University of Windsor

Scholarship at UWindor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

2007

A fruit recognition method for automatic harvesting

Linghe Yang

University of Windsor

Follow this and additional works at: <https://scholar.uwindsor.ca/etd>

Recommended Citation

Yang, Linghe, "A fruit recognition method for automatic harvesting" (2007). *Electronic Theses and Dissertations*. 4613.

<https://scholar.uwindsor.ca/etd/4613>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

A FRUIT RECOGNITION METHOD FOR AUTOMATIC HARVESTING

**By
Linghe (Steven) Yang**

A Thesis

**Submitted to the Faculty of Graduate Studies through the
Department of Electrical and Computer Engineering in Partial
Fulfillment of the Requirements for the Degree of Master of
Applied Science at the University of Windsor**

Windsor, Ontario, Canada

2007

© 2007 Linghe (Steven) Yang

Library and
Archives Canada

Bibliothèque et
Archives Canada

Published Heritage
Branch

Direction du
Patrimoine de l'édition

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 978-0-494-34931-1

Our file Notre référence

ISBN: 978-0-494-34931-1

NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

© 2007 by Linghe (Steven) Yang

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transcribed, in any form or by any means — electronic, mechanical, photocopy, recording, or otherwise — without the prior written permission of the author.

ABSTRACT

Automation of harvesting is always one of the hottest topics in greenhouse operation. But before this, a reliable method of identifying mature fruit clusters on plants is required. This thesis presents a method to detect and recognize mature tomato fruit clusters on a complex-structured tomato plant containing clutter and occlusion in a tomato greenhouse. A color stereo vision camera is applied as the vision sensor. The proposed method performs a 3D reconstruction with the data collected by the stereo camera to create a 3D environment for further processing. The Color Layer Growing (CLG) method is introduced to segment the mature fruits from the leaves, stalks, background and noise. Target fruit clusters can then be located by depth segmentation. The experimental data was collected from a tomato greenhouse and the method is justified by the experimental results.

DEDICATION

This thesis is dedicated to

My mother who has given me life and inspiration,

My father who has always been my instructor and patient with me,

My friends who have been supporting and encouraging me.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Dr. Jonathan Wu, for his precious suggestions, guidance and continuous support throughout the coursework and research.

Many thanks go to Dr. John Dickinson and Dr. Sherman Lang who cordially offered me a lot of critical and valuable ideas, suggestions and direct hands-on instructions. Their warm and generous helps have made my one-year work at the Integrated Manufacturing Technology Institute, National Research Council unforgettable and remarkable.

I would like to express my gratitude to my internal reader Dr. Kemal Tepe and external reader Dr. Shaohong Cheng for their great advice toward the fulfillment of this Master's thesis work.

I am also indebted to my fellow graduate students and friends for their friendship and support. I wish to credit Mr. Tong Wang for his outstanding work on 3D Plant Modeling and it was my pleasure to work with him.

I would also like to acknowledge the financial support provided by my supervisor Dr. Jonathan Wu and Ontario Centre of Excellence (OCE) throughout all my research.

TABLE OF CONTENTS

ABSTRACT.....	I
DEDICATION.....	II
ACKNOWLEDGEMENTS.....	III
LIST OF TABLES.....	VI
LIST OF FIGURES	VII
CHAPTER 1. INTRODUCTION	1
1.1 Introduction to Automatic Harvesting in Greenhouse Industry.....	1
1.2 Thesis Objective.....	3
1.3 Thesis Overview	5
CHAPTER 2. LITERATURE REVIEW	6
2.1 Vision System in Agriculture.....	6
2.2 Robotic Harvesting	7
2.3 Fruit Detection	9
CHAPTER 3. 3D IMAGE RECONSTRUCTION	13
3.1 3D Reconstruction Process	13
3.2 PGR BumbleBee2 3D Image Reconstruction.....	16
3.2.1 Disparity map and distance.....	16
3.2.1.1 Establishing Correspondence.....	18
3.2.1.2 Distance and Coordinate System	19
3.2.2 Data Flow in the Stereo Camera System	20
3.2.2.1 Preprocessing	20
3.2.2.1.1 Rectification.....	20
3.2.2.1.2 Edge Detection.....	22
3.2.2.2 Stereo Processing	23
3.2.2.2.1 Disparity Range	23
3.2.2.2.2 Correlation Mask	23
3.2.2.2.3 Validation Methods.....	27
3.2.2.2.4 Subpixel Interpolation.....	33

3.2.2.3	3D Position Calculation and Accuracy	33
CHAPTER 4. COLOR SEGMENTATION		38
4.1	Segmentation Definition	38
4.2	Color Segmentation Method	40
4.2.1	Hierarchical structure.....	41
4.2.2	Segmentation in island structures with color similarity and island distance	42
4.2.3	Color Layer Growing Algorithm	43
4.2.3.1	Preprocessing	43
4.2.3.2	Initialization	45
4.2.3.3	Linking.....	45
4.2.4	CLG examples	46
4.2.5	Comparison between CLG and CSC	48
CHAPTER 5. DEPTH SEGMENTATION		50
5.1	Depth segmentation based on disparity map (depth) image	50
5.1.1	Disparity map image	50
5.1.2	Point grey disparity map	53
5.2	Extracting Object Position from 3D Image	54
5.2.1	Level concept.....	54
5.2.1	Shape based recognition with 3D point cloud distribution.....	56
CHAPTER 6. EXPERIMENTS AND RESULTS.....		59
6.1	Operating Environment.....	59
6.2	Implementation and Results.....	60
CHAPTER 7. CONCLUSIONS AND FUTURE EXPECTATIONS		75
7.1	Conclusions.....	75
7.2	Future Work.....	76
BIBLIOGRAPHY		78
VITA AUCTORIS		81

LIST OF TABLES

Table 3.1 Mask size comparison.....	24
Table 3.2 The influence of mask to 3D reconstruction quality.....	26
Table 3.3 Correlation accuracy results for various resolutions and quality settings with stereo mask=11 and edge mask=9.....	36
Table 3.4 Correlation accuracy results for 320x240 resolutions, over a range of mask sizes.....	36
Table 4.1 Comparison of three filters.....	44
Table 6.1 Comparison between Measured distance range and calculated distance.....	73

LIST OF FIGURES

Figure 1.1 Tomato Greenhouse Task Model.....	2
Figure 3.1 3D Reconstruction Procedure.....	14
Figure 3.2 Camera Module Configurations.....	15
Figure 3.3 Example of matching points between stereo images.....	16
Figure 3.4 Disparity in stereo image.....	17
Figure 3.5 Correspondence Establishment.....	18
Figure 3.6 Image and world coordinate systems.....	19
Figure 3.7 Data flow.....	20
Figure 3.8 Raw (a) and rectified (b) image pair.....	22
Figure 3.9 3D point cloud with different mask size: (a) mask=5 (b) mask=15 (c) mask=23.....	26
Figure 3.10 texture validation with threshold = 2.0.....	27
Figure 3.11 Uniqueness validation with threshold = 0.5.....	29
Figure 3.12 3D point cloud (a) without surface validation (b) with surface validation (Size = 300, Difference = 0.5).....	33
Figure 4.1 Segmentation Classification.....	40
Figure 4.2 Islands of three different levels in hierarchical structures based on (a) orthogonal, (b) triangular and (c) hexagonal topology.....	41
Figure 4.3 Original color image and stereo image.....	47
Figure 4.4 2D and 3D images after color segmentation.....	48
Figure 5.1 Stereo image.....	51

Figure 5.2 Tomatoes in the office.....	52
Figure 5.3 Disparity image.....	53
Figure 5.4 Histogram of raw disparity map.....	54
Figure 5.5 Image plane levels.....	55
Figure 5.6 3D point cloud distribution in level and distance.....	57
Figure 5.7 3D image after color segmentation.....	58
Figure 6.1 Greenhouse Operating Environment.....	60
Figure 6.2 Mature fruit cluster detection and localization at 32~35cm cm (close distance), (a) raw-right image, (b) image after depth filter, (c) image after color filtering, (d) image after color segmentation, (e) image with mature cluster position, (f) Z distance histogram.....	64
Figure 6.3 Mature fruit cluster detection and localization at 65~68cm (far distance), (a) raw-right image, (b) image after depth filter, (c) image after color filtering, (d) image after color segmentation, (e) image with mature cluster position, (f) X distance histogram.....	68
Figure 6.4 Mature fruit cluster detection and localization at 50~53cm (medium distance), (a) raw-right image, (b) raw-left image, (c) image after depth filter, (d) image after color filtering, (e) image after color segmentation, (f) image with mature cluster position, (g) Z distance histogram.....	73
Figure 6.5 Comparisons between Measured and Calculated Distance of Fruit Clusters.....	74

CHAPTER 1. INTRODUCTION

1.1 Introduction to Automatic Harvesting in Greenhouse Industry

Robotic harvesting has been studied by numerous researchers across the world. Nevertheless, few developments have been adopted and put into practice. The reasons for this might be because of technical, economic, horticultural and producer acceptance issues.

In Canada, large quantities of tomatoes are produced in greenhouses in Southern Ontario and British Columbia. Currently, labour and energy costs are the largest costs incurred by a modern greenhouse operation (with each making up about 35% of operational costs). To deal with saturated market demands and fierce competition, growers are always looking for ways to improve their overall process efficiency. Hence, improving the labour efficiency or even reducing the amount of required human labour is a key issue today. Manual labour in a greenhouse is demanding, especially under poor climatic conditions. Thus, it is not unexpected that the workers may require higher wages in the future. Therefore, it is becoming more and more difficult to obtain adequate staff while there is a need to limit the cost of labour. An efficient robotic system for even one task could reduce labour costs significantly and this is one of the reasons why the use of an automated robotic system for harvesting is so attractive.

In greenhouse operations, plant tasks are divided into two main tasks for greenhouse tomato crop, which are maintenance and harvesting. Maintenance includes tasks such as clipping/winding, deleafing, lowering, support, and pruning. Their relationships are depicted in Figure 1.1. Normally, harvesting occupies a significant amount of time and demands a great number of workers to complete this task. Therefore, it becomes my immediate task to replace manual harvesting with robotic harvesting machines.

Figure 1.1 Tomato Greenhouse Task Model

There are several techniques used for harvesting fruits which are not appropriate for the fresh fruit market due to the damage caused to the fruit during its collection. These techniques include the shaking of tree limbs or tree trunks, oscillating forced-air removers and the complementary chemical treatment. Fruits are usually bruised when striking plant limbs during the landing. Hence, there is a need for a non-aggressive method to perform the harvesting of fruits as delicately as possible such as robotic harvesting

Robotic harvesting can be divided into two stages. The first one is machine vision, i.e., fruit detection, recognition and localization. And the second stage is machine grip integrated with machine vision to constitute a complete robotic harvesting. This thesis focuses on the first stage.

The use of computers to analyze images has many potential applications for automated agricultural tasks. But, the variability of the agricultural objects makes it very difficult to adapt the existing algorithms to industrial application. Agricultural systems must be flexible, and methods for including domain knowledge in algorithms should be studied as a rational way to cope with domain variability.

There are many processes in agriculture where decisions are made based on the appearance of the product. Applications for grading the fruit by its quality, size or ripeness are based on its appearance, as well as a decision on whether it is healthy or diseased. Humans are easily able to perform intensive tasks like harvesting and pruning using basically the visual sensory mechanism. This suggests that a system based on a visual sensor should be able to emulate the interpretation process of the human visual recognition system.

In machine vision, the objective is to empower a robot with one or more vision sensors. In our research work we have employed a color stereo vision camera, which can work as human eyes to locate the mature fruits in the complicated structured tomato plants.

Although a lot of methods have been studied by numerous researchers, few have been put into practice because the real greenhouse environment is much more complicated than the studied scenarios. With severe occlusion and under natural lighting, detection is made extremely difficult and thus the success detection rate is lower than the satisfied level. This problem hasn't been solved by previous study for a long time. In our work, we integrated 3D reconstruction, color segmentation and depth segmentation (depth histogram) to recognize the mature fruit clusters and calculate their distance from the camera with XYZ values.

1.2 Thesis Objective

The purpose of this thesis is to detect, recognize and locate mature tomatoes in a complex-structured tomato plant using a color stereo vision camera possibly containing occlusion so that an automatic harvesting system can be applied in the greenhouse in order to save the labor costs, thus bring down the whole costs.

Many challenges occurred during the implementation and the main difficulties can be sorted into three aspects.

- One object in one of the stereo camera images might not appear in the other camera image, which adds difficulty to object recognition. 3D reconstruction helps to eliminate this scenario so that we can further detect tomatoes without stereo occlusion.
- The complicated structure of the tomato plant makes it tough to detect mature fruits with regular shape or texture methods. Normally recognition is applied in a simple environment with controlled lighting and will have a high accuracy in correct detection. In contrast, a greenhouse tomato plant has a complex structure. Stalk, stems, branches, leaves, fruits and background are all parts of the environment. However, color segmentation provides us with a practical method to segment mature fruits from complex structure.
- After locating the object in the image, we need to map it to its corresponding position in the 3D world, but direct mapping might be less accurate than required. Depth segmentation can offer more accurate position in this stage.

Explicitly, the primary goal is to find a novel method through which multiple mature fruit clusters can be recognized and their positions (XYZ) and distances can be retrieved. Finally, this machine vision system is implemented to test the accuracy and robustness of this novel method for mature fruit detection and localization.

1.3 Thesis Overview

This thesis is organized in the following sequence; after the general introduction in Chapter 1, Chapter 2 covers the literature review which summarizes the previous work. In Chapter 3, 3D reconstruction from a stereo camera is described in detail. Chapter 4 provides us with color segmentation method and then Chapter 5 presents the depth segmentation. The experimental results can be found in Chapter 6. Lastly, the conclusion and future expectations are presented in Chapter 7.

CHAPTER 2. LITERATURE REVIEW

2.1 Vision System in Agriculture

Current areas of image analysis in agriculture can be sorted into two main groups: Research tools and Decision making. And a topology breakdown of the research can be depicted as follows:

- I. Research Tools
- II. Decision Making
 - A. Guidance
 - 1. Harvesting
 - 2. Guiding machinery
 - B. Grading

The first group of image analysis systems includes applications like plant growth monitoring, morphometry of new cultivars or biological cell counts. These types of tools allow a researcher to efficiently gather data automatically. The user monitors the performance of the system and can intervene when the system misinterprets an image. These image processing tools also allow features to be measured automatically which would be too time-consuming to do manually.

The second group of image analysis systems provides information to guide the mechanical equipment. Such systems support two different groups of applications, Grading and Guidance. The use of image processing for grading is applied to many

products, including oranges, potatoes, apples, carrots, green peppers, tomatoes and peaches. The grading may be for size and shape, color, or the presence of defects. Current guidance research includes harvesting oranges, tomatoes, mushrooms, apples, melons and cucumbers. The guidance research also focuses its attention on navigating robot vehicles using machine vision strategies or other simple sensors in order to obtain autonomous mobile capabilities.

2.2 Robotic Harvesting

Various robotic harvesting methods with distinct mechanical designs were developed based on different kinds of fruits and environment, including citrus, apples, melons, grapes and etc [1].

In the citrus harvesting environment, the use of robots to pick fruits from trees was first proposed by Schertz and Brown [2] after reviewing mechanical citrus harvesting systems. The following three steps make up their approach for fruit picking: 1) visually locate the fruit with a vision sensor, 2) guide the fruit detachment device along the line of sight to the fruit, 3) actuate the device once the fruit is “touched”. A robotic system based on their approach was built for the harvesting of apples, including a simple robotic arm, a B/W TV camera and a control computer.

An Italian company, AID Catania, designed and built a prototype of a citrus harvesting autonomous robot with a single arm, directed by a vision system which was operated both in the laboratory and in the orange grove [4][5]. This robot has a cylindrical coordinate electrical driven arm which supports a smart end-effector. The end-effector is made of a mobile helix and a fixed cylinder. An infrared proximity sensor indicates that the orange is at the right distance. Around 65% of the located fruits were successfully detached. Plans to develop a multi-arm robot for automatic picking of oranges were also proposed in their work.

Harrell presented the design of a citrus picking robot CPR [6]. The robot consisted of a single arm with a spherical coordinate system. The rotating-lip picking mechanism (PM) included a small cavity at the end of the arm, a CCD video camera, an ultrasonic ranging transducer to provide range information to objects in front of the PM, light sources and the rotating lip to cut the stem of the fruit.

The Spanish-French CITRUS project to harvest oranges included an agronomical study, the development of a visual system to locate the fruit, the design and control of a harvesting arm, the integration of the grasping and cutting device with field test [8]. There were two versions of the robot: one with cylindrical coordinate system and a more sophisticated version with spherical coordinates. The grasping method employed a vacuum sucker to detach the fruit.

For automatic apple picking, D'Esnon and Rabatel [2] invented the first version of the apple picking robot, known as MAGALI, which consisted of a hollow tube mounted in a vertical support frame. Attached to the end of the tube was a rotating cup end-effector used to detach fruits from an apple tree canopy. A B/W camera was attached to the support frame to detect the fruit. When the fruit was detected, the tube was aligned with the fruit. The tube would extend out until the end-effector contact the target fruit. The cup would rotate behind, cut the stem and allow the detached fruit to roll down the hollow tube into a collection bin. The second version of the MAGALI robot was constructed in 1986 [3]. The new design included a spherical manipulator, a camera at the center of the rotation axes and a vacuum grasper. MAGALI is a hydraulically actuated vehicle, self-propelled and totally self-guided in the pathways by four ultrasonic telemeters.

The AUFO robot was developed at the Central Enterprise for the harvesting of apples [9]. To sweep the whole volume of the tree, the robot platform is moved around the tree by small angular shifts with 6 arms to pick up apples. The position of the apples was computed by a triangulation technique using two color cameras.

The harvesting of melons was studied and a prototype harvester was constructed to selectively harvest fruits [10]. The system consists of a robot with a Cartesian manipulator mounted on a frame moved by a tractor. The robot vision system is used to locate the melons and to guide the attaching device toward the fruit.

The Japanese company, Kubota [7] developed a fruit-picking robot which used a mobile platform to approximate a small four degrees-of-freedom manipulator to the detachment area. The gripper had a mobile vacuum pad to capture the fruit and to be directed by an optical proximity sensor, a stroboscope light and a color camera.

The AGRIBOT was a Spanish project [11] to harvest fruits with the cooperation of a human operator who was responsible for fruit detection. With a joystick, the operator could move a laser pointer until the laser spot is in the middle of the fruit. The 3-dimensional coordinates were recorded and the parallelogram manipulator was controlled toward the fruit. A gripper system based on a pneumatic attaching device and an optical proximity sensor were used to detach the fruit.

2.3 Fruit Detection

Selective harvest needs to determine the location, size and ripeness of individual fruits. Four basic features can be relied on to recognize a fruit: intensity, color, shape and texture. In the following paragraphs, a review of various approaches is presented. This review is sorted with regard to the techniques employed in order to understand the impacts and effects of different approaches though some authors combine multiple approaches.

1. Intensity/Color

Parrish [12], Sites [13] and Slaughter and Harrel [14] used intensity/color based methods to recognize fruits. In Parrish's research, thresholding is done to obtain a binary image, which is then smoothed to eliminate noise and irrelevant details in the image. An

elaborate roundness measure is carried out to get the centroid and radius values. The region can be accepted as an apple only when the candidate's density is greater than a preset threshold. Sites presents a recognition method consisting 5 steps: 1) thresholding based on a constant 37% value; 2) smoothing by a binary filter; 3) segmentation by an 8-neighbor connected component labeling; 4) feature extraction (area, perimeter, compactness, elongation); 5) classification by a linear decision function or a nearest-neighbor method. Slaughter and Harrel introduced a method to locate mature oranges based on color images. This system uses the Hue(color) and Saturation(intensity) components of each pixel obtained using a color camera and artificial lighting, yielding a two-dimensional feature space. Two thresholds are employed based on a maximum value for saturation and a minimum value for hue.

Weakness: A major problem in segmenting intensity or gray level images lies in the selection of the threshold value that distinguishes an object from the background. This value depends on the illumination of the scene and there is no a priori knowledge about it because the illumination conditions can vary randomly. For instance, a fruit in the sun can appear ten times brighter than a leaf in the sun, a fruit in the shade can appear four times dimmer than the leaf in the sun. When fruits are less ripe or if there are some non-leaf background objects, e.g. sky, false recognition problems will arise.

2. Shape

Whitaker[15] and Benady[16] applied shape information to detect fruit. Whitaker applied a Circular Hough Transform to binary images to detect circles in the images. The contour of leaves becomes one of the main problems, since false detection might recognize leaves as fruits. Also this algorithm can not be performed in real time. Benady uses a laser line projector to illuminate the scene. This line of light when contacting the surface of a melon is recorded as a curved line; the deformation of the initial straight line indicates the distance to the object by a triangulation analysis. This triangulation system is used to get one profile at preset offsets. These profiles (not contours) are analyzed using the Circular Hough Transform to obtain a matrix of votes indicating the candidates for being the

center of a melon. The expected size and shape help to detect the melons. In this system, all the fruits that were visually discernible were detected by the system, and no false detection occurred.

3. Texture

Qiu and Shearer [17] and Purdue University (USA) and The Volcani Center (Israel) [18] combined shape and texture techniques to recognize fruits. Texture analysis has been used and might be a way to locate some specific fruits. Some fruits have textures different from their leaves; some are smooth while others are rough. In [18], after obtaining intensity images of the melon crop and performing an image enhancement, a thresholding and a parameter extraction, a hypothesis generation is performed. Shape and texture parameters in the neighbourhood of the hypothesized position are computed to obtain the final candidates. Then a knowledge-directed evaluation is performed by using rules which attempt to eliminate multiple occurrences. Approximately an 84% success rate and a 10% false detection rate were achieved.

4. Other techniques

Schertz and Brown [19] suggested that the location of fruits might be accomplished by photometric information, specifically by using the light reflectance differences between leaves and fruits in the visible or infrared portion of the electromagnetic spectrum. Gaffney [20] determined that "Valencia" oranges could be sorted by color using a single wavelength band of reflected light at 660 nm.

Grand D'Esnon [4] developed a vision system for the MAGALI robot. This vision system required a protective covering to get a dark background. Three color CCD cameras and the three different filters (950, 650 and 550 nm) are used to obtain three intensity images. After a preliminary study based on the spectral properties of the apple trees' leaves and the apples, some ratio features (with reference to the image filtered at 950nm) are used to decide which pixels belong to a fruit or to a leaf.

The AID robot vision system [5] was implemented to recognize oranges by pre processing the color image with an electronic filter and locating the fruits by recognizing distributions of the orientation of maximum gradients. In this system, artificial lighting is used. An analog electronic filter enhances the image and during digitization, 6 bits are used to codify the pixel value which is proportional to how close the actual pixel hue is to a preset reference hue. With this pseudo-gray image, a gradient image and a direction image are computed using the Sobel operator. Finally, the scene interpretation is done through searching for a match with an object model previously stored. This gradient direction template is moved step by step throughout the direction image.

RGB segmentation is introduced by Slaughter and Harrel [21] based on their earlier study by using the RGB information recorded by a color camera as features and a traditional Bayesian classifier method to label pixels in the images belonging either to the fruits or to the background. CIRAA in Italy built a robotic system for greenhouse operation, called AGROBOT [22]. This system employs hue and saturation histograms to perform a threshold to segment the image.

In the previous works, vision sensor was usually a B/W camera or a color one. Stereo cameras or laser scanners were seldom applied to collect data for further processing to locate target objects, such as apples, oranges, melons, peaches, etc. In most cases, artificial lighting sources and optical filters are used.

CHAPTER 3. 3D IMAGE RECONSTRUCTION

In this chapter, 3D image reconstruction by using a PGR BumbleBee2 [23] stereo vision camera is depicted in detail. In Section 3.1, some key stages are listed and camera configurations are introduced before creating a 3D image. In Section 3.2, we present the concept of a disparity map, which is applied to establish the correspondence between a pair of stereo images. The entire data flow is described in Section 3.2.2, including preprocessing and stereo processing. A number of parameters are tested to compare the difference among the results in order to approximately optimize the output.

3.1 3D Reconstruction Process

The whole 3D reconstruction procedure using PGR Bumblebee2 [23] can be depicted as the following flow chart. The five key steps are highlighted and we will explain how they are carried out using the PGR system [23].

Figure 3.1 3D Reconstruction Procedure

Before preprocessing and stereo processing, we need to set camera module configuration, which can be explained in Figure 3.2.

Figure 3.2 Camera Module Configurations

After configuration, we can apply preprocessing and stereo processing to the image in order to get a disparity map for 3D reconstruction.

Preprocessing is the process of image smoothing, rectification and edge detection, as specified by parameters in camera configuration on the input data. It involves unpacking the data, which is stripping individual channels from 32-bit packed data and putting them into 3 raw image channels, then applies a low-pass filter on these channels if requested, and corrects for lens distortion and camera misalignment, saving these images as rectified images. Finally a second derivative Gaussian edge processing is performed on the rectified images. Stereo processing is the process of stereo matching, validation, and sub-pixel interpolation, as specified by parameters in camera configuration and generates the processed image. By applying triangulation [24], we are able to extract 3D image from disparity map. After selecting proper range, we reject points out of this range for 3D display. All detailed descriptions of generating 3D point cloud are unfolded below.

3.2 PGR BumbleBee2 3D Image Reconstruction

The purpose of stereo vision is to get range measurements based on images obtained from slightly offset cameras. There are three steps required in performing stereo image processing:

- Establish correspondence between image features in different views of the scene.
- Calculate the relative displacement between coordinates of the same features in each image.
- Determine the 3D location of the feature relative to the cameras, using the knowledge of the camera geometry.

3.2.1 Disparity map and distance

Consider the following example. The following figure shows an image pair obtained from the horizontally displaced cameras. We can identify two points A and B in both images. The point A_{left} corresponds to the point A_{right}. Similarly, point B_{left} corresponds to the point B_{right}.

Figure 3.3 Example of matching points between stereo images

After measuring the horizontal distance between the left edge of the images and the points, we find that distances in the left image are greater than the distances to the

corresponding points in the right image. For example, the distance to the phone handle from the left edge of the image is greater than the distance to the phone handle in the right image. Based on this difference (also called disparity) it is possible to determine the distance to the phone handle from the camera.

Disparity for the feature A will be defined as $D(A) = x(A_{\text{left}}) - x(A_{\text{right}})$ and the disparity of point B will be derived as $D(B) = x(B_{\text{left}}) - x(B_{\text{right}})$, where $x(A_{\text{left}})$ is the x coordinate of the point A_{left} . We calculated $D(A)$ and $D(B)$ and found that $D(B) > D(A)$. This indicated that point B in the scene is closer than point A.

In the following example, we impose one image on the other, both of which are captured by right and left camera of the same stereo camera. From this picture, we can observe the disparity more clearly. Red and green images are taken from right and left cameras respectively. For instance, the disparity of the chair is much larger than the disparity of the wires on the wall, since disparity of one point is in reverse relation with its real distance.

Figure 3.4 Disparity in stereo image

According to the above examples and description, we define the disparity as the difference between the coordinates of the same features in the left and right image. We find that the distances from the top of the image to the matching features are exactly the same in both images. This is because the cameras are horizontally aligned, therefore only the horizontal displacement is relevant.

3.2.1.1 Establishing Correspondence

The system establishes correspondence between images using the Sum of Absolute Differences correlation method [24]. The idea behind the approach is to implement the following steps:

Figure 3.5 Correspondence Establishment

Comparison of neighborhoods or masks is done using the following formula:

$$\min_{d_{\min}}^{d_{\max}} \sum_{i=-m/2}^{m/2} \sum_{j=-m/2}^{m/2} |I_{\text{right}}[x+i][y+j] - I_{\text{left}}[x+i+d][y+j]| \quad \text{Sum of absolute differences}$$

Where: $d_{\max}$ and $d_{\min}$ are the minimum and maximum disparities

m is the mask size

I_{right} and I_{left} are pixels in the right and left images

The selection of disparity range is up to us. Disparity only appears in x (columns) because the two cameras are lined up in the same row, resulting in close to zero horizontal disparity.

3.2.1.2 Distance and Coordinate System

The distances from the cameras are determined using the displacement between images and the geometry of the cameras. The position of the matched feature is a function of the displacement, the focal length of the lenses, resolution of the CCD and the displacement between cameras.

Figure 3.6 Image and world coordinate systems

The above figure illustrates the coordinate system in which the system represents images and the world measurements. The origin of the image is in the top left corner of the upright image. The origin of the world measurements is in the pinhole of the reference camera.

3.2.2 Data Flow in the Stereo Camera System

The following figure shows the data flow in the camera system. The process takes raw images obtained from the camera and produces depth images. There are two main processing blocks in the library. The first processing block is the image pre-processing block that applies a low-pass filter, rectifies the images and performs edge detection. The second processing block does stereo matching, validation of results and sub-pixel interpolation. The result of this process is a depth image.

Figure 3.7 Data flow

3.2.2.1 Preprocessing

The preprocessing module of the System prepares the raw images for stereo processing. Preprocessing allows specification of the processing resolution, and the Low-Pass Filtering functionality. In order to rectify the images it is important to smooth them. If image rectification is done it is a good idea to turn on the low pass filtering. Rectification can still be done with low pass filtering disabled but the rectified images will exhibit aliasing effects. Disabling the low-pass filtering can be done in order to speed up processing.

3.2.2.1.1 Rectification

Rectification is the process of correcting input images for the distortions of the lenses. Lenses often cause distortions in raw images. For example, straight lines in the scene will often appear curved in the raw images. This effect will be particularly evident in the corners of the images. Rectified images will be corrected for these kinds of distortions. The following example shows the difference before and after rectification.

Figure 3.8 Raw (a) and rectified (b) image pair

Furthermore, rectified images will be corrected so that the rows of the images digitized from horizontally displaced cameras, i.e. the stereo camera, are aligned, and similarly that the columns of images obtained from vertically displaced cameras are aligned. Without rectification, searching along the rows and columns will not produce the correct results.

3.2.2.1.2 *Edge Detection*

Edge detection is an optional feature that allows matching on the changes in the brightness rather than the absolute values of the pixels in the images. This feature is useful because the cameras in the PGR Triclops camera module [23] have auto gain control. If the auto gains in the cameras do not change identically, the absolute brightness between images may not be the same. While absolute brightness is not the same, the

change in the intensity stays near constant. Therefore, using edge detection helps in environments where the lighting conditions change significantly.

While edge detection may improve results, there is an additional processing cost that is associated with it. Therefore we need to evaluate the resulting improvement when choosing to turn edge detection on. In the meantime, validation is only available in the edge detection mode. So in our case, we applied the edge detection.

3.2.2.2 Stereo Processing

The stereo processing module applies the Sum of Absolute Differences algorithm described in some previous section. There are a number of parameters which are used to determine the kind of depth image produced:

3.2.2.2.1 Disparity Range

Disparity range is the range of pixels that the stereo algorithm searches in order to find the best match. In our system a disparity of zero pixel corresponds to an infinitely far away object. The maximum disparity defines the closest position of an object that is to be determined. Before we carry out 3D reconstruction, we usually choose and set the disparity range that is most suitable for the task at hand with regard to the fact that reducing the disparity range will allow the system to run faster and will decrease the chance of a mismatching.

3.2.2.2.2 Correlation Mask

Correlation mask is a square neighborhood around the pixel that the system is trying to find the match for. We can specify the size of the correlation mask. The correlation mask controls the coarseness of features compared between images. Larger masks will produce

depth maps that are denser and smoother, however, they may lack precision in identifying the position of depth discontinuities. On the other hand smaller masks will produce sparser and noisier depth images, but the localization of depth discontinuities will be much better as described in the following table:

	Disparity maps	Depth discontinuities	Computational cost
Large mask	Denser and smoother	Less precise	Larger
Small mask	Sparser and noisier	Better localization	Smaller

Table 3.1 Mask size comparison

To produce similar results the size of the mask must be proportional to the resolution of the images processed. Thus, in order to produce comparable results, a 5x5 mask on a 160x120-pixel image should be increased to a 9x9 mask for a 320x240-pixel image. Mask sizes should be odd numbers so that the center of the mask is the pixel with which we search for correspondence in the reference image. In other words, valid mask sizes are 3x3, 5x5, 7x7, and invalid mask sizes are 4x4, 6x6, 8x8. The system is capable of supporting a maximum 23x23 mask to a 1x1 minimum. We illustrate the effect of the masks by providing the following figures where we set the mask value to 5, 15 and 23 respectively.

Figure 3.9 3D point cloud with different mask size: (a) mask=5 (b) mask=15 (c) mask=23

It is obvious that the larger the mask is the better correspondence it can provide, although with an increased computational cost. The following table gives us the detailed number of points with regard to the above figures and their quality (5, 15, 23 are three typical mask sizes). Apparently, the more points we have, the clearer the disparity image will be and the more information we will obtain. But after comparing the quality of figures with mask size 15 and 23, we find that there is little improvement in the quality while at the same time a significant increase in the computational cost. Therefore, we are aware that we should compromise mask size to reduce computational cost. And in this specific example, 15 is a good value for the mask size.

Mask	5	15	23
Number of points	25591	70806	84299
Quality	Poor	Good	Best

Table 3.2 The influence of mask to 3D reconstruction quality

3.2.2.2.3 Validation Methods

In some cases, such as occlusions and lack of texture, it is not possible to establish correspondence between images. If the correspondence is not correct, the obtained measurements can not be correct. In order to avoid incorrect measurements, the following validation methods are introduced:

I. Texture validation

Texture validation determines whether disparity values are valid based on levels of texture in the correlation mask. If the amount of texture is not sufficient to produce correct matches, the pixel will be marked with the texture validation mapping value in the disparity image and declared invalid. Threshold can be set for texture validation. This threshold allows us to tune the texture-based rejection of pixels. Values range from 0 (no rejection) to 128 (complete rejection), and good operating range is from 0 to 2 (float value). The following example demonstrates the effect of texture validation.

Figure 3.10 texture validation with threshold = 2.0

Threshold is set to 2.0 in Figure 3.10. As compared with Figure 3.9 (b) with the same mask size, only 33456 points are retrieved, meaning 52.8% points are eliminated by this validation (originally 70806 points). Much important information is lost since the matching criterion becomes rigid, which might result in weakened robustness to recognize the fruits in a certain circumstance. Therefore, we can turn the texture validation off or decrease the value to get 3D point cloud with good quality for further processing. The other option is to set the threshold value lower with regard to specific environment so that we keep both robustness and accuracy.

II. Uniqueness Validation

Uniqueness validation determines whether the best match for a particular pixel is significantly better than other matches within the correlation mask. Even if the correlation mask has enough texture, the correct match may not exist due to an occlusion. If the correlation result is not strong enough, the pixel will be declared invalid. There is also a threshold for uniqueness validation. The range of threshold varies from 0.0 to 10.0. The larger the number, the less rejection occurs and common operating ranges are between 0.5 and 3.0. As shown in Figure 3.11, similar thing happens as in Figure 3.10. Lots of points are lost since we set the threshold to 0.5. Only 41.9% information remains after uniqueness validation. Higher threshold levels could be considered when it comes to uniqueness validation.

Figure 3.11 Uniqueness validation with threshold = 0.5

III. Surface Validation

Surface Validation removes the so called “spikes” in a disparity image. The spikes can often cover a connected region of many pixels. This noise is not zero-mean, random, evenly distributed or Gaussian. The result is that this noise is difficult to remove with standard filtering techniques, as it appears to be a valid signal, instead of noise. Surface validation is a method to validate regions of a disparity image based on an assumption that they must belong to a likely physical surface in the image. The method segments the disparity image into connected regions. Any region that is less than a given size is suspected and removed from the disparity image.

The mapping approach described is very sensitive to noise. Depth estimate distribution about the true depth of properly matched pixels is of sufficiently low deviation to be swallowed up by quantization. However, noise due to stereo mismatches is a serious problem. Scenes containing specular surfaces, repetitive patterns, and time-varying light sources can cause errors that are more or less uniformly distributed across the disparity

range of the stereo system. They generally appear as “spikes” in the disparity image and can drastically affect the quality of the map. Spikes are characteristic of mismatches in correlation-based stereo vision.

Some researchers have tried common image filtering techniques such as median filtering or morphological filtering. These filters can improve the results, but not significantly. These techniques intend to remove or reduce noise from inputs that have unstructured or evenly distributed noise, i.e., noise that appears randomly and consequently will appear most often as single pixels in the image. In disparity images errors often do not have these qualities. Errors in stereo matching occur when two similar image features are in proximity to each other. The algorithm may match one of the features in one image to the wrong feature in the other image. A typical example of this problem is the so-called “picket fence” problem. When looking at a picket fence with binocular stereo camera, there will be regularly spaced, nearly identical image patches that can be mismatched. This causes several “ghost” fences to appear. If these kinds of errors occur, they will generally happen over a patch of the incorrectly matched feature. Thus, noise does not appear in isolated pixels but in dense, connected regions. These coherent errors will confound the above filtering approaches as they will appear as a stable signal, one to be preserved instead of rejected.

As well, filters such as described above may remove valid features if those features are insufficiently thick. This can be a problem when there are thin objects in the scene such as poles or table edges. For low resolution stereo these objects often appear only one or two pixels wide and may be removed as not sufficiently stable.

We apply an approach using surface segmentation to overcome the problem of noise rejection for coherent errors as described in the previous section. To remove spikes caused by feature mismatches, we take into account the attributes of these errors: they are locally stable, but not large and they have no support from surrounding surfaces; they are genuine spikes with sharp disparity discontinuities at all borders. True surfaces in the stereo image should be not only locally consistent, but globally part of a larger 3D surface. By segmenting the image into continuous disparity surfaces, we are able to

establish a good hypothesis based on the size of the surface whether it is a real 3D surface or a noise artifact. To segment the image into surfaces of continuous disparity we apply the following logic:

```
i = L
j = L
For all j ∈ N(i)
    If | dj - di | ≤ 1
    Then group i and j
Count number of pixels with Label L
If number > threshold
Then it is valid
```

where i is any given pixel, L is a surface label, $N(i)$ is a neighborhood of pixels around i and d_i is the disparity value at location i . Entire surfaces are invalidated from the disparity image if the number of pixels that have a given label do not pass a threshold.

This approach has two significant benefits: it can reject cohesive spikes that may fool noise rejection filters; and it can preserve thin structures that are part of a coherent structure.

Figure 3.12(a) shows us the extreme importance of surface validation. Although the number of points doubles Figure 3.9(b), it doesn't provide us with enough useful information for further processing. On the contrary, the image contains much noise and invalid surfaces and as a result it makes recognition more difficult with a higher computational cost and lower accuracy.

Two key parameters can be adjusted when applying surface validation. One is disparity difference which set the maximum disparity difference between two adjacent pixels that will still allow the two pixels to be considered part of the same surface. The other one is surface size which sets the minimum number of pixels a surface must cover and thus still

be considered as a valid surface. The larger this number is, the fewer surfaces will be accepted. Common values range from 100 to 500, depending on the image resolution. Figure 3.12(b) used surface validation with $\text{size} = 300$ and $\text{difference} = 0.5$, it keeps 92.6% points as compared to the image in Figure 3.9(b), where we set the size to 200, improving the calculation speed to some degree. Compared with Figure 3.12(a), it removes a lot of noise, while keeping the large surfaces in the image. We should be clear that the size of surface and difference must be optimized to appropriate values according to specific environment and target that we intend to recognize.

Figure 3.12 3D point cloud (a) without surface validation (b) with surface validation
(Size = 300, Difference = 0.5)

3.2.2.2.4 Subpixel Interpolation

The PGR Triclops allows matching between images to subpixel accuracy. The library takes advantage of the matching results of the neighboring pixels of the resulting disparity to determine an approximation that is within a fraction of a pixel. Accurate calibration between cameras allows an accuracy of 0.2 of a pixel, which improves the accuracy for further 3D position calculation. The strict subpixel validation option is used when the subpixel interpolation is applied. Strict subpixel validation enables a more restrictive validation method for subpixel validation. With strict subpixel validation on, there will be less data in the output image, but it will be more reliable. To reduce computational cost, it is not necessary to include this function in our work

3.2.2.3 3D Position Calculation and Accuracy

Triangulation is a classical method to establish the relationship between disparity map and 3D XYZ position.

$$Z = fB/d$$

Where

Z = distance along the camera Z axis

f = focal length (in pixels)

B = baseline (in meters)

d = disparity (in pixels)

After Z is determined, X and Y can be calculated using the usual projective camera equations:

$$X = uZ/f$$

$$Y = vZ/f$$

Where

u and v are the pixel location in the 2D image

X, Y, Z is the real 3d position

Note: u and v are not the same as row and column. We must account for the image center.

We can find u and v by:

$$u = a - b$$

$$u = \text{centerCol} - \text{col}$$

$$v = \text{centerRow} - \text{row}$$

Note: If u, v, f , and d are all in pixels and X, Y, Z are all in the meters, the units will always work i.e. pixel/pixel = no-unit-ratio = m/m.

There are two kinds of stereo errors: mismatch and estimation. Mismatch is where the disparity is incorrect. Estimation errors are where the correlation match is correct but there are some errors in estimating the position of the sub-pixel disparity value. The

purpose of validation is to remove mismatch errors only. As described in the Validation Methods section, we mainly use Texture Validation and Surface Validation as the most effective combination. Validation does not affect the accuracy of the valid stereo disparity pixels. It only throws out ones that it believes to be mismatch errors.

Using the PGR Triclops enhanced rectification and enhanced stereo allows a significant improvement of correlation accuracy. Typically the errors of the 3D points are reduced in half. Larger stereo masks provide better accuracy but more smoothing of the 3D surface mask size 15 is a good compromise mask size.

The true 3D error model is, again, more complicated than the one given below. However, this is a simple approximation that is reasonably accurate. There are two errors that affect our XYZ accuracy, the calibration error (p for “pointing”) and the correlation error (m for “matching”). Almost all 640x480 cameras have a value of p between 0.06 and 0.08, and all 1024x768 cameras have a value of between 0.1 and 0.15. This value is for the maximum resolution of the camera. It should be divided by the reduction in resolution if the stereo is not being done at full resolution. (Example a 640x480 camera at 640x480 resolution would use p as given by manufacturer. For stereo at 320x240 we would use $p/2$). Correlation accuracy, m , can be estimated from Table 3.3 and 3.4 [23].

Resolution	Stereo Quality	Rectification Quality	Correlation Accuracy (pixels)
160 * 120	Standard	Standard	0.1
160 * 120	Standard	Enhanced	0.08
160 * 120	Enhanced	Standard	0.07
160 * 120	Enhanced	Enhanced	0.06
320 * 240	Standard	Standard	0.11
320 * 240	Standard	Enhanced	0.1
320 * 240	Enhanced	Standard	0.06
320 * 240	Enhanced	Enhanced	0.05
640 * 480	Standard	Standard	0.1

640 * 480	Standard	Enhanced	0.09
640 * 480	Enhanced	Standard	0.09
640 * 480	Enhanced	Enhanced	0.08

Table 3.3 Correlation accuracy results for various resolutions and quality settings with stereo mask=11 and edge mask=9

Stereo Mask	SS-RS	SE-RS	SS-RE	SE-RE
5	0.18	0.17	0.1	0.1
7	0.18	0.14	0.1	0.08
9	0.14	0.12	0.09	0.07
11	0.11	0.1	0.06	0.05
13	0.1	0.09	0.05	0.05
15	0.1	0.09	0.05	0.05

Table 3.4 Correlation accuracy results for 320x240 resolutions, over a range of mask sizes. “SS” means “Stereo Standard”, “SE” means “Stereo Enhanced”, similarly for RE and RS and rectification. All numbers are in pixels.

For instance, for a given result of triangulation, we obtain (x, y, z) . The tolerance in (x, y) are determined by the calibration error p [23]. These are quite simple:

$$\Delta x = pz/f$$

$$\Delta y = pz/f$$

The accuracy in z is a little more complicated:

$$\Delta z = fB/(d - m) - fB/d$$

The accuracies for (x, y, z) are $(\pm \Delta x, \pm \Delta y, \pm \Delta z)$. As an example, say we have the following 3D point from stereo done at 320×240 with enhanced stereo and enhanced rectification. Typical results for a 4mm camera would be:

$$p = 0.04$$

$$m = 0.05$$

$$f = 250$$

$$B = 0.01$$

$$d = 20$$

$$(x, y, z) = (1, 0.5, 1.25)$$

Then

$$\Delta x = (0.04) * (1.25) / 250 = 0.0002$$

$$\Delta y = (0.04) * (1.25) / 250 = 0.0002$$

$$\Delta z = (250) * (0.1) / (20 - 0.05) - (250) * (0.1) / 20 = 0.003$$

or

$$\Delta x = \pm 0.2\text{mm}$$

$$\Delta y = \pm 0.2\text{mm}$$

$$\Delta z = \pm 3\text{mm}$$

CHAPTER 4. COLOR SEGMENTATION

In Chapter 3, we have reconstructed the 3D image, which also solves stereo occlusion. Normally, object detection and recognition is implemented in a simple environment and has a high accuracy in correct detection. Whereas in our research, target objects (mature tomato fruits) are located in complex structured tomato plants, which consist of stalk, stems, branches, leaves, fruits and background. The complicated structure of the tomato plant makes it tough to detect mature fruits with regular shape or texture methods. Color segmentation provides us with a practical method to segment mature fruits from the complex structure surrounding it. In this Chapter, we present our color segmentation method, Color Layer Growing (CLG), by first introducing a segmentation concept, then the 3 phases of CLG, i.e. preprocessing, initialization and linking. In the end, we have some examples to show the result of this color segmentation.

4.1 Segmentation Definition

Segmenting an image is the process of labeling the pixels so that pixels whose measurements are of the same object are given the same label. The aim of segmentation is to divide an image into possibly large non-overlapping segments. Five types of region classifications are considered and described below with a graph illustration (Figure 4.1): correct detection, over-segmentation, under-segmentation, missed, and noise [25][26].

1) Correct detection

A pair of regions R_n in the GT (ground truth) image and R_m in the MS (machine segmentation) image are classified as an instance of correct detection if

- a) At least T percent of the pixels in region R_m , in the MS image are marked as pixels in region R_n , in the GT image, and
- b) At least T percent of the pixels in region R_n , in the GT image are marked as pixels in region R_m in the MS image).

2) Over-segmentation

A region R , in the GT image and a set of regions in the MS image $R_{m_1}, \dots, R_{m_x}$, where $2 \leq x \leq M$, are classified as an instance of over-segmentation if

- a) At least T percent of the pixels in each region R_{m_i} in the MS image are marked as pixels in region R , in the GT image, and
- b) At least T percent of the pixels in region R , in the GT image are marked as pixels in the union of regions $R_{m_1}, \dots, R_{m_x}$, in the MS image.

3) Under-segmentation

A set of regions in the GT image $R_{n_1}, \dots, R_{n_x}$, where $2 \leq x \leq M$ and a region R_m , in the MS image are classified as an instance of under-segmentation if

- a) At least T percent of the pixels in region R_m , in the MS image are marked as pixels in the union of regions $R_{n_1}, \dots, R_{n_x}$, in the GT image, and
- b) At least T percent of the pixels in each region R_{n_i} , in the GT image are marked as pixels in region R_m in the MS image).

4) Missed classification.

A region R , in the GT image that does not participate in any instance of correct detection, over-segmentation or under-segmentation is classified as missed.

5) Noise classification.

A region R_m in the MS image that does not participate in any instance of correct detection, over-segmentation or under-segmentation is classified as noise.

Figure 4.1 Segmentation Classification

4.2 Color Segmentation Method

Image segmentation is a key step towards object recognition in image analysis. Common methods include edge-detection, region-growing and clustering techniques. While clustering mainly uses statistical approaches, syntactical methods grow more popular for edge-detection and region-growing. Region growing methods can be classified as local and global techniques. Local techniques are simple and fast, but have the problem of chaining mismatches: two unconnected points might be connected by a chain of similar points and thus are mismatched into one region. Global techniques use $m \times m$ neighborhoods with a sufficiently large m to solve this, which on the other hand, causes a substantial increase in computation time. As a consequence, global techniques play no role in real-time or only close-to-real-time applications for color machine vision.

Our goal of color segmentation is to detect the target objects utilizing color information, and we developed a method combining the advantages of local (simplicity and quickness)

and global (robustness and accuracy) techniques. By bottom-up local regions growing and repeating distance measurements at higher levels we stay simple but eliminate mismatches by applying the global aggregated information of the connected regions up to this level.

4.2.1 Hierarchical structure

A basic concept in this approach is island. An island of level 0 simply denotes a single pixel and an island of level 1 is a set of pixels, usually within some simple and fast growing geometrical structure. Likewise, an island of level $n+1$ is a set of islands of level n . Figure 4.1 presents three examples with different shapes. Note that the squared and triangular islands of (a) and (b) overlay an orthogonal grid of pixels, Cartesian coordinate, which is a standard camera topology, in the meantime, (c) needs a different grid of pixels, where the Euclidian distance between any two neighboring pixels is constant, the so-called hexagonal topology. Islands of level n may overlap to form an island of level $n+1$ (c), or not (a) & (b).

Figure 4.2 Islands of three different levels in hierarchical structures based on (a) orthogonal, (b) triangular and (c) hexagonal topology

Operating on a hexagonal topology leads to some difficulties in practice because nearly all vision devices scan the pixels in an orthogonal scheme. Therefore, we choose the orthogonal shape, which has 8 neighbors.

4.2.2 Segmentation in island structures with color similarity and island distance

Color similarity and island distance in segmentation within island structures is the process to determine which regions of the same level should be grouped into a higher level. Note that any region r , of level $n+1$ must be a set of sub-regions within a single island, I , of level $n+1$. Therefore, only regions within certain distance of I are candidates to be connected. Local growing methods decide whether two neighboring regions r_1 , r_2 are connected by relying on the presence of two lowest level pixels, p_1 and p_2 , who belong to r_1 and r_2 respectively. But this may lead to the unexpected chaining mismatches. Instead of measuring pixels of the lowest level, our hierarchical structure measure the color similarity between two neighboring regions r_1 and r_2 , and the regions distance as well to decide whether they should be connected.

The color similarity measure is of particular importance for the quality of the segmentation results. The color similarity measure in the color segmentation is a color predictor D . Given two color content of candidate islands c_1 and c_2 in a three dimensional color space, D is defined by:

$$D(c_1, c_2) = \begin{cases} 1 : \text{similar} \\ 0 : \text{otherwise} \end{cases}$$

The color features are usually in RGB or HSV color space. In the HSV color model a color is described by the three attributes hue, saturation and value. In the RGB model, a color is depicted by red, blue and green. The drawback of the HSV space is an irremovable singularity at the V axis, where $R=G=B$ (saturation = 0). At low intensities and at low saturations the hue value is very unstable. Thus the HSV-space is not suited for Euclidean distance measures. It is impossible to use a constant threshold over the entire HSV color space to decide the similarity of colors [27]. Therefore, our method applies similarity measurement in RGB color space. Let $c(p)$ denote the color content of a pixel p and $c(r)$ the mean of all color contents within that region. The decision if r_1 and r_2

are connected depends on the color distance of two regions, $D(c(r1), c(r2)) < t$ for some threshold. As regions grow, one may choose different thresholds t for different levels. While color similarity is one approach to decide whether two regions should be connected, region center distance is the other way to assist the decision. In our specific case to detect fruits, a simple criterion for being candidates is that two regions should be close enough and thresholds for selecting proper candidates vary according to the level they belong to.

There are a lot of ways to define a color content $c(r)$ of a region r to measure the distance of such color contents and to evaluate this measurement. Most color similarity predicates are measures calculating the ratio of the mean colors distances and their variances.

4.2.3 Color Layer Growing Algorithm

Color Layer Growing (CLG) is developed based on color structure code (CSC) [28] which can segment an image by linking its homogeneous regions and splitting the regions by the similarity in color. The CSC was firstly introduced by Hartmann [29], following a hierarchical region growing method on a special hierarchical hexagonal topology. CLG segmentation algorithm operates essentially in the following three phases: preprocessing, initialization and linking. In the preprocessing phase, color band selection is accomplished by a color band filter and noise suppression is implemented by a nonlinear filter which additionally strengthens the sharpness of contours. In an initialization phase the image is partitioned into small color regions within an island of level 0. These small color regions are grown in the linking phase in a hierarchical manner to complete regions.

4.2.3.1 Preprocessing

As mature fruit in the greenhouse has its specific color band, filtering the color within this specific color band can eliminate the complicated plant structure, including stalks,

stems, leaves, etc. The remaining image after filtering might consist of target tomato fruit clusters, background components and noise.

Symmetric Nearest Neighbor

Pietikainen and Harwood [30] suggested SNN for noise cancellation. Since we have 8 neighbors for a center point, regard the four center symmetric pairs (c_1, c_8) , (c_2, c_7) , (c_3, c_6) and (c_4, c_5) in the neighborhood of center point c_0 . From each pair choose that pixel whose color value is closer to the central pixels color value. The central pixel is then replaced by the equally weighted average of the three determined neighbors. Let $v_i = (r_i, g_i, b_i)$, $1 \leq i \leq 4$, be the three color vectors with,

$$v_1 = f(c_1), \text{ if } \|f(c_0) - f(c_1)\| \leq \|f(c_0) - f(c_8)\|$$

$$= f(c_8), \text{ otherwise}$$

$$v_2 = f(c_2), \text{ if } \|f(c_0) - f(c_2)\| \leq \|f(c_0) - f(c_7)\|$$

$$= f(c_7), \text{ otherwise}$$

$$v_3 = f(c_3), \text{ if } \|f(c_0) - f(c_3)\| \leq \|f(c_0) - f(c_6)\|$$

$$= f(c_6), \text{ otherwise}$$

$$v_4 = f(c_4), \text{ if } \|f(c_0) - f(c_4)\| \leq \|f(c_0) - f(c_5)\|$$

$$= f(c_5), \text{ otherwise}$$

$$f(c_0)^{\text{new}} = \frac{1}{4} \left(\sum_{i=1}^4 r_i, \sum_{i=1}^4 g_i, \sum_{i=1}^4 b_i \right)$$

Some other applicable preprocessing methods include Median-Filter and K-Nearest-Neighbor-Filter (KNN). SNN is the fastest filter by evaluating criterion such as color distance calculations, additions and multiplications/divisions according to Table 4.1.

Filter	Color Distance	Num. of Additions	Num. of Mult. / Div.
Median	Large	Medium	N/A
KNN	Small	Large	Small
SNN	Small	Small	Small

Table 4.1 Comparison of three filters

4.2.3.2 Initialization

In the initialization phase, regions grow with every 9 pixels. Such a region growing consists of neighboring pixels of level 0 and whose mutual color distance lies below a certain threshold, thus forming a level 1 island. This operation is a purely local operation, independently processed for each island. Instead of starting with one seed pixel, the CLG starts in all islands of the preprocessed image (points eliminated from preprocessing are not included in this step, thus the computational cost is reduced). The result of the initialization phase is a set of new regions of level 1, with each one describing a small color patch. In the following linking phase these small color patches are checked for color similarity and continuity to grow hierarchically to complete color segments.

4.2.3.3 Linking

At the beginning of linking phase, elements of level 1 are given the same label if they are physically connected. Then close islands of level n ($n > 1$) are linked to new islands of level $n+1$. Color similarity and island distance within island structures become the process to determine which islands of the same level should be grouped into a higher level. Only regions within the sub-regions of a higher level region are candidates to be connected. The color similarity between two candidate regions r_1 and r_2 and the region distance are both measured to decide whether they should be connected.

The color similarity measure employs a color predicate D . The color features are in RGB color space. Let $c(r)$ be the mean of all color contents within that region. According to the color distance of the two regions, r_1 and r_2 are connected if the color distance $D(c(r_1), c(r_2)) < t$ for some threshold. While color similarity is one criterion to decide whether two regions should be connected, region center distance is the other factor, wherein thresholds for selecting proper candidates vary according to the different level they belong to.

Those color regions which are too small are not considered for the further layer growing, meaning only large regions (over certain threshold) can participate in region growing to form higher level layer. Islands that do not find any partner for linking on level n and are under certain size are discarded from the image. The linking operations are repeated for all islands on every level, starting from level 1 until the size of the region is large enough to be considered as the target object. Thus, color segmentation results in connected regions with similar color.

4.2.4 CLG examples

We describe the result of CSC color segmentation by giving an example. The original input is as Figure 4.2, which includes an original color image and stereo image within an office environment:

Figure 4.3 Original color image and stereo image

The result after color segmentation is illustrated by the following 2D and 3D images, which clearly segment tomatoes from the rest of the environment for further processing and position localization. Some of the tomato pixels are missing because of the color filtering in the preprocessing phase, for the preprocessing cannot keep all the tomato pixels after color filtering.

Figure 4.4 2D and 3D images after color segmentation

4.2.5 Comparison between CLG and CSC

CLG outperforms CSC in computational cost in our specific application.

- Using a color filter in CLG reduces the number of pixels in an image to those within a certain color band. This tremendously decreased computations since we have a much smaller number of pixels to handle.
- CSC grows color regions in a small increment, i.e., 9 candidates at a time to form a higher level code for further growing. Take a 600*480 image for example, CSC takes at least $\log(600*480)/\log(9)=6$ rounds to finish color segmentation. Instead of doing that, CLG can set certain thresholds for layer growing. And for our specific application, usually after initialization, two rounds of linking are enough to segment the target objects.

CHAPTER 5. DEPTH SEGMENTATION

The scope of previous work has focused on finding the mature fruits in the 2D pixel image, where the work stopped. In this work, as we aim to locate the mature fruit clusters with their (X,Y,Z) positions, after locating the object in the image, we should map it to its corresponding position in the 3D space. However, direct mapping might be less accurate, while depth segmentation can offer more accurate position. In this chapter, we introduce a depth histogram for retrieving the position of the target fruits based on the properties of these sphere like fruits.

5.1 Depth segmentation based on disparity map (depth) image

After identifying a few segmented target fruits by color segmentation in the 2D pixel image, we need to map them into 3D positions so that we know which ones are nearer or farther and which ones are grouped together into mature clusters so that we are able to decide the ones to be harvested and the harvesting order. Besides, the result of color segmentation may possibly include some false decision. With depth segmentation, we can eliminate this scenario to increase the correct detection rate while determining the accurate position of the fruit in the 3D space.

5.1.1 Disparity map image

A disparity map or depth image is an efficient method for storing the depth of each pixel in a stereo image. Each pixel in the map corresponds to the same pixel in an image, and the grey level corresponds to the depth at that point, indicating the distance of that point to the camera.

An example to demonstrate disparity is illustrated in Figure 5.1. Note that objects nearer have greater separation (large disparity), and objects far away line up closer (small disparity). There is large separation between corresponding objects in foreground, but points converge as distance increases. In fact, this is how stereo vision algorithms work. The computer attempts to match every pixel in an image with every pixel in the other using a correspondence algorithm. This can't be always perfect due to occlusions, poor texture, imperfect lighting system etc., but works amazingly well quite often, as does the human visual system.

Figure 5.1 Stereo image

Disparity Map Construction can be summarized as follows:

- Find every corresponding point between the images.
- Assign a value 0 to 255 at each point based on the “disparity” or separation.
- White (255) = close, Black (0) = far

Matching as many points as possible, the Bumblebee2 can produce dense disparity maps at 48 fps (Frame Per Second). Figure 5.2 is a single image from a stereo pair of oranges on the table in the office. Figure 5.3 shows a sparse disparity map produced from Figure 5.2 using the PGR Disparity Mapping which creates a disparity map that is “balanced” visually for viewing and not to be used for calculations because of the distortion of the depth values. With this 256 bit gray level disparity map image, lighter objects are close, and darker objects are farther away. The value of 255 represents pixels which cannot be matched, for identification or exclusion in further calculations.

Figure 5.2 Tomatoes in the office

Figure 5.3 Disparity image

5.1.2 Point grey disparity map

The actual image produced for extracting depth is shown in Figure 5.3 and its histogram is shown in Figure 5.4. Note that most of the information is stored in the levels from around 80 to 140, as may be inferred from the histogram in Figure 5.4. In general, there may be more valid pixel values (grey levels) under different circumstances such as nearby objects, and the disparity value can be set in the PGR software. This value should be within depth that you are trying to detect. For instance, if a disparity value of 80 corresponds to a depth of 2 meters, and the area within 2 meters is the region of interest, the minimum value should be set to 80. Larger range will only decrease the efficiency of the PGR system because it has to search the images over a larger distance when attempting to match corresponding points. Level 240 to 255 is reserved for invalid bits, and 0 is used for points at infinity. The white regions in Figure 5. are classified to invalid

pixels that could not be mapped for various reasons. These values are ignored in our analysis. Zero values also need not be processed.

Figure 5.4 Histogram of raw disparity map.

5.2 Extracting Object Position from 3D Image

We observe that the histogram plot may help to segment the depth image and tell what and where the objects are according to their depth distribution. After separating the mature tomato fruit clusters by color segmentation, we map 2D image to 3D space (depth information). We can take advantage of this depth information to locate the exact position of each cluster by its shape property which we will explain in this section. To start, we introduce the level concept.

5.2.1 Level concept

The information contained in the disparity maps is less useful until it is processed to obtain object position and depth information. Without loss of generality, consider a disparity map as 480 pixels high, 640 pixels wide and 256 (0-255) levels in gray-level depth. Referring to Figure 5.5, visualize a depth map as 256 separate images, each depth image will have information about the image at that depth. For example, if some object is 1 meter from the camera, it might have several pixels at a disparity level 15. If it were at 3 meter it might have several pixels at a disparity value of 8, in other words, found on plane 8. Recall that larger disparity values correspond to closer objects. Depending on depth resolution and object size and orientation, objects will generally appear at two or more image depth planes.

Figure 5.5 Image plane levels

Rather than processing disparity maps, a 3-D point cloud could be created from the pixels by using the so called triangulation method referring to formula $Z = f \cdot B / d$, where Z is Depth, f is focal length, B is the baseline of the stereo camera, and d is the disparity value from the disparity map (depth map). However, the point cloud data would have to be processed resulting in several extra steps. To calculate actual X and Y position using the u and v coordinate from the disparity map, the formulas are $X = uZ/f$, and $Y = vZ/f$. The formula $Z = f \cdot B / d$ is used in this algorithm to determine the depth of each plane.

5.2.1 Shape based recognition with 3D point cloud distribution

Shape based recognition is made more efficient and accurate by using a depth histogram representation. We attach great importance to representation of objects because the performance of a classification and recognition system substantially relies on representation. The first step is to derive depth histograms of objects of interest. A statistical feature is the targeting feature that can be used to characterize the shape of target objects, based on the assumption that we are dealing with a number of objects which exist on a uniform background with the same or similar shape. Color segmentation can be used to isolate the pixels representing fruit in the 3D point cloud. After color segmentation, usually we will only see the fruits with some noise in the 3D point cloud. According to the shape property of the sphere fruit, we are able to see one fruit on multiple levels. The statistical distribution of the depth histogram of targeted fruit can be used to characterize the shape of target objects. Since a spherical fruit has the most points in its center level and fewer points on its front facing the camera and on both sides, we have the rough normal distribution. Then, according to the shape property of the fruit spheres, points scattered on different levels roughly appear in normal distributions as shown in Figure 5.6, which gives an example of point cloud distribution in both level and distance, created from a point cloud (Figure 5.7), consisting of two separate target fruits at different distances. And the two circled red local peaks are exactly where the fruits are located. This reduces the object position detection problem to a local maxima detection problem.

Figure 5.6 3D point cloud distribution in level and distance

Figure 5.7 3D image after color segmentation

Given the noise in the distance histogram curve, it is divided into groups and a local maxima is found for each group to yield the distance of each cluster from the camera.

CHAPTER 6. EXPERIMENTS AND RESULTS

It was necessary to test the performance of the aforementioned method in the field to discover its advantages and disadvantages. To do so, the stereo system was configured to detect mature fruit clusters and locate the center position of them. In this chapter, the operating environment is introduced and the results of experiments on detection and localization are presented.

6.1 *Operating Environment*

A commercial tomato greenhouse has many long rows of plants between which carts can roll on rails to support workers doing maintenance and harvesting tasks as in Figure 6.1. Usually the cart is designed with the capacity to raise and lower the worker and move forward and backward along the tracks. During our experiments the cart height was adjusted to about 55cm and the mature fruit clusters were between 60 and 100 cm in height. Since the distance between two rows is around 155cm, the camera was mounted on the cart at three different distances (close, medium and far) for every single scene to test the robustness of the method.

Figure 6.1 Greenhouse Operating Environment

6.2 Implementation and Results

The experiments were run during when the tomato plants were quite mature in April 2007 in a tomato greenhouse in South Western Ontario. The gathered data consists of raw images, stereo input images and calibration files. The image resolution used was 600*480.

As mentioned in the previous section, usually one cluster includes 4-5 fruits with occlusion to varying extents. We mounted our camera at close, medium and far positions to test the robustness of this method. We present the result in three phases:

- I. Image after color filtering
- II. Image after CGL segmentation
- III. Image showing the position of the mature fruit clusters

We selected 32~35cm as a typical value for close distance. And the detection and localization process can be depicted in the following figures.

(d)

(e)

(f)

Figure 6.2 Mature fruit cluster detection and localization at 32~35cm (close distance), (a) raw-right image, (b) image after depth filter, (c) image after color filtering, (d) image after color segmentation, (e) image with mature cluster position, (f) Z distance histogram

Figure 6.2(a) is the raw right image with some stereo occlusion in the bottom right part. We eliminate the stereo occlusion by 3D reconstruction and decrease the computational cost with a depth filter afterwards, as displayed in Figure 6.2(b). Figure 6.2(c) presents the image after color filtering and noise cancellation, but there is still some background noise left irremovable since the color filter itself is set to include as many target pixels as possible while not compromising too much in background and noise cancellation. Therefore, the CLG is employed to remove the noise and Figure 6.2(d) demonstrates the effect of color segmentation, after which, the position of the mature fruit cluster is specified in the 2D image as in Figure 6.2(e). Combining the target in the 2D image with 3D point-cloud, we have 3D mature fruit clusters image with XYZ information. As depicted in Figure 6.2(f), we have one mature fruit cluster in the raw image. By identifying the local maxima, we are able to calculate the position of the target clusters.

Finally, the mature fruit cluster was located at 34.55cm with $(X, Y, Z) = (-6.16, -0.37, 34)$ cm. This example indicates that with stereo and self occlusion, we are still able to detect the mature fruit clusters by this method while having unripe fruits on the plant.

Then we moved our camera to around 65~68cm (far distance from camera to target cluster) with more fruit clusters to test the robustness at further distance within a more complex situation.

(a)

(f)

Figure 6.3 Mature fruit cluster detection and localization at 65~68cm (far distance), (a) raw-right image, (b) image after depth filter, (c) image after color filtering, (d) image after color segmentation, (e) image with mature cluster position, (f) X distance histogram

Three mature fruit clusters were captured in this image. Without compromising the accuracy, we were capable of localizing the positions of three mature fruit clusters in this example. In Figure 6.3(f), we applied X distance histogram instead of Z distance in Figure 6.2(f). This was because sometimes when all the target clusters were in the same level along Z distance, we should have some other optional axis to build this histogram to localize the actual position of the targets. In our experiments, the X axis appeared to be an excellent option for such situations. The calculated distances of the three clusters were 81.78cm, 67.98cm and 75.96cm (starting from left). And the 3D positions were (-43.67, 1.35, 69.13) cm, (-10.4, 15.91, 70.27) cm, (24.04, 15.91, 70.27) cm respectively. This example proves that our method is applicable when there are more than one mature fruit cluster in the image, in other words, the method is robust in that sense.

For the medium distance experiments, the camera was placed about 50~53cm from the fruit. Without compromising the complexity, we tried to detect fruit clusters and separate fruits as found in the following Figure 6.4.

(b)

(c)

Figure 6.4 Mature fruit cluster detection and localization at 52~55cm (medium distance), (a) raw-right image, (b) raw-left image, (c) image after depth filter, (d) image after color filtering, (e) image after color segmentation, (f) image with mature cluster position, (g) Z distance histogram

This experiment was carried out where there was a severe stereo occlusion in the image. As in Figure 6.4 (a) and (b), the mature fruit cluster in the left part of (b) can barely be seen in (a), while the mature fruit cluster in the bottom right part of (a) totally disappears in (b). Although in (e), two clusters were separated after color segmentation, taking the stereo occlusion into consideration, we only have one target cluster in (f). In this experiment, we tried to locate every single mature cluster and as a result, we successfully located three of the mature fruits on a cluster. Due to the color filtering, we lost one single fruit on this cluster. But this is quite acceptable as the one we lost was a less mature fruit. And above all, we accurately located the mature fruit cluster once again, which proves the robustness of this method.

Min Measured Distance (cm)	Max Measured Distance (cm)	Calculated Distance (cm)
32	35	34.55
36	39	38.46
37	40	39.78
38	41	41.08
51	54	53.42
52	55	54.13
53	56	55.43
53	56	55.24
63	66	65.51
65	68	67.98
69	72	71
71	74	73.24

Table 6.1 Comparison between Measured distance range and calculated distance

Figure 6.5 Comparisons between Measured and Calculated Distance of Fruit Clusters

Table 6.1 and Figure 6.5 evaluate the accuracy and robustness of our method. Data was collected at different distances in various scenarios. The max and min distances in Figure 6 are the distance range of target fruit clusters, and the calculated distance (blue line) falls in between the two measured lines, which means that our method can recognize the fruit clusters and correctly calculate their positions in the real world.

CHAPTER 7. CONCLUSIONS AND FUTURE EXPECTATIONS

7.1 Conclusions

This thesis presents a method containing 3D image reconstruction, color segmentation and depth histogram of detecting and localizing mature fruit clusters in a complex structured tomato plant for automatic harvesting by using a stereo vision camera (PGR BumbleBee2). The most challenging issues in this work was the severe stereo and self occlusion and various noises caused by sunlight, background and the plant's own complicated structure. The main contributions of this work can be summarized as follows. (1) The recognition and localization process was carried out in a 3D reconstructed environment. Previous methods included only 2D information unless a range finder was applied. With 3D information, we could result in more accurate localization. (2) Previous selection of the vision sensor varied from a single camera to large laser scanner. In our work, the selection of a color stereo camera not only fulfilled the need of detecting and localizing purpose with a small size vision sensor but also met the economic requirement, in other words, it was practical in terms of cost. (3) We introduced a depth histogram to improve the accuracy of localization. This depth histogram was built on a layer concept by applying the statistic shape property of tomato fruit. (4) We successfully integrated 3D reconstruction, CLG color segmentation and finally depth histogram to calculate the existing mature fruit clusters in every scene. The successful detection rate was higher than previous work. (5) The experiments were implemented in a real tomato greenhouse with strong sunlight and severe noise. Thus, from the industrial perspective, our method is reliable. To sum up, this thesis presents a novel method to recognize mature fruit and

locate cluster positions in a complex 3D reconstructed environment with stereo and self occlusion and severe noise for greenhouse automatic harvest applications. To recognize fruit, 3D reconstruction eliminates stereo occluded objects and we use CLG to segment fruits from the background in a 3D reconstructed environment. A depth filter is applied to remove irrelevant image information to speed up the processing. Depth segmentation is employed to calculate the exact 3D XYZ position of a fruit cluster and thus the 3D position.

7.2 Future Work

This work can be improved and extended and particularly we can see the following improvements and extensions.

- Although we are able to locate the mature fruit clusters, in this work, we didn't put much effort on segmenting single fruit. As shown in Figure 6.4, we tried separating some mature fruits, but the successful rate is still unknown as we haven't done many experiments in this part. In particular, when there is severe self occlusion, we cannot assure the successful detection rate. Much work can be done with regard to this problem.
- The final goal of this work is to carry out automatic harvesting. Therefore, a robot grip is needed. The integration of the robot grip and the vision system will be a big issue. Probably some small touching sensors need to be mounted on the grip when they touch the fruits, while the vision system functions as a guiding machinery.
- The application of this work can also be extended to measuring the ripeness of the fruits. Instead of hiring people to watch them, we can employ the system to monitor all the fruits so that we know when the exact time is for certain operations. And it can also help to categorize the fruits by their quality, i.e., ripeness and size.

BIBLIOGRAPHY

- [1] Jimenez A. R., Jain A. K., Ceres R, Pons J. L, "Automatic fruit recognition : a survey and new results using Range/Attenuation images" Elsevier Science, vol. 32, no10, pp. 1719-1736 (1999)
- [2] C.E. Schert and G. K. Brown, Basic considerations in mechanizing citrus harvest, Transactions of the ASAE, 343-346 (1968).
- [3] A. Grand D'Esnon, Robot harvesting of apples, Proceedings of AgrMation I, ASAE and SME, St. Joseph, Michigan, 210-214 (1984).
- [4] A. Grand D'Esnon, G. Rabatel and R.Pellenc, Magali: A self-propelled robot to pick apples, ASAE paper No. 87-1037, St. Joseph, MI 49085-9659 (1987).
- [5] P. Levi, R. Falla and R. Pappalardo, Image controlled robotics applied to citrus fruit harvesting, Procedures, ROVISEC-VII, Zurich (1988).
- [6] E. Blandini, First approaches to robot utilisation for automatic citrus harvesting, Land and Water Use, Dodd&Grace (eds.), Balkema, Rotterdam (1989).
- [7] Hayashi and Ueda, Orange harvesting robot, Kubota Co., Sakai, Japan (Mimeo.) (1991).
- [8] A.K. Jain and C. Dorai, Practicing Vision: Integration, Evaluation and Applications, to appear in Pattern Recognition Journal (1997).
- [9] L. Kassay, Hungarian robotic apple harvester, ASAE Paper No. 92-7042, Charlotte, North Carolina, June 21-24 (1992).
- [10] Y. Edan and G.E. Miles, Design of an agricultural robot for harvesting melons, Transactions of the ASAE, 36(2), 593-603 (1993).
- [11] R. Ceres , J.L. Pons, A.R. Jiménez, J.M. Martín and L. Calderón, Agribot: A robot for aided fruit harvesting, AGENG96, 2 (1996).
- [12] E. Parrish and A.K. Goksel, Pictorial pattern recognition applied to fruit harvesting, Transactions of the ASAE, 20, 822-827 (1977).
- [13] Sites and Dewilche, Computer vision to locate fruit on a tree, Transactions of the ASAE, Vol.31(1), 257-263 (1988).

- [14] D. Slaughter and R. C. Harrel, Color Vision in Robotic Fruit Harvesting, Transactions of the ASAE, Vol.30(4), 1144-1148 (1987).
- [15] Whitaker, Miles, Mitchell and Gaultney, Fruit location in a partially occluded image, Transactions of the ASAE, Vol. 30(3), 591-597 (1987).
- [16] M. Benady and G. E. Miles, Locating melons for robotic harvesting using structured light, ASAE Paper No.:92-7021 (1992).
- [17] W. Qiu and S.A. Shearer, Maturity assessment of broccoli using the discrete Fourier transform, ASAE Paper No. 91-7005, St. Joseph, MI (1991).
- [18] M. Cardenas-Weber, A. Hetzroni and G.E. Miles, Machine vision to locate melons and guide robotic harvesting, ASAE Paper No. 91-7006 (1991).
- [19] C.E. Schert and G. K. Brown, Basic considerations in mechanizing citrus harvest, Transactions of the ASAE, 343-346 (1968).
- [20] J.J. Gaffney, Reflectance properties of citrus fruit, Transactions of the ASAE 16(2), 310-314 (1969)
- [21] D. Slaughter and R. C. Harrel, Discriminating fruit for robotic harvest using color in natural outdoor scenes, Transactions of the ASAE, Vol.32(2), 757-763 (1989).
- [22] F. Buemi, M. Massa and G. Sandini, AGROBOT: a robotic system for greenhouse operations, Fourth Workshop on Robotics in Agriculture & the Food Industry, IARP, Toulouse, 172-184 (1995).
- [23] Point Grey Research BumbleBee2 Specifications, <http://www.ptgrey.com/products/bumblebee2/index.asp>
- [24] Introductory Techniques for 3D Computer Vision, Trucco and Verri, Prentice Hall 1998, Page 123-172
- [25] A. Hoover, G. Jean-Baptiste, and et al. An experimental comparison of range image segmentation algorithms. IEEE PAMI, 18(7): 673-689, July 1996
- [26] J. Min, M.W. Powell, K.W. Bowyer, Automated performance evaluation of range image segmentation, IEEE Transactions on systems, MAN and Cybernetics, Vol. 34, Feb. 2004

- [27] V. Rehrmann, L. Priese. Fast and robust segmentation of natural color scenes. Third Asian Conference on Computer Vision, Hongkong, 8-10th January 1998.
- [28] P. Sturm, 3D-color-structure-code-segmentation by using a new non-plainness island hierarchy. IEEE, Vol.2 page(s): 953- 956, Oct. 2004
- [29] G. Hartmann. Recognition of hierarchically encoded images by technical and biological systems. Biological Cybernetics, 57:73-84, 1987.
- [30] M Pietikainen and D. Harwood. Segmentation of color images using edge-preserving filters. In V. Cappellini and R. Marconi, editors, Advances in Image Processing and Pattern Recognition, pages 94–99. North-Holland, 1986.

VITA AUCTORIS

Linghe (Steven) Yang, was born on February 26th, 1983 in the metropolitan city of Shanghai, China. He received his Bachelor of Science Degree in Electronics and Information Technology from East China Normal University, Shanghai, China in 2005. He got enrolled at the University of Windsor in September 2005, and graduated with a Master of Applied Science Degree in Electrical and Computer Engineering in September 2007.