

Probabilistic Methods for Robotics in Agriculture

Santosh Hiremath

Thesis committee

Promotors

Prof. Dr A. Stein

Professor of Mathematical and Statistical Methods for Geodata
ITC, University of Twente, Enschede

Prof. Dr C.J.F. ter Braak

Personal chair in the Mathematical and Statistical Methods Group
Wageningen University

Co-promotor

Dr G.W.A.M. van der Heijden

Manager and Senior Scientist, Biometris
Wageningen University and Research Centre

Other members

Prof. Dr C.A. Glasbey, BioSS, Edinburgh, UK

Prof. Dr E.J. van Henten, Wageningen University

Prof. Dr P.P. Jonker, Delft University of Technology

Prof. Dr C.H. Slump, University of Twente, Enschede

This research was conducted under the auspices of the Graduate School of
Production Ecology & Resource Conservation (PE & RC)

Probabilistic Methods for Robotics in Agriculture

Santosh Hiremath

Thesis

submitted in fulfilment of the requirement for the degree of doctor
at Wageningen University
by the authority of the Rector Magnificus
Prof. Dr M.J. Kropff,
in the presence of the
Thesis Committee appointed by the Academic Board
to be defended in public
on Thursday 3 October 2013
at 11 a.m in the Aula.

Santosh Hiremath
Probabilistic methods for robotics in agriculture,
110 pages.

Thesis, Wageningen University, Wageningen, NL (2013)
With references, with summaries in Dutch and English

ISBN 978-94-6173-641-3

To my beloved parents

Summary

Automation of agricultural tasks is helpful for sustainable and efficient food production. For autonomous vehicles to be effective in their agriculture tasks they need to deal with uncertainty in the environment. This can be achieved by using probabilistic methods which allow an explicit representation and precise manipulation of the uncertainty. Here the word uncertainty is used in a wide sense. The main goal of this thesis is to test the efficacy of probabilistic methods for autonomous robot applications in agriculture. We focus on two agricultural tasks. The first task is the automatic weed detection in a grassland where the thesis looks at the detection of a common weed named *Rumex obtusifolius* (Rumex) in pastures. The second task is autonomous navigation of a robot in a Maize field where the thesis investigates the use of probabilistic methods for autonomous row following.

Rumex is a common weed that proliferates at the cost of grass and is difficult to control. In recent times, however, robotic systems have been developed for the mechanical control of Rumex and also other weeds to meet the sustainability demands of precision agricultural practices. **Chapter 2** takes a critical look at the suitability of image analysis methods for detecting Rumex in a grassland. It starts by considering a recently developed robot with a real-time vision system (referred as the current system) for detecting Rumex with the objective of improving its detection accuracy. A careful analysis of the method revealed that the Fourier texture feature used in the current system is equivalent to local variance of the pixel intensities. Local variance is a simple descriptor not capable of capturing the uncertainties of the textures of Rumex and grass. The similarity of colour between the weed and grass along with the variety of shapes and sizes of the Rumex as well as fluctuating densities of grass increase the uncertainty of the scene. The study explores three sets of most commonly used sets of textures features: local variance, features derived from Laws' filter masks and features derived from on grey level co-occurrence matrix (GLCM). The three sets of features are tested for their ability to detect Rumex from grass using a feature selection method. The distance between algorithm-determined centre and the centre of the hand segmented image of Rumex was used as the error measure to evaluate the features. The feature selection and validation was carried out using 24 images. This yielded three

GLCM features. The segmentation algorithm with the optimal features was further tested on another set of 92 images taken under different illumination and with varying weed size. The detection accuracy of the new algorithm was found to be 90% along with an average error of 141 mm in determining the location of the taproot of Rumex compared with 308 mm of the current algorithm. The chapter demonstrates that a probabilistic representation of the interactions between the pixels while the others are simple statistical summaries.

Chapter 3 takes a Bayesian approach for the segmentation of Rumex. The uncertainties in the image is explicitly defined by describing the spatial interactions between the pixels as Gibbs distribution by means of Markov Random Field (MRF) theory. . The image is thus modelled as a Gaussian MRF (GMRF) where the parameters of the model are the texture features. The segmentation of Rumex from grass is formulated as a Maximum a Posteriori (MAP) inference problem within the MAP-MRF framework which is solved using Graph Cuts. Defining the Gibbs distribution is done by specifying the energy function which in turn is accomplished by determining single site and pair-site clique potentials. Single site clique potential is a function expressing the relationship between pixel intensity and its expected label whereas the pair-site clique potential is the energy a function expressing the relationship between the labels of two adjacent pixels. The MAP solution is found by minimizing the energy function using Graph Cuts. This new algorithm was tested over the same set of 92 image as the GLCM algorithm and yielded a detection accuracy of 97.8% and average error of 56mm in determining the location of the taproot of Rumex. The algorithm was able to process up to five images per second and thus can be applied in real-time. We concluded that the Bayesian approach developed in this chapter provided an important extension to existing segmentation approaches.

Chapter 4 looks at autonomous navigation of a robot in Maize field where the robot has to navigate through a corridor formed by two plant rows, detect the end of the rows, navigate the headland and turn into another corridor using camera as the measurement sensor under natural conditions. A particle filter based navigation method is developed for row following in a field where the robot-environment state is constantly tracked. The robot-environment state consists of six parameters namely robot heading, lateral deviation of the robot, distance between the rows, width of the rows, end of left row and end of right row. The particle filter represents the robot-environment state as a probability distribution that is updated as new measurements are acquired. It consists of a prediction-correction cycle of that it first predicts the possible particles using the motion model, second the particles are weighted by the measurement model according to their veracity with the measurement. Third they are re-sampled to filter out particles with low weights. As the robot uses only the

camera for taking measurements, the measurement model here is the camera model. Clever parametrization of the camera view into the six parameters results in a novel measurement model. We map the particles into measurement space by constricting a model-image from a particle and comparing it with the observed image to determine the likelihood of the particle. The robustness of the new navigation algorithm is demonstrated through extensive field experiments with different row patterns of varying plant sizes and also under diverse lighting conditions. This chapter concludes that the particle filter is a robust method for row following in an agricultural field.

Chapter 5 extends the navigation method developed in the previous chapter to work with a LIDAR instead of a camera. It is assumed that the LIDAR has a rectangular perspective field with the sensor at the centre. The perceptive field is divided into three striped regions according to the density of the plant material. The probability of a laser beam hitting plant material (thereby yielding an observation) in any of these regions is determined by the density of plant material present. The probability of an observation in each region is characterized by different probability distributions. This probabilistic measurement model is used in the particle filter to compute the likelihood of the particles. As a result the particle filter estimates the robot-environment state using the data from the LIDAR and the robot can navigate the rows using the LIDAR. In conclusion, the chapter proposes probabilistic LIDAR model for row following in an agricultural field. Finally, in chapter 6 the results of our work are put in perspective and some future work is suggested.

Samenvatting

Automatisering van landbouwwerkzaamheden is nuttig voor een duurzame en efficiënte voedselproductie. Autonome voertuigen dienen om te kunnen gaan met onzekerheid in de omgeving om effectief hun landbouwkundige taken te kunnen uitvoeren. Dit kan worden bereikt door gebruik te maken van probabilistische methoden, die een expliciete weergave en exacte behandeling van de onzekerheid mogelijk maken. Hierbij wordt de term onzekerheid in brede zin gebruikt. Het doel van dit proefschrift is om de effectiviteit van probabilistische methoden te testen voor autonome robottoepassingen in de landbouw. We richten hierbij de aandacht op twee landbouwkundige taken. De eerste is automatische onkruiddetectie in een grasland, waarbij in het proefschrift gekeken wordt naar de detectie van een algemeen gangbaar onkruid, genaamd *Rumex obtusifolius* (zuring) in weiland. De tweede taak is de autonome navigatie van een robot in een maisveld, waarbij in het proefschrift onderzoek wordt gedaan naar het gebruik van probabilistische methoden voor het autonoom volgen van de plantenrij.

Zuring is een algemeen voorkomend onkruid dat zich verspreid ten koste van gras en dat moeilijk te beheersen is. Recentelijk zijn er echter robotsystemen ontwikkeld voor mechanische verwijdering van zuring en ook andere onkruiden om te voldoen aan de duurzaamheidseisen van de precisielandbouw. **Hoofdstuk 2** werpt een kritische blik op de geschiktheid van diverse beeldanalyse methoden voor de detectie van zuring in grasland. Het uitgangspunt is een recent ontwikkelde robot met een realtime beeldanalyse systeem (het huidige systeem) voor detectie van zuring met als doel het verbeteren van de nauwkeurigheid van detectie. Een zorgvuldige analysemethode toont aan dat het Fourier textuur kenmerk dat gebruikt wordt in het huidige systeem equivalent is aan lokale variantie van de pixelintensiteit. Lokale variantie is een eenvoudige maat die niet in staat om de textuuronzekerheden van zuring en gras goed te beschrijven. De gelijkenis in kleur tussen onkruid en gras samen met de variatie in vorm en grootte van zuring alsook de fluctuerende dichtheid van gras verhoogt de onzekerheid in de waargenomen scene. De studie verkent drie typen veel gebruikte textuurmaten: lokale variantie, kenmerken afgeleid uit Laws filters en kenmerken afgeleid uit de zogeheten grey level co- occurrence matrix (GLCM). Deze drie typen kenmerken

worden getest op hun geschiktheid om zuring van gras te onderscheiden, gebruikmakend van een kenmerkselectie methode. De afstand tussen het hart van de zuringplant, zoals gevonden met de geautomatiseerde methode en zoals gevonden in het met de hand gesegmenteerde beeld, is gebruikt als foutmaat om de kenmerken te vergelijken. De kenmerkselectie en validatie zijn uitgevoerd op 24 beelden. Dit resulteerde in drie GLCM kenmerken. De segmentatiemethode met de optimale set kenmerken is verder getest met een andere verzameling van 92 beelden, die onder wisselende belichting zijn opgenomen en waarvan de onkruidgrootte varieert. De nauwkeurigheid van de detectie met een nieuwe algoritme was 90% met een gemiddelde fout van 141 mm in het bepalen van de plek van de penwortel van zuring, vergeleken met 308 mm met de huidige methode. Het hoofdstuk laat zien dat een probabilistische representatie van de interactie tussen de beeldpunten beter is dan de ander twee methoden, die een simpele statistische beschrijving zijn.

In **hoofdstuk 3** wordt een Bayesiaanse aanpak gevolgd voor de segmentatie van zuring. De onzekerheden in het beeld worden expliciet gedefinieerd door gebruik te maken van Markov Random Field (MRF) theorie. Hierbij wordt het beeld gemodelleerd als een Gaussisch MRF (GMRF) waarbij de parameters in het model de textuurkenmerken zijn. Het onderscheid tussen zuring en gras is geformuleerd als een Maximum a posteriori (MAP) beslissingsprobleem binnen een MAP-MRF raamwerk. Dit wordt geoptimaliseerd met behulp van Graph Cuts. De Gibbs verdeling wordt gedefinieerd door de energie-functie te specificeren die wordt bepaald door de clique potentialen van een enkelvoudige cel (pixel) en van paarsgewijze cellen. Een clique potentiaal van een enkelvoudige cel is een functie die de relatie uitdrukt tussen de pixelintensiteit en het verwachte label, terwijl de clique potentiaal van de paarsgewijze cellen de energie is van een functie die de relatie uitdrukt tussen de labels van twee naburige pixels. De MAP oplossing wordt gevonden door de energiefunctie te minimaliseren door gebruik te maken van Graph Cuts. Dit nieuwe algoritme werd getest op dezelfde set van 92 beelden als gebruikt voor het GLCM algoritme en het leverde een detectie nauwkeurigheid van 97.8% op, met een gemiddelde fout van 56 mm in de plaatsbepaling van de penwortel van zuring. Het algoritme was in staat om tot 5 beelden per seconde verwerken en kan dus real-time worden toegepast. We concludeerden dat de Bayesiaanse benadering in dit hoofdstuk een belangrijke aanvulling is op de bestaande segmentatie benaderingen.

Hoofdstuk 4 bestudeert de autonome navigatie van een robot in een masveld. De robot moet door een corridor rijden die wordt gevormd door twee plantenrijen, hij moet het eind van de rijen detecteren, op de kopakker navigeren en een andere corridor indraaien, gebruik makend van een camera als meetsensor onder natuurlijke omstandigheden. Een navigatie methode is ontwikkeld op basis van een partikel filter (particle filter of sequential Monte Carlo) voor

het volgen van de rijen in het veld, waarbij de robot-omgeving toestand continue wordt gevolgd. De robot-omgeving toestand bestaat uit zes parameters, namelijk de robotrichting, de zijwaartse afwijking van de robot, de afstand tussen de twee rijen, de breedte van de rijen, het einde van de linker rij en het einde van de rechter rij. Het partikel filter geeft de robot-omgeving toestand weer als een kansverdeling, die wordt bijgewerkt als nieuwe metingen worden verkregen. Het bestaat uit een predictie-correctie cyclus die ten eerste de mogelijke partikels voorspelt met behulp van een bewegingsmodel, ten tweede de partikels weegt met een meting model op basis van hun waarheidsgetroouwheid met de meting. Ten derde worden ze geresampled om partikels met lage gewichten uit te filteren. Omdat de robot alleen de camera gebruikt voor het nemen van metingen, is het meetmodel hier het camera model. Een slimme parameterkeuze van het camerabeeld in de zes parameters resulteert in een nieuw meetmodel. We brengen de partikels over naar de meetruimte door uit elk partikel een modelbeeld te construeren en dit te vergelijken met het waargenomen beeld om de waarschijnlijkheid (likelihood) van het partikel te bepalen. De robuustheid van het nieuwe navigatie-algoritme wordt aangetoond door middel van uitgebreide veldproeven met verschillende rij patronen met verschillende plantgrootte en ook onder uiteenlopende lichtomstandigheden. Dit hoofdstuk concludeert dat het partikel filter een robuuste methode is voor het volgen van een rij in een akker.

Hoofdstuk 5 breidt de navigatie methode die werd ontwikkeld in het vorige hoofdstuk uit met een lidar (afstandssensor) in plaats van een camera. Aangenomen wordt dat de lidar een rechthoekig veld waarneemt met de sensor in het midden. Het waargenomen veld wordt verdeeld in drie gestreepte gebieden overeenkomend met de dichtheid van plantmateriaal. De kans dat een laserstraal plantmateriaal raakt (hetgeen resulteert in een waarneming) in elk van deze gebieden wordt bepaald door de dichtheid van het aanwezige plantmateriaal. De waarschijnlijkheid van een waarneming in elke regio wordt beschreven door middel van verschillende kansverdelingen. Dit probabilistisch meetmodel wordt gebruikt in het partikel filter om de waarschijnlijkheid van de partikels te berekenen. Hierdoor schat het partikel filter de robot-omgeving toestand gebruikmakend van de meetgegevens van de lidar en kan de robot tussen de rijen navigeren met behulp van de lidar. Samengevat stelt het hoofdstuk een probabilistisch lidar model voor om rijen te volgen in een akker. Ten slotte worden in **hoofdstuk 6** de resultaten van ons werk in perspectief gezet en worden enkele toekomstige onderzoekrichtingen voorgesteld.

Table of Contents

Summary	iii
Samenvatting	vii
1 Introduction	1
1.1 Scope and Motivation	2
1.1.1 Probabilistic and Stochastic Methods	2
1.1.2 Automatic Detection of <i>Rumex obtusifolius</i>	7
1.1.3 Autonomous Navigation in Agriculture	8
1.2 Research Objectives	11
1.3 Outline of the Thesis	12
2 Textures for segmentation of Rumex	13
2.1 Introduction	14
2.2 Materials and Methods	16
2.2.1 Images	16
2.2.2 Texture features to detect <i>Rumex obtusifolius</i>	16
2.2.3 Feature selection and validation	18
2.2.4 Construction of feature sets	19
2.2.5 Training the classifier	19
2.2.6 Error measure	20
2.2.7 Validation	20
2.2.8 Segmentation Algorithm	20
2.3 Results	22
2.3.1 Performance of feature sets	22
2.3.2 Performance of the segmentation algorithm	24
2.4 Discussion	25
2.5 Conclusions	28
3 Bayesian segmentation of Rumex	31
3.1 Introduction	31
3.2 Markov Random Fields	34
3.2.1 MAP-MRF Framework	36

3.2.2	Minimizing via Graph Cuts	36
3.3	Image Model	37
3.4	Supervised Segmentation	38
3.4.1	Images	38
3.4.2	Feature Selection	39
3.5	Segmentation Performance	42
3.6	Discussion and Future Work	44
4	Navigation using camera	47
4.1	Introduction	48
4.2	Image Based Particle Filter	50
4.2.1	Local World	50
4.2.2	Motion Model	51
4.2.3	Measurement Model	52
4.2.4	Fuzzy Membership	53
4.2.5	Attenuation Factor	53
4.3	Experiments and Results	55
4.3.1	Robot	55
4.3.2	Calibration	56
4.3.3	Field Tests	59
4.4	Discussion	61
4.5	Conclusions	64
5	Navigation using LIDAR	65
5.1	Introduction	65
5.2	Materials and Methods	68
5.2.1	Maize Field	68
5.2.2	Robot	68
5.2.3	Laser Range finder (LIDAR)	68
5.2.4	Local World	69
5.2.5	Particle Filter	71
5.2.6	Motion Model	72
5.2.7	Measurement Model	72
5.2.8	Attenuation Factor	76
5.2.9	Experimental Data	76
5.3	Experiments and Results	78
5.4	Discussion	82
5.5	Conclusions	84
6	General Discussion	87
6.1	Automatic Detection of Weed	87
6.2	Autonomous Navigation in an Agricultural Field	90
6.3	Future Research Possibilities	92

Acknowledgements	104
List of Publications	106
Curriculum Vitae	107

Chapter 1

Introduction

There is a rapid increase in the demand for food production due to the growing population. Sustainable agriculture is the only feasible method of meeting this demand without causing adverse environmental impact. Precision agricultural practices and organic farming methods are cornerstones of sustainable agriculture. These practices, however, cannot meet the growing demand without the use of autonomous machines because they are labour intensive and inefficient. Autonomous operation of robotic systems in an agricultural environment is a difficult task due to the inherent uncertainty in the environment. The robot is in a dynamic, non-deterministic and semi-structured environment with many sources of noise. For instance, a robot navigating the rows autonomously in a maize field should be able to accurately estimate its position along with other parameters like the distance between the rows, the width of the rows, their relative position whilst they are constantly changing. It should also be capable of detecting the end of the row when it is in sight and differentiate it from the gaps due to missing plants. Likewise, a weed control robot should be able to identify weed(s) from the plants. Often the colour, shape, texture and size of the weed is similar to the plants making it difficult to differentiate between the two. Apart from this identification problem, the robot should also be able to estimate the position of the weed to a high degree of precision for effective treatment without damaging the neighbouring plants. These problems are further accentuated by the fact that the sensors have some inherent limitation. They are limited in their range and resolution and sensor measurements can be noisy. These hardware limitations also extend to the robot. Besides, there is also a need for the robot to operate in real-time. Due to these constraints the traditional deterministic approach of reasoning will not work for automation in an agricultural environment. The complexity of the environment is so high that there are often more exceptions than rules and a deterministic, rule based approach of reasoning becomes intractable.

A novel approach of reasoning in such complex and unstructured environment is by means of probabilistic methods. Probabilistic robotics is an

approach to robotics using these methods. In probabilistic robotics, information is represented as a probability distribution and is manipulated using probability calculus. In doing so the uncertainty in the environment and in the system is explicitly represented and managed. Thus, a row following robot will maintain and track a distribution of plausible values of the various parameters and a weed control robot will have a distribution of parameters characterizing shape, colour, size and texture of both weed and plant allowing it to differentiate between the two easily. This way of managing uncertainty explicitly provides a robot the flexibility to delay its decision and also recover from an incorrect one. According to Thrun et al. (Thrun et al., 2005) “A robot that carries a notion of its own uncertainty and that acts accordingly is superior to one that does not”.

1.1 Scope and Motivation

The central premise of the thesis is to explore the efficacy of probabilistic methods for agricultural applications. We focus on two major applications: automatic weed detection and autonomous robot navigation. In this vein, the thesis can be divided into two parts. The first part consists of automatic weed detection where two different probabilistic vision-based methods are applied for identifying and controlling *Rumex obtusifolius* L. (Rumex) in grasslands. The second part consists of autonomous robot navigation in a maize field where probabilistic methods are developed to steer the robot autonomously using a camera as well as a laser scanner. The remainder of the chapter consists of a brief introduction of probabilistic methods relevant to the thesis followed by reviews on the existing work in weed detection and autonomous robot navigation. Finally, the chapter ends by specifying the research objectives and providing an outline of the thesis.

1.1.1 Probabilistic and Stochastic Methods

In image analysis it is necessary to use contextual information to identify the object(s) or interpret the scene in the image. This contextual information is the spatial interaction between the pixels, textures or image features. These interactions are stochastic in nature due to several reasons such as noise present in the imaging process, uneven distribution of ambient light, varying reflectance properties of different surfaces, different poses of the same object, irregular object boundaries and occlusions. As a result probabilistic methods are well suited to model the spatial interaction in the images.

One of the most widely used method for expressing spatial interaction between pixels in the image is Grey Level Co-occurrence matrix (GLCM) (Haralick et al., 1973). It encodes the interaction between the pixels by recording

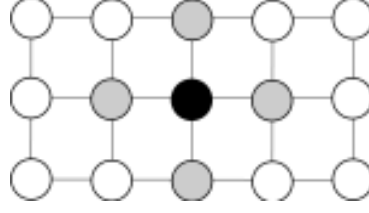


Figure 1.1: Conditional independence property of a MRF. The graph $G = (\mathcal{N}, \mathcal{E})$ consists of three disjoint set of nodes namely, interior node X in black, its neighbours N_X in grey and the exterior nodes $\bar{N}_X = \mathcal{N} - \{X, N_X\}$ in white. X is conditionally independent from $\bar{N}_X$ given N_X .

how often different combination of pixel values occur in an image. That is, each entry in the matrix is the relative probability of co-occurrence of a pair of pixel values. The neighbourhood of the interactions is defined by specifying the distance and the orientation of the pixel pairs under consideration. For instance, a neighbourhood defined as $(d, h) = (1, 0)$ where d is the distance and h is the orientation means that, for a given pixel its neighbourhood consists of pixels immediately to its right or left. In general, the three parameters that characterize the GLCM are the number of grey levels G in the image, the distance between the pixel pair d , and the orientation of the pixel pair h . The value of G is a trade-off between computational time and accuracy, while values of d and h determine the coarseness and directionality of the texture content respectively.

Another popular method used for modelling spatial context in natural images is Markov Random Field (MRF) (Efros and Leung, 1999; Zalesny and Gool, 2001) that helps to deal with the uncertainties in the image by defining the spatial interactions between the pixels as a joint probability distribution (Li, 2001; Hammersley and Clifford, 1971). MRF is an undirected graphical model of a joint probability distribution. It consists of a graph $G = (\mathcal{N}, \mathcal{E})$ where $\mathcal{N}$ are the set of nodes and $\mathcal{E}$ are the edges between them. The nodes represent the random variables and edges specify a conditional independence relation between them. This conditional independence property is called the Markov property which states that the value of any random variable is independent of all other random variables given its neighbours which is illustrated in Figure 1.1. The interior node X in black is independent from the exterior nodes in white given its neighbourhood nodes N_X in grey. More formally it is given by $P(X | \mathcal{N} - X) = P(X | N_X)$. As a result of this property, one can easily specify the conditional probability of any random variable(s) given its neighbourhood set.

The Hammersley-Clifford theorem states that the joint distribution over all the random variables of an MRF is a Gibbs distribution (Hammersley and

Clifford, 1971) given by

$$P(X) = \frac{1}{Z} \prod_C \phi_C(X_C) \quad (1.1)$$

where $\phi_C(X_C)$ is a potential function corresponding to the clique C and X_C are the nodes that belong to the clique. Here a clique is a set of nodes in which every pair is connected. Z is the normalizing constant and is called the partition function. The clique potentials can be further decomposed to

$$\phi_C(X_C) = e^{-\sum_{c \in C} V_c(X)} \quad (1.2)$$

where $U(x) = \sum_{c \in C} V_c(X)$ is called the energy function defined over the cliques. The equality established between MRF and the Gibbs distribution by the Hammersley-Clifford theorem provides a convenient way of specifying the joint distribution for an MRF by simply defining the clique potential.

Many problems in computer vision and image analysis such as image restoration, reconstruction, segmentation and interpretation can be posed as a labelling problem where the solution is a set of desired labels for the pixels. This problem however entails a lot of uncertainty due to which exact solutions are generally not feasible and one has to resort to inexact but optimal solutions. Thus the labelling problem becomes an optimization problem. In order to pose a labelling problem as an optimization problem an optimality criteria and a suitable objective function has to be defined such that finding the extremum of the objective function leads to the desired pixel labels. Various optimality criteria include Maximum Likelihood (ML), Maximum Entropy and Maximum a Posterior (MAP). It is well known that when both the prior and the likelihood distributions are known the best result is achieved by optimizing the Bayesian criteria (Li, 2001) such as MAP. In the MRF context, the MRF model of an image forms the prior, the image is the data from which the likelihood can be constructed. In a MAP-MRF framework, once the MRF is constructed the clique potentials which encodes the relationship between variables need to be defined. The appropriate energy function required is determined by the desired solution for the problem. This is application specific and often requires insight into the problem at hand as well as trial and error. However, there are several standard energy functions for different kind of problems in computer vision, the details of which can be found in (Li, 2001). After the energy function is defined an optimization algorithm is used to find the optimal solution. These optimization algorithms include Iterated Conditional Mode (ICM) (Besag, 1986), Simulated Annealing (SA) (Kirkpatrick et al., 1983) and Graph Cuts (Boykov et al., 2001).

Probabilistic methods are also used in autonomous robot navigation. An autonomous robot interacts with the environment in one of two ways namely, perception and action. Robot perception is the process in which the robot

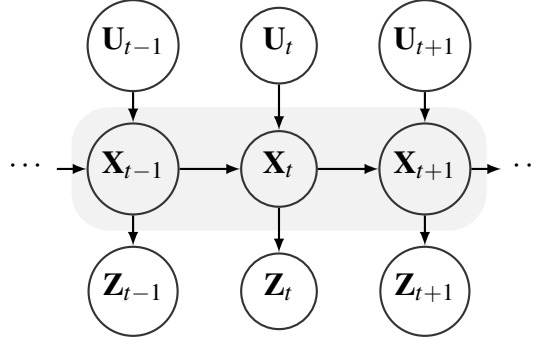


Figure 1.2: Graphical representation of a typical state estimation problem in robot localization. The nodes labelled X_t , Z_t and U_t represent the state, measurement and controls at time t , respectively.

uses its sensors to obtain information about the environment. The information so obtained at any given time t is called measurement or observation and is denoted by Z_t . Robot action is a process in which the robot influences the state of its environment through its actuators; the controls used for performing the action is denoted by U_t . Probabilistic methods are based on the idea that evolution of the robot-environment system is governed by probabilistic laws and use probability calculus to estimate the state X_t of the system at time t from the sensor measurements Z_t . Rather than a single estimate of X_t , probabilistic methods maintain a distribution of possible hypothesis of the current state along with their corresponding probabilities based on all the measurements and the controls up to time t . This probability distribution is denoted by $bel(X_t) = P(X_t|Z_{1:t}, U_{1:t})$ and is also called belief or posterior distribution which is updated constantly at each time step.

A pictorial representation of the evolution of the system in time is shown in Figure 1.2. The probabilistic relationship between the various variables is as follows. The state at time t is dependent on the state at time $t - 1$ and the control U_t and the measurement Z_t depends on the state at time t . This temporal model assumes that the state at any time is a sufficient summary of all that has happened up to that time and the next state depends on the current state of the system and the control applied. As a result, the state transition probability is given by $P(X_t|X_{t-1}, U_t)$ and the measurement probability is given by $P(Z_t|X_t)$. These probabilities are used in a Bayes filter to infer the belief at each time step.

Bayes filters make use of the Bayes theorem for estimating the belief at current time step $bel(X_t)$ from the belief at previous time step $bel(X_{t-1})$. It consists of a prediction-correction cycle. In the prediction step, the algorithm predicts the state of the system $\overline{bel}(X_t) = P(X_t|Z_{1:t-1}, U_{1:t})$ from the belief of previous time step and the state transition probability which is subsequently

updated to $bel(X_t)$ in the correction step using the measurement probability. Bayes Filter is a general algorithm and has two types of implementations. One is the parametric implementation such as the Kalman Filter (KF) and the Extended Kalman Filter (EKF) and the other is the non-parametric implementation such as The Histogram filter (HF) and Particle Filter (PF). The rest of the section briefly explains KF and PF in order to illustrate the difference between a parametric and a non-parametric implementation.

The KF is a parametric implementation of the Bayes filter as it assumes that the belief has a specific functional form. It also assumes that the system is linear and the noise can be characterized by a Gaussian distribution. In other words, the system is governed by a set of linear equations given by

$$\begin{aligned} X_t &= AX_{t-1} + BU_t + \varepsilon_t \\ Z_t &= CX_t + \delta_t \end{aligned}$$

where A, B and C are matrices and ε_t and δ_t are Gaussian noise. The first equation characterizes the state transition and the second equation characterizes the measurement system. The belief at time t , $bel(X_t)$, is represented by a set of Gaussian parameters μ_t and Σ_t which are constantly updated as new measurement Z_t arrives. The EKF is a nonlinear version of the KF in that the state transition and the measurement system are governed by non-linear equations. Details of this algorithm can be found in (Thrun et al., 2005).

A PF is a non-parametric implementation of the Bayes filter. Unlike the KF (or other parametric filters), the PF does not impose any specific form on the belief $bel(X_t)$. Instead the belief is represented by a set of m samples called particles which are denoted by $\mathcal{X}_t = \{X_t^{(1)}, X_t^{(2)}, \dots, X_t^{(m)}\}$. Each particle is a concrete instantiation of the state of the system at time t . The PF algorithm is shown in Table 1.1. Lines 2-6 are the prediction step where new particles are sampled based the state transition probability (line 3). Subsequently, their weights w_t is calculated based on the measurement probability $P(Z_t|X_t)$ (line 4). The weight w_t is an indication of how well the predicted particle 'agree' with the measurement Z_t . The predicted particles along with their weights constitute the temporary particle set $\tilde{\mathcal{X}}_t$. These particles are re-sampled based on their weights to get $\mathcal{X}_t$ that represents the belief at time t (lines 7-10). The resampling step, also referred to as importance sampling, ensures that the particles with high weight—which best represent the current state of the system—are retained while those with low weight are filtered out. The non parametric nature of PF means that it can be used in situation where the system is non linear and also when noise has a complex multimodal distribution.

In this thesis we introduce the aforementioned methods to problems specific to agricultural domain. In particular, we look at two different problems namely automatic detection of *Rumex obtusifolius* in a grassland and

Table 1.1: Particle Filter Algorithm

1	ParticleFilter($\mathcal{X}_{t-1}, U_t, Z_t$)
2	for $k = 1$ to m do
3	sample $X_t^{(k)} \sim P(X_t X_{t-1}^{(k)}, U_t)$
4	$w_t^{(k)} = P(Z_t X_t^{(k)})$
5	add $(X_t^{(k)}, w_t^{(k)})$ to $\tilde{\mathcal{X}}_t$
6	endfor
7	for $k = 1$ to M do
8	draw i with probability $\propto w_t^{(i)}$
9	add $X_t^{(i)}$ from $\tilde{\mathcal{X}}_t$ to $\mathcal{X}_t$
10	endfor
11	return $\mathcal{X}_t$

autonomous navigation of a robot in a maize field. The next two sections provide a brief introduction to these two applications motivating the research objectives.

1.1.2 Automatic Detection of *Rumex obtusifolius*

Rumex obtusifolius (broad-leaved dock or Rumex) is a common weed found in grassland. It is considered troublesome in dairy farming systems because it proliferates at the cost of grass. A widely used method of controlling weeds is by application of herbicides. This, however, is not sustainable as herbicides are found to cause ground water pollution (Kempenaar et al., 2007) and force selective bias towards herbicide-resistant weeds. There are several non-chemical methods for controlling weeds, like crop rotation, manual removal, thermal and biological control but these methods are labour intensive, non-scalable, and expensive both in time and costs (Bond and Grundy, 2001; van Evert et al., 2009). Frequent removal of the weed *per se* is insufficient to stop it from spreading (Niggli et al., 1993). To prevent its regrowth, destruction of the taproot to a soil depth of 20 cm is necessary (Zaller, 2004). As a result, many farms are often severely infested with *R. obtusifolius* (Bohm and Vershwele, 2004; Van Middelkoop et al., 2005).

Among the non-chemical control methods, robotic systems to mechanically remove weed have great potential for its automatic detection and removal. There has been a lot of research on automatic detection of weed based on computer vision (Slaughter et al., 2008). Many methods focus on the detection of weeds in row crops and are designed to detect weeds in soil background (Meyer et al., 1998; Aitkenhead et al., 2003; Tang et al., 2003; Guijarro et al., 2011). They use the colour contrast between soil and weed for

detection. Detection of broad-leaved weed in a grass background is addressed mainly by exploiting the difference in the texture of grass and weed; grass has a ‘rougher’ texture compared to weed. For instance, (Ahmad et al., 1997; Ahmad et al., 1998) developed a simple texture feature based on uniformity analysis to differentiate weed from grass in lawns using grey scale images. (Otsuka and Taniwaki, 1996; Otsuka and Taniwaki, 1999) used local variance (another measure of local uniformity) for weed detection. A more complex texture measure based on edge strength was developed by (Watchareeruetai et al., 2006); this measure exploits the fact that the density of edges of grass is higher than the density of edges of broad-leaved weeds. They extend their method further (Watchareeruetai et al., 2008) by including the texture measure in a classifier based on support vector machines. Ahmad et al. (Ahmad et al., 1999) used more sophisticated texture features based on Gray level Co-occurrence Matrix (GLCM) to differentiate weed from grass. Other methods include a texture measure based on Gabor Wavelets developed by (Tang et al., 2003) which may be some of the most complex texture features developed for the detection of weed from grass.

These systems, however, are not suitable for detection of Rumex in grasslands for several reasons. Grass and weed have the same colour, hence methods based on colour contrast are not applicable. Because of the high density of the grass in a pasture, methods such as uniformity analysis and local variance will fail as the mean and range values of weed and grass tiles overlap considerably. The veins and ribs of Rumex result in strong edges and so does damage to the leaf. This makes the use of features based on edge strength problematic. GLCM texture features are promising for correcting this problem, but which features are important and which are redundant is not yet clear. Further, the parameter settings required to compute the co-occurrence matrix so that the resulting texture features can discriminate between weed and grass efficiently are not known. The computational cost of the method proposed by (Gebhardt and Kuehbauch, 2007) precludes (at present) real-time use in a robotic system. Using wavelets is unlikely to yield an improvement over using local power spectrum, as discussed in (van Evert et al., 2009). All these shortcomings indicate that more research was required for developing a robust, real-time vision-based method for detecting Rumex in a grassland.

1.1.3 Autonomous Navigation in Agriculture

Research in autonomous navigation in agriculture dates back as far as 1950 when mechanical guidance systems were used for steering a vehicle like a tractor in the field (Richey, 1959; Rushing, 1971). The vehicle was fitted with tactile sensors and depended on fixed guidance features like field drain, ridges, furrows and crop rows for determining the relative position. These systems,

however, were not feasible for many reasons such as the absence of signal due to missing crops, incorrect signals due to lack of structure or damaged crops. Also, the infrastructure required to setup the system was expensive and not scalable. This paved the way to artificial beacon based navigation systems which made use of non-tactile sensors such as lasers or radars along with three or more (fixed) beacons to determine the position of the vehicle by means of triangulation. Complications were likely to arise if the communication link between the sensor and the beacons were broken or when the correspondence between an observation and a beacon could not be established. Other problems of such systems included low rate of observations, inflexibility and high setup cost.

The advent of Global Position System (GPS) technology in the 1990s addressed many of the shortcomings of beacon-based navigation systems and helped to accelerate the research in autonomous navigation in agriculture, forestry and horticulture (Larsen et al., 1994; Gan-Mor et al., 1996; Bell, 2000; Stoll and Dieter Kutzbach, 2000; Heidman et al., 2002). Many companies started producing farm machines fitted with Real-time Kinematic GPS and auto steering option like the John Deere Model 7800 tractor. The widespread use of GPS technology in agriculture included applications like sowing, tilling, planting, cultivating and harvesting. Despite the success, GPS-based navigation had some severe limitations. Firstly, the GPS receiver requires direct line of sight to the satellites all the time to provide consistent position accuracy. This, however, is problematic in a complex agricultural environment where the signals can be obstructed by trees and other obstacles. The signal quality can also be affected by poor weather conditions. The situation is further exasperated by the fact that the level of precision necessary for an agricultural task requires access to more than 5 satellites. Secondly, the inherent delay in determining the location pose a significant challenge to the control system. Thirdly, GPS-based navigation is not truly autonomous because it depends on remote devices like satellites or base stations.

Vision Based Navigation

During the 1990s, advances in computer processing power sparked renewed interest in the use of machine vision sensors for autonomous navigation in agriculture and in other areas. Vision-based methods use imaging cameras (monochrome and colour) to sense the world and image processing techniques to extract information about the relative position and heading of the vehicle. There are many vision-based navigation systems employed in agriculture. Good review papers on different autonomous navigating systems in agriculture can be found in (Wilson, 2000; Keicher and Seufert, 2000; Reid et al., 2000; Li et al., 2009). In most of these systems, the image processing methods used depend on segmenting the crop rows from the background

and extracting the corresponding line features which are used to determine the location and heading of the vehicle. The algorithms used for extracting line features include row-fitting algorithms like linear regression (Billingsley and Schoenfisch, 1997) and *k-means* (Han et al., 2004). Perhaps, one of the most commonly used methods to extract line features is the Hough Transform (Hough, 1962). Some examples include (Reid and Searcy, 1986; Fujii and Hayashi, 1989; Marchant and Brivot, 1995; Marchant, 1996; Hague and Tillett, 1996; Åstrand, 2005).

Navigation Based on Range Sensors

Vision based methods are sensitive to lighting conditions. Due to the large variation in ambient light in an outdoor environment like an agriculture field, most systems need frequent calibration to the specific operating conditions. Laser range finder (LIDAR) technology does not suffer from the effects of ambient lighting conditions and thus can be more reliable in an agricultural environment. Until recently, LIDARs were not preferred in agricultural domain due to their high cost but their cost is reducing rapidly and the phase of research in LIDAR-based navigation method has picked up (Ahamed et al., 2004). Barawid et al., (Barawid Jr. et al., 2007) developed a real-time guidance system for navigating an autonomous vehicle in an orchard based on LIDAR. The Hough Transform is used to extract plant rows for navigating the vehicle. They report that the method is restricted to straight line recognition and thus will have difficulty in curved rows. Another disadvantage of the method is that the vehicle will loose track if the Hough Transform fails to extract the correct plant rows. LIDAR has also been used for obstacle detection and avoidance during navigation as in the case of (Subramanian et al., 2006). More recently (Weiss and Biber, 2011) have developed a 3D LIDAR based navigation method where they use a statistical model for detection of rows of maize plants. The LIDAR acquires a 3D point cloud data which is processed to remove the points corresponding to the ground. Then a statistical model identifies clusters of points that represent the plants. While the results are promising the method will not be easily scalable to other plants because the statistical model is specific to maize plants including its shape and size. Moreover the system is specifically designed for plant phenotyping which imposes restrictions on the operating conditions like size and age of the plant and speed of the robot.

Irrespective of the type of sensors, it is necessary to represent and manage the uncertainty in the environment explicitly. Hague et al. (Hague and Tillett, 1996) were one of the first to employ a probabilistic method—the Extended Kalman Filter (EKF)—for autonomous navigation of an agricultural robot. Another application of EKF is a robot developed by Southall et al., (2002) to navigate in a field where plants are grown in a grid like pattern. Us-

ing the knowledge of the environment, a grid-based model of the local view of the robot formed the measurement model.

Despite these developments more research is required. It is well known that one of the main drawbacks of EKF is that it linearly approximates a non-linear system which results in the accumulation of errors as time progresses. Secondly, all the aforementioned methods use feature-based measurement models, in which first specific features are extracted from the images which are then used as measurements in the EKF. The Feature extraction process itself introduces uncertainties into the system. For instance, failure of the Hough Transform to extract the ‘correct’ lines will lead to navigation failure. This thesis attempts to address these two issues. We explore the use of a Particle Filter (PF) for autonomous navigation (Dellaert et al., 1999). A PF does not assume linearity of the system nor does it require the noise in the system to be Gaussian. Thus the PF becomes a natural choice for agricultural applications where the environment is complex and the measurements are non-Gaussian and may even have a multi-modal distribution. Although, there has not been much research in application of PF in agriculture, it is also not completely new. For instance, (Bergerman et al., 2012) uses a PF based method for autonomously navigating a vehicle in a orchard. They showed promising results with test runs of over 300 km. This thesis tries to further the understanding of the application of PF in agriculture as well as attempt to develop a probabilistic measurement models for camera and LIDAR.

1.2 Research Objectives

The aim of this thesis is to explore the use of probabilistic robotic methods in agricultural applications particularly for weed detection and autonomous navigation. This is achieved by addressing the following research objectives.

1. Explore the role of probabilistic texture features to improve the detection accuracy of Rumex in a grassland.
2. Model the uncertainty in the Rumex images and develop a Bayesian-based segmentation algorithm for the detection of Rumex in grassland.
3. Develop a PF based navigation method for autonomous navigation of a robot in a maize field.
4. Develop a probabilistic sensor model for a camera to be used as measurement model in PF.
5. Develop a probabilistic sensor model for a LIDAR to be used as measurement model in PF.

1.3 Outline of the Thesis

The research objectives are achieved in different chapters of the thesis. Chapter 2 takes a critical look at the existing computer vision methods for detection of Rumex in a grassland with the aim of improving upon them. Different texture features including Grey Level Co-occurrence Matrix are evaluated for their ability to detect the weed and the superiority of GLCM features (which are probabilistic in nature) over the others are demonstrated. Chapter 3 looks at the possibilities of applying Markov Random Field (MRF) theory to deal with uncertainties in an image by means of explicitly defining the spatial context. A Gaussian MRF (GMRF) is used to model a Rumex image. The GMRF model is used in a Bayesian context for a MAP-MRF formulation of the segmentation problem where the solution is given by finding the *Maximum a Posteriori* estimate, also known as the MAP estimate. The resulting real-time segmentation method is compared with the GLCM based segmentation method. The following two chapters focus on autonomous navigation of the robot in a maize field. In particular, Chapter 4 focuses on navigation of a robot equipped with a monocular camera. It describes a PF algorithm for robust row-following. It also details a novel probabilistic measurement model for the monochrome camera that is used in the PF. The novelty of the measurement model is that it does not rely on image features. Chapter 5 shifts attention to autonomous row-following using a LIDAR. It describes a probabilistic measurement model developed from first principles for a LIDAR. The LIDAR model replaces the camera model in the PF algorithm developed in the previous chapter and its performance is demonstrated in field experiments. Finally, Chapter 6 includes general discussion, conclusion and future perspectives.

Chapter 2

The role of textures to improve the detection accuracy of *Rumex obtusifolius* in robotic systems ¹

Abstract

Rumex obtusifolius is a common weed that is difficult to control in organic farming systems. Among the proposed non-chemical treatment methods, robotic systems to mechanically remove the weed have potential for its automatic detection and removal. This article considers a recently developed robot with a real-time vision system capable of detecting *R. obtusifolius* in a pasture with the objective of improving its detection accuracy. We show that the texture measure used by the current system is equivalent to local variance and has limited value for detecting *R. obtusifolius* in a complex background like pasture. To improve the system, two different sets of visual texture features corresponding to Gray level Co-occurrence Matrix (GLCM) and Laws' filter masks were investigated. Through feature selection, we determined that GLCM features of contrast, entropy and correlation were the best among the two sets of features and were 25% more accurate in estimating the taproot location than the current system. We incorporated these texture features in a new segmentation algorithm and demonstrated its robustness by testing it on a data set of 92 images with high complexity in terms of variation in illumination and weed size. The new segmentation algorithm had a detection accuracy of 90%, with an average error of 141 mm in the estimation of the location of the taproot of *R. obtusifolius*, compared with 308 mm with the former algorithm.

¹published as: Hiremath, S., van der Heijden, G., van Evert, F.K. & Stein, A. The role of textures to improve the detection accuracy of *Rumex obtusifolius* in robotic systems. In Weed Research, 2012, Vol. 52(5), pp. 430-440

2.1 Introduction

Rumex obtusifolius L. (broad-leaved dock or Rumex) is a common weed found in grassland. It is considered troublesome in dairy farming systems, because it proliferates at the cost of grass. In organic farming systems, where the use of chemicals is prohibited, various non-chemical methods are employed to control the weed (Zaller, 2006). Controlling *R. obtusifolius* with non-chemical methods is expensive and time-consuming; to prevent its regrowth, destruction of the taproot to a soil depth of 20 cm is necessary (Zaller, 2004). Frequent removal of the weed *per se* is insufficient to stop it from the spreading (Niggli et al., 1993). As a result, organic farms are often severely infested with *R. obtusifolius* (Bohm and Vershwele, 2004; Van Middelkoop et al., 2005).

Among the non-chemical control methods, robotic systems to mechanically remove the weed have potential for its automatic detection and removal. These robotic systems use computer vision for (semi) automatic detection of weed. A review of several such systems can be found in (Slaughter et al., 2008). Most of these systems focus on weeds in row crops and are designed to detect weeds against a soil background (Meyer et al., 1998; Aitkenhead et al., 2003; Tang et al., 2003; Guijarro et al., 2011) using colour contrast between soil and weed for detection. Systems that address the problem of detection of green broad-leaved weeds against a background of green grass mainly use texture-based vision methods.

Texture may be defined as the perceived characteristic pattern of irregularity of a surface. The irregularities of the surface scatter light differently at different viewing angles, which are manifest as spatially varying pixel intensities in the image. Texture analysis (a branch of computer vision) attempts to quantify these irregularities. Reviews on texture analysis can be found in (Haralick, 1979; Reed and Dubuf, 1993; Van Gool et al., 1985; Weszka and Dyer, 1976; Rao, 1990).

(Ahmad et al., 1997; Ahmad et al., 1998) developed a simple texture feature based on uniformity analysis to differentiate weed from grass in lawns using grey scale images. (Otsuka and Taniwaki, 1996; Otsuka and Taniwaki, 1999) used local variance (another measure of local uniformity) for weed detection. A more complex texture measure based on edge strength was developed by (Watchareeruetai et al., 2006); this measure exploits the fact that the density of edges of grass is higher than broad-leaved weeds. They extend their method further (Watchareeruetai et al., 2008) by including the texture measure in a classifier based on support vector machines. (Ahmad et al., 1999) used more sophisticated texture features than edge strength based on Gray level Co-occurrence Matrix (GLCM) to differentiate weed from grass. (Tang et al., 2003) developed texture features based on Gabor wavelets that may

be some of the most complex texture features developed for the detection of weed from grass.

Although the aforementioned methods are reported to work successfully for detecting weeds in soil background or in lawn grass, most of them are not applicable for the detection of *R. obtusifolius* in a pasture (Figure 3.1). Because of the high density of the grass in a pasture, uniformity analysis fails as the mean and range values of weed and grass tiles overlap considerably. (van Evert et al., 2009; Van Evert et al., 2011) developed a robot for automatic, real-time detection and removal of the weed using local spectral power as the texture feature. This method is based on local variance and requires frequent calibration, depending on the lighting condition. GLCM texture features are promising for correcting this problem, but which features are important and which are redundant are not yet clear. Further, the parameter settings required to compute the co-occurrence matrix so that the resulting texture features can discriminate between weed and grass efficiently are not known. Features based on edge strength are problematic for *R. obtusifolius*, because the leaves have strong veins and ribs that produce edges. Also, leaf damage (e.g. resulting from feeding by the chrysomelid beetle *Gastrophysa viridula* De Geer) causes additional edges. The computational cost of the method proposed by (Gebhardt and Kuehbauch, 2007) precludes (at present) real-time use in a robotic system. Using wavelets is unlikely to yield an improvement over using local power spectrum, as discussed in (van Evert et al., 2009). In summary, there are several potential texture features that can be used for the detection of *R. obtusifolius* in a dense pasture, but a thorough investigation is needed to determine the most suitable among them.

The work described here aims to improve the vision system by Van Evert et al. (2009), which, hereafter, is called the ‘current system’. The objective is to improve both the percentage of cases in which the presence or absence of *R. obtusifolius* is correctly determined and the accuracy of the estimated location of the taproot of the weed. The latter is important, because the current system controls *R. obtusifolius* by means of a cutter that is positioned in such a way so that the taproot is destroyed. The robot currently positions the cutter based on the centroid of the weed in the image. In many cases, this is close to the location of the taproot, but in 27% of cases, the distance was too large to guarantee taproot destruction (Van Evert et al., 2011). To determine the location of the taproot more precisely, a more accurate estimate of the outline of the weed leaves is necessary. To improve the detection accuracy, we identify a set of texture features that best discriminate *R. obtusifolius* from grass.

2.2 Materials and Methods

2.2.1 Images

We used 116 images collected by (van Evert et al., 2009). The images were taken in May and August 2006 under overcast conditions on several fields covering two different farms. One farm was located near Wageningen and the other was in Wilnis, Netherlands. A digital camera (Cybershot DSC-60; Sony, Tokyo, Japan) was used to capture the images and save them in JPEG format. The camera was set on automatic mode with the camera frame covering a ground area of $1.46 \times 1.09\text{m}^2$. The resolution of the captured images was 2304×1728 pixels. All images were pre-processed before use. The pre-processing consisted of two steps. First, the original colour image was converted to a grey scale image by averaging the red, green and blue channels as $I = (R + G + B)/3$. Subsequently, the grey scale image was downsampled to a resolution of 576×432 . This downscaling by a factor of four was chosen to have a consistent setting with respect to the image resolution of (van Evert et al., 2009). Of the 116 images, 24 images containing one (cluster of) *R. obtusifolius* were used in feature selection process. The other 92 images used for testing the new segmentation algorithm contained 43 grass (only) images and 49 images with one (cluster of) *R. obtusifolius*. For all images containing *R. obtusifolius*, the contour of the plant cluster was hand-segmented and used as reference for computing errors.

2.2.2 Texture features to detect *Rumex obtusifolius*

In the current vision system, to detect *R. obtusifolius* from grass, the original image is divided into a grid of tiles and the 2D discrete Fourier transform of each tile is computed. Subsequently, each tile is classified as *R. obtusifolius* (1) or grass (0), based on its spectral power (Pow) given by

$$\text{Pow} = \sum_{u=1}^{m-1} \sum_{v=1}^{n-1} F(u,v)^2 \quad (2.1)$$

where $F(u,v)$ is the Fourier transform of the image $f(x,y)$ of size $m \times n$.

This method exploits the fact that texture of grass is less smooth than texture of *R. obtusifolius*. Further, this difference is not confined to a specific frequency range but is present across the entire spectrum (van Evert et al., 2009), which allows the use of Pow as a texture feature.

The 0th frequency component $F(0,0)$ is excluded in the computation of Pow in (2.1), that is, indices start from (1,1). But

$$F(0,0) = \frac{1}{m \times n} \sum_{x=0}^m \sum_{y=0}^n f(x,y)$$

is the mean of the tile. As a result, Pow is equal to the statistical variance of the tile. This conclusion is based on Parseval's Identity, which states that the energy of a signal is the same in both frequency and spatial domain (Kammler, 2007). Effectively, the texture feature used by the current system is statistical variance of a tile. However, the computational complexity of computing the variance of an $m \times n$ tile is $O(N)$ ($N = m \times n$), whereas the complexity of calculating spectral power is $O(N \log N)$. Thus, computing the variance is $O(\log N)$ times faster than computing the spectral power. Here, $O(N)$ is a standard notation in mathematics used to describe the limiting behaviour (worst performance) of an algorithm for an input of size N . Henceforth, when variance is regarded as a texture feature, we refer to it as Var, and it is considered equivalent to Pow. Thus, all arguments that apply for Pow also apply for Var and vice versa, unless specified otherwise.

Var has limited value as texture measure, because it only captures the variation in the grey values but ignores the variation in their spatial arrangement, whereas texture is often defined as a function of spatial variation in grey values. That is, two grey scale images may have the same range (0, 255) but have different textures because of the different spatial arrangement of the grey values. A good texture measure is the one that captures the spatial dependencies between the pixels by specifying appropriate contextual constraints. GLCM and Laws' methods are two well-known approaches that provide texture measures that capture the spatial dependency between pixels. They incorporate contextual constraints by considering the distribution of pairs of pixels. In this article, we compare the two texture measures from these two methods with Var to determine which feature is the most suitable for discriminating between *R. obtusifolius* and grass.

The calculation of GLCM features starts with the construction of a data matrix that contains the relative probabilities of co-occurrence of a pair of grey values in an image (Haralick et al., 1973). It is characterised by three parameters: G , the number of grey levels; d , the distance between the pixel pair; and h , the orientation of the pixel pair. The value of G is a trade-off between computational time and accuracy, while values for d and h determine the coarseness and directionality of the texture content, respectively. The parameter settings are application dependent, although there are studies that have attempted to determine generic parameter values independent of the application. A meta-analysis of such studies can be found in (Clausi, 2001; Clausi, 2002). He carried out an exhaustive study to determine the most effective GLCM features, by investigating their behaviour as a function of the parameters G, d and h . He concludes that Entropy (Ent), Contrast (Con) and Correlation (Cor) form the optimal subset among the fourteen GLCM features, because they are uncorrelated and thus, each provides unique information about the texture. He also report that features corresponding to the

parameter setting $G = 32, d = 1$ and $h = 0^\circ$ and 90° gave relatively good results on their images. Based on the above findings, this study uses the Con, Ent and Cor features with the recommended parameter settings for the feature selection process.

Laws' energy features are extracted by convolving the original images with Laws' 2D convolution kernels (Laws, 1980b; Laws, 1980a) and passing the result through an 'Energy Filter'. The 'Energy Filter' is a statistic such as mean, variance, absolute value or any other mathematical function. The 2D convolution kernels are obtained by the outer product of separable 1D kernels. Laws suggested three different sizes of 1D kernels: 1×3 (L3), 1×5 (L5) and 1×7 (L7). A typical 2D kernel, for example, is the one formed by the outer product of L5 with itself and would be referred to as L5L5. Similarly, 25 different 2D kernels can be created from different combinations of the five 1D kernels. Thus, the main parameter involved for computing Laws' texture feature is the kernel size. Many studies (Laws, 1980a; Miller and Astley, 1992) have found the kernel size of five often works best and thus we use L5L5 kernels in our study.

2.2.3 Feature selection and validation

Feature selection is the process of selecting the best subset of relevant features for a given task. Feature selection methods can be classified into three categories: filter, wrapper and hybrid methods. Filter methods attempt to evaluate features based on measures like information content and correlation. These measures capture the general characteristics of the data without considering the objective of the application. On the other hand, wrapper methods evaluate the features indirectly, based on how well they contribute towards the application objective. The evaluation is indirect because the mining algorithm built using the features is evaluated and not the features themselves. The mining algorithm is usually designed based on the application at hand. For example, in a classification task, the features are used to build a classifier and an evaluation criterion like the predictive accuracy of the classifier determines the relevance or optimality of the features. Hybrid methods incorporate the strategies of both the filter and wrapper methods. Reviews on feature selection can be found in (Guyon and Elisseeff, 2003; Liu and Yu, 2005; Dash and Liu, 1997).

We used a wrapper method for selecting the optimal set of features, because the segmentation of *R. obtusifolius* in the image is solved by classifying each pixel in the image as belonging to either *R. obtusifolius* or grass. Thus, the feature performance is indicated by the predictive ability of the corresponding classifier. The classifier is evaluated using an Error Measure (details given below) designed to meet the application objective, that is, to detect the

taproot of the plant. The feature selection process consists of three main steps, namely construction of feature sets, training the classifier and evaluating the classifier. The validation step validates the whole feature selection process.

2.2.4 Construction of feature sets

This phase consists of computing the GLCM, Laws' and Var feature sets with a moving window of size $w \times w$ pixels. We assume a square window throughout the article and thus parametrize the window size by w . The GLCM features Con, Ent and Cor at window size w together form the feature vector $x^{g,w}$, where each individual feature (Con or Ent or Cor) is $x_i^{g,w}$. Similarly, $x^{l,w}$ and $x^{v,w}$ refer to Laws' and Var feature vectors, respectively. The number of components in $x^{g,w}$, $x^{l,w}$ and $x^{v,w}$ are three, fourteen and one, respectively. The moving window method is used to compute a feature vector for every pixel. $X^{g,w}$ is the GLCM feature set consisting of N feature vectors $x^{g,w}$ corresponding to N pixels. Similarly, $X^{l,w}$ and $X^{v,w}$ are Laws' and Var feature sets, respectively. These feature sets form the input to the classifier training phase.

2.2.5 Training the classifier

A classifier is a mapping of a feature vector into a class label. Training a classifier is learning the mapping rule from examples so that the classifier can correctly predict the class of a new unseen feature vector. This is conventionally known as supervised learning in the machine learning literature. Motivated by simplicity and low computational demand, we use Fisher linear discriminant classifier (FLD) in this study. The FLD maps a higher dimensional feature vector into lower dimensional space (one in our case) by taking the weighted linear combination of the components (individual features) in the vector. The linear combination is such that the Fisher criteria (ratio of the between-class means to the within-class means) is maximised. In other words, the learned mapping rule is such that the feature vectors corresponding to different classes are maximally separated in the lower dimensional space according to the Fisher criteria. A threshold on the projected data completes the classification process. The training targets (the desired class labels for the training images) corresponding to each feature vector are obtained by hand-segmented images (also referred to as reference images). The parameter settings used for constructing different feature sets are given in the Table 2.1.

Table 2.1: Parameter settings used for computing the GLCM, Laws' and Var features.

Features	Parameters
GLCM	$W = \{15, 21, 27, 33, 39, 45, 51, 57, 63\}$, $G = 32, d = 1, \theta = 0, 90$.
Laws'	$W = \{15, 21, 27, 33, 39, 45, 51, 57, 63\}$, $w_k = 5$.
Var	$W = \{15, 21, 27, 33, 39, 45, 51, 57, 63\}$.

2.2.6 Error measure

The result of the classification process is an image where each pixel is labelled as either 1 or 0. This binary image may contain several regions of potential *R. obtusifolius* of which only the largest is retained. The centre of this largest region is considered as the centre of the *R. obtusifolius*. The error in the algorithm-determined centre is measured relative to the centre of the hand-segmented reference image and is given by $E = \sqrt{(x - a)^2 + (y - b)^2}$, where (x, y) is the centroid of the hand-segmented object in the reference image and (a, b) is the algorithm-determined centroid. $E^{g,w}$, $E^{l,w}$ and $E^{v,w}$ are errors corresponding to GLCM, Laws' and Var feature sets, respectively, for a given window size w . The choice of the error measure was based on the objective, that is, to locate the taproot of the weed. Because the training phase consists only of images with one *R. obtusifolius*, the assumption that the largest region represents the weed is reasonable. This assumption, however, may not be valid for implementation *per se*, where one has no prior knowledge about the scene captured in the image.

2.2.7 Validation

Validation is a method to ensure the robustness and generality of the selected features. It involves repeated application of the feature selection process with different combinations of test and training images. The average error of all the iterations forms a robust estimate of the classifiers predictive performance. This study uses leave-one-out cross-validation. Each iteration of the leave-one-out process uses 23 of the 24 images for training and the remaining image for testing (evaluating) the classifier. The average error of the 24 iterations indicates the performance of the classifier and the corresponding features.

2.2.8 Segmentation Algorithm

Segmentation is solved by means of binary classification of each pixel as 1 or 0. The block diagram of the segmentation method, using the optimal

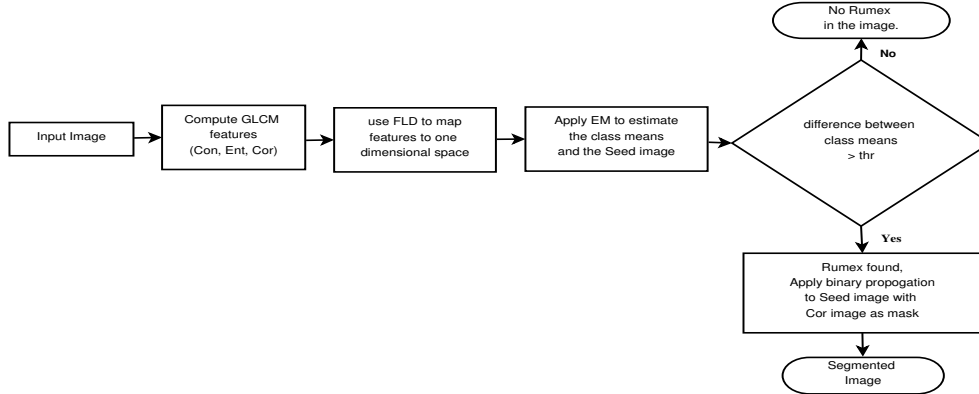


Figure 2.1: Flow chart showing the various components of the new segmentation method. The threshold value thr used in the experiment is 40 for intensity range between $[0, 255]$. This is implementation dependent and will differ according to the range of the intensity image.

features obtained from the feature selection process, is shown in 2.1 (This new segmentation method is also referred to as the new or proposed system.). The first step of the segmentation method involves calculating the selected features for each pixel in the image and mapping them to one-dimensional space using FLD. The projected data (also referred to as FLD data) are subsequently used for determining the presence or absence of *R. obtusifolius* in the image. We assume that FLD data corresponding to both *R. obtusifolius* and grass are Gaussian distributed with distinct means denoted by $\bar{\mu}_r = 1$ and $\bar{\mu}_g = 0$, respectively. The difference between the means is used to determine the presence or absence of *R. obtusifolius* in the image. The class densities (means and the variances) are estimated using expectation maximisation (EM) (Bishop, 2006). The EM step also estimates the most likely class membership of each pixel. These estimates are conservative, such that pixels with a high likelihood of being *R. obtusifolius* are classified as such, while the rest are classified as grass. This intermediate result is used as ‘seed’ image in the next stage of the segmentation process, known as binary propagation. In binary propagation, the regions around the seed points are expanded to include other ‘similar’ pixels as *R. obtusifolius* pixels. The similarity between the neighbouring pixels is measured by comparing their corresponding Cor feature values. Thus, the Cor image acts as a ‘mask’ and restricts the growth of the regions only into areas where the pixels have a correlation of 0.7 or higher. The correlation value of 0.7 is empirically determined. The algorithm stops when no more growth is possible yielding the desired segmentation of *R. obtusifolius* in the image.

Table 2.2: Variance of $E^{g,w}$ and $E^{l,w}$ for textures corresponding to different window size w

w	Variance of $E^{g,w}$	Variance of $E^{l,w}$
15	8931	8329
21	3598	8130
27	1341	8076
33	1261	4591
39	1126	3665
45	997	4346
51	1054	4321
57	1040	4203
63	1039	7143

2.3 Results

2.3.1 Performance of feature sets

The results of the classifiers corresponding to GLCM, Laws' and Var feature sets for an example image are shown in Fig 2.2. GLCM and Laws' features are better able than Var to characterise the difference between *R. obtusifolius* and grass across the entire range of w . Many Var features, for example $x^{v,w}$ where $w = (39, 45, \dots, 63)$, fail to detect the plant at all.

The box-plot of the errors $E^{g,w}$ and $E^{l,w}$ for different w obtained from the iterations of the leave-one-out analysis is shown in Fig 2.3. The variance of $E^{g,w}$ is smaller than the variance of $E^{l,w}$ across all window sizes as shown in Table 2.2. The mean errors $\bar{E}^{g,w}$ and $\bar{E}^{l,w}$ against different w are given in Fig. 2.4. We want to use the smallest window size that is not significantly different from the overall minimum. In Fig. 2.4 we see that $\bar{E}^{g,w}$ is least for $w = 27$. We also see that the mean error of GLCM features is considerably lower than the mean error of Laws' features for most values of w .

There are outliers with error values greater than 75 pixels. The images corresponding to these outliers are characterised by relatively small *R. obtusifolius*. The segmentation result of one such image using the three features sets is shown in Fig. 2.5. We can see that only GLCM features at $w = 27$ are able to detect *R. obtusifolius* in the image, while the others fail. Overall, $x^{g,w=27}$ is the best performing feature vector among all the candidate feature vectors with error $E^{g,w=27} = 60$ mm (expressed as a real-world measurement) in the estimation of the taproot of the weed in the image. In comparison, the average distance error of the current system is $E^{current} = 80$ mm (See Table 3, row 3, col 4 in of Van Evert et al., 2009). Because the feature selection process uses the same set of images as the leave-one-out, the performance of

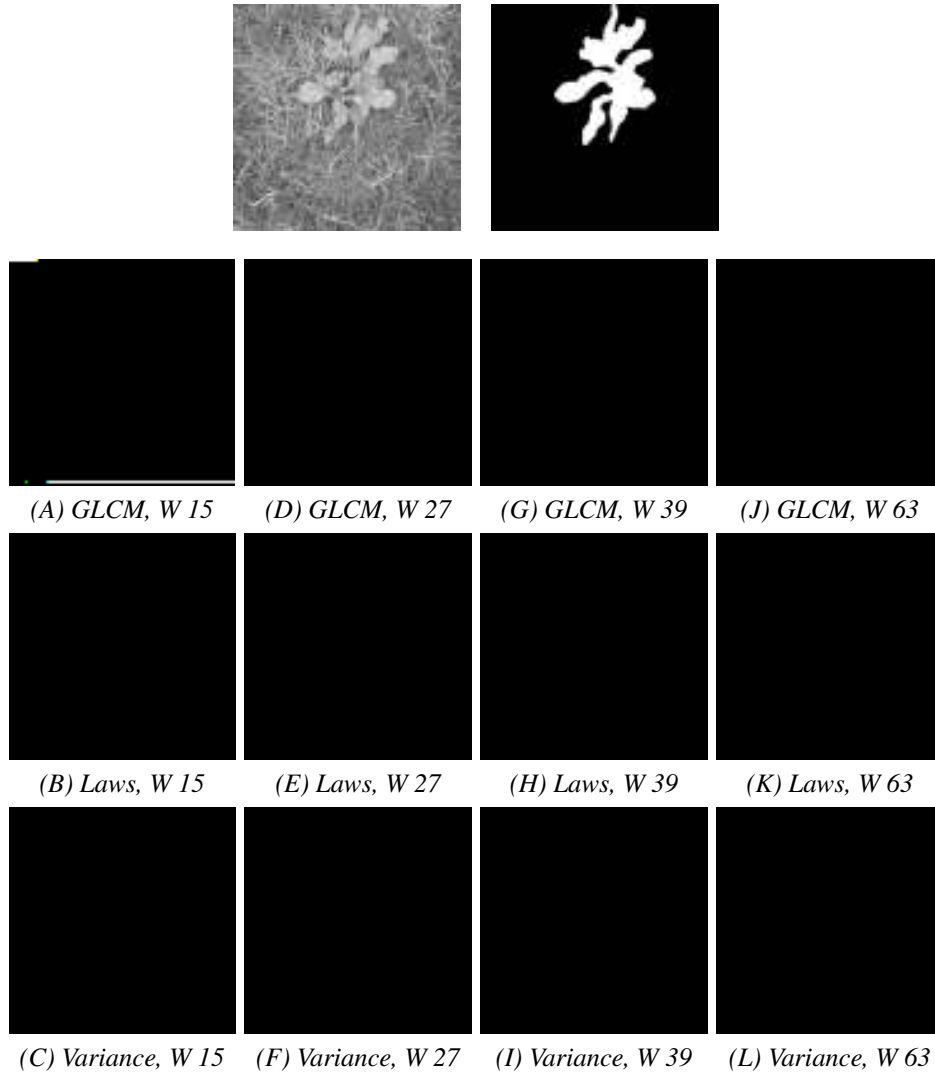


Figure 2.2: The segmentation of *R. obtusifolius* images using different feature sets. The top left image shows the grey image obtained after pre-processing. The top right image shows the expected result obtained by hand-segmenting the image. Pictures A-L show the segmentation results corresponding to the indicated feature sets.

the feature sets of the two systems is directly comparable. This indicates that GLCM feature obtained at window size $w = 27$ is a better texture feature for discriminating between *R. obtusifolius* from grass in a pasture.

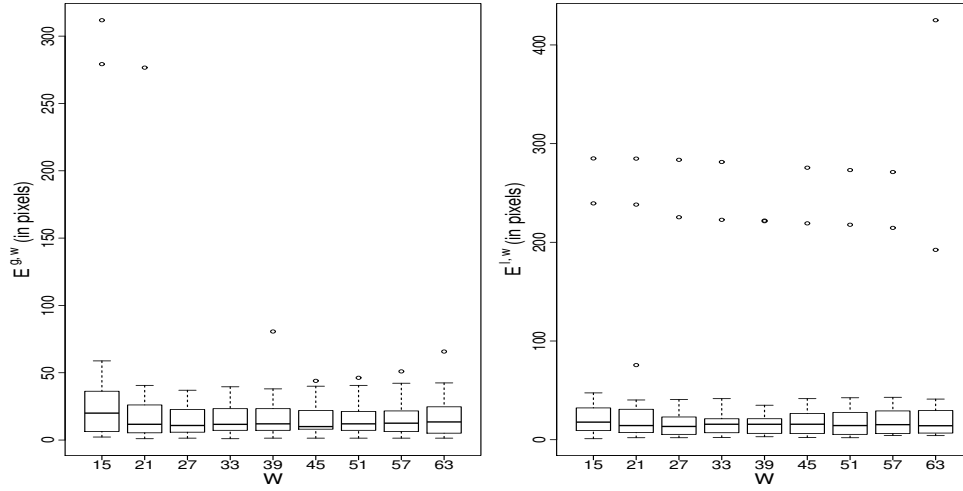


Figure 2.3: Box-plots of error measures, $E^{g,w}$ (left) and $E^{l,w}$ (right), defined as the distance, in pixels, between the centre of the image blob and the centre of the hand-segmented plant obtained from the iterations of leave-one-out analysis at different window size. The plots provide an indication of robustness and consistency. Because the plot corresponding to Gray level Co-occurrence Matrix (GLCM) features has less outliers (extreme values) compared to that of Laws' features, GLCM are more robust than Laws' features.

2.3.2 Performance of the segmentation algorithm

Based on the results of the previous section, $x^{g,w=27}$ (i.e. GLCM texture features obtained at window size $w = 27$) was used as the feature vector in the segmentation algorithm. The performance of the algorithm was tested on a set of 92 images different from the 24 training images. Of the 92 images, 49 images contained a single cluster of *R. obtusifolius*, and the remaining 43 images contained only grass. Table 2.3 shows the confusion matrix tabulating the performance of the segmentation algorithm in its ability to correctly detect the presence or absence of *R. obtusifolius* against the actual values. The columns in the table indicate algorithm performance, and the rows indicate the actual data. Several performance measures can be derived from the confusion matrix, namely detection accuracy (DA), sensitivity (SE) and specificity (SP). The DA measures the proportion of the total number of correctly identified weed and grass images and is given by $DA = (TN + TP) / (TP + FP + TN + FN)$. SE measures the fraction of correctly identified *R. obtusifolius* among the 49 images and is given by $SE = TP / (TP + FN)$. SP measures the fraction of correctly identified grass among the 43 images containing only grass and is given by $SP = TN / (TN + FP)$. Here, TP, TN, FP, FN stand for true positive, true negative, false positive and false negative, respectively. The DA, SE and

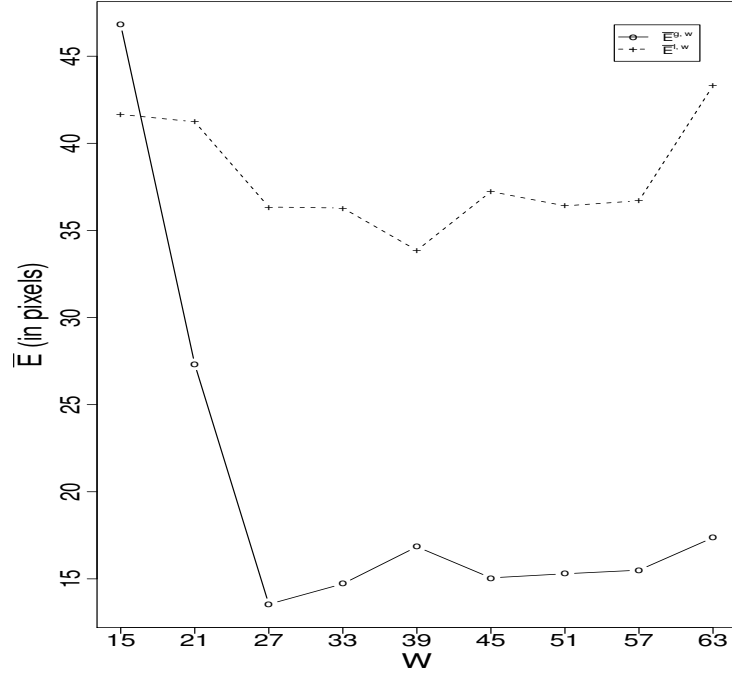


Figure 2.4: Figure showing the mean errors, $\bar{E}^{g,w}$ and $\bar{E}^{l,w}$ of the Gray level Co-occurrence Matrix (GLCM) and Laws' feature sets, respectively, computed at different window size, indicating that the GLCM features computed at a window size of 27 give the most accurate results.

SP of the new segmentation algorithm are 90%, 84%, and 98% respectively. The average error $E^{g,w=27}$ of the estimated taproot of the correctly identified images (true positives) is 141 mm. This means that, on average, the detection method will miss the centre of a *R. obtusifolius* by 141 mm. In comparison, the average error of the current system when tested with the new dataset was 308 mm.

2.4 Discussion

The results show that the segmentation of *R. obtusifolius* using either GLCM or Laws' texture features was better than Var feature. From the box-plot of the error values in Fig. 2.3, we noticed that GLCM features were more robust in capturing the differences between the texture of *R. obtusifolius* and grass. The superiority of GLCM features compared with the other features stands out in experiments with images containing small plants. These experiments illustrate the need to identify the appropriate size or range of the window, w , over which the features are calculated to facilitate the characterisation of texture differences between the two classes. This value, however, is application

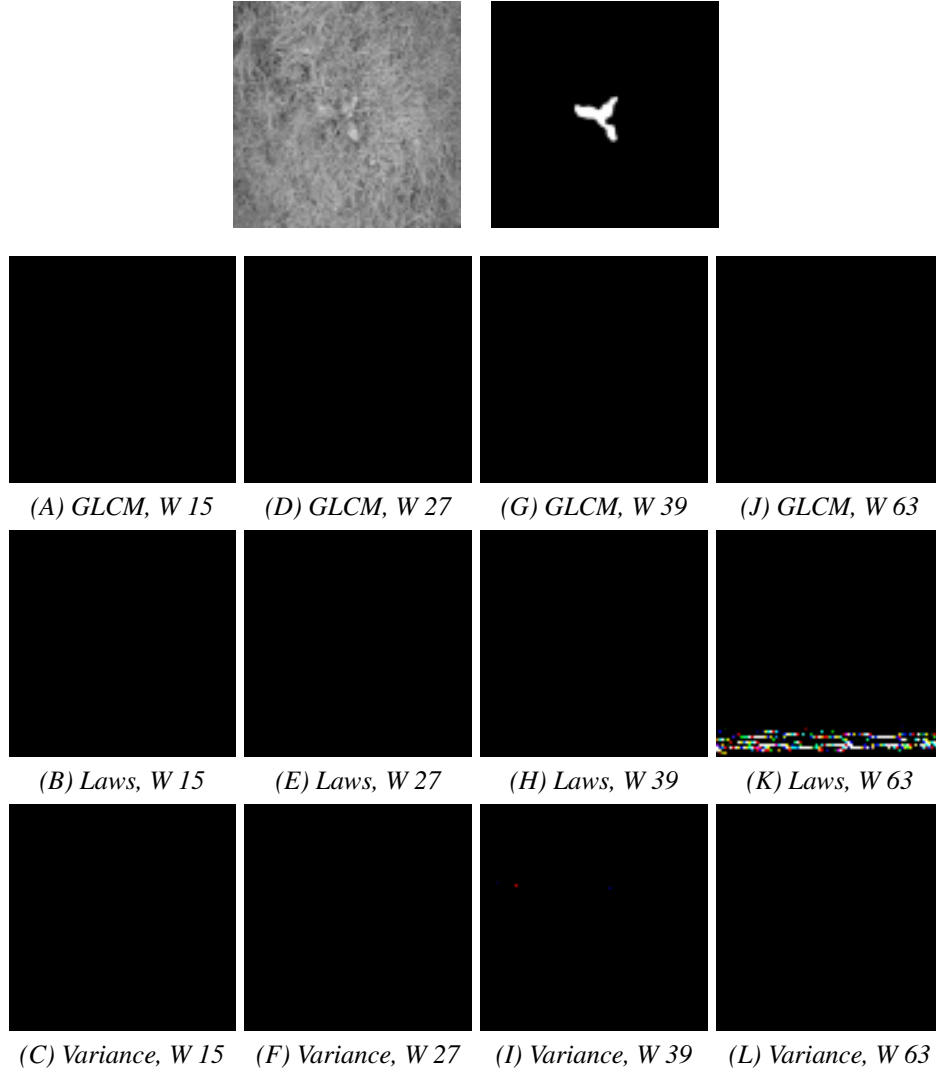


Figure 2.5: Segmentation results of an image containing small weed: The top left image shows the grey image obtained after pre-processing. The top right image shows the expected result obtained by hand-segmenting the image. Pictures A-L show the segmentation results corresponding to the indicated feature sets.

dependent, that is, it depends upon the type and resolution of images, the type of texture and features and the segmentation methods used in the application. Our results concerning the better performance of GLCM features over Laws' features are in line with those reported in the literature (du Buf et al., 1990).

The 49 images on which the new system was tested contained only one cluster of *R. obtusifolius*, suggesting that it can handle only one cluster of plant in the image. In principle, however, the new system will be able to

Table 2.3: Confusion matrix of new segmentation algorithm. The cell labels TP, FP, FN and TN indicate true positive, false positive, false negative and true negative respectively

	Algorithm	
	Rumex present	Rumex absent
Actual		
Rumex present	41 (TP)	8 (FN)
Rumex absent	1 (FP)	42 (TN)

detect multiple clusters of *R. obtusifolius*. Not being able to detect every plant in the image, however, is not a major concern, as it is not necessary that all *R. obtusifolius* in a pasture be detected in a single robot run. Repeated runs may be employed for a full control of plant occurrence.

We found that the new algorithm takes about 17.9 s for processing a single image on a laptop computer with a 2.66 GHz Intel i5 processor. This is prohibitive for real-time application, but it should be noted that the high computation cost is owing to the overlapping window (moving window) operation that yields features for every pixel; feature computation took 17.1 s, while the rest of the segmentation algorithm took 0.8 s. Feature computation at such a fine-grained level is not necessary in most cases. One can gain considerable improvement in speed by using non-overlapping window operation where features are computed for an image block rather than for every pixel. For example, computing features at every eighth pixel reduce the computation time by a factor 64 and this would allow processing of three images per second. This, however, comes at the cost of detection accuracy, and the user must decide on the trade-off between accuracy and speed of the system. Importantly, computing local features for every p th pixel on an image I of size $m \times n$ is not the same as computing the features on an image I_d that is downsampled by a factor of p , because the two images, I and I_d , have different resolution and thus different textures. Texture features from I capture more detailed spatial interactions compared with features computed from I_d . Also, for a given window size w , the neighbourhood extent is larger in I_d because the downscaling process on I brings the pixels ‘closer’, with the result that I_d has less ‘texture content’ than I .

The confusion matrix shows that the algorithm has a high specificity of 98%. This means that it does not detect many false positives. From the application point of view, this is desirable because false detections lead to unnecessary drilling causing considerable damage to the pasture while one can afford to mistake a few *R. obtusifolius* plants for grass that can be controlled during subsequent runs.

The new segmentation algorithm is modular and offers a great deal of

flexibility, making available several alternatives with negligible loss of detection accuracy. Some possible alternative configurations are as follows. The first is to omit the binary propagation step and use the seed image as a final segmented image. The second possibility is to replace the EM step. The pixels corresponding to *R. obtusifolius* and grass are maximally separated in the FLD data, and EM is a robust and elegant way to estimate the individual distributions within the mixture. At its heart, the EM algorithm analyses the shape (non-normality) of the mixture distribution to estimate the component distributions. Other statistical measures, like the skewness, however, can be used to achieve the same. The aforementioned alternative configurations of the segmentation algorithm show similar performance, a detection accuracy of 88%. Although this is 3% less than the initial configuration of the new system, it still matches the best performance of the current system. We speculate, however, that the alternative configurations will be limited in their capacity in terms of handling intensity variation and as well as those images with relatively small plants.

At the final step of the segmentation method, the location of the taproot of a *R. obtusifolius* plant is estimated by calculating the centroid of the segmented object. In reality, the taproot is not always located at this position. Moreover, in case of a cluster of overlapping plants, several roots exist and the current procedure will fail to identify all these individual roots. Handling such complexities may require incorporating an additional module in the segmentation algorithm for the detection and tracking of the mid-ribs of the leaves to identify the plant centres more reliably.

A major problem with the vision system of (van Evert et al., 2009) is its sensitivity to variation in ambient light. The vision system needs to be calibrated frequently based on the prevalent lighting conditions during the time of its operation. This problem is mitigated to some extent in the new system with the help of GLCM features, owing to their ability to handle images with a greater degree of illumination variation than the Pow feature, although the problem still persists. Illumination invariant feature extraction methods are extensively studied in the area of face recognition (Gross and Brajovic, 2003) and may be useful to tackle this problem. Also, the range of image resolution at which a linear texture measure can characterise the differences between the textures is small compared with non-linear texture measures like the GLCM features. We suppose that GLCM features can be used to differentiate between different types of broad-leaved plants.

2.5 Conclusions

This study explored two sets of texture features, namely GLCM and Laws' for the detection of *R. obtusifolius* in a pasture. Through feature selection

process, it was determined that GLCM features of contract, entropy and correlation computed at window size $w = 27$ were the best performing among the two. The experiments showed that they are 20 mm more accurate in estimating the location of the taproot of *R. obtusifolius* compared with the Var feature used in the current system. These features in the new segmentation algorithm showed a high degree of robustness to lighting conditions and weed size, as demonstrated by the tests on the data set of 92 images with large variation in illumination size of *R. obtusifolius*.

Chapter 3

Segmentation of *Rumex obtusifolius* Using Gaussian Markov Random Fields ¹

Abstract

Rumex obtusifolius is a common weed that is difficult to control. The most common way to control weeds—using herbicides—is being reconsidered due to its adverse environmental impact. Robotic systems are regarded as a viable non-chemical alternative for treating *R. obtusifolius* and also other weeds. Among the existing systems for weed control, only a few are applicable in real-time and operate in a controlled environment. In this study we develop a new algorithm for segmentation of *R. obtusifolius* using texture features based on Markov Random Fields (MRF) that works in real-time under natural lighting conditions. We show its performance by comparing it with an existing real-time algorithm that uses spectral power as texture feature. We show that the new algorithm is not only accurate with detection rate of 97.8% and average error of 56mm in estimating the location of the tap-root of the plant, but is also fast taking 0.18 seconds to process an image of size 576×432 pixels making it feasible for real-time applications.

3.1 Introduction

Rumex obtusifolius is one of the most common weeds in the Netherlands. Its rapid growth is a major concern because it competes with grass for natural resources like water, nutrients, and light reducing crop yield. A widely used

¹published as: Hiremath, S., Tolpekin, V., Heijden, G. & Stein, A. Segmentation of *Rumex obtusifolius* using Gaussian Markov random fields. In Machine Vision and Applications, 2013, Vol. 24(4), pp. 845-854

method of controlling weeds is by application of herbicides, however, the use of herbicides is being questioned due to their adverse affects on the environment. For instance, herbicides are found to cause ground water pollution (Kempenaar et al., 2007) and force selective bias towards herbicide-resistant weeds. There are several non-chemical methods for controlling weeds, like crop rotation, manual removal, thermal and biological control. But these methods are labour intensive, non-scalable, and expensive both in time and costs (Bond and Grundy, 2001; van Evert et al., 2009). As a result, robotic systems are being considered for mechanical treatment of weed (Slaughter et al., 2008; van Evert et al., 2009). These systems use vision based methods for detection of the weed.

Our focus is on detection of *R. obtusifolius* in grasslands. Most of available vision based robotic systems for detecting weeds are not suitable for real-time detection of *R. obtusifolius* in grasslands for several reasons. For instance, many methods focus on detection in soil background using colour contrast between soil and weed (Aitkenhead et al., 2003; Guijarro et al., 2011). These methods cannot be applied for detecting *R. obtusifolius* in grassland because they are characterized with similar spectral reflectance in visible range. Another method (Ahmad et al., 1997), uses uniformity-based texture measure to detect weeds in lawn field. This technique is not applicable in grasslands, because unlike in a lawn, the grass does not have a consistent texture. In (Otsuka and Taniwaki, 1999) local variance was used as the texture measure to detect broad-leaf weeds in grassland. Local variance is another measure of uniformity of texture and our preliminary analysis showed that it is unreliable due to the inconsistency of grass texture. A real-time vision system which uses spectral power as a texture measure was developed in (van Evert et al., 2009) for the detection of *R. obtusifolius* in grassland. In this method, the image is partitioned in to square tiles. Each tile is classified as weed or grass based on its spectral power. They report satisfactory performance, however, they also mention that it is sensitive to illumination condition. Local spectral power is the same as local variance (Kammler, 2007) and thus suffers from the same drawbacks as mentioned above. The authors of (Watchareeruetai et al., 2008) exploited the difference in density of edges between grass and leaf regions and developed a texture measure based on edge strength. The veins and ribs of *R. obtusifolius* form strong edges due to which differentiating it from grass based on edge strength alone is difficult. In the study of detecting weed in lawn fields (Ahmad et al., 1999), the authors use texture features based on Grey-level co-occurrence matrix (GLCM). GLCM features are one of the most widely used texture features (Haralick et al., 1973) but are computationally expensive and thus cannot be applied in real-time.

The goal of this study is to develop a robust, vision-based, real-time segmentation method for detection of *R. obtusifolius* in grassland under natural



Figure 3.1: A typical image of *R. obtusifolius*

lighting conditions. Segmentation of *R. obtusifolius* in grassland entails uncertainty in terms of shape, size, and illumination. Figure 3.1 shows a typical image of *R. obtusifolius* in grassland taken by the robot. As we can see, not only do the spatial (spectral and textural) properties of grass and Rumex pixels vary but so do the spatial properties of grass pixels in different parts of the image. Further, this pattern varies from image to image. A robust segmentation algorithm should account for these uncertainties. Markov Random Field (MRF) theory helps us to deal with uncertainties in an image by means of explicitly defining the spatial context and describing the spatial interactions between pixels as a probability distribution (Li, 2001; Hammersley and Clifford, 1971). The suitability of MRFs for modelling natural images have been demonstrated in several studies (Efros and Leung, 1999; Zalesny and Gool, 2001).

We assume that the image has only two classes, Rumex and grass and model it as a Gaussian MRF (GMRF) (Chellappa and Chatterjee, 1985). GMRF is one of the simplest MRF for encoding spatial interactions between pixels and its parameters can easily be estimated using Least Squares (Chellappa, 1985). We regard the model parameters as texture features and formulate the segmentation of Rumex from grass as a *Maximum a Posteriori* (MAP) inference problem within the MAP-MRF framework (Li, 2001). This allows us to seamlessly integrate the GMRF model of the image with a prior model (as detailed in the next section). Finally, the optimisation problem is solved via Graph Cuts (Boykov et al., 2001). The choice of Graph cuts for the optimization was based on real-time constraints. Several studies have shown they are robust and capable of operating in real-time (Boykov and Kolmogorov, 2004; Szeliski et al., 2008); they are among the most widely used optimization algorithms for real-time object segmentation (Vogiatzis et al.,

2005; Rother et al., 2004; Juan and Boykov, 2006; Boykov and Kolmogorov, 2003).

We demonstrate the real-time applicability of the proposed algorithm—referred to as A1—by comparing it with the one developed by (van Evert et al., 2009)—referred to as A2.

The rest of the paper is organized as follows. Section 3.2 gives the basic definitions and notations of MRF theory and Graph Cuts. The image model and the segmentation method is described in Section 3.3 and Section 3.4, respectively. Section 3.5 contains experiments and results followed with discussions and conclusion.

3.2 Markov Random Fields

In this section we introduce the standard definitions and terminology of MRF theory. Although this material has been reviewed numerous times, we include it here to indicate the link between MAP-MRF framework and Graph Cut as well as to make clear how they are implemented for this application.

Markov Random Field (MRF) theory is a probability based theory that provides mathematically rigorous modelling tools to model interactions between spatially varying entities. In 1984 Geman & Geman popularized the use of MRFs in image analysis (Geman and Geman, 1984). Since then, MRFs have found application in almost all areas of computer vision and image analysis (Clausi, 2001; Zhao et al., 2007; Boykov et al., 1998; Blake et al., 2004). A review of MRF based image analysis can be found in (Li, 2001).

MRF theory considers texture as a 2D stochastic process consisting of random variables in a plane, referred to as random field. The random field is characterized in terms of a conditional distribution of the constituting random variables. Consider a 2D rectangular lattice of pixels

$$S = \{i = (1, 2, \dots, m)\}$$

where m is the total number of pixels. Let x_i be a random variable associated to pixel i . It can take values from the discrete labelled set $\mathcal{L} = (d_1, d_2, \dots, d_l)$. A random variable having a specific value is said to be an instantiated random variable. A set of all instantiated random variables corresponding to all the pixels is denoted by x . That is, $x = \{x_i = d \in \mathcal{L}, i = (1, 2, \dots, m)\}$. Note that we use x_i to denote both the random variable at pixel i and its value (label). The distinction between them is made explicit if the context demands.

Definition 1: A neighbourhood is a subset of pixels η_i corresponding to pixel i given by

$$\eta_i = \{j \in S \setminus \{i\} | \text{dist}(i, j)^2 \leq r\}$$

	β		β	β	β		β_2		β_4	β_3	β_2
β	i	β	β	i	β	β_1	i	β_1	β_1	i	β_1
	β		β	β	β		β_2		β_2	β_3	β_4
$\eta_i^{(1)}$			$\eta_i^{(2)}$			$\eta_i^{(3)}$			$\eta_i^{(4)}$		

Figure 3.2: Example neighbourhood systems of pixel i . $\eta_i^{(1)}$, $\eta_i^{(2)}$, $\eta_i^{(3)}$ and $\eta_i^{(4)}$ are first-order isotropic, second-order isotropic, first-order anisotropic and second-order anisotropic neighbourhood systems, respectively. In the Figure, the pixels that are the members of the set η_i are marked with the associated model parameter β_j .

where, r is an integer that specifies the radius of the neighbourhood, $dist(i, j)$ is the Euclidean distance between pixels i and j .

The radius r and the isotropy of the neighbourhood determine the number of pixels it contains and the parameters associated with them, respectively. Figure 3.2 illustrates different neighbourhood systems. In the figure, the members of the neighbourhood set η_i are marked by the corresponding parameter β . We use the boldface notation β to represent a vector of all model parameters. $\eta_i^{(1)}$, $\eta_i^{(2)}$, $\eta_i^{(3)}$ and $\eta_i^{(4)}$ are first-order isotropic, second-order isotropic, first-order anisotropic and second-order anisotropic neighbourhood systems, respectively. Note that the pixels along the boundary of the image do not have the same number of neighbours as the interior pixels. To specify an MRF, we have to specify the conditional probability $P(x_i|x_j, j \in \eta_i)$. This is done by specifying the Gibbs distribution (Hammersley and Clifford, 1971), given by

$$P(x) = \frac{1}{Z} e^{\{-U(x)\}} \quad (3.1)$$

where $U(x)$ is called the potential function and Z is the normalizing constant known as the partition function. To define $U(x)$, we first need to define cliques associated to a neighbourhood system and the corresponding clique potentials.

Definition 2: A clique $c \in S$, is a set of pixels such that every distinct pair of pixels in the set are neighbours. Figure 3.3 shows single-site and pair-site cliques of a second order neighbourhood system. $C = \{c_1 \cup c_2 \cup c_3 \cup \dots\}$ is the set of all cliques where c_n denotes a clique with n pixels. A clique potential, $V_c(x)$, is a potential function associated with a clique type c , whose value is dependent on the structure of c . For example, the clique potential $V_1(x_i)$ of a single-site clique $c_1 = \{i\}$, is a function of the label at pixel i . Similarly, for a pair-site clique $c_2 = \{i, j\}$, the potential function $V_2(x_i, x_j)$, is a function of the labels of the pixel i and j .

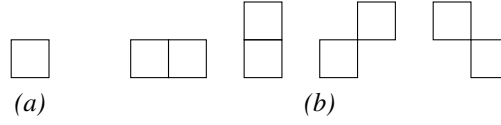


Figure 3.3: Two clique types: Figures 3.3a and 3.3b are single-site clique and pair-site cliques, respectively

Definition 3: The potential function $U(x)$, is defined as

$$U(x) = \sum_{c \in C} V_c(x) \quad (3.2)$$

where summation is done over all possible cliques C . Thus, the Gibbs distribution is specified once the clique potentials are determined.

3.2.1 MAP-MRF Framework

In this section we describe how we can use MRFs in a Bayesian context for a MAP-MRF formulation of the segmentation problem where the solution is given by finding the *Maximum a Posteriori* estimate, also known as the MAP estimate.

The image y is assumed to be a noisy realization of an underlying MRF x . The goal is to recover x using the observed y . That is, we want to find the posterior distribution, $P(x|y)$. The maximum of the posterior distribution, the MAP estimate is obtained by minimizing the negative log likelihood (Duda et al., 2001)

$$x^* = \arg \min_x U(x|y)$$

where the posterior potential $U(x|y)$ is given by

$$\begin{aligned} U(x|y) &= U(y|x) + \lambda U(x) \\ &= \sum_{i=1}^m V_1(y_i|x_i) + \lambda \sum_{i=1}^m \sum_{j \in c_2} V_2(x_i, x_j) \end{aligned} \quad (3.3)$$

Here, λ is called the region parameter and determines the importance of the prior potential $U(x)$ over the likelihood potential $U(y|x)$. $V_1(y_i|x_i)$ is the single-site clique potential and $V_2(x_i, x_j)$ is the pair-site clique potential; higher order clique potentials are ignored.

3.2.2 Minimizing via Graph Cuts

In graph cut literature (and other graph based algorithms in general), the potential function $U(x|y)$ (also referred to as energy function) specifies a graph

$\mathcal{G} = \langle \mathfrak{N}, \xi \rangle$, where $\mathfrak{N}$ consists of a set of nodes connected by edges ξ . There are two special nodes namely source and sink. Two regular nodes are connected by *n-links* and *t-links* connects each regular node to the source and sink. All the edges have an associated cost. The cost of *n-links* is analogous to $V_2(x_i, x_j)$ and the cost of *t-links* to $V_1(x_i)$. Graph cut partitions the graph into two disjoint subsets by removing edges such that the source and sink nodes are not in the same set and the total cost of the partition is minimal. This solution minimizes the energy (3.3).

3.3 Image Model

In this section we give a detailed description of the image model that we use in the study. We assume that the image contains two classes: Rumex and grass, each with a different texture. It is modelled as a GMRF given by

$$P(y_i | y_{\hat{\eta}_i}, x_i) = \frac{1}{\sqrt{2\pi v}} \exp \left\{ -\frac{(y_i - \sum_{j \in \hat{\eta}_i} \beta_j (y_i + y_j))^2}{2v} \right\} \quad (3.4)$$

where the subset $\hat{\eta}_i \in \eta_i$ is the asymmetric neighbourhood of i , such that if $j \in \hat{\eta}_i$ then $-j \notin \hat{\eta}_i$ and $-j \in \eta_i$, β_j is the model parameter associated with the members of η_i , the set $\{j, -j\}$ is the symmetric neighbour pair and e_i is the Gaussian noise with correlation structure

$$E[e_i e_j] = \begin{cases} v & \text{if } i = j \\ -\beta_j v & \text{if } j \in \hat{\eta}_i \\ 0 & \text{otherwise} \end{cases} \quad (3.5)$$

The model parameters β_j and v depend on the label x_i . They form the features that describe the textures within a region. The model parameters are estimated by means of least squares as

$$\beta^* = \left[\sum_{i \in S} Q_i Q_i^T \right]^{-1} \left[\sum_{i \in S} Q_i y_i \right] \quad (3.6)$$

and

$$v^* = \frac{1}{m^2} \sum_{i=1}^m [y_i - Q_i^T \beta^*]^2 \quad (3.7)$$

where Q_i is a column vector given by

$$Q_i = [y_j + y_{-j}, j \in \hat{\eta}_i]$$

3.4 Supervised Segmentation

A supervised approach consisting of two phases is employed for segmentation. The first phase—called the training phase—consists of feature selection in which we learn optimal GMRF model parameters (β^*, v^*) along with the region parameter λ using a set of training images. The second phase consists of testing the selected features by using them in the segmentation algorithm. The accuracy of the segmentation algorithm is measured by evaluating the segmentation error: the error between the algorithmically determined centre of the plant from the reference centre obtained from the corresponding hand-segmented image. It is given by

$$D = \sqrt{(a_1 - a_2)^2 + (b_1 - b_2)^2} \quad (3.8)$$

where (a_1, b_1) is the centre of the plant determined by the segmentation algorithm and (a_2, b_2) is its centre in the corresponding reference image. The motivation for using the above error measure is based on the application objective which is to identify the centre of *R. obtusifolius* so that it can be up-rooted by the robot. Note that the error measure is only applicable if Rumex is detected in the image. Figure 3.4 shows the flow chart of the segmentation algorithm.

3.4.1 Images

The images used in the experiments were taken using a commercial camera (Cybershot DSC-60; Sony, Tokyo, Japan), in automatic white balance mode, at a resolution of 2304×1728 pixels with the corresponding ground resolution of $1.22 \times 1.22 \text{ mm}^2$ per pixel. For experiments, however, the images were downsampled to 576×432 to be consistent with the settings used in (van Evert et al., 2009). Otherwise the comparative results will be incorrect as we will be comparing algorithms operating on different textures. The images were divided into two sets: a training set and a test set. The training set consisted of 16 images with no *R. obtusifolius*—referred to as grass-image—and 24 images consisting of a single cluster of *R. obtusifolius*—referred to as rumex-image—while the test set consisted of 43 grass-images and 49 rumex-images. Every rumex-image consists of a corresponding hand-segmented reference-image, where the contour of the plant cluster is determined manually indicating the desired segmentation.

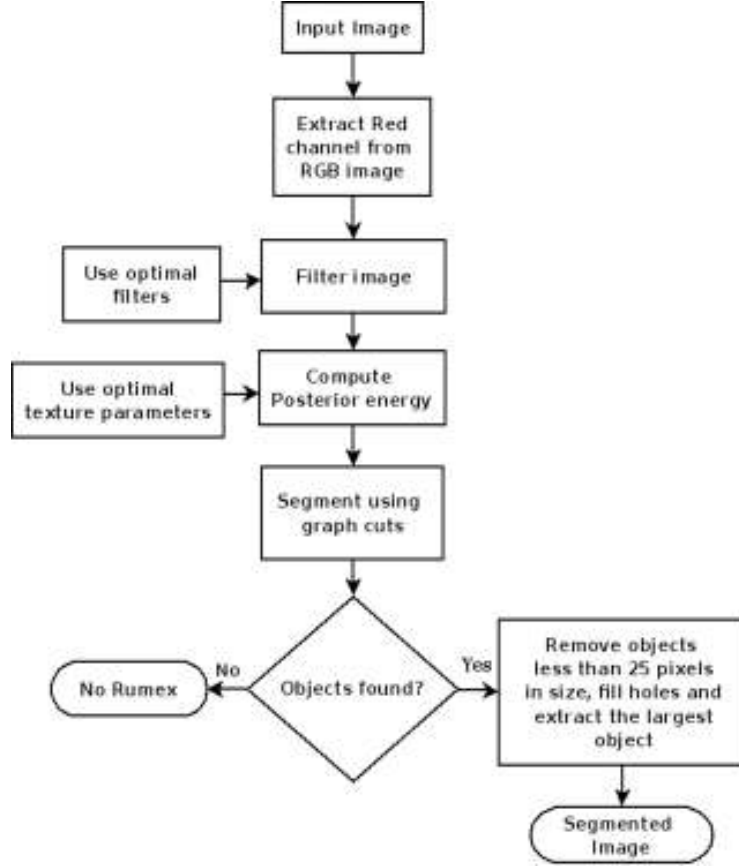


Figure 3.4: Flow chart of the segmentation algorithm

3.4.2 Feature Selection

Determining Texture Features, β^* and v^*

The features that accurately discern the label of a pixel as Rumex or grass would be called the optimal texture features. Since the texture features are the GMRF model parameters β and v , several model properties like the neighbourhood system and the scale of the texture (explained below) influence the quality of the texture features. We compare features corresponding to four different neighbourhood system at ten different scales listed in Figure 3.2. The scale of the texture in our case is the degree of smoothness of the texture which is determined by using various Gaussian filters. Through an empirical study we narrowed down the range of Gaussian filters to the ten shown in Table 3.1. A feature set corresponding to a specific neighbourhood system $\eta_i^{(r)}$ and a scale $g^{(i)}$ is computed as follows: Homogeneous texture patches of a given class (see Figure 3.5) are smoothed using a Gaussian filter, $g^{(i)}$. Subsequently, the smoothed texture is modelled as GMRF with the neighbourhood

Table 3.1: Different Gaussian filter parameters used for changing the texture scale. sz is the filter size in pixels and σ is the standard deviation of the filter

Filter id	sz	σ
$g^{(1)}$	3	0.3
$g^{(2)}$	3	0.7
$g^{(3)}$	3	1.1
$g^{(4)}$	3	1.5
$g^{(5)}$	3	2
$g^{(6)}$	4	0.3
$g^{(7)}$	4	0.7
$g^{(8)}$	4	1.1
$g^{(9)}$	4	1.5
$g^{(10)}$	4	2



Figure 3.5: Homogeneous patches of Rumex (left) and grass (right)

system $\eta_i^{(t)}$ and the estimates β^* and v^* are obtained according to (3.6) and (3.7), respectively.

The feature selection method employed in the paper is categorized as filter method (Kohavi and John, 1997) in machine learning literature. We use Fisher's criterion (Duda et al., 2001) to evaluate the quality of the texture features for their ability to differentiate between the two classes. Figure 3.6a is a plot of the Fisher score of each feature set. Each curve in the plot corresponds to a particular neighbourhood system. The x -axis indicates the scale of the texture. As we can see from the graph, features corresponding to neighbourhood system $\eta_i^{(3)}$ consistently outperform other feature sets at all scales. The best features are obtained with neighbourhood system $\eta_i^{(3)}$ and scale $g^{(7)}$. The selection process was repeated for various colour channels: red, green, blue and grey and we found that the red channel yielded the best parameters.

A filter based method for selecting features provides generic features in-

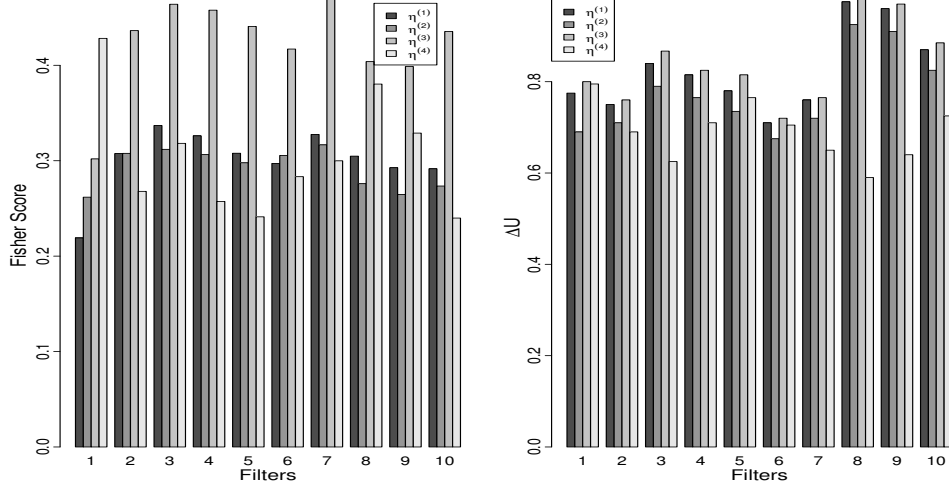


Figure 3.6: Figure 3.6a and 3.6b shows the Fisher Score and ΔU of various feature sets, respectively. The x-axis of the graph indicates the scale of the texture at which the features were computed and each curve indicates the neighbourhood system associated with the feature set. Note that the graph is not continuous; the lines connecting the Fisher score of different features is only to assist interpretation.

dependent of the segmentation algorithm (Guyon and Elisseeff, 2003). To ensure that the features are also specific to the application, they were evaluated based on an application-specific criterion: the difference in the log-likelihood of the correct and the incorrect class of a given texture patch given by

$$\Delta U(y|x) = U(y|x_r; \beta_r^*, \nu_r^*) - U(y|x_g; \beta_g^*, \nu_g^*)$$

where, y is the texture patch of a particular class. The feature set that best describes the texture corresponds to the maximum ΔU . Figure 3.6b validates the Fisher Score graph by showing that the neighbourhood system, $\eta^{(3)}$, provides the most suitable set of texture features across various scales.

Determining Region Parameter λ

The region parameter λ plays a critical role in segmentation. An incorrect λ value results in unbalanced prior and likelihood terms leading to an incorrect segmentation. Thus the optimal value of λ is determined to yield the best segmentation performance according to the error measure (3.8) using the training images. The choice of the error measure is motivated by the objective of the application—detection of the center of the *R. obtusifolius* in the image. Neighbourhood system $\eta^{(3)}$ was used in the experiments. As the earlier

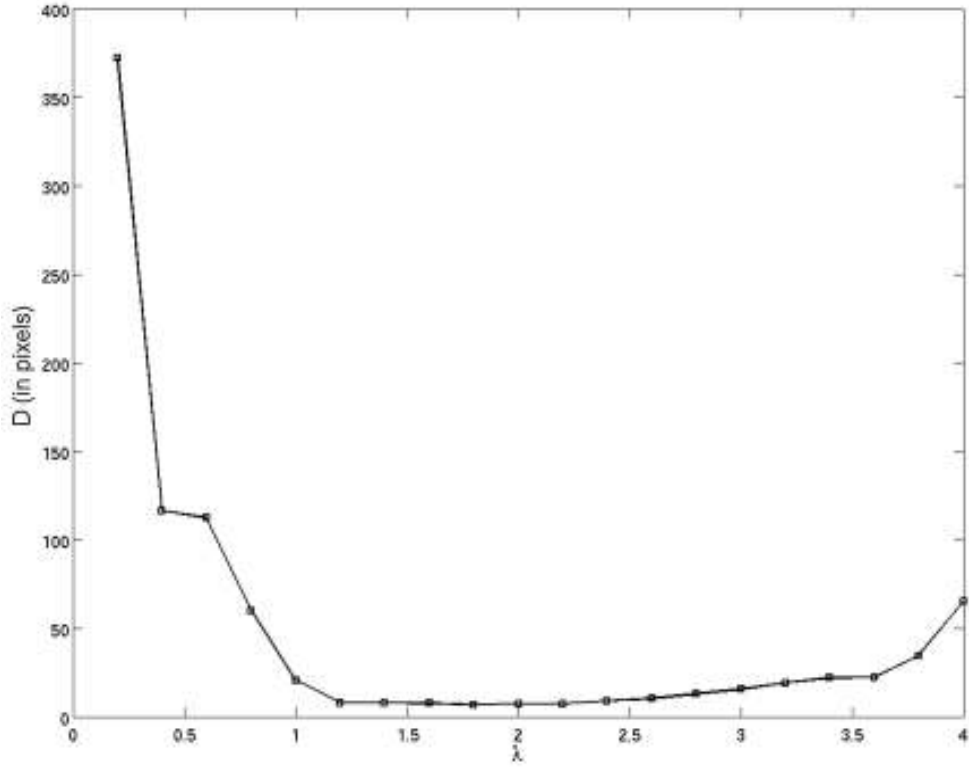


Figure 3.7: Plot of the error, D , between the estimated centre and the reference centre of *R. obtusifolius* against different values of λ averaged over 24 training images. The unit of the error measure is in millimeters

experiments could not establish the operating scale of the texture unambiguously, several different scales were used in this experiment. The plot of D at different values of λ is shown in Figure 3.7. A minimum error of $D = 6.85$ pixels was obtained at $\lambda = 1.8$. This is about which is 33.4 mm on the ground as each pixel corresponds to 1.22 mm on the ground at the original resolution. Other model parameters used in this experiment are as follows: The texture parameter values corresponding to filter $g^{(3)}$ and neighbourhood system $\eta_i^{(3)}$ are $\beta_{1,r} = 0.241$, $\beta_{2,r} = 0.275$, $v_r = 4.351$, $\beta_{1,g} = 0.242$, $\beta_{2,g} = 0.292$ and $v_g = 27.236$.

3.5 Segmentation Performance

Based on the feature selection process, the best texture features were obtained at the scale of $sz = 4$ and $\sigma = 0.7$ along with the region parameter $\lambda = 1.8$ and the neighbourhood system $\eta^{(3)}$. The performance of the segmentation algorithm was tested with 92 test images—43 grass-images and 49 rumex-

Table 3.2: Confusion Matrices of algorithm A1 and A2. TP, FP, FN and TN stand for true positive, false positive, false negative and true negative respectively.

		A1		A2	
		1	0	1	0
Actual	1	47(TP)	2 (FN)	42(TP)	7 (FN)
	0	0 (FP)	43(TN)	5 (FP)	38(TN)

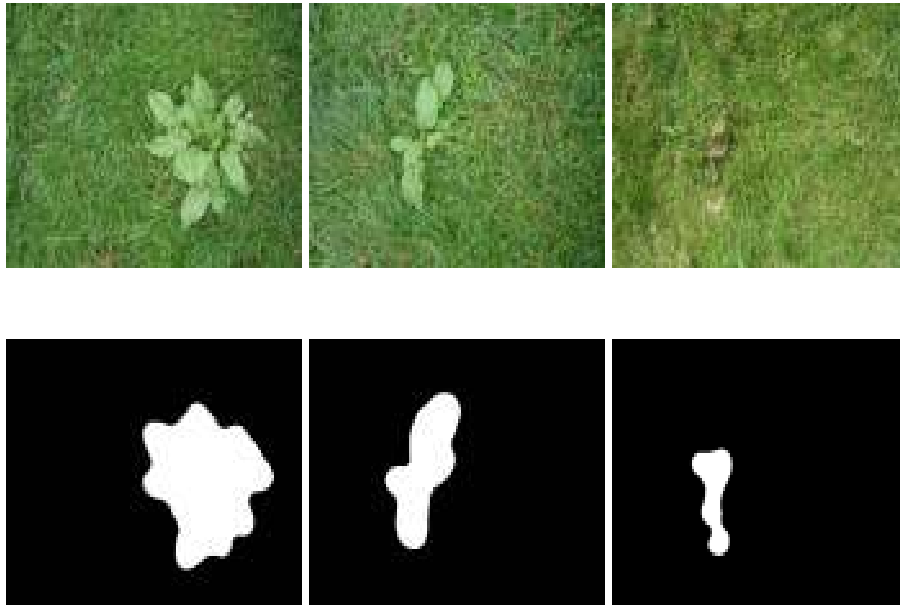


Figure 3.8: Segmentation results. The last column is an example in which A1 was successful in detecting the Rumex while the algorithm A2 failed.

images—and the results were compared with the algorithms A2. Table 3.2 shows the confusion matrix of A1 and A2. The detection rate of A1 is 97.8% while that of A2 is 87%. It is important to note that A1 also has better false negative and false positive rates than A2 showing that A1 is able to handle a wider variety of rumex-images. Finally, the segmentation error D , averaged over 49 rumex-images, of A1 and A2 are 56mm and 308mm, respectively. Figure 3.8 shows some examples of the segmentation results of A1.

The algorithm A1 takes 0.18 seconds to process a single image of size 576×432 pixels. Its computational speed, although slower than A2—which can process a single image at 0.05 seconds—is still fast enough for real-time application. The time comparisons were carried out on an Intel Core i5, 2.4 GHz processor with 4 GB RAM.

3.6 Discussion and Future Work

In this study we formulate the segmentation problem as a MAP-MRF inference problem because it has several advantages. Firstly, it is a mathematically sound way to aggregate the local relationships across the image for segmentation at a global level. Secondly, it provides a simple mechanism for incorporating prior(s). Prior models can be based on other properties of the weed and grass like plant shape, size, colour, an expert knowledge. This modularity allows the user to adapt the model based on application requirements. Thirdly, the MAP-MRF framework greatly reduces the implementation effort because a usually difficult process of determining the conditional probability is now reduced to specifying the clique potentials.

We choose GMRF to model the images over other MRFs because it was observed that the pixels intensities of both grass and rumex pixels had a Gaussian distribution. In recent years, however, more advanced methods have been developed for including spatial correlations like Conditional Random Fields (CRF). They are more advanced in the sense that they relax the assumption of conditional independence. CRF provide a rich set of modelling tools, however, optimization of these models are difficult (Lafferty et al., 2001). Also, due to the increased modelling flexibility, specifying the clique potentials becomes more involved.

Although we adopt graph cuts for optimizing the energy function, other methods like Iterated Conditional Mode (ICM) (Besag, 1986) or Simulated Annealing (SA) (Kirkpatrick et al., 1983) can also be used. They, however, suffer from severe limitations unlike graph cuts. For instance, the main drawback of ICM is its sensitivity to initial conditions which makes it unreliable. SA yields a globally optimal solution, however, it is computationally expensive and cannot be applied in real-time. The superiority of graph cuts over other optimization algorithm is discussed in detail in (Boykov et al., 2001).

During the experiment, we observed that the GMRF parameter estimates are very sensitive to homogeneity of the texture patches. Inhomogeneous training patches resulted in incorrect parameter estimates which were not representative of the desired textures leading to failed segmentation results. This constraint imposes a lower limit on the size of the weed that can be detected. For example, the algorithm fails to detect the weed shown in the Figure 3.9. Another reason for the difficulty in detection of small weeds is the bottleneck of the minimum patch size of the training image required for reliable parameter estimates. (Clausi, 2001) recommends that to get reliable GMRF parameter estimates the training patches should have the minimum size of 32×32 pixels. These shortcomings, however, bring forward new research issues like determining the robustness of the GMRF features in terms of the degree of inhomogeneity of texture according to some suitable measure(s)



Figure 3.9: An example image with a small *Rumex* plant on which the new algorithm fails

Table 3.3: Segmentation error of the new algorithm at different resolutions.

Resolution	D (in mm)	Time (in sec)
$(2304 \times 1728)/2$	137.8	1.2
$(2304 \times 1728)/3$	49.2	0.5
$(2304 \times 1728)/4$	26.6	0.2
$(2304 \times 1728)/5$	30.1	0.2
$(2304 \times 1728)/6$	28.7	0.1
$(2304 \times 1728)/7$	24.5	0.1

and quantifying the feasible size range of *R. obtusifolius* for successful detection. For the feature selection (Section 3.4.2) and results phase (Section 3.5), we restricted to work with images of size 576×432 to ensure that the segmentation algorithms operated on the same texture so that the results were comparable to earlier experiments. Further experiments, however, revealed that the performance of A1 can be improved further by using images with smaller resolution. Table 3.3 shows the segmentation error D and the computation time (for processing a single image) at different resolutions. Note that the optimal algorithm parameters are different for different image resolutions and it is necessary to recompute them every time the resolution of the image is changed. As we see the segmentation performance increases with the decrease of image resolution, however, it increases the lower-limit of the size of the *Rumex* plants that the algorithm can detect. It should be noted that images with small *Rumex* plants were excluded from this experiment as D can only be computed if the plant is detected. In conclusion it is not possible to specify a unique parameter setting that would work on *all* *rumex*-images. From an application standpoint, it is advisable to tune the algorithm based on the profile of the plant like its size, age and density.

One of the major issues with the earlier *R. obtusifolius* detection methods

is their sensitivity to variation in ambient light. This problem was either being addressed by using artificial lighting sources or by calibrating the system during each run. With this algorithm one can, however, be more efficient in handling illumination variation.

Despite satisfactory performance of the MRF and the segmentation algorithm, there is still room for improvement. For example, instead of choosing the best resolution to work with, information across multiple resolutions can be combined in a multi-resolution MRF to yield texture features invariant to image size and texture scale. This might help tackle the problem of detection of small weeds. Currently our assumption of piecewise constant prior is simplistic. A more realistic prior model will aid in determining the region boundaries more accurately. The current model also ignores the colour information and uses only the red channel of the RGB image. Thus we presume that the algorithm may confuse a large patch of soil for a plant. This, however, is not a major concern as the step of discriminating plant from soil—or any other objects with different colour than plant—can be a part of the pre-processing step. Colour information could be incorporated by combining MRFs pertaining to different colour channels to form a multi-channel MRF, similar to multi-resolution MRF. Explicitly encoding the colour information can allow detection of different weed species other than *R. obtusifolius*.

The agricultural domain is a rich platform for applications of computer vision. It poses significant challenges due to the inherent uncertainty in the environment in which the systems have to operate. As a result, despite significant development in the field of computer vision, few have found applications in the agricultural domain and are often restricted to operate in controlled environments. This is another illustration that shows the divide between theory and applications in computer vision. Often a new image processing algorithm is tested on either synthetic images or in controlled environment and as a result fails to work on natural images taken under natural conditions. By developing a new real-time segmentation algorithm for detecting *R. obtusifolius* capable of operating in natural conditions we hope to help to bridge the gap between theory and application.

Conclusions

In the current study we developed a segmentation algorithm for detection of *R. obtusifolius* based on MRF texture features and demonstrated its superior accuracy and speed by comparing it with a spectral power based algorithm. We showed that the new algorithm has a detection accuracy of 97.8% and has an error of 92mm in the estimation of the centre of the weed in the image. Furthermore, we demonstrated that it takes 0.18 seconds for processing an single image thus making it feasible for real-time application.

Chapter 4

Image-Based Particle Filtering for Navigation in a Semi-Structured Agricultural Environment ¹

Abstract

Autonomous navigation of field robots in an agricultural environment is a difficult task due to the inherent uncertainty in the environment. The drawback of existing systems is the lack of robustness to these uncertainties. In this study we propose a vision-based navigation method to address these problems. The focus is on navigation through a corn field in an outdoor environment where the robot has to navigate through a corridor formed by two plant rows, detect the end of the rows, navigate the headland and turn into another corridor under natural conditions. The method is based on a Particle Filter (PF) using a novel measurement model, where we construct a model image from the particle and compare it directly with the measurement image after elementary processing, such as down-sampling, excessive green filtering and thresholding. The new measurement model does not extract features from the image and thus does not suffer from errors associated with the feature extraction process as is the case in feature-based measurement models. In conclusion, we show how PF can be used for robust navigation of a robot in a semi-structured agricultural environment with inherent uncertainty. We demonstrate the robustness of the algorithm through experiments in several fields with different row patterns, varying plant sizes and diverse lighting conditions. To date we have logged over 5 km of successful test runs in which the robot navigates through the corridor without touching the plant stems, accurately detecting the end of the rows and traversing the headland.

¹submitted for publication

4.1 Introduction

In recent years, there is an increasing demand to reduce the use of chemicals in agriculture due to environmental, economic, and food safety reasons. Potential ways to achieve this is through precision agriculture and organic farming. These methods, however, require automation of many agricultural operations including harvesting, scouting, tilling, sowing and weed control. Automation is necessary in precision agriculture because the detection of events that might require attention (e.g. weeds, end of row, wrong planting density, lack of nutrients, presence of diseases or insects) and the subsequent actuation can be rather time consuming. This makes the process slow and thus labor intensive if a driver has to be present all the time. If a robot is able to complete the task without exploiting any human resources, even time-consuming automatic processes will be affordable. For a machine (or robot) to perform these tasks automatically, it should be able to navigate through the environment autonomously.

The importance of autonomous navigation in agriculture was realized as early as 1958 when Morgan et al., developed an autonomously navigating tractor (Morgan, 1958). Since then, many different navigational methods, using different sensors, have been developed for agricultural robots. For instance, navigation can be based on leader cables (Tillett and Nybrant, 1990), optical triangulation (Shmulevich et al., 1989), Global Position System (GPS) (Heidman et al., 2002; Slaughter et al., 2008; Stoll and Dieter Kutzbach, 2000), laser-based sensors (Ahamed et al., 2004; Barawid Jr. et al., 2007) and machine vision sensors (Kondo and Ting, 1998; Foglia and Reina, 2006; Van Evert et al., 2011). Good review papers on different autonomous navigating systems in agriculture can be found in various papers (Wilson, 2000; Keicher and Seufert, 2000; Reid et al., 2000; Li et al., 2009)

Autonomous navigation in an agricultural environment is a difficult task due to the inherent uncertainty in the environment. The robot is in a dynamic and non-deterministic environment with many sources of noise. For instance, noise can be due to uneven terrain or varying shapes, sizes and colour of the plants or inconsistency in the environment structure. The problem is further compounded by hardware related noise like imprecise sensor measurements, wheel-slippage, controller and actuator noise. Designing a navigation method to handle the total uncertainty due to the multiple and often complex sources of variation is not straightforward.

One of the main drawbacks of the existing prototype systems is the lack of robustness to the various sources of uncertainties of the environment (Li et al., 2009). The goal of this study is to develop a robust vision-based localization and navigation method based on probabilistic methods to account for the different uncertainties in the environment. We focus on navigation through a

corn field in an outdoor environment where the robot has to navigate in the corridor, detect the end of the rows, navigate in the headland and turn into another corridor under natural conditions. We restrict ourself to vision-based navigation because a camera is the most cost effective and information rich sensor compared to others (Chen et al., 2007).

The use of probabilistic methods for robot navigation has been subject to extensive studies for the past decade as it provides a good mathematical framework to deal with uncertainty and noisy conditions (Thrun et al., 2005). In agriculture, one of the earliest vision-based robots to use probabilistic methods is a horticulture robot that uses the Extended Kalman Filter (EKF) to navigate along the rows of plants (Hague and Tillett, 1996). In the correction step of the EKF, a feature-based measurement model is used, where features are lines extracted using the Hough transform (Duda and Hart, 1972) from the measurement image. The main drawback of the system is that the linear approximation of EKF results in the accumulation of errors as time progresses which may lead to system failure. The authors acknowledge this potential source of failure. Secondly, the feature extraction process itself introduces uncertainties into the system. For instance, failure of the Hough transform to extract the ‘correct’ lines will lead to navigation failure. Southall et al., (2002) developed a robot to navigate in a field where plants are grown in a grid like pattern, based on EKF. Using the knowledge of the environment they built a grid-based model of the robot view which is used as the measurement model. One of the shortcomings of this system is due to the assumption that the local view has a fixed grid structure. This will lead to errors because the grid size and shapes are not often fixed in agricultural fields. Other similar methods include (Marchant, 1996; Marchant and Brivot, 1995).

In this study, we overcome the problems due to EKF by using a Particle Filter (PF) for robot localization and navigation (Dellaert et al., 1999). The use of PF for row following in an agricultural field is promising as shown in (Bergerman et al., 2012) where a vehicle with a navigation system based on PF autonomously navigates in orchards. Secondly, we propose a novel measurement model to mitigate the uncertainty introduced by the feature extraction process in feature-based measurement models such as those that use Hough transform. We achieve this by constructing a model-image (Section 4.2) from the particle state vector and comparing it directly with the image (after elementary processing, like excessive green filtering and down-sampling). As far as we know, this is the first attempt to directly model the image from the particle state vector instead of using a feature extraction process in between.

The paper is arranged as follows. Section 4.2 describes the image based particle filter along with the details of the new measurement model. In Section 4.3 we demonstrate the performance and robustness of the new navigation algorithm in various fields with diverse environmental conditions followed by

the discussion section where we point out limitations and extensions of the method and the scope for future research.

4.2 Image Based Particle Filter

We use a small robot which has to navigate through a field of corn (Figure 4.3). The robot has a downward-looking camera at a height of 1.65 m, through which it senses the world. The task of the robot is to navigate through a corridor formed by two adjacent rows of maize plants, detect the end of the rows, navigate in the headland and return into another row under natural conditions. This can be divided into two distinct situations: navigation between the rows and navigation on the headland. The top row of Figure 4.1 show two typical camera views when the robot is between the rows and on the headland, respectively.

4.2.1 Local World

Our local world is a field with maize plants that consists of two parallel rows of equal width (rw), with a distance (rd) between them. The end of the rows is indicated by el for the left row and er for the right row. The rw , rd , el and er are measured in meters. The robot is characterized by its main axis (direction of travel straight ahead) and the control point (CP) half way between the wheels. The position of the robot in this world is parametrized by the robot heading (h) and lateral deviation (l). The heading is the angle between the robot's main axis and the reference axis (line along the centre of the rows) and is measured in radians. Lateral deviation is the perpendicular distance between the robot and the reference axis measured in meters. The parameters together represent the robot-environment state at time t and is denoted by $X_t = (h, l, rw, rd, el, er)$.

The state of the robot is estimated by a PF. The PF maintains a posterior distribution over X_t given the past sensor measurements $Z_{1:t}$ and the past controls $U_{1:t}$. This posterior distribution is approximated by a set of samples called particles. A particle is a possible realization of X_t . The posterior probability $p(X_t|Z_{1:t}, U_{1:t})$ at time t can be expressed in terms of that at time $t - 1$ in the formula

$$p(X_{1:t}|Z_{1:t}, U_{1:t}) = \frac{p(Z_t|X_t)p(X_t|X_{t-1}, U_t)}{p(Z_t|Z_{1:t-1})} p(X_{1:t-1}|Z_{1:t-1}, U_{1:t-1}) \quad (4.1)$$

where, $p(Z_t|X_t)$ represents the measurement update step and is given by the measurement model (also called the likelihood model), $p(X_t|X_{t-1}, U_t)$ represents the motion update step and is given by the motion model, $p(Z_t|Z_{1:t-1})$

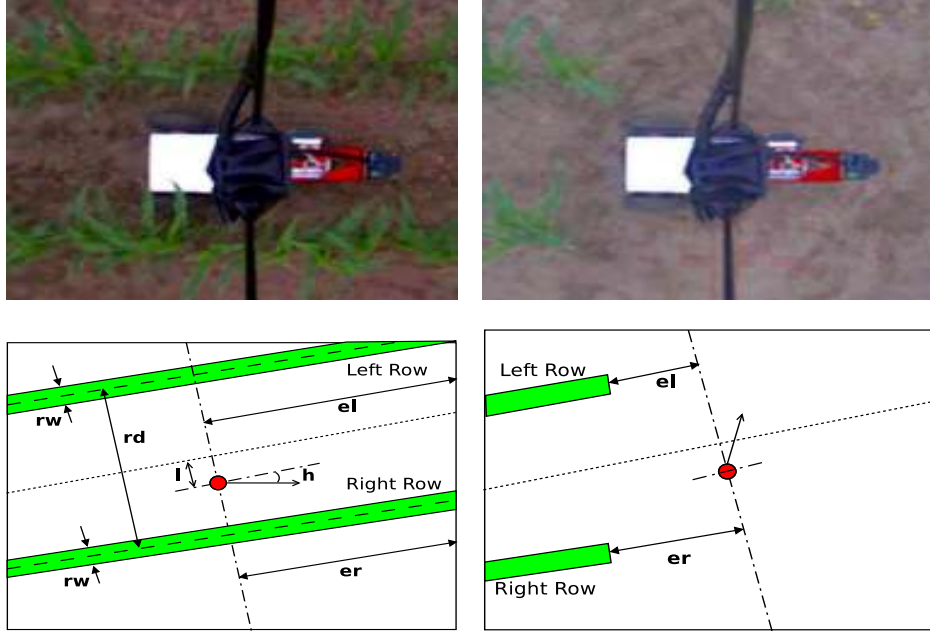


Figure 4.1: Typical camera views when the robot is in between the rows (top left) and on headlands (top right). Figures in the bottom row illustrate the parametrization of the local world. The robot control point is represented by the red circle and its travelling direction is indicated by the arrow attached to it. h is the robot heading, l is the lateral deviation, rd is the distance between the rows, rw is the row width, el is the end of left row, and er is the end of right row.

is the normalizing constant, and $p(X_{1:t-1}|Z_{1:t-1}, U_{1:t-1})$ is the posterior distribution at previous time step $t-1$. Details of the measurement model will also be discussed below. The specific form of (4.1) is indicative of the recursive nature of a particle filter where the posterior at previous time step is updated by multiplying it with the motion model and the measurement model obtained from the current time step.

When the robot is between the rows, the navigation algorithm steers the robot to follow the reference line along the centre of the rows. At the rows ends, it stops 0.6 m outside the rows, makes a 90 degree turn and aims to follow a line parallel to the end of the rows (van Evert et al., 2011).

4.2.2 Motion Model

The motion model describes how the state vector changes from one time step to another. When the robot is between the rows, it follows the path along the centre of the corridor and when travelling on headland, it follows the path along an imaginary line parallel to the line connecting the row ends. In both

cases, the robot aims to follow a specified reference line. We assume that the distribution of the initial state (X_0 at $t = 0$) of the robot is known, and that the gyroscope and wheel encoders on the robot provide the control information $U_t = (dx, dh)$ where dx is the displacement of the robot along its heading and dh is the turning angle of the robot. Now, the motion model is given by

$$\begin{aligned} h_t &= h_{t-1} + dh + \varepsilon_h \\ l_t &= l_{t-1} + dx \sin(h_{t-1} + dh) + \varepsilon_l \\ rw_t &= rw_{t-1} + \varepsilon_{rw} \\ rd_t &= rd_{t-1} + \varepsilon_{rd} \\ el_t &= el_{t-1} - dx \cos(h_{t-1} + dh) + \varepsilon_{el} \\ er_t &= er_{t-1} - dx \cos(h_{t-1} + dh) + \varepsilon_{er} \end{aligned} \quad (4.2)$$

where $\varepsilon_h, \varepsilon_l, \varepsilon_{rw}, \varepsilon_{rd}, \varepsilon_{el}, \varepsilon_{er}$ are independent Gaussian noise applied to the corresponding state variables.

A complication arises because the end of the rows is frequently not in view of the robot's camera. According to the motion model, the values of el and er are constantly decreased. When the end of row is not in robot's view, the el and er values should not be decreased. This situation is dealt with by re-initializing el and er in a fraction of the particles at regular intervals.

4.2.3 Measurement Model

The measurement $Z_t = (z_1, z_2, \dots, z_m)$ at time t is a binary image of size $m = r \times c$ pixels. Pixels for which the value is one ($z_i = 1$) are mostly green and indicate the probable presence of plant material; pixels for which the value is zero ($z_i = 0$) indicate the probable absence of plant material. The binary image Z_t is obtained from the camera image after preliminary image processing. First, the RGB image from the camera is down-sampled to $r \times c$ pixels, then it is converted to an excessive-green image $I = (-R + 2G + -B)/3$ which is thresholded to yield the binary image.

The measurement model for Z_t depends on a model image. The model image is a discretized version of the local robot world as parameterized by the particle $X_t^{(i)}$ and has the same number of pixels as the measurement image. If we assume that the pixels in Z_t are independent and Bernoulli distributed with parameters $Q = (q_1, q_2, \dots, q_m)$, then the measurement model is given by the likelihood

$$p(Z_t | X_t^{(i)}; Q) = \prod_i q_i^{z_i} (1 - q_i)^{(1-z_i)} \quad (4.3)$$

with q_i being the probability that pixel i is green.

As our local world consists of plant rows and soil, pixels in the model image belong either to a plant row (in-row region $\mathcal{S}$) or to soil (out-row region

$\mathcal{O}$). A parsimonious model is obtained by setting $q_i = \theta_{\mathcal{I}}$ when the pixel belongs to the in-row region and $q_i = \theta_{\mathcal{O}}$ when it falls in the out-row region where $\theta_{\mathcal{I}} > \theta_{\mathcal{O}}$ as we expect more green in in-row region relative to out-row region. The likelihood can then be rewritten as

$$p(Z_t|X_t^{(i)}; \Theta) = \prod_{r \in \{\mathcal{I}, \mathcal{O}\}} \prod_{i \in r} \theta_r^{z_i} (1 - \theta_r)^{(1-z_i)} \quad (4.4)$$

where $\Theta = \{\theta_{\mathcal{I}}, \theta_{\mathcal{O}}\}$.

4.2.4 Fuzzy Membership

Hard clipping the excessive green image to a binary image heavily depends on the threshold value used. This may lead to rather large differences in the binary image, under only slightly changing lighting conditions. To avoid this effect, we propose to use soft clipping over a range $[a, b]$ instead of hard clipping. This implies that z_i is no longer a binomial variable but a pseudo-binomial variable and it is calculated as follows. Let I_i be the excessive green value, which is transformed to a value z_i between 0 and 1 as follows.

$$z_i = \begin{cases} 0 & : I_i < a \\ 1 & : I_i > b \\ \frac{I_i - a}{b - a} & : a \leq I_i \leq b \end{cases}$$

Note that z is no longer a binary variable anymore, but can assume values between 0 and 1. Hence the distribution now becomes pseudo-binomial instead of binomial. This is according to the theory for analysis of fractions as e.g. described by McCullagh and Nelder (McCullagh and Nelder, 1989). The equations (4.3) and (4.4) are identical for the pseudo-binomial case.

4.2.5 Attenuation Factor

Within the PF framework the likelihood model determines the weight of each particle obtained from the motion model. After computing the weights, the particles are resampled according to their weights to yield the posterior distribution. The peakedness of the likelihood model is a measure of the diversity of the posterior distribution. A highly peaked likelihood means less diversity of particles. The likelihood as described in section 4.2.3 is highly peaked because the dependency between the pixels is not considered. Pixels in the image are dependent due to correlations prevalent in the local world. For instance, pixels are dependent because they belong to the same plant and there may be correlation at larger distances due to the regularity in their growth pattern. In the measurement model, however, every pixel is regarded as an independent measurement. This leads to a large number of measurements

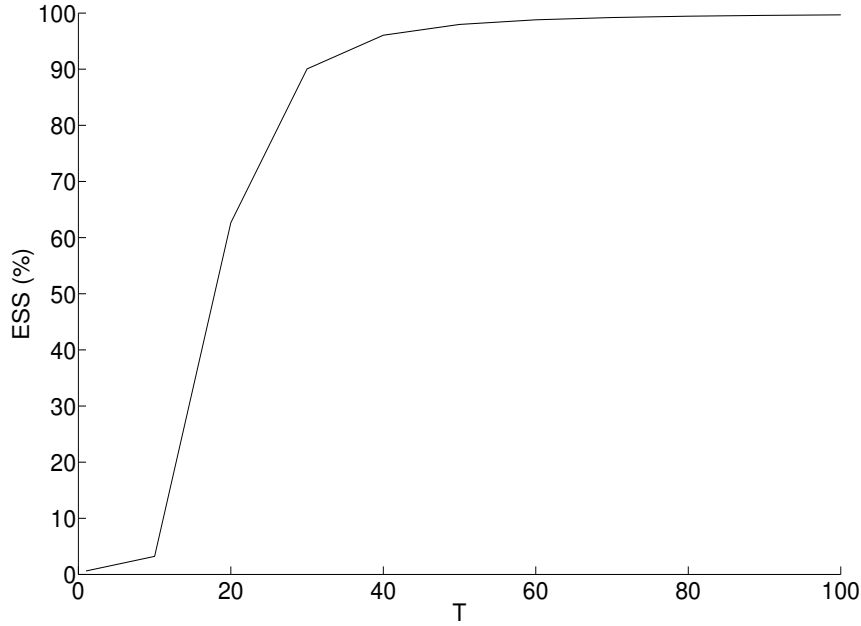


Figure 4.2: Figure shows the graph of ESS vs T , where the likelihood is tempered by taking the T^{th} root. As T increases the ESS also increases indicating a smoother likelihood function.

resulting in a highly peaked likelihood where the likelihoods of two similar particles differ by orders of magnitude. For example, for an image of size 40×60 pixels, if two particles have a difference of 0.04 m (one pixel) in lateral deviation, their likelihoods may differ up to a factor of $3.37\text{e}+216$ when $\theta_{\mathcal{J}} = 0.8$ and $\theta_{\mathcal{O}} = 0.02$. This peakedness causes particle degeneracy in the PF. This problem is addressed by expressing the dependency between the pixels by means of the attenuation factor T (Thrun et al., 2005). Instead of counting every pixel as an individual measurement, T pixels are considered as a single measurement and reduce the number of measurements by a factor T by raising the likelihood to the power $1/T$.

The appropriate value of T can be determined using the effective sample size (ESS) of the PF (Doucet and Johansen, 2009). Figure 4.2 shows the plot of ESS vs T averaged over the iterations of a PF run. As can be seen at small values of T , the ESS is close to 1% indicating most particles have negligible weights. But as the value of T is increased, the ESS increases indicating that the weights are more evenly distributed among the particles. We selected the value of T such that 80% of the particles are retained after the resample step.

4.3 Experiments and Results

4.3.1 Robot

The robot employed in this study consists of a chassis with three wheels, with overall dimensions $0.8 \text{ m} \times 0.45 \text{ m} \times 0.3 \text{ m}$. It has an actuated front wheel as the steering wheel that is affected by commands from a control program by means of a CAN-bus and it has two rear wheels that do not have the ability to steer. All wheel units are equipped with incremental encoders to measure the rotational speed. In addition, the front wheel unit is equipped with an angle sensor to measure the steering angle. The driving speed of each wheel depends upon the target speed of the control point, the location of the wheel with respect to the control point and the turning radius. An electronic box between the rear wheels houses a mini-ITX computer with a 2.4 GHz Intel Core2 Duo processor running Windows XP operating system. The robot is controlled by a custom C# software which uses OpenCV library for image processing. Energy to the computer and the wheel units is provided by three 12 V NiMH racing packs: 1 for the front wheel unit, one for both rear wheel units, and one for the PC.

The wheel units are equipped with incremental encoders to measure rotational speed of each wheel. The controller steers the robot by trying to follow a predetermined path pattern. A simple string notation is used to indicate the path pattern. For example the string '1L-2R' is an instruction to navigate to the end of the corridor and turn into the adjacent corridor on the left, navigate to the end and then turn into the second corridor to the right. When the robot is in the corridor, it follows the in-row reference line and when it is in the headland it follows the headland reference line. In either case the target steering angle of the front wheel unit is given by $\gamma = -h + \tan^{-1}(-a/b)$ where h is the robot heading, a is the perpendicular distance of the robot from the reference line and b is the target distance along the reference line as indicated in Figure 4.3d. Note that $a = l$ if the robot is between the rows. After detecting the end of the plant rows, the robot continues following the rows until its control point is at a given distance beyond the end of the rows. It then makes an on-the-spot turn to position itself parallel to the headland reference line and continues the headland navigation. Upon reaching the middle of the next corridor (new target corridor according to string pattern), it comes to a full stop, turns to position itself along the middle and starts following the new in-row reference line which is along the middle of the new corridor.

The in-row reference line is shifted so as to be along the middle of the corridor along which the robot is travelling. An element of navigation on the headland is counting how many rows have been crossed in order to follow the string pattern. To do so, a row counter is used that is initially set to zero and is updated when the lateral deviation (l) of the robot is larger than half the row

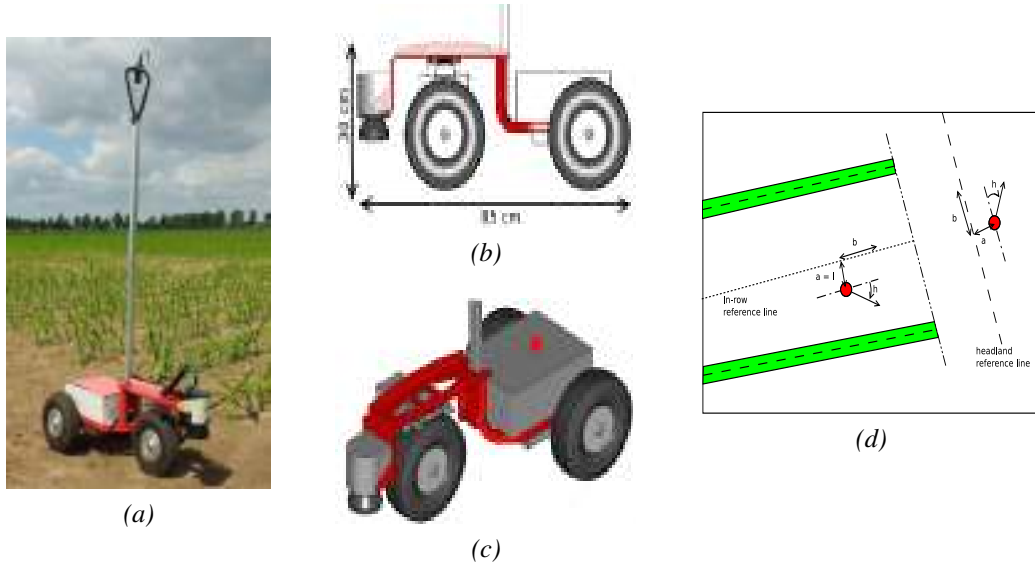


Figure 4.3: (a) The robot with the mounted camera in a field. (b) The profile view of the robot depicting the length and height of the robot drawn to scale. (c) Perspective view of the robot showing the electronics box between the rear wheels. The red dot between the two rear wheels is the designated control point that indicates the position of the robot in the environment. (d) The target steering angle γ as it is computed by the controller. Between rows the robot follows the in-row reference line (dotted line) and within the headland the robot follows the headland reference line (dashed line).

distance (rd).

The robot is equipped with a downward-looking camera (uEye UI-1220, IDS Imaging Development Systems GmbH, Obersulm, Germany) at a height of 1.65 m, through which it senses the world. The camera captures an RGB image of size $width \times height = 752 \times 480$ pixels where height is in the travelling direction and width is perpendicular to it. The input image is subjected to preliminary processing to obtain the measurement image as described in Section 4.2.3.

4.3.2 Calibration

The experimental data consists of several sequences of images taken by the robot's camera as it navigates through the field. The sequences are from different maize fields with different plant size, row structure—straight rows, curved rows, gaps within the rows—and under different lighting conditions. The number of images in each sequence depends on the length of the robot run where the camera records data at 10 Hz. For every image in the sequence,

the ground truth for the most important state variables, heading (h_r), lateral deviation (l_r), left end-of-row (el_r) and right end-of-row (er_r) was established by means of hand annotation. These ground truth values were used to evaluate the performance of the PF.

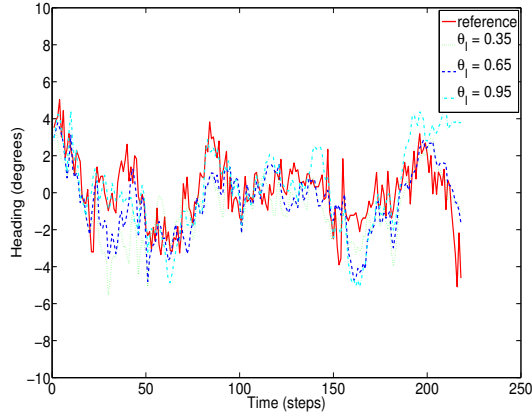
Experiments consisted of running the PF with several image sequences for various values of $\theta_{\mathcal{J}}$ and $\theta_{\mathcal{O}}$. In all the experiments, the PF was initialized with 256 particles. This number is empirically determined based on real-time computational constraints – a balance between update frequency (10 Hz) and number of particles. The particles are initialized as follows, h has a uniform distribution in the range $[-10, 10]$ degrees, l has a uniform distribution in the range $[-0.1, 0.1]$ m, rw has a uniform distribution in the range $[0.05, 0.6]$ m, rd has a uniform distribution in the range $[0.5, 1.5]$ m and el and er have a uniform distribution in range $[1.2, 1.4]$ m. During the update step, each component of the state vector is updated based on the motion model where the error components are sampled from the following distributions: $e_h \sim N(0; 1.0)$ (degrees), $e_l \sim N(0; 0.01)$ (m), $e_{rw} \sim N(0; 0.01)$ (m), $e_{rd} \sim N(0; 0.01)$ (m), $e_{el} \sim N(0; 0.02)$ and $e_{er} \sim N(0; 0.02)$ (m). The particles are resampled according to importance resampling (Thrun et al., 2005).

After the resample step, the weighted mean of the posterior distribution $\hat{X}_t$ is the PF estimate of the state X_t at time t . Thus, for every image in the sequence, its PF estimate can be compared with the corresponding ground truth. Figure 4.4 shows the result of one such comparison for $\theta_{\mathcal{O}} = 0.02$ and three different values of $\theta_{\mathcal{J}}$ in $\{0.35, 0.65, 0.95\}$. The six diagrams in the figure correspond to the six state variables. Variables h , l , el and er are plotted along with their corresponding ground-truth data h_r , l_r , el_r and er_r , respectively, while rw and rd have no reference.

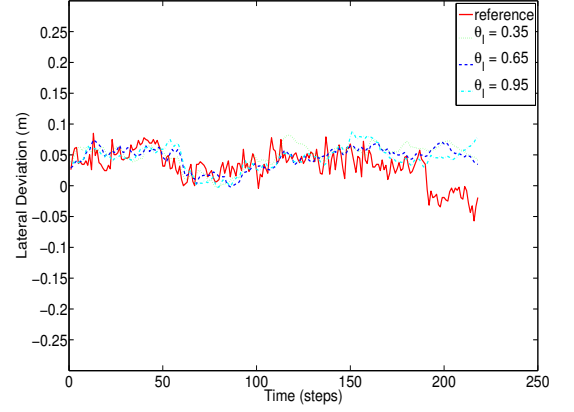
As we see in Figure 4.4a the PF successfully tracks h_r except when the robot reaches the end of the rows where it diverges. In the case of lateral deviation, the PF tracks l_r throughout the run (Figure 4.4b). When the end of the rows is in sight, the PF is able to detect this and tracks them as shown in Figures 4.4e and 4.4f, but when the end of the rows is not in sight, the values are of course not correct.

We also observe that the estimation of h , l , er , el and rd is not sensitive to a large range of values of $\theta_{\mathcal{J}}$. This is in contrast with rw (Figure 4.4c) which is inversely proportional to $\theta_{\mathcal{J}}$. Similar affects were observed for different values of $\theta_{\mathcal{O}}$ (not illustrated).

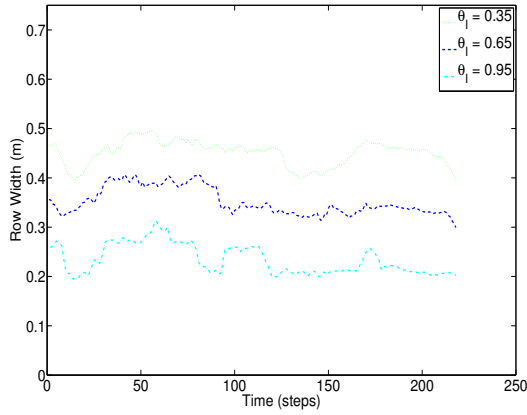
The algorithm performs well over a broad range of values of $\Theta = \{\theta_{\mathcal{J}}, \theta_{\mathcal{O}}\}$ and its actual value is thus not critical. By setting the parameter halfway in the range we expected it to perform well over the broadest range of circumstances. You can nicely see the compensating effect of the row width with the fraction Θ used. A lower Θ results in a wider row, including soil (0) pixels of the outer-row region. Hence the fraction automatically adjusts itself. This



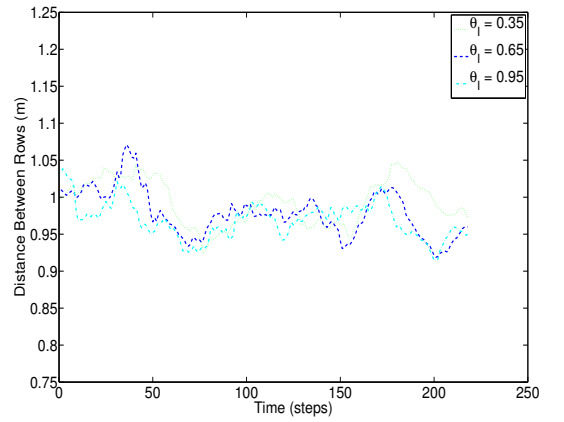
(a) Heading



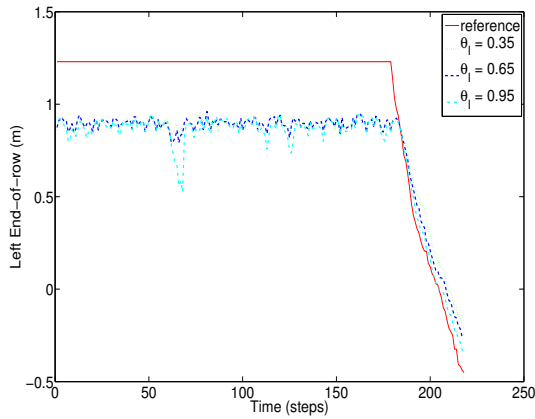
(b) Lateral Deviation



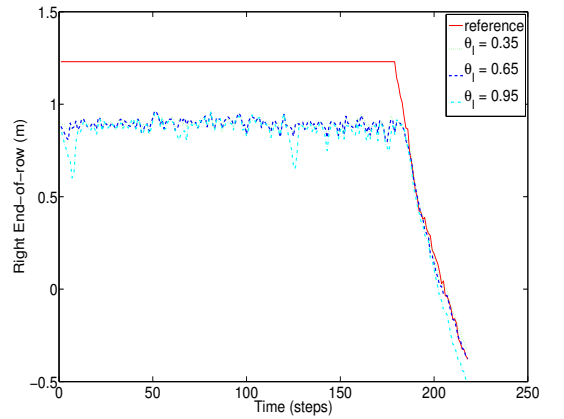
(c) Row Width



(d) Row Distance



(e) End of Left row



(f) End of Left row

Figure 4.4: The result of PF estimation of X_t of a robot run, for three values of $\theta_{\mathcal{J}}$. The dotted curve in green shows the PF estimate with $\theta_{\mathcal{J}} = 0.35$, the dashed curve in blue shows the PF estimate when $\theta_{\mathcal{J}} = 0.65$, and the dashed and dotted curve in cyan shows the PF estimate when $\theta_{\mathcal{J}} = 0.95$. The value of $\theta_{\mathcal{O}}$ was fixed to 0.02 in all three cases. The solid line in red indicates the reference data.

Table 4.1: Mean $\pm$ standard deviation of the RMSE of five test runs.

Parameter	RMSE
Heading (degrees)	3 ± 1
Lateral Deviation (m)	0.04 ± 0.007
Left end-of-row (m)	0.22 ± 0.04
Right end-of-row (m)	0.24 ± 0.05

exchangeability is one of the reasons not to estimate both fraction and row width in the same go, but to fix one of them.

The model was tested with five independent image sequences. Table 4.1 gives the mean and standard deviations of the Root Mean Squared Error of these five test runs. The tests were carried out with $\theta_{\mathcal{J}} = 0.65$, $\theta_{\mathcal{O}} = 0.02$. The values show that the state vector is estimated with a high degree of precision.

4.3.3 Field Tests

The field in which the robot was tested consists of rows of maize plants with a well-defined headland. These rows may be either straight or curved. Additionally, there may be gaps in the rows because of missing plants. The rows are spaced 0.75 m apart, being the distance at which the seeding took place. Figure 4.5 shows four example images of the various field conditions in which experiments were conducted. The first row of images shows fields with various row patterns in which experiments were carried out. The second and third rows of images show the lighting conditions and plant sizes under which the test were conducted. This includes low light causing poor visibility, bright sunlight causing shadows and plant sizes ranging from 10 cm to 50 cm.

A test run consisted of two laps of a 60 m long path alternating through the five rows. The robot started at the beginning of the first corridor, travelled along the plant rows, exited into the headland on the other end of the corridor, travelled along the headland to the next corridor and turned into it and so on. Figure 4.6 shows the trajectory that the robot followed during the test runs. Each row is approximately 12.6 m long while the headlands are about 7 m. The total distance covered by the robot in a single test run is 140 m. The path is designed such that various environmental effects during the robot run as tested for. For instance, effects of shadows due to sunlight and plant orientations due to wind direction. To date we have recorded more than 5 km in various field conditions. Table 4.2 shows the test runs conducted over the years 2011-2012 at different times of the day. In all cases the robot successfully traversed the specified path. That is, it navigates without touching the stems of the plants while in the corridor, accurately detects the end of row,

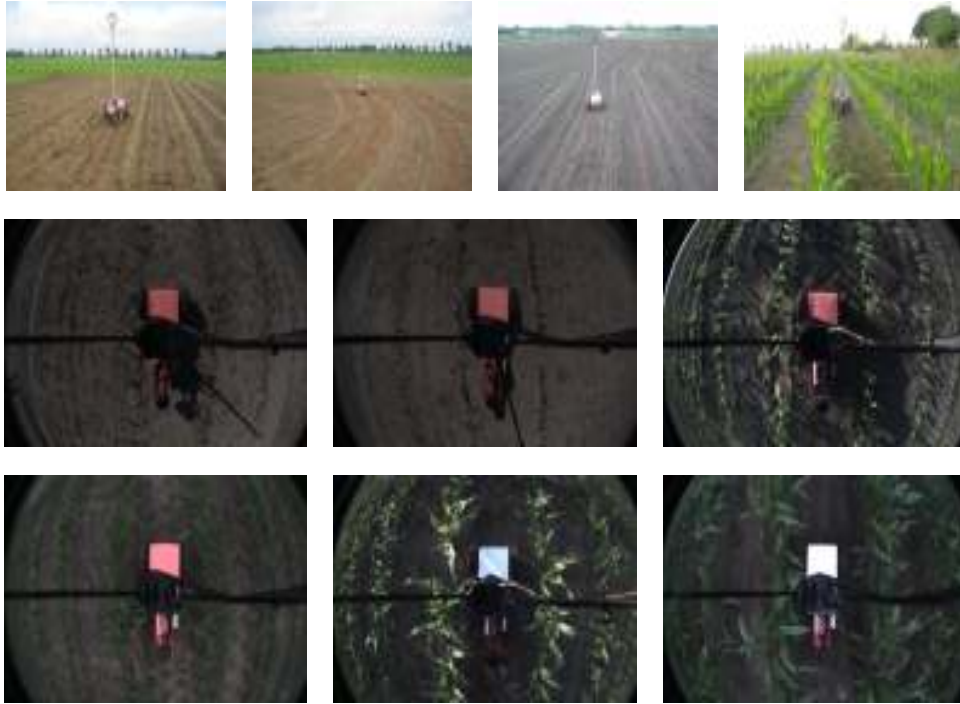


Figure 4.5: (Top row) Example of Maize fields in which the robot was tested. (Second and third row) Example images show the different lighting conditions in which the experiments were conducted along with various plants size. The poor quality of some of the images in the second and the third row illustrate the extremity of the field conditions in which the tests were conducted.

Table 4.2: Number of test runs during 2011-2012 at different times.

Date/Time	9:00-12:00	12:00-18:00	18:00-21:00
June, 2011	-	9.5	7.5
May, 2012	4.5	-	-
June, 2012	2	2.5	7.5
July, 2012	1	1	0.5

traverses the headland at the specified distance from the row ends, and turns automatically at the requested corridor.

The navigation method was also tested for its computational speed and real-time performance. As concerns the speed, the algorithm processes up to 10 images per second. This image rate allowed the robot to successfully navigate the field at a speed up to 0.7 m s^{-1} (2.5 km h^{-1}). The computational speed of the algorithm is also influenced by the size of the image, which should be small enough to ensure real-time operability of the algorithm. Therefore the camera image was downsized to retain only the essential

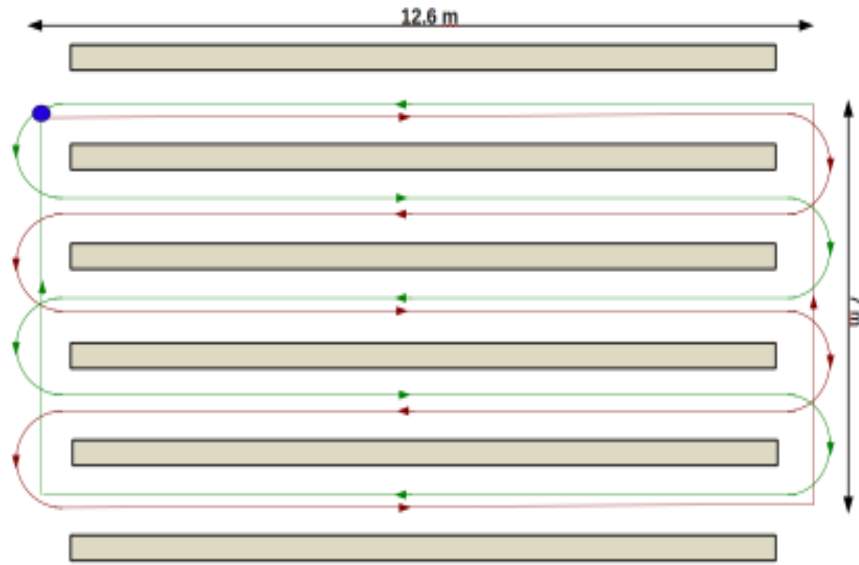


Figure 4.6: The path followed by the robot during the test runs. The blue circle indicates the starting position. For illustrative purposes, the red and green paths do not exactly run halfway between the rows.

information needed to accurately determine the orientation of the rows. Due to the difference in the information content when the robot is in the row and in the headland, different image sizes are used for the two situations. Particularly, when the robot is in the row the image is 60×47 pixels covering an area of $2 \text{ m} \times 1.5 \text{ m}$. Here 2 m is along the direction of the travel and 1.5 m perpendicular to it. In the headland the image is 90×141 pixels and covers an area of $2 \text{ m} \times 3 \text{ m}$ where 2 m is along the direction of the travel and 3 m is perpendicular to it. Increasing the image size in the headland ensures that sufficient information is captured to estimate the robot position while keeping the computational speed within the real-time operating limit. The speed on the headland was half the speed in the corridor.

4.4 Discussion

Experiments showed that estimation of all components of the state vector except rw is robust to changes in $\theta_{\mathcal{J}}$ and $\theta_{\mathcal{O}}$. rw is increased when $\theta_{\mathcal{J}}$ is large in comparison with the plant cover. Influence of values $\theta_{\mathcal{J}}$ and $\theta_{\mathcal{O}}$ is noticeable only when plants are small or plant density is low. In such situations extreme values like $\theta_{\mathcal{J}} = 0.95$ leads to erroneous estimates of rw , which eventually leads to the failure of the algorithm. This, however, is easily prevented by a rough calibration or estimation of $\theta_{\mathcal{J}}$ according to the specific stage of plant

growth. This allows the possibility to use different values for young plants (less than 5 cm in height) and older plants (greater than 50 cm in height) as confirmed in the field experiments. The results showed that in case of young plants the algorithm works best with the values $\theta_{\mathcal{J}}$ in the range $[0.2, 0.4]$. Similarly, with older plants the best values of $\theta_{\mathcal{J}}$ was in the range $[0.6, 0.8]$. It is worth noting that the value of $\theta_{\mathcal{J}}$ also influences the sensitivity of the algorithm to detect the end of row. Using high values of $\theta_{\mathcal{J}}$ makes the algorithm more sensitive to end of rows as shown by the cyan curve in Figure 4.4e between time steps 60 and 75 where the value of el decreases due to a gap in the row. This behaviour is advantageous as it can be exploited to detect gaps in the rows if necessary by adjusting the value of $\theta_{\mathcal{J}}$.

The estimation of the state vector is less accurate during headland following compared to row following. This is due to the relative lack of information in the headland. When the robot is in the headland only a small part of the rows is visible making the estimation of the orientation of the robot heading with respect to the rows less precise. This, however, did not lead to any mistakes of the robot and it was rectified automatically as the robot turned into the new row.

The heading and lateral deviation parameters are generally unimodal and could well be described by a Gaussian distribution. Therefore we could have used a Kalman Filter as well for these parameters. However, the two end of row parameters can show a multimodal distribution, e.g. due to missing plants in the rows. Since these parameters drive the determination of the end of the row, they are very critical, and there should also be a possibility to recover the correct estimate after a gap of plants. Since the Particle Filter allows for such multimodality of the parameters, it is to be preferred over e.g. a Kalman Filter.

The assumption that the rows have equal width is based on the observation that the width of the two rows are highly correlated and adding a separate variable for the width of each row mainly adds to the complexity of the model. Incorrect estimates of row width and row distance will result in a different number of foreground and background pixels in the two regions (in-row/out-row). As long as the fraction in the inrow and outrow region are substantially different, the Bernoulli likelihood works well, even if Θ is rather far from the optimum for these two regions. A smaller Θ for in-row generally leads to an increased estimate of row width since more background pixels will then be included in the in-row region. Thus the row width parameter has a compensating effect for an incorrect setting of the fraction Θ . Apart from substantial wrong settings of Θ for in-row and out-row, the row width parameters can well be estimated, as is demonstrated by the experiments with plants of different ages where the row width varies significantly.

The downsizing factor of the image is a trade-off between the computa-

tional constraints and the accuracy of the algorithm. It was determined empirically to ensure that the robot could process an image every 5-10 cm, which resulted in a speed of 0.7 ms⁻¹ and an image processing frequency of 10 Hz. The downsizing factor has an influence on the attenuation factor for reducing the peakedness of the likelihood model. The peakedness of the likelihood model is an artefact of the assumption that each pixel is an independent measurement although it is clear that the adjacent pixels are correlated. Taking the T th root overrides the independence assumption by grouping T pixels as a single measurement. A more rigorous approach would be to relax the independence assumption and explicitly model the correlation between the pixels (e.g. using a Markov Random Fields). This will also provide a more realistic representation of the plant structure.

The camera is set to auto exposure and auto white balancing. This takes care of variation of the green in different illumination situations. Even if the corn is rather bleak or yellow, it still contains a considerable amount of green compared to the background. The formula for excessive green is motivated based on the color contrast between plant and soil. It is well known in agricultural literature that excessive green is a rather invariant measure for distinguishing between plant and soil and it is used in many applications (Golzarian et al., 2012).

While developing the robotic system, we have used and tested numerous approaches, starting with a Hough Transform and Kalman Filter. Occasionally the Hough transform yielded wrong line estimates, which made the robot behave unreliably. Also the bimodality of the end of row parameter forced us to look for alternatives for the standard Kalman Filter / Hough Transform. Using a Particle Filter instead of a Hough Transform improved the end of row detection, but still caused problems with the Hough Transform. Switching to this method of image generation from a particle state yielded a reliable robotic navigation system.

Constructing a model image from a state vector to directly define the likelihood of the particles is a novel way to circumvent the difficulties and uncertainties associated with any feature extraction process. While it is preferable in environments with sparse features with simple geometric representation, it can be challenging in a complex environment with a high degree of uncertainty from many sources of noise. We believe this approach is clearly advantageous to feature extraction as it circumvents the errors of the feature extraction process. Extending the image-based measurement model to more complex situations may be a challenging field of research.

4.5 Conclusions

In this paper we presented a robust navigation algorithm for navigating a robot in an a semi-structured agricultural environment based on a particle filter. We showed that the particle filter provides an accurate estimates of the robot-environment state accounting for the uncertainty inherent in the environment. By using an image-based likelihood model we mitigate the effects of uncertainty of the feature extraction process accompanying feature-based likelihood models. The new likelihood model is also generic and can be extended to include multiple regions in the image. Peakedness of the likelihood may result from this image-based approach, but can be circumvented by using an attenuation factor based on effective sample size. We demonstrated the robustness of the algorithm through experiments in several fields with different row patterns, varying plant sizes and diverse lighting conditions. To date we have logged over 5 km of successful test runs in which the robot traversed the specified path staying along the middle of the corridor when travelling between the rows, accurately detecting the end of the rows and traversing the headland.

Chapter 5

Laser Range Finder Model for Autonomous Navigation of a Robot in a Maize Field Using Particle Filter ¹

Abstract

Autonomous navigation of field robots in an agricultural environment is a difficult task due to the inherent uncertainty in the environment. Almost all the existing systems use one or more cameras and apply computer vision methods to extract guidance direction. Vision based methods are sensitive to ambient lighting conditions. This is a major disadvantage in an outdoor environment like an agricultural field. The current study presents a novel probabilistic sensor model for a 2D range finder (LIDAR) from first principles. Using this sensor model we developed a particle filter based navigation algorithm (PF) for autonomous navigation of a robot in a maize field. The algorithm was tested in various field conditions with varying plant sizes, different row patterns and at several scanning frequencies. Testing showed that the Root Mean Squared Error of the robot heading and lateral deviation were equal to 2.4 degrees and 0.04 m, respectively. We concluded that the performance of the proposed navigation method is robust in a semi-structured agricultural environment.

5.1 Introduction

Precision agricultural takes the variation within the field into account, by observing and responding to this variation. It is considered vital for sustainable

¹submitted for publication

farming. Precision agriculture can be labour-intensive (Edan et al., 2009) therefore there is great need for automation of various agricultural tasks like crop scouting, weed control, harvesting, tilling. In this vein, robotic solutions have been applied in various agricultural domains.

A basic component of automation in agriculture is autonomous navigation. Earliest navigation systems in agricultural domain used a camera as the sensor and were based on computer vision methods (Gerrish and Surbrook, 1984; Reid and Searcy, 1987). They were perhaps the most popular in agricultural robotics due to the availability of low cost cameras and a plethora of computer vision techniques that could be readily applied. For example, several methods based on the Hough transform have been developed for row following (Hague and Tillett, 1996; Marchant and Brivot, 1995). Southall et al., (Southall et al., 2002) developed a method for navigating a cabbage field in which plants were cultivated in a grid like pattern. They use the knowledge of the environment to build a grid-based model of the local environment in the camera view to obtain the guidance information. There are also stereo based methods which try to extract the depth information for robust navigation (Kise et al., 2005). More recent vision based methods include (van Evert et al., 2011; Bergerman et al., 2012).

Vision based methods are sensitive to lighting conditions and atmospheric effects. Due to the large variation in ambient light in an outdoor environment like an agricultural field, most systems need frequent calibration to the specific operating conditions. Alternative methods to overcome these problems included those based on GPS technology (Heidman et al., 2002; Slaughter et al., 2008; Stoll and Dieter Kutzbach, 2000). But GPS technology has several critical drawbacks like lack of accuracy required in precision agriculture, interruptions in the signal or alterations in the environment which are not in the map yet, but which need to be taken into account. This may lead to navigation failure.

Laser range finder(LIDAR) technology does not suffer from the effects of ambient lighting conditions and thus can be more reliable in an agricultural environment. Also the viewing range can be larger than with a camera. Despite these advantages there is not much focus on LIDAR based navigation in agricultural mainly due to its high costs. Reducing costs in recent years, however, have sparked interest in this field. Barawid et al., (Barawid Jr. et al., 2007) developed a real-time guidance system for navigating an autonomous vehicle in an orchard based on LIDAR. Hough Transform is used to extract plant rows for navigating the vehicle. They report that the method is restricted to straight line recognition and thus will have difficulty in curved rows. Another disadvantage of the method is, if the Hough transform fails to extract the correct plant rows, then the vehicle will loose track. LIDAR has also been used for obstacle detection and avoidance during navigation as in the case

of (Subramanian et al., 2006). More recently (Weiss and Biber, 2011) have developed a 3D LIDAR based navigation method where they use a statistical model for detection of the plant rows. The LIDAR acquires a 3D point cloud. The data is processed to remove the points corresponding to the ground. Then a statistical model identifies clusters of points that represents the plants. While the results are promising the method will not be easily scalable to other plants because the statistical model is specific to the maize plants. The statistical model depends on the cluster of 3D points which in turn depends on the shape and size of the plants. Moreover the system is specifically designed for plant phenotyping and thus the system design imposes restrictions on the operating conditions like plant size, age and vehicle speed, of the robot.

One of the main shortcomings in the aforementioned methods is the lack of robustness to the uncertainties in the environment. Agricultural environment is dynamic and non-deterministic with several sources of uncertainty. For instance, there is noise due to uneven terrain or varying shapes, sizes and colour of the plants or inconsistency in the environment structure. A robot operating in such an environment will suffer from wheel-slippage, sensor noise and is further compounded by controller and actuator noise. A navigation method capable of managing multiple sources of variation is challenging. Probabilistic navigation methods proposed by (Thrun et al., 2005) are most promising. They proposed a 2D LIDAR model that characterizes different types of noise in the environment. The sensor model is used within a particle filter for autonomous navigation of the robot in an indoor environment or in an outdoor urban environment.

In this study, we adopt a probabilistic approach, similar to the one proposed in Thrun et al., (Thrun et al., 2005) for developing an algorithm for autonomous navigation of a robot equipped with LIDAR in a maize field. The navigation method uses a particle filter for localization of the robot in the environment by using LIDAR data. Novelty of our method is two fold. The first is a new probabilistic model for LIDAR data obtained during row following of a robot in a maize field. The second is the particle filter based navigation method developed using the LIDAR model. To our knowledge it is the first (probabilistic) LIDAR model developed for robot navigation in a semi-structured environment like a maize field.

The paper is arranged as follows. Section 5.2 describes materials and methods along with the details of the sensor model. In Section 5.3 we demonstrate the performance and robustness of the new navigation algorithm and in Section 5.4 we point out limitations and extensions of the method and the scope for future research.

5.2 Materials and Methods

5.2.1 Maize Field

The robot navigates in a field that consists of rows of maize plants with a well defined headland. These rows may be either straight or curved. Additionally, there may be gaps within the rows. In general, the rows are approximately 0.75 m apart, being the distance at which the seeding took place.

5.2.2 Robot

The robot employed in this study consists of a chassis with three wheels, with overall dimensions 0.8 m $\times$ 0.45 m $\times$ 0.3 m (Figure 5.1). It has an actuated front wheel as the steering wheel that is affected by commands from a control program by means of a CAN-bus and two rear wheels that do not have the ability to steer. All wheel units are equipped with incremental encoders to measure the rotational speed. In addition, the front wheel unit is equipped with an angle sensor to measure the steering angle. The driving speed of each wheel depends upon the target speed of the control point, the location of the wheel with respect to the control point and the turning radius. An electronic box between the rear wheels houses a mini-ITX computer with a 2.4 GHz Intel Core2 Duo processor running Windows XP operating system. The robot is controlled by a custom C# software which uses OpenCV library for image processing. Energy to the computer and the wheel units is provided by three 12 V NiMH racing packs: 1 for the front wheel unit, one for both rear wheel units, and one for the PC.

5.2.3 Laser Range finder (LIDAR)

The robot is equipped with a LIDAR (LMS-111, Sick AG, Waldkirch, Germany) in the front at a height of 15 cm, through which it senses the world. The LIDAR operates based on time-of-flight (TOF) principle. It emits pulsed laser beams using a laser diode. If a laser pulse is incident on an object, it is reflected. The reflection is detected using a photo diode. The distance to the object is calculated from the propagation time that the light requires from emission to reception of the reflection at the sensor. The emitted laser beams are deflected using a mirror at an angular resolution of 0.5 degrees and scan the surroundings in a circular manner with a maximum field of view of 270 degrees. The maximum range of the LIDAR is 20 m. The maximum scanning frequency of the LIDAR is 50 Hz.

Figure 5.2a shows the top view of the mount. The longitudinal axis of the robot is aligned with the axis of the LIDAR. By convention, the starting angle of the scan is -135 degrees and end angle of the scan is 135 degrees,

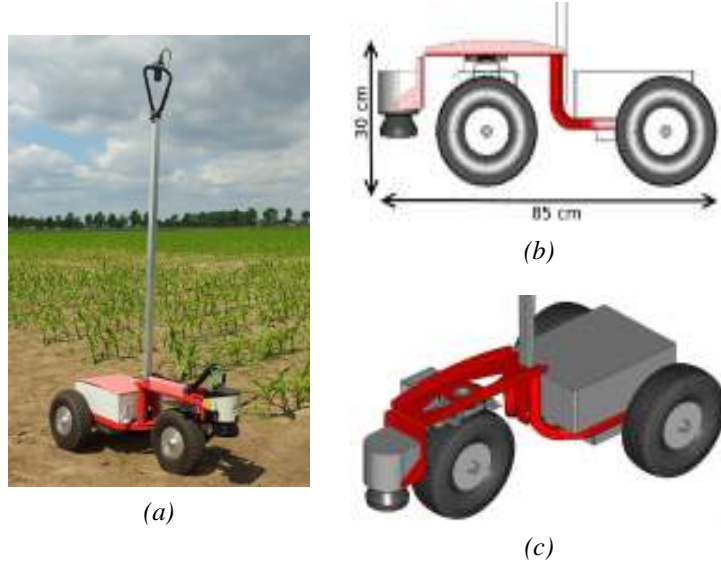


Figure 5.1: (a) The robot in a field. (b) The profile view of the robot depicting the length and height of the robot drawn to scale. (c) Perspective view of the robot showing the scanner in the front and the electronics box between the rear wheels.

depicted by Points A and C, respectively, in Figure 5.2a. A scan at any given time t consists of 541 observations $Z_t = (z^{(1)}, z^{(2)}, \dots, z^{(541)})$ corresponding to the angles $\Phi = (\phi^{(1)}, \phi^{(2)}, \dots, \phi^{(541)}) = (-135, -134.5, \dots, 135)$ where $z^{(j)}$ is the range, that is distance to a hit of an object (plant leaves or stem) measured by beam j . Figure 5.2b shows an example scan when the robot is between the rows. The data points $(\phi^{(j)}, z^{(j)})$ are represented in Cartesian coordinates for illustrative purpose. The blue circles indicates the position of the hit object with respect to the LIDAR represented in red circle.

5.2.4 Local World

A rectangular area around the robot is defined as the local world with the robot at the centre. If the robot is between the rows, the local world is approximated by two parallel rows of plants, one on either side of the robot. The rows have finite width and are a finite distance apart. It is assumed that the row ends are usually not in view as shown in Figure 5.3a. When the robot enters the headland, the end of rows are in the field of view and the geometry is modelled as in Figure 5.3b. The geometry of the local world is characterized by four parameters namely row width (rw), row distance (rd), end of left row (el) and end of right row (er), all measured in meters. The central line half-way between the rows forms the reference axis with respect to which the robot

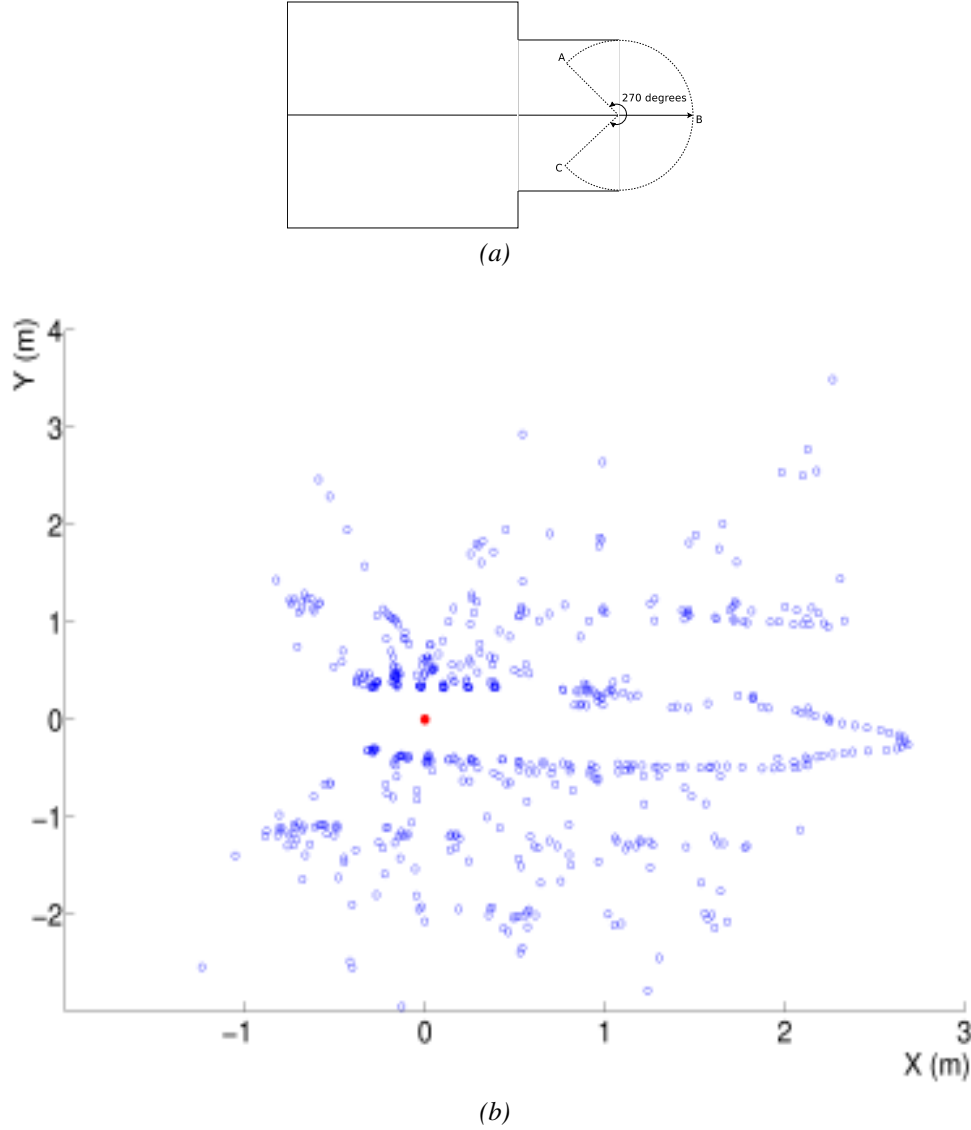


Figure 5.2: The figure on the top shows the top view of the mounted system along with the angular convention. The bottom figure shows an example scan in Cartesian coordinate system. The blue circles indicates the position of the plants with respect to the scanner (red circle).

position is determined. The robot is characterized by its main axis along the direction of travel and a central control point half way between the wheels. The position of the robot in the local world is given by robot heading (h) and lateral deviation (l). The robot heading is the angle between the main axis and the reference axis measured in degrees. Lateral deviation is the signed distance between the robot's control point and the reference axis. Jointly, the parameters represent the robot-field state vector $X_t = (h, l, rw, rd, el, er)$ that

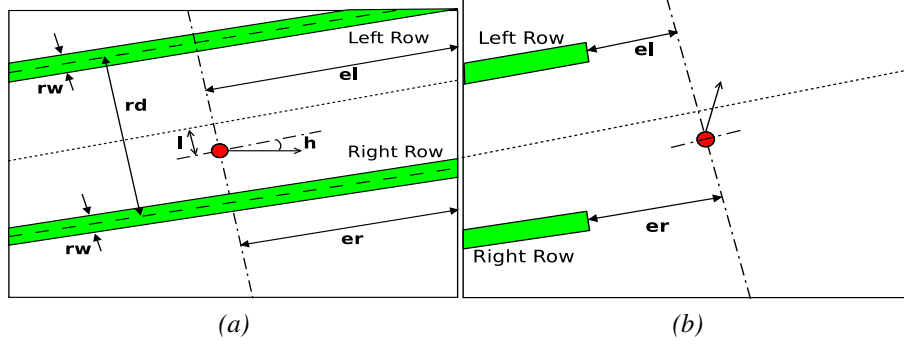


Figure 5.3: The local world of the robot when it is between the rows (left) and when it is within the headland (right). The red circle with the arrow represents the robot, with the circle representing the control point of the robot and the arrow its heading.

characterizes the system at a given time t . Successful navigation of the robot requires accurate estimation of the state vector at each time step.

5.2.5 Particle Filter

The robot-field state vector, denoted by $X_t = (h, l, rw, rd, el, er)$ characterizes the state of the robot in the field at any given time. Robot navigation can be considered as a dynamical system; the state changes at every time step and its values are uncertain due to different sources of noise in the environment. To deal with the uncertainty we represent the state of a system at any given time as a probability distribution $P(X_t|Z_{1:t}, U_{1:t})$ where $Z_{1:t}$ is the set of measurements made by the robot up to time t and $U_{1:t}$ is the set of controls applied to the robot to affect the state evolution up to time t . This distribution (also called the posterior distribution) has to be inferred at each time step. Inference of the posterior distribution is carried out by means of a particle filter algorithm. The key idea of particle filters is to represent the posterior distribution by a set of random samples called particles. These particles are recursively updated as a new measurement Z_t is acquired. The algorithm consists of two steps: prediction and update. In the prediction step the new values of the particles are calculated based on the current value and the motion model of the robot. It is discussed in detail in the next section. In the update step the predicted values are evaluated for their consistency with the measurement Z_t and importance weight assigned to them. Subsequently, the particles are re-sampled according to their (normalized) importance weights to yield the posterior distribution. Formally, it is given by

$$p(X_{1:t}|Z_{1:t}, U_{1:t}) = \frac{p(Z_t|X_t)p(X_t|X_{t-1}, U_t)}{p(Z_t|Z_{1:t-1})} p(X_{1:t-1}|Z_{1:t-1}, U_{1:t-1}) \quad (5.1)$$

where, $p(Z_t|X_t)$ represents the update step and is given by the measurement model (also called the likelihood model), $p(X_t|X_{t-1}, U_t)$ represents the prediction step and is given by the motion model, $p(Z_t|Z_{1:t-1})$ is the normalizing constant, and $p(X_{1:t-1}|Z_{1:t-1}, U_{1:t-1})$ is the posterior distribution at previous time step $t - 1$. Details of the measurement model will be discussed below. The specific form of equation (5.1) is indicative of the recursive nature of the particle filter where the posterior at the previous time step is updated by multiplying it with the motion model and the measurement model obtained from the current time step.

5.2.6 Motion Model

The motion model describes the changes in the state vector at consecutive time steps. When the robot is between the rows, it follows the path along the centre of the plant rows. We assume that the distribution of the initial state (X_0 at $t = 0$) of the robot is known, and that the gyroscope and wheel encoders on the robot provide the control information $U_t = (dx, dh)$ where dx is the displacement of the robot along its heading and dh is the turning angle of the robot. Now, the motion model is given by

$$\begin{aligned} h_t &= h_{t-1} + dh + \varepsilon_h \\ l_t &= l_{t-1} + dx \sin(h_{t-1} + dh) + \varepsilon_l \\ rd_t &= rd_{t-1} + \varepsilon_{rd} \\ el_t &= el_{t-1} - dx \cos(h_{t-1} + dh) + \varepsilon_{el} \\ er_t &= er_{t-1} - dx \cos(h_{t-1} + dh) + \varepsilon_{er} \end{aligned} \quad (5.2)$$

where $\varepsilon_h, \varepsilon_l, \varepsilon_{rd}, \varepsilon_{el}, \varepsilon_{er}$ are independent Gaussian noise applied to the corresponding state variables. It is worth noting that we treat the width of plant rows as a constant. This is because the scanner data does not provide any information about the width of the rows; it only returns the distance to the first ‘obstacle’ that the laser ray hits.

A complication arises because the end of the rows is frequently not in view of the LIDAR. According to the motion model, the values of el and er are constantly decreased. When the end of row is not in robot’s view, the el and er values should not be decreased. This situation is dealt with by re-initializing el and er in a fraction of particles at regular intervals.

5.2.7 Measurement Model

The perceptive field of the robot is modelled by a rectangular area with the LIDAR at the centre of the rectangle (the origin). When the robot is between the rows, the perceptive field is divided into five regions, named $R_1 - R_5$ based

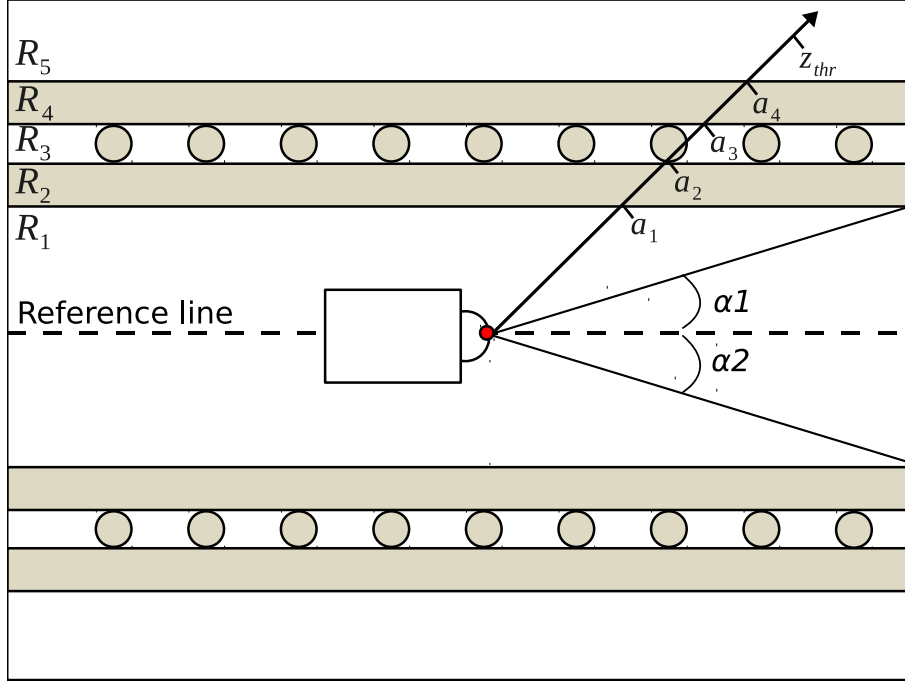


Figure 5.4: Illustration of the perceptive field of the LIDAR. R_1 represents the soil region depicted by the blank area, R_2 represents the plant leaves depicted by light grey rectangles and R_3 represents plant stems depicted by series of circles. Region R_4 is the same as R_2 and R_5 is the same as R_1 . The hatched area represents laser beams with angles in the range $[\alpha_1, \alpha_2]$ which are always in R_1 . The probability of a laser beam of hitting a plant depends on the beam angle ϕ and the resulting intervals $(a_{i-1}, a_i]$. z_{thr} is the specified threshold range.

on the density of the plant material (Figure 5.4). R_1 represents the soil region over which the robot travels and has the least density, R_2 and R_4 represent the region with foliage with intermediate density, R_3 represents the region with plant stems with highest density and R_5 represents the (soil) region beyond R_4 . Regions $R_2 - R_4$ together constitute a plant row of width r_w with, in region R_3 , stems along the centre line equidistant from each other at q m. The stems are assumed to be cylindrical with diameter $2r$ m.

The LIDAR makes observations by means of 541 laser beams corresponding to the 541 angles $\Phi = (-135, -134.5, -134, \dots, 135)$. An observation is a point in the perspective field characterized by the polar coordinates (ϕ, z) where z is the range and ϕ is the angle of observation. Due to the scanning mechanism, the angle of observations is always fixed but the range varies depending on how far an object is hit (foliage or stem). Thus z is a random variable which can take value in any of the five intervals $(0, a_1]$, $(a_1, a_2]$, $(a_2, a_3]$, $(a_3, a_4]$, and (a_4, z_{thr}) corresponding to the five regions R_1 , R_2 , R_3 , R_4 and R_5 ,

respectively, where z_{thr} is a specified threshold range value, or z falls in the category $z > z_{thr}$. The point a_i indicates the point of crossover of the laser beam from region R_i to region R_{i+1} . The intervals are different for different observation angle ϕ and thus are different for each beam.

Due to the difference in the density of plant material in each region, the probability of a laser beam hitting a plant material is also different. The region R_1 forms the robot path and thus we do not expect any objects. There can, however, be unexpected objects like an occasional overhanging leaf. The likelihood of sensing such unexpected (random) objects decreases with range and thus the probability of an observation in such situations can be described mathematically by an exponential distribution (Thrun et al., 2005). Region R_2 consists of a random configuration of plant leaves. As in the case of R_1 , the probability of an observation can be described by an exponential distribution but with a higher rate parameter than in region R_1 , because the density of the objects in R_2 is higher than in R_1 . From the assumption that plant stems in region R_3 are cylindrical, each with equal diameter $2r$ and at a fixed distance q from one other, we derived the probability of a hit which we converted to uniform density. The details of this derivation is included in Appendix 1. Region R_4 is, like R_2 , a foliage region and thus has the same exponential distribution as R_2 and region R_5 has the same density as R_1 .

It is assumed that each observation in $Z_t = (z^{(1)}, z^{(2)}, \dots, z^{(541)})$ is independent. Thus, the probability density of the measurement Z_t is the product of the probability densities of the individual observation and is given by

$$P(Z_t) = \prod_{j=1}^{541} P(z^{(j)}) \quad (5.3)$$

where $P(z^{(j)})$ is the probability density of observation $z^{(j)}$. It is the probability density of a beam j hitting a plant in the perspective field. The probability of no-hit is the probability that the beam passes through a region without hitting anything. The probability of hit is different in each region due to the difference in the plant density as well as due to the spatial arrangement of the regions. For instance, the probability of hit in R_2 not only depends on the plant density in that region but also depends on the fact that the beam does not hit any plant in region R_1 i.e., probability of no-hit of region R_1 . In general, the probability of a hit in region R_i depends on the plant density of region R_i as well as the probability of no-hit of all the preceding regions through which the beam travels. Formally, it is described as follows. If an observation $x = z^{(j)}$ is in R_1 then the probability density of a hit at range x is given by

$$P(x) = \lambda_1 e^{-\lambda_1 x} \quad (x \in (0, a_1]) \quad (5.4)$$

where λ_1 is the rate parameter of the exponential distribution. If $x = z^{(j)}$ is in

R_2 , the probability density of a hit at range x is given by

$$\begin{aligned} P(x) &= P(\text{no-hit in } R_1) \cdot P(x|\text{no-hit in } R_1) \\ &= \psi_1 \cdot \lambda_2 e^{-\lambda_2(x-a_1)} \quad (x \in (a_1, a_2]) \end{aligned} \quad (5.5)$$

where $\psi_1 = e^{-\lambda_1 a_1}$ is the probability of no-hit in R_1 and λ_2 is the rate parameter of the exponential distribution in R_2 . If the observation $x = z^{(j)}$ is in R_3 , the probability density of a hit at range x is

$$\begin{aligned} P(x) &= P(\text{no-hit in } R_1) \cdot P(\text{no-hit in } R_2 | \text{no-hit in } R_1) \cdot P(x | \text{no-hit in } R_1, \text{no-hit in } R_2) \\ &= \psi_1 \cdot \psi_2 \cdot \left(\frac{p^*}{(a_3 - a_2)} + (1 - p^*) e^{-\lambda_2(x-a_2)} \right) \quad (x \in (a_2, a_3]) \end{aligned} \quad (5.6)$$

where $p^* = \min\left(\frac{2r}{q \sin(\phi+h)}, 1\right)$, $\psi_2 = e^{-\lambda_2(a_2-a_1)}$ is the probability of no-hit in R_2 and q and r are the parameters of the row of stems. The derivation of p^* is given in Appendix 1. Further, if the observation is in R_4 the probability density of a hit at range x is

$$P(x) = \psi_1 \cdot \psi_2 \cdot \psi_3 \cdot \lambda_2 e^{-\lambda_2(x-a_3)} \quad (x \in (a_3, a_4]) \quad (5.7)$$

where $\psi_3 = (1 - p^*) e^{-\lambda_2(a_3-a_2)}$ is the probability of no-hit in R_3 (see Appendix 1). If the beam goes beyond the four regions without hitting anything then x is in the interval (a_4, z_{thr}) . In this case the probability density of a hit at range x is given by

$$P(x) = \psi_1 \cdot \psi_2 \cdot \psi_3 \cdot \psi_4 \cdot \lambda_1 e^{-\lambda_1(x-a_4)} \quad (x \in (a_4, z_{thr})) \quad (5.8)$$

where $\psi_4 = e^{-\lambda_2(a_3-a_4)}$ is the probability of no-hit in R_4 and z_{thr} is a specified threshold value. Finally, the observation may be greater than or equal to the threshold value z_{thr} . For such observations, $P(z \geq z_{thr})$ is modelled as one minus the no-hit probability in the regions up to z_{thr} .

In order to compute the probability of hit of a beam j it is necessary to compute its corresponding intervals $a_1, \dots, a_4$. The variable a_i not only depends on the beam angle $\phi^{(j)}$ but also on the components of the state vector X_t as shown in the Figure 5.5. This relationship enables the determination of the particle weights in the particle filter.

Figure 5.6 shows the probability density profile of a laser beam at $\phi = 45$ degrees for two sets of rate parameters λ_1 and λ_2 . The assumptions for creating the plot included that the plant rows are 0.2 m wide at a distance of 0.75 m. The robot is located at the centre of the two rows with heading at 0 degrees, such that, the end of the rows are in sight and at a distance of 0.8 m. Thus, the values of the components of the state vector are $h = 0, l = 0, rw = 0.2, rd = 0.75, e_l = 0.8$, and $e_r = 0.8$. The points a_1, a_2, a_3, a_4 and z_{max} on the graphs indicate the boundaries of the different regions in the perspective field.

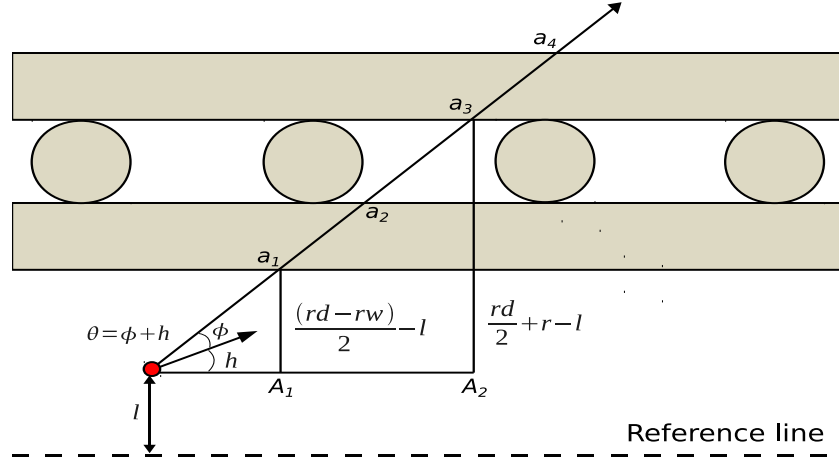


Figure 5.5: Illustration of the relationship between the intervals of a beam and the state vector.

5.2.8 Attenuation Factor

The observations obtained from adjacent beams are not independent. The observations may be correlated because they belong to the same plant. Correlation may also arise due to multiple reflections of a beam from several plants. In the aforementioned measurement model, however, every observation is regarded as independent. As a result the likelihood is highly peaked. That is, likelihoods of two similar particles differ by orders of magnitude. This problem is addressed by expressing the dependency between the beams by means of the attenuation factor v (Thrun et al., 2005) where v measurements are considered as a single unit, so reducing the effective number of measurements by a factor v . This is implemented by raising the likelihood to the power $1/v$.

The appropriate value of v is determined using the effective sample size (ESS) of the PF (Doucet and Johansen, 2009). Figure 5.7 shows the plot of ESS vs v averaged over the iterations of a PF run. At small values of v , the ESS is close to 1% indicating most of the predicted particles are ‘far’ from the measurement and thus have negligible weights. But as the value of v increases so does the ESS indicating that the particle weights are more evenly distributed. We selected the value of v such that 70% of the particles are retained after the resample step, resulting in $v = 160$.

5.2.9 Experimental Data

The experimental data consists of several sequences of scans taken by the robot as it navigates through the field. The sequences are from different maize

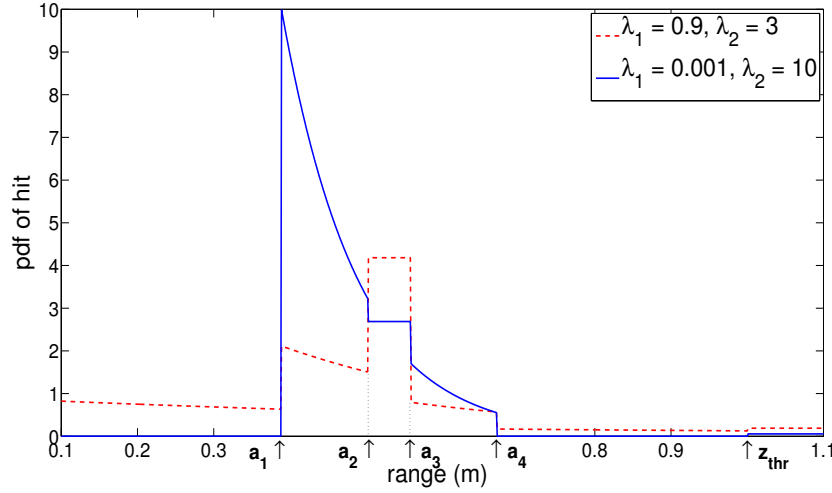


Figure 5.6: Probability density of hit of a laser beam at an angle $\phi = 45$ degrees. The values of the state vector components were $h = 0, l = 0, rw = 0.2, rd = 0.75, e_l = 0.8, e_r = 0.8$.

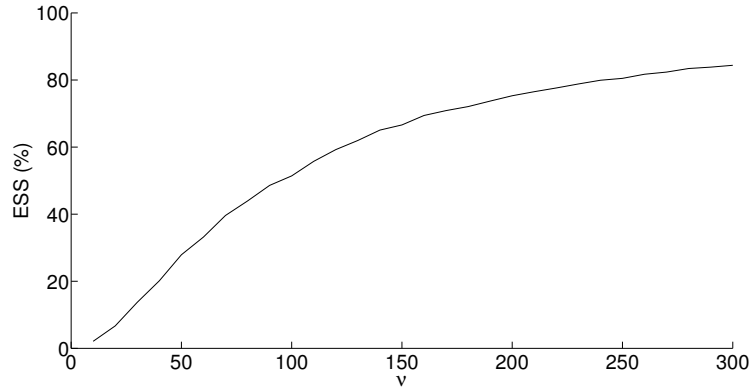


Figure 5.7: Figure shows the graph of ESS vs ν , where ν is the attenuation factor by which the likelihood is attenuated by raising it to the power $1/\nu$. As ν increases, the ESS also increases indicating a smoother likelihood function.

fields with different plant size, row structure (straight rows, curved rows, gaps within the rows). The number of scans in each sequence depends on the length of the robot run where the LIDAR records data at 10 Hz. Although the LIDAR can record data at a maximum rate of 50 Hz, it was limited to 10 Hz for computational reasons. For every scan in the sequence, the ground truth for the most important state variables, heading (h_r), lateral deviation (l_r), end of left row (el_r) and end of right row (er_r) was established by hand annotation of a corresponding image captured by a downward looking camera mounted on the robot at a height of 1.65 m. Figure 5.8 shows three example images

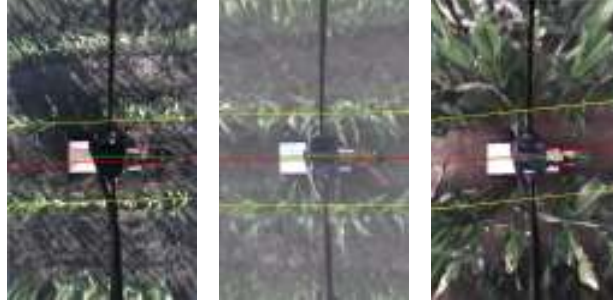


Figure 5.8: Three example images showing how the ground truth was determined. The three images correspond to three different field conditions with small plants (left), medium plants (middle) and large plants (right). In each picture, the yellow lines indicate the position and orientation of the plant rows, the green line indicates the robot and the red line is the reference line. The heading and lateral deviation of the robot is determined with respect to the reference line.

illustrating the process of establishing the ground truth. The three images correspond to three different experimental conditions where the plants are in different stages of development. These ground truth values were used to evaluate the performance of the PF. It should be noted that in case of large plants (Figure 5.8c) there is some ambiguity in determining the exact position and orientation of the rows.

The experimental data is divided into two sets namely test and validation set. The test set is used for calibrating the particle filter. It consists of a single sequence of 181 scans obtained from a field with medium sized plants (middle picture in Figure 5.8) and field conditions in which the robot is expected to operate. The validation set, on the other hand is used to validate the robustness of the PF. It consists of five sequences of scans obtained from different fields. These five sequences together contain 432 scans.

5.3 Experiments and Results

Experiments consisted of two different phases namely, testing and validation. In the testing phase the PF algorithm and its parameters were calibrated with the test data. Subsequently, the calibrated algorithm is validated using validation data. In all the experiments, the PF was initialized with 256 particles. This number is empirically determined based on real-time computational constraints – a balance between update frequency (10 Hz) and number of particles. At each time step of the algorithm, the weighted mean of the posterior distribution $\hat{X}_t$ is the PF estimate of the state X_t at time t . Thus, for every im-

Table 5.1: Mean $\pm$ standard deviation of the RMSE using validation data.

Parameter	RMSE
Heading (degrees)	2.40 ± 1.0
Lateral Deviation (m)	0.04 ± 0.02
Left end-of-row (m)	0.30 ± 0.10
Right end-of-row (m)	0.26 ± 0.10

age in the sequence, its PF estimate can be compared with the corresponding ground truth.

Calibration of the PF involved evaluating its performance on the test dataset for various combinations of rate parameters from the sets

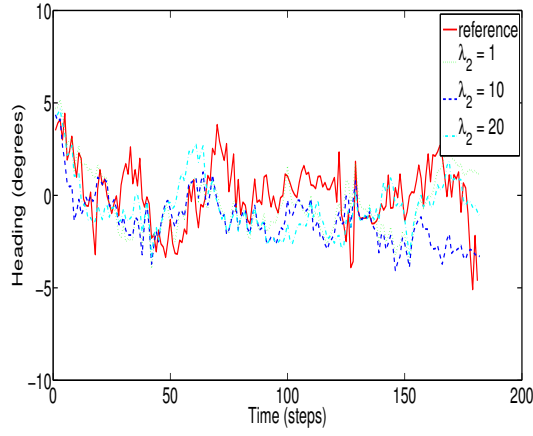
$\lambda_1 = \{0.001, 0.005, 0.001, 0.005, 0.01\}$ and $\lambda_2 = \{1, 5, 10, 15, 20, 25\}$ comparing it with the ground truth data. Figure 5.9 shows the result of the comparison for three different values of λ_2 in $\{1, 10, 20\}$ where $\lambda_1 = 0.005$. The five diagrams in the figure correspond to the five state variables. In this figure variables h , l , el and er are plotted along with their corresponding ground-truth data h_r , l_r , el_r and er_r , respectively, while rd had no reference. Note that $rw = 0.2$ m is assumed to be constant and hence not estimated by the PF.

The Figures 5.9a and 5.9b show that the PF estimate is close to the true values of h_r and l_r when $\lambda_2 = 10$ through out the entire robot run while in other two cases ($\lambda_2 = 1$ and $\lambda_2 = 20$) the PF estimate diverges from the ground truth at the end of the row. The detection and tracking of the end of the row is most accurate when $\lambda_2 = 10$. The end of row detection fails when $\lambda_2 = 1$ and there are many false detections when $\lambda_2 = 20$. We also observe (Figure 5.9c) that the estimation rd is sensitive to the values of λ_2 . The increase and decrease in the value of λ_2 leads to the increase and decrease of rd , respectively.

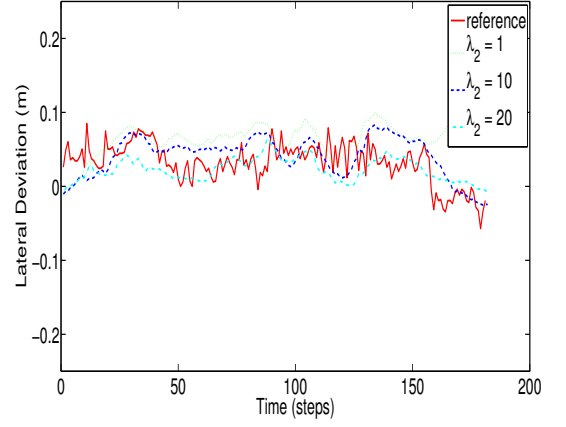
Based on the findings in the test phase, the calibrated PF used for the validation phase had the rate parameters $\lambda_1 = 0.005, \lambda_2 = 10$. The performance of the validated PF is shown in Table 5.1. It gives the mean and standard deviations of the Root Mean Squared Error (RMSE) of the PF estimate of the state variables computed using the five sequences. Note that the RMSE values for el and er is determined using only the final segments of the sequences where row ends are still in robot's view.

Other validation experiments included testing the PF at lower scanning frequency of 5 Hz. Figure 5.10 shows the comparison of the PF estimate with ground-truth data for heading and lateral deviation in two different field conditions. The results in the top and bottom row corresponds to tests done in the field with small (25 cm tall) and large (60 cm tall) plants, respectively.

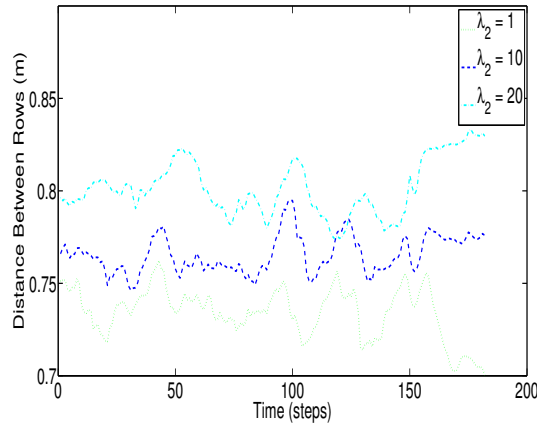
In general, extreme values of λ_2 (and λ_1) lead to erroneous estimates of the state vector components. For instance, when $\lambda_2 = 1$ the PF fails to detect the end of the rows (green dotted curve in Figures 5.9d and 5.9e). Similarly,



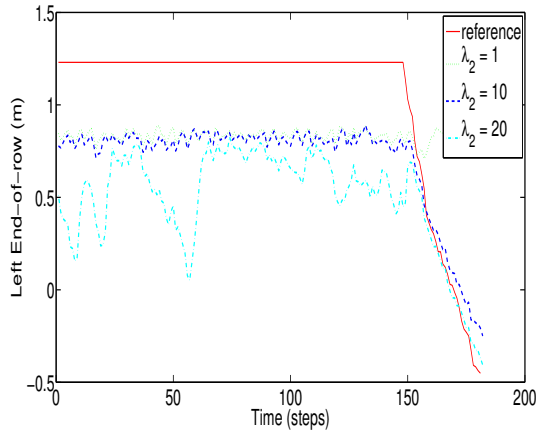
(a) Heading



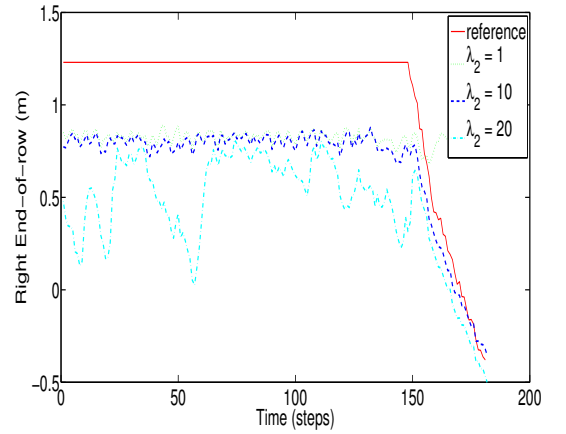
(b) Lateral Deviation



(c) Row Distance



(d) End of Left row



(e) End of Left row

Figure 5.9: The result of PF estimation of X_t of a robot run for three values of λ_2 . The dotted curve in green is the estimate when $\lambda_2 = 1$, the dashed curve in blue is the estimate when $\lambda_2 = 10$, and the dashed and dotted curve in cyan is the estimate when $\lambda_2 = 20$. The value of λ_1 was fixed to 0.001 in all three cases. The solid line in red indicates the reference data.

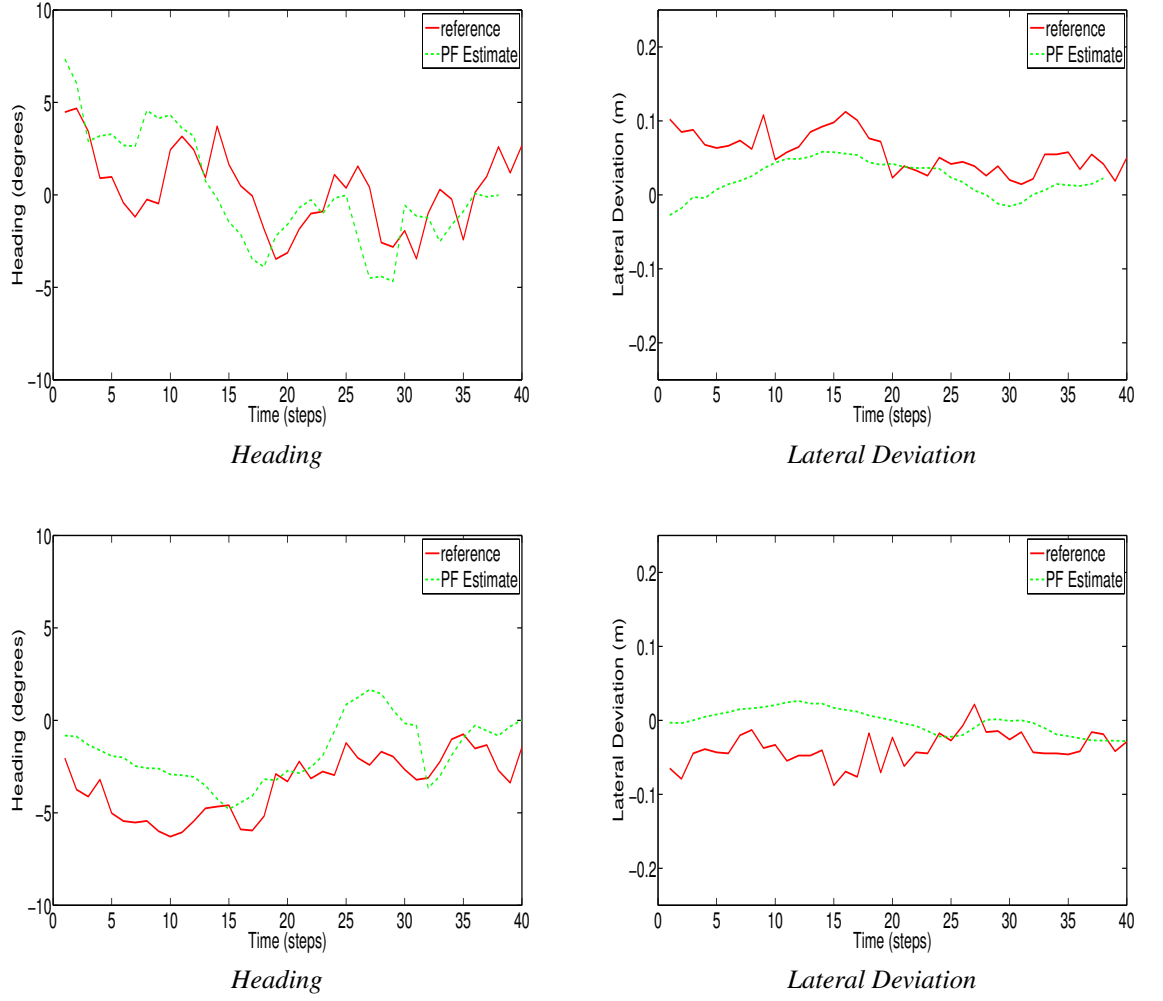


Figure 5.10: The result of PF estimation of X_t of a robot run when the scanning frequency is 5 Hz. The top row shows the results of the experiments in a maize field with small plants of 25 cm tall while the bottom row are the results from a field with large plants of size 60 cm.

when $\lambda_2 = 20$ there are many false detections due to small gaps from missing plants (cyan dotted and dashed curve in Figures 5.9d and 5.9e). In general, it was observed that the value of λ_2 is directly proportional to the ‘sensitivity’ of the PF to the row ends as well as to the gaps in the row. Increasing the value of λ_2 also results in wider row distance estimates.

Model Complexity

Modelling the details of the stem region R_3 increases the complexity of the measurement model. The trade-off between model complexity and algorithm accuracy was quantified by means of RMSE values of the PF estimate of h , l , e_l and e_r using the test data. The stem region R_3 can be excluded from the measurement model simply by setting $r = 0$. The experimental results are tabulated in Table 5.2. They suggest that the added complexity does not

Table 5.2: RMSE values using the test data.

Parameter Settings	RMSE (With R_3)				RMSE (Without R_3)			
	h	l	e_l	e_r	h	l	e_l	e_r
$\lambda_1 = 0.001, \lambda_2 = 1$	1.98	0.05	0.71	0.72	2.01	0.05	0.71	0.74
$\lambda_1 = 0.001, \lambda_2 = 5$	2	0.03	0.68	0.68	1.97	0.03	0.71	0.71
$\lambda_1 = 0.001, \lambda_2 = 10$	2.34	0.03	0.38	0.16	1.80	0.02	0.37	0.17
$\lambda_1 = 0.001, \lambda_2 = 15$	2.20	0.03	0.35	0.19	1.84	0.03	0.36	0.18
$\lambda_1 = 0.001, \lambda_2 = 20$	1.95	0.03	0.36	0.21	1.83	0.03	0.36	0.19
$\lambda_1 = 0.001, \lambda_2 = 25$	2.08	0.05	0.84	2.22	1.55	0.04	0.46	0.90

significantly improve the performance of the PF in a maize field. The implications of this is discussed in the next section.

5.4 Discussion

In this study we developed a novel sensor model for a range scanner for robot navigation in a maize field. The sensor model is in turn used to develop a particle filter based navigation method for estimating the robot-environment state. The particle filter is able to deal with the different kinds of uncertainties in the environment to provide a reliable estimate of the robot pose as it moves through the field.

There is a lower limit on the plant size in which the robot can operate. This depends upon the height of the LIDAR mounted on the robot. In the experiments, the scanner is mounted at a height of 15 cm from the ground and the plant size at which the algorithm gave reliable estimates was between 20-25 cm. Ideally, the plants should be 5-10 cm taller than LIDAR to satisfy the assumptions of the perspective field. Similarly, the algorithm does not perform very well when the plants are taller than 60 cm because of the dense leaves which extend across the middle of the rows and on to the main robot path. The resulting scanner data is unlike the one shown in Figure 5.2b and the perspective field cannot easily be divided into different regions based on the differences in the density of the plants. For instance, the foliage is dense and also covers R_1 . The large leaves of the plants obstruct the laser beams such that observations are clustered around the origin. Also, the scanner model cannot be used to reliably estimate the width of the plant rows. Due to the physics of the data acquisition process of the LIDAR as well as due to its positioning on the robot we can only observe the front of the object. This, however, does not affect the navigation of the robot as we have seen in the Section 5.3 where tests were carried out in fields with different plant sizes.

The process by which the ground truth is determined is based on the cam-

era image corresponding to a scan. The difference in the camera’s downward-looking and the LIDAR’s forward-looking perspectives results in larger RMSE values for the end of row components (e_r and e_l) than that was observed in field experiments. This is because, in the camera image, the end of a row is determined by the tip of leaf extending furthest into the headland which may not be accurate from a LIDAR perspective. The heading and lateral deviation components does not suffer from this perspective bias. The problem is further compounded by hardware errors which result in missing scans.

The experiments showed that the added complexity of the stem region (R_3) does not improve the accuracy of PF estimation in our situation. This may be explained by the fact that the density of the foliage is greater than the density of the stems for maize plants. A closer examination of Figure 5.6 shows that, in our experimental situation, the foliage density is greater than the density of the stem region (the solid blue curve where $\lambda_2 = 10$). Here the exponential density of the R_2 dominates over the density of the stem region R_3 . This effect may not hold for other crops as indicated by the dashed red curve which corresponds to a situation where the density of the stems in R_3 is greater than the foliage density in R_2 . In conclusion, the inclusion/exclusion of a particular region depends on the crop in which the robot will operate and the parameters of the measurement model can be tuned as necessary.

The scanning rate impacts the performance of the algorithm. Low scan rate leads to erroneous PF estimates while a high scan rate improves the accuracy of the PF estimate. This, however, increases the computational burden and thus the trade off between the two is a design decision based on the application. As demonstrated in Section 5.3 scanner data acquired at a rate of 10 Hz is sufficient for robust estimation of the robot-environment state.

The LIDAR model we propose is closely related to the beam model developed by (Thrun et al., 2005). The beam model is a mixture of four densities where each density characterizes a type of noise typically encountered when using a LIDAR. In comparison, the different types of noise in the beam model is analogous to the different regions in our measurement model. Another difference between the two models is the way in which the hit probability of the beam is computed. In our measurement model, it depends on the spatial arrangement of the regions as well as the sequence in which the beam encounters these regions whereas in the beam model is a weighted average. A main advantage of our model is that it takes in to consideration the geometry of the perspective region and the angles of the laser beams unlike the beam model.

Robot navigation in a maize field using laser range finder is not a trivial task because range data are noisy. The noise may be due to incorrect observations due to multiple reflections, missing observations or sometimes even missing scans. A probabilistic sensor model is a good way to characterize these various types of sensor noise. As far as we know this is the first prob-

abilistic 2D LIDAR model for robot navigation in a maize field. The sensor model can be easily extended to included more regions within the perspective field if necessary. The regions can be of a new type with its own probability distribution. As a result the model can be applied in other fields.

5.5 Conclusions

Autonomous navigation of robots in an agricultural environment is a difficult task due to the inherent uncertainty in the environment. These uncertainties include varying plant size, different row patterns or uneven terrain conditions. This study focused on developing a probabilistic navigation method based on the particle filter (PF). Such a probabilistic approach mitigates the effects of uncertainties in the environment. This is demonstrated in the performance of the PF under various field conditions, where it provided accurate estimates of the robot-environment state in a dynamic and noisy environment. The probabilistic model of the LIDAR sensor incorporated in the PF turned out to be an effective way of dealing with sensor noise. Because of its probabilistic nature, the model could easily be incorporated into the PF. Using a LIDAR sensor for sensing the world also expanded the operating conditions of a robotic system in an agricultural field as, for example, it did not suffer from the effects of varying lighting conditions. In conclusion, we showed how the PF can be used for a robust robot navigation using a LIDAR within a semi-structured agricultural environment.

Appendix of Chapter 5

Probability of an Observation in R_3

The top view of a section of the stem region R_3 is shown in Figure 5.11. It consists of a series of cylindrical stems of diameter $2r$ with centres at a distance of q from one stem to the next. Consider a laser beam j at an angle ϕ . The beam will strike the stem S1 between points m_1 and m_2 when the LIDAR is between positions A and B , respectively, where m_1 is the first point of contact and m_2 is the last point of contact. The same beam will strike stem S2 when the LIDAR is between C and D . Assuming that the stems are uniformly random with respect to the position of the robot, the probability of a beam striking a stem is related to the area covered by the stems. In other words it is proportional to the ratio $\frac{2r+2d}{q}$. From trigonometry $d = r \left(\frac{1}{\sin(\theta)} - 1 \right)$ where $\theta = \phi + h$ and thus the ratio can be rewritten as $\frac{2r}{q \sin(\theta)}$. The probability

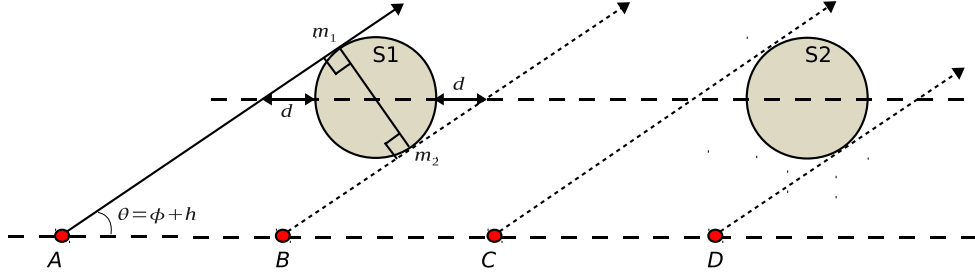


Figure 5.11: A section of the stem region. $S1$ and $S2$ represent two adjacent stems of diameter $2r$ and with centres at a distance q apart. The red circles indicate the position of the LIDAR. A laser beam j with angle ϕ strikes $S1$ between the points m_1 and m_2 . The line through A and m_1 is tangent to $S1$ and so is the line through B and m_2

of a stem hit is thus

$$p^* = \min\left(\frac{2r}{q \sin(\theta)}, 1\right)$$

If the beam does not hit a stem (probability $1 - p^*$), it may still hit foliage as in R_2 . Assuming a uniform density for a stem hit, the final pdf in region R_3 is

$$P(x) = \frac{p^*}{a_3 - a_2} + (1 - p^*)e^{-\lambda_2(x - a_2)}$$

Integrating this over the range $(a_2, a_3]$ gives the probability of hit with the resulting probability of no-hit in R_3 being $\psi_3 = (1 - p^*)e^{-\lambda_2(a_3 - a_2)}$.

Chapter 6

General Discussion

The main aim of this thesis was to explore the efficacy of probabilistic methods for agricultural robotics. In order to achieve the research goal, two applications were chosen namely automatic weed detection in a grassland and autonomous navigation of a robot in a maize field. This allowed us to structure the main goal into five objectives described in the introduction (Chapter 1) such that within each application two probabilistic methods pertinent to the domain could be tried and tested. In the preceding chapters these methods were described. Now, we discuss the achievements of each objective to put the research into perspective. First, we look at automatic detection of *Rumex obtusifolius* (Rumex) in a grassland followed by autonomous navigation of a robot in a maize field.

6.1 Automatic Detection of Weed

The main challenges in vision based weed detection methods are the non-rigid shapes of the weeds, their varying size, changing appearance at different growth stages and similarity with the crop plants. These ambiguities and overlapping characteristics between weeds and crop mean that it is very difficult to provide a consistent mathematical description of a weed based on their shape, size or colour. Texture is perhaps the best way to characterize the weeds because it is not as variable as the other parameters. As a result many texture based descriptors are being used in many weed detection methods. In Chapter 2 a critical analysis indicated that of several of the existing texture based methods such as local variance or Law's filter masks were not suitable for detecting Rumex in a grassland. The main conclusion of this chapter was that simple statistical summaries are insufficient to describe the uncertainty in natural images of weed or other such plants. Instead, features derived from GLCM which encode relationship between pixels, were more accurate and robust in differentiating between weed and grass. This probabilistic approach

also facilitated, through its parameters, the choice of the granularity of pixel interaction. For example, for discriminating Rumex from grass it was sufficient to look at interaction between adjacent pixels because of the contrast in the size of the grass blades and the leaves of Rumex.

The performance of the automatic segmentation algorithm using the optimal GLCM features (also referred to as GLCM algorithm) was measured by means of the confusion matrix. From the confusion matrix several performance metrics like accuracy, specificity and sensitivity can be derived which help us to examine the subtler aspects of the algorithm. Although it is desirable that a weed detection algorithm measures high on all metrics, this may not be possible sometimes. In such cases the performance of the algorithm can be tuned according to the requirement. For example, from a weed detection point of view, high specificity is relatively more desirable than high sensitivity because false detections lead to unnecessary drilling or spraying causing damage to the healthy crops which is costly. At the same time false negatives, like mistaking a weed for grass is not as costly because they can be controlled during subsequent runs.

Although the GLCM based segmentation algorithm provided an effective means of encoding spatial relationship between the pixels, it does not provide a means to encode prior knowledge about them. Prior knowledge is an important contextual constraint for problems in image analysis as it acts as a regularizing factor on the uncertainties in the image. For instance, two images of grass taken at different lighting conditions may exhibit different, and even contradictory, pixel relationships. A GLCM will only record these relationships but does not provide any means to specify the expected uncertainty in the interactions due the varying illumination conditions. As a result the two images might be classified to belong to different plants. To specify the structure of the uncertainty of a texture it is necessary to have a prior model. This understanding paved way to the segmentation method developed in Chapter 3 (also referred to as GMRF algorithm) where a Bayesian approach was adapted. The formulation of segmentation of Rumex from grass as a MAP-MRF optimization problem has several advantages over the existing vision-based weed control methods. From a technical point of view the MRF modelling is a means of specifying precisely the structure of uncertainty expected in the images. The optimization approach provides great flexibility where the solution is optimal to the context. That is the optimal value of the energy is a function of the data so, for example, the class of a texture in two or more images captured in different illumination conditions can both be correctly identified. Moreover, optimization is an effective way of incorporating both the local and global interactions between the pixels. More generally, the MAP-MRF framework provides an opportunity to codify the vast amount of existing knowledge about weeds in the form of prior models for exploita-

tion in computational methods. In particular, the mechanism of specifying prior(s), by means of clique potentials, in the MAP-MRF framework is relatively simple and the clique potentials can be based on shape, size, colour or other expert knowledge.

The feature selection methods used in Chapter 2 and 3 provide a systematic way to evaluate and determine the performance of a set of features. It not only provides a robust mechanism for evaluating texture features from different circumstances but also helps to determine the parameter values specific to the application. Such an automated process plays an important role in the vision based methods for weed control where many application-specific parameters lead to a large number of potential features that need to be evaluated quickly and robustly. The two chapters introduce two different kinds of feature selection methods namely, wrapper and filter method which are both valuable in their own right. A wrapper method is designed to select features that are application specific and are tightly coupled to the performance of the segmentation algorithm whereas, in a filter method, the purpose of selecting the best features is to establish the appropriate prior model that best characterizes the textures in the image. Thus the resulting features are generic in the sense that they are decoupled from the segmentation algorithm and the application. Both methods are valuable tools for agricultural robotic methods.

As both methods are based on texture, interference may occur from undesirable objects with texture that is similar to that of Rumex, such as patches of soil or dense pasture of wild grass. These problems, however, are limited as both GLCM and GMRF textures are highly sensitive to the spatial pattern of intensities that generate texture. The small range of the window sizes over which the GLCM features yield satisfactory detection results and the high sensitivity of GMRF features to homogeneity of the training patches are illustrative examples. In the event that the texture of an odd object is very similar to Rumex, then additional information can be used to discriminate between the two. For example patches of soil with a similar texture to Rumex can simply be handled, as soil has distinct colour properties. In a preprocessing step methods like excessive green transformation can then be used to discriminate Rumex from soil. Also, the GMRF algorithm provides strategies for eliminating odd objects such as multi-channel and multi-resolution MRFs.

Both GLCM and GMRF segmentation algorithms have several advantages that are not present in many other methods. For instance, both methods provide a simple and precise method of specifying the spatial context through the definition of the neighbourhood system. Further the flexibility in the definition of the neighbourhood systems means that the methods are applicable for textures with different orientation and scales. That is, the methods can be applied for detection of other weeds such as coltsfoot, common nettel, greater plantain, ground elder and ground-ivy. The most salient feature of GMRF

algorithm is its real-time operability where it can process about five images per second with an accuracy of 98%. None of the existing weed detection methods can achieve this high degree of performance. Such high speed and reliability is necessary to convince industrialists and farmers to adopt robotic weeding systems on a commercial scale.

These methods can also be used for other purposes such as plant phenotyping where the plant description is a joint probability distribution over phenotypic features yielding a precise description. Such precise description of the plants will in turn improve the performance of the automated systems. For example quality control systems for agro-products will be better equipped to identify malformed products, which will allow removal of poor quality phenotypes more effectively.

6.2 Autonomous Navigation in an Agricultural Field

Following the encouraging results of Chapter 3, Bayesian principles were used to develop a probabilistic navigation algorithm in Chapter 4 and 5. This process consisted of two phases. The first phase involved developing a vision-based navigation method using a particle filter and the development of a probabilistic camera model (Chapter 4). The second phase included extending the navigation method to work with a LIDAR instead of a camera and to develop a new probabilistic LIDAR model (Chapter 5).

The particle filter based navigation method described in Chapters 4 and 5 forms a general probabilistic framework for autonomous row following in a field. The generality of the method is due to the parametrization of the local world of the robot. That is, the components of the state vector such as robot heading, lateral deviation, row width, row distance, and the left and right row ends not only apply to maize fields but are also applicable to other fields in which the crops are planted in rows. Also, the idea of parametrizing the local world means that it can also be applied to fields with different geometry as long as the geometry can be suitably parametrized. The modular nature of the particle filter means that the local world model, motion model and the measurement model can be different depending on the geometry, the vehicle and the sensors, respectively. In Chapter 4, a camera model becomes the measurement model of the particle filter because the robot uses only a camera sensor for navigation and in Chapter 5, a LIDAR model becomes the measurement model. Henceforth the navigation methods in chapters 4 and 5 are referred to as vision-based and LIDAR-based methods, respectively.

Due to the geometric representation of the local world, the perceptual field of the robot can be reconstructed based on the state vector which is crucial for the development of probabilistic measurement models without relying on features extracted from the sensor data. For instance, in the vision-based method,

the model-image can be computed because of the parametrization and similarly in the LIDAR-based method it is possible to compute the probability of hit in different regions of the perceptual field. As we can now directly compare the predicted model image with the observed data, it is not required to extract features from sensor data. Thus the data processing required is relatively simple. In Chapter 4 the image processing step to obtain the measurement is trivial and in Chapter 5 there is no data processing at all. This makes this step hardly prone to errors, in comparison with other feature extraction methods. Thus all uncertainty can now be modelled by the particle filter, making our approach very robust. Another advantage of the parametrization is the robot centric coordinate system which makes the navigation methods invariant to the size and shape of the field.

The vision-based measurement method is characterized by two parameters: probability $\theta_{\mathcal{R}}$ of a green pixel belonging to in-row and the probability $\theta_{\mathcal{O}}$ of a green pixel belonging to the out-row. These parameters indicate the density of plant material in the rows and outside the rows. The particle filter is robust to a large range of these parameter values; thus it is applicable in many different types of fields, at different growth stages. Likewise, the LIDAR-based navigation method is also characterized by two parameters λ_1 and λ_2 . If necessary the parameter values can be calibrated to specific type of crops.

The positioning of the sensors on the robot improves the robustness of the navigation algorithm and reduces the complexity of data processing. For instance, a horizontal field of view (FOV) from a forward-looking camera complicates the geometry and additional image processing is required to extract accurate row information as compared to an orthographic FOV from a downward-looking camera. This was observed in the early versions of the robot where Hough Transform based row extraction was not as robust on horizontal FOV as on orthographic FOV. Further, a downward-looking view provides information from all around the robot: left and right and rear, in addition to front. Information from the side provides important information about the lateral displacement of the robot, whereas information from the rear improves the veracity of the robot heading obtained from the front. Finally, with the current set-up the robot can be driven backward just as easily as forward. Similarly, a forward-looking LIDAR ensures the navigation of the robot in the headland as well as between the rows; if the LIDAR is placed on one side or the other then the headland navigation will be restricted. The horizontal FOV of the LIDAR also allows the detection of the row ends which is not possible otherwise. Further, the low position of the LIDAR helps in deploying the robot at early stages of the plant growth. Therefore, appropriate positioning of the sensors on the robot improves the robustness and complexity of the navigation method in a cost effective way. This should be considered before

practical implementation of the robot in a production environment.

By virtue of the difference in the sensors and their positioning, there are subtle differences in the operating conditions of the vision-based and the LIDAR-based methods. The LIDAR-based method has a lower limit on the plant size in which it can operate. The plants should be tall enough to be detected reliably by the LIDAR, which means that in practice they should be more than 10-15 cm high, otherwise the laser beams will miss the plants more often than not. On the other hand, the vision-based method can operate with plants as small as 5 cm but runs into problems when the crop is tall and the soil can barely be seen from above. At the same time the LIDAR has a larger field of view compared to the camera. The LIDAR-based method is invariant to changes in the illumination while the vision-based method is not completely immune to the changes to the illumination. Since the two sensors are complementary they can be used simultaneously to expand the operating conditions of the robot.

6.3 Future Research Possibilities

The work in this thesis paves way to many interesting research topics. In particular, there is scope of improvement of all the methods described in the preceding chapters.

In case of weed detection, the GMRF algorithm can be improved by combining information across multiple resolutions to form a multi-resolution MRF that may yield texture features invariant to image resolution and texture scale which in turn might be helpful in tackling the problem of detection of small weeds. Information across multiple colour channels can be used by applying MRF across various channels to form a multi-channel MRF. Explicitly encoding the colour information in this manner can allow detection of different weed species.

Although, the camera and the LIDAR models are developed for 2D data they can easily be extended to 3D data. The core idea of the camera and the LIDAR models is the characterization of the density of the plant material in different regions. So just as the models can be extended to include more regions in the plane they can also be extended to include regions in space. From this perspective these models are applicable in other areas such as 3D modelling of plants.

With regards to the navigation methods, an interesting field of research is to extend them to fields with more complex geometries such as fields where crops are grown in a grid like pattern. Here more parameters will be needed to describe the geometry of the local world can easily accommodated in the navigation methods.

Another potential area of research is to investigate how to combine the

camera and the LIDAR measurement models in a single particle filter. Preliminary experiments of fusing the camera and the LIDAR data showed promising results. The robot was able to navigate the rows with both sensors as well as with either one of them. These experiments also highlighted that it is important to weigh the information from the sensors based on their accuracy before fusing them; otherwise it may lead to erroneous estimates. Typically, combining information from both the sensors will result in more accurate and robust estimates of the robot-environment state. It will also help in widening the operating conditions of the robot. For instance, when plants are very small and cannot be detected by the LIDAR, the camera can be used to estimate the state of the system and similarly information from LIDAR can be used when none is available from the camera.

Further, the navigation method described in the thesis performs only robot localization but no mapping. A navigation algorithm that can perform both localization and mapping simultaneously, provides a key advantage in that the robot can autonomously map details of large fields that can be use subsequently for more efficient management. One way to go about the mapping problem is to exploit the geometric nature of the measurement models. Since the measurement undergoes the least amount of processing it can be used to construct an occupancy grid map of the field where the map is discretized into grid cells and the value of the grid cell indicates the presence or absence of a plant material. Such maps will provide detailed information about the distribution of the plants in the field and also reveal problem areas such as large gaps in the rows.

The improvement in the navigation methods can also be realized by enhancing the particle filtering algorithm. For instance, the algorithm parameters Θ and Λ may sometimes depend on the density of plant material. Thus it is desirable that they can be estimated online like the state variables. Here the challenge is to integrate slowly changing parameters with the state variables into a single vector and to devise an appropriate update rule. Another area that may require attention is the problem of particle degeneracy in the particle filter. In this thesis this was not of great concern because the state space was small and the attenuation factor addressed the problem adequately. As the complexity of the applications increases, however, the question of how to prevent particle degeneracy and maintain particle diversity is a pressing concern where solutions include improving the proposal distribution, inducing particle diversity through Markov chain Monte Carlo sampling.

Apart from the methodological research mentioned above, there is also the opportunity to expand the scope of application of each method. For instance the two weed detection methods can be tested on different weeds or in crops other than grassland. In the same vein, both the vision-based and LIDAR-based navigation methods can be tested in fields other fields as we speculated

above.

Future farming systems capable of meeting the growing food demand in a sustainable manner need to employ automation in a distributed manner at a large scale. These farms will become more and more automated in which multiple autonomous robots are performing various tasks such as scouting, weeding and harvesting using local information as well as make independent decisions. The farmer will act as a remote supervisor to assist the machines in case of exceptional circumstances. Probabilistic methods are essential for the machines to operate in such semi-structured environment so as to manage, communicate and assess the uncertain information effectively. Through this thesis we hope to throw light on the efficacy of probabilistic methods in the agricultural domain.

Bibliography

- Ahamed, T., Takigawa, T., Koike, M., Honma, T., Yoda, A., Hasegawa, H., Junyusen, P., and Zhang, Q. (2004). Characterization of laser range finder for in-field navigation of autonomous tractor. In *Proceedings of the International Conference on Automation Technology for Off-road Equipment, 2004, Kyoto, Japan*, pages 120–130.
- Ahmad, U., Kondo, N., Arima, S., Monta, M., and Mohri, K. (1998). Algorithm to find center-point of detected weed in lawn field. *Journal of the Japanese Society of Agricultural Machinery*, 60:355–356.
- Ahmad, U., Kondo, N., Arima, S., Monta, M., and Mohri, K. (1999). Weed detection in lawn field using machine vision. *Journal of the Japanese Society of Agricultural Machinery*, 61(2):61–69.
- Ahmad, U., Kondo, N., Shibano, Y., Monta, M., Arima, S., and Nakamura, H. (1997). Feasibility of weed detection in lawn field based on gray-scale uniformity. *Journal of the Japanese Society of Agricultural Machinery*, 59:347–348.
- Aitkenhead, M., Dalgetty, I., Mullins, C., McDonald, A., and Strachan, N. (2003). Weed and crop discrimination using image analysis and artificial intelligence methods. *Computers and Electronics in Agriculture*, 39(3):157–171.
- Åstrand, B. (2005). *Vision based perception for mechatronic weed control*. PhD thesis, Chalmers University of Technology, Sweden.
- Barawid Jr., 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.
- Bell, T. (2000). Automatic tractor guidance using carrier-phase differential gps. *Computers and electronics in agriculture*, 25(1):53–66.

- Bergerman, M., Singh, S., and Hamner, B. (2012). Results with autonomous vehicles operating in specialty crops. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pages 1829–1835.
- Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal Of The Royal Statistical Society. Series B (Methodological)*, 48(3):259–302.
- 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.
- Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Blake, A., Rother, C., Brown, M., Perez, P., and Torr, P. (2004). Interactive image segmentation using an adaptive GMMRF model. In Pajdla, T. and Matas, J., editors, *Computer Vision - Eccv 2004, Pt 1*, volume 3021 of *Lecture Notes In Computer Science*, pages 428–441.
- Bohm, H. and Vershwele, A. (2004). Ampfer- und distelbekämpfung im ökologischen landbau. control of docks and thistles in organic agriculture. In Rahmann, G. and Kuhne, S., editors, *Ressortforschung für den Ökologischen Landbau 273*, pages 39–48, Bundesforschungsanstalt für Landwirtschaft (FAL), Braunschweig, Germany.
- Bond, W. and Grundy, A. (2001). Non-chemical weed management in organic farming systems. *Weed Research*, 41(5):383–405.
- Boykov, Y. and Kolmogorov, V. (2003). Computing geodesics and minimal surfaces via graph cuts. pages 26 –33 Vol.1.
- Boykov, Y. and Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions of Pattern Analysis And Machine Intelligence*, 26(9):1124–1137.
- Boykov, Y., Veksler, O., and Zabih, R. (1998). Markov random fields with efficient approximations. In *Computer Vision And Pattern Recognition*, pages 648 –655.
- Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 23(11):1222 –1239.
- Chellappa, R. (1985). Two-dimensional discrete Gaussian Markov random field models for image processing. *Progress in Pattern Recognition*, 2:79–112.

- Chellappa, R. and Chatterjee, S. (1985). Classification of textures using Gaussian Markov random fields. *IEEE Transactions On Acoustics, Speech, And Signal Processing*, Assp-33(4):959–963.
- Chen, Z., Samarabandu, J., and Rodrigo, R. (2007). Recent advances in simultaneous localization and map-building using computer vision. *Advanced Robotics*, 21(3-4):233–265.
- Clausi, D. (2001). Comparison and fusion of co-occurrence, Gabor and MRF texture features for classification of sar sea-ice imagery. *Atmosphere-Ocean*, 39(3):183–194.
- Clausi, D. (2002). An analysis of co-occurrence texture statistics as a function of grey level quantization. *Canadian Journal of remote sensing*, 28(1):45–62.
- Dash, M. and Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis*, 1:131–156.
- Dellaert, F., Fox, D., Burgard, W., and Thrun, S. (1999). Monte Carlo localization for mobile robots. In *Proceedings of IEEE International Conference On Robotics and Automation, 1999*, pages 1322–1328.
- Doucet, A. and Johansen, A. M. (2009). *The Oxford Handbook of Nonlinear Filtering*, chapter A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later. Oxford University Press.
- du Buf, J., Kardan, M., and Spann, M. (1990). Texture feature performance for image segmentation. *Pattern Recognition*, 23(3-4):291 – 309.
- Duda, R. and Hart, P. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15.
- Duda, R., Hart, P., Stork, D., et al. (2001). *Pattern classification*, volume 2. Wiley New York.
- Edan, Y., Han, S., and Kondo, N. (2009). *Automation in agriculture*, in *Handbook of Automation*. Springer.
- Efros, A. and Leung, T. (1999). Texture synthesis by non-parametric sampling. In *International Conference on Computer Vision*, pages 1033–1038.
- Foglia, M. M. and Reina, G. (2006). Agricultural robot Radicchio harvesting. *Journal Of Field Robotics*, 23(6-7):363–377.

- Fujii, Y. and Hayashi, M. (1989). Boundary detecting method and apparatus for automatic working vehicle. US Patent 4,868,752.
- Gan-Mor, S., Ronen, B., Josef, S., and Bilanki, Y. (1996). Guidance of automatic vehicle for greenhouse transportation. In *International Conference on Greenhouse Technologies 443*, pages 99–104.
- Gebhardt, S. and Kuehbauch, W. (2007). A new algorithm for automatic Rumex obtusifolius detection in digital images using colour and texture features and the influence of image resolution. *Precision Agriculture*, 8(1-2):1–13.
- Geman, S. and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 6(6):721–741.
- Gerrish, J. and Surbrook, T. (1984). Mobile robots in agriculture. In *First International Conference on Robotics and Intelligent Machines in Agriculture, 1984, St. Joseph, MI*, pages 30–41.
- Golzarian, M., Lee, M., and Desbiolles, J. (2012). Evaluation of color indices for improved segmentation of plant images. *Transactions of the American Society of Agricultural Engineers*, 55(1):261–273.
- Gross, R. and Brajovic, V. (2003). An image preprocessing algorithm for illumination invariant face recognition. In *Audio- and Video-Based Biometric Person Authentication*, pages 10–18. Springer.
- Guijarro, M., Pajares, G., Riomoros, I., Herrera, P., Burgos-Artizzu, X., and Ribeiro, A. (2011). Automatic segmentation of relevant textures in agricultural images. *Computers and Electronics in Agriculture*, 75(1):75–83.
- Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(3):1157–1182.
- Hague, T. and Tillett, N. (1996). Navigation and control of an autonomous horticultural robot. *Mechatronics*, 6(2):165–180.
- Hammersley, J. and Clifford, P. (1971). Markov fields on finite graphs and lattices. The Famous Hammersley Clifford Theorem.
- 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.
- Haralick, R. (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786 – 804.

- Haralick, R., Shanmugam, K., and Dinstein, I. (1973). Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, 3(6):610–621.
- Heidman, B., Abidine, A., Upadhyaya, S., Hills, D., and Robert, P. (2002). Application of RTK GPS based auto-guidance system in agricultural production. In *Proceedings of the 6th International Conference on Precision Agriculture and Other Precision Resources Management, 2002, Minneapolis, USA*, pages 1205–1214.
- Hough, P. (1962). Method and means for recognizing complex patterns. US Patent 3,069,654.
- Juan, O. and Boykov, Y. (2006). Active graph cuts. In *Computer Vision and Pattern Recognition, 2006*, IEEE Computer Society Conference on, pages 1023–1029.
- Kammler, D. (2007). *A First Course in Fourier Analysis*. Cambridge University Press.
- Keicher, R. and Seufert, H. (2000). Automatic guidance for agricultural vehicles in europe. *Computers and Electronics in Agriculture*, 25(1-2):169–194.
- Kempenaar, C., Lotz, L., Van Der Horst, C., Beltman, W., Leemans, K., and Bannink, A. (2007). Trade off between costs and environmental effects of weed control on pavements. *Crop Protection*, 26(3):430–435.
- Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220:671–680.
- Kise, M., Zhang, Q., and Mas, F. (2005). A stereovision-based crop row detection method for tractor-automated guidance. *Biosystems Engineering*, 90(4):357–367.
- Kohavi, R. and John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324.
- Kondo, N. and Ting, K. (1998). *Robotics For Bioproduction Systems*. American Society of Agricultural Engineers (ASAE).
- Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01*, pages 282–289.

- Larsen, W., Nielsen, G., and Tyler, D. (1994). Precision navigation with GPS. *Computers and Electronics in Agriculture*, 11(1):85–95.
- Laws, K. (1980a). Rapid texture identification. In *Image Processing for Missile Guidance*, volume 238, pages 376–381.
- Laws, K. (1980b). *Textured Image Segmentation*. PhD thesis, University Of Southern California Los Angeles Image Processing Institute.
- 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.
- Li, S. Z. (2001). *Markov Random Field Modeling in Image Analysis*. Springer-Verlag New York, Inc., Secaucus, Nj, Usa.
- Liu, H. and Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):491–502.
- Marchant, J. (1996). Tracking of row structure in three crops using image analysis. *Computers and electronics in agriculture*, 15(2):161–179.
- Marchant, J. and Brivot, R. (1995). Real-time tracking of plant rows using a Hough Transform. *Real-Time Imaging*, 1(5):363–371.
- McCullagh, P. and Nelder, J. (1989). *Generalized linear models*, volume 37. Chapman & Hall/CRC.
- Meyer, G., Mehta, T., Kocher, M., Mortensen, D., and Samal, A. (1998). Textural imaging and discriminant analysis for distinguishing weeds for spot spraying. *Transactions of the American Society of Agricultural Engineers*, 41(4):1189–1197.
- Miller, P. and Astley, S. (1992). Classification of breast tissue by texture analysis. *Image and Vision Computing*, 10(5):277–282.
- Morgan, K. (1958). A step towards an automatic tractor. *Farm mech*, 10(13):440–441.
- Niggli, U., Nsberger, J., and Lehmann, J. (1993). Effects of nitrogen fertilization and cutting frequency on the competitive ability and the regrowth capacity of rumex obtusifolius l. in several grass swards. *Weed Research*, 33(2):131–137.
- Otsuka, A. and Taniwaki, K. (1996). Round leaf weed detection in lawn field using texture analysis. In *Preprint of 55th JSAM Annual Meeting*.

- Otsuka, A. and Taniwaki, K. (1999). Extraction of broad-leaf weeds from grassland by texture analysis. *Journal of the Japan Society of Photogrammetry and Remote Sensing*, 38(6):6–13.
- Rao, A. (1990). *A taxonomy for texture description and identification*. Springer-Verlag New York, Inc., New York, NY, USA.
- Reed, T. and Dubuf, J. (1993). A review of recent texture segmentation and feature extraction techniques. *CVGIP-IMAGE UNDERSTANDING*, 57(3):359–372.
- Reid, J. and Searcy, S. (1986). Detecting crop rows using the Hough transform. *American Society of Agricultural Engineers. Microfiche collection*.
- Reid, J. and Searcy, S. (1987). Vision-based guidance of an agricultural tractor. *IEEE Control Systems Magazine*, 7(2):39–43.
- Reid, J., Zhang, Q., Noguchi, N., and Dickson, M. (2000). Agricultural automatic guidance research in North America. *Computers and Electronics in Agriculture*, 25(1-2):155–167.
- Richey, C. (1959). Automatic pilot for farm tractors. *Agricultural Engineering*, 40(2):78–79.
- Rother, C., Kolmogorov, V., and Blake, A. (2004). Grabcut - interactive foreground extraction using iterated graph cuts. *Acm Transactions On Graphics*, 23(3):309–314.
- Rushing, K. (1971). Developing the driverless tractor. *Agricultural Engineering*, 52(5):260–262.
- Shmulevich, I., Zeltzer, G., and Brunfeld, A. (1989). Laser scanning method for guidance of field machinery. *Transactions Of The Asae*, 32(2):425–430.
- Slaughter, D., Giles, D., and Downey, D. (2008). Autonomous robotic weed control systems: A review. *Computers And Electronics In Agriculture*, 61(1):63–78.
- Southall, B., Hague, T., Marchant, J. A., and Buxton, B. (2002). An autonomous crop treatment robot: Part I. a kalman filter model for localization and crop/weed classification. *International journal Of Robotics Research*, 21(1):61–74.
- 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 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.
- Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., and Rother, C. (2008). A comparative study of energy minimization methods for Markov Random Fields with smoothness-based priors. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 30(6):1068–1080.
- Tang, L., Tian, L., and Steward, B. (2003). Classification of broadleaf and grass weeds using Gabor wavelets and an artificial neural network. *Transactions of the American Society of Agricultural Engineers*, 46(4):1247–1254.
- Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. The Mit Press.
- Tillett, N. and Nybrant, T. (1990). Leader cable guidance of an experimental field gantry. *Journal Of Agricultural Engineering Research*, 45(4):253–267.
- van Evert, F., Polder, G., van der Heijden, G., Kempenaar, C., and Lotz, L. A. (2009). Real-time vision-based detection of *Rumex obtusifolius* in grassland. *Weed Research*, 49(2):164–174.
- Van Evert, F., Samsom, J., Polder, G., Vijn, M., Dooren, H.-J., Lamaker, A., Van Der Heijden, G., Kempenaar, C., Van Der Zalm, T., and Lotz, L. (2011). A robot to detect and control broad-leaved dock (*Rumex obtusifolius* L.) in grassland. *Journal of Field Robotics*, 28(2):264–277.
- van Evert, F. K., van der Bijl, M., Lamaker, A., Stravers, T., Polder, G., van der Heijden, G. W., Kroon, B., Knol, J., Dhaene, M., van der Zalm, A., Bakker, T., and Lotz, L. (2011). Hugo. In *Proceedings of the 8th Field Robot Event, 2010, Braunschweig, Germany*, pages 88–99. Available online at <http://www.digibib.tu-bs.de/?docid=00041345>.
- Van Gool, L., Dewaele, P., and Oosterlinck, A. (1985). Texture analysis anno 1983. *Computer Vision, Graphics, and Image Processing*, 29(3):336 – 357.
- Van Middelkoop, J., DE Visser, M., and Schilder, H. (2005). Control of broad-leaved dock in organic grassland in the bioveem project. In *Animal Sciences Group Report*.

- Vogiatzis, G., Torr, P., and Cipolla, R. (2005). Multi-view stereo via volumetric graph-cuts. In *Computer Vision And Pattern Recognition, Vol 2, Proceedings*, IEEE Computer Society Conference, pages 391–398.
- Watchareeruetai, U., Takeuchi, Y., Matsumoto, T., Kudo, H., and Ohnishi, N. (2006). Computer vision based methods for detecting weeds in lawns. *Machine Vision and Applications*, 17:287–296.
- Watchareeruetai, U., Takeuchi, Y., Matsumoto, T., Kudo, H., and Ohnishi, N. (2008). Modified lawn weed detection: Utilization of edge-color based SVM and grass-model based blob inspection filterbank. In *Neural Information Processing*, pages 30–39. Springer.
- Weiss, U. and Biber, P. (2011). Plant detection and mapping for agricultural robots using a 3d LIDAR sensor. *Robotics and Autonomous Systems*, 59(5):265–273.
- Weszka, J. and Dyer, C.R. and Rosenfeld, A. (1976). Comparative study of texture measures for terrain classification. *IEEE Transactions of System Man and Cybernetics*, 6(4):269–285.
- Wilson, J. (2000). Guidance of agricultural vehicles - a historical perspective. *Computers and Electronics in Agriculture*, 25(1-2):3–9.
- Zalesny, A. and Gool, L. V. (2001). A compact model for viewpoint dependent texture synthesis. In *Lecture Notes in Computer Science*, pages 124–143. Springer.
- Zaller, J. (2004). Ecology and non-chemical control of *Rumex crispus* and *R. obtusifolius* (polygonaceae): A review. *Weed Research*, 44(6):414–432.
- Zaller, J. (2006). Sheep grazing vs. cutting: Regeneration and soil nutrient exploitation of the grassland weed *Rumex obtusifolius*. *BioControl*, 51(6):837–850.
- Zhao, Y., Zhang, L., Li, P., and Huang, B. (2007). Classification of high spatial resolution imagery using improved gaussian markov random-field-based texture features. *Geoscience And Remote Sensing, IEEE Transactions On*, 45(5):1458–1468.

Acknowledgements

This thesis was possible due to the help of several people and I would like to take this opportunity to thank them. First of all, I would like to express my deepest gratitude to my supervisors Alfred Stein, Cajo ter Braak and Gerie van der Heijden for their guidance and generous support on all aspects of the thesis. It was my privilege to work under their supervision and I am indebted for their invaluable help. They taught me, challenged me, contradicted me, supported and reared me during the entire course of my doctorate studies both academically and otherwise. They picked me up and walked me through when I was down as well as let me fly during good times.

Secondly, I would like to thank Frits van Evert for spending many hours teaching me the robot software, for his assistance during field experiments as well as for providing data needed for the thesis. He also made important contributions towards formulating the research objective. My thanks also extends to Valentyn Tolpekin for all his help during our collaboration which culminated in a publication in *Machine Vision and Application*. His knack for simplicity and rigour helped me get my head around difficult topics like MRF and Graph Cuts.

My appreciation also goes to Biometris and Faculty of Geo-Information Science and Earth Observation (ITC) of the University of Twente for facilitating this joint research. Mr. Martijn van der Bijl and colleagues of Kverneland Group, Nieuw Vennep, The Netherlands, provided us with the robot including the microcontroller (low-level control) software for which we are grateful. Assistance provided by Mr. Arjan Lamaker, Wageningen, The Netherlands, in terms of writing the PC (high-level control) software is also greatly appreciated.

Thanks to my colleagues and friends at Biometris, it was a great workplace. I was in great company with my fellow PhD students Yiannis, Apri, Andries, Robert, Jeroen, Cassandra, Rianne, Thomas, Maggie, Nurudeen, Nadia and George and the post-docs Laura, Sabine, Willem, Simon, Maaike, Patricia, Marian, Elisa and other researchers Sanne, Jaap Kokorian, Martin, Wies and all the other colleagues who made Biometris an exiting place. I became good friends with Yiannis, Robert and Wies with whom I spent much quality time. Additional thanks to Apri and Yiannis for helping me with the

final formatting of the thesis. I also want to extend my gratitude to the two secretaries of Biometris, Dinie and Hanneke, for all their help with the bureaucratic work.

Living in a foreign country for a long period can be difficult. I was, however, fortunate to have fantastic friends and Wageningen was home away from home. Thanks to Andrei and Kanchi who always made me comfortable no matter the situation. My fabulous house-mates Sander, Susann, Etienne, Daniela, Bart, Charel, Maria, Angel, Jeroen and Haimil were always lively and made my stay in Wageningen more than fun.

I am greatly indebted to my former supervisor Jan Wessnitzer at Edinburgh University for introducing me to the methods of scientific enquiry during my MSc and teaching me that it can be fun. I am also grateful to Bob Fisher and Subramanian Ramamoorthy of Edinburgh University who were inspiring and helped me embark on my doctoral studies.

Lastly, but most importantly, I would like to thank my family for all their sacrifices without which I could not have started my MSc and thus my doctoral studies. I will forever be indebted to my parents who spent most of their life's earnings to provide for my MSc and for their emotional cocoon that ensured my smooth journey through my doctoral studies. Mother and Father, thank you for all your love and support. I want to thank my brother Sandesh for being an inspiration and helping me stay focused during the difficult phase of my doctoral studies.

List of Publications

Hiremath, S., van der Heijden, G., van Evert, F.K. & Stein, A. The role of textures to improve the detection accuracy of *Rumex obtusifolius* in robotic systems. *Weed Research*, 2012, Vol. 52(5), pp. 430-440.

Hiremath, S., Tolpekin, V., Heijden, G. & Stein, A. Segmentation of *Rumex obtusifolius* using Gaussian Markov random fields. *Machine Vision and Applications*, 2013, Vol. 24(4), pp. 845-854.

Hiremath, S., van Evert, F.K., ter Braak C.J.F., Stein, A. & van der Heijden, G. Image-Based particle filtering for navigation in a semi-structured agricultural environment. Submitted for publication.

Hiremath, S., van der Heijden, G., van Evert, F.K., Stein, A. & ter Braak C.J.F. Laser range finder model for autonomous navigation of a robot in a Maize field using particle filter. Submitted for publication

Curriculum Vitae

Santosh Hiremath was born on 16th June 1982 in Davangere, India. He received his BSc in Computer Science and Engineering from Visvesvaraya Technological University (VTU), India in 2004. In the final year of his undergraduate education he was recruited to work for Sonata Softwares Ltd as a software engineer. From June 2004 to 2007, he worked in the domain of Data-Warehousing where his main work involved developing reporting solutions using Microsoft SQL Server for Data Analysis. In September 2007, he went to Edinburgh University to do his MSc and in October 2008 he received his MSc in Artificial Intelligence. During his MSc, Santosh became interested in Probabilistic Robotics and started his PhD research in the same topic at Biometris Department, Wageningen University in January 2009. Currently, he is on an extended holiday in India and wishes to pursue his post-doctoral research.

PE&RC PhD Training Certificate

With the educational activities listed below the PhD candidate has complied with the educational requirements set by the C.T. de Wit Graduate School for Production Ecology and Resource Conservation (PE&RC) which comprises of a minimum total of 32 ECTS (= 22 weeks of activities)



Review of literature (4.5 ECTS)

- Probabilistic robotics, Monte Carlo simulation, particle filters (2009)

Writing of project proposal (1.4 ECTS)

- Probabilistic methods for robotics in agriculture

Post-graduate courses (7.5 ECTS)

- International computer vision summer school; University of Catania, University of Cambridge (2009)
- Front end vision and Multi-scale image analysis; Biomedical Department, TU Eindhoven (2009)
- Uncertainty propagation; PE&RC Wageningen University (2009)
- Linear models; PE&RC, Wageningen University (2012)
- Generalized linear models; PE&RC, Wageningen University (2012)
- Mixed linear models; PE&RC, Wageningen University (2012)

Laboratory training and working visits (4.5 ECTS)

- Detection of *R. Obtusifolius* Using GMRF; ITC, Enschede (2010-2011)

Deficiency, refresh, brush-up courses (3 ECTS)

- Machine learning (2012)

Competence strengthening / skills courses (1.5 ECTS)

- Moral dilemmas in daily scientific practices; Wageningen Graduate Schools (WGS) (2012)
- Competence assessment; PE&RC, Wageningen University (2012)

PE&RC Annual meetings, seminars and the PE&RC weekend (2.7 ECTS)

- PE&RC Day (2009)
- PE&RC Weekend (2009)
- PE&RC Weekend (2012)

Discussion groups / local seminars / other scientific meetings (4.5 ECTS)

- Vision and robotics (previously, green vision) (2010-2011)
- Spatial statistics (2010-2011)
- Maths&Stats (2010-2011)
- Maths and statistics network (2012)

International symposia, workshops and conferences (5.7 ECTS)

- NVPHBV (2011)
- Spatial Statistics Conference (2011)
- AgRob, Portugal (2012)

This research was carried out and financed by Biometris of Wageningen University and the Faculty of Geo-Information Science and Earth Observation (ITC) of the University of Twente.

The cover of this thesis was designed by Santosh A. Hiremath. The robot in the picture was provided by Kverneland Group, Nieuw Vennep, The Netherlands.

Printed and published by Uitgeverij BOXPress Proefschriftmaken.nl