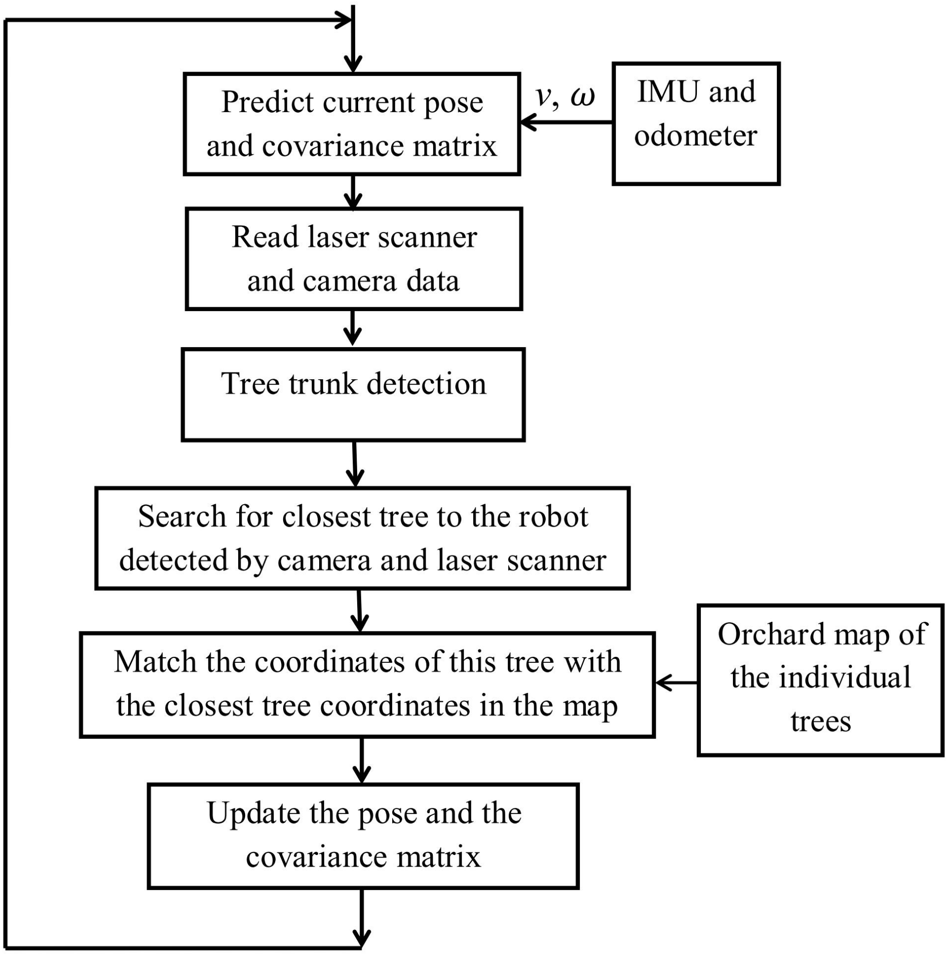

Predict current pose and covariance matrix
v, ω
IMU and odometer
Read laser scanner and camera data
Tree trunk detection
Search for closest tree to the robot detected by camera and laser scanner
Match the coordinates of this tree with the closest tree coordinates in the map
Orchard map of the individual trees
Update the pose and the covariance matrix

Figure 6.2: The developed localisation algorithm using EKF.

6.6 Determination of position accuracy

To evaluate the performance of the developed localisation algorithm, the error between the ground truth positions and the estimated positions from EKF in x and y coordinates were computed at each time step t using Equation 6.19 and Equation 6.20 respectively. The ground truth positions (x_{gt}, y_{gt}) were determined using the RTK-GPS.

$$E_x(t) = x_{gt}(t) - x(t) \tag{6.19}$$

$$E_y(t) = y_{gt}(t) - y(t) \tag{6.20}$$

The position error was also calculated through the Euclidean distance (D_E) between the ground truth position (x_{gt}, y_{gt}) and the estimated position (x, y) as follows:

$$D_E = \sqrt{(x_{gt} - x)^2 + (y_{gt} - y)^2} \tag{6.21}$$

The estimated position accuracy required for the mobile robot when moving along the row to implement tree inspection tasks is approximately ±0.15 m in x and y coordinates and 0.2 m Euclidean distance. For headland turns, the estimated position accuracy is ±0.25 m in x and y coordinates and 0.3 m Euclidean distance. The estimated heading of the mobile robot was also compared with the heading determined from the RTK-GPS ground truth position (x_{gt}, y_{gt}) to determine the heading error.

6.7 Preliminary tests of the localisation algorithm in the simulated environment

Preliminary tests were conducted using the CoroWare Explorer platform with its on-board sensors in the simulated environment to evaluate the performance of the localisation algorithm. The simulated environment in this test consisted of four simulated tree rows with each row having five simulated tree trunks. The mobile robot was remotely controlled to follow the midway between the simulated tree rows with semi-circle headland turns. This test was repeated three times from the same starting position. Table 6.1 summarises the results of the position and heading errors of the three replicates of this path. From Table 6.1, the average of the RMS of the E_x, E_y and D_E between the ground truth and the estimated positions for the three replicates were 0.067 m, 0.053 m and 0.086 m respectively, whilst the average of the RMS of the heading error was 2.86°.

Table 6.1: The results of the RMS of the D_E and the heading error of the three replicates in the simulated environment.

	Whole path RMS error			
Replicate	E_x (m)	E_y (m)	D_E (m)	Heading error (degree)
I	0.063	0.052	0.017	2.75
II	0.071	0.057	0.091	3.16
III	0.068	0.049	0.084	2.68

6.8 Experimental tests of the localisation algorithm in the orchard

The localisation algorithm using EKF was also tested in the selected area of the orchard by moving the mobile robot in different paths between the tree rows with different headland turns. The movement of the mobile robot was remotely controlled between the tree rows to collect the required data. In this study, the mobile robot must have the capability to execute different paths as this is necessary for undertaking different tree inspection tasks. The following are the main executed paths:

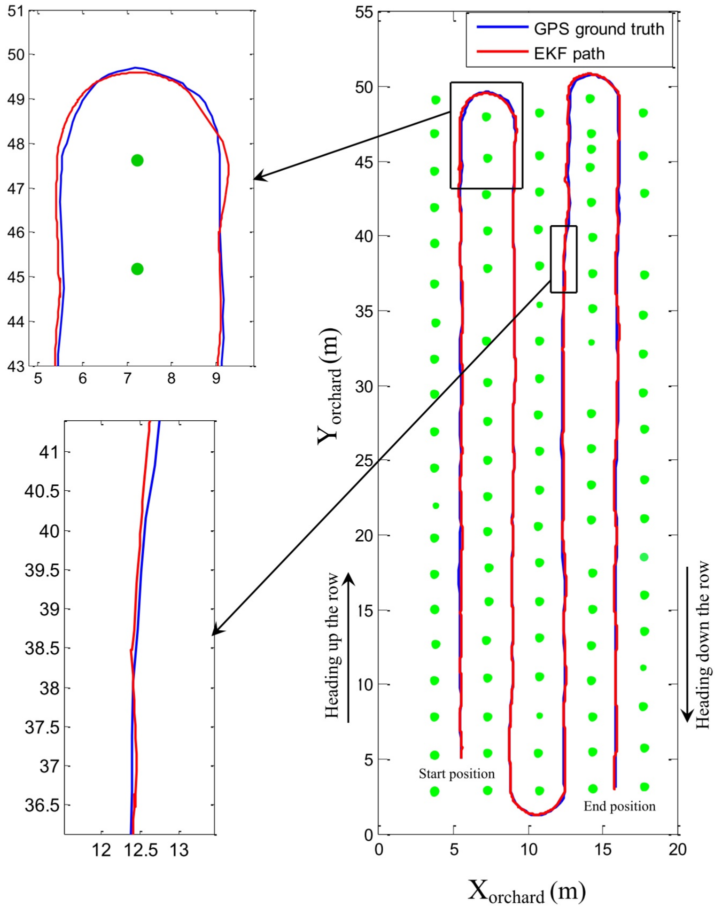
- Moving midway between tree rows.
- Moving close to a row.
- Moving between trees in a row.

The mobile robot was placed in a known starting pose which is used as the initial pose for the EKF algorithm. The starting position of the mobile robot is known by relating the robot position to the developed orchard map. The linear velocity of the mobile robot was set to approximately 0.6 m/s for straight line movement along the row and 0.3 m/s for the headland turn. The mobile robot knows its position in the orchard while moving, by relating its estimated x and y positions to the map. Extra information regarding the row number and the closest tree to the robot position could be obtained from the x and y positions of the mobile robot. The x-coordinate of the robot position relates to the row number, whilst the y-coordinate relates the robot position to the closest tree in the row. Each of the above three paths were repeated using either semi-circle or right angle headland turns. To determine the accuracy of the localisation algorithm, the estimated positions of the three main paths were determined and compared with the ground truth positions acquired from the RTK-GPS.

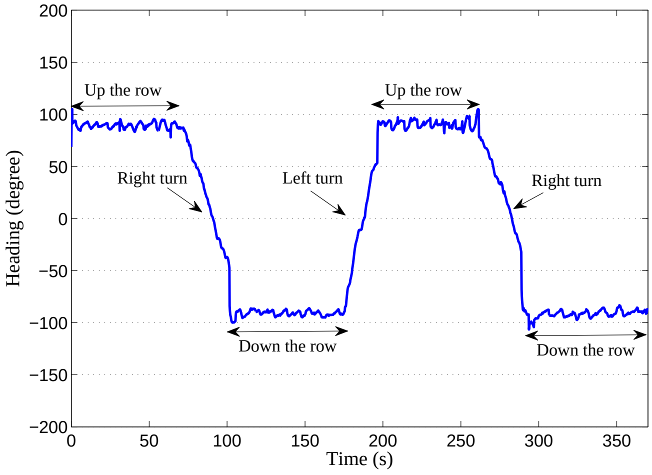
Heavy tree canopies might weaken and attenuate the RTK-GPS signal over the travel paths. This attenuation can make it difficult for the RTK-GPS receiver to track the signal from the satellites. Although the selected orchard does not have trees with a heavy canopy, there were RTK-GPS signal dropouts in several locations during the tests which may negatively affect the collection of complete path data. Hence, multiple replicates for each path were carried out on several days to increase the number of paths available for analysis and to overcome data loss due to the RTK-GPS signal dropout. Several path replicates were discarded due to the signal dropout and only paths with complete RTK-GPS position data were utilised to evaluate the performance of the localisation algorithm.

6.8.1 Estimated position and heading for different paths with semi-circle turn

The first test was achieved by moving the mobile robot from a known starting location midway between tree rows and performing a semi-circle turn from one row to another. This path is suited to tree inspection tasks for the trees in the left and right sides of the mobile robot at the same time. The robot can recognise the end of the row by comparing its y-coordinate with the y-coordinate of the last tree in that row. When the mobile robot approaches from the last tree in the row, no more trees are viewed by the camera. Therefore, the last tree in the row is used in the correction step until reaching the end of the row and during the semi-circle turn since this tree is still detected by the laser scanner. Figure 6.3a shows the estimated path and the ground truth logged using RTK-GPS. The red path shows the EKF localisation estimate of the robot positions, while the blue path shows the corresponding ground truth values. The green dots show the positions of the trees in the orchard map. Figure 6.3b shows the estimated heading of the mobile robot for the same path.



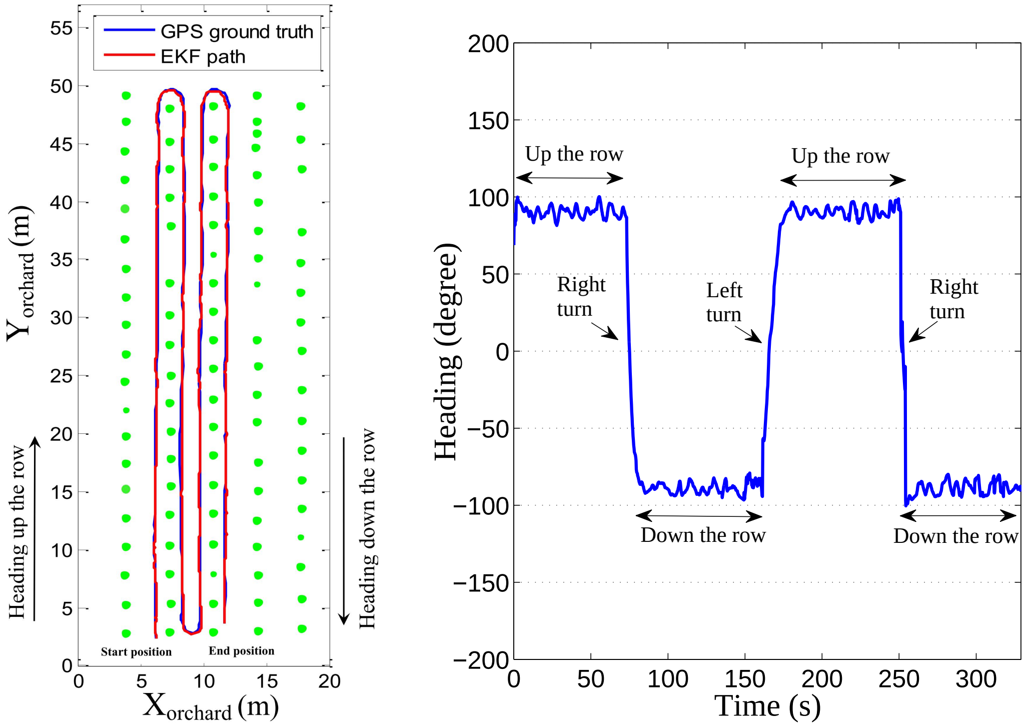
(a) The estimated path with ground truth values.



(b) The estimated heading of the mobile robot.

Figure 6.3: The estimated path and heading of the mobile robot for the midway movement between tree rows with semi-circle headland turns.

In the second test, the mobile robot was moved approximately 1 m laterally from the trees in the row. One advantage of using a small robot platform is that it can navigate close to the row without damaging anything. In addition, moving close to the row is more helpful for some inspection, pruning and harvesting tasks that need the mobile robot to move close to the row. Figure 6.4 shows the estimated path and heading angle of the mobile robot for this path.

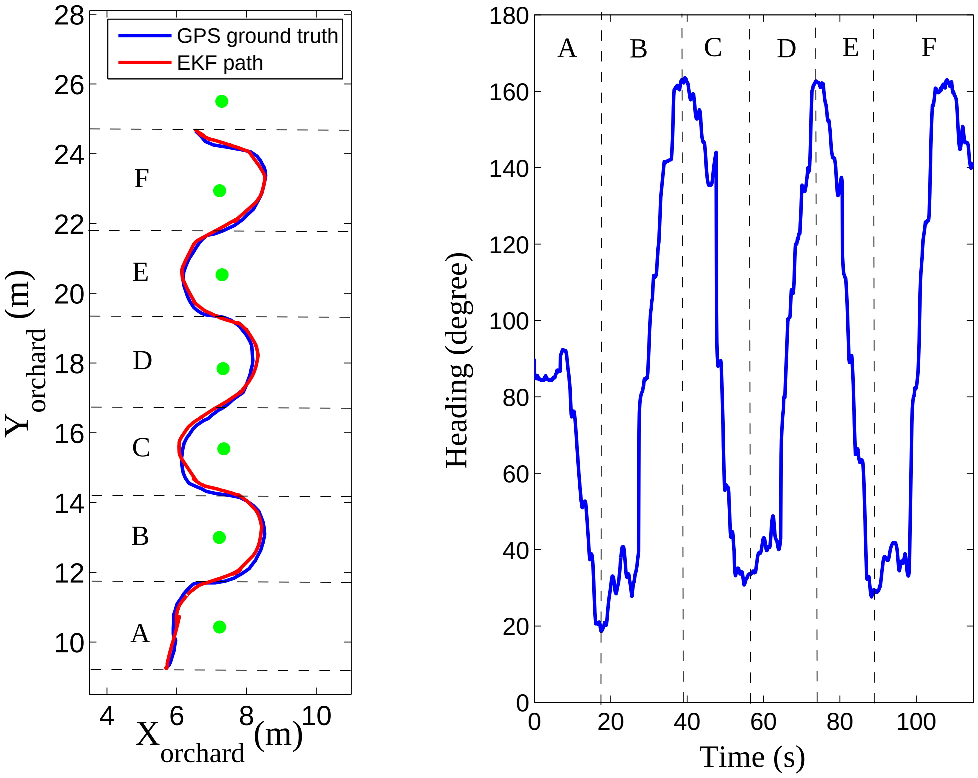


(a) Estimated path.

(b) Estimated heading angle.

Figure 6.4: Estimated path and heading angle of the mobile robot when moving close to the rows with semi-circle headland turns.

The third test was achieved by moving the mobile robot in semi-circles between trees in the row. Another benefit of using a small robot platform is that its small size provides easy maneuverability around trees in the row. This path is useful for inspecting the whole tree from different angles and also for pruning and harvesting tasks. Figure 6.5 shows the estimated path and heading angle of the robot for this path.



(a) Estimated path.

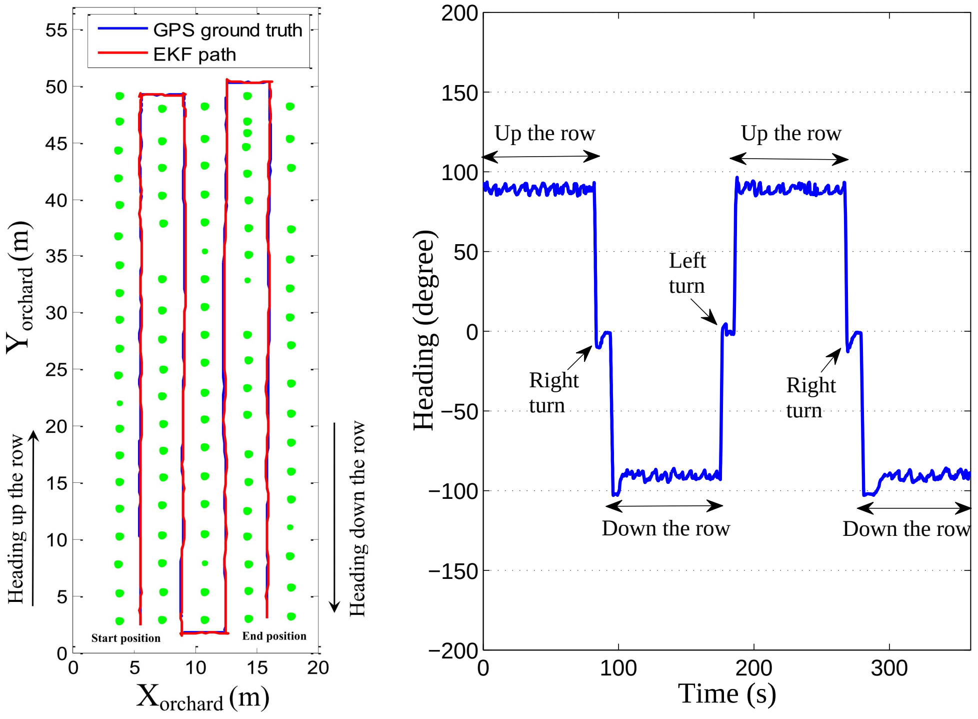
(b) Estimated heading angle.

Figure 6.5: Estimated path and heading angle when moving the mobile robot in semi-circles between trees in the row.

6.8.2 Estimated position and heading for different paths with right angle turn

A further advantage of using a small mobile robot is that it can execute a sharp right turn easily with less soil damage compared to large agricultural vehicles. In addition, the right angle turn requires less space than the semi-circle turn. When the mobile robot reaches the end of the row, it starts executing the right angle turn (i.e. the headland turning) by moving forward with the current heading for a certain distance such that the last tree in the row is still covered by the 270° scanning angle of the laser scanner. Then the robot executes the first sharp right angle turn and moves forward with a heading perpendicular to the row heading for a certain distance depending on the inter-row distance. After that, the robot performs the second sharp right angle turn and continues moving forward until entering the next row.

The previous three tests were repeated with right angle turns. The estimated path and heading angle of the mobile robot when moving in the midway between the tree rows and when turning around trees in the row with right angle turns are depicted in Figure 6.6 and Figure 6.7 respectively. From Figure 6.7a, the triangles shapes when the platform turns $+90$ or -90 degrees are expected to be due to the high slippage of the wheels when the mobile robot (with skid steering) executes a sharp right angle turn.



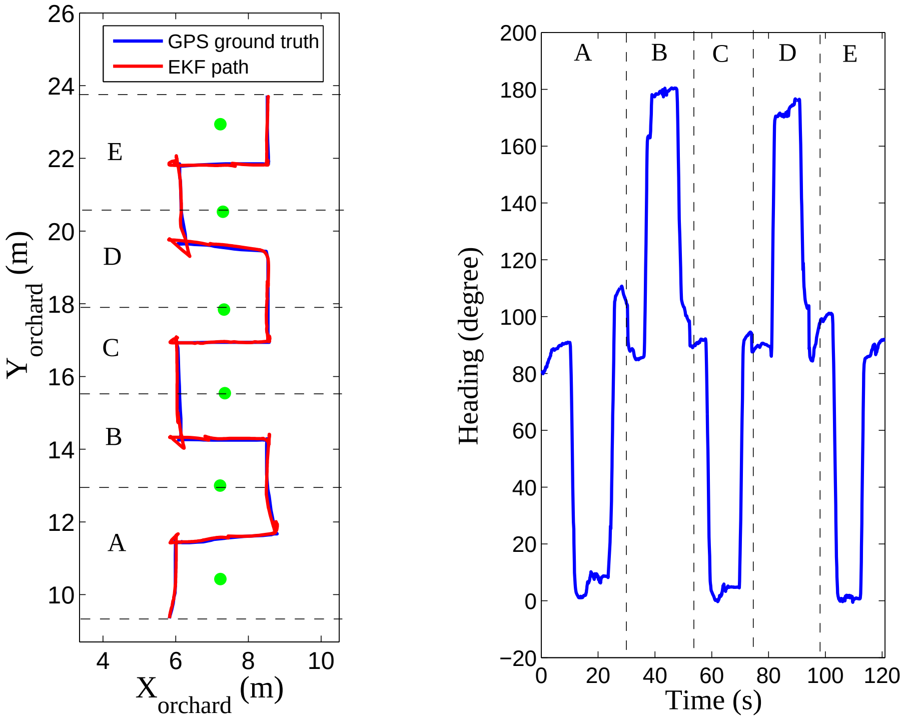
(a) Estimated path.

(b) Estimated heading angle.

Figure 6.6: Estimated path and heading angle of the mobile robot when moving between rows with right angle headland turns.

6.8.3 Position and heading errors for different paths

The localisation algorithm was evaluated against the logged RTK-GPS positions. Figure 6.8 illustrates the E_x, E_y and D_E when the mobile robot moves in the midway between tree rows with semi-circle turns, whilst Figure 6.9 depicts the heading error for the same path. From Figure 6.8, it can be seen that there is no significant difference between the amplitude of the error in x direction and y direction with maximum of ± 0.12 m when moving along the row and ± 0.15 m during the headland turns. The maximum value of D_E was 0.16 m when moving along the row and 0.2 m during the headland turns. This proves that the localisation algorithm was capable of achieving the necessary accuracy presented in Section 6.6.

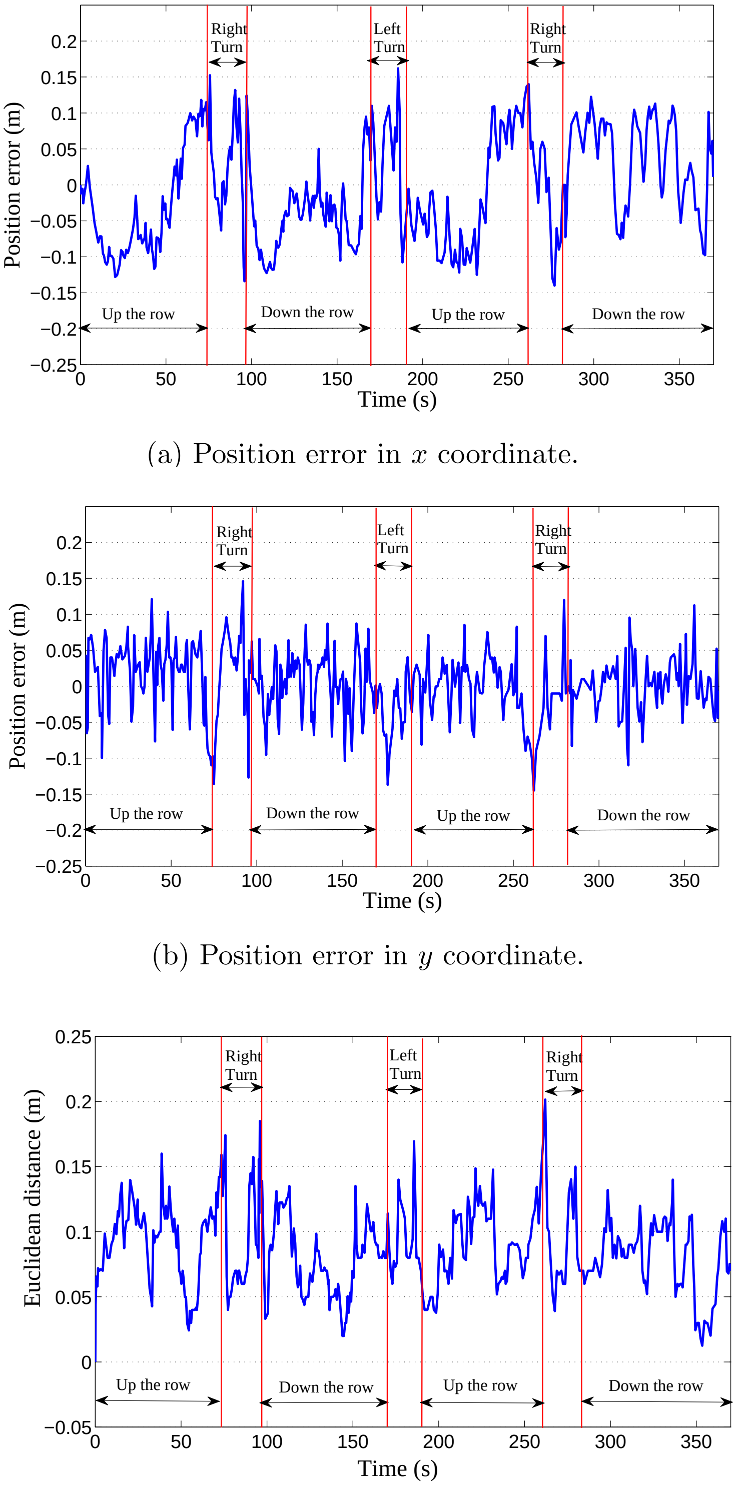


(a) Estimated path.

(b) Estimated heading angle.

Figure 6.7: Estimated path and heading angle of the mobile robot when turning around trees with right angle turn.

Table 6.2 summarises the RMS of position error based on D_E for one run of the three different paths using semi-circle headland turns. From this table, the position error when moving along the rows is less than the error in the headland turns, which is expected to be due to the high slippage of the wheels in the headland turns. From the results obtained, it can be observed that the error is inversely proportional to the turn radius.



(a) Position error in x coordinate.

(b) Position error in y coordinate.

(c) Euclidean distance.

Figure 6.8: The position errors when the robot moves midway between rows with semi-circle headland turns.

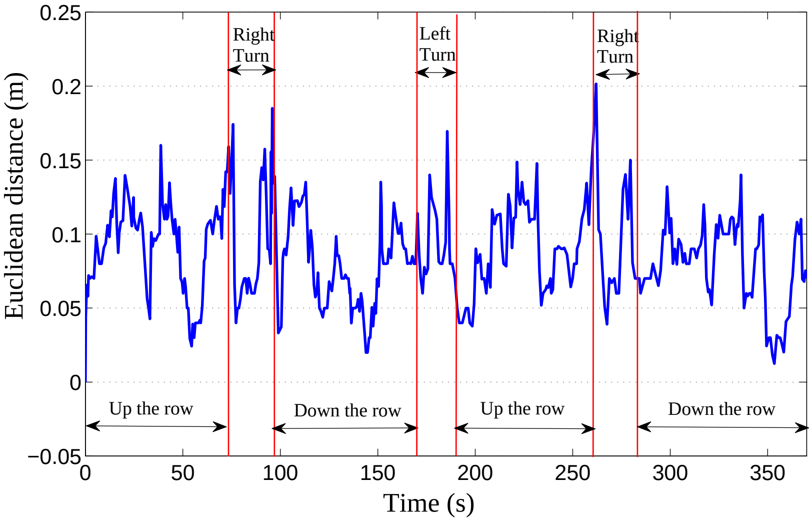


Figure 6.9: The heading error when the robot moves midway between rows with semi-circle headland turns.

Table 6.2: The results of the RMS of D_E for one run of the three different paths with semi-circle headland turns.

Localisation test type	RMS of D_E (m)		
	Along the row	Headland turns	Whole path
Midway between rows	0.089	0.104	0.092
Close to the rows	0.094	0.132	0.102
Turning around trees	-	0.116	0.116

Each of the three main paths was repeated three times with the same starting position with semi-circle turns and right angle turns to verify the performance of the designed algorithm. Table 6.3 summarises the results of the RMS of position and heading errors for the whole path of the three replicates. The results in Table 6.3 indicate that the performance of the localisation algorithm was similar for the three replicates of each path.

Table 6.3: The results of the RMS of the position and heading errors of the three replicates of each path.

			Whole path RMS error			
			E_x	E_y	D_E	Heading
Localisation test type	Replicate	Date	(m)	(m)	(m)	(degree)
Midway between rows with semi-circle turn	I	September, 27	0.075	0.054	0.092	2.25
	II	October, 18	0.081	0.058	0.101	3.06
	III	November, 1	0.077	0.06	0.097	2.78
Close to the rows with semi-circle turn	I	September, 27	0.081	0.062	0.102	3.16
	II	October, 18	0.081	0.058	0.115	3.65
	III	November, 1	0.078	0.068	0.104	3.2
Turning around trees with semi-circle turn	I	September, 27	0.086	0.078	0.116	3.78
	II	October, 18	0.094	0.085	0.127	4.02
	III	November, 1	0.082	0.08	0.114	3.8
Midway between rows with right angle turn	I	September, 28	0.075	0.056	0.096	2.52
	II	October, 19	0.075	0.056	0.093	2.34
	III	November, 2	0.083	0.061	0.103	3.25
Close to the rows with right angle turn	I	September, 28	0.082	0.069	0.107	3.48
	II	October, 19	0.092	0.075	0.119	3.75
	III	November, 2	0.088	0.067	0.11	3.28
Turning around trees with right angle turn	I	September, 28	0.089	0.083	0.122	3.85
	II	October, 19	0.083	0.08	0.115	3.50
	III	November, 2	0.094	0.09	0.13	4.21
		Average	0.08	0.07	0.103	3.32

6.9 General discussion

The developed localisation algorithm using EKF has been shown to achieve a maximum position error of ±0.12 m when moving along the row and ±0.15 m during the headland turns. The maximum value of D_E was 0.16 m when moving along the row and 0.2 m during the headland turns. This indicates that the localisation algorithm was able to achieve the required accuracy presented in Section 6.6 for localising the mobile robot along the rows and when turning from one row to another. The accuracy of the EKF localisation algorithm depends on the accuracy of the position of the trees in the map, since the EKF algorithm uses the tree positions from the map in the correction step to correct the estimated position of the mobile robot. However, in further more extensive trials, the accuracy might be further analysed if a ground truth method other than the generated map is used to assess the accuracy of localisation. Localising the mobile robot in the headland turns produces a greater error than moving along the row due to the high wheel slippage during the execution of the turns. The mobile robot was capable of performing the semi-circle and the right angle turn with acceptable error.

The developed localisation algorithm shows approximately the same behaviour for all paths with slight differences in RMS errors between straight line and turn movements. According to the results obtained from Table 6.3, it is obvious that there is no significant difference between the results of the RMS of position and heading error values of the three paths when using semi-circle turns or right angle turns. However, the right angle turn is more attractive for the navigation task since the right angle turn requires less space than the semi-circle turn. The average of the RMS of E_x, E_y, D_E and heading error of the paths in Table 6.3 were 0.08 m, 0.07 m, 0.103 m and 3.32° respectively, which are acceptable to achieve localisation along the rows and the turns in the orchard.

The orchard map of the trees and non-tree objects was constructed prior to the localisation process using the RTK-GPS, IMU, camera and laser scanner. This map was constructed only once to be used as an *a priori* map for localisation. The localisation algorithm depends on the tree map and the measurements from on-board sensors without the use of the RTK-GPS. In the localisation algorithm, the RTK-GPS was only used for collecting the ground truth positions to evaluate the performance of the developed localisation algorithm.

RTK-GPS signal dropout may occur due to heavy canopies over travel paths, which block the signal from satellites. Although the selected orchard does not have trees with a heavy canopy, there were RTK-GPS signal dropouts in several places in the conducted paths. For this, multiple replicates were carried out in multiple days to overcome data loss associated with RTK-GPS signal dropout. Several conducted paths have been neglected due to signal dropout, whilst only paths with complete RTK-GPS position data were used for evaluating the performance of the localisation algorithm.

The irregular distance between the trees in the row does not affect the localisation algorithm since the localisation algorithm is independent of the distance between the trees. In addition, missing trees in the rows does not affect the performance of the algorithm because the algorithm will use the closest detected tree in the row to correct its pose in the correction step of the EKF.

Comparing with the algorithms that used EKF to localise the mobile robot only in midway between tree rows (Hansen et al., 2009; Libby and Kantor, 2011; Christiansen, 2011; Bergerman et al., 2015), the localisation algorithm developed in this study was capable of localising the mobile robot in different paths (midway between rows, close to the rows and moving around trees in the row) and different turns (semi-circle turns and right angle turns). These multiple paths are necessary to implement different agricultural tasks such as harvesting, pruning, thinning and tree inspection tasks.

6.10 Conclusion

This chapter presented a GPS-free mobile robot localisation algorithm using EKF to localise the position of the CoroWare Explorer platform in a persimmon orchard using the trees in the orchard as landmarks to correct the pose estimation. The map of the indivitual trees is used as an *a priori* map to localise the mobile robot in the orchard.

The EKF performs well and was able to keep the robot position close to ground truth. The algorithm was tested and evaluated through extensive tests for different paths and turns. By accurately modelling the noise on the on-board robot sensors using the covariance matrices Q_t and R_t, it was able to maintain a robust position without the re-tuning of any parameters between tests.

The average of the RMS of the position errors in x and y coordinates for different paths in the orchard were 0.08 m and 0.07 m respectively, whilst the average of the RMS of the Euclidean distance between the ground truth and the estimated position for the same paths was 0.103 m. The heading error was also measured for these paths and the average of the RMS of the heading error was 3.32°. These errors are acceptable while driving along the rows and when leaving the rows for headland maneuvering. The developed algorithms rely on the on-board sensors of the mobile robot only without adding any artificial landmarks in the orchard.

Chapter 7

Conclusions and Future work

This thesis explored the use of a small mobile robot with on-board sensors to detect individual trees and non-tree objects in the tree rows of an orchard and determine the position of these trees and non-tree objects to generate a local map of the orchard. The constructed map of the individual trees was used as an *a priori* map for mobile robot localisation in different paths and turns required for tree inspection tasks in the orchard.

In this chapter, conclusions with respect to the achievement of the objectives set out in Section 1.5 are presented. Potential applications and recommendations for further development of the research are also presented.

7.1 Achievement of objectives

This section provides a brief overview of the achievement of the objectives stated early in Section 1.5 as follows:

Objective 1. Develop a tree trunk detection algorithm that can detect trees and discriminate between trees and non-tree objects using camera and laser scanner *data fusion.*

A novel tree trunk detection algorithm that has the capability of detecting trees and discriminating between trees and non-tree objects in an orchard was developed and evaluated in this research work (Chapter 4). Attempts have been made in researching the best method to consistently detect trees with natural variation present in the orchard environment. It was observed that fusing camera and 2D laser scanner data provides better detection performance than using each sensor separately. The tree trunk was selected as the suitable part of the tree to be detected since the tree trunks in the selected orchard for this study can be distinguished from the leaves and other non-tree objects. The tree trunk detection algorithm first detects the tree trunk using the laser scanner to determine its width and then projects the edge points of the tree trunk to the image plane to construct a region of interest with the required feature for colour and edge detection. The algorithm was tested in the simulated environment and real orchard to verify its performance through experimental tests. Automatic adjustment of the algorithm parameters was achieved after each test which was shown to improve the detection accuracy by 5%.

Objective 2. Develop a method for constructing a local orchard map of the individual trees and non-tree objects using the on-board mobile robot sensors to localise the mobile robot in the orchard and to enable the individual tree monitoring and inspection.

The data acquired by the mobile robot sensors was used to determine the 2D position of the detected trees and non-tree objects to construct a local-scale map of the orchard (Chapter 5). This map is used by the mobile robot to efficiently estimate its position and orientation while moving between tree rows in the orchard. The mapping results of the simulated environment and the real orchard indicated that the constructed map can be reliably used for mobile robot localisation. Mapping the orchard was done prior to the localisation process. This map was then saved to the mobile robot's on-board computer to be used for localisation.

Objective 3. Develop a data fusion algorithm to estimate the pose (position and orientation) of the mobile robot for different paths and turns in the orchard.

In this study, a data fusion algorithm based on EKF was developed to localise the mobile robot in different paths (midway between rows, close to the rows and moving around trees in the row) and different turns (semi-circle turns and right angle turns) required for tree inspection tasks (Chapter 6). Trees were used as natural landmarks for mobile robot localisation, together with the constructed map of the individual trees and the measurements from the mobile robot's on-board sensors (camera, laser scanner, odometer and IMU) to enhance in-row localisation accuracy. The results of the estimated position and heading was evaluated against the data acquired from the RTK-GPS through extensive tests to determine the position accuracy. The localisation algorithm was capable of executing the suggested paths and turns with acceptable accuracy.

Objective 4. Evaluate of the performance of the developed algorithms through extensive experimental tests using a small mobile robot platform under different illumination conditions.

The developed algorithms were evaluated through a range of outdoor experimental tests in the simulated environment and real orchard. The final findings obtained were summarised as follows:

1. The Detection Algorithm A was able to detect all the simulated tree trunks and objects in the simulated environment. In the real orchard, Detection Algorithm B was successful in detecting the tree trunks and discriminating between different tree trunks (regular and thin tree trunks) and different non-tree objects (small and big posts and tree supports) with a detection accuracy of 97% at the end of Test 4.

2. The orchard map construction method successfully localised the simulated tree trunks and the objects in the simulated environment with RMS position

accuracy of 20 mm and 18 mm in x and y coordinates respectively and RMS Euclidean distance of 27 mm. These results indicated that the proposed mapping algorithm provides a map with adequate accuracy to be used for mobile robot localisation and navigation. For the selected area of the orchard, the mapping algorithm successfully localised all the trees and non-tree objects. In the absence of ground truth methods (e.g. measuring of tree position using surveying methods), the constructed map of the selected area of the orchard was assumed to be an accurate representation of the orchard and consistent with the visual verification, and can be reliably used as an *a priori* map for localisation and navigation.

3. The localisation algorithm based on EKF was evaluated against the RTK-GPS positions for different paths and turns. In the simulated environment, the average of the RMS of the position error in x, y coordinates and Euclidean distance between the ground truth and the estimated positions were 0.067 m, 0.053 m and 0.086 m respectively, whilst the average of the RMS of the heading error was 2.86°. For the paths executed in the real orchard, the average of the RMS of the position error in x, y coordinates and Euclidean distance were 0.08 m, 0.07 m and 0.103 m respectively, whilst the average of the RMS of the heading error was 3.32°. These errors are acceptable while driving along the rows and when executing headland turns, and are adequate for both autonomous mobile robot navigation and for potential inspection tasks in orchard applications.

From these results, it can be concluded that the proposed algorithms for tree detection, orchard map construction and mobile robot localisation have answered the research questions presented in Section 1.4, and met the objectives stated in Section 1.5.

7.2 Applications of the research work

In recent years, the potential applications for autonomous mobile robots in agricultural environments have been increasing rapidly. Tree detection, orchard mapping and mobile robot localisation are fundamental tasks in many potential agricultural applications. The potential applications of a small mobile robot with the developed algorithms are presented below.

7.2.1 Tree trunk diameter measurement

Accurate estimation of tree trunk diameter is important in many applications. In this study, the tree trunk diameter can be determined from the laser scan data from Figure 7.1 using the following equation:

$$Diameter = 2R = 2r_c \sin(\frac{\Delta\phi}{2}) \tag{7.1}$$

where R is the radius of the tree trunk, r_c is laser range between the laser position on the mobile robot and the tree trunk centre determined from Equation 5.1 and $\Delta\phi$ represents the difference between ϕ_1 and ϕ_2 as explained in Section 5.2.

The developed algorithms for tree trunk detection and individual tree mapping can be upgraded to estimate and map the tree trunk diameter of the individual trees and the mean of the tree trunks' diameter for a specific row or specific area in the orchard. In addition, these algorithms provide automatic detection of the thin trees and regular trees and determination of their locations in the orchard. The diameter of the tree trunk is a good predictor for tree trunk geometry and many other features of interest (e.g. trunk volume, canopy volume and trunk cross-sectional area).

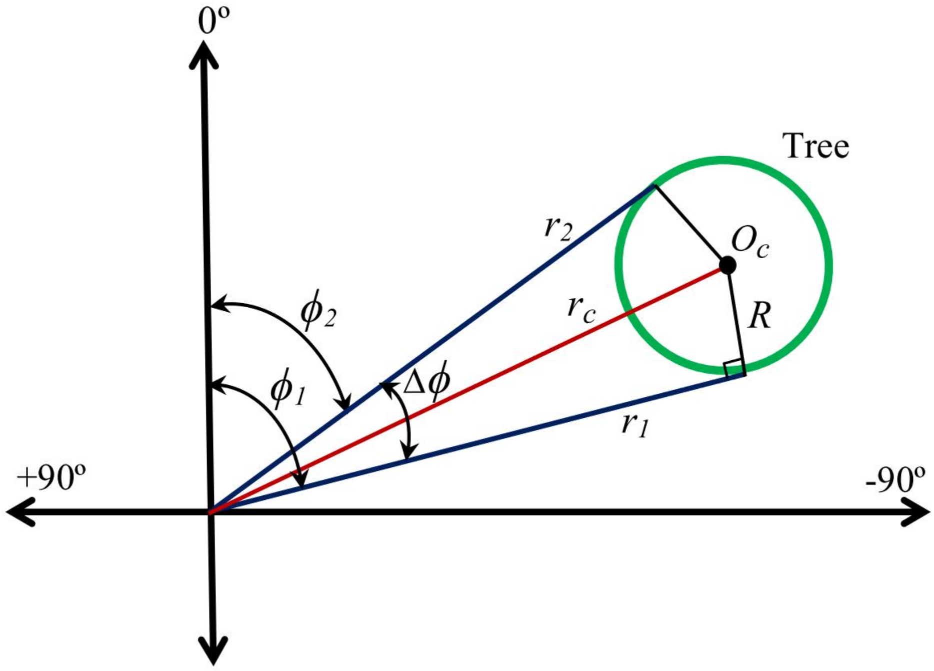


Figure 7.1: Determination of the tree trunk diameter from laser scan data.

7.2.2 Yield mapping and estimation

The developed tree trunk detection and the determination of the position of individual trees in the orchard is essential to numerous horticultural studies and management practices. Position information of individual trees are essential for the development of yield and other management maps of the orchard. These tree positions are also used for surveying tree populations and directing spray practices. Yield mapping techniques are useful tools for orchard tree production management. The yield map contains spatial information used to quantify the productivity of specific areas within the orchard. The main key to creating yield maps in an orchard lies in the ability to accurately locate the individual position of every tree in the field (Heidman and Rosa, 2008).

Trees are important crops that can produce for several years and require numerous amounts of inputs during their lifespan. These inputs have the potential to be optimised to increase crop production efficiency, reduce costs, and maximise profits.

Positional and yield data can be gathered per tree for monitoring tree yield over a number of years (Heidman and Rosa, 2008). A record of tree productivity per tree over several years provides the opportunity for a farmer to more efficiently manage inputs and remove trees that are not productive (Rains et al., 2002).

Another useful outcome is that the developed algorithm has the potential capability to record and map the trunk width and diameter of the individual trees in the orchard which can be used for yield estimation of the individual trees in the row, individual rows or a specific area in the orchard. The size of a tree is usually expressed as trunk cross-sectional area which is the most common surrogate measurement to determine the tree size and indirectly, the capacity of a tree to produce fruits (Treder et al., 2010).

7.2.3 Tree inspection and growth monitoring

The inspection task in orchards is normally undertaken by the farmer who is supposed to cross every row of the orchard in order to have up-to-date information about the planted trees. In addition, routine agricultural tasks can result in fatigue in labour under long hours of repetitive operations. Farmers benefit from mobile inspection robots as they can undertake regular inspection tasks whilst being more robust to environmental conditions than humans (Ortiz and Olivares, 2006). Mobile robots have the potential for substituting human inspection routine for individual trees according to the farmer's demands. The time and labour saving that the mobile robot will provide for the farmer will enable the farmer to spend more resources on other aspects of orchard management (Jeon et al., 2009). Small autonomous mobile robots are more suited to individual plant care task than conventional agricultural systems, and are ideal when dealing with monitoring tasks as they are more gentle on the crops and the ground than tractors. This is due to the lower weight compared to a tractors, causing less soil compaction. The degree of soil compaction is important to consider, especially when dealing with monitoring and mapping as this is often performed multiple times throughout the year (Blackmore and Griepentrog, 2002; Pedersen et al., 2005).

The mobile robot used in this study with the developed algorithm can survey the orchard instead of humans, providing the farmer with the required important information about the trees by relating the position of the mobile robot to the constructed map of the individual trees. A mobile robot of such size could meet a major requirement for commercial orchard inspection tasks such as observing the state of flowering, crop development or damage following a storm event by carrying a task-specific camera and communication system. The images captured by the mobile robot camera will help monitoring plants' growth and informing the farmer about the situation of the orchard (plants' health and diseases). For example, the data collected from the robot sensors can provide information that helps the farmer monitor individual trees or a whole tree row. This also helps to identify some problems related to tree growth (e.g. soil, irrigation problems, fruit diseases and fruit development).

The capability of the mobile robot (used in this study) to execute different paths (midway between tree rows, close to the rows and turning around the trees in the row) helps the mobile robot to implement different tree inspection and growth monitoring tasks. The mobile robot can capture images and laser scans for different parts of the trees (in relation to known robot position) from different angles and positions, and at different growth stages. These images and laser scans can be used for later analysis (off-line processing) to extract different information such as tree trunk and canopy geometry (volume, height).

7.3 Future work

In conducting this research, several areas of further work have been identified. It is recommended that future work of this research should focus on the following points.

7.3.1 Enhancements for tree trunk detection

Further improvements could be implemented to enhance the tree trunk detection algorithm as follows:

- The accuracy of the tree trunk detection algorithm could be improved by selecting several ROI above and below the original ROI for colour detection to increase the reliability of the tree trunk colour estimation. The ability of the algorithm to discriminate between the trees and non-tree objects could be enhanced by detecting the colour of the tree canopy via the setting of another ROI in the canopy area of the tree. In addition, adding another feature such as tree trunk texture could also enhance the discrimination capability of the algorithm.
- Exploring the utility of a fuzzy logic system to distinguish between all types of trees and non-tree objects.

7.3.2 Enhancements for map construction and mobile robot localisation

Map construction method and mobile robot localisation algorithm could be further improved as follows:

- For the map construction process, the heading of the mobile robot could be determined from the laser scanner data which is expected to provide more accurate heading.

- The localisation algorithm can be modified to provide an automatic update of the orchard map. This is achieved by recording the coordinates of the trees that were not present in the current map. Then the algorithm can use these coordinates to update the map.
- Testing the localisation algorithm in different terrains such as hills and uneven ground to study the effect of the angular velocity in roll and pitch on the position accuracy.
- Investigating the ability of implementing the SLAM method in such orchard environments and addressing the challenging problems related to the processing time and the computational requirements of SLAM.
- Other ground truth methods (such as 'total station' surveying techniques) should be considered in future work as these are likely to be more accurate than RTK-GPS.

7.3.3 Implementation of autonomous navigation and tree inspection

Likewise future work of this study should also focus on the following points:

- Using the suggested algorithms to develop a path planning and autonomous navigation systems using a suitable control strategy to control the movement of the mobile robot in the orchard.
- The tree trunk detection algorithm developed in this study can be modified and used for the obstacle avoidance system to detect trees and objects and to estimate the distance between them and the mobile robot in order to achieve safe navigation.

- Examining the possibility of using the developed algorithms for potential tree inspection tasks and automatic diagnosis of plant diseases, as described in Section 7.2.3.

References

Ali, W., Georgsson, F. and Hellstrom, T. (2008), Visual tree detection for autonomous navigation in forest environment, *in* '2008 IEEE Intelligent Vehicles Symposium', Eindhoven, The Netherlands, June 4-6, 2008, pp. 560–565.

Ampatzidis, Y., Vougioukas, S., Bochtis, D. and Tsatsarelis, C. (2009), 'A yield mapping system for hand-harvested fruits based on rfid and gps location technologies: field testing', *Precision agriculture* **10**(1), 63–72.

Andersen, J. C., Ravn, O. and Andersen, N. A. (2010), Autonomous rule-based robot navigation in orchards, *in* '7th IFAC Symposium on Intelligent Autonomous Vehicles', Lecce, Italy, September 6-8, 2010, pp. 43–48.

Åstrand, B. and Baerveldt, A. (2005), 'A vision based row-following system for agricultural field machinery', *Mechatronics* **15**(2), 251–269.

Auat Cheein, F. A. and Carelli, R. (2013), 'Agricultural robotics: Unmanned robotic service units in agricultural tasks', *IEEE Industrial Electronics Magazine* **7**(3), 48–58.

Auat Cheein, F., Steiner, G., Perez Paina, G. and Carelli, R. (2011), 'Optimized EIF-SLAM algorithm for precision agriculture mapping based on stems detection', *Computers and Electronics in Agriculture* **78**(2), 195–207.

Ayala, M., Soria, C. and Carelli, R. (2008), 'Visual servo control of a mobile robot in agriculture environments', *Mechanics based design of structures and machines* **36**(4), 392–410.

Bailey, T. and Durrant-Whyte, H. (2006), ‘Simultaneous localization and mapping (SLAM): Part II’, *IEEE Robotics and Automation Magazine* **13**(3), 108–117.

Bakker, T., Wouters, H., Van Asselt, K., Bontsema, J., Tang, L., Müller, J. and van Straten, G. (2008), ‘A vision based row detection system for sugar beet’, *Computers and Electronics in Agriculture* **60**(1), 87–95.

Barawid, O., Mizushima, A., Ishii, K. and Noguchi, N. (2007), ‘Development of an autonomous navigation system using a two-dimensional laser scanner in an orchard application’, *Biosystems engineering* **96**(2), 139–149.

Behfar, H., Ghasemzadeh, H., Rostami, A., Seyedarabi, M. and Moghaddam, M. (2014), ‘Vision-based row detection algorithms evaluation for weeding cultivator guidance in lentil’, *Modern Applied Science* **8**(5), 224–232.

Benson, E., Reid, J. and Zhang, Q. (2003), ‘Machine vision-based guidance system for an agricultural small-grain harvester’, *Transactions of the ASAE* **46**(4), 1255–1264.

Bergerman, M., Maeta, S. M., Zhang, J., Freitas, G. M., Hamner, B., Singh, S. and Kantor, G. (2015), ‘Robot farmers: Autonomous orchard vehicles help tree fruit production’, *IEEE Robotics and Automation Magazine* **22**(1), 54–63.

Bienert, A. and Schneider, D. (2013), ‘Range image segmentation for tree detection in forest scans’, *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* **1**(2), 49–54.

Billingsley, J. and Schoenfisch, M. (1997), ‘The successful development of a vision guidance system for agriculture’, *Computers and electronics in agriculture* **16**(2), 147–163.

Blackmore, S. and Griepentrog, H. (2002), A future view of precision farming, *in* ‘Proceedings of the PreAgro Precision Agriculture Conference’, Muncheberg, Germany, pp. 131–145.

Blackmore, S., Stout, W., Wang, M. and Runov, B. (2005), Robotic agriculture–the future of agricultural mechanisation, *in* 'European Conference on Precision Agriculture', Vol. 5, pp. 621–628.

Bloss, R. (2008), 'Simultaneous sensing of location and mapping for autonomous robots', *Sensor Review* **28**(2), 102–107.

Bonin-Font, F., Ortiz, A. and Oliver, G. (2008), 'Visual navigation for mobile robots: a survey', *Journal of Intelligent and Robotic Systems* **53**(3), 263–296.

Bouguet, J.-Y. (2009), 'Camera calibration toolbox for Matlab', `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

Brunner, A. and Gizachew, B. (2014), 'Rapid detection of stand density, tree positions, and tree diameter with a 2D terrestrial laser scanner', *European Journal Of Forest Research* **133**(5), 819–831.

Campoy, J., Gonzalez-Mora, J. and Dima, C. (2010), Advanced sensing for tree canopy modeling and precision spraying, *in* 'ASABE Annual International Meeting', Pittsburgh, Pennsylvania, 20-23 June, 2010.

Canny, J. (1986), 'A computational approach to edge detection', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(6), 679–698.

Changyi, X., Lihua, Z., Minzan, L., Yuan, C. and Chunyan, M. (2015), 'Apple detection from apple tree image based on bp neural network and hough transform', *International Journal of Agricultural and Biological Engineering* **8**(6), 46–53.

Choi, K. H., Han, S. K., Han, S. H., Park, K.-H., Kim, K.-S. and Kim, S. (2015), 'Morphology-based guidance line extraction for an autonomous weeding robot in paddy fields', *Computers and Electronics in Agriculture* **113**, 266–274.

Christiansen, M. (2011), *Localization in orchards using Extended Kalman filter for sensor-fusion*, Master thesis, University of Southern Denmark.

Cubero, S., Alegre, S., Aleixos, N. and Blasco, J. (2015), Computer vision system for individual fruit inspection during harvesting on mobile platforms, *in* 'Precision agriculture'15', Wageningen Academic Publishers, pp. 3412–3419.

Cyganek, B. and Siebert, J. P. (2011), *An introduction to 3D computer vision techniques and algorithms*, John Wiley & Sons, Ltd.

Ding, Y., Chen, D. and Wang, S. (2011), 'The mature wheat cut and uncut edge detection method based on wavelet image rotation and projection', *African Journal of Agricultural Research* **6**(11), 2609–2616.

Dudek, G. and Jenkin, M. (2010), *Computational principles of mobile robotics*, Cambridge University Press.

Durrant-Whyte, H. and Bailey, T. (2006), 'Simultaneous localization and mapping: Part I', *IEEE Robotics and Automation Magazine* **13**(2), 99–108.

Eaton, R., Katupitiya, J., Siew, K. and Howarth, B. (2010), 'Autonomous farming: modelling and control of agricultural machinery in a unified framework', *International Journal of Intelligent Systems Technologies and Applications* **8**(1), 444–457.

Emmi, L., Gonzalez-de Soto, M., Pajares, G. and Gonzalez-de Santos, P. (2014), 'Integrating sensory/actuation systems in agricultural vehicles', *Sensors* **14**(3), 4014–4049.

English, A., Ball, D., Ross, P., Upcroft, B., Wyeth, G. and Corke, P. (2013), Low cost localisation for agricultural robotics, *in* 'Proceedings of the 2013 Australasian Conference on Robotics and Automation (ACRA2013)', Sydney, Australia, December 2-4, 2013, pp. 1–8.

Ericson, S. and Astrand, B. (2010), Row-detection on an agricultural field using omnidirectional camera, *in* '2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', Taipei, Taiwan, October 18-22, 2010, pp. 4982–4987.

Fountas, S., Blackmore, B., Vougioukas, S., Tang, L., Sørensen, C. and Jørgensen, R. (2007), Decomposition of agricultural tasks into robotic behaviours, *in* ‘Agricultural Engineering International: The CIGR Ejournal’, Vol. 9.

Gao, F., Xun, Y., Wu, J., Bao, G. and Tan, Y. (2010), Navigation line detection based on robotic vision in natural vegetation-embraced environment, *in* ‘2010 3rd International Congress on Image and Signal Processing (CISP)’, Vol. 6, IEEE, pp. 2596–2600.

Garcia-Alegre, M. C., Martin, D., Guinea, M. D. and Guinea, D. (2011), *Real-Time Fusion of Visual Images and Laser Data Images for Safe Navigation in Outdoor Environments*, Sensor Fusion-Foundation and Applications, Dr. Ciza Thomas (Ed.), InTech: http://dx. doi. org/10.5772/16690.

Gongal, A., Silwal, A., Amatya, S., Karkee, M., Zhang, Q. and Lewis, K. (2016), ‘Apple crop-load estimation with over-the-row machine vision system’, *Computers and Electronics in Agriculture* **120**, 26–35.

González, J., Blanco, J.-L., Galindo, C., Ortiz-de Galisteo, A., Fernandez-Madrigal, J.-A., Moreno, F. A. and Martínez, J. L. (2009), ‘Mobile robot localization based on Ultra-Wide-Band ranging: A particle filter approach’, *Robotics and autonomous systems* **57**(5), 496–507.

Gonzalez-Mora, J., Vallespi, C., Dima, C. S. and Ehsani, R. (2010), HLB detection using hyperspectral radiometry, *in* ‘Proceeding of the 10th International Conference on Precision Agriculture’, July 18-21, 2010.

Gottschalk, R., Burgos-Artizzu, X., Ribeiro, A. and Pajares, G. (2010), ‘Real-time image processing for the guidance of a small agricultural field inspection vehicle’, *International Journal of Intelligent Systems Technologies and Applications* **8**(1), 434–443.

Griepentrog, H., Andersen, N., Andersen, J., Blanke, M., Heinemann, O., Nielsen, J., Pedersen, S., Ravn, O., Madsen, T. and Wulfsohn, D. (2009), ‘Safe and reliable: Further development of a field robot', *Precision Agriculture* **9**, 857–866.

Grift, T., Zhang, Q., Kondo, N. and Ting, K. (2008), 'A review of automation and robotics for the bioindustry', *Journal of Biomechatronics Engineering* **1**(1), 37–54.

Grocholsky, B., Nuske, S., Aasted, M., Achar, S., Bates, T. and House, G. (2011), A camera and laser system for automatic vine balance assessment, *in* 'ASABE Annual International Meeting', Louisville, Kentucky, 7-10 August, 2011.

Guivant, J., Masson, F. and Nebot, E. (2002), 'Simultaneous localization and map building using natural features and absolute information', *Robotics and Autonomous Systems* **40**(2-3), 79–90.

Habib, M. K. (2007), *Robot Mapping and Navigation by Fusing Sensory Information*, Mobile Robots: Perception & Navigation, Sascha Kolski (Ed.), InTech, Available from:http://cdn.intechweb.org/pdfs/154.pdf.

Hamner, B., Singh, S. and Bergerman, M. (2010), Improving orchard efficiency with autonomous utility vehicles, *in* '2010 ASABE Annual International Meeting', Pittsburgh, PA, June 20 - 23, 2010.

Han, S. and Burks, T. (2009), 3D reconstruction of a citrus canopy, *in* '2009 ASABE Annual International Meeting', Reno, Nevada, June 21-24, 2009, Paper No.:097398.

Han, S., Zhang, Q., Ni, B. and Reid, J. (2004), 'A guidance directrix approach to vision-based vehicle guidance systems', *Computers and Electronics in Agriculture* **43**(3), 179–195.

Hanawa, K., Yamashita, T., Matsuo, Y. and Hamada, Y. (2012), 'Development of a stereo vision system to assist the operation of agricultural tractors', *Japan Agricultural Research Quarterly: JARQ* **46**(4), 287–293.

Hansen, S., Bayramoglu, E., Andersen, J., Ravn, O., Andersen, N. and Poulsen, N. (2011), Orchard navigation using derivative free Kalman filtering, *in* 'American Control Conference', CA, USA, June 29 - July 01, 2011, pp. 4679–4684.

Hansen, S., Blanke, M. and Andersen, J. C. (2009), Autonomous tractor navigation in orchard-diagnosis and supervision for enhanced availability, *in* '7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes', Barcelona, Spain, June 30 - July 3, 2009, pp. 360–365.

He, B., Liu, G., Ji, Y., Si, Y. and Gao, R. (2011), Auto recognition of navigation path for harvest robot based on machine vision, *in* 'Computer and Computing Technologies in Agriculture IV', Springer, pp. 138–148.

Heidman, B. C. and Rosa, U. A. (2008), 'Real-time tree localization in orchards', *Applied engineering in agriculture* **24**(6), 707–716.

Hellström, T. (2002), Autonomous navigation for forest machines, *in* 'Pre-Study Report, Department of Computing Science, Umea University, Sweden'.

Hiremath, S. A., Van Der Heijden, G. W., Van Evert, F. K., Stein, A. and Ter Braak, C. J. (2014), 'Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter', *Computers and Electronics in Agriculture* **100**, 41–50.

Hokuyo (2008), 'Scanning laser range finder UTM-30LM/LN specification'. Hokuyo Automatic Co. Ltd., April 2008. `http://www.hokuyo-aut.jp/02sensor/07scanner/download/pdf/UTM-30LX_spec_en.pdf`.

Jæger-Hansen, C. L., Griepentrog, H. W. and Andersen, J. C. (2012), Navigation and tree mapping in orchards, *in* 'International Conference of Agricultural Engineering', pp. 1–6.

Jeon, H., Tian, L., Grift, T., Bode, L. and Hager, A. (2009), Improving orchard efficiency with autonomous utility vehicles, *in* '2009 ASABE Annual International Meeting', Reno, Nevada, June 21 - 24, 2009.

Ji, R. and Qi, L. (2011), 'Crop-row detection algorithm based on random Hough transformation', *Mathematical and Computer Modelling* **54**(3), 1016–1020.

Jiang, G.-Q., Zhao, C.-j. and Si, Y.-S. (2010), A machine vision based crop rows detection for agricultural robots, *in* '2010 International Conference on Wavelet Analysis and Pattern Recognition', Qingdao, China, July 11-14, 2010, pp. 114–118.

Jiang, G., Wang, Z. and Liu, H. (2015), 'Automatic detection of crop rows based on multi-rois', *Expert Systems with Applications* **42**(5), 2429–2441.

Jinlin, X. and Weiping, J. (2010), Vision-based guidance line detection in row crop fields, *in* 'International Conference on Intelligent Computation Technology and Automation (ICICTA)', Changsha, May 11-12, 2010, pp. 1140–1143.

Jutila, J., Kannas, K. and Visala, A. (2007), Tree measurement in forest by 2D laser scanning, *in* 'IEEE International Symposium on Computational Intelligence in Robotics and Automation', Jacksonville, FI, June 20-23, 2007, pp. 491–496.

Kaizu, Y. and Imou, K. (2008), 'A dual-spectral camera system for paddy rice seedling row detection', *Computers and Electronics in Agriculture* **63**(1), 49–56.

Karkee, M., Adhikari, B., Amatya, S. and Zhang, Q. (2014), 'Identification of pruning branches in tall spindle apple trees for automated pruning', *Computers and Electronics in Agriculture* (103), 127–135.

Kassir, A. and Peynot, T. (2010), Reliable automatic camera-laser calibration, *in* 'Proc of the 2010 Australasian Conference on Robotics and Automation (ACRA2010)', Brisbane, Australia, December 2010.

Kise, M., Noguchi, N., Ishii, K. and Terao, H. (2002), The development of the autonomous tractor with steering controller applied by optimal control, *in* 'Proceeding of the Automation Technology for Off-Road Equipment', Chicago, USA, pp. 367–373.

Kise, M., Zhang, Q. and Rovira Ms, F. (2005), ‘A stereovision-based crop row detection method for tractor-automated guidance’, *Biosystems Engineering* **90**(4), 357–367.

Kong, J., Ding, X., Liu, J., Yan, L. and Wang, J. (2015), ‘New hybrid algorithms for estimating tree stem diameters at breast height using a two dimensional terrestrial laser scanner’, *Sensors* **15**(7), 15661–15683.

Kurashiki, K., Fukao, T., Ishiyama, K., Kamiya, T. and Murakami, N. (2010), Orchard traveling UGV using particle filter based localization and inverse optimal control, *in* ‘2010 IEEE/SICE International Symposium on System Integration’, Sendai, December 21-22, 2010, pp. 31–36.

Lee, W., Alchanatis, V., Yang, C., Hirafuji, M., Moshou, D. and Li, C. (2010), ‘Sensing technologies for precision specialty crop production’, *Computers and Electronics in Agriculture* **74**(1), 2–33.

Leemans, V. and Destain, M.-F. (2006), Agricultural tools guidance assistance by using machine vision, *in* ‘7th National Congress on Theoretical and Applied Mechanics’, Mons, Belgium, May 29-30, 2006.

Li, M., Imou, K., Wakabayashi, K. and Yokoyama, S. (2009), ‘Review of research on agricultural vehicle autonomous guidance’, *International Journal of Agricultural and Biological Engineering* **2**(3), 1–16.

Libby, J. and Kantor, G. (2011), Deployment of a point and line feature localization system for an outdoor agriculture vehicle, *in* ‘IEEE International Conference on Robotics and Automation (ICRA 2011)’, Shanghai, China, May 2011, pp. 1565–1570.

Lindberg, E., Holmgren, J., Olofsson, K. and Olsson, H. (2012), ‘Estimation of stem attributes using a combination of terrestrial and airborne laser scanning’, *European Journal of Forest Research* **131**(6), 1917–1931.

Lulio, L. C., Tronco, M. L. and Porto, A. J. (2012), Cognitive-merged statistical pattern recognition method for image processing in mobile robot navigation, *in* '2012 Brazilian Robotics Symposium and Latin American Robotics Symposium', IEEE, Fortaleza, October 16-19, 2012, pp. 279–283.

Méndez, V., Rosell-Polo, J. R., Sanz, R., Escolà, A. and Catalán, H. (2014), 'Deciduous tree reconstruction algorithm based on cylinder fitting from mobile terrestrial laser scanned point clouds', *Biosystems Engineering* **124**, 78–88.

Meng, L., Sun, F. and Ge, S. S. (2010), Extrinsic calibration of a camera with dual 2D laser range sensors for a mobile robot, *in* '2010 IEEE International Symposium on Intelligent Control (ISIC)', Yokohama, September 8-10, 2010, pp. 813–817.

Mogensen, L. V., Hansen, S., Ravn, O. and Poulsen, N. K. (2009), Comparing mobile robot localisation algorithms using kalmtool, *in* 'Proceeding of 15th IFAC Symposium on System Identification', Saint-Malo, France, July 6-8, 2009.

Montalvo, M., Pajares, G., Guerrero, J. M., Romeo, J., Guijarro, M., Ribeiro, A., Ruz, J. J. and Cruz, J. (2012), 'Automatic detection of crop rows in maize fields with high weeds pressure', *Expert Systems with Applications* **39**(15), 11889–11897.

Moorehead, S. J., Wellington, C. K., Gilmore, B. J. and Vallespi, C. (2012), Automating orchards: A system of autonomous tractors for orchard maintenance, *in* 'Proceeding of IEEE international conference of Intelligent Robots and Systems, Workshop on Agricultural Robotics', Vilamoura, Portugal, October, 2012.

Mousazadeh, H. (2013), 'A technical review on navigation systems of agricultural autonomous off-road vehicles', *Journal of Terramechanics* **50**(3), 211–232.

Nagasaka, Y., Umeda, N., Kanetai, Y., Taniwaki, K. and Sasaki, Y. (2004), 'Autonomous guidance for rice transplanting using global positioning and gyroscopes', *Computers and Electronics in Agriculture* **43**(3), 223–234.

Negenborn, R. (2003), *Robot localization and Kalman filters*, Master thesis, Utrecht University.

Noguchi, N., Kise, M., Ishii, K. and Terao, H. (2002), Field automation using robot tractor, *in* 'Proceedings of Automation Technology for Off-Road Equipment)', Chicago, USA, pp. 239–245.

Okamoto, H., Hamada, K., Kataoka, T., Terawaki, M. and Hata, S. (2002), Automatic guidance system with crop row sensor, *in* 'Proceedings of the Automation Technology for Off-road Equipment, Chicago, Illinois, USA', pp. 307–316.

Oksanen, T., Kosonen, S. and Visala, A. (2005), Path planning algorithm for field traffic, *in* '2005 ASAE Annual Meeting', Tampa, FL.

Olofsson, K., Holmgren, J. and Olsson, H. (2014), 'Tree stem and height measurements using terrestrial laser scanning and the ransac algorithm', *Remote Sensing* **6**(5), 4323–4344.

Ortiz, J. and Olivares, M. (2006), A vision based navigation system for an agricultural field robot, *in* 'IEEE 3rd Latin American Robotics Symposium, 2006. LARS'06.', IEEE, pp. 106–114.

Parhi, D. and Singh, M. (2009), 'Navigational strategies of mobile robots: a review', *International Journal of Automation and Control* **3**(2), 114–134.

Pascual, M., Rufat, J., Villar, J., Rosell, J., Sanz, R. and Arno, J. (2011), 'Evaluation of peach tree growth characteristics under different irrigation strategies by lidar system: Preliminary results', *Acta horticulturae* pp. 227–232.

Pedersen, S., Fountas, S., Have, H. and Blackmore, B. (2005), 'Agricultural robots: an economic feasibility study', *Precision Agriculture* **5**, 589–595.

Peynot, T. and Kassir, A. (2010), Laser-camera data discrepancies and reliable perception in outdoor robotics, *in* '2010 IEEE/RSJ International Conference on Intelligent Robots and Systems', Taipei, Taiwan, October 18-22, 2010, pp. 2625–2632.

Rains, G. C., Thomas, D. L. and Perry, C. D. (2002), 'Pecan mechanical harvesting parameters for yield monitoring', *Transactions of the ASAE* **45**(2), 281–285.

Raumonen, P., Kaasalainen, M., Åkerblom, M., Kaasalainen, S., Kaartinen, H., Vastaranta, M., Holopainen, M., Disney, M. and Lewis, P. (2013), 'Fast automatic precision tree models from terrestrial laser scanner data', *Remote Sensing* **5**(2), 491–520.

Reske-Nielsen, A., Mejnertsen, A., Andersen, N., Ravn, O., Nørremark, M. and Griepentrog, H. (2006), Multilayer controller for outdoor vehicle, *in* 'Proceedings of Automation Technology for Off-Road Equipment (ATOF)', Bonn, Germany, September 1-2, 2006, pp. 41–49.

Ringdahl, O., Hohnloser, P., Hellström, T., Holmgren, J. and Lindroos, O. (2013), 'Enhanced algorithms for estimating tree trunk diameter using 2D laser scanner', *Remote Sensing* **5**(10), 4839–4856.

Rohweder, K., Vorst, P. and Zell, A. (2009), Improved mapping of RFID tags by fusion with spatial structure, *in* '4th European Conference on Mobile Robots (ECMR)', Mlini/Dubrovnik, Croatia, September 2009, pp. 247–252.

Rosell-Polo, J. R., Sanz, R., Llorens, J., Arnó, J., Ribes-Dasi, M., Masip, J., Camp, F., Gracia, F., Solanelles, F., Palleja, T. et al. (2009), 'A tractor-mounted scanning lidar for the non-destructive measurement of vegetative volume and surface area of tree-row plantations: a comparison with conventional destructive measurements.', *Biosystems engineering* **102**(2), 128–134.

Rosell-Polo, J. and Sanz, R. (2012), 'A review of methods and applications of the geometric characterization of tree crops in agricultural activities', *Computers and electronics in agriculture* **81**, 124–141.

Rovira-Más, F. (2009), 'Recent innovations in off-road intelligent vehicles: In-field automatic navigation', *Recent Patents Mechan. Eng* **2**, 169–178.

Sanz, R., Llorens, J., Arnó, J., Ribes, M., Masip, J., Camp, F., Gràcia, F., Solanelles, F., Planas, S. and Pallejà, T. (2011), 'Innovative LIDAR 3D dynamic measurement system to estimate fruit-tree leaf area', *Sensors* **11**(6), 5769–5791.

Satow, T., Matsuda, K., Ming, S., Hironaka, K. and Tan, D. (2004), Development of laser crop row sensor for automatic guidance system of implements, *in* 'Proceedings of the Automation Technology for Off-road Equipment (ATOE)', Kyoto, Japan, pp. 131–139.

Schilling, A., Schmidt, A. and Maas, H.-G. (2011), Automatic tree detection and diameter estimation in terrestrial laser scanner point clouds, *in* '16th Computer Vision Winter Workshop', Citeseer, Mitterberg, Austria, February 2-4, 2011, pp. 75–83.

Schumann, A. and Zaman, Q. (2005), 'Software development for real-time ultrasonic mapping of tree canopy size', *Computers and Electronics in Agriculture* **47**(1), 25–40.

Shalal, N., Low, T., McCarthy, C. and Hancock, N. (2013), A preliminary evaluation of vision and laser sensing for tree trunk detection and orchard mapping, *in* 'Proceedings of the 2013 Australasian Conference on Robotics and Automation (ACRA2013)', Sydney, Australia, December 2-4, 2013. http://www.araa.asn.au/acra/acra2013/papers/pap162s1-file1.pdf.

Sharifi, M. and Chen, X. (2015), A novel vision based row guidance approach for navigation of agricultural mobile robots in orchards, *in* '6th International Conference on Automation, Robotics and Applications', IEEE, Queenstown, New Zealand, February 17-19, 2015, pp. 251–255.

Stoll, A. and Dieter Kutzbach, H. (2000), 'Guidance of a forage harvester with GPS', *Precision Agriculture* **2**(3), 281–291.

Subramanian, V., Burks, T. and Dixon, W. (2009), 'Sensor fusion using fuzzy logic enhanced Kalman filter for autonomous vehicle guidance in citrus groves', *Transactions of the ASAE* **52**(5), 1411–1422.

Subramanian, V., Burks, T. F. and Arroyo, A. (2006), 'Development of machine vision and laser radar based autonomous vehicle guidance systems for citrus grove navigation', *Computers and Electronics in Agriculture* **53**(2), 130–143.

Tabile, R., Godoy, E., Pereira, R., Tangerino, G., Porto, A. and Inamasu, R. (2011), 'Design and development of the architecture of an agricultural mobile robot', *Engenharia Agrcola* **31**(1), 130–142.

Thamrin, N. M., Arshad, N. H. M., Adnan, R., Sam, R., Razak, N. A., Misnan, M. F. and Mahmud, S. F. (2012), Simultaneous localization and mapping based real-time inter-row tree tracking technique for unmanned aerial vehicle, *in* '2012 IEEE International Conference on Control System, Computing and Engineering', Penang, Malaysia, November, 23-25, 2012, pp. 322–327.

Thamrin, N. M., Arshad, N. H. M., Adnan, R., Sam, R., Razak, N. A., Misnan, M. F. and Mahmud, S. F. (2013), Tree detection profile using a single non-intrusive ultrasonic sensor for inter-row tracking application in agriculture field, *in* '2013 IEEE 9th International Colloquium on Signal Processing and its Applications (CSPA)', pp. 310–313.

Thamrin, N. M., Arshad, N. H. M., Adnan, R., Sam, R., Razak, N. A., Misnan, M. F. and Mahmud, S. F. (2014), Enhanced technique for cylindrical object diameter measurement via low-cost and innovated rotational non-intrusive sensor, *in* '2014 IEEE 10th International Colloquium on Signal Processing & its Applications (CSPA)', pp. 251–256.

Thrun, S., Burgard, W. and Fox, D. (2005), *Probabilistic robotics*, MIT Press: Cambridge, MA, USA.

Thuilot, B., Cariou, C., Cordesses, L. and Martinet, P. (2001), Automatic guidance of a farm tractor along curved paths, using a unique cp-dgps, *in* 'Proceedings. 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems', Vol. 2, IEEE, pp. 674–679.

Tian, Z., Junfang, X., Gang, W. and Jianbo, Z. (2014), 'Automatic navigation path detection method for tillage machines working on high crop stubble fields based on machine vision', *International Journal of Agricultural and Biological Engineering* **7**(4), 29–37.

Tillett, N., Hague, T. and Miles, S. (2002), 'Inter-row vision guidance for mechanical weed control in sugar beet', *Computers and Electronics in Agriculture* **33**(3), 163–177.

Torres-Sospedra, J. and Nebot, P. (2011), 'A new approach to visual-based sensory system for navigation into orange groves', *Sensors* **11**(4), 4086–4103.

Treder, W. et al. (2010), 'Crop loading studies with jonagoldapple tree', *Journal of Fruit and Ornamental Plant Research* **18**(1), 59–69.

Tumbo, S., Salyani, M., Whitney, J., Wheaton, T. and Miller, W. (2002), 'Investigation of laser and ultrasonic ranging sensors for measurements of citrus canopy volume', *Applied Engineering in Agriculture* **18**(3), 367–372.

Vougioukas, S., Fountas, S., Blackmore, S. and Tang, L. (2005), 'Combining reactive and deterministic behaviours for mobile agricultural robots', *Operational Research* **5**(1), 153–163.

Wei, J. and Salyani, M. (2005), 'Development of a laser scanner for measuring tree canopy characteristics: Phase 2. foliage density measurement', *Transactions American Society of Agricultural Engineers* **48**(4), 1595.

Weiss, U. and Biber, P. (2011), 'Plant detection and mapping for agricultural robots using a 3-D LIDAR sensor', *Robotics and Autonomous Systems* **59**(5), 265–273.

Welch, G. and Bishop, G. (2006), 'An introduction to the Kalman filter', *Department of Computer Science, University of North Carolina* .

Williams, S., Dissanayake, G. and Durrant-Whyte, H. (2002), Field deployment of the simultaneous localisation and mapping algorithm, *in* '15th IFAC World Congress on Automatic Control', Barcelona, Spain, June 2002.

Xue, J., Zhang, L. and Grift, T. E. (2012), 'Variable field-of-view machine vision based row guidance of an agricultural robot', *Computers and Electronics in Agriculture* **84**, 85–91.

Yan, J., Guorong, L., Shenghua, L. and Lian, Z. (2009), A review on localization and mapping algorithm based on extended Kalman filtering, *in* 'IEEE International Forum on Information Technology and Applications (IFITA)', Vol. 2, pp. 435–440.

Yang, X., Strahler, A. H., Schaaf, C. B., Jupp, D. L., Yao, T., Zhao, F., Wang, Z., Culvenor, D. S., Newnham, G. J., Lovell, J. L. et al. (2013), 'Three-dimensional forest reconstruction and structural parameter retrievals using a terrestrial full-waveform lidar instrument (echidna®)', *Remote Sensing of Environment* **135**, 36–51.

Yang, Y., Zhang, Y. and He, T. (2011), 'Application analysis of machine vision technology in the agricultural inspection', *Computer and Computing Technologies in Agriculture IV* pp. 316–321.

Zhang, J., Chambers, A., Maeta, S., Bergerman, M. and Singh, S. (2013), 3D perception for accurate row following: Methodology and results, *in* '2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', Tokyo, Japan, November 3-7, 2013, pp. 5306–5313.

Zhang, J., Kantor, G., Bergerman, M. and Singh, S. (2012), Monocular visual navigation of an autonomous vehicle in natural scene corridor-like environments, *in* '2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', IEEE, Vilamoura, Algarve, Portugal, October 7-12, 2012, pp. 3659–3666.

Zhang, J., Maeta, S., Bergerman, M. and Singh, S. (2014), Mapping orchards for autonomous navigation, *in* '2014 ASABE and CSBE/SCGAB Annual International Meeting', Montreal, Quebec Canada, July 13-16, 2014.

Zhang, S., Xie, L. and Xiao, W. (2008), *A Novel Feature Extraction Algorithm for Outdoor Mobile Robot Localization*, Motion Planning, Xing-Jian Jing (Ed.), InTech, Available from: http://cdn.intechweb.org/pdfs/5379.pdf.

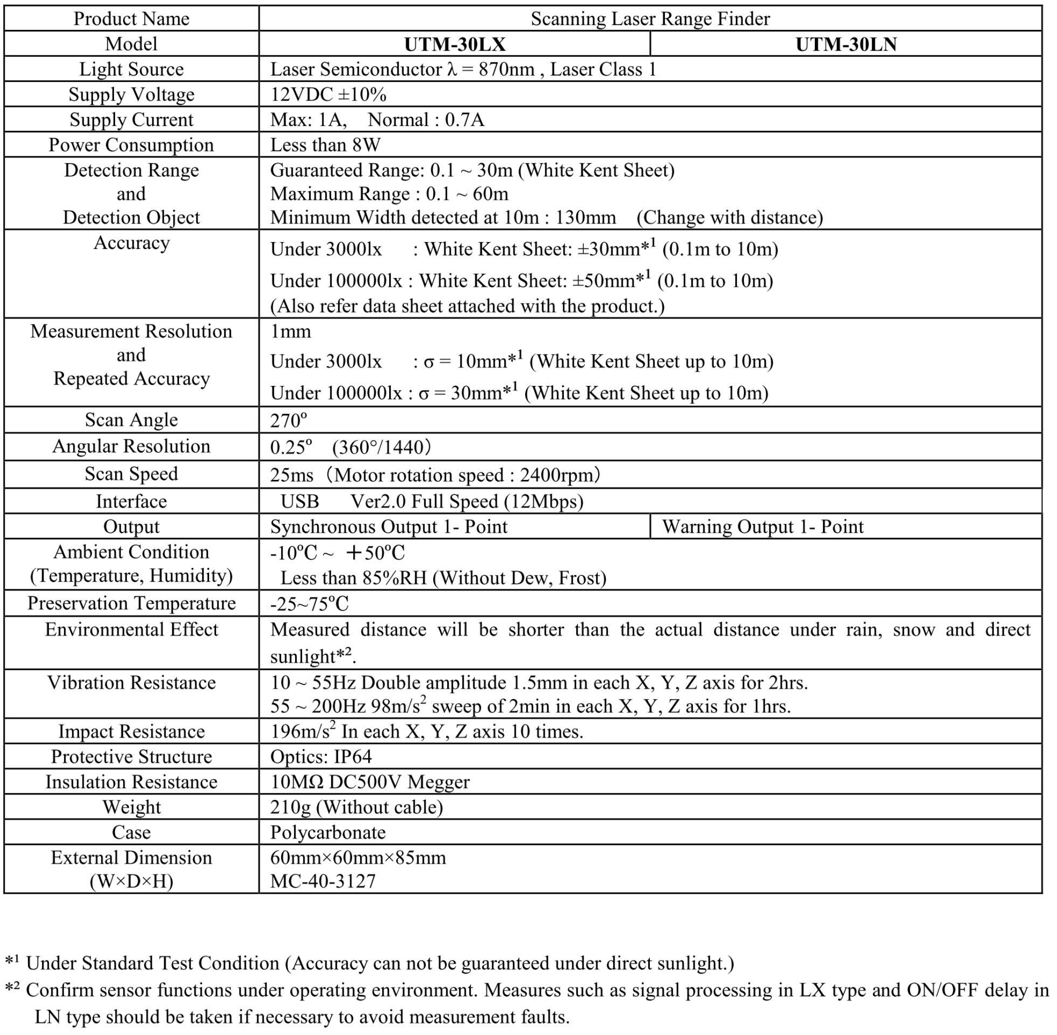
Zhao, C.-J. and Jiang, G.-Q. (2010), Baseline detection and matching to vision-based navigation of agricultural robot, *in* '2010 International Conference on Wavelet Analysis and Pattern Recognition', Qingdao, China, July 11-14, 2010, pp. 44–48.

Zheng, Y., Liu, J., Wang, D. and Yang, R. (2012), 'Laser scanning measurements on trees for logging harvesting operations', *Sensors* **12**(7), 9273–9285.

Appendix A

Sensors specifications

A.1 Laser scanner specifications

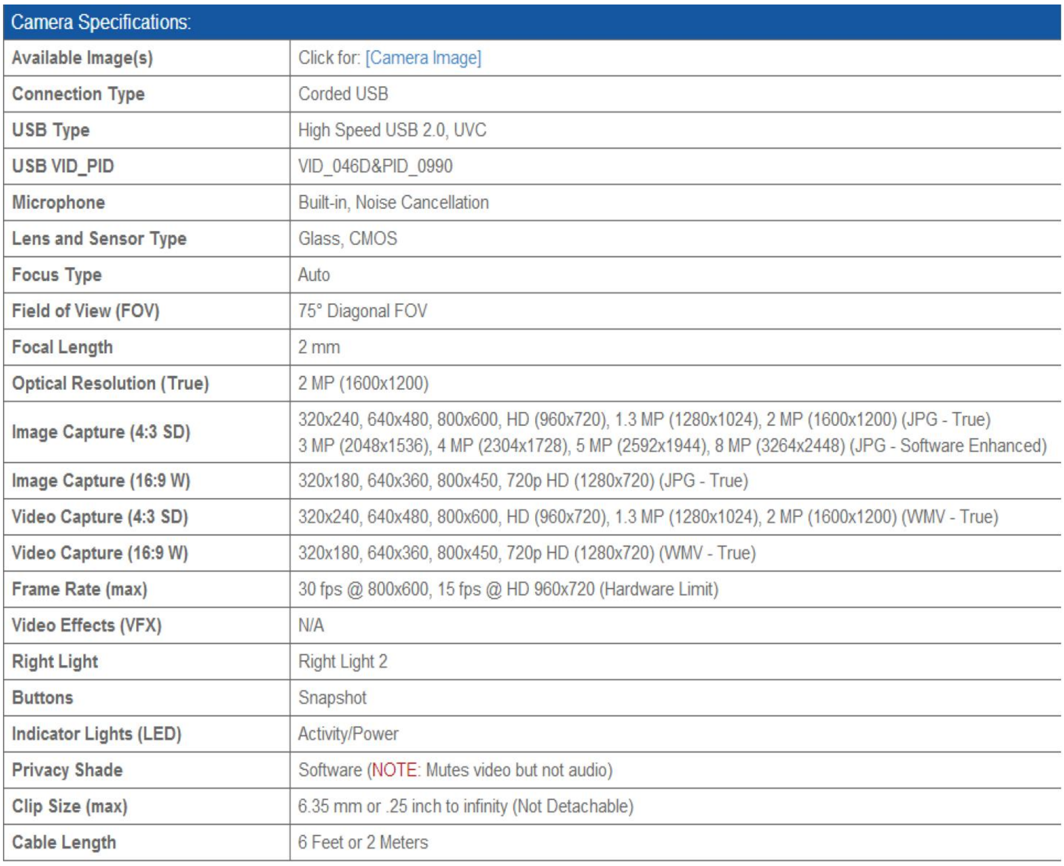


Product Name	Scanning Laser Range Finder	
Model	**UTM-30LX**	**UTM-30LN**
Light Source	Laser Semiconductor λ = 870nm , Laser Class 1	
Supply Voltage	12VDC ±10%	
Supply Current	Max: 1A, Normal : 0.7A	
Power Consumption	Less than 8W	
Detection Range and Detection Object	Guaranteed Range: 0.1 ~ 30m (White Kent Sheet) Maximum Range : 0.1 ~ 60m Minimum Width detected at 10m : 130mm (Change with distance)	
Accuracy	Under 3000lx : White Kent Sheet: ±30mm*[1] (0.1m to 10m) Under 100000lx : White Kent Sheet: ±50mm*[1] (0.1m to 10m) (Also refer data sheet attached with the product.)	
Measurement Resolution and Repeated Accuracy	1mm Under 3000lx : σ = 10mm*[1] (White Kent Sheet up to 10m) Under 100000lx : σ = 30mm*[1] (White Kent Sheet up to 10m)	
Scan Angle	270º	
Angular Resolution	0.25º (360°/1440)	
Scan Speed	25ms（Motor rotation speed : 2400rpm）	
Interface	USB Ver2.0 Full Speed (12Mbps)	
Output	Synchronous Output 1- Point	Warning Output 1- Point
Ambient Condition (Temperature, Humidity)	-10ºC ~ ＋50ºC Less than 85%RH (Without Dew, Frost)	
Preservation Temperature	-25~75ºC	
Environmental Effect	Measured distance will be shorter than the actual distance under rain, snow and direct sunlight*[2].	
Vibration Resistance	10 ~ 55Hz Double amplitude 1.5mm in each X, Y, Z axis for 2hrs. 55 ~ 200Hz 98m/s^2 sweep of 2min in each X, Y, Z axis for 1hrs.	
Impact Resistance	196m/s^2 In each X, Y, Z axis 10 times.	
Protective Structure	Optics: IP64	
Insulation Resistance	10MΩ DC500V Megger	
Weight	210g (Without cable)	
Case	Polycarbonate	
External Dimension (W×D×H)	60mm×60mm×85mm MC-40-3127	

*[1] Under Standard Test Condition (Accuracy can not be guaranteed under direct sunlight.)

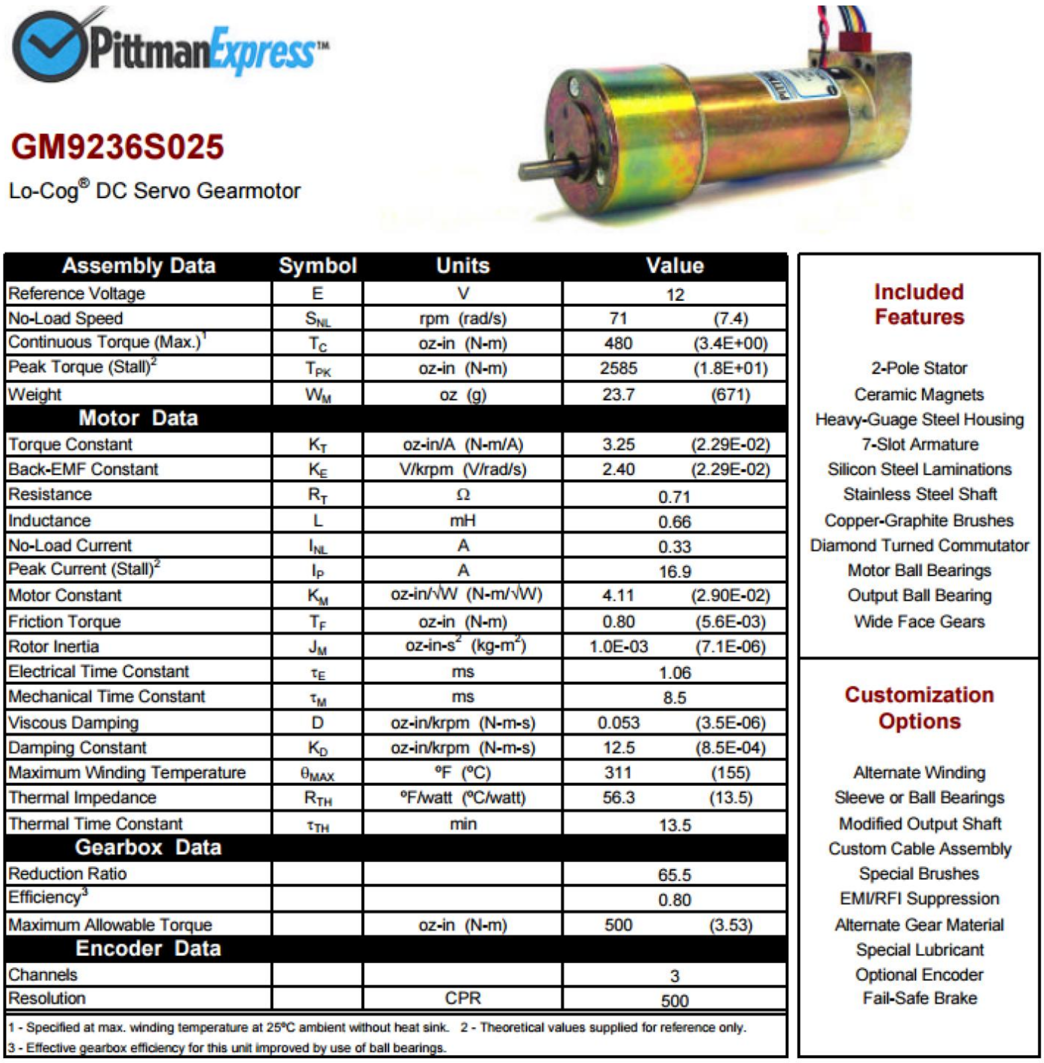
*[2] Confirm sensor functions under operating environment. Measures such as signal processing in LX type and ON/OFF delay in LN type should be taken if necessary to avoid measurement faults.

A.2 Camera specifications



Camera Specifications:	
Available Image(s)	Click for: [Camera Image]
Connection Type	Corded USB
USB Type	High Speed USB 2.0, UVC
USB VID_PID	VID_046D&PID_0990
Microphone	Built-in, Noise Cancellation
Lens and Sensor Type	Glass, CMOS
Focus Type	Auto
Field of View (FOV)	75° Diagonal FOV
Focal Length	2 mm
Optical Resolution (True)	2 MP (1600x1200)
Image Capture (4:3 SD)	320x240, 640x480, 800x600, HD (960x720), 1.3 MP (1280x1024), 2 MP (1600x1200) (JPG - True) 3 MP (2048x1536), 4 MP (2304x1728), 5 MP (2592x1944), 8 MP (3264x2448) (JPG - Software Enhanced)
Image Capture (16:9 W)	320x180, 640x360, 800x450, 720p HD (1280x720) (JPG - True)
Video Capture (4:3 SD)	320x240, 640x480, 800x600, HD (960x720), 1.3 MP (1280x1024), 2 MP (1600x1200) (WMV - True)
Video Capture (16:9 W)	320x180, 640x360, 800x450, 720p HD (1280x720) (WMV - True)
Frame Rate (max)	30 fps @ 800x600, 15 fps @ HD 960x720 (Hardware Limit)
Video Effects (VFX)	N/A
Right Light	Right Light 2
Buttons	Snapshot
Indicator Lights (LED)	Activity/Power
Privacy Shade	Software (NOTE: Mutes video but not audio)
Clip Size (max)	6.35 mm or .25 inch to infinity (Not Detachable)
Cable Length	6 Feet or 2 Meters

A.3 DC motors with built-in encoder specifications



PittmanExpress™

GM9236S025

Lo-Cog® DC Servo Gearmotor

Assembly Data	Symbol	Units	Value	
Reference Voltage	E	V	12	
No-Load Speed	S_{NL}	rpm (rad/s)	71	(7.4)
Continuous Torque (Max.)[1]	T_C	oz-in (N-m)	480	(3.4E+00)
Peak Torque (Stall)[2]	T_{PK}	oz-in (N-m)	2585	(1.8E+01)
Weight	W_M	oz (g)	23.7	(671)
Motor Data				
Torque Constant	K_T	oz-in/A (N-m/A)	3.25	(2.29E-02)
Back-EMF Constant	K_E	V/krpm (V/rad/s)	2.40	(2.29E-02)
Resistance	R_T	Ω	0.71	
Inductance	L	mH	0.66	
No-Load Current	I_{NL}	A	0.33	
Peak Current (Stall)[2]	I_P	A	16.9	
Motor Constant	K_M	oz-in/√W (N-m/√W)	4.11	(2.90E-02)
Friction Torque	T_F	oz-in (N-m)	0.80	(5.6E-03)
Rotor Inertia	J_M	oz-in-s^2 (kg-m^2)	1.0E-03	(7.1E-06)
Electrical Time Constant	τ_E	ms	1.06	
Mechanical Time Constant	τ_M	ms	8.5	
Viscous Damping	D	oz-in/krpm (N-m-s)	0.053	(3.5E-06)
Damping Constant	K_D	oz-in/krpm (N-m-s)	12.5	(8.5E-04)
Maximum Winding Temperature	θ_{MAX}	°F (°C)	311	(155)
Thermal Impedance	R_{TH}	°F/watt (°C/watt)	56.3	(13.5)
Thermal Time Constant	τ_{TH}	min	13.5	
Gearbox Data				
Reduction Ratio			65.5	
Efficiency[3]			0.80	
Maximum Allowable Torque		oz-in (N-m)	500	(3.53)
Encoder Data				
Channels			3	
Resolution		CPR	500	

1 - Specified at max. winding temperature at 25°C ambient without heat sink. 2 - Theoretical values supplied for reference only.
3 - Effective gearbox efficiency for this unit improved by use of ball bearings.

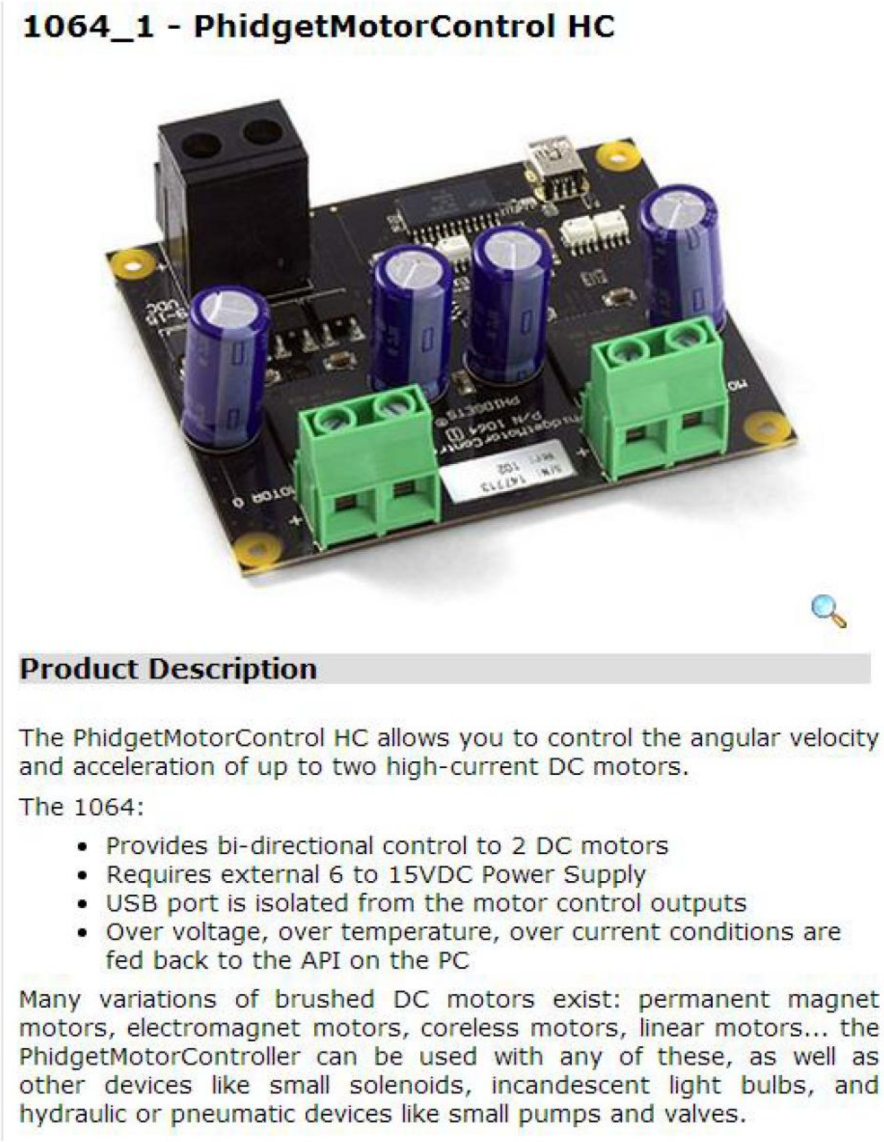
Included Features

2-Pole Stator
Ceramic Magnets
Heavy-Guage Steel Housing
7-Slot Armature
Silicon Steel Laminations
Stainless Steel Shaft
Copper-Graphite Brushes
Diamond Turned Commutator
Motor Ball Bearings
Output Ball Bearing
Wide Face Gears

Customization Options

Alternate Winding
Sleeve or Ball Bearings
Modified Output Shaft
Custom Cable Assembly
Special Brushes
EMI/RFI Suppression
Alternate Gear Material
Special Lubricant
Optional Encoder
Fail-Safe Brake

A.4 Motor control specifications



1064_1 - PhidgetMotorControl HC

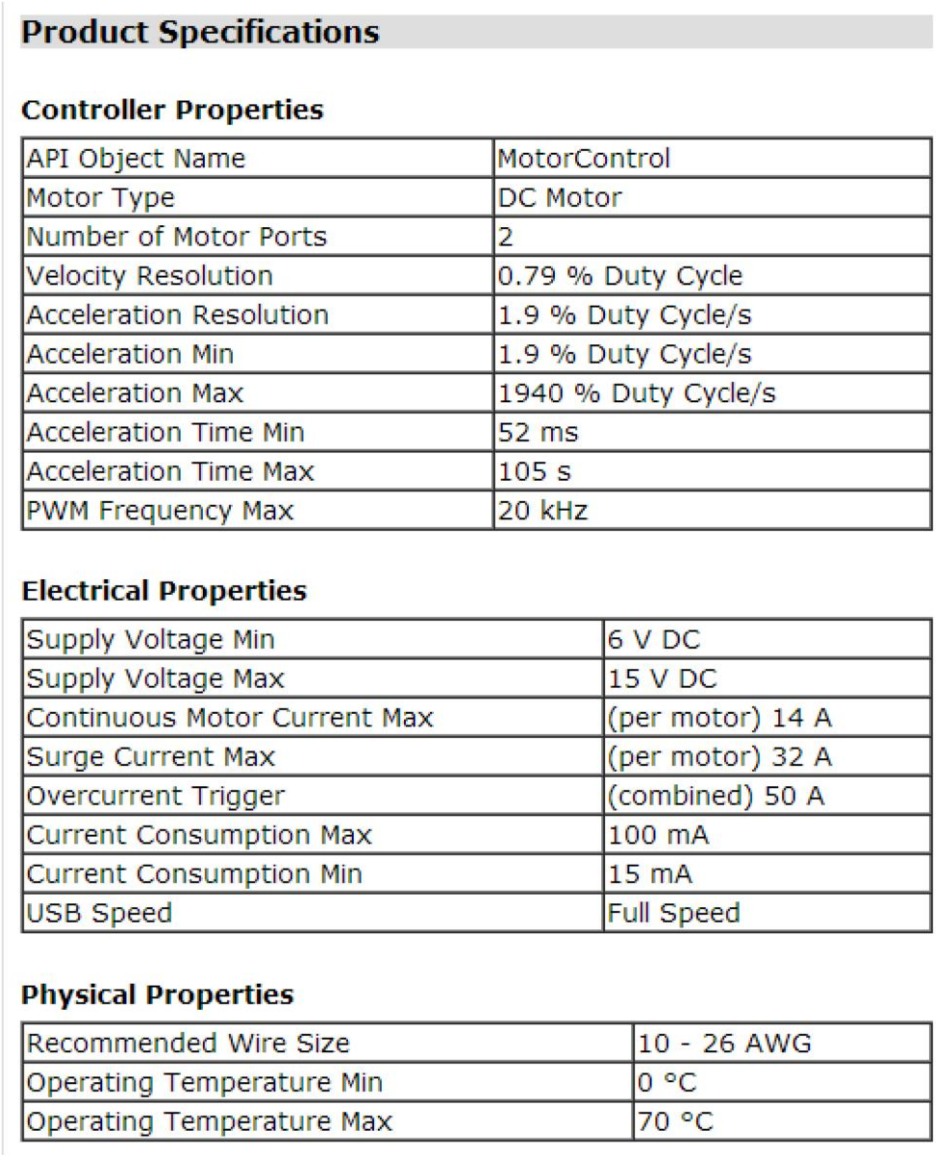
Product Description

The PhidgetMotorControl HC allows you to control the angular velocity and acceleration of up to two high-current DC motors.

The 1064:

- Provides bi-directional control to 2 DC motors
- Requires external 6 to 15VDC Power Supply
- USB port is isolated from the motor control outputs
- Over voltage, over temperature, over current conditions are fed back to the API on the PC

Many variations of brushed DC motors exist: permanent magnet motors, electromagnet motors, coreless motors, linear motors... the PhidgetMotorController can be used with any of these, as well as other devices like small solenoids, incandescent light bulbs, and hydraulic or pneumatic devices like small pumps and valves.



Product Specifications

Controller Properties

API Object Name	MotorControl
Motor Type	DC Motor
Number of Motor Ports	2
Velocity Resolution	0.79 % Duty Cycle
Acceleration Resolution	1.9 % Duty Cycle/s
Acceleration Min	1.9 % Duty Cycle/s
Acceleration Max	1940 % Duty Cycle/s
Acceleration Time Min	52 ms
Acceleration Time Max	105 s
PWM Frequency Max	20 kHz

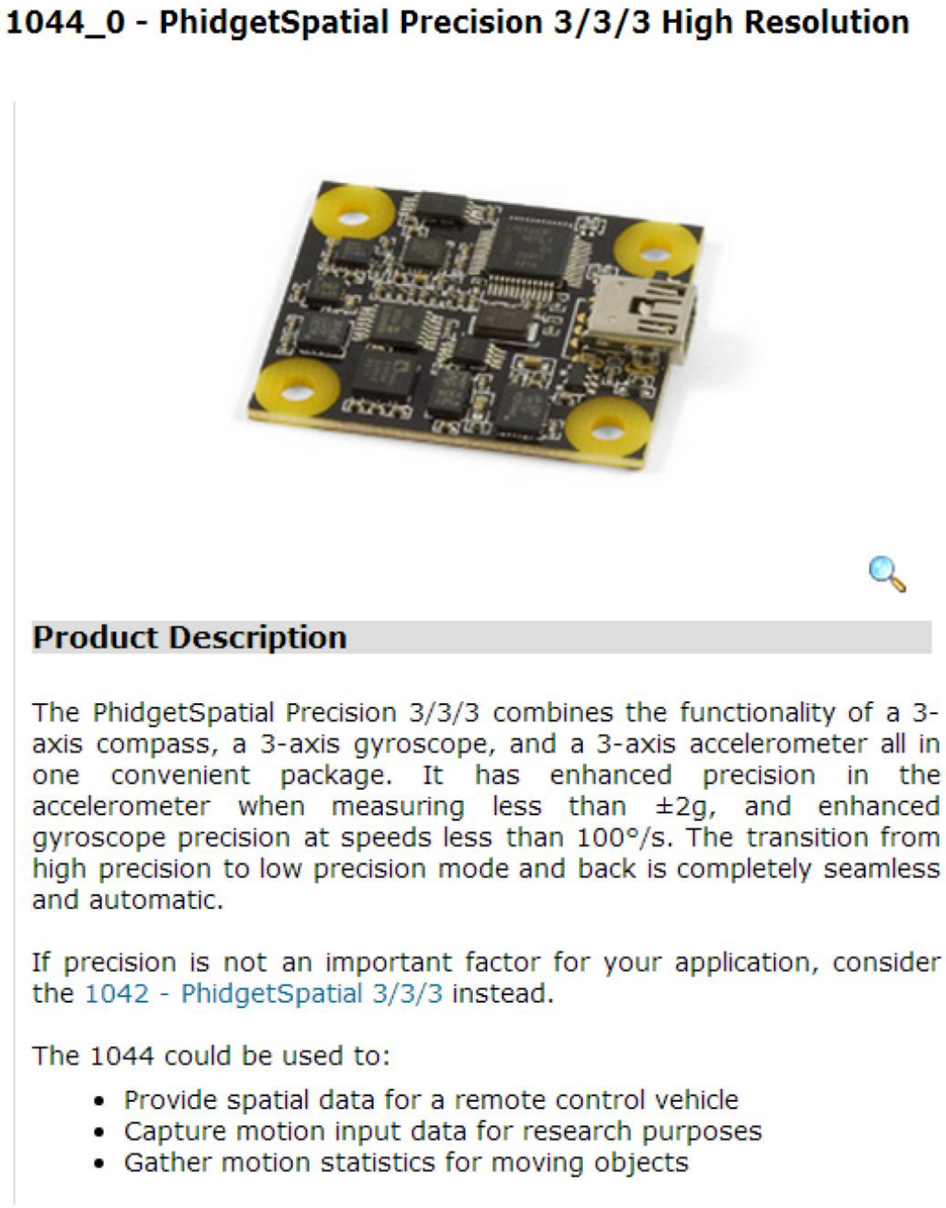
Electrical Properties

Supply Voltage Min	6 V DC
Supply Voltage Max	15 V DC
Continuous Motor Current Max	(per motor) 14 A
Surge Current Max	(per motor) 32 A
Overcurrent Trigger	(combined) 50 A
Current Consumption Max	100 mA
Current Consumption Min	15 mA
USB Speed	Full Speed

Physical Properties

Recommended Wire Size	10 - 26 AWG
Operating Temperature Min	0 °C
Operating Temperature Max	70 °C

A.5 Inertial Measurement unit (IMU) specifications



1044_0 - PhidgetSpatial Precision 3/3/3 High Resolution

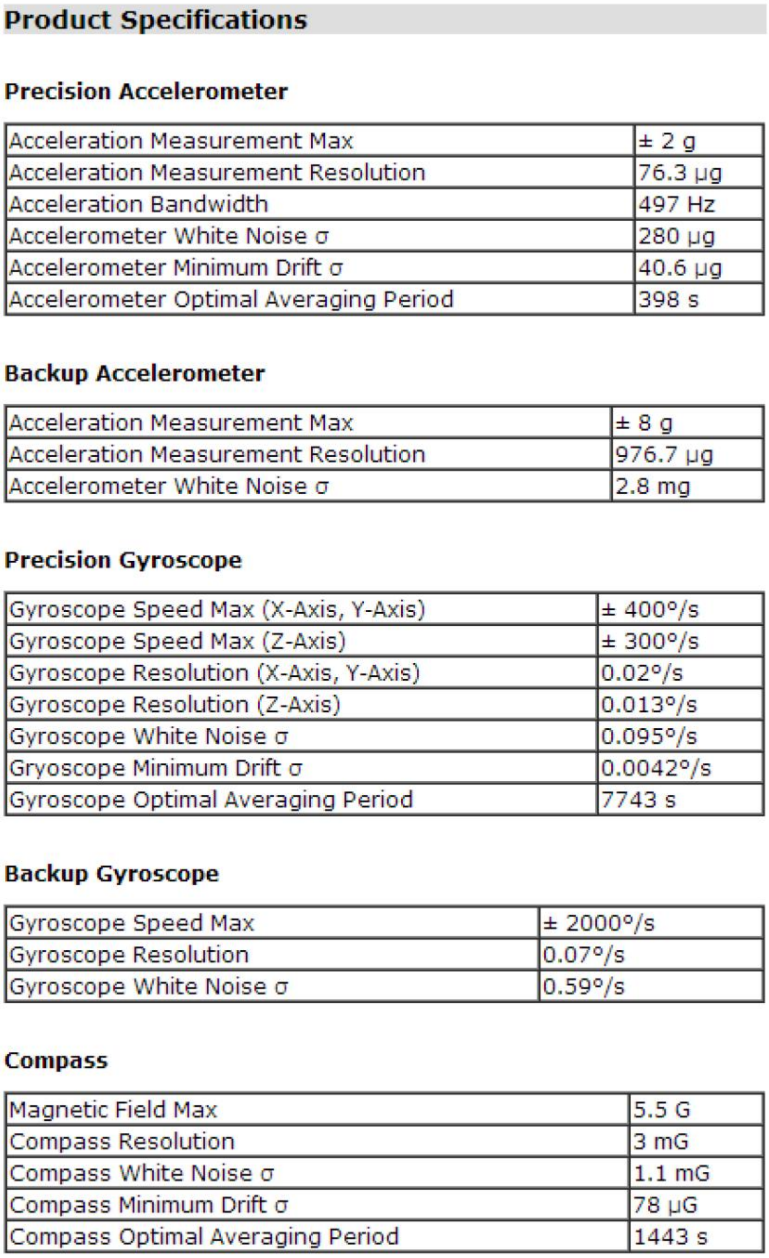
Product Description

The PhidgetSpatial Precision 3/3/3 combines the functionality of a 3-axis compass, a 3-axis gyroscope, and a 3-axis accelerometer all in one convenient package. It has enhanced precision in the accelerometer when measuring less than ±2g, and enhanced gyroscope precision at speeds less than 100°/s. The transition from high precision to low precision mode and back is completely seamless and automatic.

If precision is not an important factor for your application, consider the 1042 - PhidgetSpatial 3/3/3 instead.

The 1044 could be used to:

- Provide spatial data for a remote control vehicle
- Capture motion input data for research purposes
- Gather motion statistics for moving objects



Product Specifications

Precision Accelerometer

Acceleration Measurement Max	± 2 g
Acceleration Measurement Resolution	76.3 µg
Acceleration Bandwidth	497 Hz
Accelerometer White Noise σ	280 µg
Accelerometer Minimum Drift σ	40.6 µg
Accelerometer Optimal Averaging Period	398 s

Backup Accelerometer

Acceleration Measurement Max	± 8 g
Acceleration Measurement Resolution	976.7 µg
Accelerometer White Noise σ	2.8 mg

Precision Gyroscope

Gyroscope Speed Max (X-Axis, Y-Axis)	± 400°/s
Gyroscope Speed Max (Z-Axis)	± 300°/s
Gyroscope Resolution (X-Axis, Y-Axis)	0.02°/s
Gyroscope Resolution (Z-Axis)	0.013°/s
Gyroscope White Noise σ	0.095°/s
Gryoscope Minimum Drift σ	0.0042°/s
Gyroscope Optimal Averaging Period	7743 s

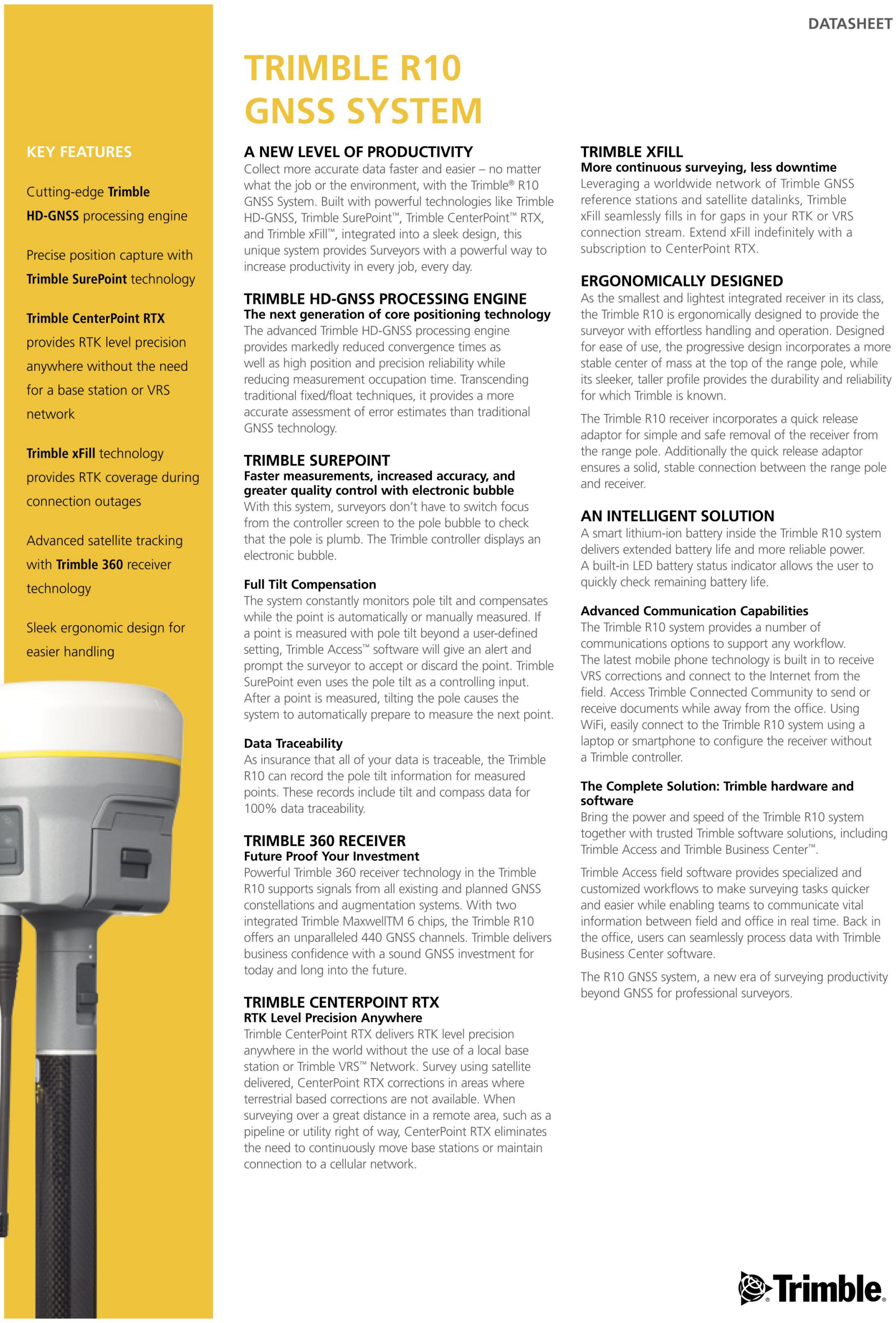
Backup Gyroscope

Gyroscope Speed Max	± 2000°/s
Gyroscope Resolution	0.07°/s
Gyroscope White Noise σ	0.59°/s

Compass

Magnetic Field Max	5.5 G
Compass Resolution	3 mG
Compass White Noise σ	1.1 mG
Compass Minimum Drift σ	78 µG
Compass Optimal Averaging Period	1443 s

A.6 RTK-GPS specifications



DATASHEET

TRIMBLE R10 GNSS SYSTEM

KEY FEATURES

Cutting-edge **Trimble HD-GNSS** processing engine

Precise position capture with **Trimble SurePoint** technology

Trimble CenterPoint RTX provides RTK level precision anywhere without the need for a base station or VRS network

Trimble xFill technology provides RTK coverage during connection outages

Advanced satellite tracking with **Trimble 360** receiver technology

Sleek ergonomic design for easier handling

A NEW LEVEL OF PRODUCTIVITY

Collect more accurate data faster and easier – no matter what the job or the environment, with the Trimble® R10 GNSS System. Built with powerful technologies like Trimble HD-GNSS, Trimble SurePoint™, Trimble CenterPoint™ RTX, and Trimble xFill™, integrated into a sleek design, this unique system provides Surveyors with a powerful way to increase productivity in every job, every day.

TRIMBLE HD-GNSS PROCESSING ENGINE

The next generation of core positioning technology

The advanced Trimble HD-GNSS processing engine provides markedly reduced convergence times as well as high position and precision reliability while reducing measurement occupation time. Transcending traditional fixed/float techniques, it provides a more accurate assessment of error estimates than traditional GNSS technology.

TRIMBLE SUREPOINT

Faster measurements, increased accuracy, and greater quality control with electronic bubble

With this system, surveyors don't have to switch focus from the controller screen to the pole bubble to check that the pole is plumb. The Trimble controller displays an electronic bubble.

Full Tilt Compensation

The system constantly monitors pole tilt and compensates while the point is automatically or manually measured. If a point is measured with pole tilt beyond a user-defined setting, Trimble Access™ software will give an alert and prompt the surveyor to accept or discard the point. Trimble SurePoint even uses the pole tilt as a controlling input. After a point is measured, tilting the pole causes the system to automatically prepare to measure the next point.

Data Traceability

As insurance that all of your data is traceable, the Trimble R10 can record the pole tilt information for measured points. These records include tilt and compass data for 100% data traceability.

TRIMBLE 360 RECEIVER

Future Proof Your Investment

Powerful Trimble 360 receiver technology in the Trimble R10 supports signals from all existing and planned GNSS constellations and augmentation systems. With two integrated Trimble MaxwellTM 6 chips, the Trimble R10 offers an unparalleled 440 GNSS channels. Trimble delivers business confidence with a sound GNSS investment for today and long into the future.

TRIMBLE CENTERPOINT RTX

RTK Level Precision Anywhere

Trimble CenterPoint RTX delivers RTK level precision anywhere in the world without the use of a local base station or Trimble VRS™ Network. Survey using satellite delivered, CenterPoint RTX corrections in areas where terrestrial based corrections are not available. When surveying over a great distance in a remote area, such as a pipeline or utility right of way, CenterPoint RTX eliminates the need to continuously move base stations or maintain connection to a cellular network.

TRIMBLE XFILL

More continuous surveying, less downtime

Leveraging a worldwide network of Trimble GNSS reference stations and satellite datalinks, Trimble xFill seamlessly fills in for gaps in your RTK or VRS connection stream. Extend xFill indefinitely with a subscription to CenterPoint RTX.

ERGONOMICALLY DESIGNED

As the smallest and lightest integrated receiver in its class, the Trimble R10 is ergonomically designed to provide the surveyor with effortless handling and operation. Designed for ease of use, the progressive design incorporates a more stable center of mass at the top of the range pole, while its sleeker, taller profile provides the durability and reliability for which Trimble is known.

The Trimble R10 receiver incorporates a quick release adaptor for simple and safe removal of the receiver from the range pole. Additionally the quick release adaptor ensures a solid, stable connection between the range pole and receiver.

AN INTELLIGENT SOLUTION

A smart lithium-ion battery inside the Trimble R10 system delivers extended battery life and more reliable power. A built-in LED battery status indicator allows the user to quickly check remaining battery life.

Advanced Communication Capabilities

The Trimble R10 system provides a number of communications options to support any workflow. The latest mobile phone technology is built in to receive VRS corrections and connect to the Internet from the field. Access Trimble Connected Community to send or receive documents while away from the office. Using WiFi, easily connect to the Trimble R10 system using a laptop or smartphone to configure the receiver without a Trimble controller.

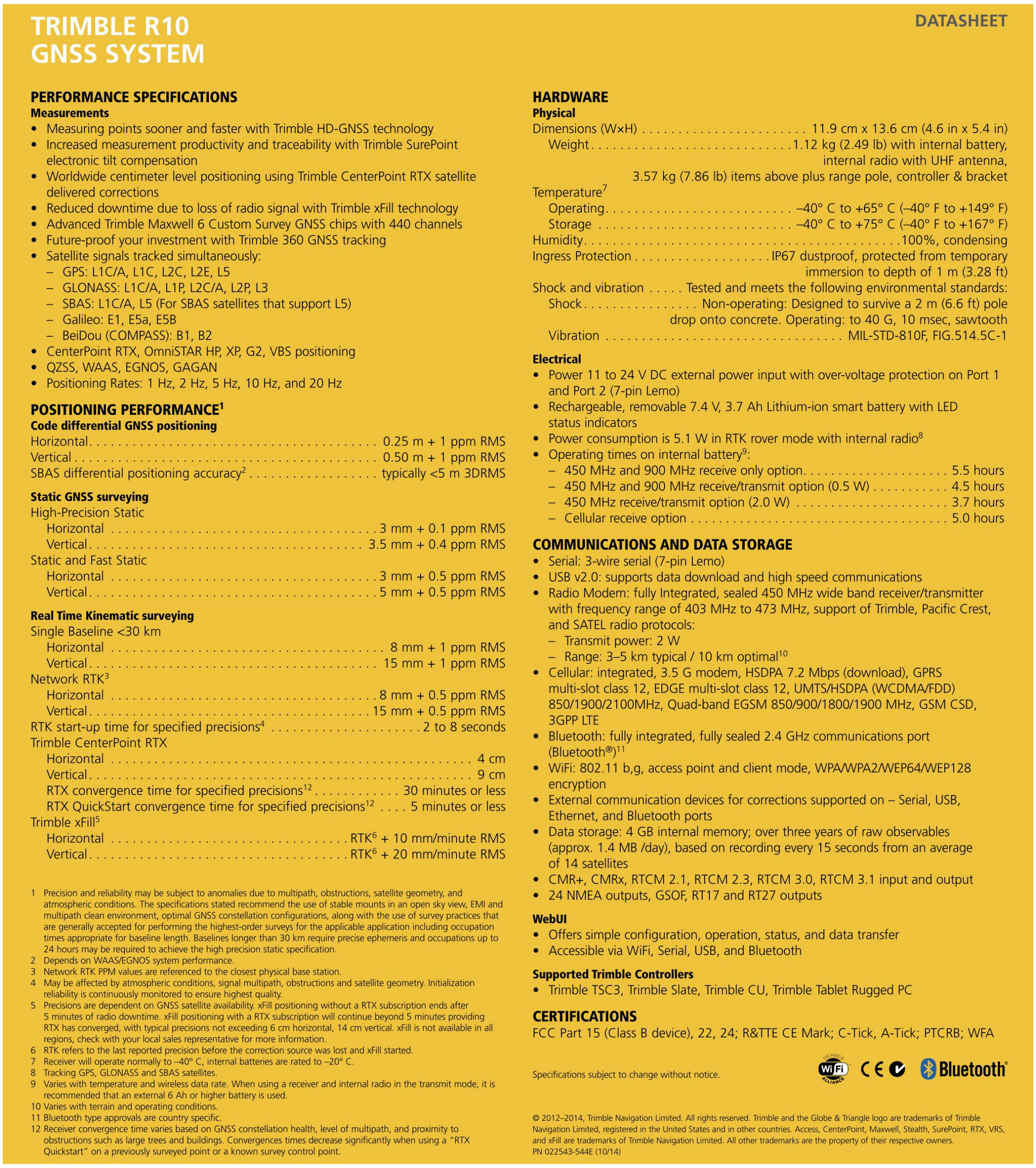
The Complete Solution: Trimble hardware and software

Bring the power and speed of the Trimble R10 system together with trusted Trimble software solutions, including Trimble Access and Trimble Business Center™.

Trimble Access field software provides specialized and customized workflows to make surveying tasks quicker and easier while enabling teams to communicate vital information between field and office in real time. Back in the office, users can seamlessly process data with Trimble Business Center software.

The R10 GNSS system, a new era of surveying productivity beyond GNSS for professional surveyors.

Trimble.



TRIMBLE R10 GNSS SYSTEM

DATASHEET

PERFORMANCE SPECIFICATIONS

Measurements

- Measuring points sooner and faster with Trimble HD-GNSS technology
- Increased measurement productivity and traceability with Trimble SurePoint electronic tilt compensation
- Worldwide centimeter level positioning using Trimble CenterPoint RTX satellite delivered corrections
- Reduced downtime due to loss of radio signal with Trimble xFill technology
- Advanced Trimble Maxwell 6 Custom Survey GNSS chips with 440 channels
- Future-proof your investment with Trimble 360 GNSS tracking
- Satellite signals tracked simultaneously:
  - GPS: L1C/A, L1C, L2C, L2E, L5
  - GLONASS: L1C/A, L1P, L2C/A, L2P, L3
  - SBAS: L1C/A, L5 (For SBAS satellites that support L5)
  - Galileo: E1, E5a, E5B
  - BeiDou (COMPASS): B1, B2
- CenterPoint RTX, OmniSTAR HP, XP, G2, VBS positioning
- QZSS, WAAS, EGNOS, GAGAN
- Positioning Rates: 1 Hz, 2 Hz, 5 Hz, 10 Hz, and 20 Hz

POSITIONING PERFORMANCE[1]

Code differential GNSS positioning

Horizontal	0.25 m + 1 ppm RMS
Vertical	0.50 m + 1 ppm RMS
SBAS differential positioning accuracy[2]	typically <5 m 3DRMS

Static GNSS surveying

High-Precision Static	
Horizontal	3 mm + 0.1 ppm RMS
Vertical	3.5 mm + 0.4 ppm RMS
Static and Fast Static	
Horizontal	3 mm + 0.5 ppm RMS
Vertical	5 mm + 0.5 ppm RMS

Real Time Kinematic surveying

Single Baseline <30 km	
Horizontal	8 mm + 1 ppm RMS
Vertical	15 mm + 1 ppm RMS
Network RTK[3]	
Horizontal	8 mm + 0.5 ppm RMS
Vertical	15 mm + 0.5 ppm RMS
RTK start-up time for specified precisions[4]	2 to 8 seconds
Trimble CenterPoint RTX	
Horizontal	4 cm
Vertical	9 cm
RTX convergence time for specified precisions[12]	30 minutes or less
RTX QuickStart convergence time for specified precisions[12]	5 minutes or less
Trimble xFill[5]	
Horizontal	RTK[6] + 10 mm/minute RMS
Vertical	RTK[6] + 20 mm/minute RMS

HARDWARE

Physical

Dimensions (W×H)	11.9 cm x 13.6 cm (4.6 in x 5.4 in)
Weight	1.12 kg (2.49 lb) with internal battery, internal radio with UHF antenna, 3.57 kg (7.86 lb) items above plus range pole, controller & bracket
Temperature[7]	
Operating	–40° C to +65° C (–40° F to +149° F)
Storage	–40° C to +75° C (–40° F to +167° F)
Humidity	100%, condensing
Ingress Protection	IP67 dustproof, protected from temporary immersion to depth of 1 m (3.28 ft)
Shock and vibration	Tested and meets the following environmental standards:
Shock	Non-operating: Designed to survive a 2 m (6.6 ft) pole drop onto concrete. Operating: to 40 G, 10 msec, sawtooth
Vibration	MIL-STD-810F, FIG.514.5C-1

Electrical

- Power 11 to 24 V DC external power input with over-voltage protection on Port 1 and Port 2 (7-pin Lemo)
- Rechargeable, removable 7.4 V, 3.7 Ah Lithium-ion smart battery with LED status indicators
- Power consumption is 5.1 W in RTK rover mode with internal radio[8]
- Operating times on internal battery[9]:

– 450 MHz and 900 MHz receive only option	5.5 hours
– 450 MHz and 900 MHz receive/transmit option (0.5 W)	4.5 hours
– 450 MHz receive/transmit option (2.0 W)	3.7 hours
– Cellular receive option	5.0 hours

COMMUNICATIONS AND DATA STORAGE

- Serial: 3-wire serial (7-pin Lemo)
- USB v2.0: supports data download and high speed communications
- Radio Modem: fully Integrated, sealed 450 MHz wide band receiver/transmitter with frequency range of 403 MHz to 473 MHz, support of Trimble, Pacific Crest, and SATEL radio protocols:
  - Transmit power: 2 W
  - Range: 3–5 km typical / 10 km optimal[10]
- Cellular: integrated, 3.5 G modem, HSDPA 7.2 Mbps (download), GPRS multi-slot class 12, EDGE multi-slot class 12, UMTS/HSDPA (WCDMA/FDD) 850/1900/2100MHz, Quad-band EGSM 850/900/1800/1900 MHz, GSM CSD, 3GPP LTE
- Bluetooth: fully integrated, fully sealed 2.4 GHz communications port (Bluetooth®)[11]
- WiFi: 802.11 b,g, access point and client mode, WPA/WPA2/WEP64/WEP128 encryption
- External communication devices for corrections supported on – Serial, USB, Ethernet, and Bluetooth ports
- Data storage: 4 GB internal memory; over three years of raw observables (approx. 1.4 MB /day), based on recording every 15 seconds from an average of 14 satellites
- CMR+, CMRx, RTCM 2.1, RTCM 2.3, RTCM 3.0, RTCM 3.1 input and output
- 24 NMEA outputs, GSOF, RT17 and RT27 outputs

WebUI

- Offers simple configuration, operation, status, and data transfer
- Accessible via WiFi, Serial, USB, and Bluetooth

Supported Trimble Controllers

- Trimble TSC3, Trimble Slate, Trimble CU, Trimble Tablet Rugged PC

CERTIFICATIONS

FCC Part 15 (Class B device), 22, 24; R&TTE CE Mark; C-Tick, A-Tick; PTCRB; WFA

Specifications subject to change without notice.

[1] Precision and reliability may be subject to anomalies due to multipath, obstructions, satellite geometry, and atmospheric conditions. The specifications stated recommend the use of stable mounts in an open sky view, EMI and multipath clean environment, optimal GNSS constellation configurations, along with the use of survey practices that are generally accepted for performing the highest-order surveys for the applicable application including occupation times appropriate for baseline length. Baselines longer than 30 km require precise ephemeris and occupations up to 24 hours may be required to achieve the high precision static specification.

[2] Depends on WAAS/EGNOS system performance.

[3] Network RTK PPM values are referenced to the closest physical base station.

[4] May be affected by atmospheric conditions, signal multipath, obstructions and satellite geometry. Initialization reliability is continuously monitored to ensure highest quality.

[5] Precisions are dependent on GNSS satellite availability. xFill positioning without a RTX subscription ends after 5 minutes of radio downtime. xFill positioning with a RTX subscription will continue beyond 5 minutes providing RTX has converged, with typical precisions not exceeding 6 cm horizontal, 14 cm vertical. xFill is not available in all regions, check with your local sales representative for more information.

[6] RTK refers to the last reported precision before the correction source was lost and xFill started.

[7] Receiver will operate normally to –40° C, internal batteries are rated to –20° C.

[8] Tracking GPS, GLONASS and SBAS satellites.

[9] Varies with temperature and wireless data rate. When using a receiver and internal radio in the transmit mode, it is recommended that an external 6 Ah or higher battery is used.

[10] Varies with terrain and operating conditions.

[11] Bluetooth type approvals are country specific.

[12] Receiver convergence time varies based on GNSS constellation health, level of multipath, and proximity to obstructions such as large trees and buildings. Convergences times decrease significantly when using a "RTX Quickstart" on a previously surveyed point or a known survey control point.

PN 022543-544E (10/14)