A STEREOSCOPIC IMAGE ANALYSIS SYSTEM TO LOCATE
AND CHARACTERIZE NURSERY PLANTS.

A Thesis

Presented in Partial Fulfillment of the requirements for
the degree Master of Science in the
Graduate School of the Ohio State University

by

Curtis W. Alden, B.S.

* * * * *

The Ohio State University

1989

Master's Examination Committee:

Robert G. Holmes, PH.D.
James C. Papritan, PH.D.
Kim L. Boyer, PH.D.

Approved By

R. J. Holmes
Advisor
Department Of Agricultural Engineering
A stereoscopic image processing system was developed to locate and evaluate nursery plants. The system was tested and evaluated using narrow upright and cone shaped plants. The information retrieved was a control point, representing the location and the height of the plant, and either the radius or apex angle of the plant. The system detected the edges of a plant in each image, then represented the edges as linear equations using the Hough transform. The control point and secondary feature in each image was determined from the linear equations representing the edges. The data from each image was combined to form the desired three dimensional information.
ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to the following:

Dr. Robert G. Holmes for his patience and guidance during my graduate studies.

Dr. James Papritan and Dr. Kim Boyer for their helpful suggestions and for serving as members on my graduate committee.

Dr. Michael Lichtenstieger, Mark Morgan and Don Wertz for their helpful suggestions and support.

OSU and USDA for providing the financial support.

Carolyn, Sandy, Sue, Val and the rest of the staff and faculty for all of their help.

Uncle Erick, Aunt Nancy and family for treating me as one of their own.

And my Family; Diane, Elizabeth, Mom, Dad, Jeff, Lori, Noel, Emily, and Chanda, without whose support, encouragement, and help I would not of achieved this degree.
VITA

October 8, 1984 . . . . . . . . . . . . . . . . . . . . . . Born - Columbus, Ohio

1986 . . . . . . . . . . . . . . . . . . . . . . . . B.S. Agricultural Engineering
The Ohio State University

1986 - 1987 . . . . . . . . . . . . . . . . . Graduate Research and Teaching
Associate, Agricultural
Engineering Department,
The Ohio State University

FIELDS OF STUDY

Major Field: Agricultural Engineering

Studies in Mechanical Engineering. Professor J. A. Collins and Associate
Professor K. Srinivasan.

Studies in Electrical Engineering. Professor K. Boyer
# TABLE OF CONTENTS

Acknowledgements ....................................................................... ii

Vita ......................................................................................... iii

Table of Figures ......................................................................... vii

List of Tables ............................................................................. x

**CHAPTERS:**

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. OBJECTIVES</td>
<td>4</td>
</tr>
<tr>
<td>III. LITERATURE REVIEW</td>
<td>5</td>
</tr>
<tr>
<td>3.2 Introduction to Image Processing</td>
<td>5</td>
</tr>
<tr>
<td>3.2 Processing Methods</td>
<td>6</td>
</tr>
<tr>
<td>3.2.1 Generalized Image</td>
<td>7</td>
</tr>
<tr>
<td>3.2.1.1 Texture Analysis</td>
<td>7</td>
</tr>
<tr>
<td>3.2.2.2 Edge Detection</td>
<td>9</td>
</tr>
<tr>
<td>3.2.2 Segmented Image</td>
<td>11</td>
</tr>
<tr>
<td>3.2.2.1 Thresholding</td>
<td>11</td>
</tr>
<tr>
<td>3.2.2.2 Boundary Detection</td>
<td>12</td>
</tr>
<tr>
<td>3.2.3 Geometric Structure</td>
<td>16</td>
</tr>
<tr>
<td>3.2.3.1 Boundary Description</td>
<td>16</td>
</tr>
<tr>
<td>3.2.3.2 Object Description</td>
<td>17</td>
</tr>
<tr>
<td>3.2.4 Relational Structures</td>
<td>17</td>
</tr>
<tr>
<td>3.3 Multiple Images</td>
<td>18</td>
</tr>
<tr>
<td>3.4 Summary of Review</td>
<td>21</td>
</tr>
</tbody>
</table>

| IV. PLANT CHARACTERISTICS | 22 |
| 4.1 Shearing Characteristics | 22 |
| 4.2 Cone Shaped Plants | 22 |

iv
<table>
<thead>
<tr>
<th>CHAPTERS</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.2 Narrow Upright Shaped Plants</td>
<td>24</td>
</tr>
<tr>
<td>4.4 Information Required From An Image</td>
<td>26</td>
</tr>
<tr>
<td>V. DETERMINING SHAPE</td>
<td>27</td>
</tr>
<tr>
<td>5.1 Image Processing</td>
<td>27</td>
</tr>
<tr>
<td>5.2 Digital Grey Level Images</td>
<td>28</td>
</tr>
<tr>
<td>5.3 Edge Transform</td>
<td>29</td>
</tr>
<tr>
<td>5.4 Hough Transform</td>
<td>30</td>
</tr>
<tr>
<td>5.5 Two Dimensional Characteristics</td>
<td>33</td>
</tr>
<tr>
<td>5.6 Three Dimensional Characteristics</td>
<td>35</td>
</tr>
<tr>
<td>VI. EQUIPMENT AND TEST DESCRIPTION</td>
<td>38</td>
</tr>
<tr>
<td>6.1 Test Overview</td>
<td>38</td>
</tr>
<tr>
<td>6.2 Experimental Layout and Equipment</td>
<td>38</td>
</tr>
<tr>
<td>6.2.1 Equipment and Programs</td>
<td>38</td>
</tr>
<tr>
<td>6.2.2 Testing Area</td>
<td>40</td>
</tr>
<tr>
<td>6.2.3 Camera Placement and Global Coordinate System</td>
<td>43</td>
</tr>
<tr>
<td>6.3 Test Procedure</td>
<td>46</td>
</tr>
<tr>
<td>6.3.1 First Stage - Two Dimensional</td>
<td>46</td>
</tr>
<tr>
<td>6.3.2 Second Stage - Three Dimensional</td>
<td>48</td>
</tr>
<tr>
<td>6.3.2.1 First part - Object</td>
<td>48</td>
</tr>
<tr>
<td>6.3.2.2 Second Part - Live Plants</td>
<td>50</td>
</tr>
<tr>
<td>VII. RESULTS AND DISCUSSION</td>
<td>52</td>
</tr>
<tr>
<td>7.1 Two Dimensional Object</td>
<td>52</td>
</tr>
<tr>
<td>7.2 Second Stage</td>
<td>57</td>
</tr>
<tr>
<td>7.2.1 Constructed Object Location and Description Determination</td>
<td>57</td>
</tr>
<tr>
<td>7.2.2 Second Part - Live plants</td>
<td>63</td>
</tr>
<tr>
<td>7.3 Further Discussion</td>
<td>78</td>
</tr>
<tr>
<td>VIII. SUMMARY AND CONCLUSIONS</td>
<td>82</td>
</tr>
<tr>
<td>IX. RECOMMENDATIONS FOR FURTHER STUDY</td>
<td>90</td>
</tr>
<tr>
<td>LIST OF REFERENCES</td>
<td>88</td>
</tr>
</tbody>
</table>
Appendices

A. Listing of Pascal Program ........................................ 91

B. Rotational Matrix Calculation ................................. 107
# Table of Figures

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Edge Operation</td>
<td>9</td>
</tr>
<tr>
<td>2.</td>
<td>Hough Transform of Points on an Image (A) to Lines in the Parameter Space M-B (B)</td>
<td>14</td>
</tr>
<tr>
<td>3.</td>
<td>Camera Reference Frame Oriented to the World Reference</td>
<td>18</td>
</tr>
<tr>
<td>4.</td>
<td>The Relationship Between the Multiplier v and The Three Dimensional Point Location</td>
<td>19</td>
</tr>
<tr>
<td>5.</td>
<td>Predominant Plant Shapes: a &gt; Cone and b &gt; Narrow Upright</td>
<td>22</td>
</tr>
<tr>
<td>6.</td>
<td>The Cone Shaped Plant Characteristics and the Information Needed to Represent the Shape.</td>
<td>23</td>
</tr>
<tr>
<td>7.</td>
<td>The Information Required to Represent the Narrow Upright Shaped Plant</td>
<td>24</td>
</tr>
<tr>
<td>8.</td>
<td>Process Flow for Determining the Crossing Point And the Secondary Feature of a Plant in an Image.</td>
<td>27</td>
</tr>
<tr>
<td>9.</td>
<td>Digital Image</td>
<td>28</td>
</tr>
<tr>
<td>10.</td>
<td>The Edge Transform of an Object in an Image</td>
<td>29</td>
</tr>
<tr>
<td>11.</td>
<td>The Hough Transform of a Single Point</td>
<td>30</td>
</tr>
<tr>
<td>12.</td>
<td>The Image Flow from the (a) Original Image to (b) the Edge Image, then (c) the Placement of the Edge Lines.</td>
<td>32</td>
</tr>
<tr>
<td>FIGURE</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>13.</td>
<td>Testing Area Schematic</td>
<td>40</td>
</tr>
<tr>
<td>14.</td>
<td>Placement of Plumbobs to Locate Camera Position</td>
<td>43</td>
</tr>
<tr>
<td>15.</td>
<td>The World Coordinate System With Respect to the Equipment Used</td>
<td>45</td>
</tr>
<tr>
<td>16.</td>
<td>The Triangle (a) and Rectangle (b) Used in the First Stage of Tests</td>
<td>46</td>
</tr>
<tr>
<td>17.</td>
<td>Constructed Cone and Cylinder Used in the Second Stage</td>
<td>48</td>
</tr>
<tr>
<td>18.</td>
<td>Caster Board Placement for the Five Runs in a Set.</td>
<td>51</td>
</tr>
<tr>
<td>19.</td>
<td>Images From Horizontal Camera Angle Tests Using the Triangle</td>
<td>53</td>
</tr>
<tr>
<td>20.</td>
<td>Images from Horizontal Camera Angle Test Using the Rectangle.</td>
<td>54</td>
</tr>
<tr>
<td>21.</td>
<td>Triangle Object Test with Branches</td>
<td>56</td>
</tr>
<tr>
<td>22.</td>
<td>Graph of the Object test results</td>
<td>59</td>
</tr>
<tr>
<td>23.</td>
<td>Procedure Flow for Conical Shaped Plants</td>
<td>64</td>
</tr>
<tr>
<td>24.</td>
<td>Procedure Flow for Narrow Upright Shaped Plants</td>
<td>65</td>
</tr>
<tr>
<td>25.</td>
<td>Cone Object Test on Caster Board Results</td>
<td>67</td>
</tr>
<tr>
<td>26.</td>
<td>Graphs For Narrow Upright Plant Test</td>
<td>71</td>
</tr>
<tr>
<td>27.</td>
<td>Line Placement for the First Set of the Narrow Upright Plant Test</td>
<td>72</td>
</tr>
<tr>
<td>28.</td>
<td>Graphs For Cone Plant Test</td>
<td>76</td>
</tr>
<tr>
<td>29.</td>
<td>Line Placement for the First Set of the Cone Plant Test</td>
<td>77</td>
</tr>
</tbody>
</table>
FIGURE: The Effect of Lighting on the Placement of Edge Lines ........................................... 81

31. Rotations About the Reference System \{A\} to Form System \{B\}. ........................................... 106
List of Tables

<table>
<thead>
<tr>
<th>TABLE</th>
<th>Title</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Area of the Parameter Space Searched</td>
<td>32</td>
</tr>
<tr>
<td>2.</td>
<td>Comparison of the Caster Board Locations</td>
<td>51</td>
</tr>
<tr>
<td>3.</td>
<td>Determined Axis Position for Two Dimensional Objects</td>
<td>55</td>
</tr>
<tr>
<td>4.</td>
<td>Object Test Results</td>
<td>58</td>
</tr>
<tr>
<td>5.</td>
<td>Cone Object Test Results</td>
<td>68</td>
</tr>
<tr>
<td>6.</td>
<td>Narrow Upright Plant Shape Results</td>
<td>70</td>
</tr>
<tr>
<td>7.</td>
<td>Cone Shaped Plant Test Results</td>
<td>75</td>
</tr>
<tr>
<td>8.</td>
<td>Sensitivity Test to Camera Angle</td>
<td>79</td>
</tr>
</tbody>
</table>
Chapter I

INTRODUCTION

In 1985, the nursery industry had over 1.7 billion dollars in sales. Twenty three percent of this amount was from the sales of conifers (a type of narrow leaf evergreen). Conifer production requires tremendous seasonal maintenance. A major task required during production is the shaping of the plant. In addition to conifers there are several types of evergreen shrubs and bushes which are used in hedges, landscaping, and as Christmas trees. All evergreens need shaping on a yearly basis. These plants require shearing to promote growth and to establish a desirable shape.

Generally, the shearing can only be performed when the new growth of the plant has just started. The time of year for shearing varies from species to species. The shearing operation requires individual assessment of each plant and is normally performed manually. The task is labor intensive and typically requires several seconds to a minute or more for each plant. Often considered a boring, seasonal job, careful attention is required to maintain plant quality. Due to the seasonal nature of shearing, a temporary work force must be hired and trained each season.
The drudgery, low pay, and temporary aspect makes it hard to attract a reliable work force. Thus, labor's contribution to the cost of shearing plants mandates that the shearing operation be further mechanized.

Prominent among the factors hindering the mechanization of the shearing operation is the natural variability of individual plants within a species. This variability dictates that each plant must be individually assessed to determine its size and shape. This task is relatively easy for a properly trained person, but difficult for the untrained and even more difficult to automate.

A way to automate the shearing operation would be to mimic the workers actions. As a worker proceeds through the field he must first assess each plant as to its location, size, and shape. After assessing the plant, the worker shears the plant to the desired shape. In order to automate the task, a system capable of determining the location, the size, and the shape of each plant is required.

Image analysis has been used in several agricultural applications to locate and/or assess various items. Examples of image analysis systems include grading tomatoes (Sarker, 1984), measuring plant architecture (Davison, 1985), locating fruit (Sites, 1988), locating and assessing asparagus (Baylou, 1984) and locating crop rows for tractor guidance systems (Reid, 1985). These systems all assess particular items so
that a related task such as picking, sorting, or guidance can be automated. For systems used to locate an object, a two image (stereoscopic) system was used to find the three dimensional location. These systems have helped automate tasks that had previously been performed manually.

Image analysis technology appears to have the potential for providing the control information necessary to automate a shearing device. The requirement of finding the three dimensional location of the plants infers that a stereoscopic system is required. The purpose of this thesis was to develop a two camera (stereoscopic) image analysis system which could provide the control information for a plant shearing device.
Chapter II

OBJECTIVES

The overall objective of this research was to develop and test a stereoscopic digital image processing system capable of providing the information necessary to control a nursery plant shearing device. Specific objectives were to:

1> Develop the processing system to determine the location and attributes of a nursery plant using two digital images obtained simultaneously.

2> Devise the optimum placement of the two cameras relative to the nursery plant being analyzed.

3> Test and validate the accuracy and repeatability of the experimental procedure for conical and narrow upright shaped plants.
Chapter III

LITERATURE REVIEW

3.2 Introduction to Image Processing

Image processing has many applications ranging from satellite image analysis to directing robots for specific tasks. As Rosenfield (1984) stated, "Its applications include document processing (character recognition, etc.), microscopy, radiology, industrial automation (inspection, robot vision), remote sensing, navigation, and reconnaissance, to name only the major areas." Using image processing methods, the applications enhance or mimic the ability of the human inspectors to evaluate scenes or images. For example Kimme et al, (1975) developed a system to inspect chest x-rays to find tumors. The location of possible tumors were highlighted in the x-ray to encourage the radiologist to look closer at those locations. Image processing systems have also provided control information for automating tasks. An example is the robotic fruit harvester developed by Kawamura and Kondo, (1984) which locates the citrus fruit and relates the location to the robot arm.
Thus, image processing is the procedures and methods which transform the data from visual sensors into useful information. The information extracted can be tabulated, used for control, and/or reinspected. Many tools are available to extract the desired information from an image. Rosenfield discussed several of the tools that are presently being used to interpret images in his 1984 paper. In his concluding remarks, he stressed that "there is no general theory of control in image analysis." In other words, there are no set guidelines for the extraction of information from an image. For one task there may be many different methods which would provide satisfactory results. For each task, the key is to investigate the different methods and determine the most applicable system.

3.2 Processing Methods

The purpose of image processing is to determine key characteristics from a scene or an image. There are several levels in processing images. Ballard and Brown(1982) described four levels of abstraction: 1) generalized images, 2) segmented images, 3) geometric structures and 4) relational structures. The transformation from one classification to another is done to highlight, quantify, and/or extract the important information. Thus the image can be transformed directly from level one to level two, three, or four.
3.2.1 Generalized Image

The generalized image represents the first order of abstraction and includes the basic image and image like structures. Initial processing of an image is included in this level. The raw image is usually portrayed as a two dimensional grey level array. The image is bulky and confused. Much of the information in each piece (pixel) of the array is the same as the next one (Ballard and Brown, 1982). Each part of the image can represent a great deal of information. The image is often processed at this level to remove unwanted information and highlight the desired data. Two of the more important general image processing methods are texture analysis and edge detection.

3.2.1.1 Texture Analysis

One of the oldest methods of image processing is texture analysis. Texture analysis is used to find relational patterns between pixels in an image. The goal of texture analysis is to relate pixel patterns in the image to a standard pattern. Texture analysis has been used in satellite imagery to classify terrain. Wezka, (et al, 1976), explained three standard methods of terrain classification and tested them on 180 LANDSAT (Land Satellite) images. The first method discussed by Wezka was the Fourier power spectrum calculated from the complex conjugate of the Fourier
transform of the image. The second method explained was the second-order grey level statistics which measured the occurrence of grey-level values. The final method was based on first order grey-level differences. Wezka concluded that the Fourier power spectrum was not as effective as the other two methods because it did not classify the terrains as accurately. Both the second order grey level statistics method and the first order grey level difference method had over a 90 percent accuracy level in identifying objects from their texture.

Texture analysis also has been used to evaluate an image for certain textural occurrences. The textural occurrences can be used to identify areas in images. Wigger (et al, 1988) used texture analysis methods to identify fungal damage on soybeans. This method used three image arrays of grey levels for the colors of red, green, and blue. The texture of the image was calculated as the ratio of the three arrays at each point in the image. Wigger’s system was able to determine fungal infection at a 98% accuracy rate. Shearer (1987) designed a system to identify species of plants using a texture analysis method. Grey level difference statistics were established for the three image arrays (red, blue, and yellow) and patterns were noted for particular species of plants. This system of texture analysis was approximately 90% accurate in determining plant species.
The goal of texture analysis is to distinguish different objects from their textures. Wigger (et al, 1988) and Shearer (1987) were both successful in using this method to identify objects from their texture.

3.2.2.2 Edge Detection

"The boundaries of objects often provide important features in pattern recognition/classification applications" said Ashkar and Modestino (1978). The first step in detecting boundaries is edge processing which identifies the pixels between objects and/or backgrounds in images. These edge points represent the boundary of an object. The primary methods of identifying edges are gradient operators, Laplacian operators and edge templates. Levine (1985) described the basic types of operators. These operators are represented by masks which contain multipliers that calculate the digital gradient. Usually there will be one mask for each direction, horizontal and vertical. The value for each pixel in the new image is calculated from a nonlinear operator using the images formed from the horizontal and vertical masks. In the example (Figure 1a) the nonlinear operator is the maximum absolute value of the two images formed. Figure 1a shows the original image, the vertical and horizontal masks, the resultant images \((S_h, S_v)\), and the final image formed. The edge pixels in the original image have greater value in the final image than the other pixels. Four basic masks are shown in Figure 1b. The
<table>
<thead>
<tr>
<th>Image</th>
<th>Processed By</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Image I(i,j) --&gt;</td>
<td>vertical --&gt; forms $S_v(i,j)$ mask</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 1 1</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 1 1</td>
<td></td>
</tr>
<tr>
<td>0 0 0 1 1 1</td>
<td></td>
</tr>
<tr>
<td>0 0 1 1 1 1</td>
<td></td>
</tr>
<tr>
<td>0 0 1 1 1 1</td>
<td></td>
</tr>
<tr>
<td>0 0 1 1 1 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Original Image I(i,j) --&gt;</td>
<td>Horizontal --&gt; forms $S_h(i,j)$ mask</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 1</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 0 1</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 1 1</td>
<td></td>
</tr>
<tr>
<td>0 0 0 0 1 1</td>
<td></td>
</tr>
<tr>
<td>0 0 1 1 1 1</td>
<td></td>
</tr>
<tr>
<td>0 0 1 1 1 1</td>
<td></td>
</tr>
</tbody>
</table>

The final image is formed by taking the maximum absolute value of either $S_h(i,j)$ or $S_v(i,j)$ for each point $(i,j)$ in the final image.

$$E(i,j) = \max(|S_v(i,j)|, |S_h(i,j)|)$$

$x$ - represent points that cannot be calculated because of being border pixel.

a) Mask Operation on Image

NonLinear Operation is $\max(|S_v(i,j)|, |S_h(i,j)|)$

<table>
<thead>
<tr>
<th>Prewitt Operator</th>
<th>Laplacian Operator</th>
<th>Sobel Operator</th>
<th>Roberts Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0 -1</td>
<td>0 1 0</td>
<td>1 0 -1</td>
<td>0 -1</td>
</tr>
<tr>
<td>1 0 -1</td>
<td>1 -4 1</td>
<td>2 0 -2</td>
<td>1 0</td>
</tr>
<tr>
<td>1 0 -1</td>
<td>0 1 0</td>
<td>1 0 -1</td>
<td></td>
</tr>
</tbody>
</table>

b) Four Common Masks

**Figure 1**

Edge Operation
corresponding horizontal masks are the same but rotated 90 degrees, except for the Laplacian which has only one mask.

Several systems already developed use edge transform as the first step in analysis (Wolfe and Sandler, 1985) (Wolfe and Swaminath, 1986). These systems used the edge transform to identify the edges of an object in an image so that the object could be quantified with further processing.

3.2.2 Segmented Image

Segmenting is the breaking of an image into groups with common properties. The purpose of segmenting is to make possible analysis of specific regions of the image. The two predominant methods of segmenting are thresholding and boundary detection.

3.2.2.1 Thresholding

The first method of segmenting is to threshold the image. This method breaks the image into two parts, one meeting the threshold and one not. Thresholding allows the computer to work with the image in parts. An example of this is the system developed to detect prune defects (Delwiche et al, 1988). This system actually
performed three separate thresholds on the image. The first threshold separated the prune from its background making it possible to determine the area characteristics of the prune. The second threshold determined which rows of prune pixels contained defects. The final threshold separated the pixels in defective areas from the pixels in the good areas of the prune. The group of pixels in the defective area of the prune was measured and then the prune was classified. The system performed with high accuracy (0 - 2.4% error of classification) and speed (20 prunes/sec).

There are many other systems which use thresholding successfully to segment images so that objects in the image may be analyzed (Guyer et al, 1986) (Rehkuglar and Throop, 1985) (Reid et al, 1985) (Hines et al, 1986). These systems used thresholding to eliminate unwanted areas of the image. The systems were then able to perform further analysis on specific areas of the image.

3.2.2.2 Boundary Detection

"Boundaries in objects are extremely important for human vision; in fact, often an object can be recognized from only an outline" said Ballard and Brown (1982). The desire is to segment the image so that the boundary of an object can be processed to obtain information. The boundary detection usually takes place after
the image has had initial processing to enhance the edges, generally in the form of an edge transform. The simplest method to determine the edge after the edge transform, is to threshold the new image. This method, though, may produce errors due to shadows and other non desired edges in the image. The determined set of edges may not be complete and/or may have non-edges included. Methods have been developed to fit the possible edge with parametric equations or to use graph search techniques to determine the true edge pixels.

The Hough technique is used to find the general trend of the edge pixel locations as they fit a parametric curve (i.e line, or conical section). The method was developed to identify straight line and curves in images (Duda and Hart, 1972). The method is now used in many different applications. The technique involves a image transformation into a parameter space based on the equation to be fit. The parameter space is defined by the curve desired. For example a line would require a two dimensional space, slope and intercept. A circle would require a three dimensional space, radius and (x,y) center location. The parameter space would be a multi-dimensional array accumulator with the same dimension as the space. The array would keep track of the number of points from the image transformed into each cell of the accumulator. This process would then produce local maximums in the parameter space at specific points. The points would correspond to a parametric curve in the image containing the most edge pixels. An example would
be the transformation to determine the equations of a line in an image. The parameter space, in the example, is a two dimensional space - intercept versus slope. Figure 2 (a) is the original image and (b) is the parameter space. In the image (a) there are several pixels which fit a line with the equation $Y = mx + b$, $m$ (slope) and $b$ (intercept) are unknown. In the example the parameter space is not an accumulator but a graph. The local maximum is found where the lines intersect. A local maximum in the parameter space occurs at the point (0.5,0.5). The equation of the line in the image is then determined to be $Y = 0.5x + 0.5$.

Figure 2
Hough Transform of Points on a Image (A) to Lines in the Parameter Space M-B (B)
The Hough transform technique has been used successfully in the detection of tumors in human chest cavities (Kimme et al, 1975), detection of fruit (Whittiker et al, 1984) and analysis of luminous cones from welding (Zavidovique et al, 1985).

Another method of identifying the edges of an object is called the graph search. This method decides the likelihood of each cell being an edge and upon finding an edge searches to find the next closest edge pixel. This method may involve extensive heuristic searches to determine the edge of an object. One method as described by Ballard and Brown (1982) is as follows.

1. Determine the first edge pixel.
2. Evaluate connected pixels for "edgeness" potential.
3. Follow the best potential edge and repeat 2.
4. In the case of running into a dead end go to the prior pixel and use the second best potential edge.
5. Repeat until goal has been met or no other possibilities exist.

For this to be successful, the ability to test pixels for "edgeness" must be available and a final goal be established. This type of edge classifier could encircle objects or establish edge streaks which could be analyzed separately from the image.
3.2.3 Geometric Structure

The ability to represent objects in images by their geometric structure allows the computer to match, describe, and/or analyze the scene. An image can be processed to find either two- or three-dimensional structures. The forming of these geometric structures can take place from raw or preprocessed images.

3.2.3.1 Boundary Description.

Two dimensional structures usually are obtained from the edges of objects in the images. There are many different methods describing the geometric structure of images from their boundaries. The problem is that the edges are just a set of points on an image. Polylines, chain codes, ¥-s curves, conic sections, and B-splines are a few methods of compiling the edge points of an object into a more descriptive form. In the more compact form the objects are easier to compare and manipulate.

Fourier descriptors have been used to form the signature of objects. The signature can then be compared to a standard signature to identify objects in an image. This method has been used to reduce the search effort for identifying landmarks (Krueger et al. 1972). The use of the Fourier descriptors allows the computer to
compare the descriptors of certain shapes to an unknown shape. The unknown shape can be matched with the known shapes. This method was used in recognizing hand printed characters (Granlund, 1972). Granlund's system scanned hand written characters (letters A - Z) of several people and matched them with a standard set of characters at a 98% accuracy level.

3.2.3.2 Object Description

Area, moment, center of mass, and other statistics are used in describing the object in an image. These characteristics are used to evaluate the corresponding object's attributes. Several system that evaluate object attributes have been designed. The majority of these systems use a geometric description to measure the object against a standard. Wright (1985) used a system to describe sweet potatoes for a sorting or grading operation. This system determined the volume of a sweet potato from the area calculated from several images.

3.2.4 Relational Structure

The relational structure, which deals with the interaction of objects in an image, is the highest level of abstraction. However, since this research involved only images containing a single object the relational structure level was not considered
and will not be discussed any further.

3.3 Multiple Images

If an image can be obtained from more than one aspect, three dimensional information can be extracted. By identifying one common object in the images, it is possible to triangulate on that object and find its three dimensional position. The position orientation of each camera is also required to be able to triangulate on the objects from the images. Tsia (1986) evaluated several methods of determining a camera's orientation. The number of unknowns in the different methods varied from 9 to 17 and was dependent on the number of assumptions made. Toscania (1986) presented a basic description of the camera orientation and its translation. Figure 3 shows the camera model. The purpose of the calculations was to translate a point from the camera system \((O_rX_rY_rZ_r)\) to the real world system \((O_wX_wY_wZ_w)\). To translate from the camera system to the world system the formula:

\[
\begin{bmatrix}
X_w \\
Y_w \\
Z_w
\end{bmatrix} = [R] \begin{bmatrix}
X_r \\
Y_r \\
Z_r
\end{bmatrix} + [T] \tag{1}
\]

where \(\begin{bmatrix}
X_r \\
Y_r \\
Z_r
\end{bmatrix}\) is the position in the camera system
Figure 3
Camera Reference Frame Oriented to
the World Reference

\[
\begin{bmatrix}
X_w \\
Y_w \\
Z_w
\end{bmatrix}
\]
is the position in the world system

\[ [R] \]
is the rotational matrix from camera to world

and

\[ [T] \]
is the translational matrix

The camera coordinate \((X_c, Y_c, Z_c)\) is calculated from the two dimensional point on
the image \((r, c)\) by the equation

\[
\begin{bmatrix}
X_i \\
Y_i \\
Z_i
\end{bmatrix} = \begin{bmatrix}
  r * v \\
  c * v \\
  f * (v-1)
\end{bmatrix}
\] (2)

Where \((r, c)\) is the point on the image
\(f\) is the focal length of the lens and
\(v\) is a multiplier for the distance.
Figure 4
The Relation Between the Multiplier \( v \) and the Three Dimensional Point Location.

Figure 4 shows the relationship between the camera coordinate and the image point. The value of the multiplier \( v \) depends on the distance the point is located from the image.

Sites (1988) developed a system to locate fruit using triangulation from two images. The system was capable of taking images from a single camera at different angles and then guiding the picking mechanism to the three dimensional location of the fruit. Baylou (1988) developed a system to locate and assess asparagus from two different camera angles. These systems used the two dimensional information gathered from each image and produced the three dimensional information required.
3.4 Summary of Review

Several image processing methods have proven successful. Representing the outline of an object in an image with equations has proven successful in locating the object. Edge detection followed by boundary representation has the potential of defining a plant in an image. Once the plant is defined within the image it could be combined with another image to produce the three dimensional information required for shearing.
Chapter IV

PLANT CHARACTERISTICS

4.1 Shearing Characteristics

Plants are unique and must be assessed individually to find the characteristics which determine the shearing procedure. Narrowleaf evergreens are the more common nursery plants that require shearing. Evergreens are classified into three plant types by Caldwell (1973), the spreading type (e.g., some Yews, Pfitzer Juniper); Upright type (e.g., Hicks Yew, Caneart Juniper, Pine, Spruce); and Rounded Type (e.g., Brown Yew, Globe Arborvitae). Due to the separate algorithms required to define the different shape types, this study was limited to the upright type plants. The upright type evergreen could be either narrow upright or conical in shape as shown in figure 5. The characteristics required are described in the following sections.

4.2 Cone Shaped Plants

Examples of cone shaped plants are the Douglas Fir, Brown Taxus and White Pine. These plants tend to grow symmetrically about a vertical center axis. The axis is
located at the point where the sides of the plant cross (figure 5a). A worker shearing a plant follows the sides to remove any growth not in the desired shape. The symmetry of the plant is maintained or promoted by the shearing. To shear the plant the location of the center axis, the angle of the sides of the plant, and the point where the sides cross the center axis must be determined. For the shearing operation the center axis was assumed to be vertical. A perfect plant was represented as a symmetrical cone, with the sides of the cone defined by the edge lines. Thus, the primary characteristic for the cone shaped plant was the location of the center axis of the plant. The secondary characteristic was the apex angle and the location of the point of side crossing (figure 6).
4.2 The Narrow Upright Shape

Several species of taxus such as the Hicks Yew are sheared to the narrow upright shape. The shape (as in figure 5b) is a cylinder with a slight angle to the sides. The angle of the sides of these plants depends on the size and type of plant. The actual shearing will reproduce the desired angle of the sides and a horizontal cut across the top. The shape of the plant was approximated to be a cylinder. After locating the plant the angle of the sides of the plant can be sheared to a desired angle. The major characteristics of the cylinder were the location of the center axis,
the average diameter, and the height (figure 7). The center axis was determined from the sides of the plant. The axis was located equidistant between the sides. The radius was the distance between the side of the plant and the center axis. The height of the plant was measured as the highest point on the plant that was on the center axis. The information required for shearing the narrow upright plant was a control point on the top of the plant (located on the center axis) and the radius of the plant.
4.4 Information Required From Image

The characteristics of a plant can be determined from the edges of that plant. The information required from each image is the location of the plant's edge(s). The location of the edge determines the position of the plant axis, the size of the plant, and the shape of the plant.
Chapter V

DETERMINING SHAPE

5.1 Image Processing

The goal of the image processing system was to determine the sides of the plant and to calculate the required features from the information obtained. An experimental system was designed to determine the sides of the plant in each image. After the sides are found the plant axis, side angle, and height are determined in each image. Finally the data from the two images are combined to determine the

Picture Grey-level Image

>> Edge Transform <<

Edge Grey-level Image

>> Hough Transform <<

Edge Line Equations

Angle and Crossing Point of the lines

Figure 8

27
three dimensional information. The system follows the flow shown in figure 8 and described in the following sections.

5.2 Digital Grey Level Images

An image was obtained in a digital form by using photodiode sensors arranged in the camera in a square two dimensional array. The image is represented by the function $I(r,c)$ where $r = 0,1,.., N_r$; $c = 0,1,.., N_c$; and the value $I = 0,1,..,(N_g-1)$ in an image with a resolution of $N_r$ by $N_c$ and $N_g$ grey-levels. The gray-level represents the light intensity at the point $(r,c)$ in the image $I(r,c)$. An example can be shown in figure 9 which is a digital image with a resolution of $5 \times 5$ pixels and 2 grey levels. The term picture element or pixel is defined as an individual cell of a digital image.

![Digital Image Values](image1)

![Digital Image](image2)

**Figure 9**
Digital Image
5.3 Edge Transform

To determine the position of the edges of the plant, a digital transform was performed on the image. The edge transform found local discontinuities in the image, which indicated an edge, by taking a digital gradient. The Sobel operator, a $3 \times 3$ neighborhood gradient, was used to perform the edge transform. The gradient used the following masks

$$
\begin{bmatrix}
1 & 0 & -1 \\
2 & 0 & -2 \\
1 & 0 & -1
\end{bmatrix} 
\quad
\begin{bmatrix}
1 & 2 & 1 \\
0 & 0 & 0 \\
-1 & -2 & -1
\end{bmatrix}
$$

Vertical    Horizontal

These masks produced $S_v$ and $S_h$ (defined in section 3.2.2.2) values for each point $(r,c)$ in the image $I(r,c)$. The value in the edge image $E(r,c)$ equals the nonlinear operation

$$
E(r,c) = (S_v(r,c)^2 + S_h(r,c)^2)^{1/2}
$$

The values of $E(r,c)$ refer to the likelihood of that pixel being an edge. The higher the value $E(r,c)$ the greater the possibility of that point being on the edge of the object. The major consideration with this transform is that the edges around the object on the image, will be two pixels thick. The new image $E(r,c)$ has less resolution than the original image $I(r,c)$. The resolution of the new image $E(r,c)$ is $(N_r-2)$ by $(N_c-2)$ if the original image $I(r,c)$ had the resolution of $N_r$ by $N_c$. Figure 10 shows a digital image of an object(a) and the corresponding edge image of the same object (b).
5.4 Hough Transform

The edge transformed image \( E(r,c) \) was processed to form linear equations which represented the plant edges. The Hough transform computed the linear equations. The edge lines were formed from the edge pixels in the image \( E(r,c) \). The Hough transform accomplished this task by transforming points from the \( x\)-\( y \) plane to the \( m\)-\( b \) plane (the line parameter space). The point \((x,y)\) was transformed to the line \( B = y - Mx \) in the \( m\)-\( b \) plane. An example is shown in figure 11 where the point \((1,1)\) is translated into the line \( b = 1 - m \) in the \( M\)-\( B \) plane. When a
group of points in the x-y plane that lie on the same line are translated into the m-b plane a group of lines are formed. These lines will intersect at the point (m,b) which represents the slope and y-intercept of the line which contains the group of edge points. The example in figure 2 contains points which lie on the line $y = 0.5x + 0.5$, the translated points form the lines in the M-B plane which intersect at the point (0.5, 0.5). Using this method, images with partial or erratic points with unknown linear parameters can be translated into the m-b plane. The point in the m-b plane which has the most intersections is the m and b for that line.
To find the intersection a two dimensional histogram was set up. The histogram is an area of the m-b plane which has been digitized into cells. When a point in the x-y plane is translated into the m-b plane those cells which lie on the translated line are incremented by one. After all of the points in the x-y plane are translated the cell with the highest value represents the m and b of the line in the x-y plane. A modified version of this was used in determining the lines that represent the edges of the plant. Two histograms were set up, one in the positive m space and one in the negative m space. This allowed the sides to be found separately. The cells were incremented by the grey-level value of the pixel in the edge image to produce a weighted histogram.

The slope and intercept for the edge line was found by taking a weighted average of the top four cells in the histogram. To limit the area which was searched in the m-b plane a general idea of the area where each of the lines would be located was determined (table 1). This area was then searched for maximum intersections. Limiting the area searched reduced computational time speeding up the determination of m (the slope) and b (the intercept) for the edge lines. The process from the initial image to the placement of the lines is shown in figure 12.
Table 1

Area of the Parameter Space Searched

<table>
<thead>
<tr>
<th>Line</th>
<th>Slope (M)</th>
<th>Intercept (B)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>Cone Shape</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Left Line</td>
<td>-0.5</td>
<td>-2.5</td>
</tr>
<tr>
<td>Right Line</td>
<td>0.5</td>
<td>2.5</td>
</tr>
<tr>
<td>Narrow Upright Shape</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Both Lines</td>
<td>0.5</td>
<td>-0.5</td>
</tr>
</tbody>
</table>

Figure 12

The Image Flow from the (a) Original Image to (b) the Edge Image, then (c) the Placement of the Edge Lines.

5.5 Two Dimensional Characteristics

The two dimensional characteristics of the plant were found from the lines determined from the Hough transform. The two shape types require different methods of calculating the characteristics. In the case of the cone shape the center axis was calculated to be the vertical line through the point where the side lines intersected. The equation to determine the point (r,c) (r=row c=column) in the image is:
\[ r = (b_2 - b_1)/(m_1-m_2) \]  

(4)

and \[ c = m_1 r + b_1, \]  

(5)

where the equations for the two lines are,

line 1: \[ c = m_1 r + b_1, \]  

(6)

and line 2: \[ c = m_2 r + b_2 \]  

(7)

This produces a set of coordinates \((r,c)\) which is the location of the point of intersection in the image plane. The angle that the sides make with the vertical was then calculated by the equation

\[
\text{Angle} = \left( |\text{Arctan}(m_1)| + |\text{Arctan}(m_2)| \right)/2
\]  

(8)

where \(m\) is the calculated slope of the line.

The center axis for the modified cylindrical shaped plant was defined as the line located equidistant between the detected edge lines. The edges were represented by the equations

\[ C_1 = b_1 \]  

(9)

\[ C_2 = b_2 \]  

(10)
The equation for the line which represents the center axis was defined as

\[ C_z = (b_1 + b_2)/2 \]  

(11)

The image was scanned along the center axis to determine the location of the control point (the highest point in the image on the center axis). This point is then stored to represent the center axis line. The radius was calculated to be:

\[ \text{Radius} = |(b_1 - b_2)/2| \]  

(12)

It is important to note that the radius was in units of pixels and not in real world units.

5.6 Three Dimensional Characteristics

The information from each image consisted of a common point on the center axis and the secondary feature, the radius or the angle, of the plant. The information was combined and translated into the world coordinate system from the camera coordinate system. A point in a camera coordinate system can be translated into the real world coordinate system (see figure 3) using the following equations:

\[ [X] = [C] [R] + [T] \]  

(13)

\[ [X] = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \]  

(14)
\[ [C] = \begin{bmatrix} v^*r \\ v^*c \\ (1-v)^*f \end{bmatrix} \]  
\[ [R] = \begin{bmatrix} a1 & a2 & a3 \\ a4 & a5 & a6 \\ a7 & a8 & a9 \end{bmatrix} \]  
\[ [T] = \begin{bmatrix} U \\ V \\ W \end{bmatrix} \]  

Where \([X]\) is the location in the world coordinate system,

\([C]\) is the location in the camera coordinate system,

\([R]\) is the rotational matrix for the camera (the method of calculating the matrix \([R]\) is described in appendix B), and \([T]\) is the translational matrix for the camera with the \(U\), \(V\), and \(W\) values corresponding to the \(x\), \(y\), \(z\) location of the camera in the world coordinate system. Also \((r,c)\) is the location of the point in the image, and \(f\) is the focal length of the lens. The value of \(v\) is a factor which relates the position of a point on the camera image plane to the corresponding position of the point in space. Figure 4 relates the value of \(v\) to the point on the image and the point in space.

In the equations; \([T]\), \([R]\), \(r\), \(c\), and \(f\) are all known, \(([T], [R], and f are determined prior to the investigation. The values \(r\) and \(c\) represent the control point found in the image). The three dimensional point \([X]\) and the multiplier \(v\) are unknown.
The calculation of \([R]\) is shown in appendix B. The method used had 3 equations and 4 unknowns per camera with 3 unknowns common to both cameras. To solve for \(v1\) and \(v2\) (the multipliers for the left or right cameras respectively) the following equation is solved:

\[
[C_i][R_i] + [T_i] = [C_i][R_2] + [T_2]
\]  

where

\[
[C_i] = \begin{bmatrix} v_i * r_i \\ v_i * c_i \\ (1 - v_i) * f_i \\ \end{bmatrix} \quad i = 1, 2
\]  

Solving for \(v1\) and \(v2\) the vector \([X]\) was determined using Equation 13. The results were compared between the left camera and the right camera. The angle in the case of the cone shaped plants did not need to be altered because it was in degrees common to both coordinate systems. The radius was translated using the equations:

\[
Sx = a_3 \times \text{radius} \times v
\]  

\[
Sy = a_6 \times \text{radius} \times v
\]  

then

\[
\text{RADIUS} = \sqrt{(Sx^2 + Sy^2)}
\]  

where \(a3\) and \(a6\) are from the rotational matrix \([R]\) and the \(v\) was the factor appropriate for that camera. This process calculated the control point and the secondary characteristic required to control a shearing machine.
Chapter VI

EQUIPMENT AND TEST DESCRIPTION

6.1 Test Overview

The experimental system was tested in two stages. The first stage tested the ability of the algorithms to determine the axis and the sides of standard two dimensional triangles and rectangles. The second stage tested the ability of the algorithms to translate the information from the two dimensional images to the three dimensional information.

6.2 Experimental Layout and Equipment

6.2.1 Equipment and Programs

The experimental system described in chapter V used two Reticon 520 cameras to obtain the images of the plants. The Reticon 520 camera is a CCD camera which used a discrete 100 x 100 matrix of photosensors to obtain the digital image of the scene. The cameras were controlled by a system developed by the Electrical
Engineering Department at The Ohio State University (Thomas, 1987). The controller translated the signal from the cameras to a digital image with 64 grey levels. The controller was the device which assigned the grey-level values. The controller displayed the images on two Panasonic monitors. The controller sent and received the images through an HPIB interface as a ten thousand 8-bit character string to the host computer.

The host computer was a Hewlett Packard 310 computer with 1 megabyte of ram, a 3.5 inch floppy disk drive, and a 20MB hard drive. The image processing algorithms were written in Pascal. Each part of the processing was a separate procedure (algorithms are presented in Appendix A). The procedures for the communication along the HPIB interface were from the HP procedure library. The procedures SEND and RECEIVE were written to send and retrieve images from the controller. The image was stored as a 10000 character string in the computer to reduce the memory requirement.

The edge transform was done in the procedure TRANSFORM2. The resulting values from the transform were limited to the values between 0 and 63. Any values greater than 63 were reassigned the value of 63. This limitation made it possible to display the edge image through the camera controller.
The Hough transform was done in procedure DETERMINE SHAPE. The procedure received the image matrix and the desired shape type. The output from the procedure depended on the shape type. The point where the side lines crossed in the image and the average vertical angle of the sides was determined for the cone shaped plant. In the case of the narrow upright shape the location of the point at the top of the plant on the center axis and the radius of the plant was determined. An axillary function of the DETERMINE SHAPE procedure was the placing of the side lines and the center axis onto the image. This function allowed a visual inspection of the placement of the lines.

The translation and triangulation algorithm was the procedure COMBINE. The procedure received the common point from each image and using the formulas presented in section 5.5, calculated the three dimensional location of the point. The procedure also calculated the secondary feature, either radius or vertical angle. The rotational and translational matrixes for the cameras used in this part were calculated by the program presented in appendix B using SPEAKEAZY on the OSU IBM 3081 mainframe computer.

6.2.2 Testing Area

The testing area was designed to simulate outdoor conditions in the laboratory. A schematic of the area can be seen in figure 13. The ‘ground’ of the area was
a raised platform with the dimensions 1.2 x 2.4 meters at the height of .5 meters. The platform divided allowing the trunk of the plant to be placed through a hole located in the center. The raised platform concealed the container in which the plant was placed. The platform was covered by a black cloth to increase the contrast. A black curtain was also placed behind the platform for the same reason. The platform was only used in the second stage of testing where the plants were being used.

The lighting for the test area consisted of four GE 2000K, 500 watt light bulbs in ten inch reflectors. The lights were placed as shown in figure 13, with two lights in the front at a height of 2.8m and two lights in the back at 1.5m height. A Gosen Luna-Pro light meter was used to measure light intensity. The scale of the meter was logarithmic with approximate foot-candles comparison. The light intensity was approximately 14 (130 fc) around the plant compared to a light intensity of 22 (3200 fc) measured on a bright day outside. The uniformity of the light intensity on and around the plant varied from 13.5 to 14.5 on the light meter and was deemed adequate.
6.2.3 Camera Placement and Global Coordinate System

The exact location and orientation of each camera was needed to calculate the distance between each camera and the plant. The camera's position and orientation was calculated as follows:

Step 1. The camera was oriented so that a whole plant was encompassed in the image plane.

Step 2. Plumbobs were placed on each side of the cameras so that the string intersected with the y-axis of the image plane. (see figure 14a)

Step 3. The points where the plumbobs intersected the floor were marked. The midpoint between these two points was determined.

Step 4. The coordinate of the point on the floor and the distance from the floor to a line through the center of the image plane was calculated. This produced \((x_0, y_0, z_0)\) for the camera (see figure 14b).

Step 5. A target was used to find a point in space which was seen at the center of the image plane. The center of the target represented that point.

Step 6. The location of the target was determined by dropping a plumbob through the target's center and finding its intersection with the floor.

Step 7. The point of the intersection determined the coordinates of the target center. This produced \((x_t, y_t, z_t)\) for the target. The line from the center of the image plane to the target represented the z axis of the camera system. The xy plane was the image plane.

Step 8. The location and orientation of the camera was calculated from the following formulas:

\[
\text{Camera location} = (x_c, y_c, z_c)
\]

\[
\text{Z-Axis Angle} = \arctan((y_t-y_i)/(x_t-x_i))
\]

\[
\text{Y-Axis Angle} = \arctan((z_t-z_i)/xy)
\]
Figure 14
Placement of plumbobs to locate camera position
(A) the plumbobs cross the center of image plane and
(B) the location with respect to the world axis system.

Where
\[(x_o, y_o, z_o)\] was determined in Step 4
\[(x_i, y_i, z_i)\] was determined in Step 7
and \[xy = \sqrt{y_i^2 + (x_i - x_o)^2}\]

Following these steps, the angles needed to determine the matrix \(\{R\}\) (see appendix B) used in the COMBINE procedure were determined. This process was followed for each of the cameras producing both relationship matrix's.
The first stage of testing did not require a global coordinate system. The camera was located approximately 3.5 meters from the shapes.

A cartesian coordinate system was used in the second stage of testing. Using the right hand rule the floor was used as the x-y plane with positive z measured in the upward direction. The approximate positioning of the equipment relative to the coordinate system is shown in figure 15.

During the second stage of testing the cameras were also located as shown in figure 15 and were oriented so that the center of the plant was approximately located at the center of the camera image. The exact position of each camera was calculated and will be presented in chapter 7.

![Figure 15](image)

**Figure 15**
The World Coordinate System with Respect to the Equipment Used.
6.3 Test Procedure

The testing procedures were outlined as follows:

I. First Stage - Two Dimensional Investigation

A. Determination of Optimum Camera Angle.
B. Determination of System's Ability to Place Edge lines on Irregular Shapes.

II. Second Stage - Three Dimensional Investigation

A. Determination of System's Ability to Locate and Assess Ideal Constructed Shapes.
B. Determination of System's Ability to Locate and Assess Live Plants.

6.3.1 First Stage - Two Dimensional

The first stage of the test was to determine the ability of the system to determine and quantify the side lines of the known two dimensional shapes. In this part of the test, two shapes, a triangle and a rectangle (figure 16), were processed by the algorithm to determine the side lines and the vertical axis. In the test the procedures RECEIVE, TRANSFORM2, and DETERMINE SHAPE were used. During this test only the front lights were used.

Part A of the first stage consisted of taking images of the known shapes with known orientations and then processing them for the location of the side lines and axis
lines. The results were compared visually with the correct placement of the lines. The test was run with horizontal camera angles of approximately 0, 10, 20, and 30 degrees. The figure was approximately 4 meters from the cameras. The camera angle represented the angle between the Z-axis of the camera system and a horizontal plane. The accuracy was then evaluated based on the quantitative results of placement of the axis, determination of secondary characteristic, and the qualitative result of placement of the side lines. The first stage of testing determined the placement for the second stage of testing.
Part B of the test determined the ability of the system to place the edge lines correctly on irregular edged shapes. The triangle shape was used in this part. The system was run three times. The first time a single branch was placed on the edge of the triangle. The second time several branches were placed on a single side of the shape. The final time several branches were on both sides of the shape.

6.3.2 Second Stage - Three Dimensional

The second stage tested the system's ability to determine three dimensional location of the center axis. Part A used fabricated shapes which approximated a perfectly symmetrical plant. Part B used one plant of each shape to verify the ability of the developed system to identify the key characteristics of the plants.

6.3.2.1 Part A - Object Detection

A constructed cone and cylinder (shown in figure 17) were used in part A of the second stage. The location of the center axis and the secondary characteristic were found for these objects. The location of the axis was found by locating the position of a plumbob at the center of the object, the tip of the cone or the center top of the cylinder. The center of the cylinder was found geometrically by placing two lines perpendicular to the sides of the top circle and marking their intersection.
The plumbob position was marked on the floor and then measured with respect to the coordinate system. The secondary characteristics were measured and noted for each of the objects being tested. The radius was found for the cylinder by determining the distance from the center of the cylinder to the side of the object.

The cameras were placed at the location as stated in chapter VII. The lighting for this test used the four lights described. The test platform was used and the object shape was placed on the platform during the test.
The test for accuracy was made by placing each object in ten different places and running the system. The location determined by the algorithm was then compared to the correct location determined by the plumbob. The secondary features were also compared.

6.3.2.2 Part B - Live Plant Detection

The tests with the live plants were performed in three sets of five runs per plant. Also three sets of runs were with the cone shaped object. These sets were used to compare the average errors and variations. The plants were in containers and were placed on a caster board during the test. The caster board allowed the plant to be rotated 90 degrees between runs while maintaining its position relative to the caster board. The runs were compared by calculating the axis positions determined relative to the caster board. The caster board was placed at the origin of the global system with the board’s sides along the x and y axis during the test. The board was then rotated clockwise 90 degrees so that the next corner was at the origin, allowing a comparison between the results of the five runs. Figure 18 shows the positioning of the caster board and Table 2 shows the corresponding comparison values \((X', Y')\).

The results from the runs were compared statistically, using the values \(X', Y', Z'\) determined in table 2, to find the mean and the variance in the placement of the axis for all of the sets of 5 runs. A mean and variance was also found for the
secondary results of angle and radius for each plant individually. An overall variance was calculated for all of the plants in all of the characteristics: placement, height (for cone shape), and radius or angle. For analysis the xy-error was used to determine the success rate, an xy-error of less then 10cm was considered a success, greater then 10cm a failure.

Table 2
Comparison of the Caster Board Locations

<table>
<thead>
<tr>
<th>Run No.</th>
<th>Corner Location</th>
<th>Translated Equivalents</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(0,0,0)</td>
<td>X</td>
</tr>
<tr>
<td>1</td>
<td>(0,0,0)</td>
<td>Y</td>
</tr>
<tr>
<td>2</td>
<td>(50.8,0,0)</td>
<td>50.8 - X</td>
</tr>
<tr>
<td>3</td>
<td>(50.8,50.8,0)</td>
<td>50.8 - X</td>
</tr>
<tr>
<td>4</td>
<td>(50.8,0,0)</td>
<td>50.8 - Y</td>
</tr>
<tr>
<td>5</td>
<td>(0,0,0)</td>
<td>X</td>
</tr>
</tbody>
</table>

NOTE - X,Y,Z is the axis position determined from the image system; X',Y',Z' is the position relative to the caster board.

Figure 13
Caster Board Placement for the Five Runs in a Set.
Chapter VII

RESULTS AND DISCUSSION

7.1 Two Dimensional Test

The ability of the experimental system to detect the edges of a constructed rectangle and triangle was evaluated. The procedure PICTURE_PLOT (described in Appendix a) was used to print images from the camera system. Figures 19 and 20 shows the images from the test described in section 6.3.1. Figure 19 a,b,c,d are the images using the triangle and figure 20 a,b,c,d using the rectangle. The four camera angles utilized were 3.13 degrees (a), 12.9 degrees (b), 19.0 degrees (c), and 28.3 degrees (d). These angles were used because after approximating the angles 0, 10, 20, and 30 these were the true angles determined. The placement of the lines in the images were observed to be on the edge of the object for all of the camera angles. In the case of the triangle object the center axis passed through the triangles apex.

Table 3 presents the quantitative results for the test. The determination of the actual axis location in the image was done manually. It was observed that the
Figure 19: Images from Horizontal Camera Angle Test Using a Triangle.
Figure 20
Images from Horizontal Camera Angle Test Using a Rectangle.
Table 3
Determined Axis Position for Two Dimensional Objects

<table>
<thead>
<tr>
<th>Figure Letter</th>
<th>Camera Shape</th>
<th>Angle</th>
<th>Axis Row Actual Determined</th>
<th>Apex Angle of Triangle</th>
</tr>
</thead>
<tbody>
<tr>
<td>19a</td>
<td>Triangle</td>
<td>3.13</td>
<td>47.5</td>
<td>47.8</td>
</tr>
<tr>
<td>19b</td>
<td>Triangle</td>
<td>12.9</td>
<td>54.5</td>
<td>54.5</td>
</tr>
<tr>
<td>19c</td>
<td>Triangle</td>
<td>19.0</td>
<td>47.5</td>
<td>47.9</td>
</tr>
<tr>
<td>19d</td>
<td>Triangle</td>
<td>28.3</td>
<td>58.5</td>
<td>58.7</td>
</tr>
<tr>
<td>20a</td>
<td>Rectangle</td>
<td>3.13</td>
<td>47.0</td>
<td>46.7</td>
</tr>
<tr>
<td>20b</td>
<td>Rectangle</td>
<td>12.9</td>
<td>55.0</td>
<td>54.7</td>
</tr>
<tr>
<td>20c</td>
<td>Rectangle</td>
<td>19.0</td>
<td>47.0</td>
<td>46.6</td>
</tr>
<tr>
<td>20d</td>
<td>Rectangle</td>
<td>28.3</td>
<td>58.5</td>
<td>58.5</td>
</tr>
</tbody>
</table>

Notes
- Units are in Pixels
- The camera angle is measured from Camera Z-Axis to the World System X-Y plane

location could be specified within +/- .5 pixels due to the resolution of the image.
The actual apex angle of the triangle was also measured manually. The apex angle of the triangle was 30 degrees. The precision of the measurement was observed to be +/- 1 degree. No significant difference between camera angles was observed.

The second part of the test was to introduce branches on the triangle to see the effect in determining the axis. This part was qualitative resulting in the figures 21a,b, and c. Figure 21a has one branch on the object. Figure 21b has several branches on one side of the object. Figure 21c has several branches on both sides
Figure 21
Placement of Edge Lines on Triangle Shape With Outlying Branches.
of the object. The placement of the center axis in the outlines was observed not to be effected by the branches. The edge lines were seen to be placed on the edges of the object regardless of the number of branches on the sides.

7.2 Second Stage

The second stage was to evaluate the ability of the system to determine the three dimensional information for constructed objects and live plants. The full algorithm (procedures RECEIVE, TRANSFORM2, DETERMINE_SHAPE, & COMBINE) was evaluated, and PICTURE_PLOT was used to print images. The testing was broken down into two major parts; one utilizing objects of constructed shape, cone and cylinder; and one using real plants, narrow upright and conical.

7.2.1 Constructed Object Location and Description Determination

The procedure outlined in section 6.3.2.1 utilized the shapes pictured in figure 17 to produce the following results. Table 4 shows the results for part A and figure 22 a-d show the corresponding graphs to that set of data. The results show a large deviation in the x coordinate having an average error of 3.33 cm and a standard deviation of 2.40cm. The y and z coordinates had average errors of -0.09cm and -0.73cm and standard deviations of 1.14cm and 0.49cm respectively. In the graph
### Table 4
Object Test Results
Axis Position Determined By Image System
Units are in Centimeters

<table>
<thead>
<tr>
<th>DETERMINED POSITION</th>
<th>ACTUAL POSITION</th>
<th>SECOND CHAR.</th>
<th>TIME</th>
<th>X ERR</th>
<th>Y ERR</th>
<th>Z ERR</th>
<th>TOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CONE</strong></td>
<td></td>
<td></td>
<td>Angle</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>43.18</td>
<td>17.02</td>
<td>118.11</td>
<td>44.20</td>
<td>17.60</td>
<td>118.24</td>
<td>15.78</td>
</tr>
<tr>
<td>2</td>
<td>36.20</td>
<td>14.92</td>
<td>118.11</td>
<td>38.25</td>
<td>15.77</td>
<td>117.22</td>
<td>15.88</td>
</tr>
<tr>
<td>3</td>
<td>27.31</td>
<td>20.32</td>
<td>118.11</td>
<td>30.89</td>
<td>20.52</td>
<td>116.94</td>
<td>16.30</td>
</tr>
<tr>
<td>4</td>
<td>18.29</td>
<td>13.34</td>
<td>118.11</td>
<td>21.77</td>
<td>14.00</td>
<td>118.03</td>
<td>15.56</td>
</tr>
<tr>
<td>5</td>
<td>20.64</td>
<td>20.19</td>
<td>118.11</td>
<td>24.28</td>
<td>20.17</td>
<td>116.87</td>
<td>16.04</td>
</tr>
<tr>
<td>6</td>
<td>31.43</td>
<td>14.12</td>
<td>118.11</td>
<td>32.89</td>
<td>14.45</td>
<td>117.20</td>
<td>16.06</td>
</tr>
<tr>
<td>7</td>
<td>44.45</td>
<td>19.05</td>
<td>118.11</td>
<td>45.14</td>
<td>18.92</td>
<td>116.51</td>
<td>16.14</td>
</tr>
<tr>
<td>8</td>
<td>36.20</td>
<td>31.37</td>
<td>118.11</td>
<td>38.94</td>
<td>30.40</td>
<td>117.35</td>
<td>16.15</td>
</tr>
<tr>
<td>9</td>
<td>19.49</td>
<td>32.07</td>
<td>118.75</td>
<td>26.19</td>
<td>31.98</td>
<td>118.34</td>
<td>15.66</td>
</tr>
<tr>
<td>10</td>
<td>6.03</td>
<td>26.04</td>
<td>118.75</td>
<td>12.42</td>
<td>25.48</td>
<td>118.11</td>
<td>15.62</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CYLINDER</th>
<th></th>
<th></th>
<th>Radius</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CONE</strong></td>
<td></td>
<td></td>
<td>Angle</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>9.86</td>
<td>13.97</td>
<td>111.13</td>
<td>13.82</td>
<td>14.96</td>
<td>110.01</td>
<td>10.15</td>
</tr>
<tr>
<td>2</td>
<td>9.40</td>
<td>27.94</td>
<td>111.13</td>
<td>16.71</td>
<td>27.58</td>
<td>109.70</td>
<td>10.46</td>
</tr>
<tr>
<td>3</td>
<td>25.10</td>
<td>22.86</td>
<td>110.82</td>
<td>28.32</td>
<td>22.94</td>
<td>110.39</td>
<td>10.14</td>
</tr>
<tr>
<td>4</td>
<td>32.08</td>
<td>8.59</td>
<td>110.82</td>
<td>33.60</td>
<td>10.13</td>
<td>109.86</td>
<td>10.07</td>
</tr>
<tr>
<td>5</td>
<td>60.33</td>
<td>13.97</td>
<td>110.49</td>
<td>58.83</td>
<td>15.24</td>
<td>110.03</td>
<td>10.25</td>
</tr>
<tr>
<td>6</td>
<td>56.64</td>
<td>33.99</td>
<td>111.13</td>
<td>58.27</td>
<td>34.80</td>
<td>110.64</td>
<td>10.47</td>
</tr>
<tr>
<td>7</td>
<td>51.31</td>
<td>46.99</td>
<td>110.82</td>
<td>53.44</td>
<td>45.03</td>
<td>111.00</td>
<td>10.62</td>
</tr>
<tr>
<td>8</td>
<td>27.94</td>
<td>46.05</td>
<td>110.82</td>
<td>33.55</td>
<td>43.51</td>
<td>110.36</td>
<td>10.45</td>
</tr>
<tr>
<td>9</td>
<td>6.35</td>
<td>40.64</td>
<td>111.13</td>
<td>13.97</td>
<td>38.20</td>
<td>110.03</td>
<td>10.49</td>
</tr>
</tbody>
</table>

**Average Error** | **3.33** | **-0.09** | **-0.73** | **3.83**
**STD of Error**   | **2.40** | **1.14**  | **0.49**  | **2.06**

*Error Is the Actual axis location minus the determined Position.*
(a) Object Tests
Determined Axis Positions Using Image System
Units Are in Centimeters

○ = Cone Object
△ = Cylinder Object
Filled Symbols are the Actual Position

(b) Object Tests
Total Error Vs. Run Number
Units are in Centimeters

○ = Cone Object
△ = Cylinder Object

Figure 22
Graph of the Object test results
Figure 22
Graph of the Object test results
of axis location (figure 22a), the distance from the actual axis location to the determined axis is observed to be proportional to the x position. Generally, the farther away the axis is from the origin the less the error. A plot of the total error vs the test number shows no relation between the two (figure 22b). A plot of the xy error (figure 22c) is observed to have most of the errors in positive x-error region. The large X-error may be the result of the x-direction being almost perpendicular to the image plane, thus represented by the z axis of the camera. The x-direction then is more dependent on the focal length of the camera. The focal length was proportionally greater then the r and c values in the image. Thus, when calculating the point in the camera space using equation 18 a small error in the multiplier v will cause a larger error in the z-coordinate than with the x or z coordinate in the camera reference space. Then when using equation 13 to calculate the point in the world reference space the larger error is transferred into the x-coordinate. Thus a larger error in the x position was observed when the cameras position and orientation were not exactly determined. A discussion of the effect of an error in camera placement is presented in section 7.3.

The errors were believed to be mainly due from miscalculation of the cameras' orientation and location. Figure 22d shows a graph of the error vs the distance of the axis from the right camera. There is a definite linear relation producing an R value of 0.876. This is further evidence that the error is due to an incorrect
camera location. Overall the XY error for all runs was within a 10cm error radius. This error radius is the standard by which all of the results were judged. Ten centimeters was selected because of the resolution of the system, discussed in section 7.3, and the shearing requirements.

The (X, Y, Z) position of the cameras for this test were (310.8cm, -66.3cm, 194.6cm) for the left and (313.6cm, 165.7cm, 194.3cm) for the right. The rotational matrices for the test were the following:

\[
\text{left camera } [R] = \begin{bmatrix} -0.27582 & -0.26269 & -0.92462 \\ -0.92462 & 0.075379 & 0.26532 \\ 0.0 & 0.96193 & 0.27329 \end{bmatrix}
\]

\[
\text{Right Camera } [R] = \begin{bmatrix} 0.46941 & -0.22078 & -0.85493 \\ -0.88298 & -0.11737 & -0.45450 \\ 0.0 & 0.96823 & -0.25004 \end{bmatrix}
\]

The COMBINE procedure was changed after this part of the test. In the first version of COMBINE, to find \(v_1\) and \(v_2\) equation 18 was solved by solving all three resulting equations (X, Y, Z) simultaneously. The final version of COMBINE found \(v_1\) and \(v_2\) by only solving the X and Y equations. After determining \(v_1\) and \(v_2\) the point (x,y,z) was determined by equation 13 in both versions of COMBINE. The final version appears to determine the point (x,y,z) more accurately than the first version. The first version had larger errors in the x position and a large variance.
The final version was used there after in the research because the accuracy in determining the x-y position of the axis was better.

7.2.2 Part B - Live Plants

Figure 23 is the flow for the conical plant and figure 24 is the flow for the narrow upright shape. In the figures (a) is the view of the plant, (b) is the digital image of the plant, (c) is the edge image of the plant, and (d) is the placement of the edge lines determined by the Hough Transform.

The three dimensional test used the caster board method described in section 6.2.2.2. Using the caster board to rotate the object, or plant, the plot of determined axis locations ideally produced a square centered about the point (25.4cm, 25.4cm), the center of rotation. The square would be produced since a point rotated 90 degrees around a center point forms a square. The (X, Y, Z) position of the cameras for this set of test was (297.5cm, -62.5cm, 184.1cm) for the left and (293.7cm, 208.9cm, 184.2cm) for the right. The rotational matrices were calculated to be the following:

\[
\text{left camera } [R] = \begin{bmatrix}
-0.29979 & -0.21542 & -0.92937 \\
-0.95401 & 0.067693 & 0.29205 \\
0.0 & 0.97417 & -0.22580
\end{bmatrix}
\]
Figure 23
Procedure Flow for the Conical Shaped Plant
Figure 24
Procedure Flow of the Narrow Upright Shaped Plant
Right Camera $[R] = \begin{bmatrix} 0.56130 & -0.16609 & -0.81077 \\ -0.82761 & -0.11265 & -0.54988 \\ 0.0 & 0.97966 & -0.20068 \end{bmatrix}$

The actual results using a constructed cone on the caster board for the first set of runs was a square centered about a position (25.53cm, 25.92cm), as shown in figure 25a. The results from the two other sets of runs utilizing the cone produce squares centered about the points (26.02cm, 26.00cm) and (25.88cm, 25.81cm). After calculating the average position for each set of runs, using the method described in section 6.3.2.2, the standard error determined for the three sets was 0.56cm, 0.52cm, and 0.98cm for the x, y, and z directions. These average errors and deviations are significantly less than in the previous test. This is due to the change in the COMBINE procedure and the recalculation of the camera position.

The average position of the axis was determined by computing $(X',Y')$, from table 2, for each run and finding the average $(X',Y')$ for each set. The average position $(X',Y')$ was rotated 90 degrees 4 times about the point (25.4,25.4) (the center of the caster board). The average position, and the standard errors calculated for the location of the cones are shown in table 5. The errors were defined as the average position of the axis minus the axis location determined by the image system. figure 25b shows the xy error plotted. This plot shows that the error was well within a 10cm radius. When the average position was calculated and plotted for
(a) Cone Object Test
Plot of Axis Position Determined by the Image System

- O---O - First Set
- △---△ - Second Set
- □---□ - Third Set
Filled Symbols Represent Average Positions

(b) Cone Object Test
Plot of Determined Axis Position Error
Error is the Determined Position Subtracted from the Average Position

 Figure 25
Cone Object Test on Caster Board Results
<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>Apex Angle</th>
<th>X</th>
<th>Y</th>
<th>X-Err</th>
<th>Y-Err</th>
<th>Z-Err</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Set 1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>29.24</td>
<td>17.27</td>
<td>139.29</td>
<td>16.14</td>
<td>29.83</td>
<td>17.16</td>
<td>0.59</td>
<td>-0.11</td>
<td>0.78</td>
</tr>
<tr>
<td>2</td>
<td>33.45</td>
<td>29.87</td>
<td>137.62</td>
<td>16.89</td>
<td>33.64</td>
<td>29.83</td>
<td>0.19</td>
<td>-0.04</td>
<td>-0.90</td>
</tr>
<tr>
<td>3</td>
<td>20.70</td>
<td>34.44</td>
<td>140.00</td>
<td>16.25</td>
<td>20.97</td>
<td>33.64</td>
<td>0.27</td>
<td>-0.80</td>
<td>1.49</td>
</tr>
<tr>
<td>4</td>
<td>17.15</td>
<td>21.69</td>
<td>136.60</td>
<td>17.23</td>
<td>17.16</td>
<td>20.97</td>
<td>0.02</td>
<td>-0.72</td>
<td>-1.91</td>
</tr>
<tr>
<td>5</td>
<td>30.84</td>
<td>17.68</td>
<td>139.83</td>
<td>15.59</td>
<td>29.83</td>
<td>17.16</td>
<td>-1.01</td>
<td>-0.52</td>
<td>1.31</td>
</tr>
<tr>
<td><strong>Set 2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>56.16</td>
<td>27.20</td>
<td>139.47</td>
<td>16.56</td>
<td>55.65</td>
<td>27.91</td>
<td>-0.51</td>
<td>0.71</td>
<td>-0.22</td>
</tr>
<tr>
<td>2</td>
<td>22.61</td>
<td>56.95</td>
<td>138.73</td>
<td>16.41</td>
<td>22.89</td>
<td>55.65</td>
<td>0.28</td>
<td>-1.30</td>
<td>-0.95</td>
</tr>
<tr>
<td>3</td>
<td>-4.06</td>
<td>24.00</td>
<td>139.52</td>
<td>16.20</td>
<td>-4.85</td>
<td>22.89</td>
<td>-0.79</td>
<td>-1.12</td>
<td>-0.17</td>
</tr>
<tr>
<td>4</td>
<td>29.36</td>
<td>-4.17</td>
<td>141.02</td>
<td>15.90</td>
<td>27.91</td>
<td>-4.85</td>
<td>-1.45</td>
<td>-0.69</td>
<td>1.33</td>
</tr>
<tr>
<td>5</td>
<td>55.32</td>
<td>28.02</td>
<td>140.39</td>
<td>16.16</td>
<td>55.65</td>
<td>27.91</td>
<td>0.33</td>
<td>-0.10</td>
<td>0.70</td>
</tr>
<tr>
<td><strong>Set 3</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>32.03</td>
<td>25.73</td>
<td>138.94</td>
<td>16.16</td>
<td>31.38</td>
<td>24.90</td>
<td>-0.65</td>
<td>-0.83</td>
<td>0.34</td>
</tr>
<tr>
<td>2</td>
<td>26.19</td>
<td>31.42</td>
<td>138.81</td>
<td>16.38</td>
<td>25.90</td>
<td>31.38</td>
<td>-0.29</td>
<td>-0.04</td>
<td>0.22</td>
</tr>
<tr>
<td>3</td>
<td>20.12</td>
<td>26.70</td>
<td>138.96</td>
<td>16.29</td>
<td>19.42</td>
<td>25.90</td>
<td>-0.70</td>
<td>-0.80</td>
<td>0.37</td>
</tr>
<tr>
<td>4</td>
<td>25.17</td>
<td>19.41</td>
<td>137.67</td>
<td>16.60</td>
<td>24.90</td>
<td>19.42</td>
<td>-0.27</td>
<td>0.02</td>
<td>-0.93</td>
</tr>
<tr>
<td>5</td>
<td>31.37</td>
<td>24.89</td>
<td>138.46</td>
<td>16.17</td>
<td>31.38</td>
<td>24.90</td>
<td>0.01</td>
<td>0.01</td>
<td>-0.14</td>
</tr>
</tbody>
</table>

### Center Point Average Average
<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Height</th>
<th>Apex Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set 1</td>
<td>25.53</td>
<td>25.92</td>
<td>138.51</td>
</tr>
<tr>
<td>Set 2</td>
<td>26.02</td>
<td>26.00</td>
<td>139.69</td>
</tr>
<tr>
<td>Set 3</td>
<td>25.88</td>
<td>25.81</td>
<td>138.60</td>
</tr>
</tbody>
</table>

*a* - The Average Position is the average axis position relative to the Caster board rotated with the board through the five rotations.

*b* - The Error is the Average axis location minus the Determined axis position.
each turn of the caster board an ideal square was produced. The square center is at the point (25.4, 25.4) (the center for the caster board). In figure 25a, it is observed that the square formed by the average position plots are the same size and orientation as the squares produced by the determined axis plots only offset. The offset is due to the camera not finding the true axis location.

The upright plant shape results are plotted in figure 26, and the data is in table 6. The overall average error for this shape was -3.04 cm in the x direction, -2.17 cm in the y direction, and -0.26 cm in the z direction with standard deviations of 4.24 cm, 2.63 cm, and 3.53 cm respectively. A plot of the xy error (figure 26a) shows that there is only one error that is outside a 10 cm error radius. The error was in the second set, third run and of the value 16.8 cm in the xy plane. The other errors were considerably less. The next largest error was the value 7.42 cm in the xy plane.

In observing the plot of the determined axis locations (figure 26b) the first and third plots are fairly square while the second has one corner improperly located. The center points of the squares are (25.30 cm, 26.98 cm) for the first set, (32.09 cm, 29.34 cm) for the second set, and (27.28 cm, 25.36 cm) for the third set. These are offset from the center of rotation (25.4 cm, 25.4 cm). The offset is largest for the second set which corresponds to the set with the largest errors. The actual placement of the sides on the plants can be shown in figure 27. These images correspond to the first set of runs for the upright plant.
### Table 6
Upright Plant Shape Test Results
Determined Axis Position and Radius
Distances are in Centimeters

<table>
<thead>
<tr>
<th>Axis Position Determined X</th>
<th>Y</th>
<th>Z</th>
<th>Radius</th>
<th>Average Position a X</th>
<th>Y</th>
<th>X-Err</th>
<th>Y-Err</th>
<th>Z-Err</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Set 1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>44.78</td>
<td>18.67</td>
<td>151.69</td>
<td>30.43</td>
<td>42.06</td>
<td>14.31</td>
<td>-2.72</td>
<td>-4.36</td>
</tr>
<tr>
<td>2</td>
<td>39.40</td>
<td>40.06</td>
<td>157.81</td>
<td>33.02</td>
<td>36.49</td>
<td>42.06</td>
<td>-2.91</td>
<td>2.00</td>
</tr>
<tr>
<td>3</td>
<td>6.91</td>
<td>38.35</td>
<td>159.11</td>
<td>33.15</td>
<td>8.74</td>
<td>36.49</td>
<td>1.83</td>
<td>-1.86</td>
</tr>
<tr>
<td>4</td>
<td>11.84</td>
<td>12.32</td>
<td>157.68</td>
<td>34.65</td>
<td>14.31</td>
<td>8.74</td>
<td>2.47</td>
<td>-3.58</td>
</tr>
<tr>
<td>5</td>
<td>43.08</td>
<td>17.20</td>
<td>159.41</td>
<td>31.14</td>
<td>42.06</td>
<td>14.31</td>
<td>-1.02</td>
<td>-2.89</td>
</tr>
<tr>
<td><strong>Set 2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>46.13</td>
<td>24.94</td>
<td>157.73</td>
<td>33.43</td>
<td>39.59</td>
<td>23.14</td>
<td>-6.54</td>
<td>-1.80</td>
</tr>
<tr>
<td>2</td>
<td>28.32</td>
<td>45.31</td>
<td>157.15</td>
<td>36.47</td>
<td>27.66</td>
<td>39.59</td>
<td>-0.67</td>
<td>-5.73</td>
</tr>
<tr>
<td>3</td>
<td>26.01</td>
<td>35.71</td>
<td>163.73</td>
<td>26.59</td>
<td>11.21</td>
<td>27.66</td>
<td>-14.80</td>
<td>-8.06</td>
</tr>
<tr>
<td>4</td>
<td>27.89</td>
<td>11.40</td>
<td>155.22</td>
<td>30.45</td>
<td>23.14</td>
<td>11.21</td>
<td>-4.74</td>
<td>-0.19</td>
</tr>
<tr>
<td>5</td>
<td>42.32</td>
<td>25.32</td>
<td>157.23</td>
<td>34.59</td>
<td>39.59</td>
<td>23.14</td>
<td>-2.73</td>
<td>-2.18</td>
</tr>
<tr>
<td><strong>Set 3</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>36.20</td>
<td>24.46</td>
<td>156.90</td>
<td>34.70</td>
<td>39.40</td>
<td>26.26</td>
<td>3.21</td>
<td>1.80</td>
</tr>
<tr>
<td>2</td>
<td>29.24</td>
<td>39.47</td>
<td>158.14</td>
<td>35.15</td>
<td>24.54</td>
<td>39.40</td>
<td>-4.69</td>
<td>-0.07</td>
</tr>
<tr>
<td>3</td>
<td>13.94</td>
<td>25.55</td>
<td>158.93</td>
<td>33.93</td>
<td>11.40</td>
<td>24.54</td>
<td>-2.55</td>
<td>-1.01</td>
</tr>
<tr>
<td>4</td>
<td>29.74</td>
<td>11.96</td>
<td>147.83</td>
<td>31.57</td>
<td>26.26</td>
<td>11.40</td>
<td>-3.48</td>
<td>-0.56</td>
</tr>
<tr>
<td>5</td>
<td>45.64</td>
<td>30.28</td>
<td>159.59</td>
<td>29.34</td>
<td>39.40</td>
<td>26.26</td>
<td>-6.24</td>
<td>-4.02</td>
</tr>
</tbody>
</table>

**Center Point**

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>25.30</td>
<td>26.98</td>
<td>158.50</td>
<td>32.99</td>
<td>Average Error</td>
</tr>
<tr>
<td>32.09</td>
<td>29.34</td>
<td>158.46</td>
<td>31.74</td>
<td>STD of Error</td>
</tr>
<tr>
<td>27.28</td>
<td>25.36</td>
<td>155.45</td>
<td>33.84</td>
<td></td>
</tr>
</tbody>
</table>

---

a - The Average Position is the average axis position relative to the Caster board rotated with the board through the five rotations.

b - The Error is the Average axis location minus the Determined axis position.
(a) Narrow Upright Plant
Plot of Determined Axis Position Error
Error is the Determined Position subtracted from the Average Position

(b) Narrow Upright Plant Test
Plot of Axis Position Determined by the Image System

Figure 26
Graphs for Narrow Upright Plant Test
Figure 27
Line Placement for The First Set Runs using The Narrow Upright Plant.
Figure 27
(Continued)
The cone shaped plant results produced the greatest errors. The average errors were -0.84cm in the x direction, 2.01cm in the y direction, and -1.17cm in the z direction with corresponding standard deviations of 8.24cm, 4.25cm, and 9.48cm respectively. The results for the cone shaped plant are shown in table 7. An xy error plot (figure 28a) shows that the third set of runs had several points outside the 10cm radius. The plot of axis locations for the sets show that the general square shape holds true for the first two sets but does not hold for the last (see figure 28b). The center of the squares for the sets are (22.87cm, 23.80cm), (23.65cm, 25.49), and (31.91cm, 21.74) respectively. The third set has the largest offset and corresponds to the set that has the largest errors. The images of the first set are in figure 29a-d. It is noted that the lines representing the edges of the plant are not all placed identically.

The test with the cone object on the caster board established the ability of the developed system to locate the center axis. The average error and standard deviation for the cone object established the tolerance of the system. By comparing the results of the cone object to that of the live plants the ability of the system to represent the plants was measured. The standard deviations for the cone object were all less than 1cm indicating that the determination of the control point was not dependent on the view. A comparison of the standard deviations for the live plants shows that the cone plant is the more difficult shape to represent with a standard
<table>
<thead>
<tr>
<th>Axis Position Determined</th>
<th>Average Position</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
<tr>
<td><strong>Set 1</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>37.59</td>
<td>33.66</td>
</tr>
<tr>
<td>2</td>
<td>3.96</td>
<td>42.19</td>
</tr>
<tr>
<td>3</td>
<td>6.91</td>
<td>8.86</td>
</tr>
<tr>
<td>4</td>
<td>44.30</td>
<td>10.46</td>
</tr>
<tr>
<td>5</td>
<td>36.32</td>
<td>33.68</td>
</tr>
<tr>
<td><strong>Set 2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>46.33</td>
<td>0.15</td>
</tr>
<tr>
<td>2</td>
<td>46.23</td>
<td>45.39</td>
</tr>
<tr>
<td>3</td>
<td>1.04</td>
<td>47.09</td>
</tr>
<tr>
<td>4</td>
<td>0.99</td>
<td>9.35</td>
</tr>
<tr>
<td>5</td>
<td>39.70</td>
<td>2.82</td>
</tr>
<tr>
<td><strong>Set 3</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>52.50</td>
<td>9.19</td>
</tr>
<tr>
<td>2</td>
<td>33.58</td>
<td>45.49</td>
</tr>
<tr>
<td>3</td>
<td>33.12</td>
<td>23.04</td>
</tr>
<tr>
<td>4</td>
<td>8.43</td>
<td>9.22</td>
</tr>
<tr>
<td>5</td>
<td>53.04</td>
<td>11.33</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Center Point</th>
<th>Average</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Y</td>
<td>Height</td>
</tr>
<tr>
<td><strong>Set 1</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22.87</td>
<td>23.80</td>
<td>211.97</td>
</tr>
<tr>
<td><strong>Set 2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.65</td>
<td>25.49</td>
<td>211.52</td>
</tr>
</tbody>
</table>

a - The Average Position is the average axis position relative to the Caster board rotated with the board through the five rotations.

b - The Error is the Average axis location minus the Determined axis position.
Figure 28
Graphs For Cone Plant Test
Figure 29
Line Placement for The First Set Runs using The Cone Shaped Plant.
deviation of 8.24cm, 4.25cm, and 9.48cm in the x, y, and z directions respectively. Though most of the errors were within the 10 cm radius, the large standard deviation is evidence that the control point determined varies greatly, depending on the view of the cameras. The variance in determining the upright plant was less than for the cone shaped plant. The standard deviations for the upright shaped plants were about half that of the cone shaped plants. The variance in placing the axis can be seen in the images from the first set of runs of each image (figures 27 and 29) where the lines are not in the same place for all of the views. The placement of the edge lines on the plants is observed to be predominately on the edge of the plants. The narrow upright plant had a few placements that were within the plant but generally they were located on the edge. The placement of the edge lines on the conical plant was mainly in the edge of the plant to off the plant. Eighty-seven percent of the runs performed within a 10cm error radius.

7.3 Further Discussion

The experimental system was tested under some ideal situations and with several assumptions. Ancillary investigations were done to further probe the accuracy of the system. One question that came up during the tests was the sensitivity of the system to camera angle. To test the sensitivity three runs were done with the left camera angles changed. Table 8 shows the camera angles and the resulting position
Table 8
Sensitivity Test to Camera Angle

<table>
<thead>
<tr>
<th>Angle Alteration</th>
<th>Angle of Rotations</th>
<th>Determined Position</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Z-Angle</td>
<td>Y-Angle</td>
</tr>
<tr>
<td>Correct Angles</td>
<td>17.445</td>
<td>103.05</td>
</tr>
<tr>
<td>+1 Deg Y Angle</td>
<td>17.445</td>
<td>104.05</td>
</tr>
<tr>
<td>+.5 Deg Z angle</td>
<td>17.945</td>
<td>103.05</td>
</tr>
</tbody>
</table>

found. The tests show that the greatest error was introduced if the Z Angle was not determined correctly. This also shows the need for a method that finds the angles accurately. In determining the orientation of the cameras, described in section 6.2.3, a one half degree error could be introduced by a mistake of location of 3.07cm when the camera is at a 350cm distance. This may be critical because the resolution of the 100 by 100 matrix camera at 350cm (approximate distance between the cameras and the plant) is approximately 1.84cm per pixel. Therefore the placement of the target can only be accurate to 1.8cm in any direction.

Another factor related to accuracy is the ability of the investigator to accurately measure the position (usually about +/- 0.5cm). The combination of these errors can create up to a 2.5 cm error resulting in a 0.5 degree error in the camera orientation. A mistake in placement could be the reason for the large average error for the results presented in section 7.2.1. The small error in the results for the second part of section 7.2 shows that the camera angles can be determined accurately.
An investigation was also performed to investigate the effect of lighting in finding the axis of the plant. It was noted during the test that the lighting had to be uniform about the plant or the lines could not be determined correctly. This particular problem was most prominent with the cone shaped plant. The system was run three times with the cone shape plant unmoved, the first run was with adequate lighting, the second with left back light slightly turned, and the final with left back light not facing the plant. In figure 30 the placement of the edge lines for the three runs is shown. The change in the side lines and the placement of the axis is slight between the first two, but the third run was a definite failure with the right line on the images not on the right edge of the plant.
(a) Correct Lighting

(b) Left Light Slightly Turned

(c) Left Light Facing Opposite Direction

Figure 30
The Effect of lighting in Determining The Edge Lines.
Chapter VIII

Summary and Conclusions

The intent of this research was to develop a stereoscopic vision system capable of providing control information to an automated trimmer. The hypothesis was that a stereoscopic vision system could determine the shape and location of a plant that was to be trimmed. The exact research was limited to the cone and narrow upright shaped evergreens.

It was determined that the key characteristic in the image of a plant was the edge of the plant. From the edges, the axis, the top, and the shape of the plant could be determined. After reviewing several image processing methods, the process of edge transform followed by the Hough transform was selected to determine the edges of the plant. This process provided the equations of the lines which represented the two edges of the plant in an image. From the two lines the location of the center axis and either the radius for the narrow upright shape or the side angle for the cone shape could be calculated. The center axis and the height of the plant were represented by a point at the top of the plant and on the axis.
The information gathered from both images in the system was combined to calculate the three-dimensional information.

This research employed a camera control system, developed by The Ohio State University Electrical Engineering Department using two Reticon cameras, to obtain the digital images. The digital images were 100 by 100 pixels with 64 grey-levels. The contrast between the plant and the background was near optimal for most of the test. The developed system was tested in two parts, 1> the ability to determine the sides of an object and center axis in an image and 2> the ability to determine the shape and location of an object in a three dimensional setting.

The first part used two-dimensional shapes, triangle and rectangle, to test the accuracy of determining the center axis. The system was capable of determining the center axis within +/- .5 pixels on the image. It was determined that the sides of the object did not have to be straight but could be jagged. The test to determine the best horizontal camera angle revealed that the angle did not effect the system's ability to determine the center axis when the angle was less than 30 degrees.

The three-dimensional test for determining the objects (a cone and a cylinder) location showed an average error of 3.83cm. The second part using a caster board to rotate the cone object displayed better results with all the errors less then 1.5cm.
The tests using the live plants located the plant position within +/- 10cm xy error at least 87% of the time.

As a result of this research the following conclusions were reached:

1. A stereoscopic image processing system can determine the location and attributes of a upright type nursery plant.

2. The control information needed to shear a plant is the location of the center axis of the plant, the top of the plant and a shape characteristic (the radius for narrow upright plants, and side angle for conical plants).

3. The horizontal angle of the cameras does not effect the accuracy of the system when in the range of 0 to 30 degrees.

4. The stereoscopic image processing system was capable of determining the control information for ideal shapes.
5. The stereoscopic image processing system was capable of determining the shape characteristic (radius, side angle) of plants with accuracy of 85 percent.

6. The stereoscopic image processing system was capable of determining the center axis xy location within +/- 10 cm at least 87% of the time.

The system developed can provide control information to an automated shearer and conceivably could be used for other applications. This method has the potential to give control information to sprayers so that spot spraying could be performed. Automated diggers may be able to use the control information to remove plants for transplanting. The system described has the potential to replace inexperienced and transient labor with a consistent automated system.
Chapter IX

RECOMMENDATIONS FOR FURTHER STUDY

This research demonstrated that a stereoscopic vision system can obtain plant information; the first step in the development of a real time plant shearing system. During the research several questions arose that were not answered in the scope of this thesis. The questions will need to be answered prior to further development of the system. The following are the areas which require further study:

1. The effect of the contrast between the plant and its background. A high contrast material was used as the background of the image. What would be required if the image had a different background (e.g. grass, soil, weeds, etc.)?

2. The effect of another plant in the image. A single plant was used in each image. What if another plant was located within the image?
3. The effect of image resolution on the accuracy of the system. A 100 by 100 image resolution was used in this research. Would the system be more accurate if a camera with higher resolution were used?

4. The processing time needs to be shortened. Presently the system is slow. There might be several methods of speeding up the process. A quicker method needs to be determined.

5. The present system determines linear equations for the sides of the plants. Would the system be more accurate if the edges were fitted to an arc or some other curve rather then a line?

6. The system needs to be expanded to include other plant shapes. What equation would best fit the other shapes (eg. spherical plant shape, cubic plant shape, etc)?
LIST OF REFERENCES


Appendix A

Listing of Pascal Program
PROGRAM Plant_Characteristics(OUTPUT, KEYBOARD);
lable 100;
IMPORT SYSXORLS, SYSXORG, IOCOMNRM, TODECKLARATIONS,
              GENERAL_1, GENERAL_2, GENERAL_4, HPIB_0, HPIB_1, HPIB_2, HPIB_3;

type
    scales = array[0..63] of integer;
    string = string[255];
    Rmatrix = array[1..9] of real;
    Imatrix = array[1..3] of real;
    integers = array[0..8] of integer;

const
    eaxbytes = 10239;

  type
    byte = 0..10236;
    ary = array[0..eaxbytes] of byte;
    frames = array[1..10000] of char;
    frame = *frames;
    frames2 = array[1..10000] of integer;
    frame2 = *frames2;
    cameras = (right, left, both);

var
    image, image, origin, origin: frame;
    picts: file of frames;
    buffer: buf_info_type;
    rmat1, rmat2: Rmatrix;
    cam: cameras;
    clrn : integer;

  type
    err = array[-8..17] of integer;
    MArr = array[1..15, 1..15] of real;
    xy = (r, c);
    ntr = array[xy] of real;
    pointptr = array[0..1000] of integer;
    ntr = *pointptr;
    xyz = (x, y, z);
    realworld = array[xyz] of real;
    liars = array[0..11] of real;
    shapes = (cone, cylinder);

var
    answer : char;
    camangle : real;
    string : strings;
    VALID : BOOLEAN;
    SHAPE : SHAPES;
    second, second, second, 0IF : real;
    realpoint : realworld;
const
pi = 3.141592654;

{matr = {matrix[115.625, 82.25, 72.5];
{matl = {matrix[117.125, -24.625, 72.5];
{frl = 0.468;
{fr = 0.567;

procedure delayji(integer);
var
  tin : integer;
begin
  tin := sysclock;
  while sysclock((tin+1000)*di)
do;
end:

procedure command(var buffer:buf_info_type;string:string);
var
  strg: strings;
begin
  while buffer_busy(buffer) do;
    strg:="Sending the command ";
    strappend(strg,string);
    buffer_reset(buffer);
    writebuffer_string(buffer,string);
    transfer_end(703, SERIAL_UART, FROM_MEMORY, buffer);
end:

procedure receive(var buffer:buf_info_type;var image:frame;string:string);
var
  clr:integer;
  temp: char;
  strg: strings;
begin
  strappend(string,"P");
  command(buffer,string);
  while spall(703) = 0 do;
    strg:="Receiving the image";
    strinsert(strg,string,15);
    transfer(703, SERIAL_UART, TO_MEMORY, buffer, 1000);
    for clr:=1 to 10000 do begin
      readdbuffer(buffer,temp);
      image[clr]:=chr(ord(temp) mod 64);
    end;
end;

var
cnt:integer;
strg: strings:

begin
strappend(strg, 'G');
command(buffer, strg);
for cnt := 1 to 10000 do
  writelnbuffer(buffer, image[cnt]);
writelnbuffer(buffer, 'V');
wsnok(buffer, 'V', FROM_MEMORY, buffer);
end;
PROCEDURE writepicture(image :FRAME):

TYPE
  LEVELS = ARRAY[0..8] OF INTEGER;
VALUES = ARRAY[0..255] OF INTEGER:
VAR
  I,J,K,L : INTEGER;
  A : ARRAY[0..1000] OF INTEGER;
  A, B, C, E = STRING(100);
  U1, U2, U3, U4 : VALUES;
  U5 : STRING(10);
  CONS;
  LEVEL = LEVELS[0, 8, 20, 64, 85, 93, 111, 172, 255];
  LEVEL2 = LEVELS[16, 40, 32, 170, 174, 245, 254, 255];

PROCEDURE setup(var U1, U2, U3, U4 :VALUES):

VAR
  I, J, K, L, T : INTEGER;

BEGIN
  FOR I := 0 TO 7 DO
    FOR J := 1 TO (I+1) DO
      FOR K := 1 TO (I+1) DO
        FOR L := 1 TO (I+1) DO BEGIN
          I := I+J+K+L;
          U1[I] := LEVEL[I];
          U2[I] := LEVEL2[I];
          U3[I] := LEVEL3[I];
          U4[I] := LEVEL4[I];
        END;
    END;
  END;
END:

BEGIN
  SETUP(U1, U2, U3, U4);
F := CHR(27) + "c1280s" + CHR(27) + "oR";
G := CHR(27) + "6264" + "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
LISTEN(7, 1);
TALK(7, MY_ADDRESS(7));
WRITECHAR(7, CHR(10));
WRITESTRING(701, F);
FOR J := 0 TO 99 DO BEGIN
  FOR I := 1 TO 100 DO BEGIN
    AL[I] := ord(image"[I + 100]"[J + 100]);
  END;
  E := CHR(27) + "61000";
  WRITESTRING(701, E);
  FOR J := 1 TO 100 DO BEGIN
    K := AL[I] DIV 2;
    IF K > 31 THEN K := 31;
    IF (K = 0) THEN K := 0;
    WRITECHAR(7, CHR(VICK));
  END;
  WRITESTRING(701, E);
  FOR J := 1 TO 100 DO BEGIN
    K := AL[I] DIV 2;
    WRITECHAR(7, CHR(VICK));
  END;
  WRITESTRING(701, E);
  FOR J := 1 TO 100 DO BEGIN
    K := AL[I] DIV 2;
    WRITECHAR(7, CHR(VICK));
  END;
  END;
END;
E := CHR(27) + "m8";
WRITESTRING(701, E);
END;

procedure setup_cpu(var buffer:buf_info_type);
begin
  initialize;
  iobuffer(buffer, 10001);
  buffer_reset(buffer);
  clear(703);
end;
delay(2);
end;
procedure initialize(var image, inager: frame);
var
  ctr: integer;
begin
  new(image);
  new(inager);
  for ctr := 1 to 10000 do begin
    image^[ctr] := ord(ctr0);
    inager^[ctr] := ord(ctr0);
  end;
end;

procedure transform(var image: frame);
var
  ctr, i, j, k, s, d1, d2, r1, r2, r: integer;
  ins: ary;
const
  matrix1 = integers[1, 2, 1,
                     0, 0, 0,
                     -1, -2, -1];
  matrix2 = integers[1, 0, -1,
                     2, 0, -2,
                     1, 0, -1];
begin
  for ctr := 1 to 10000 do begin
    image^[ctr] := ord(ctr0);
    for i := 0 to 8 do begin
      dl := 0;
      dr := 0;
      for ctr := 0 to 8 do begin
        dl := dl + matrix1^[ctr];
        dr := dr + matrix2^[ctr];
      end;
      if dl = 0 then dl := 1;
      if dr = 0 then dr := 1;
      for j := 3 to 97 do begin
        for k := 2 to 90 do begin
          i := k*100 + j;
          a := matrix1^[i] + matrix1^[i+100] + matrix1^[i+200] + matrix1^[i+300];
          b := matrix2^[i] + matrix2^[i+100] + matrix2^[i+200] + matrix2^[i+300];
          c := matrix1^[i] + matrix2^[i] + matrix2^[i+100] + matrix2^[i+200];
          rl := abs(a + b + c) div dl;
          if rl > 63 then rl := 63;
        end;
      end;
    end;
  end;
end;
a := matrix2[i] + img[i-100] + matrix2[i+1] + matrix2[i] + img[i-99];
b := matrix2[i] + img[i-1] + matrix2[i] + matrix2[i] + img[i+1];
r2 := xabs((a + b) div d1);
if r2 < d1 then r2 := d1;
if r2 < r1 then image[i] := chr(r1)
else image[i] := chr(r2);
end;
end;
end;

Procedure Hough (Var Hough :frame2; A, B: integer);
{
   Hough = Hough Area
   A = Point Of Maximum Value (to be calculated)
   B = Maximum Value (to be Calculated)

   Var
      I, Temp : integer; (Free Variable)
      Temp : integer; (temporary Placement)

   Begin
      Temp := 1;

      For I := 1 to 10000 do
         Begin
            If Hough[Temp] > Hough[I] then
               Temp := I;
         End;
      A := Temp;
      B := Hough[CA];
      Hough[Ia] := 0;
   End;

   Procedure Locate_Line (Hough: frame2; Bin : integer; Slo, Inc : real; 
                           Var Line : list);
   {
      Hough = Hough Area
      Bin = Minimum Intercept
      Slo = First Slope Value
      Inc = Slope Increase
      Line = Line Equation (to be Calculated)
         Line[0] = Slope
         Line[1] = Intercept

      Type

\( A[i] = \text{array}[0..n] \) of Integer:

Var

\( n, i, r, x, i, j, \text{TEMP} \) : integer;

Begin

\( \text{Hough}(\text{Hough}, i, \{0\}) ; \)

\( \text{ROI} := (j \text{ and } 100) \times \text{ROI} ; \)

\( \text{ROI} := (j \text{ div } 100) \times \text{ROI} ; \)

\( \text{TEMP} := j ; \)

\( i := 1 ; \)

WHILE \( i < \text{n} \) DO

Begin

\( \text{Hough}(\text{Hough}, i, \{1\}) ; \)

\( \text{ROI} := (j \text{ and } 100) \times \text{ROI} ; \)

\( \text{ROI} := (j \text{ div } 100) \times \text{ROI} ; \)

IF \( \text{ROI}(i) \times 100 \neq \text{ROI}(j) \times 10 \) THEN

IF \( \text{ROI}(i) \times 100 \neq \text{ROI}(j) \times 10 \) THEN

\( i := i + 1 ; \)

End;

\( \text{Line}[0i] := (\text{ROI}[i] + \text{ROI}[1] + \text{ROI}[2] + \text{ROI}[3]) / (\text{ROI}[0] + \text{ROI}[1] + \text{ROI}[2] + \text{ROI}[3]) ; \)

\( \text{Line}[1i] := (\text{ROI}[i] + \text{ROI}[1] + \text{ROI}[2] + \text{ROI}[3]) / (\text{ROI}[0] + \text{ROI}[1] + \text{ROI}[2] + \text{ROI}[3]) ; \)

\( \text{Line}[2i] := \text{Slo} + \text{Line}[0i] \times \text{Inc} ; \)

\( \text{Line}[1i] := \text{Line}[1i] \times \text{Inc} ; \)

End;

Procedure \( \text{Place Points}(\text{Image} : \text{frame} ; \text{Hough} : \text{frame}^2 ; \text{I}, \text{Threshold}, \text{Min} : \text{Integer} ; \text{Slo}, \text{Inc} : \text{real} ) ; \)

\( \{ \)

\( \text{Hough} - \text{Hough} \text{Area } 10,000 \text{ Points} \)

\( \text{I} - \text{Point In Image to be transformed} \)

\( \text{Threshold} - \text{Threshold Value} \)

\( \text{Min} - \text{Minimum Intercept Value} \)

\( \text{Slo} - \text{Starting Slope Value} \)

\( \text{Inc} - \text{Slope Increment} \)

\( \} \)

Var

\( r, c, i, j, l, \text{ROI} \) : integer;

\( \text{ROI} \) : real;

Begin

\( \text{ROI} := \text{I} \text{ div } 100 ; \)
\( \ell := I \mod 100; \)

For \( L := 1 \) to 50 do

Begin

\( R := L + \text{Inc} \times \text{Slo}; \)
\( R := \text{Round}(C - R 	imes C + \text{Min}); \)
If \((B > 0)\) and \((B < 100)\) then

Begin

\( I := B + 100 + L; \)
\( \text{Hough}[I] := \text{Hough}[I] + \text{ord}(\text{image}[J]) \times \text{Thresh}; \)
end;
end;
end;

Procedure HoughTransform(image : frame, thresh, Up, Low, Min line integer;
Slo, Inc := real; Var line: line);

{ Hough Transform}

Var

Hough : frame;
I : integer;

Begin

H(i) := 0;

for \( I := 1 \) to 10000 do

Begin

\( I := I + \text{Inc} \)
\( \text{Hough}[I] := \text{Hough}[I] \times \text{Thresh}; \)
End;

for \( I := \text{Up} \) to \( \text{Low} \) do

Begin

IF \((I \mod 100) \equiv 76\) THEN

It Ord(image[I]) > thresh THEN

\( \text{Place Points}(\text{image}, \text{Hough}, I, \text{thres}, \text{min}, \text{slo}, \text{inc}); \)
\( \text{End Loop} \)

End;

End:

Locate Line(Hough, Min, Slo, Inc, Line); (Call Procedure to Find the Line)
Dispose(Hough); (Remove Hough From Memory)

End; (End Hough Transform Procedure)
Procedure Line_intersect(Line1, Line2 : liar; var point: point; var angle : real);
{
Line1 & Line2 = The lines that the sides of the plant make
with lineY(0) = slope
lineX(1) = y-intersect
Point = the point that the lines intersect at
angle = angle between the lines
}

var
xx, yy, zz : real;

{ The intersection of two lines can be calculated by the
  equation X = (b1 - b2)/(a2 - a1)
  and Y = a2 * X + b1

  with the angle of intersection calculated to be

  angle = arctan((a1 - a2)/(1 + a1 * a2))
}

begin

xx := ((line1[1] - line2[1])/(line2[0] - line1[0]));
yy := xx * line1[0] + line1[1];
point[0] := yy;
point[1] := xx;
angle := (abs(arctan(line1[0])) + abs(arctan(line2[0]))) / 2;
angle := angle * 180 / pi;

end;

Procedure lines_parallel(frame : frame; line1, line2 : liar;
var point : point; var radius : real; col : integer):

var
i, j, k, l : integer;
xx, yy, zz : real;
const
th = 25;

begin

yy := (col * line1[0] + line1[1]) + (col * line2[0] + line2[1]) / 2;

k := round(yy);
l := 0;
for i := 5 to 70 do begin
j := ord(image[(k + 100 + i) * chr(th)]) + ord(image[(k + 1) * 100 + i] * chr(th))
  + ord(image[(k + 2) * 100 + i] * chr(th)) + ord(image[(k - 1) * 100 + i] * chr(th))
  + ord(image[(k - 2) * 100 + i] * chr(th)) + ord(image[(k - 5) * 100 + i] * chr(th));

if j > l then begin
xx := i;
L := j;
end;
}
end;

yy := (X*X + line[1] + line[1]) + (X*X + line[2]) + line[2]) / 2;
point[1] := xx;
point[2] := yy;
radius := abs((X*X + line[1]) - (X*X + line[2])) / 2;
POINT[1] := POINT[1] + 0.2 + radius;
end;

procedure draw_shape(var image : frame; linetype : LINE);
var
i,j,l : integer;

begin
for j := 2 to 90 do
begin
if abs(line[1]) < 0.001 then
i := round(line[1])
else
i := round(j * (HE) + line[1]);
if (j/10) AND (i<90) AND (j>90) then
image[1][i][j][l] := chr(32);
end;
for j := 2 to 98 do
begin
if abs(line[1]) < 0.001 then
i := round(line[1])
else
i := round(j * (HE) + line[1]);
if (j/10) AND (i<98) AND (j>98) then
image[1][i][j][l] := chr(32);
end;
end;

procedure scan_step(image : frame; q : real; var line : line);
label 20:
var
x,y,z,k : real;
i,j : integer;
begin
x := line[1] - 4;
y := 0;
for j := 2 to 98 do begin
if abs(line[1]) < 0.001 then
i := round(k)
ext:
else
i := round(j * (HE) + k);
if image[1][i][j][l] = chr(9) then
x := x + 1;
end;
y := q;
k := l + q;
20: if YX THEN X:=Y;
y := 0; k := k + q;
for j := 2 to 98 do begin
if abs(line[1]) < 0.001 then
i := round(k)
  else
i := round(j + LINE[0] + k);
if image[i][100+j][1] <> 29) then
  y := y + 1;
end;
if y > 0.7 * x then goto 20;
line[i] := k - y;
end;

procedure center(image :frame; var row, col: integer);
var
  area, momentr, momentc, x, y : real;
  i, j, k : integer;
var begin
  area := 0;
  momentr := 0;
  momentc := 0;
for i := 100 to 999 do begin
  momentr := momentr + ord(image[i][1]) * (i div 100);
  momentc := momentc + ord(image[i][1]) * (i mod 100);
end;
x := momentr / area;
y := momentc / area;
row := round(x * 100);
col := round(y * 100);
end;

Procedure determine_shape(image, old :frame; Shapel :shapes;
  var Point : unit; var second : real):
{
  Point - this is the point of intersection for the sides. It relates the
  center axis position and the height of the plant.
  Second - this is the secondary characteristic of the plant
  for the cone shape  = angle of the sides
  for the cylinder  = radius of the plant
  Shapel - declares what type of shape the plant is
  Cone - for cone shape
  Cylin - for cylinder shape
}

var
line1, line2, line : int;
i, j, k, l, row, col : integer;

const
thresh = 20;
minol = 0;
nin2 = 0;
nins1 = 0;
nins2 = 0;
sloc = 0.15;
incc = 0.04;
sloa = -0.01;
incs = 0.0009;
upcl = 1000;
ups2 = 100;
loc1 = 5900;
loc2 = 9000;
ups1 = 100;
ups2 = 5000;
locx2 = 9500;

begin
  if shapet = cone then begin
    hough_transform(image, thresh, upcl, loc1, nin1, sloa, incc, line1);
    hough_transform(image, thresh, ups2, loc2, nin2, sloa, -incc, line2);
    line_intersect(line1, line2, point, second);
  end:
  if shapet = cylin then begin
    center(image, row, col);
    hough_transform(image, thresh, ups1, row, nin1, sloa, incs, line1);
    scanside(old, -1, line1);
    hough_transform(image, thresh, row, ups2, nin2, sloa, incs, line2);
    scanside(old, 1, line2);
    lines_paral(image, line1, line2, point, second, col);
    cline(0) := line1[0];
    cline(1) := point[1];
    draw_shape(old, cline, cline);
  end;
  draw_shape(old, line1, line2); WRITE('UNIF');
end:

Procedure Combine(point0, point1, point2, point3, Realpoint : Realworld;
  shape: shapes; var second1, second2, secondM: Real;
  Realn1, Realn2: Realign; var fl, fr : real);

Var
  01, 02, 03, 04, 05, 06, 07, 08, 09,
  u1, u2, u3, v1, v2, v3, w1, w2, w3,
  u2, 51, 52, 53 : real;
  i, j, k : integer;

Begin
  pointfr := (pointfr - SD) * 0.0024;
  pointfc := (pointfc - SD) * 0.0024;
  pointfr := (pointfr - 50) * 0.0024;
  pointfc := (pointfc - 50) * 0.0024;
readIn(shape);

ranf[11] := -0.29579 ;
ranf[12] := -0.2542 ;
ranf[13] := -0.52657 ;
ranf[14] := -0.35401 ;
ranf[15] := 0.067635 ;
ranf[16] := 0.29285 ;
ranf[17] := 0.0 ;
ranf[18] := 0.59147 ;
ranf[19] := -0.2259 ;
ranf[20] := 0.5613 ;
ranf[21] := -0.16609 ;
ranf[22] := -0.01077 ;
ranf[23] := 0.02761 ;
ranf[24] := -0.11255 ;
ranf[25] := -0.59989 ;
ranf[26] := 0.0 ;
ranf[27] := 0.59766 ;
ranf[28] := -0.20968 ;

end:

Begin
WRITEK(\"ENTER TITLE OF RUN SET\"); 
KCONS(STRING);
TALS7,MY_ADDRESS(7); 
LISTEN(7,1); 
WRITEK(\"701,STRING\"); 
WRITEK(\"701,STRING\"); 
WRITEK(\"701,TIME\"); 
setup(ranf, ranf, shape); 
nat(cir); 
HDD := 1; 
100:Setup_kpub(buffer); 
Initialize_imag,imag); 
initialize(origl,origl); 
THEG := SYSCLOCK; 
Receive(buffer,imag,"L","left"); 
Receive(buffer,imag,"R","Right"); 
Receive(buffer,origl,"L","left"); 
Receive(buffer,origl,"R","Right"); 

(IF SHAPE = CUBE THEN BEGIN)
transf_2(imag);

Transform3(Buffer);
send(Buffer,"X","left");
send(Buffer,"Y","right");

Determine_shape(Buffer,origl,shape,pointl,secondl);
Determine_shape(Buffer,orig2,shape,point2,second2);
Combine(pointl,point2,realpoint,shape,secondl,second2,secondl,second2,secondl,second2,

Pointl,Point2,fr);

 Radius := SYSHELP-TABS;
 writeln('Radius of Plant = '); 
 writeln(secondl);
 writeln('TIME = ',HR);

 send(Buffer,origl,"L","left");
 send(Buffer,orig2,"R","right");
 writeln('Print Images Y/N ?');
 Readin(answer);
 if answer='Y' then begin
 writestring(origl);
 writestringl(70,"left");
 writestring(orig2);
 writestringl(70,"right");
 end;

 WRITESTRING(STRING,1,1,MAG=7,REALPOINT1=(-3:2,REALPOINT2)=(-3:2,REALPOINT3)=(-3:2,
 DIF=(-3:4,SECOND1=(1:2,POINT1)=(1:2,POINT2)=(1:2,
 POINT3)=(1:2,POINT4)=(1:2);
 WRITESTRINGMT(70,STRING);
 writeln('Done');

MAG := MAG+1;

 dispense(origl);
 dispense(orig2);
 dispense(Buffer);
 dispose(inage);
 readin(second);
 release(clrln);
 inomint:=; 
goto 100;

End.
Figure 31
The Rotations About the Reference System \{A\}
To Form the New System \{B\}.

This produces a 3x3 matrix which is the rotational matrix from the \{A\} reference system to the \{B\} system. The inverse of this matrix is the rotational matrix from the \{B\} system to the \{A\} reference system. It is important to note that the order of rotation and multiplication is critical in producing the correct matrix and all of the angles are in the \{A\} reference frame.

The following is the SPEAKEASY program used to obtain the rotational matrix. The program calculates the matrix to translate from the world system to the camera system using the angle of rotations about the camera system. The calculated matrix is then inverted to form the the rotational matrix from the camera system to the world system.
Appendix B

Rotational Matrix Calculation

The rotational matrix for each camera was calculated as described by John J. Craig in *Introduction to Robotics Mechanics and Control*, 1986, pp 40-41. The following is a condensed form of the method described:

The method of orienting the coordinate system from the reference system \( \{A\} \) to the new system \( \{B\} \) is done by a rotation about each axis. Following figure 31 to obtain the new system three rotations are performed about the reference system axis. The first rotation is about \( X_\alpha \) by the angle \( \text{ALPHA}(a) \), next rotate about \( Y_\beta \) by the angle \( \text{BETA}(b) \), the final rotation is about \( Z_\gamma \) by the angle \( \text{GAMMA}(g) \). The rotational matrix can be calculated from the angles of rotation from the equation

\[
^b_0[R](a,b,g) = \text{ROT}(Z_\gamma g) \text{ROT}(Y_\beta b) \text{ROT}(X_\alpha a)
\]

\[
= \begin{bmatrix} c(g) & -s(g) & 0 \\ s(g) & c(g) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c(b) & 0 & s(b) \\ 0 & 1 & 0 \\ -s(b) & 0 & c(b) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(a) & -s(a) \\ 0 & s(a) & c(a) \end{bmatrix}
\]

where \( c(a) \) represents \( \cos(a) \) and \( s(a) \) represents \( \sin(a) \)
\begin{verbatim}
// T1 = 10
// T2 = (0,14)
// FPC1 = 1024
// J0 = 0
// LINKS = 2000, CNTS = 0, DISK10 = 5000, TAPF10 = 0

// 11000
// SYSTEM_DM

PROGRAM DET

ANGLES IN DEGREES

A = 0
B = 105.274
C = 10.54

HZ = RHT(1,4;CSS(A),-SIN(A),0,SIN(A),COS(A),0,0,0,1)
Hv = RHT(1,4;CSS(B),0,SIN(P),0,1,0,-SIN(P),0,0,COS(P))
Hz = RHT(1,4;CSS(C),-SIN(C),0,SIN(C),COS(C),0,0,0,1)
F = Hz
C = 1/F
PRINT F
PRINT C
END
END DET

//
//
\end{verbatim}