A RIPE TOMATO RECOGNITION AND
LOCALIZATION SYSTEM

A Thesis
Presented to
the Faculty of School of Engineering
of
the University of Guelph

by
JIN PEI

In partial fulfillment of requirements
for the degree of
Master of Applied Science
May, 2011

© Jin Pei, 2011
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.

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

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

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 protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

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

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

A RIPED TOMATO RECOGNITION AND LOCALIZATION SYSTEM

Jin Pei
University of Guelph, 2011

Advisors:
Prof. Simon X. Yang
Prof. Gauri S. Mittal

A real-time system for the recognition and localization of ripe tomatoes in greenhouses is developed, which includes binocular camera calibration, stereo matching, ripe-tomato recognition, and localization. It can automatically calibrate the binocular camera system, recognize and localize ripe tomatoes for the robot to pick off.

A low-cost calibration method for the binocular camera system is developed using asymmetric data flow and an asymmetric energy function. The method is capable of providing the position and orientation of the right camera with reference to the left camera with high consistency and accuracy, which requires no high-cost 3D measuring equipment. A new point sampling method is developed for the calibration algorithm, where the intersection of two straight lines is used as the sample point. An exponential evaluation function is built for line extraction, which considers more on the valuable points in the line matching process, and makes the process consistent and fast. A new simple algorithm for binocular image rectification is developed, which transforms the original binocular image pair into an ideal one for stereo matching. The stereo matching segmentation algorithm classifies pixels into groups of inner pixels and edge pixels, which match both the characteristics and the functions of the pixels, making the stereo matching process faster and more reliable. A specialized simple equation for ripe tomato localization is derived based on the calibrated binocular camera system. Experimental results demonstrate the effectiveness of the developed system.
# Contents

Table of Contents i

List of Figures iv

List of Tables vi

List of Symbols vii

List of Abbreviations xi

1 Introduction 1

1.1 Problem Statement 3

1.1.1 Camera Calibration 3

1.1.2 Stereo Matching 6

1.1.3 Localization 8

1.2 Objectives of This Thesis 8

1.3 Contributions of This Thesis 9

1.4 Organization of This Thesis 10

2 Background and Literature Review 12

2.1 Binocular Camera System Calibration 12

2.1.1 Analytical Binocular Camera System Calibration 13

2.1.2 Artificial Neural Network Based Binocular Camera Calibration 16

2.2 Stereo Matching 17

2.2.1 Feature Based Stereo Matching 17
2.2.2 Correlation Based Stereo Matching .................................. 19
2.2.3 Energy Minimization Based Stereo Matching ...................... 19
2.2.4 Other Stereo Matching Approaches .................................. 20
2.3 Ripe Tomato Recognition and Localization .............................. 22
2.4 Summary ........................................................................... 22

3 Calibration of the Binocular Camera System ......................... 25
3.1 Introduction ........................................................................ 25
3.2 Definitions and Expected Output ......................................... 27
3.3 Spatial Relationship between Two Cameras ........................... 29
3.4 Modification of the Binocular Camera System Model ............... 30
3.5 Asymmetric Data Flow and Energy Function .......................... 33
   3.5.1 Asymmetric Data Flow and Symmetric Data Flow .............. 34
   3.5.2 The Structure of the Data Flow and the Energy Function ... 36
   3.5.3 Details of the Functions in Data Flow ............................ 38
3.6 Calibration Algorithm for the Binocular Camera System .......... 40
3.7 Data Preparation for Calibration ......................................... 42
   3.7.1 Edge Detection .......................................................... 42
   3.7.2 Line Extraction ......................................................... 42
   3.7.3 Corner Point Calculation ............................................ 46
3.8 Comparison of the Algorithm with Existing Algorithms .......... 48
3.9 Experimental Results ....................................................... 50
3.10 Summary ......................................................................... 50

4 Segment Feature Based Stereo Matching ............................... 53
4.1 Introduction ........................................................................ 53
4.2 Image Rectification .......................................................... 54
   4.2.1 Epipole Calculation .................................................... 54
   4.2.2 Image Rectification .................................................... 55
4.3 Segmentation ..................................................................... 59
4.4 Segment Combination ....................................................... 61
5 Recognition and Localization Algorithms of Ripe Tomatoes

5.1 Introduction ........................................... 71
5.2 Ripe Tomato Recognition .............................. 72
   5.2.1 Transformation from RGB Space to sRGB Space 72
   5.2.2 Transformation from sRGB Space to XYZ Space 73
   5.2.3 Transformation from XYZ Space to CIELAB Space 73
5.3 Ripe Tomato Localization ............................... 74
5.4 Harvesting Robot Calibration .......................... 76
5.5 Experimental Results ................................ 77
5.6 Summary ............................................. 79

6 Conclusion and Future Work ............................ 80

6.1 Conclusion ............................................ 80
6.2 Future Work ........................................ 82

References .............................................. 84
# List of Figures

1.1 An example of colour differences between two images. ................................. 6
1.2 The vertical position difference between two images. ................................. 7

2.1 A binocula camera based RBF network (from Zong et al. 2006). ................. 16

3.1 Illustration of a camera coordinate system. .................................................. 28
3.2 Illustration of focal length measurement. ...................................................... 33
3.3 Illustration of the asymmetric data flow and symmetric data flow. ............... 35
3.4 The relationship between two coordinate systems, the camera-plane co­ordinate system and the image one. ....................................................... 37
3.5 The data flow of binocular camera calibration ............................................... 38
3.6 Extracted edge points. .................................................................................... 44
3.7 The geometric model of a line to be extracted. .............................................. 45
3.8 Lines extracted from edge points. ................................................................. 46
3.9 Satellite points for corner recognition ........................................................... 47
3.10 Lines and corners extracted from both images. ............................................ 48
3.11 Input data structure of the data flow. ......................................................... 49

4.1 An illustration of left image rectification ....................................................... 56
4.2 An illustration of right image rectification .................................................... 57
4.3 A sample stereo image pair before rectification ............................................ 63
4.4 The rectified sample stereo image pair ........................................................ 64
4.5 A segmentation result of the sample stereo image pair .............................. 65
4.6 Non-complete segments before segment combination .............................. 66
4.7 Complete segments after segment combination 67
4.8 Feature based stereo matching result. 67
4.9 The colour calibration curves. 68
4.10 The colour rectification result. 69
5.1 A screen shot of ripe-tomato localization 76
List of Tables

3.1 Part of the input data for calibration ........................................ 51
3.2 BCS calibration results ............................................................ 52
5.1 The average CIELAB values of ripe tomatoes ......................... 77
5.2 A typical ripe-tomato localization result ................................. 78
## List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>The $3 \times 3$ camera rotation matrix between $H$ and $U$</td>
</tr>
<tr>
<td>$A_i$</td>
<td>$A$ between $H$ and $U_i$</td>
</tr>
<tr>
<td>$A_{ij}$</td>
<td>$A$ between $H_j$ and $U_i$</td>
</tr>
<tr>
<td>$a^*$</td>
<td>$a^*$ Component of a colour in CIELAB space</td>
</tr>
<tr>
<td>$B$</td>
<td>The 3D translation vector between $H$ and $U$</td>
</tr>
<tr>
<td>$B_i$</td>
<td>$B$ between $H$ and $U_i$</td>
</tr>
<tr>
<td>$B_{ij}$</td>
<td>$B$ between $H_j$ and $U_i$</td>
</tr>
<tr>
<td>$B$</td>
<td>The blue element of a colour in $RGB$ colour space</td>
</tr>
<tr>
<td>$B_l, B_r$</td>
<td>The blue elements of the corresponding left-image colour and right-image colour in a colour-mapping function</td>
</tr>
<tr>
<td>$B_s$</td>
<td>The blue element of a colour in $sRGB$ colour space</td>
</tr>
<tr>
<td>$b^*$</td>
<td>The $b^*$ Component of a colour in CIELAB space</td>
</tr>
<tr>
<td>$C$</td>
<td>A $4 \times 3$ matrix representing a camera’s external and internal parameters</td>
</tr>
<tr>
<td>$D$</td>
<td>The distance between an object and the origin of a camera</td>
</tr>
<tr>
<td>$d$</td>
<td>The disparity value of an image point</td>
</tr>
<tr>
<td>$d_i$</td>
<td>The distance from a point to a line</td>
</tr>
<tr>
<td>$E$</td>
<td>The energy of optimization</td>
</tr>
<tr>
<td>$f$</td>
<td>The focal length of a camera</td>
</tr>
<tr>
<td>$f_i$</td>
<td>The focal length of the $i$th camera</td>
</tr>
<tr>
<td>$G$</td>
<td>A vector representing the extended coordinates of a point in a 3D coordinate system, $G = \begin{pmatrix} X &amp; Y &amp; Z &amp; 1 \end{pmatrix}^T$</td>
</tr>
</tbody>
</table>
The green element of a colour in RGB colour space

The green elements of the corresponding left-image colour and right-image colour in a colour-mapping function

The green element of a colour in sRGB colour space

The transformative function of a colour from XYZ space to CIELAB space

A vector representing the coordinates of a 3D coordinate system,

\[ \mathbf{H} = \begin{pmatrix} X & Y & Z \end{pmatrix}^T \]

\( \mathbf{H}_i \)  
H in the ith camera coordinate system

The intensity of an image

The Intensity of an image in sRGB colour space

The length of a line

The \( L^* \) element of a colour in CIELAB space

The length of the image of a line

Image width and height

A point on an object

A point representing the projection of an object point (P) on the image plane of the ith camera

A point representing the rectified position of \( P_i \)

The centre point of the left image

Epipoles of the left image (1) and the right image (2)

Focal points of the left camera (1) and the right camera (2)

The intersection of the vertical centre line of the left image and the Epipolar line through \( P_1 \) in the left image

The 3 × 3 The camera rotation matrix of the right camera against the left camera

The 3 × 3 rotation matrix of the robot coordinate system against the left-camera one

The red element of a colour in RGB colour space
\[ R_l, R_r \] The red elements of the corresponding left-image colour and right-image colour in a colour-mapping function

\[ R_s \] The red element of a colour in \( sRGB \) colour space

\[ r_{ij} \] An element of \( R \) at the \( i \)th raw and the \( j \)th column

\[ s \] An offset of a line

\[ T \] A 3D translation vector of the right camera against the left camera, \( T = \begin{pmatrix} T_1 & T_2 & T_3 \end{pmatrix}^T \)

\[ T' \] A 3D translation vector of the robot coordinate system against the left camera, \( T' = \begin{pmatrix} T'_1 & T'_2 & T'_3 \end{pmatrix}^T \)

\[ T_1, T_2, T_3 \] Elements of \( T \)

\[ T'_1, T'_2, T'_3 \] Elements of \( T' \)

\[ t \] A free value of a line

\[ U \] A vector representing extended coordinates of a 2D camera-plane coordinate system, \( U = \begin{pmatrix} u & v & 1 \end{pmatrix}^T \)

\[ U_i \] \( U \) for the \( i \)th camera, \( U_i = \begin{pmatrix} u_i & v_i & 1 \end{pmatrix}^T \)

\[ u, v \] Coordinates of \( U \)

\[ u_{ei}, v_{ei} \] Coordinates of \( P_{ei} \) \((i = 1, 2)\)

\[ u_i, v_i \] Coordinates of \( U_i \)

\[ \mathbf{W}_{BCS} \] Binocular camera system parameter vector

\[ w \] The input of Function \( g(w) \)

\[ X \] A vector representing extended image coordinates, \( X = \begin{pmatrix} x & y & 1 \end{pmatrix}^T \)

\[ X_i \] \( X \) for the \( i \)th camera, \( X_i = \begin{pmatrix} x_i & y_i & 1 \end{pmatrix}^T \)

\[ X_{DF} \] The input-layer vector of data flow in binocular camera calibration

\[ X, Y, Z \] Coordinates of a point on an object in a coordinate system

\[ X_i, Y_i, Z_i \] \( X, Y, Z \) in the \( i \)th camera coordinate system

\[ x_{i1}, y_{i1} \] Coordinates of \( P_{i1} \)
$X_{D65}, Y_{D65}, Z_{D65}$ Components of a colour with $D65$ white point in $XYZ$ colour space

$X_w, Y_w, Z_w$ Component normalizers for a colour with $D65$ white point in $XYZ$ colour space

$x, y$ Coordinates of $X$

$x_i, y_i$ Coordinates of $P_i$

$x'_i, y'_i$ Coordinates of $P'_i$

$\hat{x}_2, \hat{y}_2$ The computed result of $x_2, y_2$ following the data flow of binocular camera calibration

$x_{c1}, y_{c1}$ Coordinates of $P_{c1}$

$x_{ei}, y_{ei}$ Coordinates of $P_{ei}$

$Y_{Actual}$ The actual output of a binocular camera system corresponding to the data flow output

$Y_{DF}$ The output layer vector of data flow in binocular camera system calibration

$\alpha, \beta, \gamma$ Rotation angles of the right camera against the left camera

$\alpha', \beta', \gamma'$ Rotation angles of the robot coordinate system against the left camera

$\theta$ Theta angle of a line

$\lambda$ A scalar value representing the depth of a 3D point from a camera

$\lambda_i$ $\lambda$ between $H$ and $U_i$

$\lambda_{ij}$ $\lambda$ between $H_j$ and $U_i$

\footnote{$i = 1$ for the left camera and $i = 2$ for the right camera.}
# List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN</td>
<td>Artificial Neural Network</td>
</tr>
<tr>
<td>DF</td>
<td>Data Flow</td>
</tr>
<tr>
<td>BCS</td>
<td>Binocular Camera System</td>
</tr>
<tr>
<td>LC</td>
<td>Left Camera</td>
</tr>
<tr>
<td>LI</td>
<td>Left Image</td>
</tr>
<tr>
<td>LSM</td>
<td>Least Square Method</td>
</tr>
<tr>
<td>MI</td>
<td>Mutual Information</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
</tr>
<tr>
<td>RC</td>
<td>Right Camera</td>
</tr>
<tr>
<td>RI</td>
<td>Right Image</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

This thesis research work is a part of the project “Robotic Systems for Tomato Harvesting”, a group research project of the Advanced Robotics and Intelligent Systems Lab in the School of Engineering at the University of Guelph. This project is funded by the Ontario Centres of Excellence.

Agriculture is a major part of Canada’s economy. Greenhouse industry is very important to Canada due to the cold weather, and it is widespread in recent years. The total annual revenue of the Canadian greenhouse industry was $2.4 billion in 2009. As for greenhouse tomato planting, there were 53 million square-foot area of greenhouses producing 507 million pounds of tomatoes, and about $434 million annual outcome was created from tomato production in 2009 ("Greenhouse, Sod and Nursery Industries", Statistics Canada, http://www.statcan.gc.ca/bsole/olc-cel/olc-cel?catno=22-202-XWE&lang=eng).

Tomato harvesting is a labour-intensive and tedious type of operation, and the high humidity conditions in greenhouses are harmful to human operators. Robotic harvesting systems can save the cost of burdensome labour, protecting workers from this risk. This system is an autonomous robot that can automatically pick off tomatoes from tomato branches. It is composed of three subsystems:

**Mobile Navigating system:** An intelligent and automatic mobile system to navigate the robot to the designated working location
**Sensing system:** A pair of cameras to capture the images of tomatoes, and a computer equipped with a corresponding software system to recognize and localize the tomatoes and to provide data to control the robot to complete the picking off task.

**Actuating system:** A combination of manipulator and end-effector to pick off tomatoes from tomato branches.

Some parts of the robotic tomato harvesting system have been developed in the Advanced Robotics and Intelligent Systems Lab. A package of manipulator-control software has been developed based on a position-related artificial neural network (ANN) (Pei et al. 2010). An end-effector integrating both gripping and cutting functions has also been designed (Jia et al. 2009). A ripe-tomato recognition and localization system was proposed that can successfully recognize ripe tomatoes (Yin et al. 2009), where the tomato localization is performed using a single camera, a laser distance detector, and a computer system.

This project is focused on the sensing system, which is also called the ripe-tomato recognition and localization system. A new configuration for tomato localization is developed, in which a binocular camera system (BCS) is used to recognize and localize tomatoes, eliminating the laser distance detector and the aiming mechanism in previous works. The ripe-tomato recognition component is inherited from Yin et al. (2009).

The ripe-tomato recognition and localization system includes both hardware and software. The hardware of the system is typically a BCS and an image processing unit (a computer and, in some cases, a set of accelerating hardware such as GPU). The software is a functional code package that uses the images captured from the BCS as an input, processes the images, recognizes ripe tomatoes, determines the 3D locations of the ripe tomatoes, and develops the parameters required by the robot to pick off the tomatoes.

This project uses two identical CCD cameras to construct a BCS, and the video streams of both the cameras are fed into the computer using USB cables. A GPU is
an option to increase the processing speed. However, the system can complete all the designed tasks without any GPU, if the processing speed is not a concern. Because the system must be commercially competitive, off-the-shelf components are used to reduce the cost. The camera system is built with commercially available webcams, the computer is a personal computer with Windows XP, and Microsoft VS 2010 C# is used as the programming language. A more rugged compact industrial PC will be used aboard the robot.

1.1 Problem Statement

The most challenging part of the ripe-tomato recognition and localization system is algorithm development, and it is also the main part of this thesis. This thesis explores BCS calibration, stereo matching, and ripe-tomato recognition and localization.

1.1.1 Camera Calibration

Camera calibration is the very first step of this project, and it is an important part. It is a pre-requisite for BCS-based ripe-tomato recognition and localization. In an ideal situation, the BCS would be designed and manufactured with zero errors; i.e., it would have the following properties:

1. The image obtained on the camera plane would be a geometrically precise copy of the scene and the objects without any distortion or deformation.

2. All the CCD cells in the cameras would be identical. The same intensity of an object applied to different CCD cells would obtain the same output.

3. The two cameras would be identical, which means the cameras would have the exactly same image if they are installed in the same position against the objects.

4. The two cameras would be installed exactly in the designated position and direction, and one camera’s position would be an exact offset of the other one’s position.
If the above conditions are satisfied, the cameras would not need to be calibrated and the data from the cameras could be directly used for stereo matching and object localization. Unfortunately, the ideal condition is difficult to be fulfilled, so that camera calibration is required.

Corresponding to the four conditions for an ideal BCS, camera calibration has the following two classification methods.

The first classification method is to follow the calibration steps. There are two steps, the first step is to calibrate the parameters related to one single camera, which is called internal parameter calibration, and the second step is to determine the relationship between both cameras, which is called external parameter calibration.

The calibration algorithm of this project mainly focuses on external parameter calibration, because it is difficult to assemble cameras in the exact positions, and external parameters are difficult to be determined without software-based calibration, while internal parameters are related to single products, and it is possible to determine the internal parameters from camera specifications.

The second classification method is functional, in which the calibration is divided into spatial (or geometric) parameter calibration and colour calibration. This project deals with both of them; however, because the image colour changes with every image, it is more related to images than to cameras. The BCS calibration algorithm in this project only focuses on spatial parameter calibration. Colour calibration is separated from this algorithm, and it is developed in the corresponding part of this thesis. Unless it is mentioned, BCS spatial parameter calibration is referred by BCS calibration, while colour calibration is referred by image colour calibration.

BCS calibration algorithms developed to date can be classified into two categories, which are analytical methods and ANN-based methods. Both of the two categories have advantages and drawbacks. Analytical methods display the physical features of the system, which is suitable for applications such as stereo matching, but its models are too complex, and are always nonlinear. The computing load is very heavy, preventing it from being feasible as a real-time system. Another drawback is that they are always inconsistent and unreliable.
ANN-based methods were developed in recent years to overcome the drawbacks of analytical methods. ANN-based methods eliminate the complex and nonlinear models, but they can result only in facsimile models that are not related to the real physical parameters of the system. For a BCS calibrated using an ANN algorithm, the location of a 3D point is determined only if the projection of the 3D point onto both camera planes is determined; however, this prerequisite is not always true in stereo-matching cases, because the correspondence between any two points is undetermined before stereo matching. It is necessary to develop an algorithm that is simple, consistent, and able to offer true physical parameters of the system. This is the first problem this project faces. To solve this problem, an innovative calibration algorithm with asymmetric data flow and an asymmetric energy function is developed.

It could be found from the literature that most BCS calibration algorithms rely on 3D coordinate measurement (Tsai 1987, Liu et al. 2008, Zong et al. 2006, Ge et al. 2008). Expensive measurement equipment is employed to obtain the 3D coordinates. High expenses make this series of BCS calibration methods suitable only for research purposes in lab situations, but not suitable for commercial use. Developing a low-cost BCS calibrating system is important for robotic harvesting systems to be competitively viable in the market. This is the second problem the project tackles. The algorithms of this project would reduce the cost of BCS calibration by reducing or eliminating the use of 3D coordinate measuring equipment.

Beside the spatial calibration described above, image colour calibration or re-correction is another item that needs to be addressed. Each camera has different colour parameters, and most cameras can automatically adjust their exposure; i.e., the image colour of an identical object is different with different cameras, and the image colour is different even if the same camera is used at a different time. As shown in Figure 1.1, the left image is much darker than the right image. Consequently, colour re-correction is a process for every single image.

This project uses the result of segment-based stereo matching (Chapter 4) to re-correct colour.
In summary, the problem that needs to be solved for BCS calibration is the development of a fast and low-cost BCS calibration method and a simple and consistent algorithm that determines the real physical parameters of the system.

1.1.2 Stereo Matching

Stereo matching is the basis of the object recognition and localization program. The objective of stereo matching is to find the relationship between pixel pairs from the left image (LI) and the right image (RI).

A problem that has to be solved before stereo matching is image rectification. Stereo matching requires the BCS to perform as well as an ideal BCS; i.e., the images of the same 3D point would be located on the same horizontal line on both camera planes; however, there are big vertical differences between the two points in real captured images. In Figure 1.2, the height of the same plate corner is apparently different between the LI and the RI. The function of the image rectification algorithm is to move the corresponding points to the same horizontal line.

Rectification algorithms used to move the pixels of both images to form the images of an ideal BCS have not yet been found to be mature. Building an algorithm to ap-
proximate the images of an ideal BCS is a problem that must be solved in this project. This project builds a simple but efficient and effective algorithm to approximate the images of an ideal BCS.

Stereo matching has been established and developed over the last half century. El-Etriby et al. (2007) has classified stereo-matching algorithms that were developed in several trends such as feature-based, correlation-based, and energy-minimization-based stereo matching. Correlation-based stereo matching is also called local methods, and energy-minimization-based stereo matching is also called global methods. Correlation-based matching and energy-minimization-based matching can also be classified as pixel-level stereo matching, because both of them use pixel information, such as intensity and colour vector, to construct the matching function.

As inboard real-time software serving an onsite robot, the software of this project has to use fast algorithms. Because pixel-based matching methods require iterative computing, and they always spend quite a few minutes to obtain a result, so that they are not suitable for real-time robot control. The stereo-matching algorithm of this project is feature-based stereo matching, which is fast and suitable for tomato localization. Feature-based stereo matching uses less data load, and gives the 3D
location coordinates of an object directly.

The first step of feature-based stereo matching is segmentation, through which the image pixels of an object form a cluster called a segment, so that the system can focus on a segment to extract features. Many segmentation algorithms have been developed so far; however, most segmentation methods are not reliable. Different algorithms give different results for the same image. Because of this, most algorithms are modified, combined, and some extra processes are added to the original algorithms to make the segmentation results consistent and reliable. As a result, the segmentation algorithms become more and more complex, increasing the computing load and making the algorithms unsuitable for real-time robot control. Building a simple, but reliable segmentation algorithm is a challenging problem this project is facing.

To sum up, stereo image rectification, segmentation, and stereo matching are the three problems that are solved in this project.

1.1.3 Localization

The last problem needing to be addressed is tomato localization. Theoretically, it is simple to compute the 3D world coordinates of an object providing that a disparity map is obtained by stereo matching; however, the 3D world coordinates calculated from the disparity map are not the actual 3D world coordinates of the object, because the images used for stereo matching are rectified images instead of original ones. The general algorithms for localization are not suitable for rectified images, so that a new localization algorithm has to be built.

1.2 Objectives of This Thesis

The main objective of this project is to develop a ripe-tomato recognition and localization system in which a bunch of ripe tomatoes are recognized and localized utilizing both the left and right images obtained from a binocular camera system. The effectiveness of the system is ensured by a complete development of all parts of the system including binocular camera system calibration, stereo image rectification,
stereo matching, and ripe-tomato recognition and localization. Binocular camera system calibration and stereo image rectification are prerequisites for stereo matching, while stereo matching is the pre-condition for ripe-tomato recognition and localization. The application of the system is robotic tomato harvesting systems. The output of the project is a user friendly software package including all the above functions.

1.3 Contributions of This Thesis

The contributions of this thesis can be summarized as follows:

This project develops a complete software system for ripe-tomato recognition and localization. The software can automatically calibrate the camera system, recognize and localize ripe tomatoes in a natural environment, and offer 3D coordinate information for a robot to pick off the tomatoes. This system is a mandatory part of the robotic tomato harvesting system.

This project develops a low-cost BCS calibration method that uses only a pair of commercial level cameras and a 3D object point, eliminating the requirement for 3D coordinate measurement equipment. The developed method reduces the cost and increases the precision of BCS calibration, thus making it viable in the market.

A new kind of asymmetric data flow and a new asymmetric energy function are proposed in the BCS calibration algorithm. These features eliminate the transformation of data from 2D to 3D, making the algorithm consistent.

This project develops a robust image point identification algorithm for BCS calibration, which uses the intersection of two straight lines as the sample point. This algorithm avoids the difficulty of recognizing an image point directly.

This project proposes a new exponential evaluation function in the line matching algorithm. The new evaluation function stresses the effect of points closer to the matching line and eliminates the effect of points further from the line. It makes the line matching algorithm more efficient and precise.

A new simple binocular image rectification algorithm is developed for stereo matching, which can create a fair approximation to the images of an ideal BCS
for stereo matching, and can avoid the complexity of Epipolar geometry-based image rectification.

The segmentation algorithm of this project divides pixels into inner pixels and edge pixels, which makes the process faster and more reliable. A segment combination algorithm is developed to increase the segment-correlation quality.

A colour calibration algorithm is developed using the least square method (LSM). The result is used to calibrate the image colour, making the image of the same sample point be the same colour in both images.

A simple tomato localization equation is derived from Epipolar geometry. Experimental results show that it precisely represents the real location of tomatoes.

The program developed in this project can be directly integrated into a robot system to construct a prototypical robotic ripe-tomato harvesting system for further development.

1.4 Organization of This Thesis

This thesis has three parts, introduction, algorithm, and conclusion. The introduction part includes two chapters, which are Chapter 1, introduction (this chapter) and Chapter 2, literature review. The algorithm part is the main body of this thesis, which is comprised of three chapters addressing three topics of the project. The conclusion is the last chapter (Chapter 6). This part summarizes the achievements of this project and provides some potential future works in different directions inspired during the research.

Chapter 1 introduces the background and overview of the project. This chapter first reveals the background of the project. It starts from an economic perspective, then it introduces the project "Robotic Systems for Tomato Harvesting", to which the project of this thesis work belongs. The structure of the system of this thesis work is also described. The problem statement section lists the problems that the project is facing and the corresponding solutions. A consecutive section identifies the objectives of this thesis. The contributions of this thesis are then listed in another
section, and the organization of this thesis is described in this section.

Chapter 2 is a literature review. As described in Chapter 1, the main parts of ripe-tomato recognition and localization are BCS calibration and stereo matching. The literature review mainly includes these two parts and some works for tomato recognition and localization. The summary part of this chapter evaluates the key technologies in the literature, exposes the gaps that need to be filled, and introduces the significance of this project in future works.

Chapter 3 deals with the first topic of the main body, BCS calibration. The main contributions of this project, the asymmetric data flow and the asymmetric energy function, are shown in this chapter. This chapter first introduces the role that this topic plays in the whole system, then the spatial relationship between the two cameras is derived from Epipolar geometry. This relationship is a general model of a BCS. To make it practical, some modifications are introduced to the model, and a simplified BCS model is obtained. An algorithm featured with asymmetrical data flow and energy function is proposed, then input data preparation is discussed including some innovative techniques, image line extraction and corner point calculation. The proposed algorithm is compared with existing ones to show its advantages. Lastly, the effectiveness of the algorithm is verified with experimental results.

Chapter 4 describes the feature-based stereo matching, which consists of image rectification, segmentation, and stereo matching. Each step includes some contributions of this project to increase the processing speed, so as to make the algorithms viable for real-time applications.

Chapter 5 is focused on ripe-tomato recognition and localization. A ripe-tomato recognition algorithm is constructed based on colour information in CIELAB colour space. The tomato localization algorithm uses a reversal data flow of the image rectification algorithm and the BCS Epipolar geometry built in this project.

Chapter 6 includes conclusion and discussion. This part summarizes the contributions and experimental results of this project, and discusses some potential developments that could be based on the results of this project, but are yet to be finished because of the scale of this project.
Chapter 2

Background and Literature Review

A ripe-tomato recognition and localization system consists of BCS calibration, stereo matching, and ripe-tomato recognition and localization. This chapter is a literature review of the topics consecutively.

2.1 Binocular Camera System Calibration

Binocular camera system calibration is a pre-requisite for BCS-based stereo matching, ripe-tomato recognition and localization. Binocular camera systems are widely used in robotic vision applications, such as ripe-tomato recognition and localization and robotic ripe-tomato harvesting. In robotic vision systems, two or more identical cameras are mounted on a robot and are connected to an image processing unit to obtain the stereo data of the environment through a package of programs. Plenty of stereo-matching algorithms have been developed for this purpose (Ruling et al. 2008, Sun et al. 2010). However, most stereo-matching algorithms are based on ideal Binocular camera systems, where the two cameras are identical and are precisely mounted on a frame. This ideal condition is difficult to fulfill, so that software-based BCS calibration algorithms are required.

BCS calibration consists of two parts, spatial calibration and colour calibration. Because color calibration is more related to images than cameras, it is always called image color calibration. Regularly, BCS calibration means BCS spatial calibration.
The objective of BCS calibration is to identify the spatial relationship between the two cameras of a BCS from a series of binocular stereo image pairs captured by the dual-cameras, so that the images can be rectified as the images of an ideal BCS, and the stereo-matching algorithms can be employed to identify objects.

BCS calibration (spatial) is further divided into two fragments, which are internal parameter calibration and external parameter calibration. Internal parameter calibration is focused on identifying the parameters of each camera, such as radial and peripheral distortion (Zhang 2000, Baba et al. 2005, Baba et al. 2002), while external parameter calibration is addressed on the spatial relationship between the two cameras. Because cameras can be made with very high precision along with the development of camera manufacturing technology, internal camera parameter calibration is not frequently mentioned in recent years, and the main trend of BCS calibration research is focused on external parameters.

BCS calibration algorithms developed to date can be classified into two categories, which are analytical methods and ANN-based methods.

2.1.1 Analytical Binocular Camera System Calibration

Analytical binocular camera system calibration methods are traditional methods developed earlier (Tsai 1987). The principle of analytical BCS calibration methods is based on a parameterized analytical model of the BCS derived from Epipolar geometry (Gomez & Villanueva 2005, Akhloufi et al. 1999, Hauck et al. 1999). The model is a set of equations containing BCS parameters and a set of sample data including the coordinates of 3D sample points and the coordinates of their projections on both camera planes. When a set of parameters are chosen, the model acts as a data flow from one part of the sample data set to another. An energy function is defined to evaluate the precision of the model. An optimization method is utilized to change the parameters until the energy function reaches the minimum value.

In Tsai (1987), a camera’s position and direction in an object coordinate system, and its focal length, distortion, and other parameters are computed using some salient points of a template plate. Since this method implements only a pinhole model, it
cannot give precise results of the defocus effect. A camera model with a thin lens system is proposed in Baba et al. (2002). The distance from an object to a camera is computed from blur effect and zooming translation. It is successful for internal parameter calibration. The result of Baba et al. (2002) is extended in Baba et al. (2005) by computing both the blur effect and the geometry of sample points.

The above conventional BCS calibration methods mostly employ expensive 3D coordinate measuring equipment to measure the coordinates of the 3D points. Some flexible methods have been invented in which 3D coordinate measuring equipment is replaced with pattern templates to reduce the cost.

The research in Zhang (2000) is focused on internal parameter calibration. A template is placed in different positions and directions, then closed form solution of equations and nonlinear refinement are used to obtain precise parameters of cameras. This calibration method does not need to know the 3D coordinates of the template, and it can be performed without measuring equipment.

Consequently, a large number of modifications and techniques have been developed to increase the speed, precision, consistency, and reliability, and to reduce the cost of calibration in different directions.

One direction is to combine the flexible method with other algorithms. LSM and triangulation methods (Jiang et al. 2010) are combined with flexible methods to obtain a simple and fast calibration method. The limit of this work is that it requires the cameras be parallel to each other, a condition that does not always happen. Least squares support vector machines (LSSVM) method can be applied into BCS calibration (Liu et al. 2008) to eliminate the computing of BCS parameters. The image is divided into two regions in radial direction, making the calibration more accurate.

The flexible method can also be extended to the case of large camera base lines using two plates to be placed in parallel positions (Hu 2009). The plates can be placed in different views of a BCS.

Another direction is linearization (Xu et al. 2000), where two templates are used together. The formulations are simplified, then the calibration becomes faster and
more consistent in this work; however, the accuracy is decreased. In some cases, such as indoor environments, there are plenty of parallel lines that can be used for BCS calibration (Xu et al. 2006).

Two 3D model reconstruction algorithms, triangular measure based on camera model with distortion and imitated direct linear transforming were applied to the BCS calibration (Gao et al. 2006) to calibrate both internal and external parameters simultaneously. The accuracy of the results were improved by the introducing of the genetic algorithm and the particle swarm algorithm (Gao et al. 2008).

Flexible methods reduce the cost of calibration, but the data flow starts from small values, such as image data and template data, introducing more noise to the data flow, and making the algorithm inconsistent. One effective solution for improving the consistency is to seek the appropriate direction of the data flow and energy function. Many researches were made in this field. Mostly, the distance between the real 3D position of an object point and the point’s computed position from camera images is used as the energy function (Gao et al. 2006, Gao et al. 2008, Hung & Tang 2006, Fiala & Shu 2008), and the data flow points to the energy function. A different energy function is formed from rectification errors (Bradley & Heidrich 2010), which gives more accurate results; however, rectification error is related only to the vertical differences between left and right images, the horizontal accuracy is not improved.

This paper proposes the asymmetric data flow and the asymmetric energy function into the BCS calibration algorithm. The algorithm employs one 3D point as sample device, and only a distance is measured. This feature eliminates the expensive 3D coordinate measuring equipment. The data flow uses the distance and the left camera image as input, avoiding the data flow moving from small values to big values, thus increasing the consistency and converging speed.
2.1.2 Artificial Neural Network Based Binocular Camera Calibration

In recent years, ANN related methods have been developed in camera calibration area.

The radial basis function (RBF) ANN for BCS calibration was introduced in Zong et al. (2006). This algorithm uses the projections of an object point in both camera images as an input, and uses the 3D coordinates of the object point as an output (see Figure 2.1). The complex nonlinear model of the BCS is eliminated in this work. This algorithm processes faster than traditional algorithms, and it does not need human intervention; it is easy to operate. The ANN was then optimized with the genetic algorithm for camera calibration (Liu & Xie 2007). The experimental results show that the method is fast, practical, and reliable.

![Figure 2.1: A binocular camera based RBF network (from Zong et al. 2006). X, Y, Z: world coordinates; u_1, v_1: left camera plane coordinates; u_2, v_2: right camera plane coordinates.](image)

In Ge et al. (2008), the parameters of Gauss function are tuned to achieve faster and more consistent convergence, and the camera matrices are replaced by the ANN function and weights.

ANN-based camera calibration methods eliminate the complicated computation
of Epipolar geometry that is nonlinear, so that the algorithms are more efficient than traditional methods; however, they can result only in facsimile models that are not related to the real physical parameters of the system.

The camera calibration algorithm of this project is inspired by ANN technology, but it has more characteristics of optimization algorithms than ANNs.

2.2 Stereo Matching

When camera calibration is completed, the rectification parameters can be derived from the calibration results. The image data from cameras then can be rectified using the parameters. The rectified image data follows the constraints of an ideal BCS strictly, so that it can be used to find out the 3D coordinates of object points from the image coordinates of their projections. As described before, for an interested point in the RI, a correlative point from the LI can be found. These two points form a correlative pixel pair. Correlative pixel pairs represent real points on the object surface. Following Epipolar geometry, the 3D coordinates of an object point can be found from the disparity value, the coordinate offset between the two correlated points in a correlated pixel pair. Stereo matching is used to find the correlative pixel pairs and to construct a disparity map.

Stereo matching algorithms are classified as three categories, which are feature based, correlation based, and energy-minimization based stereo-matching algorithms. The following is a review of stereo-matching developments in the three categories.

2.2.1 Feature Based Stereo Matching

The earliest work concerning feature-based stereo matching is Marr & Poggio (1977), in which the basic structure of feature-based stereo matching is described. After that, some modifications were made, and some other technologies were added to the family of feature-based stereo matching to increase the performance.

Feature-based stereo matching is a type of object level algorithms, while the other two categories of stereo-matching methods are pixel level methods. Feature-based
stereo-matching algorithms match the features of segments from both images of a BCS to find the correspondence and to form corresponding segment pairs. Each corresponding segment pair represents a 3D object. The 3D coordinates of the object are calculated using the disparity value of the corresponding segments. Most feature-based stereo-matching processes have two steps, which are segmentation and stereo matching.

Segmentation is the first step of feature-based stereo matching. Image points projected from the same object surface always have similar features, so that they can be segmented to extract the image of the object. The most popular segmentation method is mean-shift colour segmentation (Comaniciu & Meer 2002, Wang & Lim 2010), in which the colour intensity of all image points is mapped into a 3D colour space where the colour points are distributed as clouds of colour points in the colour space. The points are clustered into different partitions, and a mean colour point for each cluster is found by shifting the guess of the mean toward a higher density position. The image is segmented according to the clusters around means.

Another segmentation method is edge detection. The original edge detection was introduced in Canny (1986), where edges are extracted according to the intensity differences between contiguous pixels. The rank of edges (Wang 2004) has been added to edge detection methods to make it adaptive to different contrasts. Morphological methods, such as edge combination, noise filtering, and object extraction, can be used to enhance the reliability of the algorithm (Balakrishnan et al. 2004).

The next step is stereo matching. In feature-based stereo matching, the stereo-matching step can be performed at the segment level to reduce computing load and increase the matching speed. Some frequently used algorithms are Plane fitting (Tao & Sawhney 2000), second order curve fitting (Woodford et al. 2009), disparity distribution model (Li & Chen 2010), ground control point model (Xiao, Xia, Lin & Zhang 2010), salient object detection (Park et al. 2002, Bjorkman & Kragic 2004), and trainable selective attention model (Choi et al. 2004).

In recent years, feature-based stereo-matching methods are combined with other methods to obtain better performance that are hard to achieve by any single al-
algorithm. The algorithm can be combined with dynamic programming along segment edges (Xiao, Xia & Lin 2010, Shao & Da 2010) and belief propagation (Sun et al. 2003, Klaus et al. 2006, Yang et al. 2009).

Some feature-based stereo-matching algorithms use windows with fixed sizes instead of segments, which result in identical structure (Resko & Baranyi 2005, Humenberger et al. 2010). This kind of feature-based stereo matching can be implemented as parallel computation, so as to construct a real-time system using parallel-computing hardware.

2.2.2 Correlation Based Stereo Matching

The original theorem of correlation-based stereo matching is simply to match the pixels based on intensity similarity; however, this intensity similarity method does not work well, because the corresponding pixels do not always have the same intensity. The intensity values of pixels are affected by noise, reflect factor, and camera parameters. Intensity compensation is considered to reach the correlation goal, and other properties of pixels and their corresponding support windows are used to work as assistance to the algorithm. As a result, some variations of correlation methods were developed just after the original method was introduced. The most famous variations are spatial information correlation (Ogale & Aloimonos 2005), colour-weighted correlation (Yoon & Kweon 2006), self-organizing neural network (Venkatesh et al. 2007, Vanetti et al. 2009), and intensity histogram correlation (Liu et al. 2009).

2.2.3 Energy Minimization Based Stereo Matching

Energy-minimization-based stereo matching is focused on finding the minimum situation of a global energy function using a kind of global iteration algorithm, so that they are called global methods. There are several categories of algorithms developed for global iteration.

The first category of Energy-minimization based stereo matching is belief propagation, which is an algorithm to pass the belief properties among the nodes of a
graphical network. It was first founded in Pearl (1982), and was introduced into stereo matching in Sun et al. (2003). Some developments in this category are the replacement of ordering constraints using visibility constraints (Sun et al. 2005), density gradients based parameter regularization (Zhang & Seitz 2005), and hierarchical belief propagation (Felzenszwalb & Huttenlocher 2004, Yang et al. 2006).

Loopy belief propagation method (Yang et al. 2008) is used to improve the precision for low textured regions. A message compressing technology (Yu et al. 2007) can compress belief information on different compressing ratio to save the memory for propagation. Using high-dimensional global energy function (Pock et al. 2008) can help finding the minimization solution more efficiently.

The second category of Energy-minimization based stereo matching is graph cuts. Graph-cuts method is a method to cut a network into several parts by cutting boundaries called cuts. There are labels assigned to each part, and an energy function is formed according to the labels and the properties of the parts themselves. There is also another kind of energy function related to the smoothness between adjacent parts. Graph cuts related algorithms are developed to find optimized cuts that have minimum energy function over the graph. Graph cuts theory was initially introduced into computer vision in Greig et al. (1989), where maximum a posteriori estimate of a binary image is obtained. An $\alpha - \beta$ swap algorithm and an $\alpha$—expansion algorithm were proposed to solve the graph cuts problem (Boykov et al. 2001), and the two algorithms are improved in Kolmogorov & Zabih (2001) and were introduced into correspondence computing and stereo matching.

### 2.2.4 Other Stereo Matching Approaches

There are some other technologies added to the family of stereo-matching algorithms. They are helpful to improve the performance of stereo matching.

One technology for stereo matching is Pixel dissimilarity, which is a feature vector representing how a pixel stands out from its surroundings, and it is used to replace colour-space distance measure; however, it needs more computing load. Pixel dissimilarity was introduced into stereo matching in Birchfield & Tomasi (1998), and
Mutual Information (MI) is another newly developed technology in stereo matching. It is a probability value measuring the dependency between random variables. In stereo-matching area, the corresponding pixel pairs from LI and RI would have high values of MI, while non-corresponding pixel pairs would have low MI values. Some stereo-matching algorithms, such as Kim et al. (2003), Hirschmuller (2005), and Banno & Ikeuchi (2009) were developed based on MI. MI-based stereo-matching algorithms eliminate the hypothesis of colour discontinuities corresponding to object boundaries, and they are robust to non-Lambert objects and intensity differences between different cameras.

Dynamic programming is a stereo-matching algorithm that focuses on finding an optimal solution along a scan line at one time. When one scan line is finished, it moves to another scan line. In dynamic programming, the energy values with all the disparity values of all the pixels along a scan line are computed to form a 2D matrix, then the optimal disparity path is found in the matrix using the cost aggregation method. The algorithm is fast because it does not need iteration. The drawback of the algorithm is that the smoothness between contiguous scanned lines is not considered. Some improvements have been proposed to solve the problem (Veksler 2005, Deng & Lin 2006, Wang et al. 2006, Mattoccia et al. 2007, Salmen et al. 2009).

Bhusnurmath & Taylor (2008) has introduced linear programming into stereo matching. The disparity of one segment is approximated as a linear function. So that the energy function can be calculated upon segments, instead of pixels. This method reduces computing load. Interior point methods are used to solve the linear equations.

Cooperative programming (Brockers 2009) is a new stereo-matching technology that constructs the energy function using the probability distribution of the disparity. The minimization of energy function results in an optimized disparity probability distribution. The peak point of the distribution is chosen as the final disparity.

Different algorithms have different performances. They improve stereo-matching
technology in different directions. While a technology improves the stereo-matching performance, it also has limitations and drawbacks. Recently, multiple technologies are joining together to exceed the limitations and construct stronger algorithms (Brockers et al. 2005, Wang & Zheng 2008, Kosov et al. 2009, Banno & Ikeuchi 2009). These researches combine similarity measure, mean-shift colour segmentation, plane fitting, variational method, mutual information, and belief propagation to achieve high precision; however, heavier computing load is required in these methods because multiple algorithms are run in the processes.

2.3 Ripe Tomato Recognition and Localization

The first robotic tomato harvesting system was proposed in Gotou et al. (2003), where infrared laser beam sensors are used to scan the area and recognize ripe tomatoes. This is the earliest work that can be found from the literature concerning ripe-tomato recognition and localization.

A ripe-tomato recognition system was developed using the K-means method to segment tomatoes in L*a*b* colour space (Yin et al. 2009). In this work, Morphology methods are used to process the shape information of the tomatoes and to extract one tomato from a tomato cluster. This system obtains an image of a tomato using a single camera and provides spherical coordinates to the systems computer. It uses a laser beam to detect the distance between the tomato and the camera.

2.4 Summary

A tomato recognition and localization system includes BCS calibration, stereo matching, and tomato recognition and localization. Plenty of works in this field are related to BCS calibration and stereo matching, and these two fields are the main contents of this project.

In the area of BCS camera calibration, both analytical calibration methods and ANN methods were developed. Both of them can obtain a model of the BCS.
Most analytical BCS calibration methods were developed based on Epipolar geometry. The analytical model of a BCS consists a set of equations that include unknown BCS parameters and input data from image sampling. The unknown BCS parameters are obtained by solving the Equations. Numerous of optimization methods were developed to solve the equations in different ways. Data flow and the energy function are two important elements of defining the type of an optimization algorithm, and the construction of the two elements is critical to the consistency, sensitivity, and precision of the optimization. A variety of algorithms were developed employing different data flow and energy functions to increase the consistency and precision.

ANN algorithms were developed as an alternative choice of BCS calibration. An ANN can assimilate any complex system with several layers of neural elements. It makes the analysis of extremely complex systems possible; however, the models obtained from ANN methods provide only input-output relationships, and they do not offer the structures and the inside parameters of the system.

The major trend of BCS calibration development is to improve the speed, consistency, and precision of the algorithms, and to improve the algorithms to adapt to low-cost cameras and measuring equipment.

Stereo matching algorithms are divided into three groups, which are feature based, correlation based, and energy-minimization based algorithms.

Feature-based stereo matching obtains object clusters using segmentation, and matches objects based on feature vectors. It is suitable for fast algorithms of object localization. The most popular segmentation methods are Mean-shift segmentation and edge detection based segmentation. In addition to the basic feature-based stereo-matching methods, some technologies, such as Plane fitting, adaptive methods, foveal and peripheral camera combination, and ground control points, were developed to increase the performance. Feature-based stereo matching is a stereo-matching technology on the segment level, so that it is fast and is suitable for real-time systems.

Energy-minimization methods are a group of methods to seek the best disparity map holding the minimum energy at the global scale. These methods get the most precise stereo matching for the whole image, but they need heavy computing load.
The methods are suitable for static image processing, but are unsuitable for a real-time vision system. The most popular energy-minimization methods are belief propagation and graph-cuts.

Correlation-based stereo-matching methods are pixel scale local matching methods, and their precision and speed are between the other two groups. The two major groups of correlation-based stereo-matching methods are line segmentation and dynamic programming.

Some other performance-improving technologies were developed recently, which are pixel dissimilarity, mutual information, cooperative programming, and census-based correlation.

Feature-based stereo matching is the fastest among the three groups. It is suitable for robot vision systems. Segmentation is a major part of the algorithm. Improving the speed and the reliability of segmentation is critical to real-time robot vision and control.

Ripe tomato recognition algorithms are based only on the colour information to date. L*\(a*b*\) colour space is used to eliminate the effect of light intensity, which is sensitive to light conditions. The algorithms use the elements corresponding to tomato ripeness. Tomato localization algorithms can be derived from Epipolar geometry.

Although tomato localization algorithms can be derived from Epipolar geometry, the derivation from rectified images needs more efforts, and it is not shown in the literature to date.

To conclude, increasing the speed and consistency is a continuous trend both in BCS calibration and in stereo matching, and it is mandatory for a ripe-tomato recognition and localization system to move to industrial applications. This trend is also the focus of this project.
Chapter 3

Calibration of the Binocular Camera System

3.1 Introduction

Calibration is the prerequisite for stereo matching. A binocular camera system is constructed with two cameras, the left camera (LC) and the right camera (RC). Theoretically, both cameras would be symmetrical to each other; however, it is expensive to make the cameras symmetrical. Most binocular cameras have both displacive and angular differences, which result in the displacive and angular differences in the images captured by the cameras. Figure 1.2 is an example of a pair of binocular images in which the vertical positions of the same template board are apparently different in both images. Algorithms used to obtain the spatial relationship between the dual-cameras are called spatial calibration, or geometrical calibration. A majority of researches are focused on this kind of calibration.

Another aspect of calibration is related to the colour difference between both the cameras. Most commercial level cameras have different colour parameters. For example, the same object has different image colours in different cameras. The colour changes in different images even if the images are captured using the same camera. Most binocular camera stereo matching is based on colour matching. The colour difference between the two cameras should be determined in the first place prior to
colour matching. The process to locate the laws governing the colour difference is called colour calibration. Because colour differences happen in each capture, it is not fixed as a camera’s property. It is more appropriate to call it image colour calibration than camera colour calibration, so that this part of calibration is elaborated in another chapter.

As described above, calibration includes spatial calibration and colour calibration. It is a dilemma to decide which calibration should be performed first between spatial calibration and colour calibration. Any image processing, including spatial calibration, is based on the colour information. It is reasonable that colour calibration should be performed before spatial calibration; however, colour calibration is a process to determine the colour difference between both cameras; correlated pixels in different images from the same object point are used to perform colour calibration. It is hard to locate correlated pixels without the BCS being spatially calibrated.

To solve this problem, a spatial calibration algorithm is proposed that does not strongly rely on the colour information, so that spatial calibration can be performed before colour calibration.

The proposed algorithm focuses on one salient point of an object. The projection points of the salient point into both camera images, \((x_1, y_1), (x_2, y_2)\), and the distance \((D)\) between the point and the origin of the LC are used as input of the algorithm.

The salient point is obtained from a corner of a piece of white paper in this project.

There are several ways to obtain the input data for BCS calibration. One way is to use a piece of rod (Medioni & Kang 2005), and another way is to use a printed pattern template (Zhang 2000). The algorithm used in this project has the following advantages over the above existing algorithms.

1. This algorithm uses only a piece of white paper with green paper background attached to a plate (Figure 1.2). This calibration device is the cheapest one, and it is the easiest to be made, because no high accuracy machining or assembling process is requested.

2. A straight line is easy to be extracted because it is an edge between two segments
with different colours (white and green).

3. A corner point can be easily calculated from the intersection of two lines.

4. Lines are apparently long, so that there are enough extracted edge points to support the lines. This feature makes the algorithm robust to noise while keeping the precision.

BCS spatial calibration is based on Epipolar geometry in most literature. Epipolar geometry describes the relative positions of the two cameras against each other. When Epipolar geometry parameters are known, the real 3D positions of objects can be computed from their images.

Equations used for BCS calibration is built based on Epipolar geometry in the following sections, and an innovative BCS calibration algorithm with asymmetric data flow and an asymmetric energy function is proposed.

3.2 Definitions and Expected Output

In a robotic ripe tomato harvesting system, the objective of the calibration of the BCS is to find out the position and orientation of each of the two cameras with reference to the robot coordinate system. The expected output of the calibration includes the position and orientation of the LC with reference to the robot coordinate system and the position and orientation of the RC with reference to the robot coordinate system.

In this chapter, only the RC’s position and orientation with reference to the LC is calibrated for stereo matching purpose. The position and orientation of the LC with reference to the robot coordinate system is obtained in Section 5.4. When the above two types of calibration is completed, the position and orientation of the RC with reference to the robot can be derived from the results, thus the BCS calibration in robot system is completed.

The following is a detailed description of the expected output with the assistance of the definition of the camera coordinate systems. The definition of a camera coordinate system is shown in Figure 3.1.
The 3D camera coordinate system in Figure 3.1 is fixed to the camera, and is defined as follows:

- The origin point is coincident to the focal point of the camera;
- $X_i$-axis points to the right direction of the camera;
- $Y_i$-axis points to the bottom direction of the camera;
- $Z_i$-axis points to the front direction of the camera.

The objective of the BCS calibration in this chapter is to find the position and orientation of the RC with reference of the LC, the expected output includes three displacements and three rotating angles as $\alpha$, $\beta$, $\gamma$, $T_1$, $T_2$, $T_3$. They are defined as follows. Suppose the RC is coincident with the LC initially, and it is transferred through the following steps:

- **Step 1.** Rotate the RC for an angle of $\gamma$ around $Z_2$-axis;
- **Step 2.** rotate the RC for an angle of $\beta$ around $Y_2$-axis;
- **Step 3.** rotate the RC for an angle of $\alpha$ around $X_2$-axis;
- **Step 4.** translate the RC for an offset of $T$ in the RC coordinate system, where

$$
T = \begin{pmatrix} T_1 & T_2 & T_3 \end{pmatrix}^T. 
$$

In summary, the vector $\alpha$, $\beta$, $\gamma$, $T_1$, $T_2$, $T_3$ is the expected output of BCS calibration. All the geometrical parameters used in the following chapters can be obtained
from this vector. For example, the baseline, or the distance between the two cameras is $B = \sqrt{T_1^2 + T_2^2 + T_3^2}$.

### 3.3 Spatial Relationship between Two Cameras

This section describes a general model for the spatial relationship between the two cameras before calibration. This section is based on Epipolar Geometry theory. Detailed explanations about Epipolar Geometry theory can be found in Medioni & Kang (2005). Some adaptations are made to suit the requirement of this project.

Any 3D point $G = \begin{pmatrix} X & Y & Z & 1 \end{pmatrix}^T$ in a world coordinate system can be projected onto a camera’s image plane to form a 2D point as $U = \begin{pmatrix} u & v & 1 \end{pmatrix}^T$. There is a $4 \times 3$ matrix $C$ to link the two points as

$$\lambda U = CG,$$

where $\lambda$ is a scalar value representing the depth of the 3D point. For the convenience of derivation, let

$$
C = \begin{pmatrix}
A & B \\
H & -
\end{pmatrix},
G = \begin{pmatrix}
1
\end{pmatrix},
$$

where $A$ is a $3 \times 3$ matrix representing the rotation of the camera against the world coordinate system, and $H = \begin{pmatrix} X & Y & Z \end{pmatrix}^T$ is a 3D vector representing the coordinates (world or camera) of the 3D point, and $B$ is another 3D vector representing the spatial translation, then Equation (3.2) becomes

$$\lambda U = A H + B.$$
For a designated camera, Equation (3.5) can be rewritten as

\[ \lambda_i U_i = A_i H + B_i, \]

which is the universal model of camera systems. In the BCS case, this equation becomes

\[
\begin{align*}
\lambda_1 U_1 &= A_1 H + B_1, \\
\lambda_2 U_2 &= A_2 H + B_2
\end{align*}
\]

where \( A_1, B_1, A_2, \) and \( B_2 \) are property parameters of the BCS, which are called BCS properties; \( U_1, U_2, \) and \( H \) are object and image data that are either measured or captured using cameras, which are called observation data. The purpose of BCS calibration is to decide the BCS properties from the observation data.

It can be found from Equation (3.7) that there are 24 BCS property parameters that are unknown and yet to be obtained by solving the equations. For each set of observation data, six equations can be created; however, only two new immediate parameters, \( \lambda_1 \) and \( \lambda_2 \), are created for each observation data set if \( U_1, U_2, \) and \( H \) are all known. There should be at least six sets of observation data to make the unknown parameters fully determined. If the observation data is more than six sets, an optimization method can be used to obtain the optimized result.

This universal model appears effective for BCS calibration, but it is not practical, because there are too many unknown parameters that give rise to heavy computation load. Besides, the relationship between the observation data and the BCS properties is too weak, so that the iterative process is inconsistent, and is difficult to converge. Modifications should be found to build a practical process, which is described in the next section.

### 3.4 Modification of the Binocular Camera System Model

The main purpose of BCS model modification is to reduce the number of undecided BCS property parameters by introducing some decided ones. A new structure of the
BCS model is derived based on some proven knowledge of cameras.

Beside the camera coordinate system, camera plane coordinate system is also useful in BCS calibration. Camera plane is an imaging plane showing the geometrical relationship between the camera image and the camera coordinate system. Refer to Figure 3.1, camera plane is a plane perpendicular to Z-axis with an offset \( f \) from the origin point of the camera. \( f \) is also called focal length.

The 2D camera-plane coordinate system in Figure 3.1 is defined as bellow:

- The origin point is at the intersection of \( Z_i \)-axis and the camera plane (the central point of the camera plane in this case);
- \( u_i \)-axis points to the right direction of the camera plane;
- \( v_i \)-axis points to the bottom direction of the camera plane.

Let \( \mathbf{H}_j \) be a 3D point’s coordinates in a camera coordinate system (see Section 3.2). To simplify the derivation, suppose the world coordinate system is the same as the camera coordinate system of one of the cameras, then \( \mathbf{H} \) in Equation (3.6) can be replaced by \( \mathbf{H}_j \). Equation (3.6) becomes

\[
\lambda_{ij} \mathbf{U}_i = \mathbf{A}_{ij} \mathbf{H}_j + \mathbf{B}_{ij}. \tag{3.8}
\]

For the camera coordinate system, it has

\[
\begin{align*}
\mathbf{A}_i &= \begin{pmatrix}
    f_i & 0 & 0 \\
    0 & f_i & 0 \\
    0 & 0 & 1
\end{pmatrix}, \\
\mathbf{B}_i &= \begin{pmatrix}
    0 & 0 & 0
\end{pmatrix},
\end{align*}
\tag{3.9}
\]

where \( f_i \) is the focal length of the camera. Detailed derivation refers to Medioni & Kang (2005). Combining Equations (3.8) and (3.9), it gives

\[
\lambda_{ii} \mathbf{U}_i = \begin{pmatrix}
    f_i & 0 & 0 \\
    0 & f_i & 0 \\
    0 & 0 & 1
\end{pmatrix} \mathbf{H}_i. \tag{3.10}
\]

In the BCS case, without loss of generality, the world coordinate system is assumed
to be the same as the 3D camera coordinate system of the LC, i.e., \( H = H_1 \). Now the relationship between \( H_1 \) and \( H_2 \) is derived as follows.

If the BCS is an ideal one, the position of the RC would be an exact offset of the LC along \( X \) axis. Actually, the RC has offsets along all the axes and has angles against the LC as well. The offsets and angles are defined in Section 3.2 as \( \alpha, \beta, \gamma, T_1, T_2, T_3 \). Following the steps described in Section 3.2, the BCS model is derived as

\[
H_2 = \begin{pmatrix}
1 & 0 & 0 \\
0 & \cos \alpha & \sin \alpha \\
0 & -\sin \alpha & \cos \alpha \\
\end{pmatrix}
\begin{pmatrix}
\cos \beta & 0 & -\sin \beta \\
0 & 1 & 0 \\
\sin \beta & 0 & \cos \beta \\
\end{pmatrix}
\begin{pmatrix}
\cos \gamma & \sin \gamma & 0 \\
-\sin \gamma & \cos \gamma & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
H_1 - T.
\]

Let

\[
R = \begin{pmatrix}
1 & 0 & 0 \\
0 & \cos \alpha & \sin \alpha \\
0 & -\sin \alpha & \cos \alpha \\
\end{pmatrix}
\begin{pmatrix}
\cos \beta & 0 & -\sin \beta \\
0 & 1 & 0 \\
\sin \beta & 0 & \cos \beta \\
\end{pmatrix}
\begin{pmatrix}
\cos \gamma & \sin \gamma & 0 \\
-\sin \gamma & \cos \gamma & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
, \quad (3.12)
\]

then

\[
H_2 = RH_1 - T.
\]

(3.13)

Equations (3.10) and (3.13) together represent the modified BCS model, and they give the relationship between the two cameras.

For abbreviation, \( \lambda_i \) is used to represent \( \lambda_{ii} \), and for convenience of analysis, Equations (3.10) and (3.13) is rewritten as

\[
\lambda_1 U_1 = \begin{pmatrix}
f_1 & 0 & 0 \\
0 & f_1 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
H_1
\]

\[
H_2 = RH_1 - T
\]

(3.14)

\[
\lambda_2 H_2 = \begin{pmatrix}
f_2 & 0 & 0 \\
0 & f_2 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
U_2
\]

32
where \( R \) refers to Equation (3.12).

The above equations represent a modified model of the BCS. It is already known that \( f_1 = f_2 = 860 \text{ pixel} \) from measurement that is described later. The remaining unknown BCS property parameters are \( \alpha, \beta, \gamma, T_1, T_2, \) and \( T_3 \), totally six unknown parameters. Compared to the original model of the BCS used by most other BCS calibration algorithms including 24 unknown parameters, the computation load of this model is dramatically reduced.

The focal length of a camera is measured in this way (See Figure 3.2): place a line with already known length \( L \) in front of the camera. The distance from the line to the origin of the camera is measured as \( D \), and the image length of the line is measured as \( l \). The focal length of the camera is

\[
\frac{d}{L} = \frac{DL}{L}.
\]

Figure 3.2: Illustration of focal length measurement. \( L \): the length of a sample line; \( D \): the distance from the line to the origin of the camera; \( l \): the length of the image of the sample line; \( f \): the focal length of the camera.

### 3.5 Asymmetric Data Flow and Energy Function

The BCS constraint equations are obtained in Section 3.4. When a group of equations are given from data sampling, an algorithm can be used to solve the equations.
This project uses optimization to solve the equations and obtain the unknown BCS parameters.

The principle of optimization is to seek a parameter set of a system that makes the energy function reach its minimum or maximum value. The energy function is also called the cost function; it is a value that is expected to be as little as possible. It is always related to the error between the algorithm’s expected output and the system’s real output. There is no explicit output value for equation solving problems such as BCS calibration; many values could be output, input, or immediate. Choosing the appropriate output value as well as the energy function is challenging because it is critical to the data flow and performance of an algorithm. Various data flow and energy functions have been chosen in different works, which result in various consistency, performance, and precision.

From an investigation of the existing algorithms, it can be found that, the inconsistency of a system is caused by high ratio signal amplification. Most existing BCS calibration algorithms have one step of computation in which 3D object coordinates are computed from 2D image coordinates on both camera planes. A small noise from image processing could create big errors in 3D object coordinates, especially when the object is far from the cameras. This project proposes asymmetric data flow and an asymmetric energy function to avoid the data flow from small value variables to big value ones, so as to avoid the noise amplification.

### 3.5.1 Asymmetric Data Flow and Symmetric Data Flow

Figure 3.3 is an illustration showing the difference between asymmetric data flow and symmetric data flow. Most traditional BCS calibration algorithms are based on symmetric data flow, where the input of the data flow consists of all the coordinates of the projections of the 3D sample point (G in Figure 3.3) onto both the camera planes (P₁ and P₂ in Figure 3.3), the output consists of the computed coordinates of the 3D sample point (Ĝ in Figure 3.3) through the data flow. The data flow is symmetric between the LI and the RI. The energy function (E) is the distance between the computed coordinates of the 3D sample point (Ĝ) and the measured
coordinates of the real 3D sample point \((G)\). It is also symmetric between both the images.

In symmetric data flow, the coordinates of the 3D sample point as big values are computed from small values, the coordinates of the projection of the 3D sample point onto the images. Any noise can be amplified, so that algorithms based on symmetric data flow are inconsistent. Asymmetric data flow proposed in this project (Figure 3.3) does not include any step that transforms data from small-value variables to big-value variables, it is consistent.

The input of the asymmetric data flow consists of the projection of the 3D sample point onto the LI \((P_1)\) and the distance \((D)\) between the 3D sample point and the origin of the LC. The coordinates of the 3D sample point are computed from the input, then the coordinates of the projection of the 3D sample point \(G\) onto the RI \((P_2)\) are computed (as \(\hat{P}_2\)) from the above computed result of \(G\). Because the input of the data flow includes a big value variable \(D\), the noise amplification is restrained, so
that the algorithm is more consistent than most of traditional algorithms.

3.5.2 The Structure of the Data Flow and the Energy Function

The structure of the data flow is inspired by the ANN. Although it is an analytical system, it has the similar structure as an ANN, so that ANN concepts are used for convenience of description and understanding.

The unknown parameters in the BCS are $\alpha$, $\beta$, $\gamma$ from Equation (3.12) and $T_1$, $T_2$, $T_3$ from Equation (3.1). A vector of BCS property parameters can be constructed as

$$ W_{BCS} = \begin{pmatrix} \alpha & \beta & \gamma & T_1 & T_2 & T_3 \end{pmatrix}^T, $$

which is the counterpart of the weights in the ANN.

Similar to the ANN, the data flow also has an input layer, one or more immediate layers, and an output layer.

In Zong et al. (2006), the input layer is the image of a 3D point projected onto both cameras’ image planes, the output layer is the 3D point coordinates (Figure 2.1). In this work, the coordinates of the 3D point have to be measured, so that large coordinate measuring equipment is requested. In this project, a new method to construct the layers of the data flow is proposed, which eliminates the use of 3D coordinate measuring equipment.

The proposed method uses the projections of 3D points onto camera planes as its input. The projections are sampled from captured stereo images. Each image has its own coordinate system called the image coordinate system (Figure 3.4), which is defined as follows: The origin point is located at the centre of the bottom-left pixel, the $x_i$-axis points to the right direction of the image, and the $y_i$-axis points to the top direction of the image.

The relationship between a camera-plane coordinate system and the image coordinate system of an image captured by the same camera is also shown in Figure 3.4,
Figure 3.4: The relationship between two coordinate systems, the camera-plane coordinate system and the image one. \( u_i, v_i \): camera-plane coordinates; \( x_i, y_i \): image coordinates.

and it is

\[
\begin{align*}
    u_i &= x_i - \frac{M-1}{2}, \\
    v_i &= -y_i - \frac{N-1}{2},
\end{align*}
\]  

(3.17)

where \( M \) and \( N \) are the width and the height of the image.

Assume a BCS has captured images of a 3D point, then the coordinates of the projections of the 3D point onto the LI and the RI are \((x_1, y_1)\) and \((x_2, y_2)\), respectively, and the distance from the 3D point to the origin of the LC is measured as \( D \) simultaneously. A vector

\[
X_{DF} = \begin{pmatrix} x_1 & y_1 & D \end{pmatrix}^T
\]  

(3.18)

is used as the input layer of the data flow.

There are 2 immediate layers that correspond to the 3D coordinates of the 3D point in both the camera coordinate systems, which are

\[
\begin{align*}
    H_1 &= \begin{pmatrix} X_1 & Y_1 & Z_1 \end{pmatrix}^T = F_1(X_{DF}, f_1) \\
    H_2 &= \begin{pmatrix} X_2 & Y_2 & Z_2 \end{pmatrix}^T = F_2(H_1, W_{BCS})
\end{align*}
\]  

(3.19)

The details of functions \( F_1, F_2, \) and \( F_3 \) (appears later) is defined in Subsection 3.5.3.

The output layer is a vector of the image coordinates of the 3D point in RI
calculated from the data flow (It is not the actual image coordinates):

$$Y_{DF} = \left( \hat{x}_2 \quad \hat{y}_2 \right)^T = F_3(H_2, f_2).$$  \hspace{1cm} (3.20)

The error function is the distance between the calculated image coordinates of the 3D point in the RI and the actual image coordinates captured in the RI,

$$E = (\hat{x}_2 - x_2)^2 + (\hat{y}_2 - y_2)^2,$$  \hspace{1cm} (3.21)

where the square root part is eliminated to simplify the computation. The data flow is shown in Figure 3.5. It is called asymmetric data flow because it starts from the LI and ends with the RI.

3.5.3 Details of the Functions in Data Flow

The derivation of $F_1$:

Apply Equation (3.10) to the LC, it gives

$$\lambda_1 U_1 = \begin{pmatrix} f_1 & 0 & 0 \\ 0 & f_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} H_1.$$  \hspace{1cm} (3.22)
It is apparent that

\[ D = \sqrt{X^2 + Y^2 + Z^2}. \]  \hspace{1cm} (3.23)

Combine Equations (3.17), (3.22), and (3.23), then perform a couple of derivations, the function \( F_1 \) is obtained as

\[
F_1 \mid (X_{DF} \mapsto H_1) = \begin{cases} 
    u_1 = x_1 - \frac{M-1}{2} \\
    v_1 = -y_1 - \frac{N-1}{2} \\
    Z_1 = \frac{D}{\sqrt{\frac{u_1^2}{f_1^2} + \frac{v_1^2}{f_1^2} + 1}}. \\
    X_1 = \frac{u_1}{f_1} Z_1 \\
    Y_1 = \frac{v_1}{f_1} Z_1 
\end{cases} \hspace{1cm} (3.24)
\]

Function \( F_2 \) already exists in Equation (3.13) as

\[
F_2 \mid ((H_1, W_{BCS}) \mapsto H_2) = \begin{cases} 
    H_2 = R H_1 - T. 
\end{cases} \hspace{1cm} (3.25)
\]

Function \( F_3 \) is derived from Equations (3.14.3) and (3.17), as

\[
F_3 \mid ((H_2, f_2) \mapsto Y_{DF}) = \begin{cases} 
    u_2 = f_2 \frac{X_2}{Z_2} \\
    v_2 = f_2 \frac{Y_2}{Z_2} \\
    \hat{x}_2 = u_2 + \frac{M-1}{2} \\
    \hat{y}_2 = -v_2 - \frac{N-1}{2} 
\end{cases} \hspace{1cm} (3.26)
\]

Actually, the three functions can be combined together to form a comprehensive function as

\[
F_{DF} \mid ((X_{DF}, W_{BCS}) \mapsto Y_{DF}) = \begin{cases} 
    H_1 = F_1 (X_{DF}, f_1) \\
    H_2 = F_2 (H_1, W_{BCS}) \\
    Y_{DF} = F_3 (H_2, f_2) 
\end{cases} \hspace{1cm} (3.27)
\]

Equation (3.27) is the core function of the data flow. It is implemented as a subroutine in the BCS calibration program.
3.6 Calibration Algorithm for the Binocular Camera System

The input data structure is \( \{x_1, y_1, D, x_2, y_2\} \), which is the information of a 3D point extracted from captured dual-images by the BCS. Equations (3.21) and (3.27) can be further combined to form the criterion function

\[
F_E| ((x_1, y_1, D, x_2, y_2, \alpha, \beta, \gamma, T_1, T_2, T_3) \rightarrow E)
\]

\[
\begin{align*}
X_{DF} & = \begin{pmatrix} x_1 & y_1 & D \end{pmatrix}^T \\
W_{BCS} & = \begin{pmatrix} \alpha & \beta & \gamma & T_1 & T_2 & T_3 \end{pmatrix}^T \\
Y_{DF} & = F_{DF}(X_{DF}, W_{BCS}) \\
E & = (\hat{x}_2 - x_2)^2 + (\hat{y}_2 - y_2)^2
\end{align*}
\]  \hspace{1cm} (3.28)

If define

\[
Y_{Actual} = \begin{pmatrix} x_2 & y_2 \end{pmatrix}^T
\]  \hspace{1cm} (3.29)

and function \( F_4 \) as

\[
F_4| (\langle Y_{DF}, Y_{Actual} \rangle \rightarrow E) \begin{cases} E = (\hat{x}_2 - x_2)^2 + (\hat{y}_2 - y_2)^2, \end{cases}
\]  \hspace{1cm} (3.30)

then the criterion function can be represented as \( F_E(X_{DF}, Y_{Actual}, W_{BCS}) \), and Equation (3.28) becomes

\[
F_E| (\langle X_{DF}, Y_{Actual}, W_{BCS} \rangle \rightarrow E) \begin{cases} Y_{DF} = F_{DF}(X_{DF}, W_{BCS}) \\
E = F_4(Y_{DF}, Y_{Actual}) \end{cases}
\] \hspace{1cm} (3.31)

Optimization is used to solve the equations. An algorithm is implemented as Algorithm 1. For convenience of programming, \( W_{BCS} \) is represented as a standard array, e.g., \( W_{BCS}[1] = \beta \).

Up to now, the BCS calibration algorithm is fully formed; however, this algorithm has a prerequisite that the input data needs to be ready to use. The preparation of the input data is discussed in the following section.
Algorithm 1: BCS calibration

Data: \( W_{BCS} \) ← values from a previous process cycle, or, \( W_{BCS} ← 0 \) if no previous values are available

Result: Converged \( W_{BCS} \)

begin
Create an array of \( W_{BCS} \) in the form of \( A_w[13] \) and an array of error values in the form of \( E[13] \) to save BCS parameters and error values for the BCS parameter sets with different offset values

Value \( e = \) a random number > \( e' \) (a tolerance value)

while \( e > e' \) do

for each set of input data do

\( W_{BCS0} = W_{BCS} \) (keep a record of the original BCS parameters)
\( W_{BCS} = W_{BCS0} \)
\( A_w[0] = W_{BCS} \)
\( E[0] = F_E(X_{ANN}, Y_{Actual}, W_{BCS}) \)

for \( i \) from 0 to 5 do

\( W_{BCS} = W_{BCS0} \)
\( W_{BCS}[i] = W_{BCS}[i] - \delta \)
\( A_w[i + 1] = W_{BCS} \)
\( E[i + 1] = F_E(X_{ANN}, Y_{Actual}, W_{BCS}) \)

for \( i \) from 0 to 5 do

\( W_{BCS} = W_{BCS0} \)
\( W_{BCS}[i] = W_{BCS}[i] + \delta \)
\( A_w[i + 7] = W_{BCS} \)
\( E[i + 7] = F_E(X_{ANN}, Y_{Actual}, W_{BCS}) \)

Scan all the elements in \( E[13] \), and locate the element \( E[i] \) with the maximum value

\( e = E[i] \)

\( W_{BCS} = A_w[i] \)
3.7 Data Preparation for Calibration

The input of the BCS calibration algorithm is a series of data set as \( \{x_1, y_1, x_2, y_2, D\} \) (See Section 3.6). Each set of data includes the image coordinates of a 3D point captured by both cameras and the measured distance between the 3D point and the origin of the LC. This section discusses the algorithm of extracting the data set from a pair of binocular images.

The input data extraction program is composed of three parts, edge detection, line matching, and point calculation.

3.7.1 Edge Detection

Generally, the most popular method for edge detection is Canny edge detector; however, in this project, a specific edge detection algorithm is proposed to detect only the interesting edge points of the white paper in the green background (Algorithm 2). The edge points are shown as red points in Figure 3.6.

3.7.2 Line Extraction

Although the edge points are found by edge detection, they are only a stack of points in the computer sense. The computer does not know it is a line in spite of a human being doing this well. The objective of line extraction is to create a parameterized line that matches the edge points appropriately.

The geometric model of a line is constructed in this way: First, draw a line through the origin of the image with an angle \( \theta \); then, move the line by an offset variable \( s \) (as in Figure 3.7). The line is represented as

\[
\begin{align*}
  x &= t \cos \theta - s \sin \theta, \\
  y &= t \sin \theta + s \cos \theta,
\end{align*}
\]

(3.32)

where \( t \) is a free variable ranged from \(-\infty\) to \(+\infty\).

The next task is to match the line with the obtained edge points. Traditionally, the LSM is used for matching. In the LSM, the square root of the distance between
Algorithm 2: Edge Detection

Data: An image of a peace white paper with green background

Result: A collection of edge points

begin

Start the edge detection process along the vertical direction
The image is divided into grids with a size of 16 × 16

for each grid do

Check the colour of the top-middle pixel and the bottom-middle pixel

if the colour of the top-middle pixel ≠ the colour of the bottom-middle pixel & both of them are white or green then

Check the colour of pixels starting from the top of the grid, and go down pixel by pixel, if the colour of the current pixel is different from the colour of the previous pixel, mark the current pixel as an end pixel
Repeat the above process from the bottom of the grid in the opposite direction, obtain another end pixel
Set an edge point as the middle point of the line linking the two end pixels

Start the edge detection process along the horizontal direction
Repeat the steps similar to the ones in the vertical direction
each point and the line is calculated, and all the square roots are added up, then the
line is moved by translating and/or rotating to make the sum reach a minimum value;
however, LSM has a drawback: generally, points are ranged from the most important
ones, which are located very close to the line, or, in other words, lie on the line, to the
least important ones, which are a little bit far away from the line. If LSM is used, the
least important points always have a bigger distance from the line, and therefor they
have bigger weights to the evaluation function than the most important points. This
situation is opposite to the reasonable performance. Furthermore, it is undetermined
about which point belongs to which line before matching, so that a point belonging
to one line may have much more effect upon the matching result of another line.

To solve this problem, a different evaluation function is proposed to reflect the

Figure 3.6: *Extracted edge points, which are represented by red spots.*
distance from a point to a line, which is

\[ E = \sum_i e^{-d_i}, \]

(3.33)

where \( d_i \) is the distance from point \((x_i, y_i)\) to line \((s, \theta)\),

\[ d_i = |s + x_i \sin \theta - y_i \cos \theta|. \]

(3.34)

It can be induced from Equation (3.34) that, the output of the function is bigger when the distance is smaller, and vice versa.

The matching process is performed with all the points based on the proposed algorithm. it is not necessary to classify which line a point belongs to.

Theoretically, the line’s initial location could be anywhere, and it can converge to the matching points; however, it does not always converge to the expected location in the experiments of this project. There are some hidden factors that need to be investigated further. So far, if the line matching is started with the line’s initial location crossing two randomly selected points that are close to each other (whether or not they belong to one line), the lines converge very well to the expected positions (Figure 3.8).
Corner points are among the intersections of lines. The following is the equation of point calculation, which is derived from Equation (3.32).

\[
\begin{align*}
x &= \frac{s_1 \cos \theta_2 - s_2 \cos \theta_1}{\sin (\theta_2 - \theta_1)} \\
y &= \frac{s_1 \sin \theta_2 - s_2 \sin \theta_1}{\sin (\theta_2 - \theta_1)}
\end{align*}
\] (3.35)

Usually, four lines are extracted from an image; however, in some cases, more than four lines could be extracted if there are any edges that appear to be the expected edges but they are some other edges actually, so that it is possible that more than four intersection points are obtained but some are not the expected corner points. The challenge is how to filter out the non-corner points.

This project proposes an algorithm to recognize a point as an expected corner
point or not. In the algorithm, four satellite points are assigned for each intersection point using the following steps (Figure 3.9):

The first satellite point is an offset of the intersection point along the direction of the angle \((\theta_2 + \theta_1)/2\), and the offset value is 16 pixel units. Other 3 satellite points are rotations of the first satellite point according to the intersection point with an angle index of \(\pi/2\), \(\pi\), and \(3\pi/2\). Check the colour of the satellite points (or the average colour of a window for the robust purpose), if exactly one satellite point belongs to the white region and three satellite points belong to the green region, the intersection point is judged as an expected corner point, and vice versa.

![Figure 3.9: Satellite points for corner recognition. All the satellite points are positioned between the lines. If exactly one satellite point belongs to the white region and three satellite points belong to the green region, the intersection point is judged as an expected corner point, and vice versa.](image)

The real corner points are exactly identified using this algorithm (see blue points in Figure 3.10).

Furthermore, the direction of each corner point is defined as the direction from the corner point to the satellite point whose colour is white, so that the direction always points toward the inside of the white paper frame.

If the directions of two corner points from both LI and RI are close to each other, they are judged to be the images of the same corner. This correspondence is used for stereo matching in BCS calibration.

Up to now, the four corner points from both images are successfully extracted, and
Figure 3.10: Lines and corners extracted from both images. The blue square blocks represent corners.

the correspondence of the points between both images is obtained. A data structure can be constructed as in Figure 3.11.

If one capture is successful, four input data instances will be extracted. In some cases, if a corner is too close to the image edge, or is located outside the image, fewer instances will be extracted. If one capture is not successful, the system will automatically skip the process and waits for the next capture.

3.8 Comparison of the Algorithm with Existing Algorithms

The data flow proposed in this project is asymmetric, which starts from the LC and ends with the RC, while most of the existing algorithms use symmetrical data flow, which starts from both cameras and ends to 3D object coordinates, or in the reversal direction. When 3D object coordinates are included either in the input or in the output, expensive and high accuracy 3D coordinate measuring equipment is mandatory for the process. This project uses only image data and the distance from the 3D point to the origin of the LC as input and output, eliminating the measuring
A majority of existing algorithms start from image data to 3D object coordinates. Image data has small values while 3D object coordinates are large values. Noise is inevitably amplified when data is transported from small-value variables to large-value ones. This amplification makes the algorithms inconsistent. In this project, the 3D object coordinates are computed based on a large-value variable, which is the distance of the 3D point to the origin of the LC, as one of the input values, so that the noise amplification is eliminated.

The data flow of the algorithm of this project is originally inspired by the ANN, so that it has the similar structure as the ANN. In the original ANN, there are a set of variables called Weights \( (w_{ij}) \), which linearly determine the flow rate of the information stream from input layer cells to hidden layer cells, and from hidden layer cells to output layer cells. In other words, the values in \( w_{ij} \) linearly determine the relationship between adjacent cells, so as to construct a network. In this algorithm, the weights are directly corresponding to physical parameters. For example, in the BCS, the weights \( \alpha, \beta, \text{ and } \gamma \) have physical meanings, while general weights rarely
have physical meanings.

The advantage of this data flow structure over the ANN is that, when the data flow has converged or, in the other word, the weights are stabilized, the physical parameters become known, and it provides a clear view of the structure of the system. A majority of further artificial intelligent works can be done based on the obtained structures. For example, stereo matching and object identification can benefit from this data flow structure.

### 3.9 Experimental Results

Several experiments are carried on. The experiments consist of two parts, input data preparation and BCS calibration. There are 51 sets of input data extracted from the captured images by a BCS (Table 3.1), and the proposed algorithm is applied to the calibration of the BCS. The initial weights and resulting weights are shown in Table 3.2.

### 3.10 Summary

An innovative binocular camera system calibration algorithm is developed in this chapter, which can automatically obtain the position and the orientation of the right camera with reference to the left camera. A new algorithm with asymmetric data flow and an asymmetric energy function is proposed. The data flow starts from the left camera and ends with the right camera. This algorithm does not require the measuring equipment, thus reducing the cost of calibration. This algorithm avoids the data flow from small-value variables to large-value ones, making itself more consistent than most existing algorithms.

The data flow of the algorithm of this project is originally inspired by the ANN, but it uses real physical parameters as weights in an ANN. The advantage of this data flow structure over the ANN is that, when the system has converged, the physical parameters become known; it can provide a clear view of the structure of the system,
Table 3.1: Part of the input data for calibration

<table>
<thead>
<tr>
<th>variable name</th>
<th>$x_1$</th>
<th>$y_1$</th>
<th>$x_2$</th>
<th>$y_2$</th>
<th>$D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>unit</td>
<td>pixel</td>
<td>pixel</td>
<td>pixel</td>
<td>pixel</td>
<td>cm</td>
</tr>
<tr>
<td>1:</td>
<td>484.1</td>
<td>346.9</td>
<td>278.7</td>
<td>387.1</td>
<td>27.94</td>
</tr>
<tr>
<td>2:</td>
<td>280.4</td>
<td>346.7</td>
<td>81.1</td>
<td>382.7</td>
<td>27.94</td>
</tr>
<tr>
<td>3:</td>
<td>72.3</td>
<td>311.2</td>
<td>329.0</td>
<td>376.7</td>
<td>27.94</td>
</tr>
<tr>
<td>4:</td>
<td>547.2</td>
<td>336.6</td>
<td>329.0</td>
<td>376.7</td>
<td>27.94</td>
</tr>
<tr>
<td>5:</td>
<td>230.7</td>
<td>52.2</td>
<td>206.1</td>
<td>88.8</td>
<td>152.40</td>
</tr>
<tr>
<td>6:</td>
<td>491.4</td>
<td>68.2</td>
<td>437.0</td>
<td>109.7</td>
<td>152.40</td>
</tr>
<tr>
<td>7:</td>
<td>70.0</td>
<td>35.2</td>
<td>34.4</td>
<td>67.3</td>
<td>457.20</td>
</tr>
<tr>
<td>8:</td>
<td>109.0</td>
<td>30.1</td>
<td>74.1</td>
<td>68.3</td>
<td>457.20</td>
</tr>
<tr>
<td>9:</td>
<td>566.4</td>
<td>382.7</td>
<td>536.0</td>
<td>418.4</td>
<td>457.20</td>
</tr>
<tr>
<td>10:</td>
<td>88.5</td>
<td>381.6</td>
<td>58.7</td>
<td>412.9</td>
<td>457.20</td>
</tr>
<tr>
<td>11:</td>
<td>279.0</td>
<td>232.3</td>
<td>248.3</td>
<td>253.3</td>
<td>368.30</td>
</tr>
<tr>
<td>12:</td>
<td>414.3</td>
<td>201.3</td>
<td>360.9</td>
<td>229.4</td>
<td>152.40</td>
</tr>
<tr>
<td>13:</td>
<td>317.2</td>
<td>211.9</td>
<td>112.0</td>
<td>240.8</td>
<td>27.94</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

while an ANN can not.

An innovative point extracting algorithm is developed that uses the intersection of two straight lines as the object point. The structure of two straight lines with one intersection is easily made from a piece of paper attached to a plate, and the algorithm for extracting the lines is robust to noise.

This project proposes a new evaluation function in the line matching algorithm, in which an exponent function is built that is monotonously correspond to the distance between a sample point and the matching line in the reverse direction (Section 3.7.2). The new function is opposite to the LSM, but it is advantageous over the latter, because it gives bigger weight to the points closer to the matching line and eliminates the effects of the points further from the line. This feature increases the efficiency and
<table>
<thead>
<tr>
<th>variable name</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\gamma$</th>
<th>$T_x$</th>
<th>$T_y$</th>
<th>$T_z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>unit</td>
<td>$^\circ$</td>
<td>$^\circ$</td>
<td>$^\circ$</td>
<td>cm</td>
<td>cm</td>
<td>cm</td>
</tr>
<tr>
<td>initial value</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>final value</td>
<td>-2.01</td>
<td>1.20</td>
<td>0.52</td>
<td>5.98</td>
<td>0.07</td>
<td>0.31</td>
</tr>
</tbody>
</table>

The new function can be used in most other applications, such as stereo matching and data modulation, where the LSM may be used to date.
Chapter 4

Segment Feature Based Stereo Matching

4.1 Introduction

As discussed in Section 2.2, there are multiple groups of stereo-matching methods, which are feature-based, correlation-based, and energy-minimization-based stereo-matching methods. The first group of methods are mainly used for object recognition and localization, while other groups of methods are used for stereo image rendering. This project does not use the most complicated algorithms such as the recently developed comprehensive methods, because the system would be a feasible real-time stereo-matching system. The proposed algorithm is tightly related to its application, robotic harvesting systems, so that the algorithm should be fast, but its accuracy is enough for a robot to identify and pick off ripe tomatoes. The system should be reliable and robust for the greenhouse or farm environment.

This project uses a feature-based stereo-matching method, taking the benefit of its characteristics. Feature-based stereo matching does not provide the whole view of the stereo image in the pixel level, but it is a fast algorithm to obtain the 3D positions of objects.

The first step of stereo matching is to rectify the stereo image pair, so that it looks like being captured using an ideal BCS. One rule of the images in an ideal BCS is
that correlated image points are in the same vertical position, i.e.,

$$y_1 = y_2.$$  \hfill (4.1)

Image rectification is based on the camera calibration result in Chapter 3. Section 4.2 describes image rectification. Then more steps of feature-based stereo matching, segmentation, feature extraction, and segment correlation, are described later in this chapter.

### 4.2 Image Rectification

Theoretically, when the Epipolar geometry of a BCS is available, the BCS can be rectified into an ideal BCS by spatial transformation; however, the transformation is very complicated and it causes heavy computing load. This project proposes a simplified rectification method that can fairly approximate the ideal BCS.

Equation (3.27) is the core function of the structural data flow of the BCS calibration algorithm developed in Chapter 3, and it is also the core function of the BCS Epipolar geometry model once the BCS calibration is completed.

#### 4.2.1 Epipole Calculation

The Epipole is the projection of the origin of one camera onto the other camera’s image plane. The Epipolar line is a line in the camera image plane located through the Epipole. According to Epipolar geometry, all the 3D points that are projected onto one point in the image plain of one camera are projected to the image plane of the other camera along a specific Epipolar line. The Epipole is useful for stereo matching, because, for any image point in the image plain of one camera, the correlative image point in the other camera image plane is located along a specific Epipolar line. This restraint reduces the searching scope and simplifies the stereo-matching process. Both Epipoles can be obtained based on Epipolar geometry.

Let $$P_{e1} = \begin{pmatrix} x_{e1} \\ y_{e1} \end{pmatrix}^T$$ be the Epipole in the LI, and $$P_{e2} = \begin{pmatrix} x_{e2} \\ y_{e2} \end{pmatrix}^T$$ be the Epipole in the RI.
For the origin point of the LC,

\[ H_1 = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T. \]  

(4.2)

According to Equations (3.25) and (3.26),

\[
\begin{aligned}
H_2 &= R H_1 - T = -T \\
P_{e2} &= F_3(H_2)
\end{aligned}
\]

(4.3)

Equations (4.2) and (4.3) are used to calculate \( P_{e2} \).

The calculation of \( P_{e1} \) is the reverse of the above steps. The formulae is derived as follows. For the origin point of the RC,

\[ H_2 = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}^T. \]  

(4.4)

Then Equation (3.25) becomes

\[
F_2^{-1}\left( (H_2, W_{BCS}) \mapsto H_1 \right) \left\{ \begin{array}{l}
H_1 = R^{-1}(H_2 + T) = R^{-1}T ,
\end{array} \right.
\]

(4.5)

and Equation (3.24) becomes

\[
F_1^{-1}\left( H_1 \mapsto P_{e1} \right) \left\{ \begin{array}{l}
\begin{aligned}
u_{e1} &= f_1 \frac{X_1}{Z_1} \\
v_{e1} &= f_1 \frac{Y_1}{Z_1} \\
x_{e1} &= u_{e1} + \frac{M-1}{2} \\
y_{e1} &= -v_{e1} - \frac{N-1}{2}
\end{aligned}
\end{array} \right.
\]

(4.6)

Equations (4.5) and (4.6) are used to calculate \( P_{e1} \).

### 4.2.2 Image Rectification

The image rectification processes are different between the LI and the RI in this image rectification algorithm. Figure 4.1 is an illustration of left image rectification. The process of left image rectification is as follows.

Draw an Epipolar line through point \((x_1, y_1)\) and \(P_{e1}\), the line intersects the vertical centerline at point \(P_{i1}(x_{e1}, y_{e1})\).
Figure 4.1: An illustration of left image rectification. $P_1$: the original position of an image point; $P'_1$: the rectified position; $P_{cl}$ the center point of the left image; $P_{il}$: an immediate point.

Draw a horizontal line through point $P_{il}$.

Move the original point $P_1(x_1, y_1)$ along the vertical direction to the horizontal line to get the rectified point $P'_1(x'_1, y'_1)$.

It can be derived from Figure 4.1 that

$$
\begin{align*}
  x'_1 &= x_1 \\
  y'_1 &= y_1 + (y_{e1} - y_1) \frac{x_{e1} - x_1}{x_{e1} - x_1}, \\
\end{align*}
$$

or,

$$
\begin{align*}
  x_1 &= x'_1 \\
  y_1 &= y'_1 + (y_{e1} - y_1) \frac{x_{e1} - x_{e1}}{x_{e1} - x_{e1}}.
\end{align*}
$$

(4.7) (4.8)
Equation (4.8) is implemented as function $F_L$:

$$
F_L: \{ P'_1 \mapsto P_1 \} \begin{cases} 
    x_1 = x'_1 \\
    y_1 = y'_1 + (y_{e1} - y'_1) \frac{x_1 - x'_{e1}}{x_{e1} - x'_{e1}} 
\end{cases} \quad (4.9)
$$

Let $I_1 (P_1)$ be the intensity of the image before rectification, and $I'_1 (P'_1)$ be the intensity after rectification, then it gives

$$
I'_1 (P'_1) = I_1 (P_1) = I_1 (F_L (P'_1)) \quad (4.10)
$$

As for the RI, both $x_2$ and $y_2$ are changed to make correspondence with the LI.

The concept of plane-at-infinity, a plane located infinitely far away from the camera, needs to be used at this point.

One of the characteristics of an ideal BCS is that, any point on the plane-at-infinity has the same projected image coordinates in both cameras. Based on this rule, the rectification algorithm of the RI is constructed as follows.

Figure 4.2 is the illustration of the rectification of the RI. Similar to the LI, $P_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix}^T$ is defined as the original position of an image point in the RI, $P'_2 = \begin{pmatrix} x'_2 \\ y'_2 \end{pmatrix}^T$ is its rectified position.

![Diagram of image rectification](image)

**Figure 4.2**: An illustration of right image rectification. $P_1$, $P_2$: the original positions; $P'_1$, $P'_2$: the rectified positions.

The relationship between $P'_2$ and $P_2$ can be obtained from the relationship between $P'_2$ and $P'_1$, the relationship between $P'_1$ and $P_1$, and the relationship between $P_1$ and $P_2$. 

57
As in an ideal BCS, or a rectified BCS,

\[
\begin{align*}
  x'_2 &= x'_1 \\
  y'_2 &= y'_1
\end{align*}
\] (4.11)

This is the relationship between \( P'_2 \) and \( P'_1 \).

The relationship between \( P'_1 \) and \( P_1 \) is already established in Equation (4.8).

Equation (3.27) shows the parameterized relationship between \( P_1 \) and \( P_2 \), and it does not offer the fixed relationship between the two points because \( D \) is not fixed.

For a point on the plane-of-infinity, the distance between the point and the origin of the LC is

\[ D = \infty. \] (4.12)

If \( D \) is set by Equation (4.12), the relationship between \( P_1 \) and \( P_2 \) will be fully established. There are two options to get the approximation of the relationship in the computer program. The first option is to set to \( D \) a number that is much bigger than all the numbers being dealt with. The second option is to set \( T = 0 \) and \( Z = 1 \) in the computation program, since \( RW_1 \gg T \). In order to keep the unity of the program, this project uses the first option.

Let \( D_{\text{large}} = 1.0 \times 10^{20} \) be a large enough number, then the RL image rectification function \( F_R \) is constructed as

\[
\begin{align*}
  x'_1 &= x'_2 \\
  y'_1 &= y'_2 \\
  P_1 &= F_L (P'_1) \\
  X_{DF} &= (x_1, y_1, D_{\text{large}})^T \\
  W_{BCS} &= (\alpha, \beta, \gamma, T_1, T_2, T_3)^T \\
  P_2 &= F_{DF} (X_{DF}, W_{BCS})
\end{align*}
\] (4.13)

Let \( I_2 (P_2) \) be the intensity of the image before rectification, and \( I'_2 (P'_2) \) be the intensity after rectification, then it gives

\[ I'_2 (P'_2) = I_2 (P_2) = I_2 (F_R (P'_2)) \] (4.14)
Figure 4.4 is an example of the rectified stereo image using the algorithm described above. The result precisely approximates the image from an ideal BCS. This result approves both the BCS calibration algorithm and the BCS rectification algorithm.

### 4.3 Segmentation

The segmentation method is shown in Algorithm 3. The method is colour-based, where the colour values of two contiguous pixels are compared. If the difference of their colour values is smaller than a threshold constant, the two pixels will be judged to be in the same segment, otherwise, they will be judged to belong to different segments.

#### Algorithm 3: Image segmentation

**Data:** \( I[M, N] \) ← The intensity of a \( M \times N \) image

**Result:** \( S[] \) An array of segments

```
begin
  Assign a label \( L[i,j] \) to each pixel. The value of the label is among
  Not segmented, Inner pixel, and Edge pixel
  The values of all the labels are set to Not segmented
  for each pixel \( I[i,j] \) in the image do
    if \( L[i,j] = \text{Not segmented} \) then
      Creat a new segment and add the pixel to the new segment as the
      first pixel
      Call Algorithm 4 (Extend the segment from the pixel) using \( I[i,j] \)
      as Input
```

Two kinds of pixels are also defined in the algorithm, which are:

1. Inner pixels - the pixels whose all neighbours are in the same segment, and

2. Edge pixels - the pixels that have at least one neighbour belonging to another segment.
A label property is assigned to each pixel corresponding to the above pixel type definition. The values of the label can be Not segmented, Inner pixel, and Edge pixel. A data structure is constructed to store the segments, and a sub-algorithm (Algorithm 4) is called by Algorithm 3 to extend the segment pixel by pixel. Algorithm 4 also calls itself to extend the segment from its four neighbours.

**Algorithm 4: Extend from a pixel**

**Data:** $I[i, j] \leftarrow$ The intensity of a pixel already segmented into a segment

**Result:** $L[i, j]$ The label and the segment number of the pixel

**begin**

- if $L[i, j] \neq \text{Not segmented}$ then
  - return
- Set $L[i, j] = \text{Edge Pixel}$
- for all the four adjacent pixels of $I[i, j]$ do
  - if $L[\text{neighbour}] = \text{Not segmented}$ then
    - if $I[\text{neighbour}]$ is close to $I[i, j]$ then
      - Add the adjacent pixel to the same segment as $\text{Pixel}[i, j]$ in.
      - Call Algorithm 4 (Extend the segment from the pixel) using the adjacent pixel as Input
    - if all the four adjacent pixels are in the same segment as $I[i, j]$ in then
      - Set $L[i, j] = \text{Inner Pixel}$

**end**

Figure 4.5 is the segmentation result of Figure 4.4, and Figure 4.6 is the segmentation result of another pair of stereo images of a tomato bunch. Coloured spots represent edge pixels. The spots being the same colour are in the same segment, while the spots being different colours are in different segments, taking into account that only the segments with more than 100 pixels are shown in the figures.

When the images are segmented, the process comes to the second step, which is to build feature vectors for each segment. The feature vectors are composed of the three components of the segment mean colour and some spatial features such as top,
bottom, height, and width. This project uses images directly captured by off-the-shelf cameras containing noises, so that no individual feature is reliable for correlation, and comprehensive feature vector is the best trade off for correlation.

4.4 Segment Combination

There is a small default in this segmentation algorithm. As shown in Figure 4.6, one object such as a bunch of tomato is supposed to be in one segment because they have the similar mean colour; however, parts of the tomato bunch are segmented into different segments. This is because there is no continuous connection between the parts. This result can affect the quality of the segmentation-based stereo matching.

To solve this problem, a segment combination algorithm is built. In this algorithm, if two segments have the similar mean colour, and they are close to each other, then they are combined to make one new segment.

Figure 4.7 shows the segmentation result of combination, in which the whole bunch of tomato is combined into one segment.

4.5 Stereo Matching

Stereo matching is fulfilled by segment correlation. For each segment feature vector from the left image, its weighted least square differences from all the vectors from the right image are computed, then a 2D matrix of vector differences is formed. The segment pairs possessing the smallest least square differences are picked out as segment correlation pairs.

Figure 4.8 is the segment correlation result of Figure 4.4. There are several segments besieged with coloured spots in the figure. The spots being the same colour represent correlated segments between the two images. There are 38 correlation pairs for this instance, and all these correlation pairs are obviously true based on human-eye judgements.
4.6 Image Colour Calibration and Colour Rectification

The purpose of colour calibration is to determine the colour difference between the colours of the images in both cameras projected from the same object colour. Colour calibration is based on segment correlation, the result of the previous section. Segment correlation pairs are used to perform colour calibration.

To simplify the task without the loss of precision, assume any value of a colour component in the RI is a monotone function of the value of the similar colour component in the LI, and the functions concerning the three components are mutually independent. Let the three components of right image colour be $R_r$, $G_r$, $B_r$, and the three components of left image colour be $R_l$, $G_l$, $B_l$, then the three colour mapping functions can be defined as

$$
\begin{align*}
R_r &= F_r(R_l) \\
G_r &= F_g(G_l) \\
B_r &= F_b(B_l)
\end{align*}
$$

The colour mapping functions are further supposed to be quadratic, then

$$
\begin{align*}
R_r &= a_r R_l^2 + b_r R_l + c_r \\
G_r &= a_g G_l^2 + b_g G_l + c_g \\
B_r &= a_b B_l^2 + b_b B_l + c_b
\end{align*}
$$

Because the three components are processed independently, one equation is used to represent the three equations:

$$I_r = a I_l^2 + b I_l + c$$

For each segment correlation pair, the mean colour of both correlated segments is used as sampling data, and the optimized colour mapping functions are obtained using the LSM, as in Figure 4.9.

Colour mapping functions can be used for colour rectification. As described above, colour mapping function is the mapping from a left image colour to a right image.
colour. If the left image colour is transformed using the same function, the colour property of the left image will be the same as that of the right image. In other words, the same object will have the same colour in both images. Figure 4.10 is the colour rectification result of Figure 4.4. The colour difference between the two images in Figure 4.4 is reduced in Figure 4.10.

### 4.7 Experimental Results

Experiments are carried on for all the software packages in feature-based stereo matching. The experiments include stereo image rectification, segmentation, segment correlation, segment combination, and colour rectification.

Figure 4.4 shows one of the experimental results of stereo image rectification. The original image is in Figure 4.3 for comparison. It can be found, from the images in Figure 4.4, that both projections of any object in the LI and the RI are placed in the same vertical position, and for the objects in the background, e.g., the pictures on the wall, are even placed in the same horizontal position. This result precisely approximates the characteristics of an ideal BCS. This result verifies both the BCS calibration algorithm and the stereo image rectification algorithm.

![left image right image](image)

Figure 4.3: A sample stereo image pair before rectification in which the correlated objects are not at the same horizontal line.
Figure 4.4: *The rectified sample stereo image pair in which the correlated objects are at the same horizontal line.*

Figure 4.5 is the segmentation result of Figure 4.4, and Figure 4.6 is the segmentation result of another pair of stereo images of a tomato bunch. A piece of program code is written to automatically apply one colour to the edge pixels of each segment, so that the edge pixels of different segments have different colours. The colours are applied only to the output images, and the internal database is not affected. The segments are sorted from big sizes to small sizes. The sequence of the colours from bigger segments to smaller segments are green, yellow, blue, white, grey, red, brown, etc.. In the figures, the spots being the same colour are in the same segment, while the spots being different colours are in different segments. These figures show that, this algorithm can successfully construct segments from colour information, no matter how complex the segments are.

Figure 4.6 also shows that one tomato bunch is segmented into different segments. (It can be seen from different edge spot colours such as white, red, and black.) The separated segments are combined using the combination algorithm developed in this chapter, as shown in Figure 4.7, where one tomato bunch is segmented as one segment.

In Figure 4.7, the left image is highly different from the right image. This is because of three factors. First, on purpose of testing the adaptability of the algo-
algorithm to noisy environments, very complex and noisy background is chosen for the experiment; second, the segmentation process is applied to each image independently, so that the process creates highly different segments for the background; third, the light exposure for the same object is different between two cameras. However, the under-test object, the tomato bunch, has similar features (e.g. shape, mean color) between the two images, so that the correlation of the tomato bunch between the two images can be successfully identified.

The feature-based stereo-matching result is shown in Figure 4.8. The spots being the same colour represent correlated segments between the two images. There are 38 correlation pairs for this instance, and all the correlation pairs are obviously true if judged by human eyes.

Figure 4.9 shows the colour calibration result. In this figure, each coloured spot represents the information regarding one component of the mean colour of a pair of matched segments from both the LI and the RI. The colour of the spot represents the component, the horizontal coordinate equals the value of the component of the mean colour in the RI, while the vertical coordinate equals the value of the component of the mean colour in the LI. For example, if the G component of the mean colour of a pair of correlated segments has a value of 210 in the RI, and has a value of 140 in the
Figure 4.6: Non-complete segments before segment combination. Different colour spots on the same tomato bunch means it is segmented into different segments.

LI, there would be a green spot in the graph in the position (210, 140).

One spot in the graph represents one component of the mean colour of a correlated segment pair. Since there are 38 correlated segment pairs in this instance, there are 38 red spots, 38 green spots, and 38 blue spots. The red, green, and blue curves are colour component calibration curves obtained from the data of the spots using the LSM. They reasonably show the colour difference between the LI and the RI.

Figure 4.10 is the colour rectification result of Figure 4.4. The colour difference between the two images in Figure 4.4 is reduced in Figure 4.10.
**Figure 4.7:** Complete segments after segment combination

Complete segments after segment combination. Same colour spots on the same tomato bunch means it is segmented into the same segment.

**Figure 4.8:** Feature based stereo matching result. Objects besieged by the same colour spots in both images represent the same object indicated by correlated segments.
Figure 4.9: The colour calibration curves. $R$, $G$, $B$: colour rectification curves for red, green, and blue components; $R_l$, $G_l$, $B_l$: colour intensity values for red, green, and blue components in the left image; $R_r$, $G_r$, $B_r$: colour intensity values for red, green, and blue components in the right image.
Figure 4.10: The colour rectification result. The colour differences between the left image and the right image are reduced.

4.8 Summary

A feature-based stereo-matching program is developed in this chapter. This algorithm is the second part of the ripe-tomato recognition and localization system. The algorithm includes stereo image rectification, segmentation, segment combination, stereo matching, image colour calibration and rectification.

The first step of stereo matching is image rectification. A new simple binocular image rectification algorithm is developed in this chapter. The BCS parameters obtained from BCS calibration are used for image rectification.

The next step of the algorithm is segmentation. The Segmentation algorithm of this project is based on colour continuity. Pixels in a segment are divided into inner pixels and edge pixels. This division matches both characteristics and functions of pixels. Inner pixels are less affected by pixels from other objects, and they are more consistent and are more related to the object, so that inner pixels are used to extract the colour information and the area value. Edge pixels are a small amount of pixels that are located on the edge of a segment, which are used for geometrical statistics.
This separation makes the process faster and more reliable.

Since one object may be segmented into different parts, a segment combination algorithm is developed, which combines the separated parts of an object together. The stereo-matching algorithm developed in this chapter can find out the correlated segment pairs based on segment features.

The LSM is used to extract the colour difference function between the two images. The result is used to calibrate the colour of stereo images, making the image of the same object point be the same colour. Colour calibration is useful for colour-related stereo matching, including the stereo-matching algorithm of this project.

Experiments are carried on for two pairs of stereo images. All the steps in feature-based stereo matching and colour calibration are performed. The experimental results verify the algorithms.
Chapter 5

Recognition and Localization
Algorithms of Ripe Tomatoes

5.1 Introduction

Ripe tomato recognition and localization are the key functions of the developed system of this project. The rest parts of this project, BCS calibration, image rectification, and stereo matching serve these two functions.

The Advanced Robotics and Intelligent Systems Lab has developed an algorithm for ripe-tomato recognition (Yin et al. 2009), and it works successfully. This chapter redesigns a simplified version of the algorithm to show the structure of the developed system. The original algorithm can also be used in the same system to make complete integration.

Tomato localization is not too difficult if the stereo images fulfill the conditions of the ideal BCS; however, the tomato localization of this project is based on the calibrated BCS and rectified images, and the conventional algorithms are not suitable for this condition. A specialized algorithm for this condition is built in this chapter.
5.2 Ripe Tomato Recognition

Tomato recognition is based on colour feature. CIELAB colour space is used for tomato recognition in this project.

Similar to RGB colour space, CIELAB colour space is another way to represent colour. It is well known that, RGB values of a colour is heavily affected by the illumination factor, or, illuminance. CIELAB colour space uses different coordinates to represent colour. It describes "true" colour attributes and illuminance using separate components, so that the colour components are not affected by the illuminance. There are three components in CIELAB colour space, \( L^* \), \( a^* \), and \( b^* \). \( a^* \) and \( b^* \) are the colour components that are not affected by illuminance. They have the same range \((-50 \sim 50)\). \( a^* \) represents the colour's position between green and magenta (negative value for green and positive value for magenta), while \( b^* \) represents the colour's position between blue and yellow (negative value for blue and positive value for yellow). \( L^* \) (0 \sim 100) represents the illuminance.

The tomato recognition program of this project uses \( a^* \) and \( b^* \) as a measure to recognize the ripeness of tomatoes. Because \( a^* \) and \( b^* \) are not highly affected by illuminance, this system can work well in various light conditions. The following are the equations of colour transformation from RGB to CIELAB excerpted from Haeghen et al. (2000) and Yin et al. (2009).

5.2.1 Transformation from RGB Space to sRGB Space

Let \( I \) represent \( R \), \( G \), or \( B \) in RGB colour space, and \( I_s \) represent \( R_s \), \( G_s \), or \( B_s \) in sRGB colour space, then

\[
I_s = \begin{cases} 
(I + 0.055 \times 255)^{2.4} & \text{if } I > 0.04045 \times 255 \\
12.92 \times \frac{I}{255} & \text{if } I \leq 0.04045 \times 255
\end{cases}
\] (5.1)
5.2.2 Transformation from sRGB Space to XYZ Space

For D65 white point,

\[
\begin{pmatrix}
X_{D65} \\
Y_{D65} \\
Z_{D65}
\end{pmatrix} = 100 \times
\begin{pmatrix}
0.4124 & 0.3576 & 0.1805 \\
0.2126 & 0.7152 & 0.0722 \\
0.0193 & 0.1192 & 0.9505
\end{pmatrix}
\begin{pmatrix}
R_s \\
G_s \\
B_s
\end{pmatrix}, \quad (5.2)
\]

where \(X_{D65}, Y_{D65}, Z_{D65}\) are the colour components of a colour with D65 white point in XYZ space.

5.2.3 Transformation from XYZ Space to CIELAB Space

\[
\begin{align*}
L^* &= 116g(Y_{D65}/Y_w) - 16 \\
a^* &= 500(g(X_{D65}/X_w) - g(Y_{D65}/Y_w)), \\
b^* &= 200(g(Y_{D65}/Y_w) - g(Z_{D65}/Z_w))
\end{align*}
\]

(5.3)

where

\[
\begin{align*}
X_w &= 95.047 \\
Y_w &= 100.000 \\
Z_w &= 108.883
\end{align*}
\]

(5.4)

for D65 white point, and

\[
g(w) = \begin{cases} 
7.787w + 16/116 & \text{if } w < 0.008856 \\
\\w^{1/3} & \text{if } 0.008856 \leq w \leq 1 
\end{cases}.
\]

(5.5)

Colour transformation equations include exponent items, and need heavy computing load. In this project, not all the pixels are transformed into CIELAB colour space, only the mean colours of segment correlation pairs are transformed. The average \(a^*\) and \(b^*\) values are obtained as shown in Table 5.1 based on experimental results. When the mean colour of a segment is close to this colour value, it can be concluded that the segment is representing a ripe tomato.

It is apparent that more features, such as shapes and patterns, could be used to recognize tomatoes. More features will be implemented in the future to increase the recognition quality.
5.3 Ripe Tomato Localization

The content of this section is a reversal derivation of the equations obtained in camera calibration (Chapter 3) and image rectification (Section 4.2).

When the stereo-matching algorithm is applied to a rectified stereo image pair, segment correlation pairs are obtained. If a segment correlation pair is recognized as the images of a bunch of ripe tomatoes, the geometrical center points of the correlated segments in the segment correlation pair can be used to localize the tomatoes.

Following the definitions from Section 4.2, the geometrical center points are referred as \( P'_1 \) and \( P'_2 \), where \( P'_1 = (x'_1, y'_1)^T \) and \( P'_2 = (x'_2, y'_2)^T \).

The 3D coordinates of the object in the LI coordinate system, \( H_1 \), can derived from \( P'_1 \) and \( P'_2 \) as follows.

There are two steps in obtaining the 3D coordinates, which are computing the original image coordinates of the object point and computing the 3D coordinates of the object point from the obtained original image coordinates.

Remember that, from Section 4.2, \( P'_1 \) and \( P'_2 \) are mappings of the original image points \( (P_1 \) and \( P_2 \)) of the object point. Equations 4.9 and 4.13 can be used to obtain the original image points \( P_1 \) and \( P_2 \).

Let

\[
R = \begin{pmatrix}
  r_{11} & r_{12} & r_{13} \\
  r_{21} & r_{22} & r_{23} \\
  r_{31} & r_{32} & r_{33}
\end{pmatrix},
\]

where each \( r_{ij} \) is known because \( R \) is known when the BCS has been calibrated.
Combine Equations (3.14) and (3.17), and rewrite in a separate format as

\[
\begin{align*}
\begin{cases}
  u_1 &= x_1 - \frac{M-1}{2} \\
  v_1 &= -y_1 - \frac{N-1}{2} \\
  u_2 &= x_2 - \frac{M-1}{2} \\
  v_2 &= -y_2 - \frac{N-1}{2} \\
  \lambda_1 u_1 &= f_1 X_1 \\
  \lambda_1 v_1 &= f_1 Y_1 \\
  \lambda_1 &= f_1 Z_1 \\
  X_1 &= r_{11} X_1 + r_{12} Y_1 + r_{13} Z_1 - T_1 \\
  Y_1 &= r_{21} X_1 + r_{22} Y_1 + r_{23} Z_1 - T_2 \\
  Z_1 &= r_{31} X_1 + r_{32} Y_1 + r_{33} Z_1 - T_3 \\
  \lambda_1 u_2 &= f_2 X_1 \\
  \lambda_1 v_2 &= f_2 Y_1 \\
  \lambda_2 &= f_2 Z_2
\end{cases}
\end{align*}
\] (5.7)

noticing that there are 13 equations but 12 unknown values \((\lambda_1, X_1, Y_1, Z_1, \lambda_2, X_2, Y_2, Z_2, u_1, v_1, u_2, \text{ and } v_2)\). The ninth equation is related to \(Y_2\), which can be ignored. Performing some derivations, then get

\[
\begin{align*}
\begin{cases}
  u_1 &= x_1 - \frac{M-1}{2} \\
  v_1 &= -y_1 - \frac{N-1}{2} \\
  u_2 &= x_2 - \frac{M-1}{2} \\
  v_2 &= -y_2 - \frac{N-1}{2} \\
  Z_1 &= \frac{u_1}{r_{11} f_1 + \frac{v_1}{f_1} r_{12} + r_{13} f_1} - \frac{u_1}{r_{31} f_1 + \frac{v_1}{f_1} r_{32} + r_{33} f_1} \frac{u_2}{f_2} \\
  X_1 &= \frac{u_1}{f_1} Z_1 \\
  Y_1 &= \frac{v_1}{f_1} Z_1
\end{cases}
\end{align*}
\] (5.8)
This equation is the final equation for tomato localization.

Figure 5.1 is a screen shot of ripe-tomato localization. The 3D coordinates shown in the figure reveals the real location of the ripe-tomato bunch. More experimental results are discussed in Section 5.5.

5.4 Harvesting Robot Calibration

To this point, the program of localizing a 3D point from binocular images captured by a BCS is ready to use. Beside the program is used in stereo matching, it is also used in harvesting robot calibration. In a robotic harvesting system, it is mandatory to localize a tomato in the robot coordinate system, so that the robot can move its end-effector to the location. The relationship between the BCS coordinate system and the robot coordinate system has to be established. This process is called harvesting robot calibration.

Let the robot coordinate system be $W_3 = \begin{pmatrix} X_3 & Y_3 & Z_3 \end{pmatrix}^T$, the relationship between the BCS LC coordinate system and the robot coordinate system can be built as follows:

$$W_3 = R'W_1 - T',$$  \hspace{1cm} (5.9)

where

$$T' = \begin{pmatrix} T'_1 & T'_2 & T'_3 \end{pmatrix}^T,$$  \hspace{1cm} (5.10)
and

\[
\mathbf{R}' = \begin{pmatrix}
1 & 0 & 0 \\
0 & \cos \alpha' & \sin \alpha' \\
0 & -\sin \alpha' & \cos \alpha'
\end{pmatrix}
\begin{pmatrix}
\cos \beta' & 0 & -\sin \beta' \\
0 & 1 & 0 \\
\sin \beta' & 0 & \cos \beta'
\end{pmatrix}
\begin{pmatrix}
\cos \gamma' & \sin \gamma' & 0 \\
-\sin \gamma' & \cos \gamma' & 0 \\
0 & 0 & 1
\end{pmatrix}.
\]

(5.11)

The six unknown parameters are \( \alpha', \beta', \gamma', T_1', T_2', \) and \( T_3' \).

Move the robot end-effector to any random position \( \mathbf{W}_{3i} \), a corresponding vision system localization result of the end-effector, \( \mathbf{W}_{1i} \), can be obtained using the tomato recognition and localization program developed in this project. When multiple sets of the data pairs are obtained, an optimization method can be used to determine the six unknown parameters described above, then the robot system is calibrated, and it is ready to localize tomato in harvesting robot coordinate.

### 5.5 Experimental Results

Experiments for plenty of photos of tomatoes are tested to obtain the average CIELAB values of ripe tomatoes using the algorithm developed in this chapter. The results are in Table 5.1.

<table>
<thead>
<tr>
<th>Component name</th>
<th>( a^* )</th>
<th>( b^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average value</td>
<td>45</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 5.1: The average CIELAB values of ripe tomatoes

Several tomato recognition experiments are performed. One screen shot is shown in Figure 5.1, and some experimental results are in Table 5.2.

In Table 5.2, \( X_0, Y_0, Z_0 \) are measured 3D coordinates of a tomato, while \( X, Y, Z \) are computed 3D coordinates of the tomato using the ripe tomato recognition and localization algorithm. Each line shows one experiment. Following is the analysis of the results from Table 5.2:

1. Line 1 shows a result in which the tomato is positioned close to the centerline of the left camera. In this case, the error is less than 1.0 cm.
Table 5.2: A typical ripe-tomato localization result

<table>
<thead>
<tr>
<th></th>
<th>X₀</th>
<th>Y₀</th>
<th>Z₀</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>91.00</td>
<td>0.37</td>
<td>0.92</td>
<td>91.85</td>
</tr>
<tr>
<td>2</td>
<td>28.50</td>
<td>0.00</td>
<td>91.00</td>
<td>31.33</td>
<td>1.08</td>
<td>98.16</td>
</tr>
<tr>
<td>3</td>
<td>10.70</td>
<td>0.00</td>
<td>91.00</td>
<td>11.92</td>
<td>-0.60</td>
<td>87.23</td>
</tr>
<tr>
<td>4</td>
<td>0.00</td>
<td>-15.00</td>
<td>91.00</td>
<td>0.98</td>
<td>-14.55</td>
<td>88.90</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>-0.00</td>
<td>21.00</td>
<td>3.23</td>
<td>1.40</td>
<td>19.65</td>
</tr>
</tbody>
</table>

2. Line 2 shows a result in which an offset of 28.50 cm is added to X-axis. In this case, the error is approximately 7 cm. This error is caused by a big horizontal angle of the tomato’s position apart from the left camera’s centerline.

3. Line 3 shows another result with an offset along X-axis.

4. Line 4 shows a result in which an offset of 15.00 cm is added to Y-axis. In this case, the error is approximately 1 cm.

5. Line 5 shows a result with a very short distance between the tomato and the camera. In this case, the error is approximately 1 cm.

6. From the experimental results, the effective distance range of the binocular camera system is 21 cm ~ 91 cm. The algorithm needs to be improved if it is required to work beyond this range.

7. The vertical angle range of the effective area is 18.72° (calculated from the data in Line 3), and the horizontal angle range of the effective area is 13.40° (calculated from the data in Line 4). This angle range is limited by the specifications of the binocular camera system.

Harvesting robot calibration experiments will be performed in the system integration period, so that the experimental result is not available in this stage.
5.6 Summary

This chapter mainly develops a ripe-tomato recognition algorithm and a ripe-tomato localization algorithm. A harvesting robot calibration algorithm is also proposed as an application of the two algorithms, because the robot calibration process requires the localization of the end-effector using the algorithms developed in this project.

A simplified ripe-tomato recognition algorithm is designed with the main contents inherited from former works. CIELAB colour space is used for ripe-tomato recognition, where the effect of light intensity is eliminated, and the remaining part of the colour property is directly related to a tomato itself. This part of the colour property is used to judge the ripeness of the tomato and recognize them.

A specialized tomato localization algorithm is contrived based on the results of BCS calibration and stereo matching. The results of tomato localization precisely represent the real locations of tomatoes.
Chapter 6

Conclusion and Future Work

6.1 Conclusion

This project develops a complete software system to provide 3D coordinates of ripe tomatoes for a harvesting robot. It includes binocular camera calibration, stereo matching, tomato recognition, and tomato localization. With a pair of cameras installed on a harvesting robot, this software can automatically calibrate the camera system, recognize and localize ripe tomatoes in a natural environment, and provide 3D coordinate information to a robot to pick off the tomatoes. The engineering value of this software is apparent.

An innovative binocular camera system calibration algorithm is developed, in which a new algorithm featured with asymmetric data flow and an asymmetric energy function is proposed. This algorithm can automatically determine the spatial relationship between the two cameras in the system, without using expensive 3D coordinate measuring equipment, reducing the cost of BCS calibration. It avoids the data flow from 2D small-value variables to 3D large-value ones, making itself more consistent than most existing algorithms.

The data flow of the calibration algorithm is originally inspired by the ANN, but it uses real physical parameters as weights in an ANN. The advantage of this data flow structure over the ANN is that, when the system has converged, the physical parameters become determined. It can offer a clear view of the structure of the
system, while the ANN can not.

Extracting a point from an image is challenging, because the point is among many other noisy points. This project develops an innovative algorithm that uses the intersection of two straight lines as the object point. The structure of two straight lines with one intersection is easy to make from a piece of paper attached to a plate, and the algorithm for extracting the lines is robust to noise.

This project proposes a new evaluation function in the line matching algorithm, which uses an exponential function. The function is monotonously correspond to the distance between a sample point and the matching line in the reverse direction (section 3.7.2). This function is opposite to the current evaluation functions frequently used in the least square method.

In the least square method, the distance is directly used as the evaluation function, where the points further from the matching line have bigger values, so that they have bigger weights, while the points closer to the matching line have smaller weights. It is apparent that the points closer to the matching line are more important than the ones further from the line. The least square method treats the points opposite to the real requirement of the problem. The proposed evaluation function gives bigger weights to the points closer to the matching line and eliminates the effects of the points further from the line, making the line matching algorithm more efficient and precise. The new function can be used in many other applications, such as stereo matching and data modulation, where least square method may be used currently.

A new simple binocular image rectification algorithm is developed for stereo matching, in which the left image is rectified along the vertical direction, while the right image is rectified according to Epipolar geometry. This algorithm can create a fair approximation to an ideal binocular camera system, where a pair of corresponding points from both images are positioned on the same horizontal line. It makes stereo matching as easy as in an ideal binocular camera system.

The segmentation algorithm of this project is based on colour continuity. Pixels in a segment are divided into inner pixels and edge pixels. This division matches both the characteristics and the functions of pixels. Inner pixels are less affected by the
pixels from other objects, so they are more consistent and have more correspondence to the object, so that inner pixels are used to extract the colour information and the area value. Edge pixels are a small amount of pixels that are located on the edge of a segment, which are used for geometrical statistics. This separation makes the process faster and more reliable. Segment combination algorithm is also developed to increase the segment correlation quality.

The least square method is used to extract colour differences between the two images captured by a binocular camera system. The result is used to calibrate the image colour, making the images of the same object point be the same colour. Colour calibration is useful for colour-related stereo matching, including the stereo-matching algorithm of this project.

A simple tomato localization equation is derived from Epipolar geometry. The result of tomato localization precisely represents the actual location of tomatoes.

The program developed in this project can be installed into a robot system to construct a robotic ripe-tomato harvesting system.

6.2 Future Work

This project develops a ripe-tomato recognition and localization system. The project focuses on the system level. The programs developed in this project are ready to be used for ripe-tomato harvesting robot production development. Beside the system level construction, some modules still have some potential to be enhanced to improve the total performance of the system.

The rectification algorithm can be modified to make it more precisely to approximate the ideal binocular camera system. Right now, the rectified image is not an exact representation of an ideal binocular camera system, and the algorithm can not get the symmetrical result between the two cameras. More precise rectification algorithms could be developed in the future; however, the current algorithm is sufficient for stereo matching and ripe-tomato localization, because it fulfills the fundamental requirements of an ideal BCS and it can precisely localize tomatoes.
More features can be used for stereo matching. This project uses sizes, positions, and mean colours for stereo matching. In the future, the algorithm can be improved by using other features to increase stereo-matching quality. For example, colour pattern can be used for stereo matching. As all tomatoes have approximately spherical shape, circles can be used to match tomato edges, so that individual tomatoes can be separated from the bunches. A great amount of state of the art stereo-matching algorithms could be found in the literature, and the algorithms are continuously progressing. New and more advantageous algorithms will be developed to improve the performance.

Contour curves can also be used as feature vectors for stereo matching. The curves may not be a complete representation of an object, but a partial correlation algorithm can be developed for this situation. In a partial stereo-matching algorithm, if a part of a curve in the left image matches a part of a curve in the right image, correlation pairs can be built based on this correlation. In most cases, one object may be partially covered behind other objects. Partial correlation can find the correlation pairs in this situation.

This project has constructed the ripe-tomato recognition and localization program, and it is ready for robotic tomato harvesting system integration. There is potential for the improvement of the software and algorithms during the system integration period. It is possible that new problems will be found waiting for or requiring a new solution in this period.
References


Gotou, K., Fujiura, T., Nishiura, Y., Ikeda, H. & Dohi, M. (2003), 3-d vision system of


Jia, B., Zhu, A., Yang, S. & Mittal, G. (2009), Integrated gripper and cutter in a


Liu, T., Zhang, P. & Luo, L. (2009), Dense stereo correspondence with contrast


Pei, J., Yang, S. & Mittal, G. (2010), Vision based robot control using position specific


Sun, J., Zheng, N. N. & Shum, H. Y. (2003), ‘Stereo matching using belief prop-
agation', *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25(7), 787–800.


