Plants detection, localization and discrimination using 3D machine vision for robotic intra-row weed control

by

Jingyao Gai

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Agricultural and Biological Engineering

Program of Study Committee:
Lie Tang, Major Professor
Brian L. Steward
Kathleen Delate

Iowa State University

Ames, Iowa

2016

Copyright © Jingyao Gai, 2016. All rights reserved.

TABLE OF CONTENTS

LIST OF FIGURES iv

LIST OF TABLES xii

ACKNOWLEDGMENTS xiv

ABSTRACT xv

CHAPTER 1. GENERAL INTRODUCTION 1

1.1 Introduction 1
1.2 Robotic Mechanical Weeding 2
1.3 Machine Vision in Robotic Weeding 5
2D vision applications: 5
3D sensing applications: 7
1.4 Research Objectives 10
1.5 Thesis Overview 11
1.6 References 11

CHAPTER 2. METHODS FRAMEWORK AND DATA COLLECTION 14

2.1 Introduction 14
2.2 Methods Framework 14
2.3 Sensor Evaluation 15
2.4 Data Collection 19
2.4.1 Target species 19
2.4.2 View angle and working distance selection 22
2.4.3 Apparatus design and field design for data acquisition 24
2.5 Algorithm Design Overview 26
2.6 Reference 27

CHAPTER 3. PLANTS DETECTION, LOCALIZATION AND DISCRIMINATION USING 3D MACHINE VISION FOR ROBOTIC INTRA-ROW WEED CONTROL– PART A: CROP PLANT CANDIDATE DETECTION 29

3.1 Abstract 29
3.2 Introduction 30
2D vision applications: 31
3D sensing applications: 32
3.3 Materials 36
3.3.1 Sensor 36
3.3.2 Location and agronomic trial 36

3.4 Algorithm Design 37
3.4.1 Preprocessing on depth image 39
3.4.2 Segmentation using depth and color 45
3.4.3 Pixels clustering & crop plant candidate extraction 57
3.5 Results and Discussion 59
3.6 Conclusion 61
3.7 References 62

CHAPTER 4. PLANTS DETECTION, LOCALIZATION AND DISCRIMINATION USING 3D MACHINE VISION FOR ROBOTIC INTRA-ROW WEED CONTROL– PART B: PLANT DISCRIMINATION 65

4.1 Abstract 65
4.2 Introduction 66
4.3 Plant Features 69
4.4 Algorithm Design 72
4.4.1 Plant leaf segmentation 74
4.4.2 Grouping & localization 76
4.4.3 Supervising user interface 86
4.4.4 Classification methods 87
4.5 Experimental Results 97
4.5.1 Grouping & localization 98
4.5.2 Crop/weed classification 99
4.6 Discussion and Conclusion 105
4.7 References 106

CHAPTER 5. SUMMARY AND CONCLUSIONS 108

LIST OF FIGURES

Figure 2-1. (a) Kinect v2 sensor used in this study. (b) The colored point cloud output from fusing 3D depth information and 2D color information. 16

Figure 2-2. Teardown picture of a Kinect v2 sensor. Image was taken from (Fankhauser et al., 2015)..... 16

Figure 2-3. Axial noise level (standard deviation of measured depth values) at different working distances (z-distance) and different angles of observed object surface (θ), in both indoor (a) and outdoor (b) tests (Fankhauser et al., 2015). The θ is defined as the angle difference between the direction of sensor and the surface normal. 19

Figure 2-4. Example images of weeds in this study. From up left to bottom right: bromegrass (Bromus inermis Leyss), pigweed (Amaranthus spp.), Lambsquaters (Chenopodium album), waterhemp (Amaranthus rudis), barnyardgrass (Echinochloa crus-galli), bindweed (Convolvulus arvensis), purslane (Portulaca oleracea), and white clover (Trifolium repens). 22

Figure 2-5. The sensor pose and the desired working distance while acquiring data. 24

Figure 2-6. The data flowchart of the image processing system 27

Figure 3-1. (a) Kinect v2 sensor used in this study. (b) Teardown picture of a Kinect v2 sensor (Fankhauser et al., 2015). (c) Example 3D point cloud registered with

color..... 36

Figure 3-2. The data flowchart of the image processing system 39

Figure 3-3. An example of depth image indicating the z-direction distance acquired using a Kinect v2 sensor. Noise level is higher in the non-center area, especially at corners. Unit in mm. 40

Figure 3-4. The strategy of the neighbor counts filter. 43

Figure 3-5. The performance shift of the proposed method compared with methods implemented in PCL and stated by . Time measured in seconds. 45

Figure 3-6. Corn plant example showing the differences before (a) and after (b) preprocessing. The point cloud is generated from corn crops in the laboratory. After applying preprocessing procedure, spare noise and bad points are removed. 45

Figure 3-7. The upper leaf and bottom leaf figures show the assumptions of the illuminant-invariant method. The right figure shows the ill-inv space and projection of different color under different illuminations. It is clear that same color under different light conditions lies a specific line. Different colors are on different lines. Thus different colors can be separated. 47

Figure 3-8: The effect of illuminant-invariant map. In the left figure, the light conditions are different between shaded and unshaded area. And in the right figure, the illuminant-invariant map is generated and compensates the effect of changing light conditions. Green pixels have lower values in illuminant-invariant maps. 48

Figure 3-9. The RANSAC detection algorithm 50

Figure 3-10. Fig (a) shows a synthetic 2D point cloud of a plant with a ground plane to visualize the principle of the ground detection. With RANSAC algorithm, two points are randomly selected (blue), and a line is fitted using these points (blue line). Then the model is refined using robust linear regression (red line). The points between the brown lines become inlier standing the ground. Fig (b) shows a point cloud with only outliers after ground detection. The points of the ground are successfully removed. 52

Figure 3-11. The standard logistic response function curve with one variable, with $\beta_0 = \beta_1 = 1$, which transforms $-\infty, +\infty$ to $[0,1]$. 54

Figure 3-12. The generation of a ROC curve. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. 54

Figure 3-13. The ROC space and plots of 4 example predictions. The higher the curve is, the more likely the classifier will give the correct result. The predictions on the diagonal dashed line are equivalent to randomly guessing. 55

Figure 3-14. Sample segmentation successes. The left figure shows the one color image, and the right figure shows the segmentation result point cloud. The background is eliminated, and the pixels belonging to crops are extracted and separated as clusters. 57

Figure 3-15. Sample segmentation failures. The left figure shows a segmentation failure case in tomato field, in which the ground is covered with weeds. The right figure

shows a lettuce field case, in which the depth-based segmentation is failed, because of the low crops height. Color-based segmentation will still work on the latest case..... 57

Figure 3-16. A run-time example showing the plants clustering procedure. Fig (a) is the point cloud generated from Kinect v2 sensor after preprocessing. And Fig (b) is the result after background removal and clustering. Different clusters are assigned with different colors. Ideally, each cluster stands for a plant..... 59

Figure 3-17. Examples of the crop plant candidate detection run-time result. In those images, the detected crop plant candidates are circled..... 61

Figure 4-1. The flowchart of the image processing procedures..... 73

Figure 4-2. A run-time output example showing the leaf extraction procedures. Left: Color image of on cluster after segmentation and clustering, with connected crops, occlusions, as well as broken leaves. Middle: Combined gradient magnitude image on both color and depth. Right: Leaves extraction result example. Most of the leaves are extracted and labeled with different colors..... 76

Figure 4-3. One example image of broccoli collected in the field, with damaged leaves and interconnected canopies. Leaf venation was found a robust feature to determine the leaf directions..... 78

Figure 4-4. Surfaces with different eigenvalues of their Hessian matrixes. A flat surface has eigenvalues to be both zero, positive eigenvalues correspond to valleys or saddles, and negative eigenvalues correspond to ridges or peaks. 79

Figure 4-5. A run-time output example showing venation extraction methods. The left figure shows the map of lower eigenvalues of Hessian Matrixes at each location. Ridges are colored with brighter colors. The right figure shows the veins extracted by thresholding, with noise. 80

Figure 4-6. A run-time output example showing the venation skeletons extracted from the same leaf combination. The left figure shows the potential combinations of leaves. The right figure shows the skeletonized venation map extracted from the bottom leaves combination, which is passed to the center finding step. 81

Figure 4-7. The patterns which the joints involve after thinning with the method proposed by Zhang & Suen, (1984). The upper row shows the ‘Y’ pattern in 4 directions, and the lower row shows the ‘ㅏ’ pattern in 4 directions. All the pixels with these patterns are joints. 83

Figure 4-8. The algorithm of defining the weight factor of each line segment when calculating the center. 83

Figure 4-9. A run-time output example showing the results after the first and second steps in center finding. The left figure shows the venation skeletons of one potential leaves combination. In the right figure, the joints are colored in red, and the fitted line segments are drawn in green, with their lengths equal to their weight factors. 84

Figure 4-10. The algorithm of center finding using vein line segments. 85

Figure 4-11. A run-time output example showing the iterations of finding the

center. The green cross indicates the resultant center location in the current iteration. The line segments with white color are the active line segments (inliers, with non-zero calc-weights). And the line segments in red are the inactive line segments (outliers, with zero calc-weights) in the current iteration. 85

Figure 4-12. The supervising user interface. It can realize plant-based supervising and leaf-based supervising. Plants detected are labeled with circles. With buttons on the right side, the operator can identify the crops and weeds. The supervising input as well as the features of each plant are saved. 86

Figure 4-13. Example of k-NN classification. The prediction is voted by its k nearest neighbors. The classification result should be *Red* when k = 3 (solid line), and should be *Blue* if k = 6 (dashed line). Adopted from K-nearest neighbors algorithm, ln Wikipedia, n.d., Retrieved May 28, 2007, from https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm. ©2007 by Antti Ajanki. Adopted with permission. 88

Figure 4-14. An example of logistic regression, showing the probability curve fitted using the data points. Ln Wikipedia, n.d., Retrieved August 17, 2015, from https://en.wikipedia.org/wiki/Logistic_regression. ©2015 by Michaelg2015. Adopted with permission. 90

Figure 4-15. An example of QDA visualization using generated data, showing the data points and the quadratic decision boundary of two classes. 91

Figure 4-16. An example of three-layer artificial neural network visualization,

showing the structure of the network, and the coefficients of each neuron. Each circle indicates a neuron, and the values above arrows indicates the weight of each input in each neuron. 93

Figure 4-17. An example of linear SVM classification. The problems is to find a hyperplane with maximum margin using data points. 94

Figure 4-18. The concept of forest, In Random forest, different trees are trained with random predictors combinations. When predicting, voting is performed by each tree in the forest. Adopted from Random Forest とその派生アルゴリズム, ln Hatena Blog, n.d., Retrieved Dec 4, 2013, from http://kazoo04.hatenablog.com/entry/2013/12/04/175402 ©2013 by kazoo04. Adopted with permission. 95

Figure 4-19. The algorithm of two-class discrete AdaBoost. 96

Figure 4-20. The concept of Adaboost and the procedure to ensemble weak classifiers (e.g. h_1, h_2 and h_3) into a strong classifier H. The size of each circle indicates the weight in each iteration. The prediction is based on the weighted votes of those weak classifiers. Adopted from Computer Vision Final Project: Viola-Jones & Morphology-based Face Detector, Retrieved 2005, from http://www.cc.gatech.edu/~kihwan23/imageCV/Final2005/CS7495%20Computer%20Vision%20Final%20Project.pdf ©2005 by Kihwan Kim. Adopted with permission. 97

Figure 4-21. Examples of the plants grouping and localization run-time result. In

those images, the detected and separated plants are circled the positions are labeled with crosses. For detecting Broccoli, segmented leaves are also labeled with squares. 99

Figure 4-22. Paired plot of predictors used in leaf classification stage of broccoli classification, showing the distribution of the training data. Red data points were supervised as crop leaves, and blue data points were supervised as weed leaves or incorrect segmentation. 100

Figure 4-23. Paired plot of predictors used in plant-scale crop/weed classification in the broccoli dataset, showing the distribution of the training data. Red data points were supervised as crops, and blue data points were supervised as weeds or incorrect segmentation. 102

Figure 4-24. Paired plot of predictors used in plant-scale crop/weed classification using the lettuce dataset, showing the distribution of the training data. Red data points were supervised as crops, and blue data points were supervised as weeds or incorrect segmentation. 104

LIST OF TABLES

Table 2-1. Kinect v2 Specifications 17

Table 2-2 Examples of 2D images taken using Kinect v2 sensor from different species at different time, as well as distinctive features that can be observed using Kinect v2 sensor..... 21

Table 2-3. Field of view and spatial resolution at different working distance for top view..... 24

Table 3-1 Fitted logistic response functions: coefficients and their corresponding AUC values. 55

Table 3-2. Table showing the description of different clustering methods, as well as their advantages and disadvantages. 58

Table 3-3. The candidate detection true positive rate and precision for broccoli and lettuce at different growth stages. 61

Table 4-1. The average localization error for broccoli and lettuce at different growth stages. 98

Table 4-2. Models evaluation and comparison result for broccoli leaf classification. Listing tuning parameters, training errors and CV errors. Adaboost model yields the best performance. 101

Table 4-3. Models evaluation and comparison result for broccoli plant

classification, listing tuning parameters, training errors and CV errors. 103

Table 4-4. Models evaluation and comparison result for lettuce plant classification, listing tuning parameters, training errors and CV errors. AdaBoost performed the best in plant classification. 104

ACKNOWLEDGMENTS

I would like to thank my committee chair, Dr. Tang, and my committee members, Dr.Steward and Dr. Delate, for their guidance and support throughout the course of this research.

In addition, I would also like to thank my colleagues, Yin Bao, Hang Lu, Dylan Shah, Jafni Johari Jiken, Safal Kshetri, who gave me helps and suggestions in this research. Also thank the department faculty and staff for making my time at Iowa State University a wonderful experience.

Finally, thanks to my parents and my wife for their encouragement, patience, respect and love.

ABSTRACT

Weed management is vitally important in crop production systems. However, conventional herbicide-based weed control can lead to negative environmental impacts. Manual weed control is laborious and impractical for large scale production. Robotic weeding offers a possibility of controlling weeds precisely, particularly for weeds growing close to or within crop rows. The fusion of two-dimensional textural images and three-dimensional spatial images to recognize and localize crop plants at different growth stages were investigated. Images of different crop plants at different growth stages with weeds were acquired. Feature extraction algorithms were developed, and different features were extracted and used to train plant and background classifiers, which also addressed the problems of canopy occlusion and leaf damage. Then, the efficacy and accuracy of the proposed methods in classification were demonstrated by experiments. Currently, the algorithms were only developed and tested for broccoli and lettuce. For broccoli plants, the crop plants detection true positive rate was 93.1%, and the false discover rate was 1.1%, with the average crop-plant-localization error of 15.9 mm. For lettuce plants, the crop plants detection true positive rate was 92.3%, and the false discover rate was 4.0%, with the average crop-plant-localization error of 8.5 mm. The results have demonstrated that 3D imaging based plant recognition algorithms are effective and reliable for crop/weed differentiation.

CHAPTER 1. GENERAL INTRODUCTION

1.1 Introduction

In recent years, with the development of greater health consciousness, consumers are becoming more interested in vegetables, especially natural, organic vegetables. In 2013, U.S. vegetable production resulted in a revenue of 11.4 billion dollars for fresh products, up 14 percent from 2012, with a total harvested area of 1.63 million acres (USDA-NASS, 2014). In organic vegetable production, there were 1.2 billion dollars in total for fresh productions, with total harvested area of 164,000 acres (USDA-NASS, 2014).

One important factor that affects crop yield is weed competition. Weeds are very competitive in obtaining moisture, sunlight, and nutrients, all of which adversely affect crop yield and quality. The National Organic Farmers' Survey conducted by Walz (2004) reported that organic farmers indicated weeds were one of the major causes of production losses, second only to weather-related losses. In addition, weed management is one of the most costly operations in vegetable production, especially for organic farming. Organic farmers have major production costs associated with weed control, which are mainly caused by the reliance on manual weeding. Earthbound Farms, the largest producer of organic vegetables in North America, reported that some of their farmers spent up to $1,000 per acre to control weeds manually (Zimdahl, 2013). It is obvious that labor costs have made manual weed

control impractical. For some farmers, moreover, the use of herbicides is becoming less desirable due to the emergence of herbicide-resistant weeds, the environmental impact of herbicide runoff, as well as societal demand for chemical-free foods (McErlich & Boydston, 2014). Realizing that, many tools have been developed to increase mechanical weeding efficacy in recent years.

Inter-row (between crop rows) mechanical weeding is relatively easy to achieve by using commercial tools. However, intra-row (within or close to crop rows) mechanical weeding has a risk of damaging crop plants. To date, there are limited tools for intra-row weeding in vegetable crops with desirable weed control efficacy. With the advancement of computational technology, automated robotic weeding offers a possibility of controlling weeds in a precise fashion, particularly for weeds growing near crops or within crop rows. Since identification and localization of plants have not yet been fully automated, research to address these problems is thus in great demand.

1.2 Robotic Mechanical Weeding

Mechanical weeding currently is the most promising way to control the weeds in non-chemical situations, especially in organic farming. Manual weeding is a traditional solution, using hand or hand-held tools. It is one of the most environmentally friendly weed management methods. However, manual weeding becomes impractical due to the labor costs. The efficacy of hand weeding is also limited by the operator’s experience and skill. In 1975,

short-handled hoe weeding was permanently banned by the California Industrial Safety Board. In 2004, the ban was extended to manual weeding.

With the advancement of agriculture technology, as well as people's growing demand for healthy living, many non-chemical weed management tools have been developed for large-scale use (Finney, 2008), in which very good performance was obtained for inter-row weeding (between crop rows). Most of the farmers managed weeds in one of three methods: burying, cutting, or uprooting. Machines have difficulty in locating space between crops in crop rows without human interaction. Only a few machines can accomplish intra-row weeding, which uses a selective mechanism or material, such as finger weeders, torsion weeders, and spinning brush/tine weeders. Among these, the brush weeders were reported to be the most promising technology because they could uproot, bury, and break weeds (Upadhyaya & Blackshaw, 2007). The Eco-weeder is a brush-type intra-row mechanical weeder that is widely used due to its simplicity as well as its high weed control effectiveness. But the Eco-weeder requires a human operator to guide two rotating brushes in and out of the crop row. Although such mechanical weeding tools were reported to be effective in controlling intra-row weeds, they will cause crop damage unless their end effectors are accurately guided (Slaughter, 2008).

In contrast, the use of automated robotic weeding has great potential for reducing the economic and environmental cost while providing high performance in intra-row weed control (Thompson et al., 1991). Several weeding robots have been developed (Slaughter,

2008), which have the ability to distinguish plants from weeds, as well as control weeds with actuators.

Three operating modes are widely applied on agricultural weeding robots (Slaughter, 2008). The first mode is based on row-following systems, which is used for locating crop rows then accordingly control carriers' steering and weeding actuators' positions at the same time. Several real-time crop row detection methods were developed. For instance, a method described by Søgaard & Olsen (2003) took advantage of infrared images as well as optimization algorithms in order to identify crop rows, and obtained an accuracy of ±12 mm. The limitation of this mode is that only crop rows can be found, without identifying crops. Thus, this method can only accomplish inter-row weeding.

The second mode is based on GPS mapping systems. Some research groups equipped planters with RTK GPS to generate crop maps during planting (Ehsani, Upadhyaya, & Mattson, 2004; Sun et al., 2010). With this method, only plants detected at pre-recorded planting positions were recognized as crops. Ehsani et al. (2004) tested this type of maize seed mapping system and reported that the average error between seed maps and actual plant positions after germination was about 34 mm. The error sources of the system are: RTK GPS's error, seed bounce in the furrows, as well as different soil conditions (Griepentrog, Nørremark, Nielsen, & Blackmore, 2005). On the other hand, this method highly depends on the data collected during planting, which is not flexible.

The third mode is using a machine vision based plant species identification system.

Computer vision has become an option in the inspection of agricultural plants, particularly when color and shapes need to be analyzed at high speed. Many applications in agricultural robotics such as plant discrimination and self-guidance can be realized with the power of computer vision (Astrand & Baerveldt, 2002). Plant morphology and structure have been focused on, which remains one of the most consistent methods of plant identification (Du, Wang, & Zhang, 2007).

1.3 Machine Vision in Robotic Weeding

By comparing those three methods used by previously developed weeding robots, the machine vision method seems to be the most reliable and flexible. There are many methods developed for robotic weeding based on computer vision. With sensing method as criteria, most of those computer vision systems can be categorized into two classes: two-dimensional (2-D) vision systems and three-dimensional (3-D) vision systems.

2D vision applications:

A 2D sensor is a type of sensor that can record light or other electromagnetic radiation reflected or emitted from objects, by focusing it on a light-sensitive surface. The word “2D” means horizontal and vertical dimensions in the image space, as it is a projection of the 3D real world.

Regular 2D cameras were the earliest ones used in robotic weeding. The problem of

plant identification was addressed by extracting color or morphological features from a leaf or a whole plant, such as length, width, perimeter dimensions, roundness, circularity, convexity and moment. Slaughter et al. studied these types of systems and concluded that they demonstrated high recognition accuracy only under ideal conditions, in which light was controlled and plants were sparseley distributed in images (Slaughter, 2008). Also, they are not robust to occlusion problems, or defects of the plants caused by insect damage or wind, which are common in the field. Moreover, with complex illumination conditions, such as strong sunlight, images will be saturated. Most of the algorithms will fail to segment plants out of the background.

Spectral reflectance characteristics of plants were reported to be effective in vegetation segmentation and crop/weed discrimination (Scotford & Miller, 2005; Zwiggelaar, 1998). However, the selected spectral wavebands for classification are generally different for different weed and crop pairs (Zwiggelaar, 1998). Therefore, selecting wavebands and designing algorithms for distinguishing crop plants from different weed species is complex. On the other hand, an NDVI (normalized difference vegetation index) map, which can be generated from an image from an infrared camera or NDVI sensor, is reported to be effective in vegetation segmentation. Sui et al. (2008) have developed a vision-based system for weed mapping using an NDVI camera. However, in order to accomplish crop/weed discrimination with NDVI images, morphological features are still needed. The advantages of spectral reflectance based methods include: they are less sensitive to environmental light, and the

infrared reflectance of plants can include additional information for plant discrimination.

3D sensing applications:

A 3D sensor is a type of sensor that can measure the distances between objects and sensors, otherwise known as “depth”. The word “3D” corresponds to the three dimensions of the real world. In recent research, with the development of sensing technology, a 3D sensor promises to address problems in 2D vision systems, such as occlusion. A 3D sensor can also give reliable information to perform plant discrimination and plant localization. In several studies (J. Li, 2014; Jin & Tang, 2009; Nakarmi & Tang, 2012), 3D sensors were applied in agricultural applications which provided a good performance. The advantages of 3D sensors for plant discrimination and localization are obvious: 3D sensors can provide fundamental depth information, making it is much easier to obtain the 3D structural and morphological data of the plants.

Today, three types of state-of-art 3D sensors are mainly used on mobile robots: stereo vision, laser, and PMD time-of-flight. In several articles (Sansoni, Trebeschi, & Docchio, 2009; Weiss & Biber, 2011), the authors have compared and evaluated those three types of sensor for mobile robot:

To receive 3D data using stereo vision, typically triangulation of two cameras or structure-from-motion technique are used. In the study by Jin & Tang (2009), a real-time sensing system for corn phenotyping was developed based on a stereo vision sensor.

However, due to the stereo camera's passive operation mode, it is hard to provide reliable data for accurate sensing: When receiving 3D data from stereo vision, the disparity calculation has a high dependency on the structures or features of the objects in images. Further, the precision and maximum depth are limited by the baseline between the cameras. Also, qualities of distance values decrease very fast as depth increases. The advantage of stereo vision is its high image resolution, with color information. One the other hand, cameras in stereo vision systems can also be modified into near-infrared cameras by replacing their filters with NIR filters, in order to take advantage of spectral reflectance information (Hunt et al., 2010).

The accuracy, resolution, frame rate as well as price of different 3D laser sensors can be vastly different. Some of the 3D laser sensors are implemented from 2D laser scanners which uses the line-scanning method, such as Lidar. One example is Kurt3D (Surmann & Nüchter, 2003) which equips a rotating 2D SICK® laser sensor to realize 3D laser scanning. However, it needs a stop-and-go mode for traveling to receive consistent 3D data, since a 2D line sensor was not built for 3D applications. Generally, 3D laser sensors usually have the properties of high weight, high power consumption, as well as high price. Those make those 3D sensors less desirable for small mobile robots.

The semiconductor based PMD time-of-flight (TOF) camera is the latest technique. It measures distance and infrared reflectance intensity information based on time-of-flight technique. A modulated light signal is emitted by sensor, reflected by objects then received

by sensor receiver (L. Li, 2014). The distances are calculated by the phase shift of the signal as well as the reflection intensity. There are many research applications using TOF cameras as sensing devices in agriculture, such as phenotyping (Alenyà, Dellen, Foix, & Torras, 2012), and plant spacing (Jin & Tang, 2009). In a study by Kazmi et al. (2014), the authors summarized the advantages along with drawbacks of TOF cameras. The advantages are: they deliver high frame rates as well as accurate depth data under suitable conditions. The limitations are: resolutions of depth images are often low; the sensors are sensitive to ambient sunlight, which usually leads to poor performance while working outdoors; the quality of depth values depends on the color of objects, and some sensors have blurring problems while sensing moving objects.

Although there are limitations for 3D cameras, 3D sensing is still beneficial in agricultural applications. For indoor applications like phenotyping facilities, very accurate depth measurements of plant organs are required to rebuild fine 3D models of plants; In field operations like weed control, 3D information can help not only in improvement of plant recognition and localization (J. Li, 2014) by resolving problems of occlusion, but also in estimation of infection, in order to apply precise amount of chemicals onto the crops (Nielsen, Andersen, Slaughter, & Giles, 2004).

In addition, 3D cameras have not yet reached their full potential: from version to version their resolutions and robustness are increasing while their prices are decreasing such as Kinect 2 developed by Microsoft.

It is clear that 2D images have higher resolution, more detail, and 3D images contain spatial information of plants. As a result, in order to accomplish plant localization and discrimination, two-dimensional textural data and three-dimensional spatial data can be fused together, which takes advantages of both sensor types.

1.4 Research Objectives

The goal of this project was to develop a computer vision based high-throughput crop plant discrimination and localization system for robotic weed control, by fusing 2D color and textural data with 3D spatial data.

The specific objectives were to:

(1) Explore the possibility of accomplishing crop plant candidates' detection and localization accurately against different weed species at different growth stages by using image processing techniques, and then evaluate the performance of the image processing system.

(2) Develop a statistical classification system using supervised machine learning techniques, with features from plant color and morphology as inputs, to improve the crop/weed discrimination accuracy, and identify the most suitable classification method by evaluating the error rates of different classification methods.

1.5 Thesis Overview

In Chapter 2, the methods framework is stated, including the sensor evaluation, data collection and the algorithm design overview.

In Chapter 3, a novel 2D and 3D fusion based crop plant candidate detection system is introduced. The image processing algorithms of preprocessing, segmentation, and clustering are discussed in detail, along with the analysis of candidate detection results.

In Chapter 3, a feature-based crop/weed classification system is developed, using color, morphological and structural features extracted from the image processing procedures. The problem of connecting plants was also considered as solved. Different classification methods are evaluated and compared in this study to select the most suitable one. The classifier's performance is discussed, and the error sources are analyzed.

In Chapter 4, the general conclusion, as well as suggestions for future research are presented.

1.6 References

Alenyà, G., Dellen, B., Foix, S., & Torras, C. (2012). *Robotic Leaf Probing Via Segmentation of Range Data Into Surface Patches.* Paper presented at the IROS Workshop on Agricultural Robotics: Enabling Safe, Efficient, Affordable Robots for Food Production, Vilamoura, Portugal.

Astrand, B., & Baerveldt, A.-J. (2002). An Agricultural Mobile Robot with Vision-Based Perception for Mechanical Weed Control. *Autonomous Robots*, 21-35.

Du, J., Wang, X., & Zhang, G. (2007). Leaf shape based plant species recognition. *Applied mathematics and computation, 185*(2), 883-893.

Ehsani, M., Upadhyaya, S. K., & Mattson, M. L. (2004). Seed location mapping using RTK GPS. *Transactions of the ASAE, 47*(3), 909.

Finney, D. M. e. a. (2008). Weed Management on Organic Farms. *Organic Production*.

Griepentrog, H. W., Nørremark, M., Nielsen, H., & Blackmore, B. (2005). Seed mapping of sugar beet. *Precision Agriculture, 6*(2), 157-165.

Hunt, E. R., Hively, W. D., Fujikawa, S. J., Linden, D. S., Daughtry, C. S., & McCarty, G. W. (2010). Acquisition of NIR-green-blue digital photographs from unmanned aircraft for crop monitoring. *Remote Sensing, 2*(1), 290-305.

Jin, J., & Tang, L. (2009). Corn Plant Sensing Using Real-Time Stereo Vision. *Journal of Field Robotics*, 591–608.

Kazmi, W., Foix, S., Alenyà, G., & Andersen, H. J. (2014). ndoor and outdoor depth imaging of leaves with time-of-flight and stereo vision sensors: Analysis and comparison. *ISPRS Journal of Photogrammetry and Remote Sensing*, 128-146.

Li, J. (2014). *3D machine vision system for robotic weeding and plant phenotyping.* (Dissertation), Iowa State University, http://lib.dr.iastate.edu/cgi/viewcontent.cgi?article=4743&context=etd. (13736)

Li, L. (2014). Time-of-Flight Camera – An Introduction. *Technical White Paper*.

McErlich, A. F., & Boydston, R. A. (2014). Current state of weed management in organic and conventional cropping systems *Automation: The Future of Weed Control in Cropping Systems* (pp. 11-32): Springer.

Nakarmi, A., & Tang, L. (2012). Automatic Inter-plant spacing sensing at early growth stages using a 3D vision sensor. *Computers and Electronics in Agriculture*.

Nielsen, M., Andersen, H. J., Slaughter, D. C., & Giles, D. K. (2004). *Detecting leaf features for automatic weed control using trinocular stereo vision.* Paper presented at the Precision Agriculture and Other Precision Resources Management, Minneapolis, MN, USA.

Sansoni, G., Trebeschi, M., & Docchio, F. (2009). State-of-The-Art and Applications of 3D Imaging Sensors in Industry, Cultural Heritage, Medicine, and Criminal Investigation. *Sensors*, 9, 568-601.

Scotford, I. M., & Miller, P. C. H. (2005). Applications of Spectral Reflectance Techniques in Northern European Cereal Production: A Review. *Biosystems Engineering*, 235-250.

Slaughter, D. C. (2008). Autonomous robotic weed control systems: A review. *computers and electronics in agriculture*.

Sui, R., Thomasson, J. A., Hanks, J., & Wooten, J. (2008). Ground-based sensing system for weed mapping in cotton. *Computers and Electronics in Agriculture*, 31–38.

Sun, H., Slaughter, D. C., Ruiz, M. P., Gliever, C., Upadhyaya, S. K., & Smith, R. F. (2010). RTK GPS mapping of transplanted row crops. *Computers and Electronics in Agriculture*, 32-37.

Surmann, H., & Nüchter, A. (2003). An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, Volume 45, Issues 43–44.

Søgaard, H. T., & Olsen, H. J. (2003). Determination of crop rows by image analysis without segmentation. *Computers and Electronics in Agriculture*, 141-158.

Upadhyaya, M. K., & Blackshaw, R. E. (2007). *Non-chemical Weed Management: Principles, Concepts and Technology*: CABI.

USDA-NASS (Producer). (2014). http://www.nass.usda.gov/Publications/Ag_Statistics/2014/. *2014 Agricultural Statistics Annual*.

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

Zimdahl, R. L. (2013). *Fundamentals of Weed Science*: Elsevier Science.

Zwiggelaar, R. (1998). A review of spectral properties of plants and their potential use for crop/weed discrimination in row-crops. *Crop Protection*, 189-206.

CHAPTER 2. METHODS FRAMEWORK AND DATA COLLECTION

2.1 Introduction

Weed management is important in crop production systems. However, conventional herbicide based weed control can lead to negative environmental impacts. Robotic weeding offers a possibility of controlling weeds precisely, particularly for weeds growing near or within crop rows. In robotic weeding, computer vision is the most competent method when color and shapes need to be analyzed at high speed. In this study, the main purpose was to develop a computer vision based high-throughput plant recognition and localization system for robotic weed control using both 2D and 3D images. In this chapter the research methodology adopted in this research is presented. In more details, the sensor evaluation, the data acquisition, and the image processing algorithm framework are stated.

2.2 Methods Framework

The goal of this project was to develop a computer vision based high-throughput plant recognition and localization system for robotic weed control, by fusing 2D color and textural data with 3D spatial data. The methods framework of the study is listed as follows:

1. Sensor selection and evaluation.
2. Material (field with crop and weed plants) preparation, and data acquisition.
3. Fusion of 2D and 3D based image processing algorithms development.
4. System evaluation based on recognition and localization results.

The rest of this chapter will be organized as follows:

At first, the sensor used in this study was evaluated by specification and depth measurement error level.

Secondly, the target crop species in this study are defined. The sensor view angle and working distance are decided. The field design and the data acquisition apparatus are stated as well.

Lastly, the image processing algorithm framework is introduced, which was designed to accomplish crop/weed discrimination and crop plant localization.

The algorithms developed will be described in detail in Chapter 3 and Chapter 4, and the system evaluation will be introduced in Chapter 4.

2.3 Sensor Evaluation

Kinect v2 sensor (Figure 2-1(a)) was used in this study. It is developed by Microsoft, and provides color (RGB), IR (Infrared) and depth information. The 3D sensor in Kinect v2 is a semiconductor based PMD (photon mixer devices) sensor based on Time-of-Flight principle. The 2D color information can be registered into the 3D space point with sensors' relationship provided in Kinect SDK (Figure 2-1(b)). It offers an easy way to combine the advantages of both 2D color camera's high resolution as well as 3D depth sensor's illuminant-change insensitiveness together. It is widely adopted for various computer vision applications, including 3D reconstruction, object recognition as well as some Human-

Computer Interface applications. The specification of Kinect v2 is listed in Table 2-1. The resolution of depth is 512x424, which is much higher than other commercial depth sensors, for instance, Swiss Ranger. Another advantage of the Kinect v2 sensor is that it uses three strong IR emitters as light sources (Figure 2-2). They enable the Kinect v2 to work outdoors under indirect sunlight, though it is still not operational under direct sunlight.

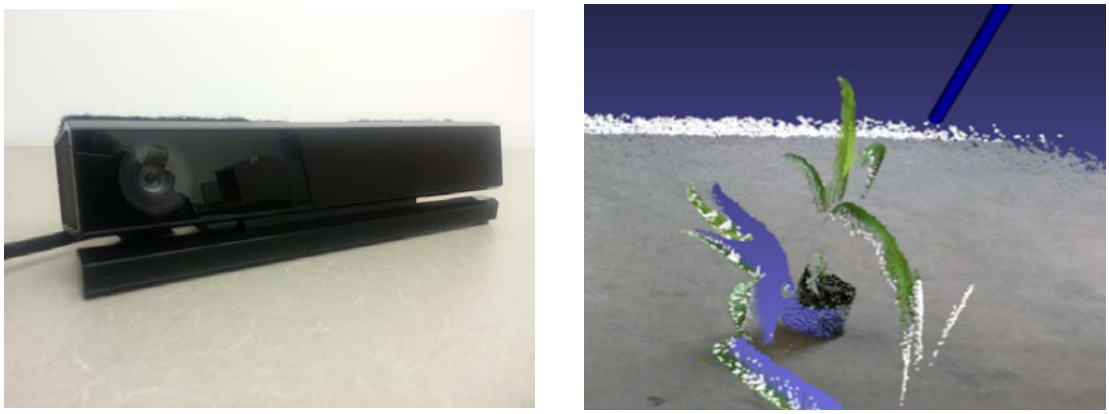


Figure 2-1. (a) Kinect v2 sensor used in this study. (b) The colored point cloud output from fusing 3D depth information and 2D color information.

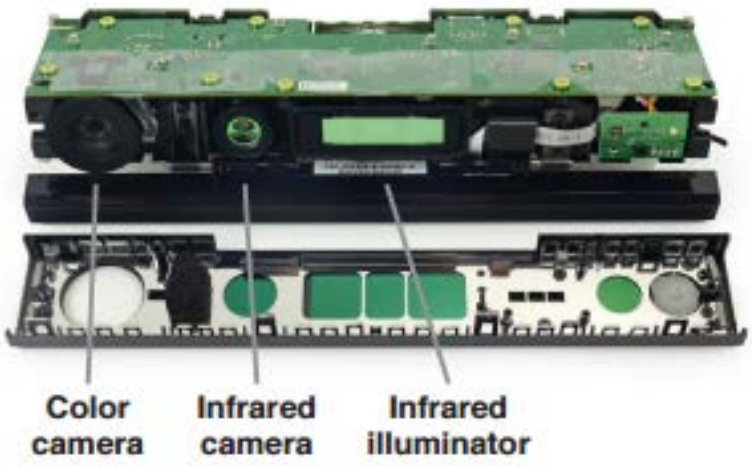


Figure 2-2. Teardown picture of a Kinect v2 sensor. Image was taken from (Fankhauser et al., 2015)

Table 2-1. Kinect v2 Specifications

Infrared/depth camera	Resolution	512 x 424 px
	Field of view (h x v)	70º x 60º
	Operating range	0.5 – 4.5 m
	Depth resolution	1 mm
Color camera	Resolution	1920 x 1080 px
	Field of view (h x v)	84º x 54º
	Color depth	256 bit, 3 channels
Frame rate		30 Hz
Shutter type		Global shutter
Voltage		12 VDC
Power usage		15 W
Dimensions(w x d x h) (mm)		249 x 66 x 67
Mass		970 g
Price		199 USD

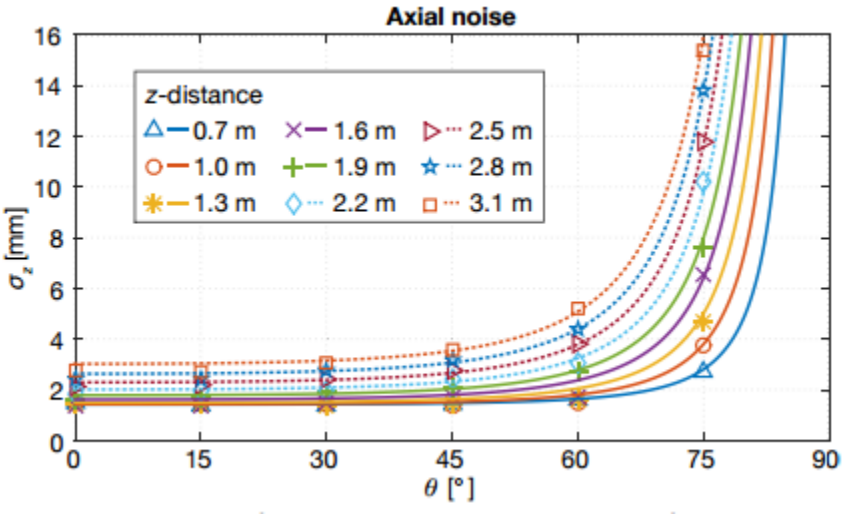
On the other hand, Kinect v2 sensors also have some drawbacks. The most severe one is the minimum range requirement: the sensor can detect nothing within a minimum distance of 0.5 meters. This limits the density of data points on objects, which will increase the difficulty and reduce the accuracy of object detection. Another drawback is delaying, which is not severe, but still influences the performance of real-time applications.

Other than these specifications, accuracy is another important parameter to evaluate a sensor in this project. The error is composed of both non-systematic and systematic error. The non-systematic error is characterized by statistical uncertainty or noise level. A comprehensive report of evaluating the noise level of Kinect v2 was reported by (Fankhauser et al., 2015). The author tested the axial noise σ_z (noise level along the z-axis) of the depth output at different distances and angles of the observed surface. Both indoor and outdoor situations are tested. The results are shown in Figure 2-3. The author also found that the noise

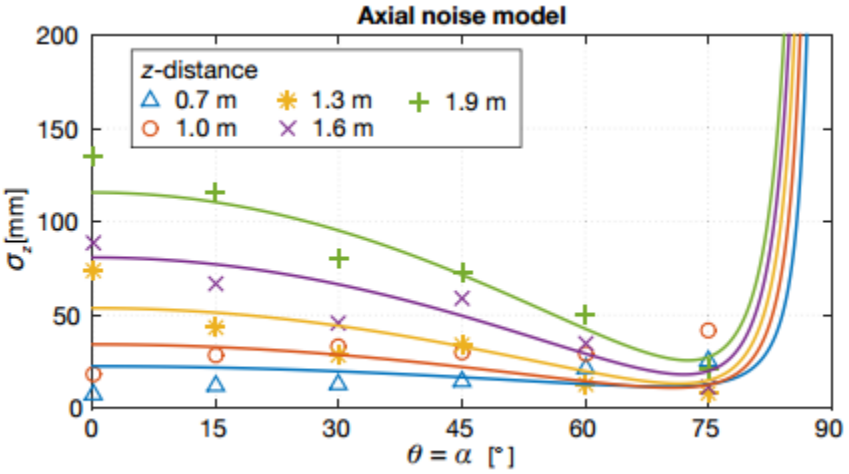
levels are similar when testing indoor and testing outdoors with overcast sky conditions, and the noise level is significant when testing in direct sunlight. For this research project, the distance between the sensor and objects is within 0.75-1.25 m, and a sunshade will be used to block direct sunlight. Thus, the noise level σ_z should be within 4 mm. Thus, it is clear that the sensor is competent to output reliable data under indirect sunlight.

Equally important, systematic error, which is a type of error that deviates by a fixed amount from the true value of measurement, is also evaluated by Fankhauser (2015). The author analyzed the systematic errors such as depth distortion, amplitude-related error, and temperature-related error. The composited systematic error is still in millimeter level when working at a short working distance and perpendicular to the object surface. But it increases with working distance, or angle of the object surface. In addition, Corti et al. (2015) also found that different materials, surfaces, as well as different colors, will result in small offset in depth measurement. Still, it is not significant, which was estimated to be ±1 mm.

Overall, the depth measurement uncertainty of the Kinect v2 sensor should be within 5 mm. Thus, the Kinect v2 sensor was found competent for this project, as a result of its high resolution, tolerance to various illumination conditions, acceptable error level and low price.



(a) Axial noise level in indoor test



(b) Axial noise level in outdoor test

Figure 2-3. Axial noise level (standard deviation of measured depth values) at different working distances (z-distance) and different angles of observed object surface (θ), in both indoor (a) and outdoor (b) tests (Fankhauser et al., 2015). The θ is defined as the angle difference between the direction of sensor and the surface normal.

2.4 Data Collection

2.4.1 Target species

The target crop species in this study includes some common crop species, such as

lettuce (Lactuca, L.), broccoli (Brassica oleracea L. var. botrytis L.), kale (Brassica napus L. var. pabularia (DC.) Rchb.), pak choi (Brassica chinensis L.), tomato (Solanum lycopersicum L.), pepper (Capsicum, L.) and green beans (Phaseolus vulgaris L.). Some sample images are shown in Table 2-2.

The negative input of the system are various types of weeds that are common in Iowa, including bromegrass (Bromus inermis Leyss), pigweed (Amaranthus spp.), Lambsquaters (Chenopodium album), waterhemp (Amaranthus rudis), barnyardgrass (Echinochloa crus-galli), bindweed (Convolvulus arvensis), purslane (Portulaca oleracea), and white clover (Trifolium repens). The example images are shown in Figure 2-4. Some weed species have similar shapes as the target plants; sometimes weed plants are higher than the crops, and sometimes weeds partially occlude the crop canopies.

It can be observed that, plant morphological features such as overall canopy shape, leaf length, leaf shape, and leaf curvature, as well as plant structural features such as venation are ideal for crop/weed discrimination. As they are distinctive between different crop species and weed species.

Table 2-2 Examples of 2D images taken using Kinect v2 sensor from different species at different time, as well as distinctive features that can be observed using Kinect v2 sensor.

Species	Stage 1	Stage 2	Stage 3	Features
Lettuce				Height, Diameter, Color saturation
Broccoli				Height, Diameter, Leaf length, Leaf width, Leaf aspect ratio, Venation
Kale				Height, Diameter, Leaf shape, Crenate margin, Venation
Pak Choi				Height, Diameter, Orbicular leaf, Venation



Figure 2-4. Example images of weeds in this study. From up left to bottom right: bromegrass (Bromus inermis Leyss), pigweed (Amaranthus spp.), Lambsquaters (Chenopodium album), waterhemp (Amaranthus rudis), barnyardgrass (Echinochloa crus-galli), bindweed (Convolvulus arvensis), purslane (Portulaca oleracea), and white clover (Trifolium repens).

2.4.2 View angle and working distance selection

In order to achieve a higher recognition accuracy in identification of different crop species at different growth stages, a suitable sensor setup was needed to be selected. To obtain higher classification accuracy, the features to be used for classification must be observed and extracted with less “distortions”. In this study, the data was collected using cameras. Concerning the simplicity of the mechanical design of our robot, the view angle would not be changed frequently. Thus, the view angle of the camera was selected carefully to acquire the features according to the different crop species.

Table 2-2 lists pictures of some different species at different growth stages, and their features respectively. According to the table, we can see that: most of the distinctive features for crop/weed discrimination are leaf-shape-based, for which it is best to place the sensor

directly above the leaves. At the same time, there are no obvious features that can only be observed from side views for the target crop species in this study (compared to stalk shape for maize). There is an exception for broccoli, in which the side view features (structural) might be observable but only in very early growth stages. After that, those structural features would be blocked by its own leaves. Thus, it was not economical to place a side view only for acquiring those features. As a result,, top view was selected for those species.

For the purpose of making full use of the sensor, analysis was conducted for this project in order to find the best working distance with consideration of spatial resolution and noise level. This was because the sensor should be placed as close as possible for better spatial resolution (pixels per inch) and less noise, in order to collect finer features. However, the working distance should not be too close, because of the sensor's minimum working distance, as well as the field of view. Since the built-in depth sensor has a lower resolution than the color sensor in Kinect v2, and the depth sensor is more critical in this project, the analysis was only conducted for the depth sensor.

The depth sensor has a resolution of 512x424, with angles of view 70 deg x 60 deg. The sensor is required to be placed more than 19.7" (500mm) away from the plants, due to the minimum distance requirement. Thus, the pixel resolution (ppi, pixels per inch) at different working distances are calculated in Table 2-3.

Table 2-3. Field of view and spatial resolution at different working distance for top view

Distance	Field of view in inch (in mm)		Spatial resolution in ppi (in mm/pix)	
	Vertical	Horizontal	Vertical	Horizontal
<19.7” (<500mm)	Not available			
19.7” (500mm)	22.7 (577)	27.6 (701)	18.7 (1.36)	18.5 (1.37)
29.5” (750mm)	34.1 (866)	41.3 (1050)	12.4 (2.04)	12.4 (2.05)
39.4” (1000mm)	45.4 (1154)	55.1 (1400)	9.34 (2.72)	9.30 (2.73)

As the height of plants varies in the field, as well as the ground is uneven, there must be flexibility when selecting the working distance. After all, the height of the sensor H is selected by estimating the average crop height:

$$H = avg(Crop\ height) + 750mm$$

With the selected working distance, we obtained a spatial resolution of 12.4 ppi in both vertical and horizontal directions on observed plants. The noise level is 2mm, referred to noise analysis stated in the previous section. This arrangement allows this sensor to extract most of the desired features of the plants.

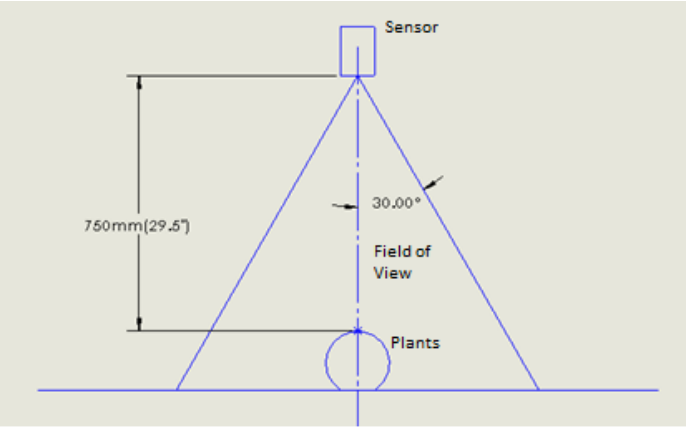


Figure 2-5. The sensor pose and the desired working distance while acquiring data.

2.4.3 Apparatus design and field design for data acquisition

With the analysis of the sensor setup, a simple data collection apparatus was built by

mounting the Kinect v2 sensor and a laptop on a modified cart. The adjustability of the sensor height and view angle are enabled for this system, in order to be compatible with different crop species at different growth stages. By pushing the cart and running the data capturing program, infrared image, depth image and color information were collected and stored for crop plant recognition study.

The data used in this study were obtained in the horticulture research station of the Iowa State University in Story County (42.11 º N, -93.59 º E). The soil is nearly level to moderately sloping landscape near the Skunk River. Soil type was mostly Clarion loam, moderately eroded, with 5 to 9 percent slope. The average annual temperature is 49.45ºF and the average annual precipitation is 35.83 inch.

The field reserved for data acquisition in this study was 1/8 acres. The crop plants were started in a greenhouse and transplanted in the horticulture research station with row spacing of 30 in and plant spacing within the same row of about 12 in. Weeding on the field was not fully performed, in order to make sure crop plants and weeds are observed in the same images.

The image data was collected with Kinect v2 sensor in afternoons of both sunny and cloudy days, in June, July and August. While acquiring data in sunny days, the average illuminance was about 80,000 lux. An umbrella was used to block the sunlight to decrease the illuminance to about 9,000 lux and increase the quality of the images. In cloudy days, the average illuminance was about 35,000 lux, and no umbrella was used. In this project, the data collection cart was pushed at the speed of about 0.3m/s. 3D depth images, as well as 2D color

images, were taken about every two seconds. In total, 2212 images were taken from 546 plants at 4 different times for 7 crop species.

2.5 Algorithm Design Overview

In order to accomplish plants detection and localization, 3D spatial and 2D color information are fused together. The framework of the algorithm is shown in the flowchart (Figure 2-6) and stated as follows:

The algorithm takes a depth image which contains 3D spatial information with corresponding 2D color information acquired through a Kinect v2 sensor as input.

Step 1: Preprocessing. Remove the invalid pixels and remove noise points in point clouds. This procedure is done by using a usable-area filter, a cut-off filter and a simplified neighbor count filter.

Step 2: Registration and segmentation. 2D and 3D information are fused. The ground is detected using both 2D color and 3D depth information, and a plane equation is fitted using RANSAC. All vegetation pixels are extracted in the images.

Step 3: Clustering & candidate extraction. The remaining points which belong to plants are separated into different clusters. Each cluster may contain one plant, or multiple objects which are connected in space, with species not determined. Crop plant candidates can be extracted by simple thresholding on cluster size to shift small objects which can be regarded as non-crop objects.

Step 4: Feature extraction. Canopy morphological and structural features in each crop plant candidate are extracted. If possible, leaves are segmented, and leaf morphological and structural features are extracted.

Step 5: Grouping. The features extracted (venation for broccoli, and canopy shape for lettuce in this study) are used for grouping, in order to separate the pixels within the same candidate into different plants. Locations can be calculated as well.

Step 6: Classification. Crop/weed classification is applied to all the separated plants based on their features (canopy, leaf) using machine learning techniques.

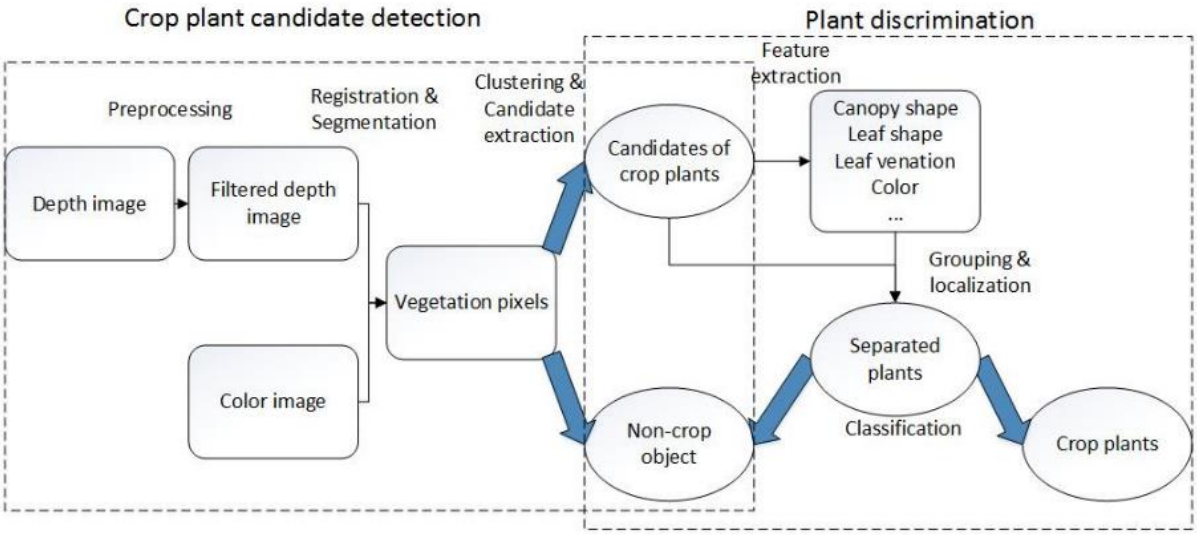


Figure 2-6. The data flowchart of the image processing system

2.6 Reference

Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*: " O'Reilly Media, Inc.".

Corti, A., Giancola, S., Mainetti, G., & Sala, R. (2015). A metrological characterization of the Kinect V2 time-of-flight camera. *Robotics and Autonomous Systems*, Available online

Fankhauser, P., Bloesch, M., Rodriguez, D., Kaestner, R., Hutter, M., & Siegwart, R. (2015). *Kinect v2 for mobile robot navigation: Evaluation and modeling.* Paper presented at the Advanced Robotics (ICAR), 2015 International Conference on.

CHAPTER 3. PLANTS DETECTION, LOCALIZATION AND DISCRIMINATION USING 3D MACHINE VISION FOR ROBOTIC INTRA-ROW WEED CONTROL– PART A: CROP PLANT CANDIDATE DETECTION

3.1 Abstract

Robotic weeding offers a possibility of controlling weeds precisely, particularly for weeds growing near or within crop rows. Computer vision is the most competent method when color and shapes of the plants need to be analyzed at high speed. This paper presents a novel robust crop plant candidate detection algorithm using three-dimensional spatial data fused with two-dimensional textural data to enhance the detection capability. The procedures include filtering data, detecting vegetation pixels, clustering vegetation pixels, then detecting and extracting crop plant candidates using obvious features such as size. The algorithm detects crop plant candidates at different growth stages with disturbances from weeds in different species, under varying illumination conditions. Currently, the method was only developed and tested for broccoli and lettuce. The algorithm was developed and tested on data collected at different growth stages under different illumination conditions (sunny and cloudy), and also with weeds of different species. During the test, 98% of target crop plants were detected as crop plant candidates (true positive rate). Among the candidates, about 66% of them are individual target crop plants or crop plants connected with other plants (precision), and 34% of them are weeds (false discovery rate).

3.2 Introduction

In recent years, with the development of greater health consciousness, consumers are becoming more interested in vegetables, especially natural, organic vegetables. One important factor that affects crop yield is weed competition. It is also one of the most costly operations in vegetable production, especially for organic farming.

With the advancement of computational technology, automated robotic weeding offers a possibility of controlling weeds in a precise fashion, particularly for weeds growing near crops or within crop rows (Slaughter, 2008). Since identification and localization of plants have not yet been fully automated, research to address these problems is thus in great demand.

Computer vision has been shown to provide an option in the inspection of agricultural products, particularly when color and shape need to be analyzed at high speed. Many applications in agricultural robotics such as plants discrimination and self-guidance can be realized with the power of computer vision (Astrand & Baerveldt, 2002). Plant morphology and structure have been focused on, which remains one of the most consistent methods of plants identification (Du et al., 2007).

With sensing method as criteria, most of those computer vision systems can be categorized into two classes: tow-dimensional (2-D) vision systems and three-dimensional (3-D) vision systems.

2D vision applications:

A 2D sensor is a type of sensor that can record light or other electromagnetic radiation reflected or emitted from objects, by focusing it on a light-sensitive surface. The word “2D” means horizontal and vertical dimensions in the image space, as it is a projection of the 3D real world.

Regular 2D cameras were the earliest ones used in robotic weeding. The problem of plant identification was addressed by extracting color or morphological features from a leaf or a whole plant, such as length, width, perimeter dimensions, roundness, circularity, convexity and moment. Slaughter (2008) reviewed these types of systems and concluded that they demonstrated high recognition accuracy only under ideal conditions, in which light was controlled and plants were sparse in images. Also, they are not robust to occlusion problems, or defects of the plants caused by insect damage or the wind, which are common in the field. Moreover, with complex illumination conditions, such as strong sunlight, images will be saturated. Most of the algorithms will fail to segment plants out of the background.

Spectral reflectance characteristics of plants were reported to be effective in vegetation segmentation and crop/weed discrimination (Scotford & Miller, 2005; Zwiggelaar, 1998). Zwiggelaar (1998) also reported that the selected spectral wavebands for classification are generally different for different weed and crop pairs. Therefore, selecting wavebands and designing algorithms for distinguishing crop plants from different weed species is complex.

On the other hand, an NDVI (normalized difference vegetation index) map, which can be generated from an image from an infrared camera or NDVI sensor, is reported to be effective in vegetation segmentation. Sui et al., (2008) have developed a vision-based system for weed mapping using an NDVI camera. However, in order to accomplish crop/weed discrimination with NDVI images, morphological features are still needed. The advantages of spectral reflectance based methods are: they are less sensitive to environmental light, and the infrared reflectance of plants can include additional information for plant discrimination.

3D sensing applications:

A 3D sensor is a type of sensor that can measure the distances between objects and sensors, often referred to as “depth”. The word “3D” corresponds to the three dimensions of the real world. In recent research, with the development of sensing technology, a 3D sensor promises to address problems in 2D vision systems, such as occlusion. A 3D sensor can also give reliable information to perform plant discrimination and plant localization. In several studies (J. Li, 2014; Jin & Tang, 2009; Nakarmi & Tang, 2012), 3D sensors were applied in agricultural applications which provided a good performance. The advantages of 3D sensors for plant discrimination and localization are obvious: 3D sensors can provide fundamental depth information, making it is much easier to obtain the 3D structural and morphological data of the plants.

Today, three types of state-of-art 3D sensors are mainly used on mobile robots: stereo vision, laser, and PMD time-of-flight. In several articles (Sansoni et al., 2009; Weiss & Biber,

2011), the authors have compared and evaluated those three types of sensor for use on mobile robots.

To receive 3D data using stereo vision, typically triangulation of two cameras or structure-from-motion technique are used. In the study by Jin & Tang (2009), a real-time sensing system for corn phenotyping was developed based on a stereo vision sensor. However, due to the stereo camera's passive operation mode, it is hard to provide reliable data for accurate sensing: When receiving 3D data from stereo vision, the disparity calculation highly depends on the structures or features of the objects in images. Further, the precision and maximum depth are limited by the baseline between the cameras. Also, qualities of distance values decrease very quickly as depth increases. The advantage of stereo vision is its high image resolution, with color information. One the other hand, cameras in stereo vision systems can also be modified into near-infrared cameras by replacing their filters with NIR filters, in order to take advantage of spectral reflectance information (Hunt et al., 2010).

The accuracy, resolution, frame rate as well as price of different 3D laser sensors are widespread. Some of the 3D laser sensors are implemented from 2D laser scanners which uses the line-scanning method, such as Lidar. One example is Kurt3D (Surmann & Nüchter, 2003) which equips a rotating 2D SICK laser sensor to realize 3D laser scanning. However, it needs a stop-and-go mode for traveling to receive consistent 3D data. Generally, 3D laser sensors have the properties of high weight, high power consumption, as well as high price. Those make those 3D sensors less desirable for small mobile robots.

The semiconductor based PMD time-of-flight (TOF) camera is the latest technique. It measures distance and infrared reflectance intensity information based on time-of-flight technique. A modulated light signal is emitted by the sensor, reflected by objects then received by sensor receiver (L. Li, 2014). The distances are calculated by the phase shift of the signal as well as the reflection intensity. There are many research applications using TOF cameras as sensing devices in agriculture, such as phenotyping (Alenyà et al., 2012), and plant spacing (Jin & Tang, 2009). In a study by Kazmi et al. (2014), the authors summarized the advantages along with drawbacks of TOF cameras. The advantages include: they deliver high frame rates as well as accurate depth data under suitable conditions. The limitations are: resolutions of depth images are often low; the sensors are sensitive to ambient sunlight, which usually leads to poor performance while working outdoors; the quality of depth values depends on the color of objects, and some sensors have blurring problems while sensing moving objects.

Although there are limitations for 3D cameras, 3D sensing is still beneficial in agricultural applications. For indoor applications like phenotyping facilities, very accurate depth measurements of plant organs are required to rebuild fine 3D models of plants; In field operations like weed control, 3D information can help not only in improvement of plant recognition and localization (J. Li, 2014) by resolving problems of occlusion, but also in estimation of infection, in order to apply precise amount of chemicals onto the crops (Nielsen et al., 2004).

It is clear that 2D images have higher resolution, more detail, and 3D images contain

spatial information of plants. As a result, in order to accomplish plant localization and discrimination, two-dimensional textural data and three-dimensional spatial data can be fused together, which takes advantages of both sensor types.

In this research, we focused on the development and implementation of a time-optimal crop plants detection and localization system based on machine vision for robotic intra-row weed control. In contrast to the most common technologies in crop plants detection based on machine vision, the fusion of 2D color images and 3D depth images are used. Thus, the methods proposed can benefit from the advantages of both types of sensors. In this paper, a crop-plant-candidate detection algorithm was developed and implemented with the fusion of 2D and 3D images, which includes preprocessing, color registration, vegetation pixels segmentation and pixels clustering. The performance was evaluated on dataset collected from broccoli and lettuce under different illuminantion conditions.

This paper is arranged as follows. Section 3.3 introduces the sensor used in this study with its accuracy evaluation, as well as the data acquisition location and agronomic trial design. Section 3.4 describes the proposed method for crop plant candidate detection using both color and spatial information, while Section 3.5 presents the experimental results and discussion. Finally, a summary is presented. A companion paper sets out the subsequent component of crop plant detection and localization, which solves problem of connected plants and classify plants using machine learning methods, based on the candidate detection as presented in this current paper.

3.3 Materials

3.3.1 Sensor

Kinect v2 sensor (Figure 3-1 (a)) was used in this study, which provides RGB, IR (Infrared) together with depth information. The 3D sensor in Kinect v2 is a semiconductor based PMD (photon mixer devices) sensor based on Time-of-Flight principle. The 2D color information can be registered to the 3D space point with sensor's relationship provided in Kinect SDK (Figure 3-1 (c)). It offers an easy way to combine the advantages of both 2D color camera's high resolution as well as 3D depth sensor's light changing insensitiveness together. The Kinect v2 sensor uses three strong IR emitters as light sources (Figure 3-1 (b)). They enable the Kinect v2 to work outdoors, with indirect sunlight. The resolution of depth is 512x424, and the depth measurement accuracy is within 5 mm, when working under indirect sunlight outdoors.

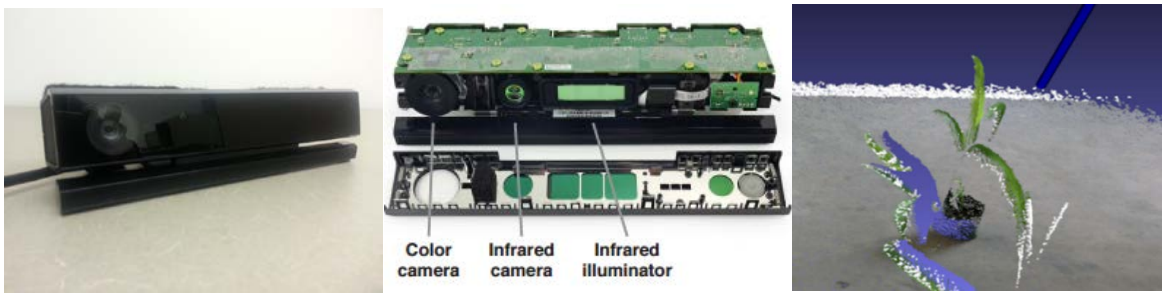


Figure 3-1. (a) Kinect v2 sensor used in this study. (b) Teardown picture of a Kinect v2 sensor (Fankhauser et al., 2015). (c) Example 3D point cloud registered with color.

3.3.2 Location and agronomic trial

The data used in this study were obtained in the horticulture research station of the Iowa State University in Story County (42.11 º N, -93.59 º E). The soil is nearly level to

moderately sloping landscape near the Skunk River. Soil type was mostly Clarion loam, moderately eroded, with 5 to 9 percent slope. The average annual temperature is 49.45 ºF and the average annual precipitation is 35.83 inch.

The field reserved for acquisition in this study was 1/8 acres. The crop plants were started in greenhouses and transplanted in the horticulture research station with row spacing of 30 in and plant spacing within the same row of about 12 in. Weeds were controlled manually in the field with some weeds reserved, in order to make sure crop plants and weeds are observed in the same images.

The image data was collected with Kinect v2 sensor in afternoons on both sunny and cloudy days, in June, July and August. While acquiring data in sunny days, the average illuminance was about 80,000 lux. An umbrella was used to block the sunlight to decrease the illuminance to about 9,000 lux and increase the quality of the images. In cloudy days, the average illuminance was about 35,000 lux, and no umbrella was used. In this project, the data collection cart was pushed at the speed of about 0.3m/s between crop rows. 3D depth images, as well as 2D color images, were taken about every two seconds. In total, 203 images were taken from 49 broccoli plants and 376 images were taken from 97 lettuce plants at four different times.

3.4 Algorithm Design

In order to accomplish plant detection and localization, 3D spatial and 2D color

information were fused together. The framework of the algorithm is shown in the flowchart (Figure 3-2) and stated as follows:

The algorithm takes a depth image which contains 3D spatial information with corresponding 2D color information acquired through a Kinect v2 sensor as input.

Step 1: Preprocessing. Remove the invalid pixels and remove noise points in point clouds. This procedure is done by using a usable-area filter, a cut-off filter and a simplified neighbor count filter.

Step 2: Registration and segmentation. 2D and 3D information are fused. The ground is detected using both 2D color and 3D depth information, and a plane equation is fitted using RANSAC. All vegetation pixels are extracted in the images.

Step 3: Clustering & candidate extraction. The remaining points which belong to plants are separated into different clusters. Each cluster may contain one plant, or multiple objects which are connected in space, with species not determined. Crop plant candidates can be extracted by simple thresholding on cluster size to shift small objects which can be regarded as non-crop objects.

Step 4: Feature extraction. Canopy morphological and structural features in each crop plant candidate are extracted. If possible, leaves are segmented as well, and leaf morphological and structural features are extracted as well.

Step 5: Grouping. The features extracted (venation for broccoli, and canopy shape for lettuce in this study) are used for grouping, in order to separate the pixels within the same candidate detected into different plants. Locations can be calculated as well.

Step 6: Classification. Crop/weed classification is applied to all the separated plants based on their features (canopy, leaf) using machine learning techniques.

The image processing steps (step 1-5) were developed using OPENCV library, C++ (Bradski & Kaehler, 2008), and the classification step (step 6) was developed using R language (R Development Core Team, 2010). In this section, step 1, 2, 3 are used to accomplish crop plant candidate detection. Step 4, 5, 6 are used for plant discrimination, which will be introduced in the companion paper.

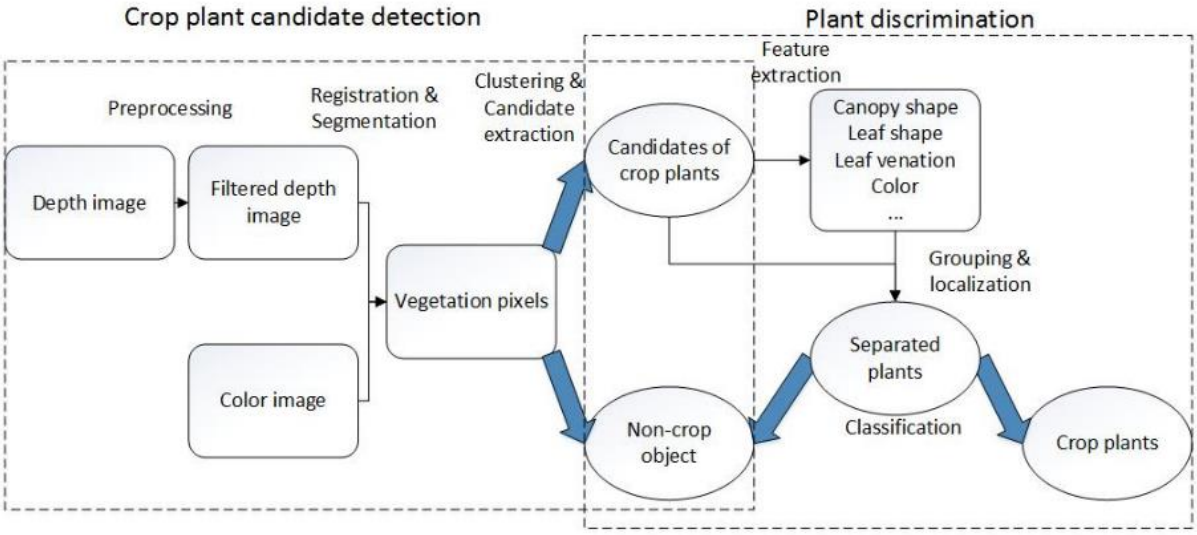


Figure 3-2. The data flowchart of the image processing system

3.4.1 Preprocessing on depth image

In general, data collected by optical sensors contains noise, which will cause unexpected results if left unfiltered and uncorrected. This is also true for Kinect v2 depth sensor, especially when working outdoors. Within preprocessing, useful data (in contrast to sparse noise & sensor bad points) is extracted from the depth image, against the disturbances

from noise and useless information (such as the wheels of the data collection cart). In our algorithm, three simple filters are applied on the depth image:

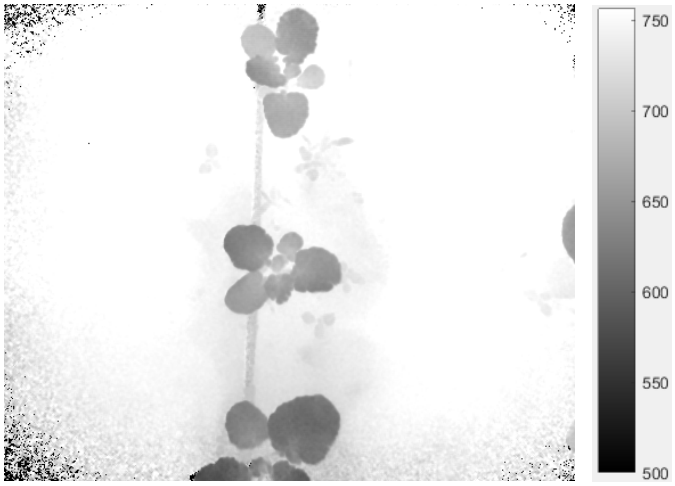


Figure 3-3. An example of depth image indicating the z-direction distance acquired using a Kinect v2 sensor. Noise level is higher in the non-center area, especially at corners. Unit in mm.

3.7.2.1 Useable-area filter:

The first filter to apply is a reliable area filter. Because of the ambient light, the off-center pixels in depth images of Kinect v2 sensor have a higher chance of carrying incorrect depth information. For instance, Figure 3-3 shows a depth image of broccoli crops collected by Kinect v2 sensor, in overcast sky conditions. Noise (pixels similar to salt-and-pepper noise) can be observed in the off-center area, especially at the corners. Because the off-center pixels have a higher angle difference than center pixels, which will result in higher noise levels. It is also a result of the sensor structure as well. In this study, a round area with a radius of 220 pixels was selected as reliable area by observing and testing. In this process, about 30% of the points are deleted. This operation can be expressed as:

$$dst(x,y) = \begin{cases} src(x,y) & if\ ||(x,y) - (256,212)|| < 220 \\ 0 & otherwise \end{cases}$$

where (x, y) is the position of a pixel in depth image in (column, row), and (256, 212) is the principal point of the camera or the center pixel of the depth image.

3.7.2.2 Depth cut-off filter:

The second filter applied is a cut-off filter in depth images, which is used to remove pixels laying outside of a predefined depth range. Bad points from the sensor (usually have a distance of zero or infinity) and points with significant depth value (noise in most cases) are removed. In this study, the distance upper limit was selected to be the sensor height + 200 mm, to ensure the ground can be observed by the sensor. And the lower limit is 500 mm, which is also the minimum working distance of Kinect v2. The operation can be expressed as:

$$dst(x,y) = \begin{cases} src(x,y) & if\ 500 < src(x,y) < h + 200 \\ 0 & otherwise \end{cases} \quad (1)$$

where (x, y) is the position of a pixel, and h is the sensor height when taking this image.

3.7.2.3 Neighbor count filter:

Finally, a simplified neighbor count filter was modified from Statistical Outlier Filter developed by (Rusu, 2009) to remove sparse noise. The objects of interest (such as plants and soil surface) have flat surfaces. For depth sensors such as Kinect v2, the pixels are denser on flat surfaces than on uneven surfaces. In other words, the pixel on a flat surface has more surrounding pixels in 3D space. Those “connected” pixels are called “neighborhood” in 3D point cloud, which is similar to the concept of “connectivity” in 2D image processing. The

neighborhood of a pixel *P* in pixels set *G* in Euclidian space *E* can be defined as an induced subset of *G* consisting of all pixels whose Euclidian distance *d* to *P* is:

a. Shorter than defined value *r*, or

b. one of the *k* shortest.

The criteria (a) is called *RNS* (radius neighbor search), and the one using criteria (b) is called *KNN* (k-nearest neighbor) search. The most general method to search neighbors in a 3D point cloud is the k-d tree method (Cormen & Leiserson). Most of the neighborhood searching algorithms implemented in Point Cloud Library (PCL) (("Point Cloud Library (PCL)," 2015) are using k-d trees. K-d tree is a binary tree in which every node is a k-dimensional point. At every non-leaf node, there is a virtual splitting hyperplane that divides the space into two parts. Constructing a k-d tree has a worst complexity of $O(kn\ log\ n)$.

By using the tree properties of the k-d tree, the neighbor search can be done very efficiently without large portions of the search space. The complexity of the searching algorithm is $O(log\ n)$, which is much less than searching by comparing all the distances to each point.

Although k-d tree, as well as other searching algorithms such as Octree, bd-tree, can greatly improve the neighborhood searching speed, it is still a heavy burden for the CPU to search neighbor points for every point in a point cloud in this study. However, the problem can be simplified in this application. As the output of the depth sensor of Kinect v2 is well organized in rows and columns, the concept of “connectivity” in 2D images can be applied in this case. The potential neighbors can only be the surrounding pixels on depth image.

Thus, in this neighbor searching algorithm, the searching area is limited to a 5 x 5 window. For point P_{ij}, the searching is limited to a point set

$$P \in \{P_{xy} | i-2 \leq x \leq i+2, j-2 \leq y \leq j+2\} \quad (2)$$

As discussed in Section 2.4, the estimated distance between two adjacent pixels is 2 mm if the surface is flat and facing to the sensor. If the surface angle is 60 degrees, the distance between pixels becomes 4 mm, where the noise level is ±3 mm. Then 10 mm can be defined as the limit of the distance of deciding neighborhood, and that is sufficient even considering the varying surface angles of the plants and measurement uncertainty.

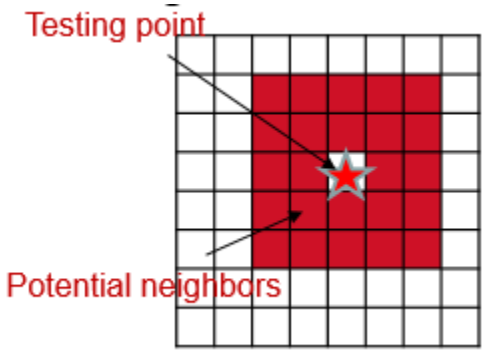


Figure 3-4. The strategy of the neighbor count filter.

Instead of calculating the Euclidean distance *d* in 3D Cartesian space in *L2* norm form, the square distance d^2 can be used for judgment by comparing d^2 with 100 mm^2. This method was proposed in other studies as well (J. Li, 2014; Rusu, 2009), as the square root operations are computationally expensive. However, in this study, the distance d was simplified as the depth value difference between two pixels:

$$d_{ij} = abs(src(x_i, y_i) - src(x_j, y_j)) \quad (3)$$

After testing, the result is similar to using the *L2* norm distance. This simplified

definition of distance reduced three minus and three multiplication operations to one single minus operation. In addition, it also eliminates the large computation cost of calculating 3D point cloud from the depth image. OpenMP parallel computing technique was also used for optimization. Therefore, the efficiency increased greatly in this neighborhood searching procedure.

The neighbor count filter first calculates neighbors for each pixel, then removes the pixel with fewer neighbors than the threshold (Figure 3-4). The filtering operation can be expressed as:

$$dst(x,y) = \begin{cases} src(x,y) & if\ RNS(x,y) > 15 \\ 0 & otherwise \end{cases} \quad (4)$$

After evaluating the resultant point cloud with different thresholds c, c = 15 was selected to be the most reasonable value. The results before and after filtering are shown in Figure 3-6. The time consumption of filtering using a). The proposed method, b). Method stated by J. Li, (2014), and c). Method implemented in PCL are also compared respectively in Figure 3-5:

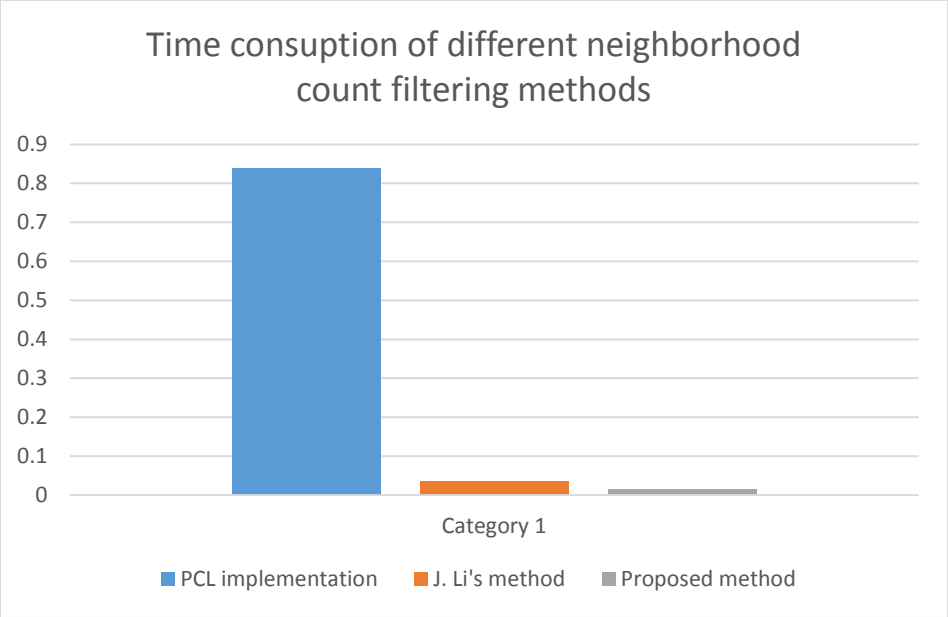
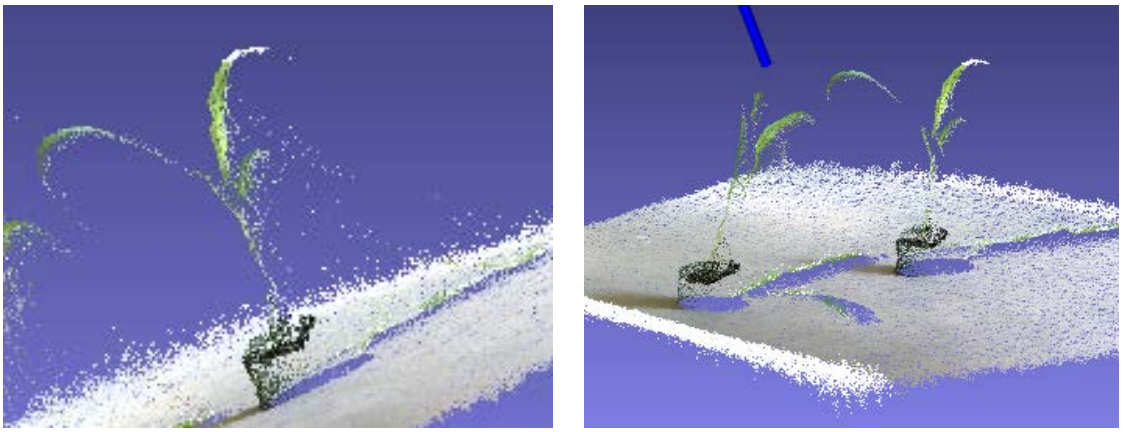


Figure 3-5. The performance shift of the proposed method compared with methods implemented in PCL and stated by J. Li, (2014). Time measured in seconds.



(a) Point cloud with sparse noise (b) Point cloud after filtering

Figure 3-6. Corn plant example showing the differences before (a) and after (b) preprocessing. The point cloud is generated from corn crops in the laboratory. After applying preprocessing procedure, spare noise and bad points are removed.

3.4.2 Segmentation using depth and color

Color registration is to assign colors to their corresponding pixels in depth images.

With this procedure, RGBD images and colored point clouds can be generated. Color registration also unifies the data structure then simplifies the program. At the same time, color registration reduces the memory storage cost. The color registration is based on the intrinsic parameters of the Kinect v2 sensor calibrated by the manufacturer, and the registration function was provided in Kinect v2 SDK provided by the manufacturer.

After preprocessing and color registration, the next step is segmentation. In segmentation, the pixels are divided into two different subsets: one is the vegetation pixels set, and the other is the background pixels set. Both depth and color information are utilized in this procedure, as biomass pixels are higher in green color and higher in depth, while background pixels are darker in color and lower in depth.

3.4.2.1 Color based segmentation

In color based segmentation, the main strategy is to find the green pixels. Segmentation using only RGB channels is unstable. The complexity of illumination conditions such as shadows will challenge the color-based segmentation algorithms. However, illuminant invariant space and HSV color space are found useful in extracting green pixels, which are less sensitive to illumination conditions.

Finlayson et al. (2002) proposed a method using illuminant invariant (ill-inv) space to remove the shadows. The theory is built on the assumptions of narrow-band color sensors, Lambertian Surface (on which Lambert’s cosine law is obeyed), and Planckian Light (Color Temperature). It transfers the color of each pixel from RGB color space into ill-inv color

space, where the Y axis is $\log(B/R)$, and the X-axis is $\log(G/R)$. In ill-inv space, the same color under different light conditions forms a line, and all lines of different colors are approximately parallel. If project those ill-inv color space points onto a specific line, which is called invariant axis, all different colors can be separable. (Figure 3-7)

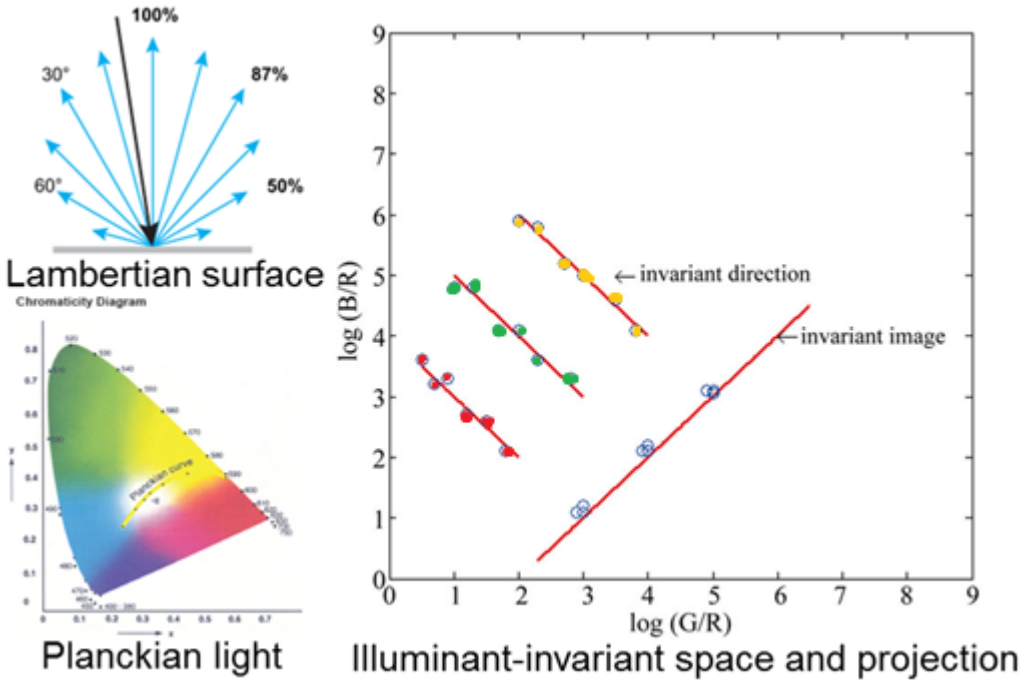


Figure 3-7. The upper leaf and bottom leaf figures show the assumptions of the illuminant-invariant method. The right figure shows the ill-inv space and projection of different color under different illuminations. It is clear that same color under different light conditions lies a specific line. Different colors are on different lines. Thus different colors can be separated.

The invariant axis is an internal parameter of a color sensor, and can be obtained by using the calibration method provided in the study of Alvarez (2008). In this study, the invariant axis of Kinect v2 color sensor was calibrated to be a line passing through the origin with a slope of 40 degrees.

After projecting colors of pixels onto the invariant axis, an illuminant-invariant image

I_{il} can be generated and saved to assistant the segmentation procedure. One example of the illuminant invariant image is shown in Figure 3-8(b), which reduced the effect of shadow compared with the original color image Figure 3-8(a).

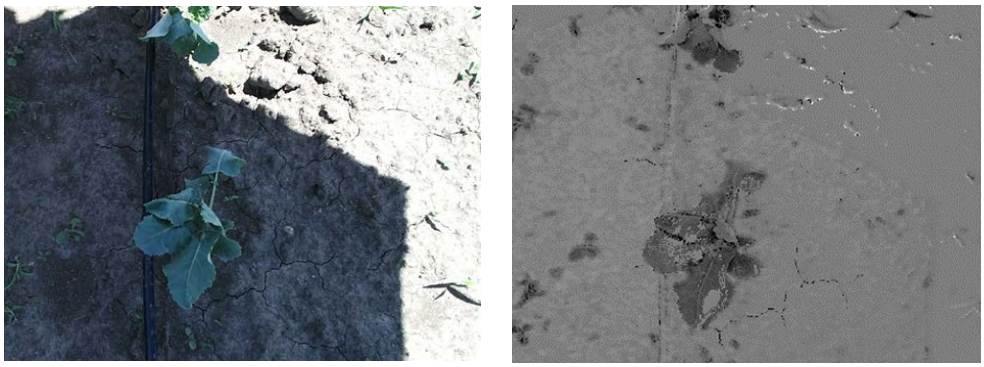


Figure 3-8: The effect of illuminant-invariant map. In the left figure, the light conditions are different between shaded and unshaded area. And in the right figure, the illuminant-invariant map is generated and compensates the effect of changing light conditions. Green pixels have lower values in illuminant-invariant maps.

Philipp & Rath, (2002) found the HSV (Hue-Saturation-Value) color space to be one of the most reliable color space to distinguish green plants from the background. The hue values (H channel) of plants don't change significantly with different light intensity. Thus, the HSV color space image I_{HSV} is also stored for the next segmentation step.

3.4.2.2 Depth based segmentation

In depth-based segmentation, the strategy of the algorithm is to identify the ground, which can be assumed as a plane. The ground identification is based on depth information, as well as the color information extracted in the previous step. The ground is visible in images because the sensor is looking down at the plants. Ideally, the individual plants could be separated after fitting the ground plane and eliminating the ground pixels.

In order to fit the ground plane, the weighted Random Sample Consensus (RANSAC)

algorithm, as well as the robust regression are used. RANSAC is used as a rough ground fitting method, and the robust regression is used for refining the ground plane model.

In this study, the homogeneous coordinate is used to describe points (can be transferred from the depth image) and planes. A finite point $(x, y, z) \in \mathbb{R}^3$ in Cartesian space has homogeneous coordinate $p = t(x, y, z, 1), t \neq 0$. A plane in homogeneous coordinate has the form of $n = r(a, b, c, d), r \neq 0$, corresponding to the general plane equation $ax + by + cz + d = 0$. Then in order for p to lie on the plane n, the dot product of p and n must vanish. That is,

$$p \cdot n = 0 \tag{5}$$

Then the perpendicular distance dis from an arbitrary normalized point $p_0 = (x, y, z, 1)$ to a normalized plane $n = (a, b, c, d)$ can be obtained by

$$dis = p_0 \cdot n \tag{6}$$

However, the general RANSAC evaluates the resultant model of each iteration by only counting inliers based on Euclidean distances. In some cases, the general RANSAC algorithm will fail. For instance, when weeds cover the ground and have height close to the target crops, the general RANSAC will result in an incorrect ground plane. Thus, it is necessary and possible to improve the general RANSAC algorithm with additional information.

In this study, the general RANSAC was modified to make use of the color information. The definition of distance $\widehat{d_i}$ was modified to a distance function $f(d_i, c_i)$, in which d_i is the perpendicular distance between each point p_i and the plane n. And c_i is

the corresponding color. The pseudo code is listed in Figure 3-9:

Algorithm: RANSAC Ground detection

Input: Color registered point cloud $(P, C), in\ which\ P, C\ are\ bijectional$
Output: The best fit model $\hat{n}$, inlier point set G, outlier point set $\bar{G}$
Procedure:

1. Set iteration count $t = 0$
2. While $t < k$, the minimum iteration count requirement
3. Select randomly 3 points $p_j (j = 1,2,3)$ to determine the ground plane parameters n_t.
4. Calculate the distance form each point to the plane $\widehat{d_i} = f\left[d(p_i, n_t), c_i\right], p_i \in P, c_i \in C$
5. Evaluate the cost of the estimated model $C_t = \sum \widehat{c_i}$
6. Find the minimum C_t
7. Extract inlier points G based on criteria $\widehat{d_i} < threshold$
8. Output the best ground plane model $\hat{n}$

Figure 3-9. The RANSAC detection algorithm

In this study, with the consideration of the program's simplicity, the distance definition is determined to be:

$$\widehat{d_i} = f(d_i, c_i) = w(c_i) * d(p_i, n_t), \tag{7}$$

in which the color information acts as a weight factor of the distance function, indicating the importance of each pixel in fitting the ground. The determinant of the weight function $w(c_i)$ is stated in the next section. The cost $\widehat{c_i}$ of each pixel is defined as a piecewise function

$$\widehat{c_i} = \begin{cases} w(c_i) & if\ abs(p_i \cdot n) > 3 * stddev(p \cdot n) \\ 0 & else \end{cases}, \tag{8}$$

In which "inliers" (points close to the ground) have zero cost, and "outliers" (points far from the ground) have costs equal to their weight.

After RANSAC, the refinement of the plane is done by using robust regression. Robust regression is a variant of the least square regression method, which is developed for data are contaminated with outliers. It is also one kind of weighted least square, in which the weight of each data is negatively correlated to its residual. This algorithm may also be

iterative, and the weight of each data point is updated in each iteration.

In this study, the robust regression is used to solve minimization problem

$$argmin_n \sum {d_i}^2 = argmin_n \sum (w_i * dis(p_i \cdot n))^2, \quad (9)$$

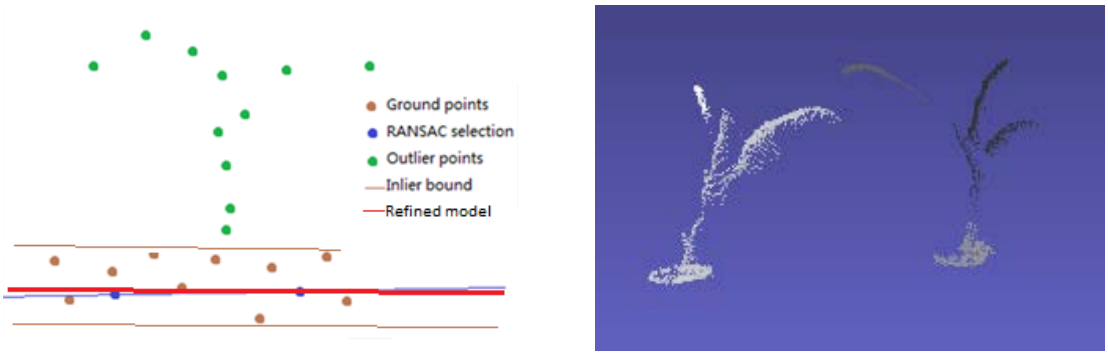
in which $p_i \in P$ for each point in the point set, and w_i is its weight. This least square problem can also be expressed by solving an over constrained equation

$$\begin{pmatrix} w_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_n \end{pmatrix}_{n*n} \begin{pmatrix} p_1^T \\ \cdots \\ p_n^T \end{pmatrix}_{n*4} n_{4*1} = 0. \quad (10)$$

This over constrained equation can be solved by using SVD (singular-value-decomposition) method. In addition, in this case, the weight function is defined as a piecewise function, which is based on distance, color, as well as the weight from the latest iteration

$$w_i = \begin{cases} w(c_i) & if\ abs(w_i p_i \cdot n) < 3 * stddev(wp \cdot n) \\ 0 & else \end{cases}. \quad (11)$$

In the equation above, $w(c_i)$ is the weight function of the corresponding color information of the pixel, which is also used in RANSAC. Thus, in this equation, the “inliers” have weights equal to their weight used before, and “outliers” have zero weight. The iteration continues until the result no longer changes or the iterations exceed a certain number. The best fit ground plane model n is obtained by then.



(a) Principle for ground detection (b) Result after ground removal

Figure 3-10. Fig (a) shows a synthetic 2D point cloud of a plant with a ground plane to visualize the principle of the ground detection. With RANSAC algorithm, two points are randomly selected (blue), and a line is fitted using these points (blue line). Then the model is refined using robust linear regression (red line). The points between the brown lines become inlier standing the ground. Fig (b) shows a point cloud with only outliers after ground detection. The points of the ground are successfully removed.

After ground detection, the outliers with weight $w = 0$ are considered as biomass pixels. Then, those outlier pixels are extracted into plant data point set P_t. At the same time, most of the weeds that are relatively short are excluded as background as well.

3.4.2.3 Weight function determination

When fitting the ground plane model, it is not reliable to use the depth information only. For instance, when the most of the ground was covered by weed, a tilted plane model would be produced. In this study, in order to solve such scenarios, weights that are calculated by a function of color information are assigned to those pixels when fitting the ground, which assists the ground fitting algorithm by telling the importance of each pixel.

Since pixels belonging to the ground should be close to the fitted ground plane, they should have higher weights while performing fitting. In contrast, the plant pixels are more likely to be the outliers when fitting the ground, then they should have lower weights. One

solution is to define the weight function to be:

$$Weight(p) = 1 + P(p \in Ground), \tag{12}$$

in which p is one pixel in the point cloud, and the $P(p \in Ground)$ means the probability of p is belonging to the ground pixel set.

The probability function should transform $(-\infty, +\infty)$ to $[0,1]$. In this study, the model used is:

$$P[Y = 1|X = x_i] = \frac{e^{(\beta_0 + \beta_1 X)}}{1 + e^{(\beta_0 + \beta_1 X)}} = \frac{1}{1 + \exp(-\beta_0 - \beta_1 x_i)} \tag{13}$$

In the equation above, β_0 and β_1 are the coefficients of the model, and X is the variable vector. In this study, considering the computational cost of the program, X is selected from values in three color spaces: RGB, HSV, or Illuminant variant.

The properties of logistic response function are obvious: it is either monotonic increasing or decreasing depending on the sign of β_1; it gradually approaches 0 and 1; and it is almost linear in the middle section. Similar to linear regression which determines the parameters of a linear model, the method to fit the parameters of this logistic response function is called logistic regression, which is one of machine learning methods for classification. The advantages of the logistic regression methods are:

1. Logistic response function model is based on maximum likelihood estimation theory, and it doesn't rely on any assumptions (e.g. normally distributed error, and equal variance of different parameters) of data, except for i.i.d (independent, identically distributed).
2. Relatively low computational cost in fitting the model, as well as calculating the

probability.

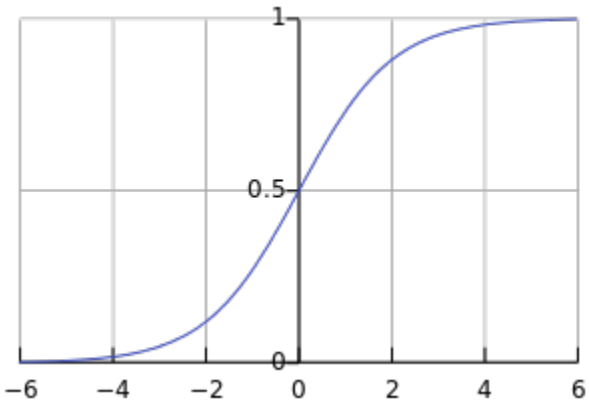


Figure 3-11. The standard logistic response function curve with one variable, with $\beta_0 = \beta_1 = 1$, which transforms $(-\infty, +\infty)$ to $[0,1]$.

In order to fit the logistic response function, a binomial distributed dataset was generated by manually separate the image pixels into two categories: ground pixels (with label 1) and plant pixels (with label 0). After that, three models with different variable options (RGB, HSV, and Illuminant-invariant) are fitted for ground detection with different crop species.

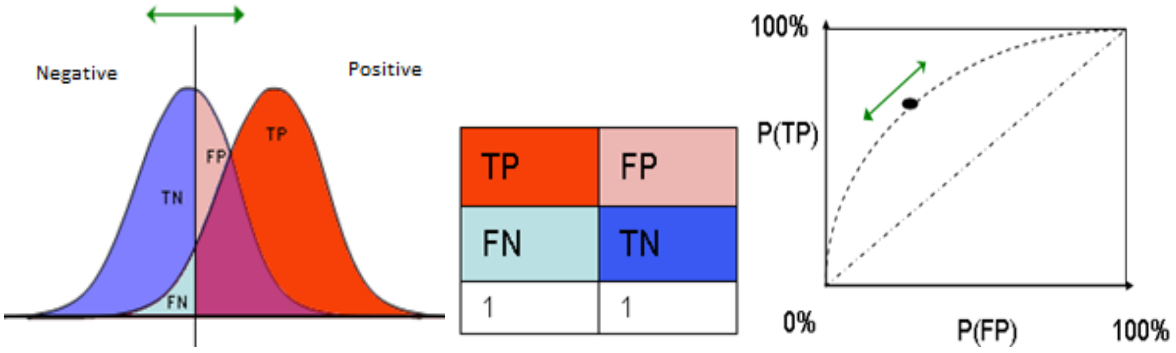


Figure 3-12. The generation of a ROC curve. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

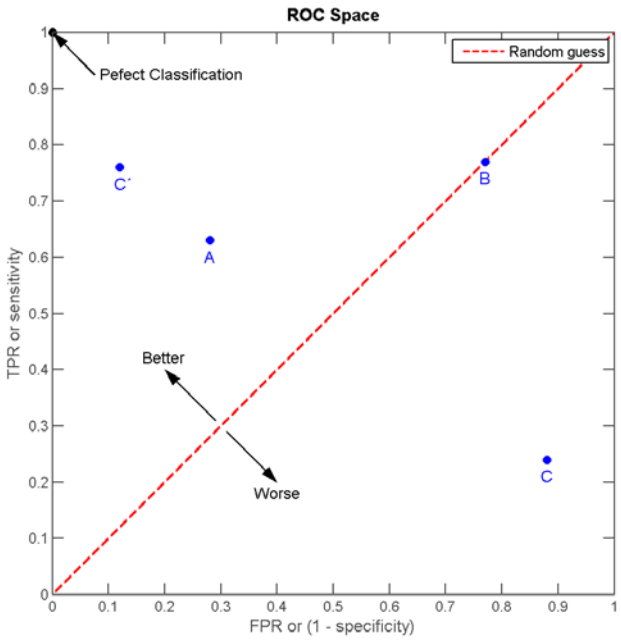


Figure 3-13. The ROC space and plots of 4 example predictions. The higher the curve is, the more likely the classifier will give the correct result. The predictions on the diagonal dashed line are equivalent to randomly guessing.

Table 3-1 Fitted logistic response functions: coefficients and their corresponding AUC values.

Species	Lettuce			Broccoli		
Model	RGB	HSV	Ill-inv	RGB	HSV	Ill-inv
β_0	-0.486825	-3.606021	-2.444331	-3.437759	-7.045284	-1.429986
β_1	-0.793238, 0.0948187, -0.019799	-0.013694, 0.0405418, 0.0123583	-5.830809	-0.15641, 0.073231, 0.087175	-0.002772, 0.0482402, 0.0277419	-6.41317
AUC	0.92	0.94	0.95	0.90	0.86	0.73

Those models are validated and compared using the AUC (area under the curve) of ROC (receiver operating characteristic) curves. ROC (Figure 3-12, Figure 3-13) is a statistic graphical plot that indicates the performance of a binary classifier system with discrimination threshold is varied. The AUC is equal to the probability that a classifier will give the right result for a randomly chosen positive instance. It is often used in machine learning for model

comparison.

Currently, only the results of lettuce and broccoli detection are analyzed. The fitting results are listed in Table 3-1. It is obvious that when detecting lettuce, the Illuminant-invariant model is the best model, and when detecting broccoli, the RGB model is the best model.

Finally, the weight functions are decided to be:

Lettuce:

$$Weight(p) = 1 + \frac{1}{1 + \exp(2.444 + 5.831 * ill)} \tag{14}$$

Broccoli:

$$Weight(p) = 1 + \frac{1}{1 + \exp(3.438 + 0.156 * R - 0.073 * G - 0.087 * B)} \tag{15}$$

3.4.2.4 Segmentation performance

After testing the segmentation algorithm on the data collected, we found that about 95% of the images could be segmented correctly, in which background pixels were eliminated correctly. The reasons for the 5% failures are:

1. Sometimes, the weeds covered most of the area in the field of view. As the ground segmentation relies on the depth information of the ground pixels, it will cause incorrect results if the ground is not observable.
2. Sometimes, especially at early growth stages, the crops are lower in height. Thus, it is hard to tell the height difference between the ground and the crops. In such cases, the color-based segmentation will be the only solution.

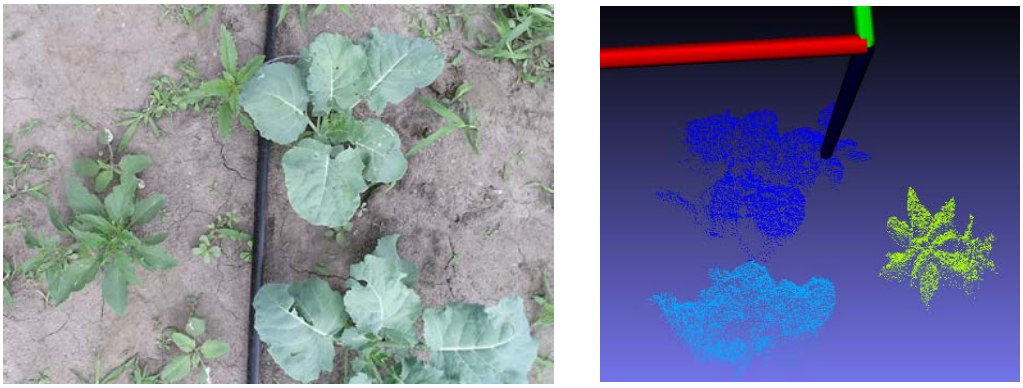


Figure 3-14. Sample segmentation successes. The left figure shows the one color image, and the right figure shows the segmentation result point cloud. The background was eliminated, and the pixels belonging to crops were extracted and separated as clusters.



Figure 3-15. Sample segmentation failures. The left figure shows a segmentation failure case in tomato field, in which the ground was covered with weeds. The right figure shows a lettuce field case, in which the depth-based segmentation was failed, because of the low crops height. Color-based segmentation will still work on the latest case.

3.4.3 Pixels clustering & crop plant candidate extraction

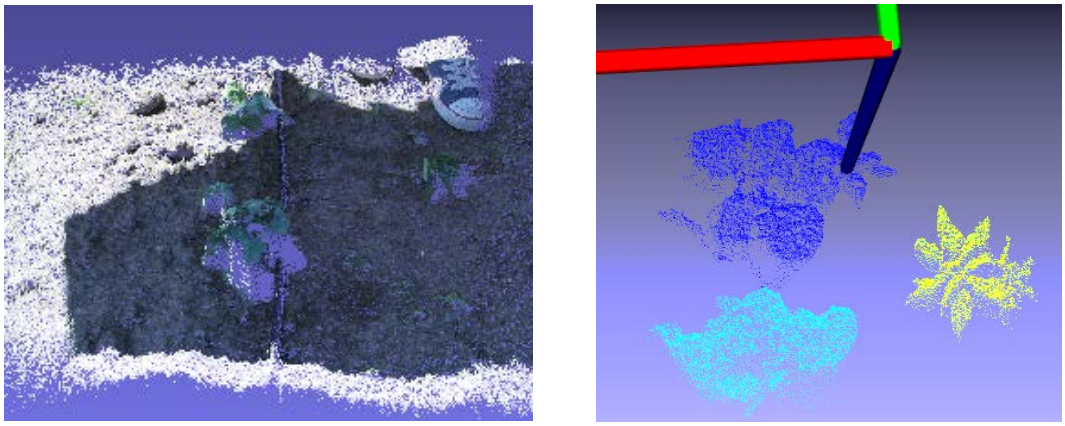
With segmentation by using the both depth information and color information, outlier points which belong to green plants (crop plants and weeds) can be extracted. Then clustering is applied to the plant data point set P_t, in order to assign the remaining points into different clusters representing different plant candidates. In this step, the clustering relies on spatial continuity of pixels only. Therefore, two or more overlapping plants may be grouped into a

same cluster. This overlapping problem will be solved after analyzing each leaf in this cluster.

This clustering problem can be described as an unsupervised 3D clustering problem. It is unsupervised because the spatial continuity is the only information can be relied on in this clustering problem. Different methods have been implemented (except for Superparamagnetic clustering (Blatt, Wiseman, & Domany, 1996)) and applied to solve this clustering problem. A brief comparison is in Table 3-2.

Table 3-2. Table showing the description of different clustering methods, as well as their advantages and disadvantages.

Name	Description	Pros	Cons
K-Mean	Iteratively find K clusters to minimize the sum of with-in cluster variation.	High efficiency	K need to be specified
ISODATA	Modified K-Mean for unknown number of clusters	High efficiency, Algorithm splits and merges clusters	Unexpected results when plants are connected
Superparamagnetic clustering	Simulation of the ferromagnet movement	High quality results	Not open source, hard to implement
Region Growing	Growing from seeds to neighbors, until the whole cloud is processed	High quality results	Low efficiency, since hard for compiler to optimize for parallelization
Connected components in 2D	Find connected components in depth image	Best efficiency	Change in depth (z-axis) not considered when clustering



(a) Point cloud of broccoli crops in field (b) Clustering result

Figure 3-16. A run-time example showing the plants clustering procedure. Fig (a) is the point cloud generated from Kinect v2 sensor after preprocessing. And Fig (b) is the result after background removal and clustering. Different clusters are assigned with different colors. Ideally, each cluster stands for a plant.

After considering both advantages and drawbacks, the 2D connected components method was finally selected, because of its high efficiency, and its negligible drawbacks. Due to the fact that the data's range in depth (or z-axis) is very narrow compared to the other two dimensions (x, and y-axis), thus it is difficult for any clustering method to rely on depth information to separate individual plants perfectly. Further analysis will be performed after extracting features.

With clustering, small clusters containing fewer pixels than a predefined threshold are no longer kept, since they can only be weeds or noise. Thus, the plant data point set P_t is divided into different subsets indicating different crop plant candidates S_i $(i = 1 \cdots n)$.

3.5 Results and Discussion

In this study, the algorithm is tested on all Broccoli images (203 in total) and Lettuce

images (376 in total) collected. The detection ground truth was obtained by human inspection using manually selecting method frame by frame. Some run-time results examples are shown in Figure 3-17. In those images, the background pixels were eliminated, and the crop plant candidates were labeled with circles. The candidate detection algorithm's true positive rate, precision, false discovery rate are listed in Table 3-3. In average, during the test, 98% of target crop plants were detected as crop plant candidates (true positive rate). Among the candidates, about 66% of them are individual target crop plants or crop plants connected with other plants (precision, TP/(TP+FP)), and 34% of them are weeds (false discovery rate, FP/(TP+FP)).

There are several error sources contributed to the miss-detection errors The first source is error from the Kinect v2 sensor. Since working outdoors will increase the noise level of Kinect v2, even with sunlight shaded; The second is error from the algorithms. With the disturbances from weeds, sometimes the image processing algorithm will fail in detecting ground.

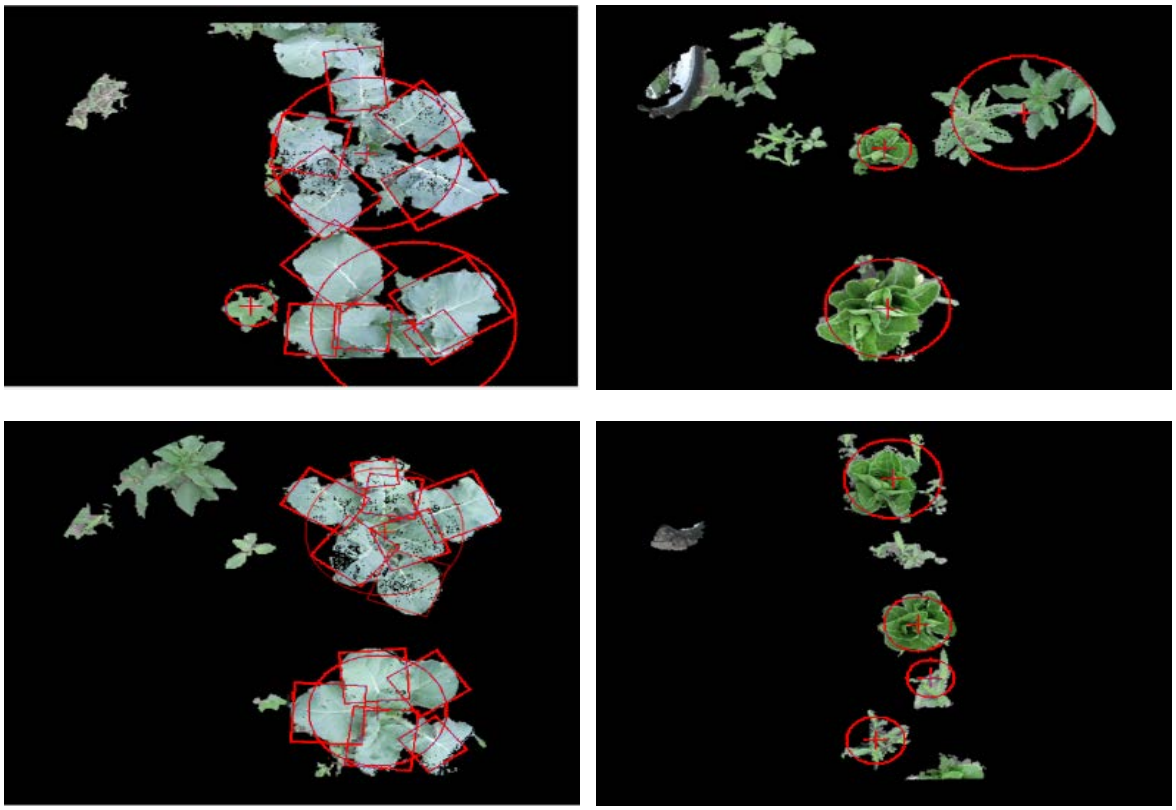


Figure 3-17. Examples of the crop plant candidate detection run-time result. In those images, the detected crop plant candidates are circled.

Table 3-3. The candidate detection true positive rate and precision for broccoli and lettuce at different growth stages.

Collection data	Days after transplant	Broccoli candidate detection True positive rate/ precision	Lettuce Candidate detection True positive rate/ precision
June 13, 2015	12	100% / 66%	100% / 91%
June 17, 2015	16	100% / 72%	100% / 77%
June 23, 2015	22	100% / 76%	99% / 45%
June 27, 2015	26	96% / 75%	95% / 44%

3.6 Conclusion

This paper demonstrates a new method for detecting crop plant candidates by fusing

2D color information and 3D spatial information. The developed algorithm relies only on the on-board sensors of the mobile robot, without using any pre-recorded localization data. Enhanced by the utilization of both 2D and 3D information, the candidate detection algorithm is robust to inconsistent illumination conditions, as well as high weed intensity in images.

Required data was collected from multiple fields of different crop species, with multiple weed species, in different illumination conditions. Candidate detection algorithms have been implemented, tested and evaluated with broccoli and lettuce datasets. The algorithms were successful in detecting the crop candidates with an average true positive rate of 98%, and a precision of 66%. Further processing methods based on the morphological and structural features of plants were applied in order to improve the precision, and will be described in the companion paper, namely plant discrimination. Future work will focus on enhancing the algorithm to improve the performance by optimizing the code.

3.7 References

Alenyà, G., Dellen, B., Foix, S., & Torras, C. (2012). *Robotic Leaf Probing Via Segmentation of Range Data Into Surface Patches.* Paper presented at the IROS Workshop on Agricultural Robotics: Enabling Safe, Efficient, Affordable Robots for Food Production, Vilamoura, Portugal.

Astrand, B., & Baerveldt, A.-J. (2002). An Agricultural Mobile Robot with Vision-Based Perception for Mechanical Weed Control. *Autonomous Robots*, 21-35.

Blatt, M., Wiseman, S., & Domany, E. (1996). Superparamagnetic Clustering of Data. *Physical Review Letters, 76*(18), 3251-3254.

Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*: " O'Reilly Media, Inc.".

Cormen, T. H., & Leiserson, C. E., Rivest, Ronald L.. Introduction to Algorithms.: MIT Press and McGraw-Hill.

Du, J., Wang, X., & Zhang, G. (2007). Leaf shape based plant species recognition. *Applied mathematics and computation, 185*(2), 883-893.

Fankhauser, P., Bloesch, M., Rodriguez, D., Kaestner, R., Hutter, M., & Siegwart, R. (2015). *Kinect v2 for mobile robot navigation: Evaluation and modeling.* Paper presented at the Advanced Robotics (ICAR), 2015 International Conference on.

Finlayson, G. D., Hordley, S. D., & Drew, M. S. (2002). *Removing Shadows from Images*. Paper presented at the Computer Vision — ECCV 2002.

Hunt, E. R., Hively, W. D., Fujikawa, S. J., Linden, D. S., Daughtry, C. S., & McCarty, G. W. (2010). Acquisition of NIR-green-blue digital photographs from unmanned aircraft for crop monitoring. *Remote Sensing, 2*(1), 290-305.

J.M. Alvarez, A. L. o., R. Baldrich. (2008). Illuminant-Invariant Model-Based Road Segmentation. *IEEE Intelligent Vehicles Symposium*.

Jin, J., & Tang, L. (2009). Corn Plant Sensing Using Real-Time Stereo Vision. *Journal of Field Robotics*, 591–608.

Kazmi, W., Foix, S., Alenyà, G., & Andersen, H. J. (2014). ndoor and outdoor depth imaging of leaves with time-of-flight and stereo vision sensors: Analysis and comparison. *ISPRS Journal of Photogrammetry and Remote Sensing*, 128-146.

Li, J. (2014). *3D machine vision system for robotic weeding and plant phenotyping.* (Dissertation), Iowa State University, http://lib.dr.iastate.edu/cgi/viewcontent.cgi?article=4743&context=etd. (13736)

Li, L. (2014). Time-of-Flight Camera – An Introduction. *Technical White Paper*.

Nakarmi, A., & Tang, L. (2012). Automatic Inter-plant spacing sensing at early growth stages using a 3D vision sensor. *Computers and Electronics in Agriculture*.

Nielsen, M., Andersen, H. J., Slaughter, D. C., & Giles, D. K. (2004). *Detecting leaf features for automatic weed control using trinocular stereo vision.* Paper presented at the Precision Agriculture and Other Precision Resources Management, Minneapolis, MN, USA.

Philipp, I., & Rath, T. (2002). Improving plant discrimination in image processing by use of different colour space transformations. *Computers and Electronics in Agriculture, 35*(1), 1-15. doi:http://dx.doi.org/10.1016/S0168-1699(02)00050-9

Point Cloud Library (PCL). (2015). Retrieved from http://pointclouds.org/

Rusu, R. B. (2009). *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. Retrieved from

Sansoni, G., Trebeschi, M., & Docchio, F. (2009). State-of-The-Art and Applications of 3D Imaging Sensors in Industry, Cultural Heritage, Medicine, and Criminal Investigation. *Sensors*, 9, 568-601.

Scotford, I. M., & Miller, P. C. H. (2005). Applications of Spectral Reflectance Techniques in Northern European Cereal Production: A Review. *Biosystems Engineering*, 235-250.

Slaughter, D. C. (2008). Autonomous robotic weed control systems: A review. *computers and electronics in agriculture*.

Sui, R., Thomasson, J. A., Hanks, J., & Wooten, J. (2008). Ground-based sensing system for weed mapping in cotton. *Computers and Electronics in Agriculture*, 31–38.

Surmann, H., & Nüchter, A. (2003). An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, Volume 45, Issues 43–44.

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

Zwiggelaar, R. (1998). A review of spectral properties of plants and their potential use for crop/weed discrimination in row-crops. *Crop Protection*, 189-206.

CHAPTER 4. PLANTS DETECTION, LOCALIZATION AND DISCRIMINATION USING 3D MACHINE VISION FOR ROBOTIC INTRA-ROW WEED CONTROL– PART B: PLANT DISCRIMINATION

4.1 Abstract

Robotic weeding offers a possibility of controlling weeds precisely, particularly for weeds growing near or within crop rows. In robotic weeding, computer vision is the most competent method when color and shapes need to be analyzed at high speed. Crops localization and discrimination are challenging tasks in robotic weeding with compter vision, since it is difficult for a computer to understand the differences of human-use high-level descriptions of the appearance.

In this section we are introducing a feature extraction and classification system for robotic weeding, based on the crop plant candidate detection algorithms in the companion paper. Morphological and structural features were extracted from leaves and canopies, with which the detected candidates can be further separated into different plants. Supervised machine learning technique was used to identify crop plants of different growth stages. In addition, different machine learning methods were also evaluated and compared. Finally, the algorithmic efficiency and classification accuracy of the proposed discrimination methods were validated and demonstrated. Based on the data collected, the AdaBoost algorithm performs the best, with error rates of 3.1% (broccoli) and 6.8% (lettuce).

4.2 Introduction

Crops/weeds differentiation plays an important role in robotic weeding. But it is challenging for a computer to accomplish the discrimination of crops and weeds, due to the difficulty for a computer to understand the human-use high-level descriptions of the appearance differences between crops and weeds. Supervised machine learning is a competent technique, in which computer learn general rules to map the inputs to outputs. In machine learning, the crop/weed discrimination can be characterized as a classification problem.

Many supervised learning methods are available. Most of the conventional methods use vectors of numerical and categorical features as input, as well as the supervising. Then, a probability function is selected, and a training method is used to determine the probability function using the input. In the end, the output could be predicted given a new input feature vector. The performance of supervised learning heavily depends on feature selection, which represents the raw data, as well as the learning method (Bengio, Courville, & Vincent, 2013).

However, a breakthrough in feature learning and deep learning was initiated by Hinton et al., (2006). Later on, great progress has been made that significantly increased the performance of machine learning. The central idea of deep learning is referred to as greedy layer-wise unsupervised pre-training, which is to learn a hierarchy of features on one level at a time, using unsupervised feature learning to learn a new transformation at each level to be composed with the previously learned transformations (Bengio et al., 2013). With the support of the advancement of computing capacity of computers, especially GPUs, a deep supervised

predictor such as a neural network classifier with many layers can be trained. Its ability has been proved in many areas, such as Language Processing (**Seide**, **Li**, & **Yu**, 2011) and Computer Vision (Szegedy et al., 2014). In many Data mining competitions such as ImageNet 2014, deep learning methods covered first places. Compared with traditional feature-based machine learning methods, the advantages of deep learning are obvious: it requires less work for operators to extract and select features. Theoretically, the best feature can be selected by operators is also obtained in deep learning. The disadvantage of deep learning is that a large training dataset is needed, which increases the complexity of generating training datasets.

In this study, limited by knowledge and dataset size, only conventional feature-based classification methods were considered. Plant candidates were detected at first using the computer vision system stated in the previous section. After detecting plant candidates, leaf and canopy features were extracted from the plant clusters for recognition. The features can be divided into three categories: color based, plant morphology based, and plant structure-based features.

Color based features indicate the physical properties of objects, mainly the reflection properties. They can be useful when the colors of crop plants are different from that of weeds, e.g., broccoli has stronger blue color than weeds, and lettuce has more saturated green than weeds. However, color-based features are not sufficient, as crops may share some similar colors with some weed species, and color can be influenced by changing illumination conditions. As a result, plant morphology and structure-based features are both utilized in our

research.

Structural features are descriptions of the structure of plants, such as the stems, the leaves and their spatial relationship. Structural features are not commonly used as they are typically described by high-level descriptive languages, which are difficult to parameterize. In addition, the structural features need to be extracted from relatively comprehensive 3D models of plants.

Morphological features are descriptions of shapes of plants. Morphology features are most commonly used in agricultural computer vision due to their reliability and parameterization simplicity. For instance, plant canopy shapes such as height and diameters, as well as leaf shapes such as length and width, are scalars and easy to extract using 2D or 3D sensors. There are many research projects using morphological features for plants classification. and about 90% to 95% recognition rates in average were achieved (Q. Wu, Zhou, & Wang, 2006; S. G. Wu et al., 2007). In the research of Weiss et al., (2010), the authors extracted plant canopy morphological features from LIDAR data for species discrimination, and achieved a 98% recognition rate. However, their experiments were mostly performed indoor. In our study, the challenge is that the classification needs to be performed outdoor, with influences of weeds, damaged leaves, lighting condition changes, implying a high data noise level.

Based on a literature search, classification methods addressing outdoor plant classification problems are very few, and their performances are not robust enough for automated weeding operation. In this chapter, a crop/non-crop discrimination (classification)

system for robotic weeding will be introduced.

The crop/non-crop classification system used supervised machine learning technique, and used plant color, morphological and structural features as input. Different classification methods were evaluated and compared in this application. All of the classification programs were implemented using R language.

4.3 Plant Features

In this section, the definitions of the plant leaf and canopy features which are extracted and used in this study will be introduced. The features include morphological features, structural features, as well as some color features. The morphological features consist of features which can describe dimensions and shapes of canopies or leaves, such as plant height, diameter, leaf aspect ratio and leaf area convex hull ratio, etc. Color features include average hue and saturation, as well as illuminant invariance to take use of the color difference between crops and weeds if possible. And the structural features describe the structures of plants, such as the venation of leaves. With those features, classifiers can be built for different species at different growth stages. The definitions of these features to be extracted are listed below, and some of them are referred to the work of Du et al. (2007).

Leaf features:

(1) Leaf venation (vein)

Binominal variable, indicating whether the venation of the leaf can be extracted.

(2) Leaf average height (havg):

The distance between the field ground and the center point of the leaf.

(3) Leaf area (area):

The area of the leaf after projection to the ground plane.

(4) Leaf length (leaf_l):

The maximum distance on the leaf to the center of the plant.

(5) Leaf width (leaf_w):

The maximum width on the leaf, which is perpendicular to the length direction.

(6) Leaf aspect ratio (ratio):

The ratio between L_{max}: the maximum length of the leaf measured from the center of the plant, and W_{max}: the maximum width on the leaf

$$ratio = L_{max}/W_{max} \tag{16}$$

(7) Leaf roundness (leaf_roundness):

The ratio between the leaf contour length C_{leaf} and the bounding ellipse circumference C_e.

$$leaf_roundness = C_{leaf}/C_e \tag{17}$$

(8) Leaf rectangularity (leaf_recti):

The ratio between the leaf area A_{leaf} and the leaf's bounding rectangle area A_c.

$$leaf_recti = A_{leaf}/A_c \tag{18}$$

(9) (10) Leaf hue, leaf saturation (l_clr_hue, l_clr_setra):

The average values of Hue and Saturation channel in HSV (Hue-Saturation-Value)

color space respectively of the leaf pixels.

(11) Leaf illuminant-invariant value (l_clr_ill):

Finlayson in (Finlayson et al., 2002) proposed the way using illuminant invariant (ill-inv) space method to remove the shadows. The theory is built on the assumptions of the narrow-band color sensor, Lambertian Surface (on which Lambert's cosine law is obeyed), and Planckian Light (Color Temperature). It transfers the color from RGB color space into ill-inv color space, where the Y axis is $\log(B/R)$, and the X-axis is $\log(G/R)$. Then if project those colors in ill-inv color space onto a specific line, which is called invariant axis, all different colors can be separable with different illuminant-invariant values. This feature is defined as the average ill-inv values of the leaf pixels.

Canopy morphological features:

(12) Canopy height (z_height):

The maximum perpendicular distance between the field ground and the highest point of the plant.

(13) Canopy radius (radi)

The average length of the long leaves in all directions. The short new grown leaves which are shorter than their underlying leaves are not considered.

(14) Leaf number (leaf_n):

The number of leaves that can be segmented from the plant data.

(15) (15) Canopy hue, Canopy saturation (clr_hue, clr_setra):

The average values of Hue and Saturation channel in HSV (Hue-Saturation-Value) color space respectively of the canopy pixels.

(16) Canopy illuminant-invariant value (clr_ill):

The average illuminant-invariant value of the plant data points.

4.4 Algorithm Design

In order to accomplish plants detection and localization, 3D spatial and 2D color information were fused together. The framework of the entire algorithm is shown in the flowchart (Figure 4-1).

Step 1: Preprocessing. Remove the invalid pixels and remove noise points in point clouds. This procedure is done by using a usable-are filter, a cut-off filter and a simplified neighbor count filter.

Step 2: Registration and segmentation. 2D and 3D information are fused. The ground is detected using both 2D color and 3D depth information, and a plane equation is fitted using RANSAC. All vegetation pixels are extracted in the images.

Step 3: Clustering & candidate extraction. The remaining points which belong to plants are separated into different clusters. Each cluster may contain one plant, or multiple objects which are connected in space, with species not determined. Crop plant candidates can be extracted by simple thresholding on cluster size to shift small objects which can be regard as non-crop objects.

Step 4: Feature extraction. Canopy color, morphological and structural features in each crop plant candidate are extracted. If possible, leaves are segmented as well, and leaf morphological and structural features are extracted as well.

Step 5: Grouping. The features extracted (venation for broccoli, and canopy shape for lettuce in this study) are used for grouping, in order to separate the pixels within the same candidate detected into different plants. Locations can be calculated as well.

Step 6: Classification. Crop/weed classification is applied to all the separated plants based on their features (canopy, leaf) using machine learning techniques.

In this section, step 1, 2, 3 are used to accomplish crop plant candidate detection and are stated in the companion paper. Step 4, 5, 6 are used for plant discrimination, which will be introduced in this paper.

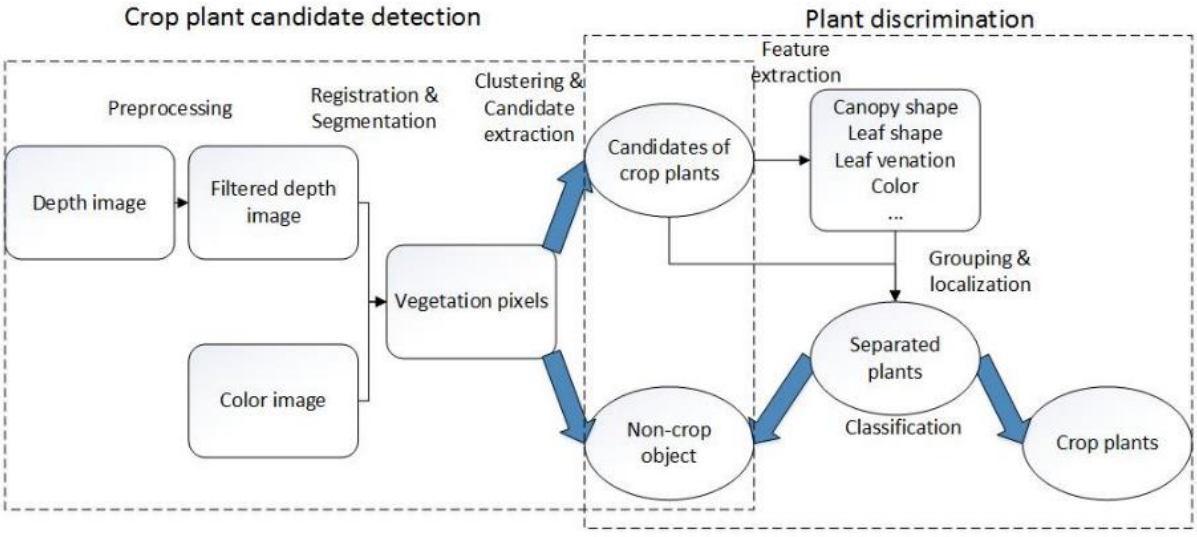


Figure 4-1. The flowchart of the image processing procedures.

4.4.1 Plant leaf segmentation

To accomplish plant localization and recognition with leaf features, it is very important to segment each leaf out of the plant candidate clusters if possible. However, in the dataset collected, leaf damage was found common, which increases the difficulty of leaf segmentation if using the leaf shape as patterns.

Thus, the main strategy is to use marker-controlled watershed segmentation algorithm on the depth image and color information (Hue channel in HSV color space) to segment leaves.

Watershed segmentation is a simulation of filling the valleys with water on a topographic surface, where the topographic surface is generated from a grayscale image. The marker-controlled watershed algorithm enhanced the watershed by starting the filling of water from a previously defined set of markers.

The “watersheds” in this study can be obtained by finding the edges of leaves. The edges are the discontinuities in depth and color. This is because different leaves usually have different heights, and sometimes they may have different color intensities due to the sunlight shadow. Those discontinuities are quantified using the magnitudes of gradients in these images.

In geometry, the gradient of a function $f(x_1, x_2)$ in 2D Cartesian coordinate system is given by

$$\nabla f = \frac{\partial f}{\partial x_1} i + \frac{\partial f}{\partial x_2} j \tag{19}$$

where i, j are the standard unit vectors. The magnitude of gradient is then defined as

$$|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x_1}\right)^2 + \left(\frac{\partial f}{\partial x_2}\right)^2} \tag{20}$$

In image processing, those partial derivatives can be approximated by convolving Sobel operators

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \; and \; S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{21}$$

to a source image A for horizontal and vertical derivatives respectively. Thus the magnitude G of the gradient becomes the root of the square sums of above two derivatives

$$G = \sqrt{(S_x * A)^2 + \left(S_y * A\right)^2} \tag{22}$$

where the operator * here is the 2D signal processing convolution operation.

After obtaining the gradient magnitude of both depth image and color image, edge images $E_d \; and \; E_h$, can be obtained by thresholding both depth and color gradient magnitude image ($G_d \; and \; G_h$, respectively). These edge images are merged into a watershed image E.

The label map L is obtained from the edge image E. The area where its distance to any edges is smaller than a threshold is labeled as one of foreground labels (label 2~n, n>2). The pixels identified as background in previous steps are labeled as background (label 0). The rest of the pixels are labeled as unknown (label 1).

Then the depth image, as well as the label map, are passed to the watershed function from OpenCV library. An example resultant label image is shown in Figure 4-2.

In this study, leaves were segmented from the broccoli dataset. As the leaves are not obvious in the lettuce dataset, leaf segmentation was not performed for lettuce.

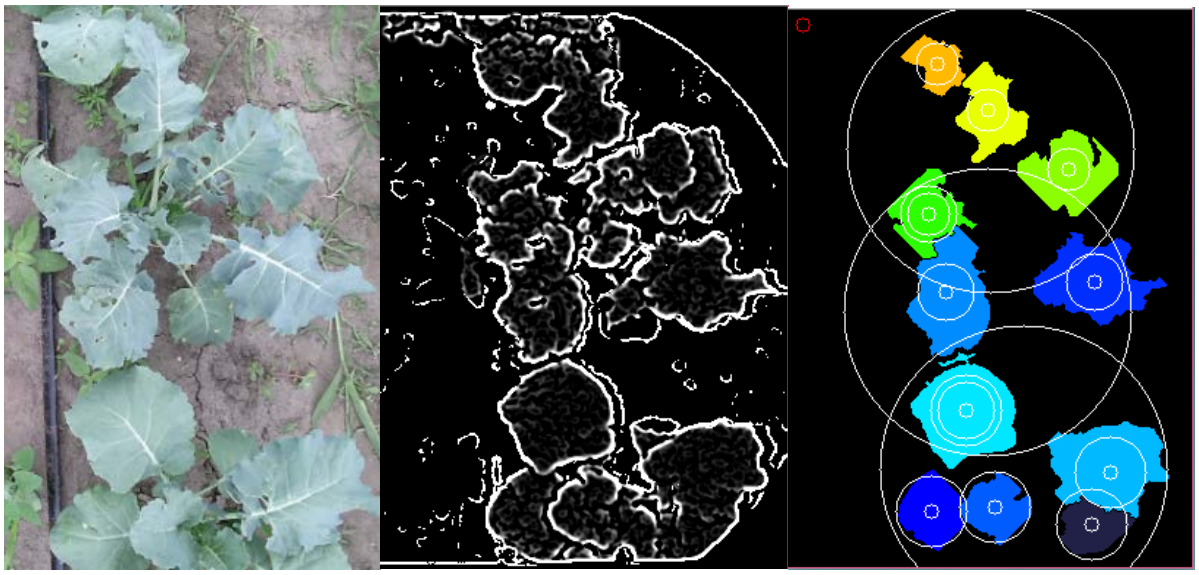


Figure 4-2. A run-time output example showing the leaf extraction procedures. Left: Color image of on cluster after segmentation and clustering, with connected crops, occlusions, as well as broken leaves. Middle: Combined gradient magnitude image on both color and depth. Right: Leaves extraction result example. Most of the leaves are extracted and labeled with different colors.

4.4.2 Grouping & localization

As mentioned before, during crop plant candidate detection stage, only the spatial relationship of the pixels was considered while clustering. Thus there may be connected plants within the same candidate cluster. Such connected cases can be detected by analyzing the pixel distribution using PCA (principal component analysis). It is necessary to separate those connected plants for plant localization and discrimination.

There are researchers addressing the connected problem of plants in past years. Mainly two categories of features are used to separate the connected plants. The first one category is based on color features. For instance, Tang (2002) built a color based Bayes classifier to distinguish centers from leaves of corn plants at early growth stages, with which connected corn plants can be separated. However, methods using color features are still

sensitive to varying light conditions and growth stages.

The other category is based on structural features, which localize the plant based on the features such as overall shapes, pixel distribution, or the topology of leaves. Many types of research in the past separated plants by finding the area with high biomass pixel densities for simplicity (Tillett, Hague, Grundy, & Dedousis, 2008). Such methods only work for separating crop plants which are easily separable or whose overall shapes are approximately spherical, such as lettuce in this study.

Also, some researchers separate connected plants by analyzing the leaf topology (leaf directions, leaf distributions, etc) based on the morphological features of leaves, such as curvature (Jin & Tang, 2009), leaf contours (Persson & Åstrand, 2008) and leaf tips (Midtiby, Giselsson, & Jørgensen, 2012). Due to the occlusion problems as well as broken leaves in our dataset were very common (Figure 4-3), the morphological features such as leaf tips are hard to detect. The surfaces of broccoli leaves can also be irregular, thus the curvatures are not usable.

However, leaf directions were found robust when separating plants in the broccoli dataset. Moreover, in this study, venation was found the most stable feature in determining the directions of the leaves. As the venation close to the center is obvious and is not likely to be occluded by other plants, or damaged by small animals.



Figure 4-3. One example image of broccoli collected in the field, with damaged leaves and interconnected canopies. Leaf venation was found a robust feature to determine the leaf directions.

In this section, a new developed venation-based leaf grouping algorithm is proposed for separate broccoli plants. This algorithm works on species with border leaves and obvious straight veins. The algorithm has two steps, the first is venation extraction, in which extract the vein pixels and convert them into line segments. The second is center finding using those line segments.

4.4.2.1 Venation extraction

Veins are found obvious in the Red channel from RGB color space images. As the pixels from the veins have higher values in red compared to surrounded pixels, those vein pixels are considered as “ridges” in the images.

In image processing, the concept of “ridge” is derived from differential geometry. In differential geometry, a ridge point is a point where a smooth surface has a local maximum or minimum of principal curvature on lines of curvature. The ridges are the set of ridge points from curves on the surface. Equally, a “ridge” point has a lower negative principal curvature,

and a “valley” point has a higher positive principal curvature.

For surface with format $\sigma(x,y) = \big(x, y, f(x,y)\big)$, if assume only the fluctuation is considered, the principal curvatures of the surface patch can be simplified as the eigenvalues of the second fundamental form of the surface. In this case, the second fundamental form of the surface becomes the Hessian matrix of $f(x,y)$

$$H\big(f(x,y)\big) = \begin{pmatrix} \dfrac{\delta^2 f}{\partial x^2} & \dfrac{\delta^2 f}{\partial x \partial y} \\ \dfrac{\delta^2 f}{\partial x \partial y} & \dfrac{\delta^2 f}{\partial y^2} \end{pmatrix} \tag{23}$$

which reflects the second-order partial derivatives of $f(x,y)$ at each pixel. The eigenvalues of the Hessian matrixes indicate the maximum and minimum curvatures on different curves at certain pixel. Their corresponding eigenvectors indicates the directions on which the extremum curvatures are obtained.

As shown in Figure 4-4, a flat surface has eigenvalues to be both zero, positive eigenvalues correspond to valleys or saddles, and negative eigenvalues correspond to ridges or peaks.

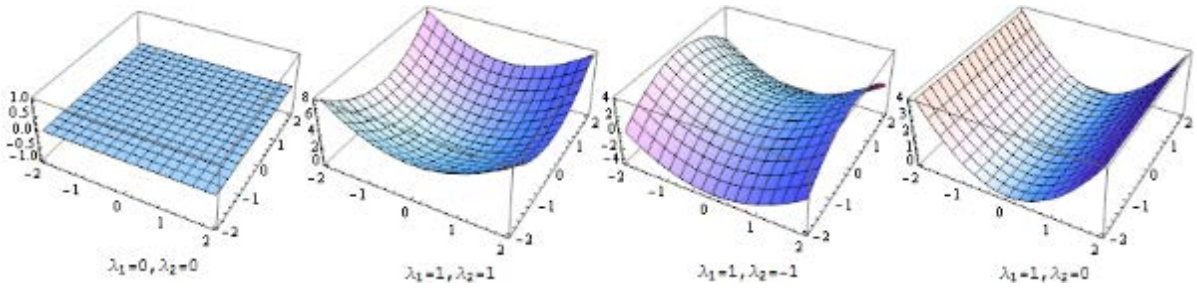


Figure 4-4. Surfaces with different eigenvalues of their Hessian matrixes. A flat surface has eigenvalues to be both zero, positive eigenvalues correspond to valleys or saddles, and negative eigenvalues correspond to ridges or peaks.

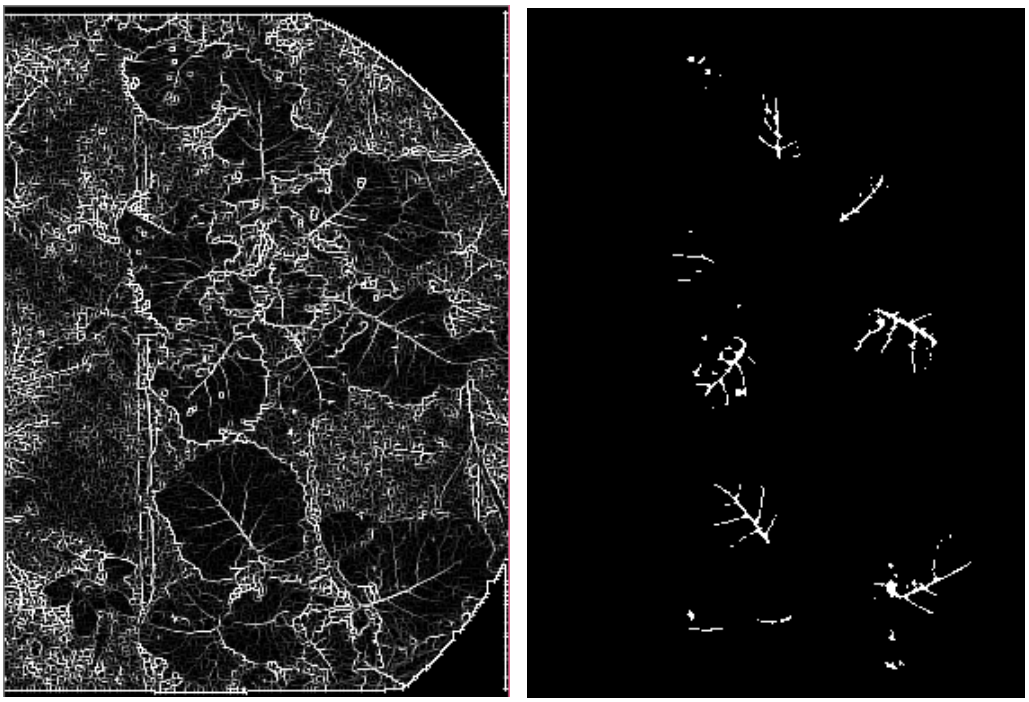


Figure 4-5. A run-time output example showing venation extraction methods. The left figure shows the map of lower eigenvalues of Hessian Matrixes at each location. Ridges are colored with brighter colors. The right figure shows the veins extracted by thresholding, with noise.

Thus, for each cluster found in previous steps, a map of the lower eigenvalue of each pixel in the red channel can be generated (Figure 4-5). The eigenvalues are inversed, such that the brighter pixels are more likely to be ridge points. It can be seen that the veins are obvious on leaves.

By limiting the processing areas to those leaves' areas (obtained in leaf segmentation step), the veins can be segmented by thresholding. The threshold used in this study was determined by experiments and have been tested promising. The veins extracted by thresholding might be noisy (Figure 4-5). Then image dilating, thinning and connected components algorithms are used to skeletonize the veins, and eliminate the unconnected pixels, which are noise pixels.

4.4.2.2 Crop plant center detection

Based on the field of view of the sensor while working, maximum three crop plants can be observed within the same image. So in this method, three circles are created to represent three potential combinations of the leaves for each crop plant candidate with connection. Those circles are initialized at different positions at first. Then the circles move to the new positions by calculating the mean positions of pixels from their surrounding leaves iteratively, until their positions are converged. Therefore, three potential combinations of leaves can be generated (Figure 4-6).

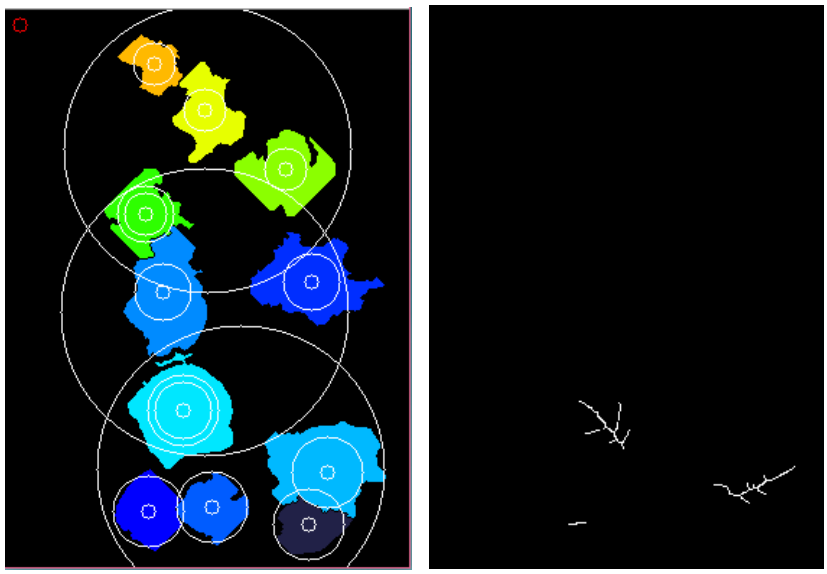


Figure 4-6. A run-time output example showing the venation skeletons extracted from the same leaf combination. The left figure shows the potential combinations of leaves. The right figure shows the skeletonized venation map extracted from the bottom leaves combination, which is passed to the center finding step.

The center detection method can be applied to the veins extracted from those potential leaf combinations. Suppose the primary veins' direction can be extracted perfectly, then all the primary veins can be estimated as a line segment. Then, ideally, the center point

lies on every extension lines of each line segment. If there are only two lines unparalleled, the point of intersection is unique. If there are more than three lines, the problem becomes finding the point, which is closest to all those extension lines.

The challenge of this method is the difficulty for the computer to distinguish the primary vein with other branches by analyzing only the target leaf, due to the imperfect extractions (noise pixels, missing branches). However, we can find out all the branches in each leaf, and estimate their probability of being primary branches. The algorithm can be summarized as three steps:

1. Find all the branches, and simplify them into line segments.
2. Estimate the probability of being a primary vein for each line segment, and assign the probability to each line segment as a weight factor.
3. Solve the center-detection problem by using the weighted robust least square method.

In the first step, all the joints of branches are found to separate the branches. In many thinning algorithms with small thinning windows, it can be observed that the joints always involve some certain patterns. For instance, in the algorithm of Zhang & Suen (1984), all the joints involve patterns of either ‘Y’ or ‘ꞏ’ (Figure 4-7). With those joints erased, the separated vein branches were fitted into line segments using least square method.

The second step is to assign each line segment a weight factor, which reflects the probability of being a primary vein. It is obvious that a primary vein is the longest and have most joints on it for most of the times on a leaf. Thus, the lengths of line segments l and the

counts of joints n lying on them are used for calculating the weight factors w. The algorithm is stated in Figure 4-8:

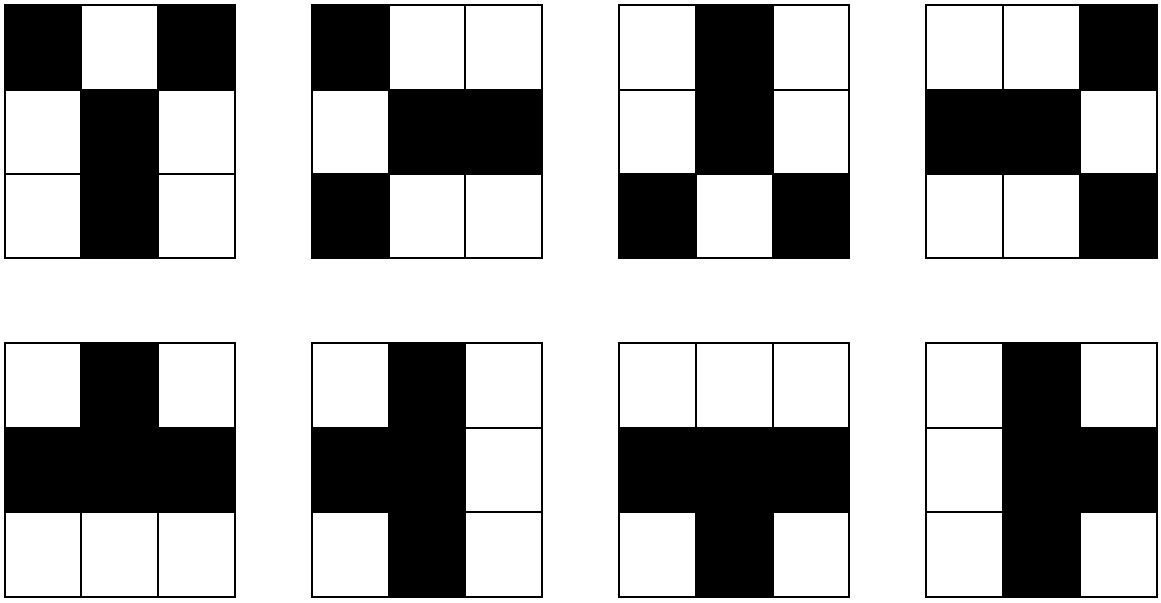


Figure 4-7. The patterns which the joints involve after thinning with the method proposed by Zhang & Suen, (1984). The upper row shows the 'Y' pattern in 4 directions, and the lower row shows the '⊦' pattern in 4 directions. All the pixels with these patterns are joints.

Algorithm: Weight factor calculation with in leaf

Input: Joints *p*, separated line segments (equation *n*, length *l*)
Output: Weight factor *w* for each line segment
Procedure:

1. for each line segment *(n, l)*
2. Set on-line joint count $t = 0$
3. for each joint *p*
4. Calculate the vertical Euclidean distance *d* from *p* to *(n, l)*
5. if distance *d < 5* (in pixel)
6. Joint count++
7. end for
8. Weight multiplier *c = max(1, t * 3)*
9. end for
10. Find the maximum weight multiplier c_{max}
11. Normalize weight multiplier $c = c / c_{max}$
12. for each line segment *(n, l)*
13. Weight factor *w* = multiplier *c* * line segment length *l*
14. end for

Figure 4-8. The algorithm of defining the weight factor of each line segment when calculating the center.

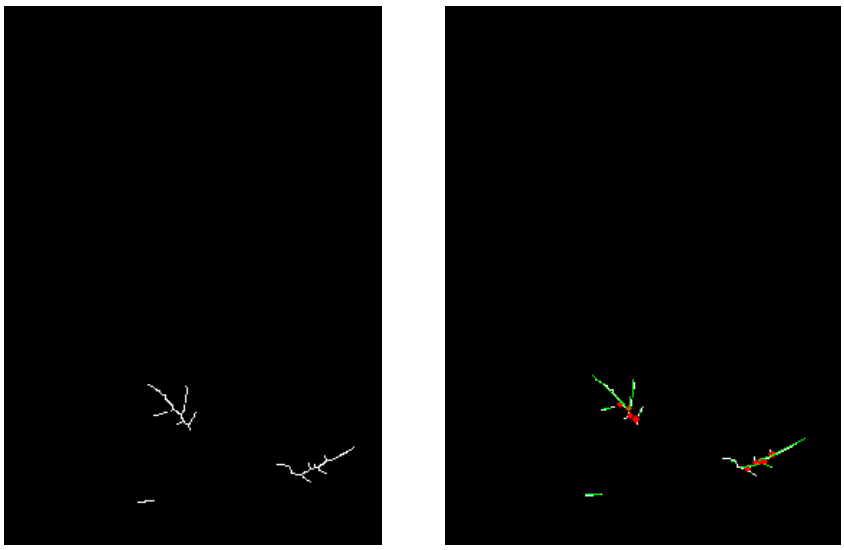


Figure 4-9. A run-time output example showing the results after the first and second steps in center finding. The left figure shows the venation skeletons of one potential leaves combination. In the right figure, the joints are colored in red, and the fitted line segments are drawn in green, with their lengths equal to their weight factors.

The last step is to perform the weighted robust least square algorithm using the segmented line segments to finish the center finding. The theory is similar to the algorithm used for refinement of depth based segmentation, which uses the inliers to perform the least square method, iteratively. The algorithm is summarized in Figure 4-10:

With the proposed algorithm, the potential leaf combinations with proper centers output (the centers are surrounded by leaves, not on leaves) are regarded as correct leaf combinations, which indicates individual plants, and the problem of connection is solved. For each individual plant, all the canopy features can be extracted again if necessary, and all the leaf features can be taken from the pre-extracted leaf feature set.

Algorithm: Center finding

Input: Separated line segments (normalized line equation *n*, weight *w*), stop criterial ε
Output: The center position *p*
Procedure:

1. Iteration *i* <- *0*
2. Previous position p_{i-1}
3. Current position p_i <- initial position p_0
4. *while* $|p_i - p_{i-1}| < \varepsilon$
5. $p_{i-1} = p_i$
6. *for each* line segment *(n,w)*
7. calc-weight *cw* <- *0*
8. Calculate the vertical Euclidean distance *d* from p_{i-1} to *(n, l)*
9. *if* distance *d < mean(d) + 3 * stddev(d)*
10. *cw* <- *w*
11. *else cw* <- *0*
12. *end for*
13. Perform least square to solve: $p_i = argmin_p \sum(cw_j * dis(p, l_j))^2$
14. *end while*
15. $p = p_i$

Figure 4-10. The algorithm of center finding using vein line segments.

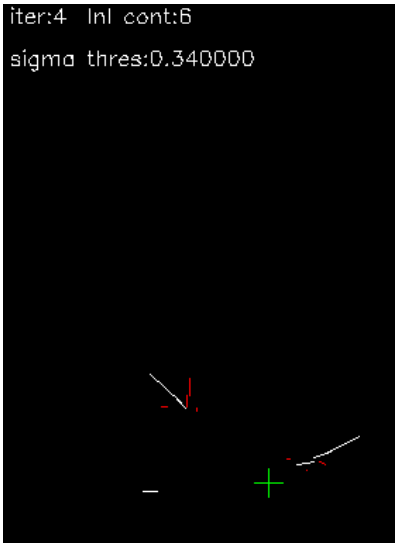


Figure 4-11. A run-time output example showing the iterations of finding the center. The green cross indicates the resultant center location in the current iteration. The line segments with white color are the active line segments (inliers, with non-zero calc-weights). And the line segments in red are the inactive line segments (outliers, with zero calc-weights) in the current iteration.

4.4.3 Supervising user interface

In this application, a user interface was built in order to generate supervised data for crop/weed discrimination (Figure 4-12). For each separated plant (obtained from previous steps), the user can specify whether it is a crop or weed, or it is a wrong detection. In addition, if any leaf can be segmented from the plant, the user can also tell whether it is a crop leaf, a weed leaf, a piece of ground, or a wrong detection.

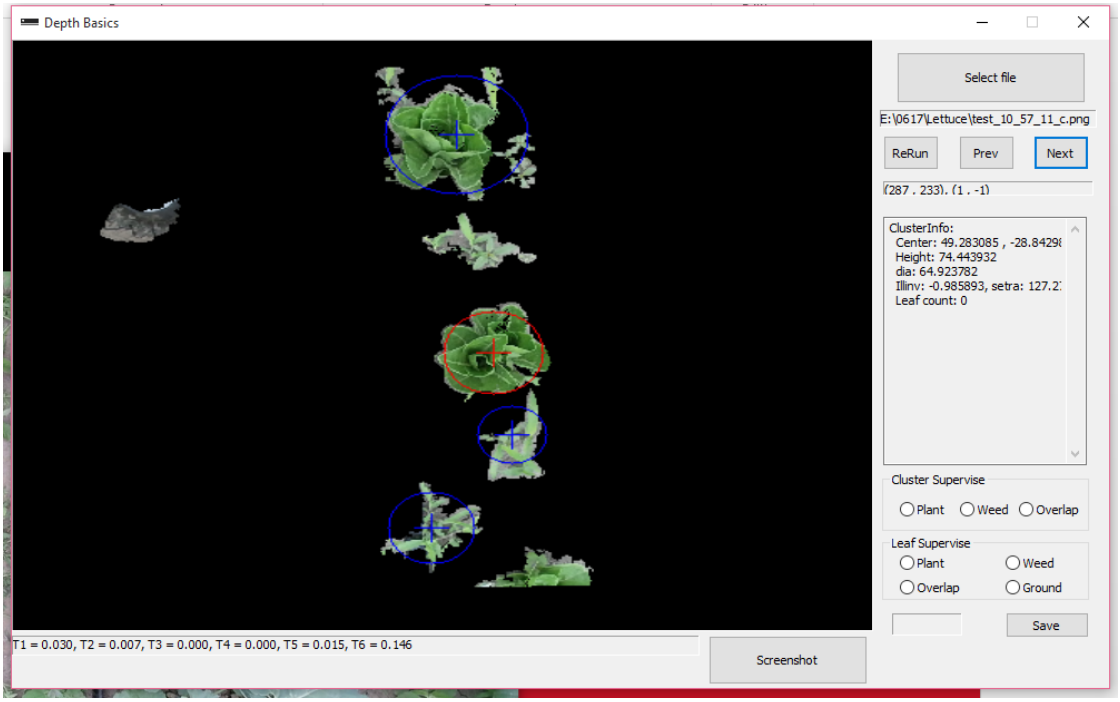


Figure 4-12. The supervising user interface. It can realize plant-based supervising and leaf-based supervising. Plants detected are labeled with circles. With buttons on the right side, the operator can identify the crops and weeds. The supervising input as well as the features of each plant are saved.

In this study, 403 sets of plant features were extracted from the Broccoli field dataset, with 289 of them are from broccoli crop plants, and others are from non-crops. 861 sets of plant features were extracted from the lettuce dataset, with 376 sets of them are from lettuce crop plants, and the other are from non-crop objects.

4.4.4 Classification methods

The purpose of classification is to predict the class of a new unknown sample point based on database that has been separated into different classes. All numeric features can be combined to feature vectors in order to build classifiers. As the leaf features can be extracted from plants of some species, such as broccoli, two classifiers are built. One is for leaves, and another is for plants. In leaves classification, the result should be whether those leaves are from crops or weeds, or they are incorrectly segmented. In the plant classification, the classifier uses the features of classified crop leaves, as well as the plant features, then output whether the plant is a crop, or a non-crop.

There are many classification algorithms available, for instance the logistic regression (LR), the k-nearest neighbors (KNN), the artificial neural network (ANN), the Bayes classifiers such as LDA and QDA, the support vector machine (SVM), and tree-based classifiers such as decision tree and random forest (RF). The performance of each classifier highly relies on the distributions of data. Thus, it is not suitable to subjectively decide which model is the best. In this study, all the above models are fitted using the features extracted. Then those models are evaluated and compared by using Cross-Validation (CV) results. The evaluation program was implemented using R language (R Development Core Team, 2010).

In this section, different classification algorithms are introduced, as well as their tuning parameters, and some cautions.

K-Nearest neighbors

The K-Nearest Neighbors algorithm (KNN) is a non-parametric method used in classification. A new sample point is classified by the votes of its k nearest neighbors. The concept of distance to determine the “nearest” neighbors is usually the Euclidean distance in feature space. Thus, in order to equalize the importance of each predictor, it is important to normalize the predictors in preprocessing.

It can be seen that the parameter k would affect the result. Theoretically, the higher k will result in a smoother decision boundary, then reduce the noise on the classification, but make the boundary between classes less distinct. Thus, 10-fold Cross-Validation is used to select the best k in this study.

The KNN function such as prediction and cross-validation were implemented in the package “class” (Venables & Ripley, 2002) in R.

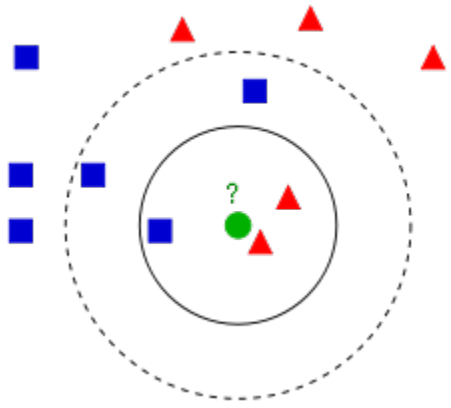


Figure 4-13. Example of k-NN classification. The prediction is voted by its k nearest neighbors. The classification result should be *Red* when k = 3 (solid line), and should be *Blue* if k = 6 (dashed line). Adopted from K-nearest neighbors algorithm, ln Wikipedia, n.d., Retrieved May 28, 2007, from https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm. ©2007 by Antti Ajanki. Adopted with permission.

Logistic regression

The logistic regression model arises from the desire to model the posterior probabilities of the classes via linear functions while at the same time ensuring that they sum to one and remain in [0,1] (Hastie, H. Friedman, & Tibshirani, 2009). This method fits a logistic function, which is:

$$F(x) = \frac{1}{1 + \exp(-\beta_0 - \beta_1 x)} \tag{24}$$

The advantage of this method is that it is based on maximum likelihood theory, and there is no assumption on the distribution of data.

In this method, as all the predictors are used to fit the logistic regression model, it will result in sparse data points in the predictor space, which is also called the “curse of dimensionality”. Thus, subset selection is performed to reduce the dimension. In this study, “Best subset selection” method is used, which find the best model (simple and less CV error) among all the combinations. Since only less than 10 predictors are extracted in this study, only 2^10 = 1024 combinations are considered, which is not a big deal for current computers.

The logistic regression function such as fitting, prediction, subset selection were implemented in the packages “stats” (R Development Core Team, 2010) and “bestglm” (McLeod & Xu, 2010) in R.

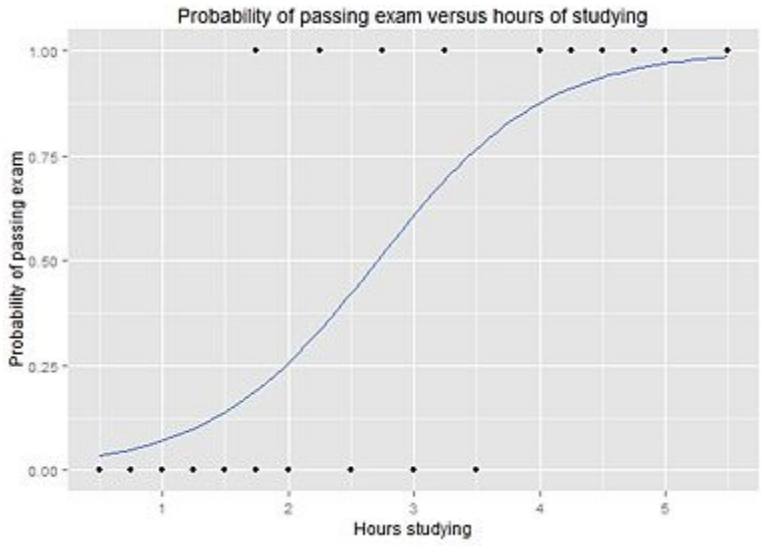


Figure 4-14. An example of logistic regression, showing the probability curve fitted using the data points. Ln Wikipedia, n.d., Retrieved August 17, 2015, from https://en.wikipedia.org/wiki/Logistic_regression. ©2015 by Michaelg2015. Adopted with permission.

Bayes' classifier

In these approaches, distribution of the predictors *X* are modeled in each of the response classes *Y*, and use Bayes' theorem to estimate P[$Y = k \mid X = x$]. When these distributions are assumed to be normal, these methods become discriminant analysis.

When assuming the variance of predictors are equal across all classes. The class prediction $\hat{Y}(x)$ is which the posterior probability is the largest:

$$\begin{aligned}\hat{Y}(x) &= \underset{k}{\operatorname{argmax}}\, logP[Y = k|X = x] \\ &= \underset{k}{\operatorname{argmax}}[x \cdot \tfrac{\mu_k}{\sigma^2} - \tfrac{\mu_k^2}{2\sigma^2} + \log(P[Y = k])] \qquad (d = 1) \\ &= \underset{k}{\operatorname{argmax}}[x^T \textstyle\sum^{-1} \mu_k - \frac{1}{2}\mu_k^T \textstyle\sum^{-1} \mu_k + \log(P[Y = k])] \ (d > 1),\end{aligned} \tag{25}$$

where d is the number of predictors. This method is called linear discriminant analysis (LDA), as the decision boundaries are linear.

When not assuming equal variance across all classes, the prediction is:

$$\begin{aligned}\hat{Y}(x) &= \underset{k}{\operatorname{argmax}}\, logP[Y = k|X = x] \\ &= \underset{k}{\operatorname{argmax}}[-\frac{1}{2}\log|\sum_k| - \frac{1}{2}x^T{\sum_k}^{-1}x + x^T{\sum_k}^{-1}\mu_k - \frac{1}{2}\mu_k^T{\sum_k}^{-1}\mu_k + \\ &\log(P[Y = k])]\end{aligned} \tag{26}$$

This method is called quadratic discriminant analysis (QDA), in which the decision boundaries are quadratic.

When assuming each the predictor X_i is conditionally independent of every other feature, the method becomes Naïve Bayes classifier.

In this study, the assumption of equal variance is not reasonable, and predictors are not independent to each other. Thus, QDA is used. The QDA function such as fitting, prediction were implemented in package "MASS" (Venables & Ripley, 2002) in R.

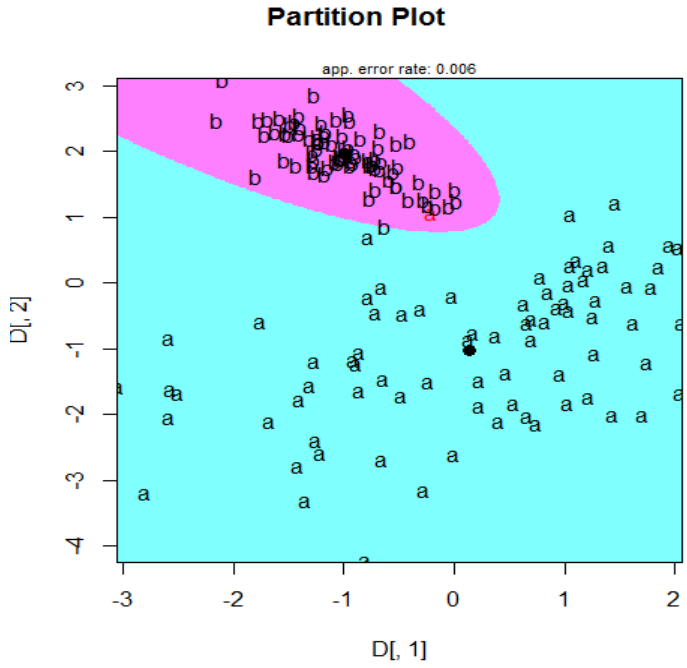


Figure 4-15. An example of QDA visualization using generated data, showing the data points and the quadratic decision boundary of two classes.

Artificial neural network

A neural network emulates the structure of a human brain as a network of neurons that are interconnected to each other. In classification problems, each neuron is equivalent to a logistic regression unit in general. The learning happens via an iterative feedback mechanism where the error of training data output is used to adjust the corresponding weights of input and the coefficients of each neuron. This learning algorithm is known as “backpropagation”. The advantage of this method can create complex decision boundaries compared with directly using LR, QDA.

The tuning parameters in a neural network include the number of hidden layers, the number of neurons in each layer, as well as learning rate. Larger layer number will lead to more complex decision boundaries, but will require longer training time. Higher learning rate will speed up the learning procedure, but results in a risk of failing to converge.

The neural network functions such as fitting, prediction were implemented in package “neuralnet” (Günther & Fritsch, 2010) in R.

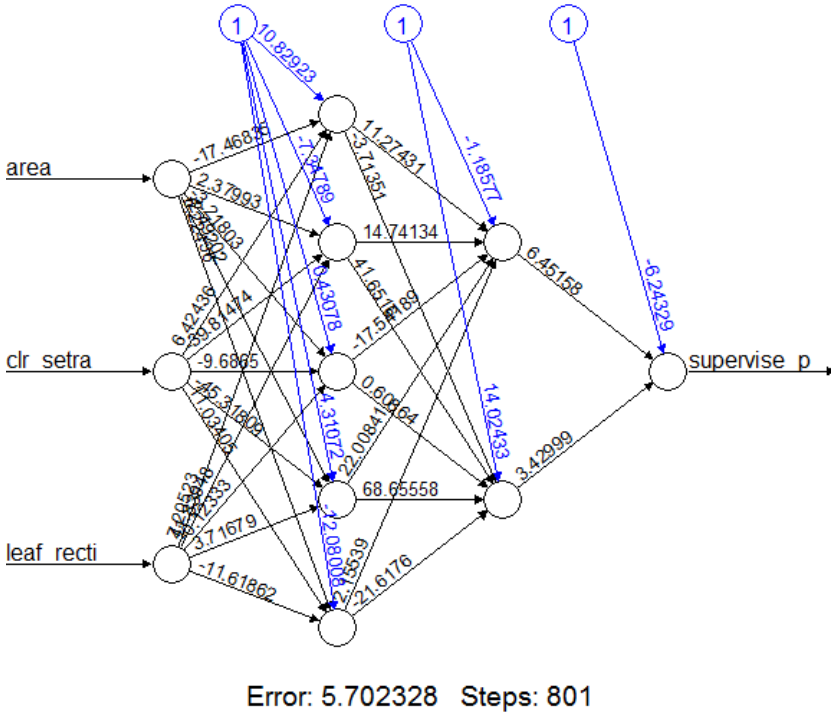


Figure 4-16. An example of three-layer artificial neural network visualization, showing the structure of the network, and the coefficients of each neuron. Each circle indicates a neuron, and the values above arrows indicates the weight of each input in each neuron.

Support vector machine

Support Vector Machine (SVM) is a binary classification method based on finding a hyperplane between a set of samples with positive and negative outputs, when assuming the data is linearly separable. It is a non-probabilistic binary classifier. In addition to performing linear classification, non-linear classification can also be performed by mapping data points into high-dimensional feature spaces, which is called “kernel trick”. However, operators still need to know the distributions of the data point before deciding which kernel to use.

The tuning parameters in SVM include the kernel type, the parameter needed for kernels, as well as the cost of constraints violation using in Lagrange formulation. The SVM functions such as fitting, prediction, as well as parameters tuning were implemented in

package “e1071” (Meyer, Dimitriadou, Hornik, Weingessel, & Leisch) in R.

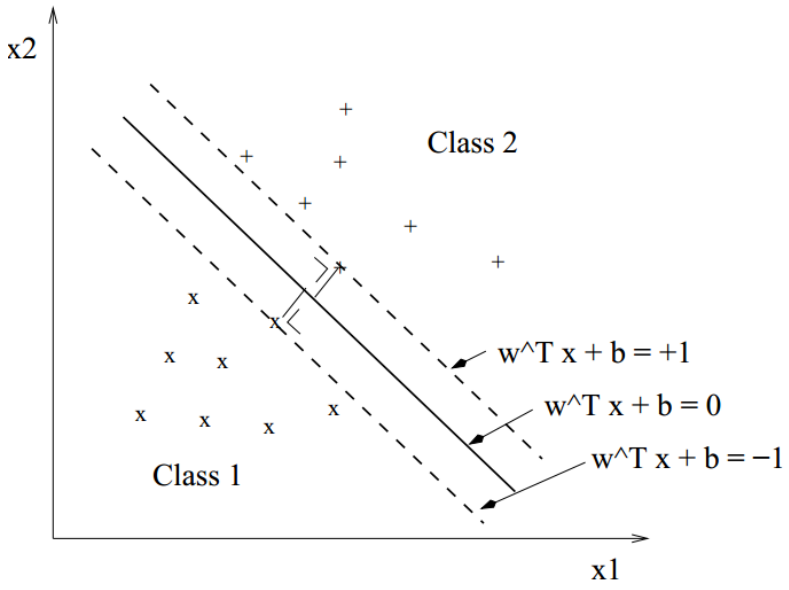


Figure 4-17. An example of linear SVM classification. The problems is to find a hyperplane with maximum margin using data points.

Random forest

Random forest is one of the tree-Bagging (Bootstrap Aggregation) models, which is built based on decision trees. In the Random Forest, it randomly selects k input predictors from the total N input predictors (k ~ N^0.5), then learns a decision tree from the data. After sampling for several times, different decision trees can be learned, and therefore a forest is built. During predicting, each tree in the forest votes for the result. This method is popular because it is non-parametric, robust, and is compatible with non-linear data.

The tuning parameters include the number of trees, and the depth of each tree. Higher tree number and tree depth will increase the complexity of the decision boundary, but also result in overfitting. The Random forest functions such as fitting, prediction are implemented in package “randomForest” (Liaw & Wiener, 2002) in R.

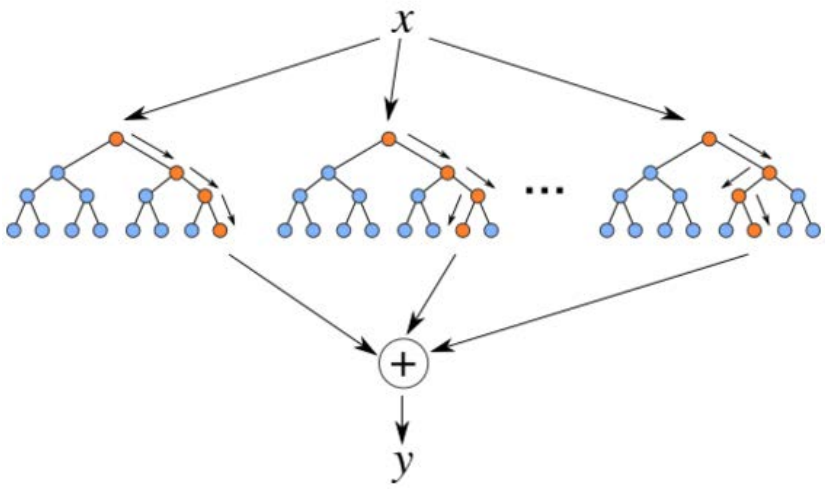


Figure 4-18. The concept of forest, In Random forest, different trees are trained with random predictors combinations. When predicting, voting is performed by each tree in the forest. Adopted from Random Forest とその派生アルゴリズム, ln Hatena Blog, n.d., Retrieved Dec 4, 2013, from http://kazoo04.hatenablog.com/entry/2013/12/04/175402 ©2013 by kazoo04. Adopted with permission.

AdaBoost

Boosting methods are methods that can combine the outputs of many “weak” (error rate is only required to be better than chance) classifiers to produce a powerful “committee” (Hastie et al., 2009). AdaBoost, short for “Adaptive Boosting”, formulated by Freund & Schapire (1997) who won the Godel Prize in 2003, is the most popular boosting algorithm.

The purpose of AdaBoost is to sequentially apply the weak classification algorithm to repeatedly modified versions of the data, therefore, produce a sequence of weak classifiers (trees are mostly used). After that, in prediction, all the weak classifiers are combined through a weighted vote to produce the final prediction. There are several variants of AdaBoost known, such as Discrete AdaBoost, Real AdaBoost, Logit AdaBoost, and Gentle AdaBoost. All of them are very similar in their overall structures. Figure 4-19 shows the

algorithm of Discrete AdaBoost as an example.

The tuning parameters of AdaBoost include the variant selection, the maximum depth of the trees in each iteration, the iteration number, and the shrinkage parameter. Since there are too many tuning parameters, only a local optimum parameter set was found, which is not guaranteed to be the global optimum. The AdaBoost functions such as fitting, prediction were implemented in package "ada" (Culp, Johnson, & Michailidis, 2006) in R.

Algorithm: Two-class Discrete AdaBoost

Input: M Samples (x_i, y_i) where $x_i \in \chi$, $y_i \in \{-1, +1\}$
Output: Final hypothesis $H(x)$
Procedure:

1. Initialize sample weight for each sample $W_1(i)=1/m$ for $i = 1,\ldots,m$.
2. for each iteration $t = 1, .. , T$
3. Train weak classifier using weight W_i
4. Get weak hypothesis $h_t : \chi \in \{-1, +1\}$ using the classifier
5. Compute error $err_t = E_w[1_{(yi=ht(xi))}]$
6. Choose classifier weight for current iteration $\alpha_t = \frac{1}{2}\ln(\frac{1-err_t}{err_t})$
7. for each sample, update sample weight:
8. $W_{t+1}(i) = W_t(\mathrm{i})\exp(-\alpha_t y_i h_i(x_i))$
9. end for
10. Renormalize W_{t+1}, so that $\sum W_{t+1} = 1$
11. end for
12. Output the final hypothesis:
13. $H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$

Figure 4-19. The algorithm of two-class discrete AdaBoost.

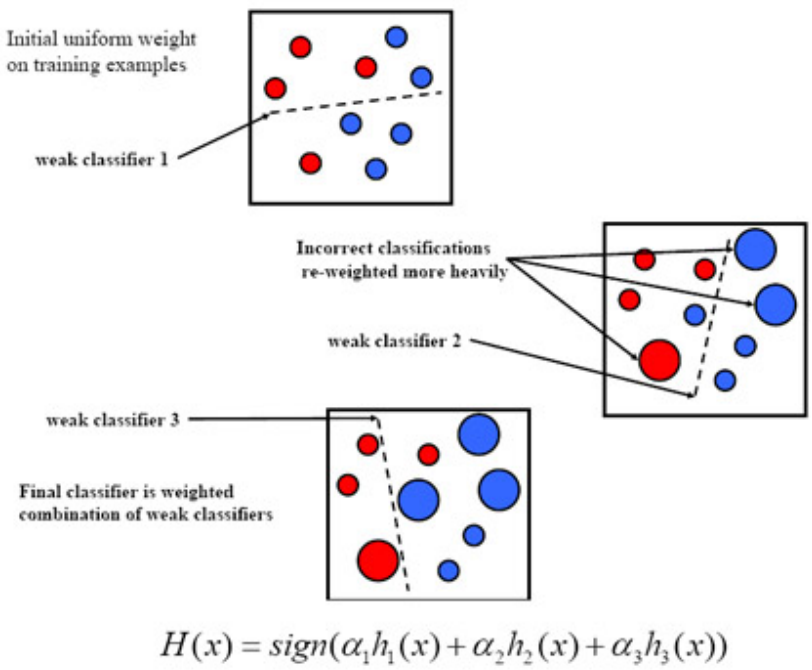


Figure 4-20. The concept of Adaboost and the procedure to ensemble weak classifiers (e.g. h_1, h_2 and h_3) into a strong classifier H. The size of each circle indicates the weight in each iteration. The prediction is based on the weighted votes of those weak classifiers. Adopted from Computer Vision Final Project: Viola-Jones & Morphology-based Face Detector, Retrieved 2005, from http://www.cc.gatech.edu/~kihwan23/imageCV/Final2005/CS7495%20Computer%20Vision%20Final%20Project.pdf ©2005 by Kihwan Kim. Adopted with permission.

4.5 Experimental Results

Currently, only two species: broccoli and lettuce, can be classified. When extracting features, leaf features could be extracted for most of the broccoli leaves, but not for the lettuce leaves. Thus the strategies of grouping and classification are both different for those two species. The algorithm was tested on all the crop-plant candidates detected with the algorithm described in the companion paper, in which the broccoli and lettuce datasets were collected in the horticulture research station of the Iowa State University. The data collection

details and the agronomic trail was described in the companion paper as well. In this section, the results of grouping and classification will be reported, and results of each step for these two species are introduced separately.

4.5.1 Grouping & localization

For broccoli, the strategy is to segment the leaves first, then used the venation to separate connected plants, and find the center at the same time. For lettuce, as the leaves are not easy to extract, only the pixel distributions were used to separate connected plants, and the center of each plant was obtained by finding the geometric center of the plant cluster. Figure 4-21 shows some run-time grouping results, in which connected plants within the same candidates were separated. And Table 4-1 shows the average localization error for broccoli and lettuce data collected at the different time, in which the golden values were obtained by human inspection using manually selecting method frame by frame in images using the supervising GUI.

Table 4-1. The average localization error for broccoli and lettuce at different growth stages.

Collection data	Days after transplant	Broccoli localization error	Lettuce localization error
June 13, 2016	12	14.8 mm	5.2 mm
June 17, 2016	16	10.6 mm	6.9 mm
June 23, 2016	22	17.2 mm	9.9 mm
June 27, 2016	26	21.1 mm	12.1 mm
Average		15.9 mm	8.5 mm

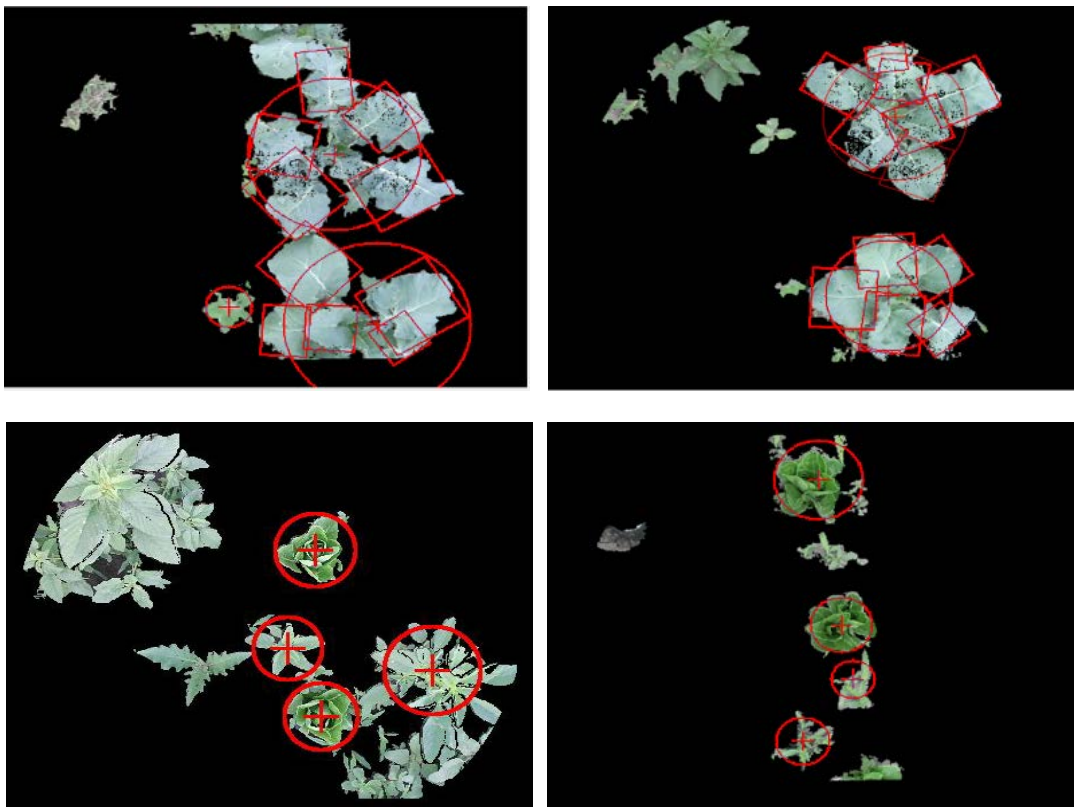


Figure 4-21. Examples of the plants grouping and localization run-time result. In those images, the detected and separated plants are circled the positions are labeled with crosses. For detecting Broccoli, segmented leaves are also labeled with squares.

4.5.2 Crop/weed classification

Seven classification models were applied in this study: logistic regression (LR), k-nearest neighbor (KNN), artificial neural network (ANN), Bayes classifiers QDA, support vector machine (SVM), random forest and tree-based AdaBoost. After fitting models with different methods and tuned to minimize the CV result, the best performance of each model was recorded.

Broccoli classification

The broccoli classification was separated into two stages in this study, one was leaf

classification using leaf features, and the other was plant classification.

In leaf classification, with paired plots of every variable (Figure 4-22), three predictors: leaf area, leaf length, leaf width, can be observed highly correlated. PCA (principal component analysis) was performed to reduce those three predictors into two, named pc_1 and pc_2.

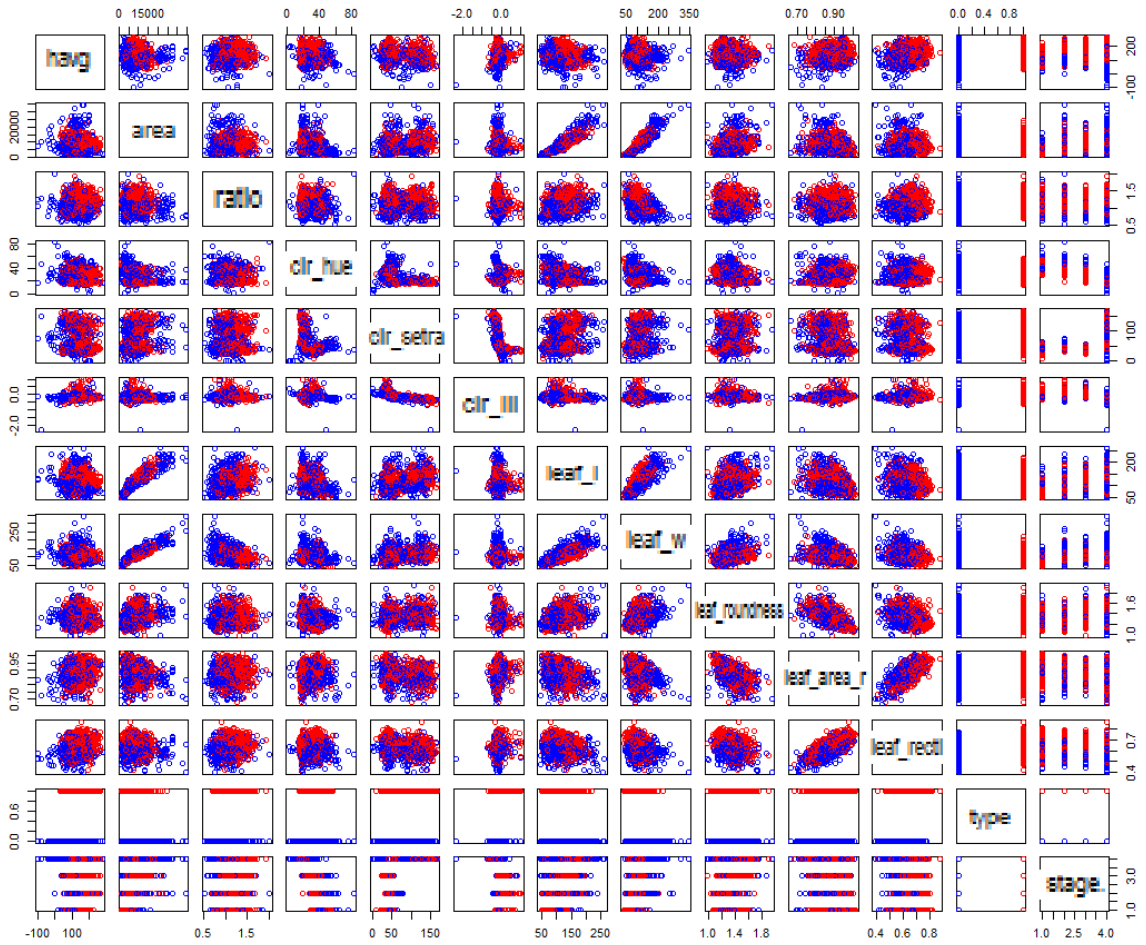


Figure 4-22. Paired plot of predictors used in leaf classification stage of broccoli classification, showing the distribution of the training data. Red data points were supervised as crop leaves, and blue data points were supervised as weed leaves or incorrect segmentation.

In leaf classification, with paired plots of every variable (Figure 4-22), we can observe that three predictors: leaf area, leaf length, leaf width are highly correlated. Principal

component analysis (PCA) is performed to reduce those three predictors into two, named pc_1 and pc_2.

As a result (Table 4-2), AdaBoost performed the best in leaf classification with the minimum CV error. After classifying each leaf, for each separated plant, the classified crop leaf parameters was used to form a new set of predictors for crop/weed classification. In this study, the Maximum, Minimum, Mean, Median, Standard deviation were considered for each leaf parameter as a new predictor. By testing their variance across classes, the max(leaf height), min(pc_1), max(pc_2), max(ratio), max(leaf roundness), max(leaf area rate), max(leaf Rectangularity) were chosen. The color features were found not stable for leaves. Thus they were not used for plant classification.

Table 4-2. Models evaluation and comparison result for broccoli leaf classification. Listing tuning parameters, training errors and CV errors. Adaboost model yields the best performance.

Model	Tuning parameter	Training error	CV error
LR	None	0.052	0.086
ANN	Layer = (5, 2)	0.051	0.154
KNN	K=5	0.083	0.113
SVM	Kernel = radial Gamma = 0.01 Cost = 100	0.053	0.107
QDA	none	0.064	0.095
RandomForest	N=500	0.138	0.170
AdaBoost	TreeDepth=4 Iter = 100 Nu = 0.2	0.057	0.073

In plant-scale crop/weed classification, the canopy features as well as the classified crop-plant leaf features of each separated plant were used. However, for some plants, as the leaves were failed to be segmented, the plant classification became a problem with missing

data, for which only some tree-based classification methods (Decision tree and AdaBoost) were compatible. Thus, in this study, LR, KNN, ANN, SVM, QDA, RandomForest were tested without leaf parameters, and AdaBoost were tested with leaf parameters.

As a result, AdaBoost performed the best in broccoli plant-scale classification and achieved an error rate of 3.1% in average during Cross-Validation, with the false positive rate of 1.1% and the false negative rate of 2.0% (Table 4-3).

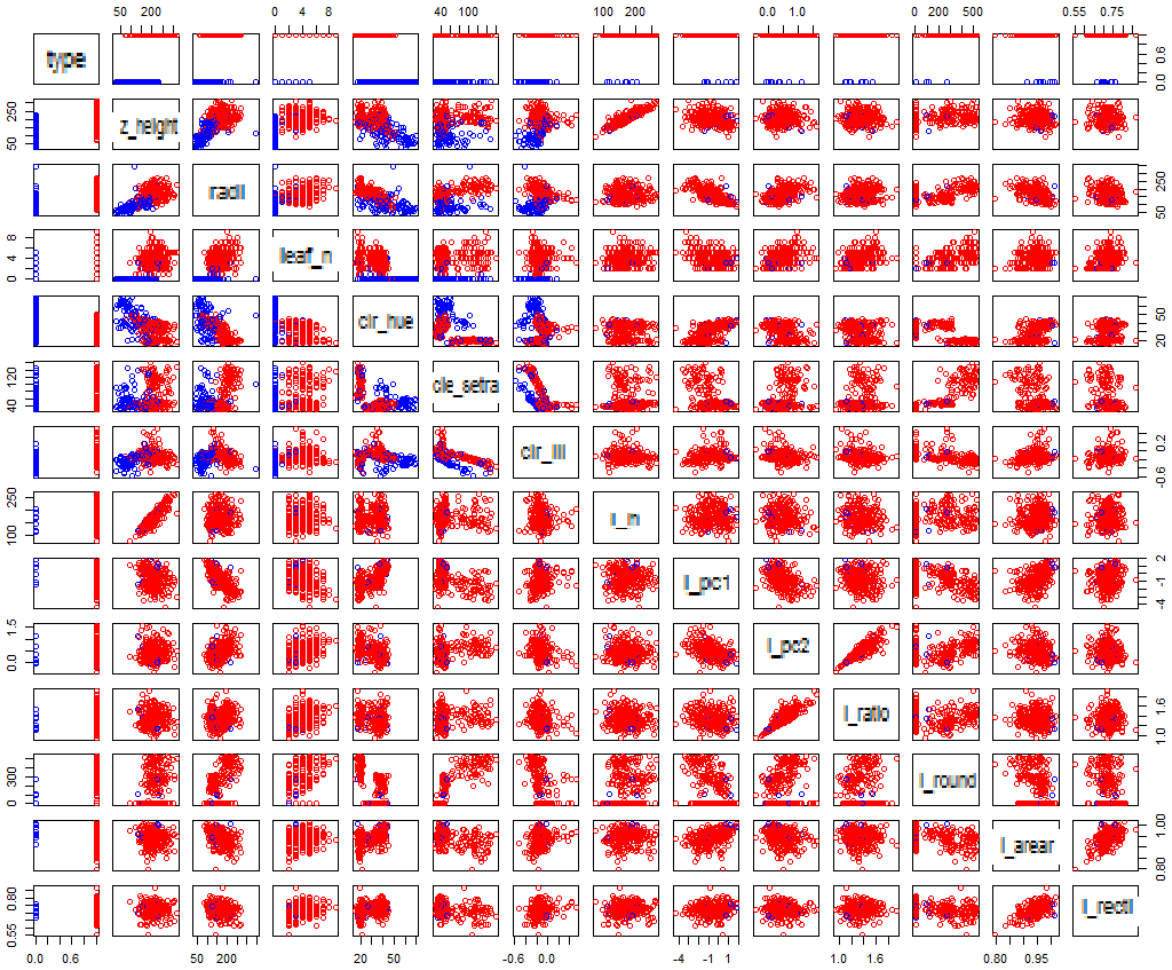


Figure 4-23. Paired plot of predictors used in plant-scale crop/weed classification in the broccoli dataset, showing the distribution of the training data. Red data points were supervised as crops, and blue data points were supervised as weeds or incorrect segmentation.

Table 4-3. Models evaluation and comparison result for broccoli plant classification, listing tuning parameters, training errors and CV errors.

Model	Tuning parameter	Training error	CV error
LR	None	0.072	0.102
ANN	Layer = (5, 2)	0.037	0.119
KNN	K=3	0.105	0.143
SVM	Kernel = radial Gamma = 0.01 Cost = 100	0.036	0.056
QDA	none	0.044	0.104
RandomForest	N=500	0.077	0.084
AdaBoost	TreeDepth=4 Iter = 50 Nu = 0.1	0.026	0.031 (FPR=0.011, FNR=0.020)

Lettuce classification

Since only a few leaves can be extracted from the lettuce dataset, the plant-scale classification was performed only, using only canopy features of the separated plants (Figure 4-24).

As a result, AdaBoost performed the best in lettuce plant-scale classification and achieved an error rate of 6.8% in average during Cross-Validation, with the false positive rate of 4.0% and the false negative rate of 2.8%. (Table 4-4).

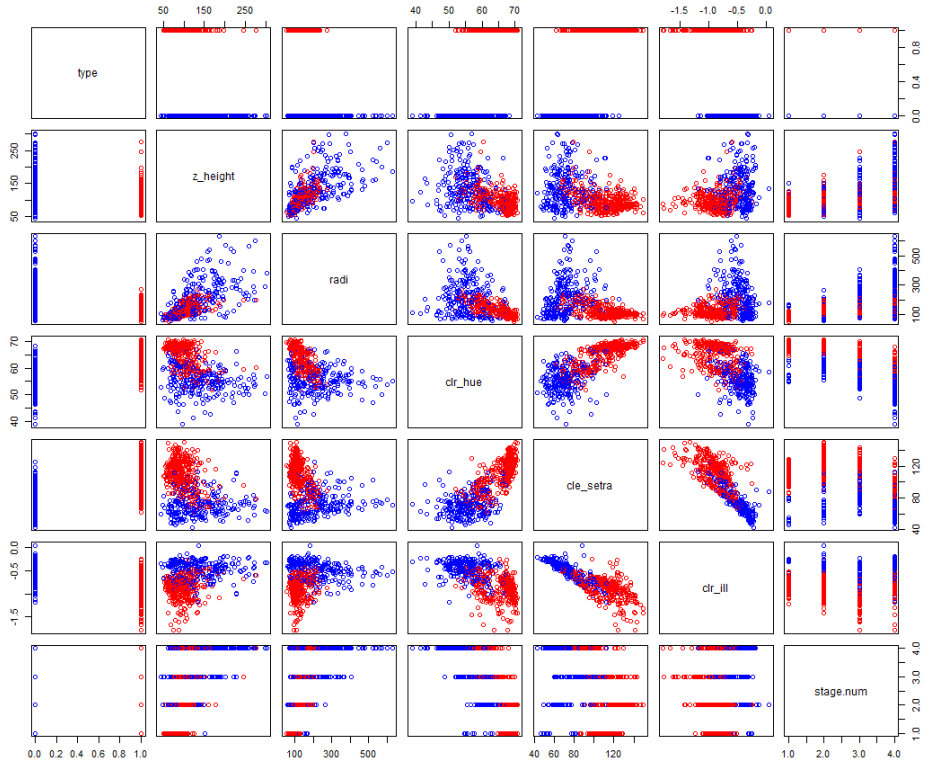


Figure 4-24. Paired plot of predictors used in plant-scale crop/weed classification using the lettuce dataset, showing the distribution of the training data. Red data points were supervised as crops, and blue data points were supervised as weeds or incorrect segmentation.

Table 4-4. Models evaluation and comparison result for lettuce plant classification, listing tuning parameters, training errors and CV errors. AdaBoost performed the best in plant classification.

Model	Tuning parameter	Training error	CV error
LR	None	0.112	0.134
ANN	Layer = (5, 2)	0.079	0.173
KNN	K=5	0.105	0.135
SVM	Kernel = radial Gamma = 0.01 Cost = 100	0.087	0.154
QDA	none	0.072	0.115
RandomForest	N=500	0.097	0.127
AdaBoost	TreeDepth=4 Iter = 50 Nu = 0.1	0.051	0.068 (FPR=0.040, FNR=0.028)

4.6 Discussion and Conclusion

In this study, a feature extraction and classification system was developed for robotic weeding of broccoli and lettuce. The problem of connected plants was solved by grouping using features extracted. With features from plant color and morphology extracted with image processing, crop plants at different growth stages can be discriminated with non-crop objects. Different classification methods were also evaluated and compared in this application.

The average localization errors for broccoli and lettuce were 15.9 mm and 8.5 mm, respectively. By evaluating and comparing different classification methods, AdaBoost was found the best in this application. After feature extraction and classification, crop plants can be recognized with error rates of 3.1% and 6.8% with AdaBoost for broccoli and lettuce in average, respectively. With the crop-plant candidate detection result presented in the companion paper, the average detection rates of 93.1% and 92.3% were obtained for broccoli and lettuce weeding, with average false discovery rates of 1.1% and 4.0%, respectively. The results show that this plant discrimination system is feasible for use in robotic weeding on broccoli and lettuce.

The reasons why AdaBoost performed the best are: Firstly, crop/non-crop classification is a two-class classification problem, with non-linear decision boundary. Crops are in the same patterns, and others are all non-crops. Then using linear models (LR, QDA, ADNN, SVM) with sigmoid-shaped responses is not suitable. As Adaboost was designed to create complex decision boundaries, it is more suitable in this study. Secondly, in this study,

only about ten predictors are used, with limited information available. Thus, Random Forest, which is mainly used with many predictors available and to find the most valuable features, is less suitable than AdaBoost.

The classification error comes from four sources: The first is sensor error. As the data is collected outdoors with Kinect v2 sensor, the noise can affect the accuracy even with sunlight shaded. Also, the resolution of depth image is still limited. The second is from the image processing algorithms. As the situations are complex in the field, the algorithms such as ground fitting and leaf extraction may have inaccurate results, then affect the accuracy of extracted features. The last source is that insufficient features are extracted. The features extracted in this study still cannot fully parameterize the crops and non-crops including all differences.

In the future, the image processing algorithm should be enhanced to be more robust, thus extract features more accurately. More features addressing the differences between crops and non-crops should be developed. In addition, since the deep learning algorithm has a great potential in detection and recognition, it should be tried in the future.

4.7 References

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation Learning: A Review and New Perspectives (Vol. 35.8, pp. 1798-1828): *Pattern Analysis and Machine Intelligence, IEEE Transactions*.

Du, J., Wang, X., & Zhang, G. (2007). Leaf shape based plant species recognition. *Applied Mathematics and Computation*, 883-893.

Fankhauser, P., Bloesch, M., Rodriguez, D., Kaestner, R., Hutter, M., & Siegwart, R. Kinect v2 for Mobile Robot Navigation: Evaluation and Modeling.

Finlayson, G. D., Hordley, S. D., & Drew, M. S. (2002). *Removing Shadows from Images*. Paper presented at the Computer Vision — ECCV 2002.

Freund, Y., & Schapire , R. E. (1997). A decision-theoretic generalization of on-line learning

and an application to boosting (pp. 119-139). Journal of Computer and System Sciences.

Hastie, T., H. Friedman, J., & Tibshirani, R. (2009a). Logistic Regression *The elements of statistical learning* (pp. 119): Springer-Verlag.

Hastie, T., H. Friedman, J., & Tibshirani, R. (2009b). *The elements of statistical learning:*

Data Mining, Inference, and Prediction: Springer-Verlag.

Hinton, G. E., Osindero, S., & Teh, Y. (2006). A fast learning algorithm for deep belief nets. (Vol. 18, pp. 1527-1554): *Neural Computation*.

Seide, F., **Li, G.**, & **Yu, D.** (2011). **Conversational speech transcription using context-dependent deep neural networks** (pp. 437-440): **in Proc. Interspeech**.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2014). Going Deeper with Convolutions: Cornell University Library.

Weiss, U., Biber, P., Laible, S., Bohlmann, K., & Zell, A. (2010). ***Plant Species Classification using a 3D LIDAR Sensor and Machine Learning***. Ninth International Conference on Machine Learning and Applications.

Wu, Q., Zhou, C., & Wang, C. (2006). Feature Extraction and Automatic Recognition of Plant Leaf Using Artificial Neural Network. *Advances in Artificial Intellingence* 3-10.

Wu, S. G., Bao, F. S., Xu, E. Y., Wang, Y.-X., Chang, Y.-F., & Xiang, Q.-L. (2007). A Leaf Recognition Algorithm for Plant

Classification Using Probabilistic Neural Network (Vol. *IEEE International Symposium on*. IEEE, 2007.): *Signal Processing and Information Technology*.

CHAPTER 5. SUMMARY AND CONCLUSIONS

This study consisted of two main efforts, the first was to develop an image processing system based on the fusion of 2D color and 3D spatial images, and explore the possibility to accomplish crop-plant candidate detection accurately by segmentation and clustering. The second was to develop a classification system using supervised machine learning technique, with features from plant color, morphology and structure, to accomplish discrimination of crop plants at different growth stages with non-crop objects.

In the first section, the accuracy of the Kinect v2 sensor was evaluated, then thousands of pictures were taken from different crop species with weeds existing. Data collection was performed at different growth stages of the crop plants, from germination or transplantation to maturity. Using the data collected, image processing algorithms were developed and implemented for broccoli and lettuce to segment the crop-plant candidates.

In the second problem, a leaf-and-canopy feature extraction algorithm was developed and implemented which also addressed the problem of connecting plants. A classification system was developed using supervised machine learning technique, with features of plant color and morphology extracted, to accomplish the discrimination of crop plants at different growth stages with non-crop objects (e.g. weeds). Different classification methods were also evaluated and compared in this application, and AdaBoost was found to perform the best in this application.

After testing the algorithm with the broccoli and lettuce dataset, detection rates (true

positive rate) of 93.1% and 92.3% with average false discovery rates of 1.1% and 4.0% for broccoli and lettuce were obtained, respectively. And for the actual crop plants, the average localization errors were 15.9 mm and 8.5 mm for broccoli and lettuce, respectively. The results indicate that the developed system has the ability to accomplish crop plant discrimination and localization for broccoli and lettuce in the field. The performance still needs to improve.

There are several error sources contributed to misdetection error and localization error: the first is the error from the Kinect v2 sensor, since working outdoors will increase the noise level, even with sunlight shaded; The second is that the with the disturbances from weeds, sometimes the image processing algorithm will fail in detecting the ground and extracting leaves. The third is that the ground is not strictly flat in the field, which will also affect the ground fitting results.

The machine-learning based classification error came from four sources: The first is the sensor error in data collection. As the data is collected outdoors with Kinect v2 sensor, the noise can still affect the accuracy even with sunlight shaded. In addition, the resolution of depth image is still limited. The second is from the image processing algorithms. As the situations are complex in the field, the algorithms such as ground fitting and leaf extraction may have the inaccurate result, then affect the accuracy of extracted features. The last source is that insufficient features are extracted. The features extracted in this study still cannot fully parameterize the crops and non-crops including all differences.

In the future, the system’s performance can be improved by enhancing current image

processing algorithms, extracting more features, as well as developing algorithms for other species. In addition, since the deep learning algorithm has a great potential in detection and recognition, it should be tried in this project in the future.