Abstract

Mummert, Craig, The development of a machine vision system to measure the shape of a sweetpotato root (Under the direction of Dr Michael D. Boyette, Dr Larry F Stikeleather, and Dr. M. K. Ramasubramanian)

The objective of this project is to develop and investigate whether a machine vision system can quantitatively describe the physical characteristics of a sweetpotato and then produce a unique number that is a more accurate descriptor of the root. Machine vision has been shown to accurately measure the size and shape of various agricultural commodities by mapping their surfaces. This map can then be used to describe the characteristics of the object. For this study a system was developed to rotate the root in front of a CCD imaging device that maps the entire surface of the sweetpotato. Algorithms were then developed to quantitatively measure, from this map, the length of the sweetpotato; the width of the sweetpotato; the ratio of length to width and how much the spine of the sweetpotato bends. These characteristics are then used to create a 10 digit numerical identifier that would uniquely describe each sweetpotato. The developed system was initially tested using five standard shaped objects. The system was shown to accurately measure both the length and width and produced straightness numbers that were indicative of the object being tested. 10 sweetpotatoes were then tested under the same procedure as the spheres. The results showed that the system could accurately measure the length and width for each sweetpotato within several millimeters. The system produced straightness numbers that were unique and reasonable for each sweetpotato.

The development of a machine vision system to measure the shape of a sweetpotato root

by

Craig Nevin Mummert

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Biological Engineering

Raleigh

2004

APPROVED BY:

_____ _____

_____

Chair of Advisory Committee

This thesis is dedicated to my wife, Melissa, and my family.

Biography

Craig Nevin Mummert was born September 12, 1980 in Pittsburgh, Pennsylvania. His family moved to Hendersonville, North Carolina in 1986 and lived there until he graduated from West Henderson High School in 1998. He attended North Carolina State University from August 1998 to May 2002 and obtained a Bachelor of Science degree in Biological Engineering, with a concentration in agriculture. He married his wife, Melissa (Nicholson) Mummert, in June 2002 and started his graduate studies at North Carolina State University in August 2002. He finished his masters in Biological and Agricultural Engineering in August 2004.

Table of Contents

List of Tables v
List of Figures vi

1. Introduction 1

1.1 Sweetpotato Industry 1
1.2 USDA Sweetpotato Grade Specifications 1
1.3 Variability in Human Observation 4
1.4 Use of Machine Vision to classify agricultural product 5
1.4.1 Machine Vision 6
1.4.2 Classifying Mushrooms and Tomatoes using Machine Vision 7
1.4.3 Classifying Apples and Potatoes using Machine Vision 8
1.5 The use of Machine Vision with Sweetpotatoes 9
1.6 Objectives of the thesis 10

2. Materials and Methods 12

2.1 Hardware 12
2.1.1 Viewing Platform 13
2.1.2 Imaging Device 19
2.2 Software 21
2.2.1 Stepper Motor Control 23
2.2.2 Physical Characteristics 24
2.2.2.1 Width 25
2.2.2.2 Length 27
2.2.2.3 Length per Width Ratio 28
2.2.2.4 Straightness 29
2.2.3 Numerical Identifier 33

3. Results and Discussion 35

3.1 Standard Objects 35
3.2 Sweetpotatoes 41

4. Conclusions 47

5. Bibliography 51

6. Appendices 53
Appendix A 53
Appendix B 63

List of Tables

Results and Discussion

3-1: Results of Five standard objects ……..……..……..……..……36
3-2: Standard Deviations for the Standard Objects …….…….38
3-3: Standard Deviations Normalized by the Object's Length …… 40
3-4: Standard Deviations Normalized by the Characteristic……. 40
3-5: Results from Inverting Ten Sweetpotatoes ……..... 45

List of Figures

Introduction

1-1: Standard Sweetpotato Shapes …….……3
1-2: Imaging flowchart ……...…….……... 6

Materials and Methods

2-1: Picture of Completed Base ……..…….. 13
2-2: Orientation of the system ……...14
2-3: Stepper Motor Control Circuit ……...16
2-4: Image of Viewing Platform ……... 18
2-5: 542c Smartimage Sensor and Tamron lens …….…….. 20
2-6: Front Panel from Labview Program ……..22
2-7: Inspection algorithm ……..……...……23
2-8: Illustration of Width Algorithm ……...……..……26
2-9: Illustration of Length Algorithm ……..……. 28
2-10: Illustration of Computed Centers of Mass ……...32
2-11: Location of Characteristics inside the Numerical Identifier …….……...34

Results and Discussion

3-1: Results for Width, Length, Ratio, and Straightness of Sweetpotatoes and Mini Footballs .. 42
3-2: Illustration of Sweetpotato Three and Ten …….…….. 44

1. Introduction

1.1 The Sweetpotato Industry in North Carolina

The sweetpotato industry has become a significant vegetable commodity in the United States, with over 14.6 million cwt being harvested in 2001 and a production value of almost 225 million dollars. North Carolina produced 38% of that total, making North Carolina the largest producer of sweetpotatoes (Ipomoeas batatas, L.) in the United States (USDA, 2003). However even though the production of sweetpotatoes has been increasing over the past several decades, the methods of classifying and grading the sweetpotato roots based on human observation have remained relatively the same. Consumers have been shown to buy produce that has a uniform and pleasing appearance (Wills *et al*. 1998). Therefore it is imperative to deliver sweetpotatoes to the consumer market that not only taste good, but also have a good shape and appearance. Several automated systems have been developed to aid growers in grading sweetpotatoes. However, due to the high cost of these systems their use on small farms and in breeding research is highly limited.

1.2 USDA sweetpotato grade specifications

The standard developed by the USDA for grading sweetpotatoes consists of four standard grades; with each grade having its own requirements on the size, weight and physical appearance of sweetpotatoes. For example the highest quality sweetpotatoes,

U.S. Extra No. 1, must have a length between three & nine inches with a diameter between 1-3/4 and 3-1/4 inches. The physical appearance of U.S. Extra No. 1 must have, "similar characteristics which are firm, smooth, fairly clean and fairly well shaped…" (USDA, 1997). While the size of a sweetpotato can easily be measured using a grading board (The Postharvest Handling of Sweetpotatoes, p 37), measuring the other physical characteristics has been left open to human observation and subjectivity. This subjectivity is illustrated by the definition of fairly well shaped (USDA, 1997):

> "Fairly well shaped", means the sweetpotatoes are not so curved, crooked, constricted or otherwise misshapen as to materially detract from the appearance of the individual sweetpotato or the general appearance of the lot"

There is no precise definition on exactly how much the sweetpotato has to be curved, crooked or misshapen in order for the sweetpotato to no longer be U.S. Extra No. 1. Iit is this subjectivity in the grading standards that has led to several different published methods of classifying sweetpotato shapes. The most complete process of classifying the roots was published in Descriptors for Sweet Potato (CIP 1981). This book outlines, among a large number of other characteristics, four different categories to describe the physical appearance of sweetpotato roots. These categories include root shape, root surface defects, root cortex thickness and root skin color. Each category has a set of corresponding numbers that represent a different characteristic for that corresponding category. For example, in the storage root shape category; there are nine different common shapes that sweetpotato might resemble (CIP 1981). Using this method a sweetpotato would be classified into which ever of these nine shapes the root resembles the most, based on subjective human judgment. The nine shapes outlined in Descriptors for Sweet Potato (CIP 1981) are illustrated below in **Figure 1-1.**

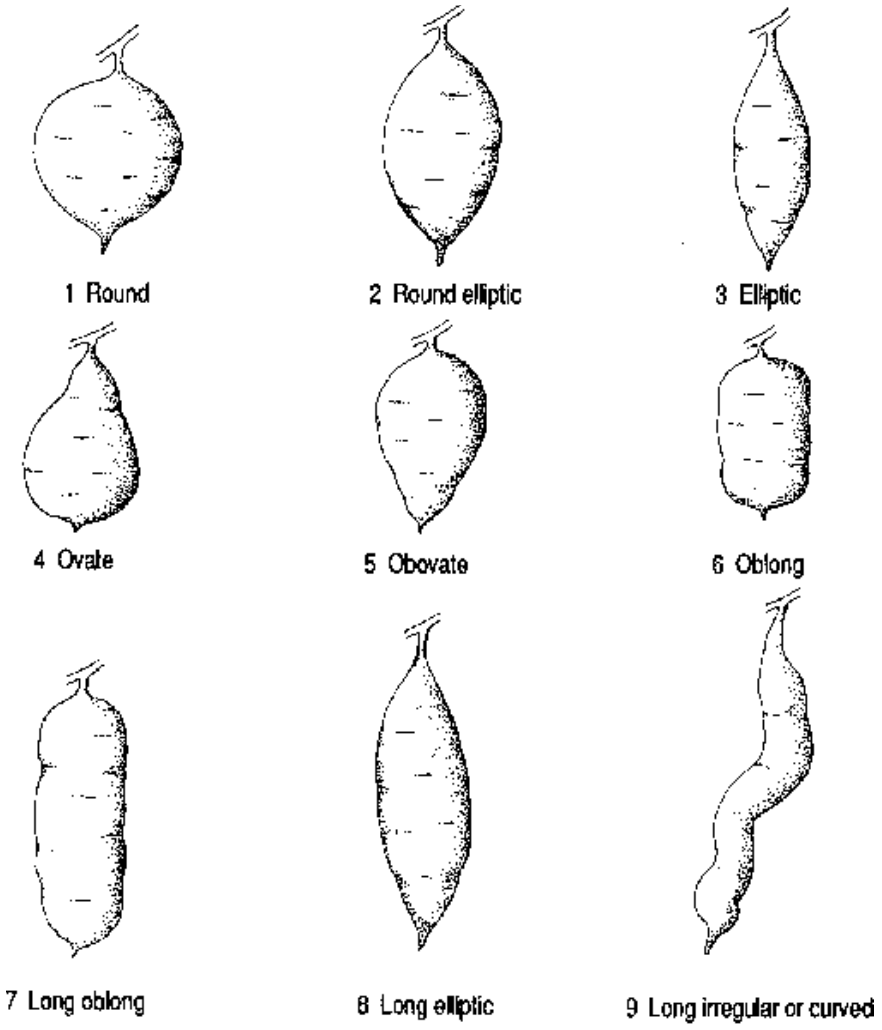


Figure 1-1: Standard Sweetpotato Shapes (CIP 1981)

A second study conducted by Jones, Steinbauer and Pope (1969) looked into which traits of a sweetpotato are heritable. In this study the shape was classified as a number from one to five. Using this method it was concluded that shape had a 62 percent heritability estimate (Jones *et al*., 1969). A final study by Martin and Rhodes looked at set of characteristics that could be used to describe a sweetpotato and how the characteristics correlated to each other. In this study they looked at a set of characteristics that could describe a sweetpotato root. For each characteristic they assigned a numerical

value that corresponded to set observation for that characteristic. For example, uniformity of color was rated with a one, two or three; with a one standing for very irregular and a three corresponding to very regular. They then looked into how each of these characteristics was affected by the other characteristics and came up with correlations between characteristics. They concluded that sweetpotatoes, based on the correlations from all the factors can be classified into seven different groups. They also concluded that shape was mainly influenced by length and width (Martin *et al*, 1983). These two studies not only provided two different means of classifying sweetpotato properties, they showed that shape was a breedable quality of sweetpotatoes and it had an effect on other characteristics that might be important to consumers. However, without a quantitative method to measure the shapes, the ability for researchers to select the best shapes remains limited to their own opinion of the shape.

1.3 Variability in Human observation

The procedures from each of the studies above in conjunction with the USDA standards can be used to classify a sweetpotato into a certain shape class. The problem that remains is that another person could repeat the same procedure and come up with a different classification for the same sweetpotato. This problem was clearly illustrated in a study of mushrooms by Heinemann, Hughes, Morrow, Sommer, Beelman and Wuest (1994). This study investigated grading mushrooms using a machine vision system. When testing the system, two expert inspectors each classified 25 mushrooms based on the stem cut. The first inspector classified 12 mushrooms as good and 13 mushrooms as

bad. The second inspector classified 3 mushrooms as good and 22 mushrooms as bad (Heinemann *et al*, 1994). A second study conducted by Paulus, Busscher and Schrevens (1997) used image analysis to investigate the human efficiency of the classification of apples. The goal of the study was to investigate the inconsistency of human classification by developing an image analysis system that could simulate human classification. Based on the system, it was shown that the inconsistency of human classification of apples between two grading sessions was 35.2 percent (Paulus *et al*, 1997). While it is possible to have people grade agricultural commodities for sale, these two studies illustrate that when relying on human classification of agricultural products based on appearance; the difference between inspectors can be significant. This difference could result in produce, with varying physical appearances, being graded into the same category. A machine vision system would eliminate this difference and act as if there was only one inspector, who graded the products, the same every time.

1.4 Use of Machine Vision to classify agricultural products

Machine vision is the use of a computer to analyze a picture in order to extract meaningful information out of the picture. Using this powerful tool, accurate information, such as the images shape, size or appearance, can be obtained from an object that could not be easily obtained by human observation. To better classify the shape and appearance of agricultural products several studies have looked into using machine vision to classify various agricultural products. These include studies by Nielsen et al 1998, Paulus et al 1997, and Heinemann et al 1994.

1.4.1 Machine Vision

The acquisition of an image that is both focused and illuminated is one of the most important parts of any machine vision system. **Figure 1-2** shows the general steps required in obtaining results from an image of an object.

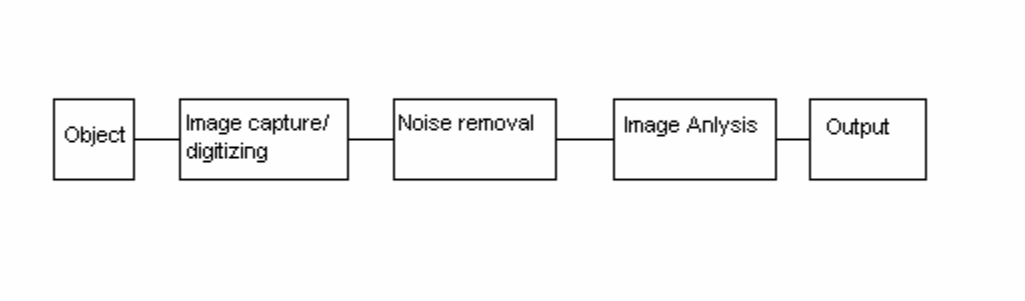


Figure 1-2: Imaging flowchart

Originally the image capture and digitizing of the image was accomplished by using a combination of a video camera and a frame grabber program (Russ, 2002). This method has almost been entirely replaced with CCD (charge-coupled device) and CMOS (complementary metal oxide semiconductor) chips. These chips use electrical circuits to directly convert light intensities into a digital image. They combine the video camera and frame grabber into one tool that can operate faster and with less distortion of the image. These chips also have the advantage that they can produce images at a much higher resolution then the frame grabber method (Russ, 2002).

Noise is the incorrect representation of a pixel inside an image. It is best observed in variations in the color of a uniformly colored surface. Noise can be caused by numerous electrical sources and its removal is important, since the noise can cause the features of an image to appear distorted. This distortion can cause the features to be measured and

classified incorrectly. While many algorithms have been proven useful to remove noise, the simplest method is to take multiple images of the same object and averaging the images together (Russ, 2002). Since the noise is not the same in every image, when averaged the noise will blend into its surroundings, making the resulting image much clearer. Preprocessing of an image can include thresholding, cropping, gradient analysis, and many more algorithms (Russ, 2002). All of the processes permanently change the pixel values inside an image so that it can be analyzed by a computer. For example, in grey scale thresholding, a value for the intensity is selected and any pixel whose intensity value is less then the selected value intensity is set to 0 (black), if greater the value is set to 255 (white). After thresholding, the resulting grey scale image can easily have features classified and measured. The outputs from a machine vision system can be varied, in robotics the output might represent the location of a object to be moved, in inspections the output would be a pass or fall result, or in the case of this study the output is the sweetpotatoes size and shape.

1.4.2 Classifying Mushrooms and Tomatoes using Machine Vision

A study by Nielsen and Paul showed that tomatoes can be more accurately described by machine vision system than what is required by the USDA standards (Neilsen *et al*, 1998). A study by Heinemann et. al (1994) used machine vision to grade mushrooms. The system was trained from the results produced by one expert grader. The trained system was then shown to misclassify mushrooms on average by 20% for all the characteristics. This was significantly lower than the disagreement between two graders

which averaged 25%. The study concluded that misclassification was no worse for the trained system than between the two expert graders (Heinemann *et. al*,1998). The study showed that mushrooms can be described using a machine vision system. In summary, these studies show the power that a machine vision system can have in classifying agricultural products. Heinemann, et al. also showed that these systems are not perfect and can still misclassify a product.

1.4.3 Classifying Apples and Potatoes using Machine Vision

Paulus, Busscher and Schrevens (1997) showed that a machine vision system could be developed to describe an apple's shape and color along with describing the sources of human inconsistencies in grading; this was done by using a decision tree to model human classification. Paulus and Screvens (1999) then expanded on this line of research and looked into describing the shape of apple cultivars by using Fourier expansion. Using principal component analysis, they indicated that the first two principal components from the Fourier expansion is all that is needed to describe the overall shape of an apple. The first principle component describes the height to width ratio and the second principle component describes the conical profile of the apple. Then using the developed Fourier expansion algorithms, the 12 "ideal" apple shapes as defined by the "International Board for Plant Genetic Resources" were modeled in order to find their fourier coefficients and principal components (Paulus *et al*, 1999). The ideal apples' principal components were tested against six experimental cultivars and it was concluded that the developed system could correctly place apples into the predefined standard

shapes. The second part of the study was to determine the number of profiles required to accurately describe the apple. The conclusion was that four images at 90 degree intervals can accurately describe an apples shape (Paulus *et al*, 1999).

Tao *et. al* (1995) investigated using machine vision on grading of potatoes on the basis of shape. The shape of potatoes was determined by describing the boundary of the profile using a radius fourier transform. Using the shape information, the algorithms were tested and shown to obtain 89.2 percent correlation between human graders and the computer. The ability to implement this system into a grading application is limited, since the time required per root was shown be as high as two seconds.

In summary, these studies showed that the complex shapes of apples and potatoes can be described using machine vision. The studies also showed that improvements needed to be made in decreasing the inspection time in order to implement these processes into a commercial grading procedure.

1.5 The use of Machine Vision with Sweetpotatoes

The work with classifying sweetpotatoes using machine vision has been limited, with only two in depth studies being conducted. Both of the studies showed that the shape of a sweetpotato could be classified in a quantitative manner using machine vision. The first study by Wooten *et. al* (2000) looked into developing a yield and quality system that would be used when harvesting sweetpotatoes. This study used a common machine vision technique called segmentation, to separate sweetpotatoes and classify sweetpotatoes into grades as they traveled along a sweetpotato harvester. The system

classified sweetpotatoes into four categories depending on their size and shape. In this study shape was calculated using the polar moment of inertia. The study demonstrated that based only on size the system would place the sweetpotato in the correct grade 80 percent of the time in the worse case scenario. When shape was included the success rate would fall to 77 percent. The study also suggested that the system was more accurate if the cull and jumbo sweetpotatoes were removed prior to inspection using the machine vision algorithms (Wooten *et al*, 2000). The second study conducted by Wright, Tappan and Sistler (1986) investigated the size and shape of sweetpotatoes. In the experiment a sweetpotato was suspended in front of an imaging device and slowly rotated in order to map its entire surface. The data was analyzed and it was suggested that the most logical geometric shape match for a sweetpotato is a prolate spheroid and that a slight curve in the major axis should be expected, based on the researchers expectations. Using the equations for volume and surface area of a prolate spheroid the corresponding surface area and volume were calculated. It was shown that the percent error in calculating both surface area and volume of a sweetpotato when compared to the measured surface area and volume was below six percent (Wright et al., 1986). This study indicated that the prolate spheroid is a reasonable choice of a shape to represent the shape of a sweetpotato.

1.6 Objectives of this thesis

Consumers want produce to be of similar size and shape. They want the produce to not only to look similar at the time of purchase, but from purchase to purchase at different time periods (Will *et al.* 1998). Previous research in describing sweetpotatoes

has been left to human interpretation, based on a set of standards (CIP, 1981 and Jones *et al.,* 1969). Studies conducted for various agriculture crops, have demonstrated that using machine vision can produce more accurate and uniform classification of the products (Paules *et al.,* 1997 and Heinemann *et al.,* 1998). Several studies using machine vision have been conducted on sweetpotatoes(Wooten *et al,*. 2000 and Wright *et al,*. 1986); however none of these studies have produce a process to quantitatively measure the size and shape of sweetpotatoes. Therefore, the goal of this project is to develop a machine vision system that can accurately describe the visual physical characteristics of a sweetpotato and create a numerical identifier that will contain all the characteristics in a single sequence of numbers. The physical characteristics to be measured include the sweetpotato's width, length, length to width ratio and straightness. The description of these characteristics inside a single number will allow for easier reporting of the sweetpotatoes appearance instead of five individual numbers. This system will give researchers a quantitative way a measuring the physical attributes of sweetpotatoes in a way not available before. In order to accomplish this study the work was broken down into three parts.

1. A hardware platform was developed to rotate the roots in a uniform manner. The findings of Paulus *et al*, 1999, Wright et al., 1986 and Wooten *et al*, 2000 were used as a guide in developing the system.
2. The use of an imaging device and lighting scheme was developed in order obtain the best possible image.
3. Algorithms were developed to control the system and describe the physical characteristics of sweetpotatoes.

2. Materials and Methods

This section discusses all of the components that were developed in order to achieve the goals stated in the introduction. This section covers the design of the hardware platform; the selection of the viewing device, and the development of all of the algorithms used to measure the physical characteristics.

2.1 Hardware

A steady base was designed to provide a location that a viewing platform could be attached as well as a location for the imaging device and lighting. The base was constructed of 1.5 inch aluminum square tubing and 1.5 inch aluminum angle. The base was built 42 inches high. The 1.5 inch angle was placed 7.5 inches down from the top and 12 inches up from the bottom. The aluminum angle lengths were cut so that the rectangular base would have an outside diameter of 23.875 X 27.375. When assembled the aluminum angle was positioned so that there was a flat base that viewing platform could be mounted onto. Last, ½ inch plywood was mounted around the base in order to block out the majority of the light and provide a location to mount lights and the imaging device to. The plywood was mounted flush with the bottom of the 1.5 inch angle and to a height of 18 inches above the top of the base. The base was made rectangular to provide a long dimension that would provide enough distance to obtain the correct field of view and a short dimension to allow for easy placement of sweetpotatoes onto the viewing platform. **Figure 2-1** shows the completed base, along with the position of the viewing platform, imaging device and lighting.

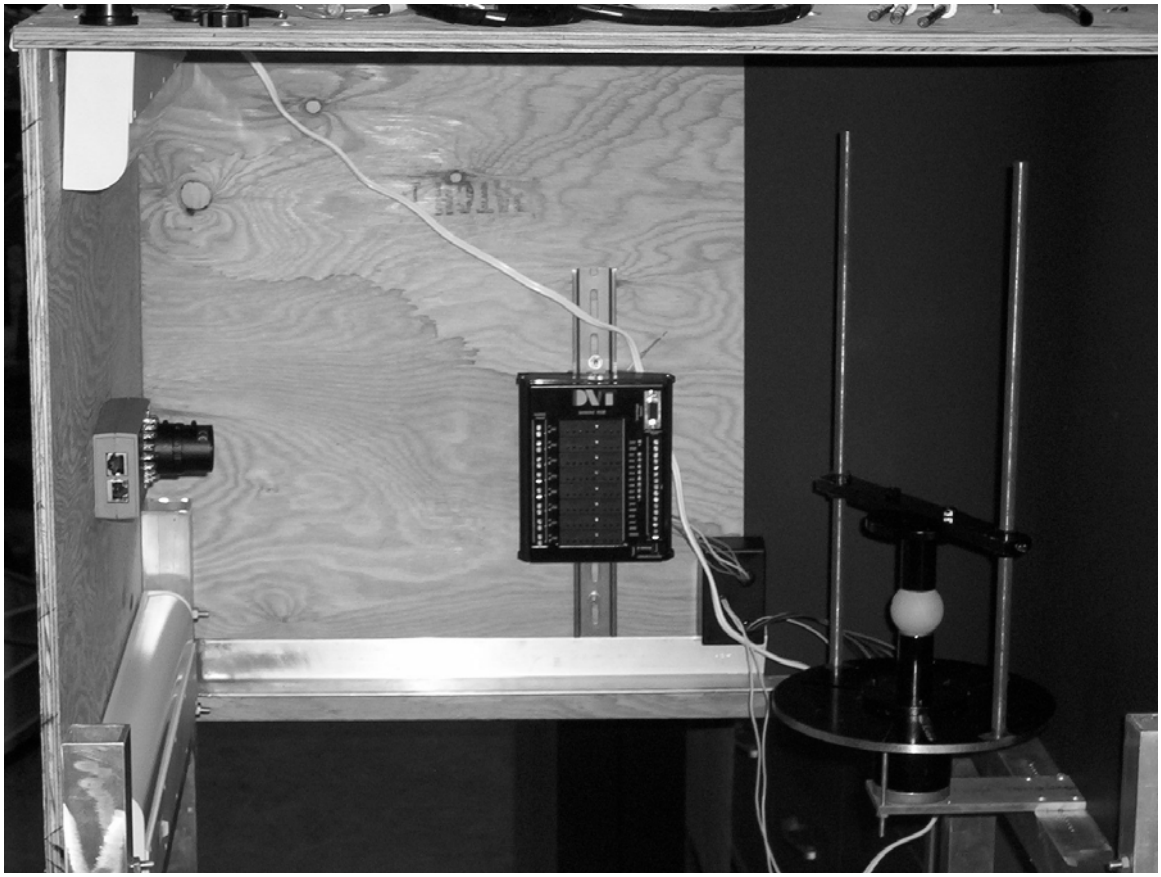


Figure 2-1: Picture of Completed Base

2.1.1 Viewing Platform

The viewing platform was responsible for holding the sweetpotato roots in the correct orientation and to slowly rotate at a constant angular velocity. The rotation of the roots is necessary, since Paulus *et al* (1999) showed that at least four views are required to accurately describe the shape of an apple. Since both apples and sweetpotatoes have irregular shapes, it was reasonable to assume that at least four views are required to accurately describe a sweetpotato's characteristics. In order to calculate

the straightness of the sweetpotato, it was important that the angle the root has been rotated can be accurately measured for each root.

The orientation of the sweetpotatoes inside the system was setup so that a line drawn through the two ends of the sweetpotato would be parallel to the x-axis in the produced images. The orientation of the sweetpotatoes is illustrated in **Figure 2-2.** The z-axis comes straight out of the image and the grey dots represent the locations of edge points found by the imaging device.

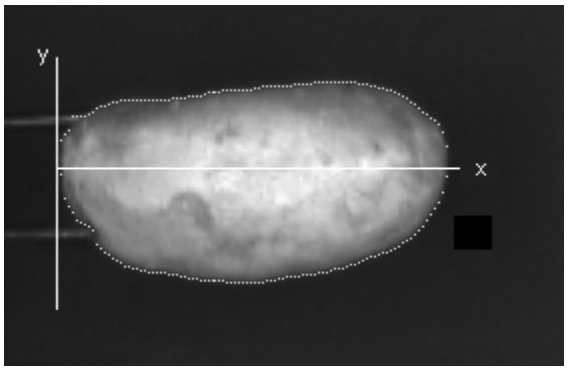


Figure 2-2 Orientation of the system

The original design had two power rollers positioned next to each other so the sweetpotatoes would constantly rotate during the inspection process. A potentiometer was attached to the power supply to control the speed at which the rollers would rotate. It was originally assumed that the sweetpotatoes would become positioned and rotate, at the same velocity of the rollers, and these rollers would provide the steady base. Several trials were run with sweetpotatoes on this setup and it was found that angle the sweetpotatoes has rotated could not be determined. This was due to the irregular shape of

sweetpotato roots, which caused sweetpotatoes of different shapes to rotate at different rates.

To solve this problem, it was decided that a stepper motor with an attached platform would provide the constant rotation necessary to make the required measurements. The ability of the stepper motor to turn a set number of degrees provided an extra benefit, in that the motion could be stopped after the required rotation had been achieved. This allowed the imaging device to capture an image of a still object instead of a rotating object, which provided an image that was less blurry.

An Applied Motion unipolar 5V with 200 steps per revolution was used in the system. This stepper motor provided an angular resolution of 1.8 degrees per step. The first attempt at controlling the stepper motor was to use a computer's parallel port to trigger a circuit that would rotate the stepper. Due to the low amount of power that a parallel port can source, this approach was unable to consistently operate a circuit that would operate the stepper motor. It was then decided that the IO ports from the imaging device could be used to control the motor. The following circuit, (**figure 2-3)**, was designed and implemented to use the IO ports from the imagine device.

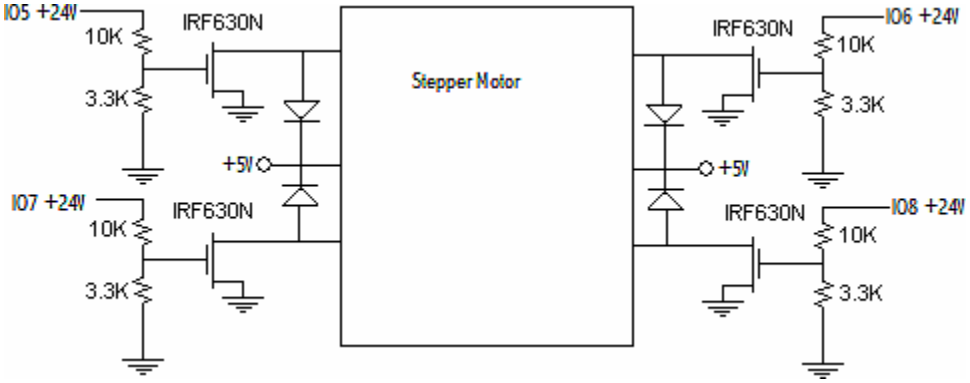


Figure 2-3: Stepper Motor Control Circuit

This circuit works by setting in sequential order the corresponding IO line, "high", while setting the other IO ports low. Using the equation for a voltage divider as shown below, the resister values was chosen so that the voltage supplied to the MOSFETs (Metal-Oxide-Semiconductor Field Effect Transistors) equaled 6 volts.

$$V_2 = \frac{R_2}{R_1 + R_2} V_1 \quad \text{..... (1)}$$

V_2= output voltage
V_1= input voltage = 24 volts
R_2= 3.3 KΩ
R_1= 10 KΩ

The reduction from 24 volts to six volts was necessary to insure there was no damage to the power MOSFETS. When the six volts are applied to the gate of the MOSFET, it allows the current to flow through the corresponding coil of the stepper motor. When the IO lines are run in succession, the stepper motor will rotate 1.8 degrees every time an IO line is set high. The maximum speed at which the stepper motor can turn without a load attached is determined by the turn-off time of the MOSFETS, which is 27 nanoseconds.

Once the stepper motor was operating a viewing platform that would be placed on the shaft of the stepper motor was designed and built. This viewing platform needed to

hold the sweetpotatoes in the correct orientation, during the entire inspection process. The first viewing platform that was designed and built consisted of an 8 inch aluminum disk with three slots cut 120 degree apart. Each slot is 3 inches long and a quarter inch wide. The slots provided a location that three pegs can be positioned. Properly positioned, these pegs held the sweetpotatoes upright and in the center of the disk. A hole was bored in the center of the disk and a collar was inserted into the hole which allowed the platform to be secured to the shaft of the stepper motor. The disk was made from aluminum to help keep the weight of the platform at a minimum. This helped to ensure that the disk's rotational inertia would not exceed the holding torque of the motor, which would cause the motor to skip a position and lose track of the angle the sweetpotato had been rotated. With the addition of the platform the maximum speed at which the motor could operate was reduced. Additionally, the maximum speed at which to motor could now be operated was reduced since at high speed the sweetpotatoes often fell off the platform. This disk was modified into a mandrel, when it was realized that is was difficult to ensure that the sweetpotatoes were positioned correctly, so that the two tips produced a line that was parallel to the y-axis in the images, during the entire inspection process. The mandrel worked by using collars, with an inside diameter of 1/2", to hold the ends of the sweetpotatoes in the correct position. The bottom collar was attached to the original plate by using the three slots. The top collar was attached to a bar that could have it's height above the plate adjusted and then secured by a thumb screw. The bar was allowed to slide on two 16 inch long 0.375" round rods, that were screwed into the plated. By using the mandrel the correct orientation of the sweetpotatoes could be maintained throughout the

entire inspection process. **Figure 2-4,** below, is an image of the completed viewing platform.

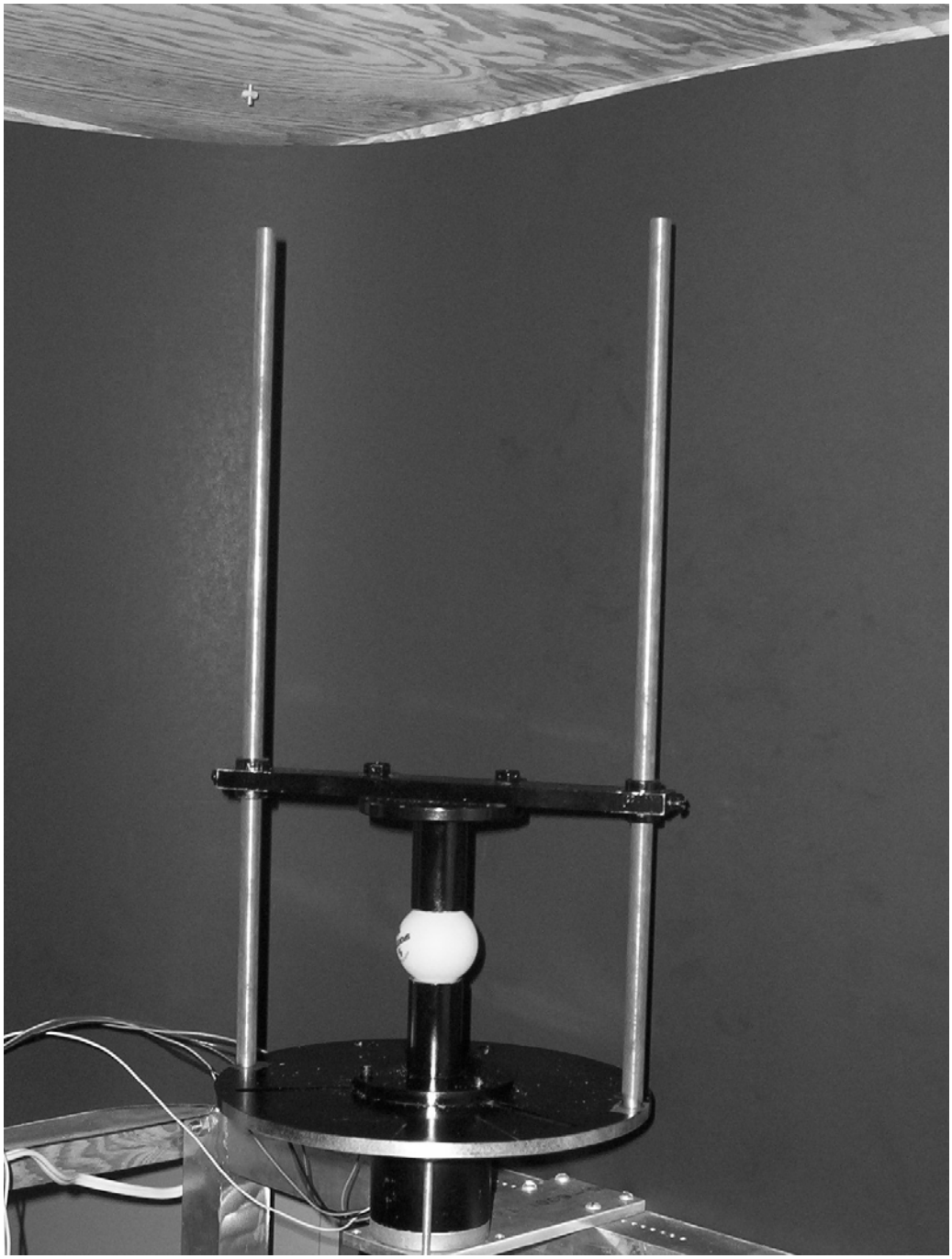


Figure 2-4: Image of Viewing Platform

2.1.2 Imaging Device

The imaging device that was selected for this project was a DVT 542c Smartimage sensor (DVT). The 542c contains a 1/4" color CCD (charged-coupled device) chip, with a 640 by 480 pixel resolution and contains a ring of 24 integrated white LEDs to provide illumination. All DVT Smartimage sensors have a CS lens mount; are powered by a 24V power supply and have both TCP/IP and serial communications. DVT provides its own operating software, Framework, to control all its sensors. Framework provides numerous preprogrammed image analysis algorithms and the ability for additional algorithms, called scripts, to be developed and implemented on the 542c using C++. Since the main purpose of any DVT system, is to be used in industrial inspection, all of the preprogrammed DVT algorithms have been design for speed and efficiency. This allows for quicker implementation since algorithms to remove noise, detect edges and many more common machine vision processes do not need to be written and optimized.

Using equation 2, a lens was selected to be used on the 542c. Knowing that the 542c has a CCD size of ¼", the FOV needed to be 15" and the working distance could be estimated at 20". The focal length was determined to be 4mm.

$$\frac{CCD}{FL} = \frac{FOV}{WD} \quad \text{..... (2)}$$

Where: CCD= CCD size
FL=Focal length
FOV= Field of View
WD=Working Distance

A Tamron CS mount 3-8mm zoom lens was selected for the system. The adjustable focal length provides for adjustments in obtaining the proper focus for the system. Since the lens is a CS mount it could be attached to the 542c Smartimage sensor without any additional spacers. **Figure 2-5,** shows the 542c Smartimage sensor and lens attached to the plywood base. The black wire on the left provides the 24V supply and serial communications. The white wire provides the TCP/IP communications. The 542c was position in the center of one of the plywood sides facing the opposite 23.875 side. The Smartimage sensor was placed so that the CCD chip was positioned so that the resulting images are oriented in such a fashion that the y-axis will run parallel to the minor axis of the sweetpotato and the x-axis will run parallel to the major axis.



Figure 2-5: 542c Smartimage Sensor and Tamron lens

To provide lighting, two 18" fluorescent tubes were used. The fluorescent tubes, were placed at 6 inches above and below the Smartimage sensor. The placements of these

tubes were determined by experimentation so that the position of the lights provides an image that is both well illuminated and free from shadows.

2.2 Software

Once the hardware had been designed and built, software was written to control the system and describe the physical characteristics of the sweetpotato. On power up, the entire system is initialized. This initialization step powers up the camera and rotates the stepper motor 4 steps. Since it is possible to rotate the platform freely when the power is not connected to the motor, it is possible that the motor could be in aligned with any of the four possible steps. Cycling through the four steps initially ensures that the motor will rotate immediately when the inspection program is run.

The inspection program uses both Labview and DVT's Framework to inspect and determine the physical characteristics of the sweetpotato root. The Labview program is setup as the user interface and its front panel is illustrated in **Figure 2-6.**

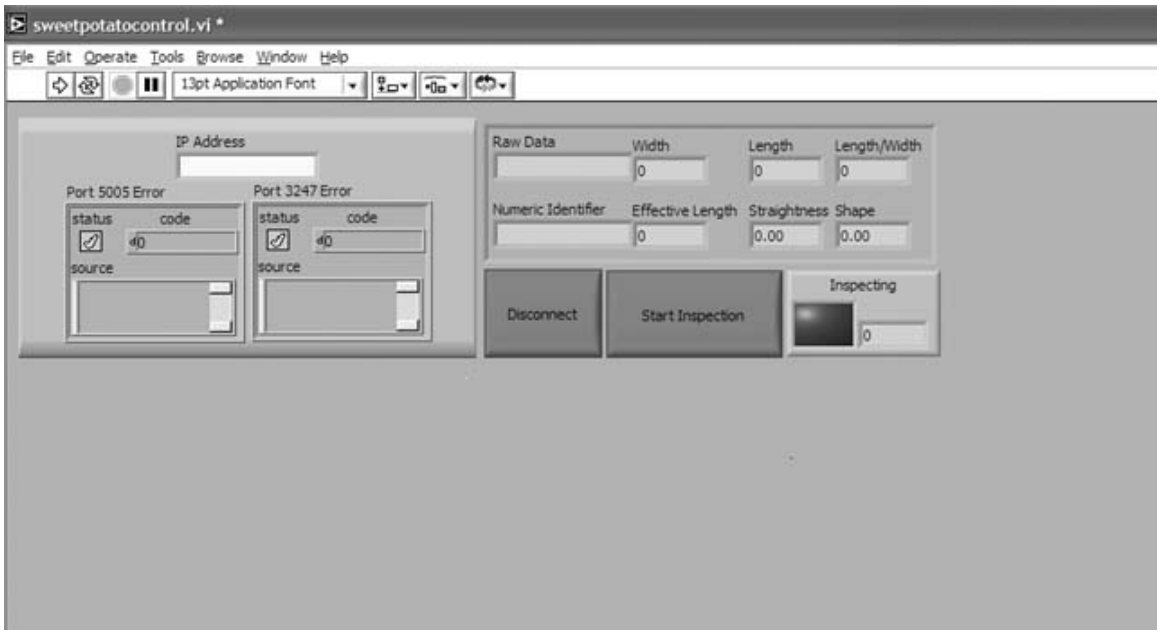


Figure 2-6: Front Panel from Labview Program

The interface allowes the user to input the IP address of the 542c Smartimage sensor. It also allows the user to control when an inspection is started along with ending the inspections, which will close all communications between Labview and the 542c Smartimage sensor. A green led was placed in the lower right corner to allow the user to see when the program is inspecting. All the measured physical characteristics along with the numerical identifier are displayed in the upper right hand side.

Once the start inspection button is pressed, the program enters the inspection algorithm, shown in **Figure 2-7**. The 542c Smartimage sensor is programmed to run the inspections of the root and will be explained in detailed in **section 2.2.2**.

```
{-------------------------------------------------------------------------------}
If start inspection                    // waits until the start inspection button is pressed
{
for image=0 to 6                       // cycle through required images
{
Inspect root                           // runs the algorithms for all characteristics
Rotate stepper motor 36 degrees        // turn stepper motor for next inspection
If image=6 -- report characteristics and numerical identifier
Else – report busy                     // allows user to monitor progress of system
}
}
{-------------------------------------------------------------------}
```

Figure 2-7: Inspection algorithm

The 6 images at 36 degree intervals were chosen because that number provides the smallest number of images above four that had a uniform number of degrees between images that could be obtained using this stepper motor. Four images at 90 degee intervals were shown by Paulus *et al*, 1999 as the minimum number of profiles needed to describe the shape of an object. A 36 degree interval was chosen based on the 1.8 degrees per step resolution of the stepper motor. 36 degrees is achieved by stepping through each of the four steps, five times. Once the inspection is completed the results are shown both on the front panel of Labview and saved inside a user defined document. This inspection program controls both the stepper motor and the inspection of the roots, whose algorithms will be describe in the next two sections.

2.2.1 Stepper Motor Control

The control of the stepper motor was achieved by a combination of Labview and a DVT background script. When the inspection algorithm is triggered by the start inspection button, the string “trigger” followed by an end on line character is sent to the DVT background script from Labview. The string is sent via a TCP protocol that is setup between port 5005 of the computer and the 542c Smartimage sensor. The background script, then triggers the inspection of the root. When the inspection is completed, IO lines 5-8 turned on and off in order five times. IO line eight is left on after the 5^{th} cycle to both stop the motion of the disk which allows for a cleaner image to be taken and to hold the disk in place during the times that the sweetpotatoes are switched. After the 5^{th} cycle the background script waits until Labview sends the start inspection string again before

triggering the IO lines. During the entire inspection process the string is sent from Labview to the background script a total of 6 times, once for each image.

2.2.2 Physical Characteristics

A total of four physical characteristics were measured in this study. These characteristics are the sweetpotato's width, length, length to width ratio and straightness. Each of these characteristics contains their own unique algorithms to compute its characteristic. Each characteristic is converted from number of pixels to millimeters of length by multiplying the characteristic by a scale factor. The scale factor of .61 represents the physical size in millimeters that one pixel represents in the image. After the sixth image is taken and inspected, all of the physical characteristics are computed and sent to Labview via a TCP protocol set up between port 3247 on the computer and the 542c Smartimage sensor. Labview takes this raw data and converts it into the numerical identifier. The physical characteristics along with the numerical identifier are then reported in the front panel of LabView, (**Figure 2-6)**. All of the DVT script algorithms are Labview program are located in Appendix B.

2.2.2.1 Width

Width has traditionally been one of the characteristics that could be measured without the aide of machine vision system. A pair of calipers for example, is all that is needed to measure the width. In this study, width is the only physical characteristic that is

calculated by directly using a predefined DVT Softsensor. This characteristic is measured using the Measure Across the Area Softsensor. A rectangular area inside the image was setup from x equal to 9 to 200 and y equal to 39 to56. These numbers represent the x and y pixel locations inside the image. **Figure 2-8**, illustrates the width algorithm. The Softsensor is then set so that there is a scan line running parallel to the y-axis every other pixel in the x direction. The scan lines work by scanning from the outside of the box to the center of the box. The scan lines are set up to record the y position when a pixel's intensity goes above a threshold value; with each scan line reporting two edge locations, an upper location and a lower location. Since the sweetpotatoes being measured are smaller than the area being scanned, any scan lines that do not report an edge pixel are ignored. The change in intensity represents the scan line passing over an edge.



Figure 2-8: Illustration of Width Algorithm

The light grey pixels that surround the sweetpotato represent the location of the edge pixels. The edge pixels were found by setting the cutoff value for scan lines at 22 and conducting an open and close algorithm supplied by DVT. The open algorithm is a machine vision process that erodes the object by set number of pixels, set at ten for this system, and then dilates the object by the same number of image. The close algorithm works in the same way expect it dilates and then erodes the object. When used in conjunction these two algorithms help to eliminate any stray edge pixels that might have be found on the mandrel used to hold the roots. The distance between the two edge points on each scan line is calculated by the Softsensor. The maximum distance was then reported as the width for each image. The largest width found from any of six images is reported as the sweetpotato's width in the raw data, which is sent to labview. This process is the same as taking a caliper measurement every two pixels across the entire length of the sweetpotato and recording the largest value obtained.

2.2.2.2 Length

Length is another characteristic that can be measured accurately without the aide of machine vision. For this study, length was measured using the data collected from the width algorithm. Using a DVT script Softsensor, length was defined as the x distance between the first and last significant scan line. A significant scan line is any scan line that records an edge. The largest length from all images is recorded as the sweetpotatoes

length. The length that is being reported is shorter then the true length of the sweetpotato. This is a result from the ends of the sweetpotato are hidden for view inside the two collars used to hold the sweetpotato. However, it was assumed that ends that were being hidden from view would not be of much use commercially, since they would have a diameter smaller then 1/2'' (12.7 mm). And the length that was being reported by the system was a better representation of the sweetpotato's length than just measuring tip to tip. **Figure 2-9**, illustrates the output from the length algorithm. The grey line the runs through the sweetpotato, illustrates the length that was measured by the length algorithm.



Figure 2-9: Illustration of Length Algorithm

2.2.2.4 Length per Width Ratio

This characteristic is the only characteristic not computed using the 542c Smartimage sensor. The length to width ratio is determined by dividing the reported length from the Smartimage sensor by the reported width inside the Labview program. This is important characteristic, since it provides a good gauge on the elongation of the root. A small ratio would indicated that the sweetpotato is round, while a large ratio would indicate a long narrow sweetpotato.

2.2.2.5 Straightness

The final characteristic this study will describe is straightness. Straightness is defined as how straight the sweetpotato is along its long axis. This important feature, which cannot be measured accurately or as fast by hand, has typically been included in the shape classifications determined by CIP 1981, Jones *et al*., 1969, and Martin *et al*, 1983. The value of the straightness factor should equal zero when the sweetpotato is perfectly straight and it should increase as the sweetpotato shape becomes more curved. The expected number for the straightness of an ideal root should be somewhat above zero, since Wright et al, 1986 showed that a slight curve in a sweetpotato's shape should be expected.

The straightness is computed by using the scan lines used in the width Softsensor to break the sweetpotato up into uniform sections that have a diameter equal to the distance between the edge points and a width of 1.5 pixels. The center of mass for each section is then used to compute the straightness factor. The next step in the straightness algorithm is to compute the y and z coordinates of the center of mass for each section. The x-coordinates of the center of mass for each section represent the significant scanline that produced that section. Equation five was used to calculate the y and z coordinates of the center of mass.

$$t = \frac{\sum t'}{N} \quad \text{..... (5)}$$

Where: t is the coordinate of the center of mass being calculated
t′ is the coordinate of the edge point
N is the total number of edge points

In computing the centers of mass for each section, the first step is to move the origin so that the x-axis runs through a center of the sweetpotato root. The center of the sweetpotato, y_c', is determined by averaging the two y′ values, of the edge points, in the first scan line. The y′ stands for the y value measured by the 542c Smartimage sensor. The first scan line was arbitrarily selected since the sweetpotato is placed on the viewing platform vertical making the center axis of the viewing platform orthogonal to the middle of the root. After y_c' is computed, two radii are computed from y_c' to the y′ position of the two edge points for each section. These two radii represent the y′ distance from the center, y_c', to the y′ position of the edge. Using the two radii as hypotenuses of two right triangles and knowing the angle that the root has turned the y and z coordinates for every

edge point discovered by the width algorithm can be computed. Equations six and seven are the equations used to compute the y and z values for each scan line.

$$y = radius_{upper} * \cos(\theta) + radius_{lower} \cos(\theta) \quad \text{..... (6)}$$

$$z = radius_{upper} * \sin(\theta) + radius_{lower} \sin(\theta) \quad \text{..... (7)}$$

Where: θ equals the angle the root has been rotated.
$Radius_{upper}$= y′ distance from center to upper edge point
$Radius_{lower}$= y′ distance from center to lower edge point

The new x,y,z coordinate system were setup to match the coordinate system reported by the 542c Smartimage sensor to avoid confusion. The y direction wass set to travel vertically through the image and the z direction is then setup to come out of the image at orthogonal to both the x and y axis, as illustrated in **figure 2-11**. **Equation 6 and 7** sum the two radii on each scan line for that image. After all y and z values have been computed for every section in the image. The process of computing the y and z values is repeated for each of the 5 remaining images. The y and z values from all six image are then summed together from each section. The resulting sums result in the numerator of **equation 5** for both the y and z directions. Since six images are being summed together the total number of edge points for each section equals 12. Using this information and **equation 5** the y and z coordinates for the centers of mass for each section can then be determined. The locations of computed the centers of mass for the straightness algorithm is illustrated in **Figure 2-10.** The top curved line represents the location of the z center of mass. The top straight line represents the location of the z center of mass if the sweetpotato has no bend. The bottom curved lines illustrates the positions of the y centers of mass. The bottom straight line represents where the y center of mass would be if there was no bend in the sweetpotato.



Figure 2-10: Illustration of Computed Centers of Mass

Using the centers of mass computed by the routine above the straightness factor can now be calculated. Starting with the second significant scan line, the differences between the coordinates of the center of mass between the current scan line and the one prior are calculated and then squared. Since only the amount of bend is being investigated the squaring the results allow for negative values to be eliminated. The squaring also protects against the case of a "S" shaped sweetpotato, whose straightness would be reported as zero since it would have equal positive and negative differences. Only the y and z coordinates are used. The x-distance is ignored, since it has nothing to do with the straightness of the sweetpotato. This process is repeated with each squared difference

being added together, until the difference between last significant scan line and its predecessor is calculated. The total change in both the y and z directions is then added together. This resulting sum has its square root calculated to nullify the squares being calculated previously and is reported as the straightness factor. The square root was calculated to help reduce the size of the number caused by squaring all of the numbers.

2.2.3 Numerical Identifier

The numerical identifier is used to combine the results from all the physical characteristics into one 15 digit code. Since the numerical identifier contains six physical characteristics that have been measured quantitatively, the numerical identifier will be a more accurate way of describing sweetpotatoes than the processes described by CIP 1981, Jones *et al*., 1969, and Martin *et al*, 1983. This numerical identifier is produced by a combination of two DVT scripts and Labview. The first DVT script receives data from the width, length and effective length factors after each image is analyzed. This script compares the results from each image and reports the largest length and width found. The second background script reports the straightness and shape factors it receives after the sixth image is analyzed. These two scripts are combined into a single string of data that contains all the characteristics, with each characteristic separated by a comma. This string is then sent via a DVT application call Datalink to Labview. Datalink sets up the TCP protocol between Labview and the 542c Smartimage sensor. As Labview receives the data it breaks the string apart into the characteristics. Wooten *et al* in 2000 reported that as the size of a sweetpotato increases the polar moment of inertia increases. It is

justifiable to believe that both the shape and straightness factor computed would behave in a similar fashion. To solve this problem the straightness number was normalized by dividing both characteristics by the sweetpotato's length. This results in straightness to be reported on a per millimeter of length basis and allows for comparison straightness between roots that are not of the same size. The modification of straightness is acceptable, since only the magnitude is needed for the comparison to other sweetpotatoes, is needed. All of the results are converted into an integer value to get rid of the decimal places, except for straightness which remains a floating point number with three numbers following the decimal place. The new results for length to width ratio and straightness are then combines in Labveiw into the 10 digit numerical identifier, illustrated in **Figure 2-11.**

W	**W**	**L**	**L**	**L**	**R**	**ST**	**.**	**ST**	**ST**	**ST**

W- Width R= Ratio of Length to Width
L= Length ST= Straightness

Figure 2-11: Location of Characteristics inside the Numerical Identifier

3. Results and Discussion

This section provides the results from the inspections of five standard shaped objects and ten sweetpotato roots. The five standard objects include three round spheres of diameters of 37, 62, and 65 mm respectively and two different sized mini footballs. The sweetpotato roots were purchased on the same day at a local supermarket. Each object was inspected 30 times to produce a set of thirty numerical identifiers. The results were then averaged to produce the arithmetic average measured value for each characteristic. These results were analyzed to illustrate the effectiveness of the developed system and algorithms.

3.1 Standard Objects

The five standard shaped objects were selected to illustrate the ability of the system to produce accurate results. The three spheres, with a sphere being a prolate spheroid with equal major and minor axis, were chosen to demonstrate that the system would produce a straightness factor equal to zero. The footballs were chosen since they have a shape that is similar to a sweetpotato and have a straight major axis. The following chart (**Table 3-1**) illustrates the arithmetic average of the characteristics for the standard shaped objects.

	Width (mm)	Length (mm)	Length/Width	Straightness (per mm of length)
Ping Pong Ball	35.27	30.2	0.9	0.0696
Wooden Ball	61.00	56.8	0.9	0.0539
Tennis Ball	65.63	60.0	0.9	0.1678
Football 1	75.20	123.0	1.6	0.0821
Football 2	88.73	126.0	1.4	0.0701

Table 3-1: Results of Five standard objects

All three spheres and the two footballs had their widths measured within two millimeters of their actual diameters. This small amount of error is the result of the method of finding the edge pixels inside the width algorithm. The width algorithm was set up to detect an edge pixel when the normalized intensity reached 20. This results on edge pixels being placed on the inside of the edge, instead of the outside. Doing this, resulted in the widths being reported 2 pixels or 1.22 millimeters small. The algorithm could have been set up to report the edge pixels on the outside of the edge; however this would have resulted in the widths being 1.22 millimeters to large. The rest of the two millimeters of error is the result of the placement of the scan lines at every other x value. This could result scan line not running along the true maximum width; thus reporting a width slightly smaller the actual width is possible. The lengths for the three spheres were reported to be approximately five millimeters smaller then the widths. This was expected since the length algorithm was developed to report the length from that the object has a diameter greater then 1/2". The lengths reported for the footballs were both approximately ten millimeters smaller then the measured lengths. This larger difference for the footballs is the result of the conical ends of the footballs fitting farther into the collars of the mandrel. The ratios of length to width for all three spheres were correctly

measured. The straightness factors for the ping-pong ball, wooden ball and tennis ball was measured at 0.0696, 0.0539, and 0.1678 respectfully. The small difference from zero, for the ping pong ball and the wooden ball is the result of rounding during the computation of the straightness factor. The large value for the tennis ball can be attributed to the black lettering on the ball. This lettering only occurs on one side of the ball could, on some views, cause an incorrect reporting of the edge pixels. These incorrect edge pixels would then cause the centroids to be calculated incorrectly. The two footballs also small straightness factors of 0.0821and 0.0701 respectively. These results were slightly larger then the straightness numbers for the ping pong ball and the wooden ball. This small difference is believed to be caused by the plastic laces on the footballs showing up in a view.

Overall the results from the standard objects showed that the system would measure all of the physical characteristics and produce results that were reasonable. The width for all of the standard objects measured by the system agreed within 2mm of the width measured by hand. The measured lengths were smaller then the lengths measured by hand. This was expected, since the length was set up to measure a length shorter then the actual length. The straightness numbers produced by the system agreed with what was expected. The tennis ball showed that discoloration can cause results for shape and straight to diverge from what is expected, by producing incorrect locations for edge pixels.

It is important to comment on how uniform the results for the physical characteristics were. **Table 3-2,** below, provides the standard deviations for each of the standard objects for the 30 inspections.

	Width	Length	Length/Width	Straightness
Ping Pong Ball	0.450	0.551	0.017	0.018
Wooden Ball	0.000	0.551	0.009	0.005
Tennis Ball	0.490	0.000	0.007	0.012
Football 1	0.407	0.000	0.009	0.006
Football 2	0.583	0.000	0.009	0.004

Table 3-2 Standard Deviations for the Standard Objects

For all of the objects the largest standard deviation for the width or length was 0.583 millimeters. The uniformity of the results suggested that the system will produce an accurate measurement of symmetrical object every time it is inspected, regardless of its initial position in the system. If the standard deviations have had been larger, it would have suggested that position of the object or the viewing platform had some affect on the measurement of the characteristics. The standard deviations for straightness was largest for the ping pong ball, this is believed to be the result of the small size of the ping pong ball. The tennis ball also had a large standard deviation when compared to the wooden ball or either of the footballs. The large standard deviation was due to the black lettering on the tennis ball. The reasoning that the lettering could cause the larger standard deviations was its different locations in different inspections, in some inspections the lettering fell on the edge of the profile causing larger error while during others it did not. The variation in the results footballs and the wooden ball are believed to be mainly caused by rounding errors, however part of the variation in the results is believed to be caused by the mandrel used to hold the objects blocking part of the profile. To get a better idea of how the error caused by the calculation of the characteristics or the mandrel used to hold the objects might have affected the results, the standard deviation for all of the

characteristics were normalized by two different methods. The normalization allows for the comparison of the deviations of objects of different sizes, to see if there was any possible correlation between the deviations. The first method for normalization was conducted by dividing the standard deviation by the measured length of the object. The second method for normalization divided the standard deviation of the characteristic by characteristic's measured value. The normalized data was placed in **Tables 3-3** and **3-4**. Since the objects being inspected are symmetrical objects, each of their six profile images should be the same regardless of the objects initial orientation. Any variation in the profiles captured by the 542c SmartImage sensor would be caused by the mandrel used to hold the objects, obstructing the view of the 542c. The mandrel had been painted black to help them blend into the background. However it was still possible that it could cause variations in the results. For the system to be functioning correctly, it is important to show that the system was not causing any uniform error that could be adjusted adding a correction factor to any of the characteristics. Correction factors could easily have been included in any of the algorithms to nullify any effects the system might have on an image.

Normalized by Length				
	Width	Length	Length/Width	Straightness
Ping Pong Ball	0.0149	0.0182	0.0006	0.0006
Wooden Ball	0.0000	0.5916	0.1674	0.0001
Tennis Ball	0.0082	0.0000	0.0409	0.0001
Football 1	0.0033	0.0000	0.1069	0.0001
Football 2	0.0046	0.0000	0.1350	0.0001

Table 3-3: Standard Deviations Normalized by the Object's Length

Normalized by Characteristic				
	Width	Length	Length/Width	Straightness
Ping Pong Ball	0.013	0.018	0.020	0.264
Wooden Ball	0.000	0.010	0.010	0.095
Tennis Ball	0.007	0.000	0.007	0.073
Football 1	0.005	0.000	0.005	0.075
Football 2	0.007	0.000	0.007	0.053

Table 3-4: Standard Deviations Normalized by the Characteristic

In both methods of normalization; the normalized standard deviations do not appear to have any pattern that would indicate any common error that is being caused by the system. This would suggest that the variations in the results are from computing the measurement or rounding the results into integers and not the mandrel used to hold the objects. This would suggest that the mandrel not causing any error in the calculations and a correction factor for any of the characteristics is not needed. The only object that appears to be affected by the system was the ping pong ball, which had a normalized standard deviation significantly larger then the others. Since, the ping pong ball is of such small size, it is logical to conclude that the any common error between to objects would appear to be magnified in an object that small. This error should not be a problem in the future since the system was not designed to handle sweetpotatoes of the same size of a ping-pong ball. These results would suggest that a correction factor is not needed for any

of these characteristics. However a correction factor of 1.22 millimeters will be added to width to correct for the under measurement of the width caused by method edge pixels are located.

Overall the system was shown to produce measurements for the width, length, ratio of length to width and straightness for all of the standard objects that are independent of the position of the mandrel or their orientation in the system. The system produced straightness factors that matched what was expected, with all the objects having very small straightness numbers. The lettering on the tennis ball showed that a black discoloration on an object could cause the system to misplace edge points, resulting in incorrect measurements for straightness.

3.2 Sweetpotatoes

After the system had been shown to measure several standard objects, ten sweetpotatoes were inspected to show that the system could measure all of the physical characteristics of sweetpotatoes. The results from the inspections will also be used to illustrate the amount of variation in the size and shape of sweetpotatoes that can be found at a local supermarket. The results from the two mini footballs will be included in the data to provide a comparison of a sweetpotato to an object with approximately the same profile.

As with the standard shapes, all ten sweetpotatoes were inspected 30 times. The results for each characteristic were then averaged together to produce an arithmetic average for each characteristic. The results for width, length and effective length are

illustrated in **Figure 3-1** and the results from shape and straightness illustrated in **Figure 3-3.**

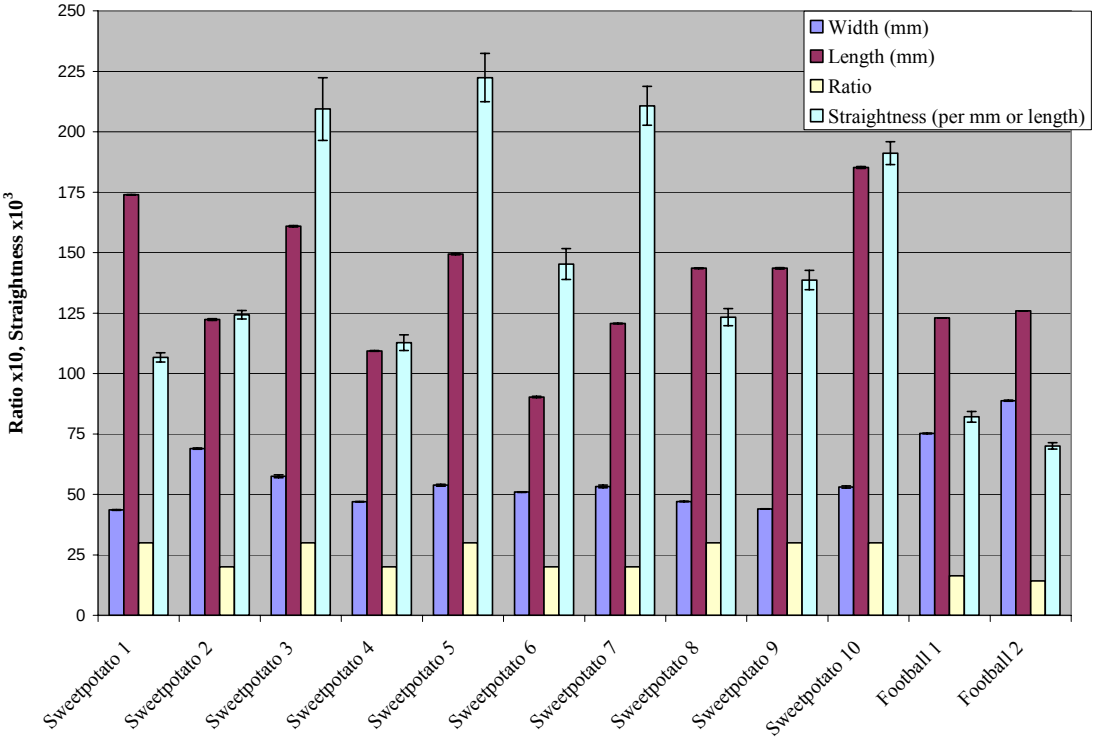


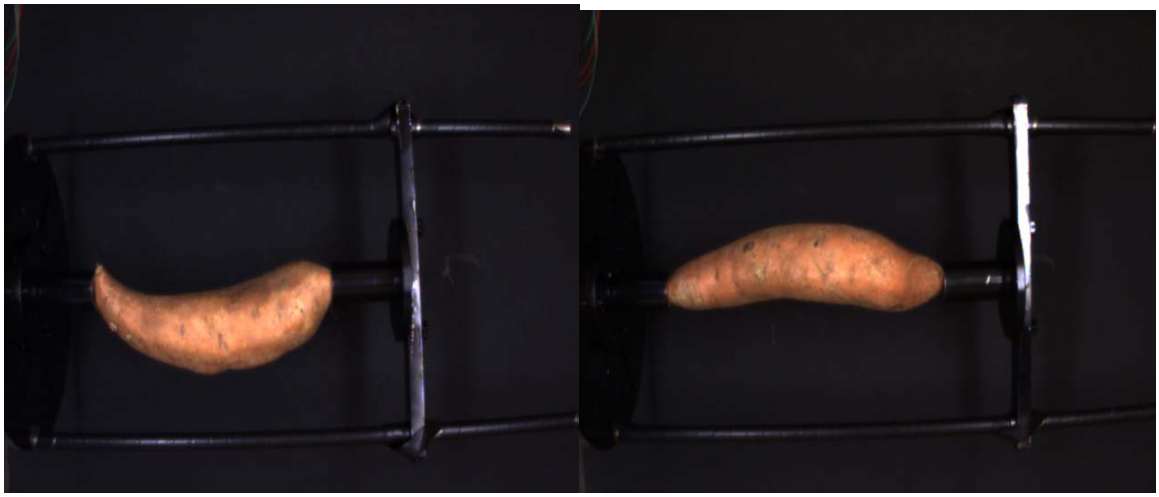
Figure 3-1: Results for Width, Length, Ratio, and Straightness of Sweetpotatoes and Mini Footballs

The system measured the average widths of the ten sweetpotatoes as 43.6, 69, 57.6, 57.5, 53.1 53.8, 51, 47.1, 44, and 53 millimeters. Using the data from all 30 inspections, 95 percent confidence intervals for each of the sweetpotatoes width were constructed, shown as the plus/minus bars on **Figure 3-1**. The produced confidence intervals did not contain any of the widths measured by hand of 45, 67, 59, 55, 54, 55, 49, 42, 48, and 56 millimeters, respectively. Since the 95% confidence intervals were of such small ranges, approximately plus/minus 0.28 millimeters on average, and the size of the standard objects in section 3.1 were correctly measured; the results illustrate that the

system can measure the correct value for width and just how difficult it is to measure the size of a sweetpotato. The lengths that were measured produced results with small 95% confidence intervals. Knowing that standard objects had their lengths measured correctly and the sweetpotatoes had small confidence intervals in the results; the data suggests that the system is measuring the correct lengths. These results demonstrated that the system can measure that width, length, and ratio of length to width of sweetpotatoes. The uniformity of the results as illustrated by small standard deviations, suggests that the same results for a sweetpotato will be obtained regardless of the original position of the sweetpotato in the system.

The straightness numbers were larger for the majority of the sweetpotatoes than the straightness numbers for the footballs. This appears to help to validate the findings of Wright et al. (1986) that showed a slight bend in the major axis of the sweetpotato. The confidence intervals for straightness were considerably larger than the confidence intervals for the other characteristics. The larger confidence interval is believed to be caused by only using six images. By only using six images and given the sweetpotato's naturally irregular shape, it is possible to not capture the same six profiles each time an inspection is run. The variation in these profiles would create varying straightness numbers and result in the larger confidence intervals. This is best illustrated by the range in the straightness confidence intervals for the footballs compared to the sweetpotatoes. The footballs, which have a uniform profile from any angle had smaller confidence intervals than the intervals for the sweetpotatoes.

The straightness number was not a sensitive to the bending of the major axis as was hoped. This is best illustrated by looking at sweetpotatoes three and ten, **Figure 3-4.**



Sweetpotato Three **Sweetpotato Ten**

Figure 3-2 Illustration of Sweetpotato Three and Ten

Visual inspection of sweetpotato three and ten would suggest that sweetpotato three should have a much larger straightness number. The results however in **Figure 3-1** would seem to indicate that the curvature of sweetpotato three was not much larger then sweetpotato ten, the 95% confidence intervals for sweetpotatoes three and ten. While the straightness number did not increase as much as was expected, it did show an increase in the result as the amount of bend increased. Overall the straightness factor did not have as large of variation as was hoped, however it did show that the amount of bend could be measured.

The observation that the system would measure the same characteristics, regardless of initial position, was tested by reinspecting all ten sweetpotatoes. For this test each sweetpotato was inverted from it's original position and tested again. The numerical averages for width, length and effective length from this test are reported in **Table 3-4,** below.

	Width (mm)		Length (mm)		Ratio		Straightness (per mm of Length)		
		Inverted		Inverted		Inverted		Inverted	p-value
Sweetpotato 1	43.6	43.50	173.9	172.7	4	4	0.1067	0.1371	0.311
Sweetpotato 2	69.0	68.73	122.3	122.1	2	2	0.1243	0.1262	0.083
Sweetpotato 3	57.5	57.10	160.9	161.0	3	3	0.2094	0.2288	0.008
Sweetpotato 4	47.0	47.67	109.4	109.7	2	2	0.1128	0.1201	0.158
Sweetpotato 5	53.8	53.80	149.4	149.3	3	3	0.2224	0.2099	0.109
Sweetpotato 6	51.0	51.63	90.3	91.5	2	2	0.1453	0.1184	1.08E-05
Sweetpotato 7	53.3	53.30	120.7	120.6	2	2	0.2107	0.1960	0.001
Sweetpotato 8	47.1	47.23	143.5	143.0	3	3	0.1233	0.1294	0.052
Sweetpotato 9	44.0	44.00	143.5	143.6	3	3	0.1386	0.1448	0.031
Sweetpotato 10	53.1	52.80	185.2	185.0	3	3	0.1911	0.1896	0.656

Table 3-5: Results from Inverting Ten Sweetpotatoes

The largest variation between the average results of width and length were observed to be approximately 0.67 millimeters for the width and 1.2 millimeters for the length. The variations accounted for only approximately 1% of the total width or length and are considered acceptable given the large variations in the size of sweetpotatoes allowed in the US No. 1 grade for sweetpotatoes. This demonstrates that the position of the sweetpotato has little effect on the measurement of its width or length. The results for the straightness number appear to be similar. As a check to see if the results for straightness were not significantly different, a paired two-sample student's t-test, with an alpha level of 0.05 and hypothesized mean difference of zero, was conducted on the straightness numbers. The p-values from this test showed that for sweetpotatoes three, six, seven and nine had means that were significantly different. While statistically significant, the variations were still inside the confidence intervals produce in **Figure 3-1.** This would suggest that the variation is a result in only using six images to describe the sweetpotato's straightness.

The results from the ten sweetpotatoes illustrate that the system will accurately measure the length, width, effective length. By inverting the sweetpotatoes the system demonstrated that these measurements can be made regardless of the sweetpotatoes position. The system was demonstrated to have the ability to quantify the sweetpotatoes' straightness. The quantification method straightness however, causes some slight variation in the results, of multiple inspections of the same sweetpotato; depending on the orientation of the system and the sweetpotato. This is believed to be primarily, the result of only using six profiles to inspect the root. The main causes of error in the calculation of length, width and ration of length to width are believed to be caused by rounding errors in the calculations of the characteristics. The findings for straightness appeared to confirm the findings of Wright et al. (1986) that a slight bend in the major axis of a sweetpotato can be expected. The results for the sweetpotatoes show that the amount of variation in shape of sweetpotatoes that are sold in grocery stores is highly variable. This is the opposite of expectations of consumers, who typically desire produce to have similar size and shape (Wills *et al*. 1998).

4. Conclusions

This study looked into the use of machine vision to produce numerical identifiers that represent several physical characteristics of sweetpotato roots. The developed system was tested using five standard objects and ten sweetpotatoes. The results demonstrated that the developed algorithms could measure the physical characteristics. The resulting system achieved all three objectives that were stated in the introduction.

1. The developed system provided a method for uniformly rotating the sweetpotatoes. The rotation was achieved by a stepper motor that is controlled by I/O ports on the DVT 542c SmartImage Sensor. The developed viewing platform provided the stable and adjustable location needed to hold the sweetpotatoes in the correct position for inspection. An aluminum frame covered by plywood was constructed to block the majority of the outside light and provide a platform to mount the hardware to.

2. A DVT 542c SmartImage sensor and a Tamron 3-8mm zoom lens were selected to acquire the images of the sweetpoatoes. The selected lens provided the 15" field of view needed to image all possible sweetpotato sizes. The 542c SmaertImage sensor was placed in the center of the plywood side opposite of the viewing platform. Two white 18" fluorescent lights were placed at 6" above and one six inches below 542c. The location for the lights provided the best illumination, with the least shadowing of the objects. The viewing platform and the plywood side behind the sweetpotatoes were colored black to provide the best image of a sweetpotato.

3. The developed algorithms were demonstrated to produce accurate numerical identifiers of the sweetpotatoes. A Labview program was developed and implemented to produce the numerical identifiers from data sent from the 542c SmartImage sensor and provided the user interface for the system. Algorithms were written for the 542c to produce measurements for all of the characteristics being measured. The width algorithm was developed using the DVT Measure across the Area Softsensor. The resulting width algorithm produced results that were accurate and uniform over thirty inspections of an object. The developed algorithm for length produced results that uniform, the accuracy of the length algorithm had the possibility to be off by a few millimeters. This inaccuracy was caused by the spacing of the scan lines used to measure the length. The straightness factor was measured by computing the change in the centers of mass of slices made in the sweetpotato by the scan lines from the width algorithm. The straightness factor was not as sensitive to bending as was hoped but it produced results that were reasonable with the curvature of the objects. Overall the entire system produced uniform results for multiple inspections on the same object, illustrated by the small standard deviations for each of the characteristics. The straightness factor had the largest variation in its results. This variability can be explained by the non uniform shape of the sweetpotatoes and only using six images to describe the profile, allowing for different profiles to be taken on different inspections. The system was tested and shown to produce the same

numerical identifiers for individual sweetpotatoes, regardless of the sweetpotatoes position in the system.

Overall the results from the ten sweetpotatoes demonstrated just how varied the shape of sweetpotatoes can be. The results also demonstrated that the findings of Wright et al. (1986) appear to be correct. Based on the observations and the inspection results there are several opportunities for future research in this area.

1. Using the developed system a study could be conducted into the color and appearance of the skin of the sweetpotatoes. The variation of color on the skin can be attributed to insect damage, pathological damage or skinning. The skinning of sweetpotato roots is a major problem both from an appearance stand point and a pathological stand point. There have been numerous studies that have looked into skinning of sweetpotatoes several of these include Blankenship et. al 2002, Bouwkamp et al. 1972, and La Bonte et al 1993. Using the color feature of the 542c an in-depth study on the variation of color of the skin would give researchers an idea on amount of skinning on the surface. This knowledge could then be used to develop new processes that cause less damage to the surface of the sweetpotato.
2. A Study looking into reducing the inspection time of this system would be beneficial if this system is to ever be used in a major inspection capacity. The system was designed for use only in a breeding program to aid researchers in selecting the best roots so speed of the process was not the main issue; however the system was made a fast as possible to allow the researchers to inspect a large number of roots per day. Currently the system takes six

seconds per inspection, without including the time required to place the sweetpotato in the system. The incorporation of a conveyer belt and more imaging devices might allow for faster and constant inspections.

3. A study could be conducted on different methods of measuring the straightness characteristics, to improve the sensitivity to bending. This study used the finding of Wright et al. 1986 to quantitatively and quickly measure the straightness. The straightness algorithm used in this study was written to limit complex math, which would aide in the speed of the system, at a cost of possible sensitivity. A Fourier Transform method, such as the method described by Tao et al. 1995 for potatoes. Might produce results that would be more sensitive to the bending of the sweetpotato
4. A study looking into matching the shape of the sweetpotato to either a prolate spheroid, as was suggested by Wright et al. (1986) or some other standard shape. In this study an algorithm had be developed to produce a number that matched the sweetpotato to a prolate spheroid. However, this algorithm did not produce results that were accurate or repeatable and was removed from the system prior to final testing.

Bibliography

Blankenship, S and Boyette, M.D. (2002). Root Epidermal Adhesion in Five Sweetpotato Cultivars during Curing and Storage. *HortScience* 37(2), pp 374-377

Bouwkamp, J.C., Scott, L.E., and Harris, W.L. (1972). Rapid Determination of "Skin-toughness" in Sweetpotato. *HortScience* 7(5), pp 473-474

DVT Script Reference Manual 4th edition, (2002). DVT Corporation

Cheng, Z. and Ling, P. P. (1994). Machine vision techniques for somatic coffee embryo morphological feature extraction. *Transactions of the ASAE* 37 (5), pp. 1663–1669.

CIP, AVRDC, IBPGR. 1991. *Descriptors for Sweet Potato.* Humán, Z., editor. International Board for Genetic Resources, Rome, Italy.

Hamiltion, M.G., Jones, A., Dukes, P.D. and Schalk, J.M. (1986). Selection criteria for breeding sweetpotatoes for industrial uses. *HortScience* 21 (6), pp. 1426-1428

Heinemann P.H., Hughes R., Morrow C.T., Sommer H.J., Beelman R.B. and Wuest P.J. (1994). Grading of mushrooms using a machine vision system. *Transactions of the ASAE* 37 (5), pp. 1671–1677.

Heinemann, P.H., Pathare, N.P. and Marrow, C. T. (1996). An automated inspection station for machine-vision grading of potatoes. *Machine Vision and Applications* 9, pp. 14-19.

Jones, A., Steinbaur, C. E., and Pope, D. T. (1969). Quantitative inheritance of ten root traits in sweetpotatoes. *Journal American Society of Horticultural Science* 94, pp. 271-275.

La Bonte, D.R. and Wright, M.E., (1993) Image Analysis Quantifies Reduction in Sweetpotato Skinning Injury by Preharvest Canopy Removal. *HortScience* 28(12), pp. 1201

Martin, F. and Rhodes, A (1983). Correlation among characteristics of sweet potato roots, and Intraspecific grouping. *Euphytica* 32, pp. 453-463

Nielsen, H.M. and Paul, W., (1998). Modeling image processing parameters and consumer aspects for tomato quality grading. In: Mathematical and Control Application in Agriculture and Horticulture. Proceedings of the Third IFAC Workshop, Pergamon/Elsevier, Oxford, UK.

Paulus, I., De Busscher, R., and Schrevens, E. (1997). Use of image analysis to investigate human quality classification of apples. *Journal of Agricultural Engineering Research* 68, pp. 341-353

Paulus, I., and Schrevens, E. (1999). Shape characterization of new apple cultivars by Fourier expansion of digitized images. *Journal of Agricultural Engineering Research* 72, pp. 113–118.

The Postharvest Handling of SweetPotatoes: With Construction guidelines for Negative Horizontal Ventilation Curing and Storage Facilities. North Carolina Cooperative Extension Service. Publication: AG-413-10-B

Tao, Y., Morrow, C.T., Heinemann, P.H. and Sommer, H.J. (1995). Fourier based separation techniques for shape grading of potatoes using machine vision. *Transactions of the ASAE* 38 (3), pp. 949–957.

USDA, (1997) United States Standards for Grades of Sweetpotatoes. USDA Agricultural Marketing Service, Washington D.C.

USDA, (2003) *U.S. SWEET POTATO STATISTICS (03001)*. Retrieved Jun 15, 2004 from http://jan.mannlib.cornell.edu/data-sets/specialty/03001/

Wills, R., McGlasson, B., Graham, D. and Joyce, D (1998) *Postharvest An Introduction to the Physiology & Handling of Fruit, Vegetables and Ornamentals.* Cab International, New York, NY.

Wooten, J., White, J., Thomasson, J. and Thompson, P (2000). "Yield and Quality Monitor for Sweetpotato with Machine Vision", 2000 ASAE Annual International Meeting, Paper No. 001123. ASAE, 2950 Nile Road, St Joseph, MI 49085-9659 USA.

Wright, M.E., Tappan, J.H., and Sistler, F.E. (1986). The size and shape of typical sweet potatoes. *Transactions of the ASAE* 29 (3), pp. 678-682.

Appendix A

Code for Algorithms

Background script (Based on Example script, DVT Script Manual, 2002, p 170)

```
class sweetpotato
{
	public static void main()
	{
		//declare socket objects to use and the necessary variables
		Socket socket1;
		Socket Socket2;
		int Result;
		int time=clock();
		int i,counter;
		int delay=50;
		int stringlength=20;
		long Bit=1;
		byte output[]=new byte[3];
		byte input[]=new byte[20];
		byte closeconn[]=new byte[15];
		byte trigger[]=new byte[7];
		boolean connection = true;
		socket1 = new Socket();
		Socket2 = new Socket();
		// initialization of motor, cycles through each output
		SetOutputs((Bit<<63),(Bit<<60));
		sleep(delay);
		SetOutputs((Bit<<62),(Bit<<63));
		sleep(delay);
		SetOutputs((Bit<<61),(Bit<<62));
		sleep(delay);
		SetOutputs((Bit<<60),(Bit<<61));
		sleep(1000);
		for(i=0;i<5;i++)
		{
			Inspect();
			sleep(1000);
		}
		output[0]=67;
		output[1]=78;
		output[2]=79;
		//binds socket1 to port 5005. Sets up socket1 to act as a server
		Result = socket1.Bind(5005);
		Result = socket1.Listen();
```

```
DebugPrint("Success 5005");

//Progam loops until a connection as beeing made to socket1. Socket2
// is set up to send and recieve information
do
{
Result=socket1.Accept(Socket2); // Result=0 for a successful connection
} while(Result!=0);
// Read in information from connection,
while(connection)
{
        Result=Socket2.Send(output);
        for(i=0;i<stringlength;i++)
        {
                input[i]=0; // resets the input array to all zeros
        }
        i=-1;
        do
        {
//check for a Timeout Failure, Result=-14. If Timeout does not occur then increment i
                if (Result!=-14)
                {
                        i++;
                }
                Result = Socket2.Recv(input,i,1); // reads the input
//keep reading until a Line Feed  ASCII character is found or timeout error.
        } while(input[i]!=10 || Result==-14);
// Checks the first 7 bytes from the input array
        i=0;
        while(i<7)
        {
                trigger[i]=input[i];
                i++;
        }
        // checks to see if word trigger was read in from the connection
        // if true the inspection sequence is run
        if(String(trigger).compareTo("trigger")==0)
        {
                //runs the inspection on the current position
                Inspect();
                // turn the stepper motor to next viewing position
                for(counter=0;counter<5;counter++)
                {
                        SetOutputs((Bit<<63),(Bit<<60));
                        sleep(delay);
                        SetOutputs((Bit<<62),(Bit<<63));
```

```
                                        sleep(delay);
                                        SetOutputs((Bit<<61),(Bit<<62));
                                        sleep(delay);
                                        SetOutputs((Bit<<60),(Bit<<61));
                                        sleep(delay);
                                }
                                sleep(500);
                        }
                        DebugPrint(String(input));
                        i=0;
                        while(i<15)
                        {
                                closeconn[i]=input[i];
                                i++;
                        }
                        if(String(closeconn).compareTo("closeconnection")==0)
                        {
                                socket1.Close(); // closes the connection to port 5005
                                Socket2.Close(); //oses the send/recieve port
                                connection = false;
                                DebugPrint("connection closed");
                        }
                }
        }
}
```

Length

```
class Length
{
   public void inspect()
   {
   int num;
        float xmin,xmax,ymin,ymax;
        num=Width.NumEdgePoints;
        xmin=Width.EdgePoint.X[0];
        xmax=Width.EdgePoint.X[num-1];
        ymin=(Width.EdgePoint.Y[0]+Width.EdgePoint2.Y[0])/2;
        ymax=(Width.EdgePoint.Y[num-1]+Width.EdgePoint2.Y[num-1])/2;
        Length.Distance=xmax-xmin;
        MarkImage(xmin,ymin,1);
        MarkImage(xmax,ymax,1);
        MarkImage(xmin,ymin,xmax,ymax,1);
        Length.String= xmax-xmin*Scale.Scale; // Computes Length


   }
}
```

Straightness

```
class straightness
{
   public void inspect()
   {
     int count,number,i,markerz,markery,x,yo;
     float theta[],center;
     float Radiuscenter,radiusupper,radiuslower;
     float ztemp,ytemp,z,y, zcenter[],ycenter[],sumy,tempsumy,SSy,sumz,tempsumz,SSz;
     float slopey1, slopez1;
     number=Width.NumEdgePoints;
     zcenter=new float[number];
     ycenter=new float[number];
     theta=new float[6];
     theta[0]=0;
     theta[1]=36;
     theta[2]=72;
     theta[3]=108;
     theta[4]=144;
     theta[5]=180;
     count=RegisterReadByte(12);
     Radiuscenter=(Width.EdgePoint.Y[0]+Width.EdgePoint2.Y[0])/2;
// Computes Centroids of all slices
//--------------------------------------------------
      if (count>0)
      {
         for (i=0; i<number; i++)
         {
                  markery=32+i*4;
                  markerz=4090-i*4;
                  radiusupper=Width.EdgePoint.Y[i]-Radiuscenter;
                  radiuslower=Width.EdgePoint2.Y[i]-Radiuscenter;
                  ytemp=RegisterReadFloat(markery);
                  y=radiusupper*cos(theta[count]*3.1415/180)+radiuslower*
                           cos(theta[count]*3.1415/180)+ytemp;
                  RegisterWriteFloat( markery, y);
                  ztemp=RegisterReadFloat(markerz);
                  z=radiusupper*sin(theta[count]*3.1415/180)+radiuslower*
                           sin(theta[count]*3.1415/180)+ztemp;
                  RegisterWriteFloat( markerz, z);
         }
     }
     else  // executes when count=0
         {
                  for (i=0; i<number; i++)
```

```
                {
                        markery=32+i*4;
                        ytemp=0;
                        radiusupper=Width.EdgePoint.Y[i]-Radiuscenter;
                        radiuslower=Width.EdgePoint2.Y[i]-Radiuscenter;
                        y=radiusupper*cos(theta[count]*3.1415/180)
                                +radiuslower*cos(theta[count]*3.1415/180)+ytemp;
                        RegisterWriteFloat( markery, y);
                        markerz=4090-i*4;
                        ztemp=0;
                        z=radiusupper*sin(theta[count]*3.1415/180)
                                +radiuslower*sin(theta[count]*3.1415/180)+ztemp;
                        RegisterWriteFloat( markerz, z);
                }
        }
     if(count==5) // computes the centers of mass after six inpections
     {
        for(i=0;i<number;i++)
        {
                markery=32+i*4;
                markerz=4090-i*4;
                y=RegisterReadFloat(markery);
                ycenter[i]=y/12;
                z=RegisterReadFloat(markerz);
                zcenter[i]= z/12;
                MarkImage(Width.EdgePoint.X[i],y/12+140,1);
                MarkImage(Width.EdgePoint.X[i],z/12+130,2);
                MarkImage(Width.EdgePoint.X[i],140,3);
                MarkImage(Width.EdgePoint.X[i],130,4);
        }
     }

// Computes Straightness
//-----------------------------------------------
     if (count==5)
     {
        for(i=1;i<number-1;i++)
        {
                tempsumy=SSy;
                sumy=ycenter[i-1]-ycenter[i];
                SSy=sqrt(pow(sumy,2))+tempsumy;
                tempsumz=SSz;
                sumz=zcenter[i-1]-zcenter[i];
                SSz=sqrt(pow(sumz,2))+tempsumz;
         }
         straightness.String=(SSy+SSz)*Scale.Scale;
```

```
            straightness.Distance=(SSy+SSz)*Scale.Scale;;
    }
    else
    {
        straightness.String="busy";
     }
  }
}
```

Output 2: reports shape and straightness in the raw data

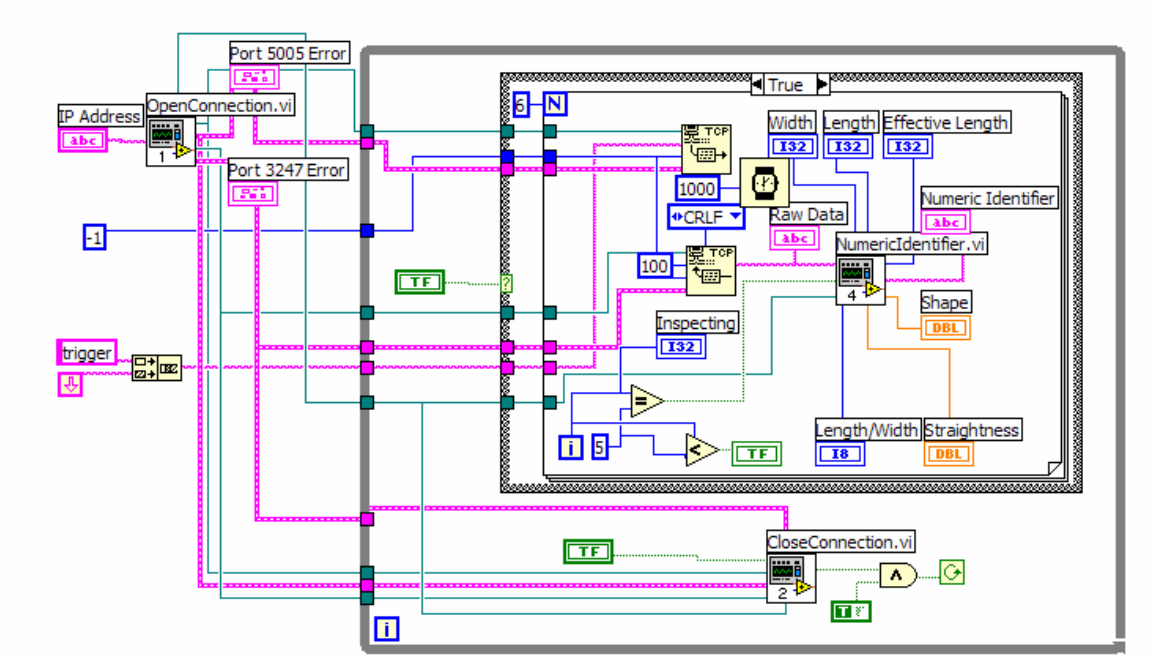
```
class Output2
{
   public void inspect()
   {
        byte count;
        int shpe,straight;
        count=RegisterReadByte(12);
        if(count==5)
        {
                straight=straightness.Distance;
                shpe=RegisterReadShort(19);
                count++;
                Output2.String=shpe+","+straight;
        }
        else
        {
                count++;
                Output2.String=count;
        }
   }
}
```

Output 1: reports length, width, and effective length in raw data

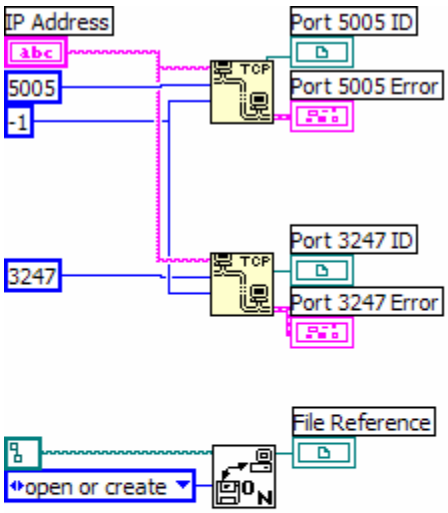
```
class Output
{
   public void inspect()
   {
         byte count;
         int  width,len,efflen;
         int  width2,len2,efflen2;
         int  maxwidth,maxlen,minefflen;
         count=RegisterReadByte(12);
         if(count==0)
         {
                  width2=0;
                  len2=0;
                  efflen2=10000;
         }
         else
         {
                  width2=RegisterReadShort(13);
                  len2=RegisterReadShort(15);
                  efflen2=RegisterReadShort(17);
         }
         width=Width.Distance*Scale.Scale;
         efflen=Efect_length.Distance*Scale.Scale;
         len=Length.Distance*Scale.Scale;
// Finds the maximum width, length and minimum effective length
         if(width>width2)
         {
                  width2=width;
                  RegisterWriteShort(13,width2 );
         }
         if (len>len2)
         {
                  len2=len;
                  RegisterWriteShort(15,len2);
         }
         if (efflen<efflen2)
         {
                  efflen2=efflen;
                  RegisterWriteShort(17,efflen2);
         }
//------------------------------
         if(count==5)
         {
                  count++;
```

```
                maxwidth=RegisterReadShort(13);
                maxlen=RegisterReadShort(15);
                minefflen=RegisterReadShort(17);
                Output.Distance=count;
                Output.String=maxwidth+","+maxlen+","+minefflen;
        }
        else
        {
                count++;
                Output.Distance=count;
                Output.String=count;
        }
    }
}
```

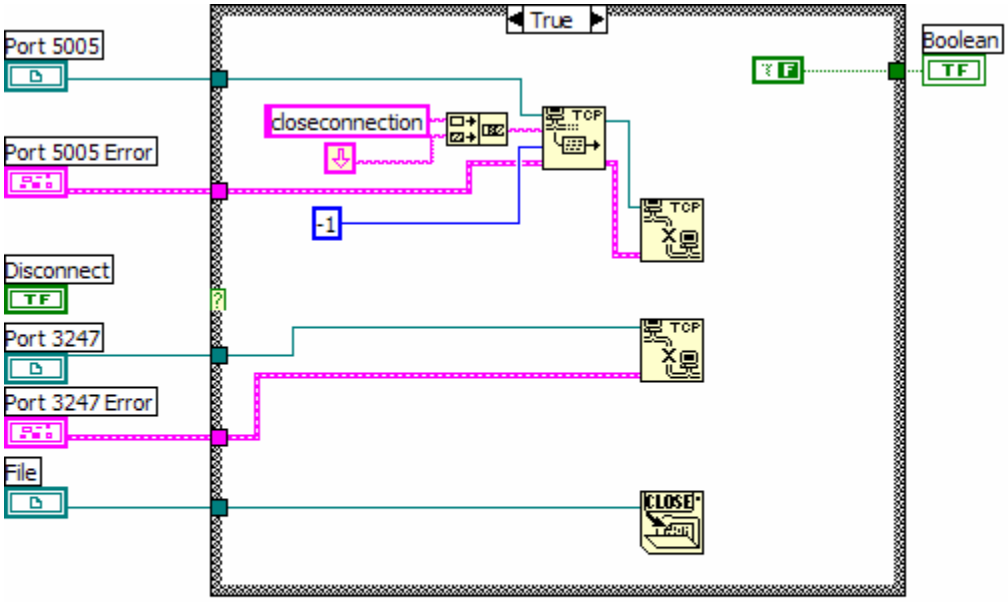
Labview main.vi



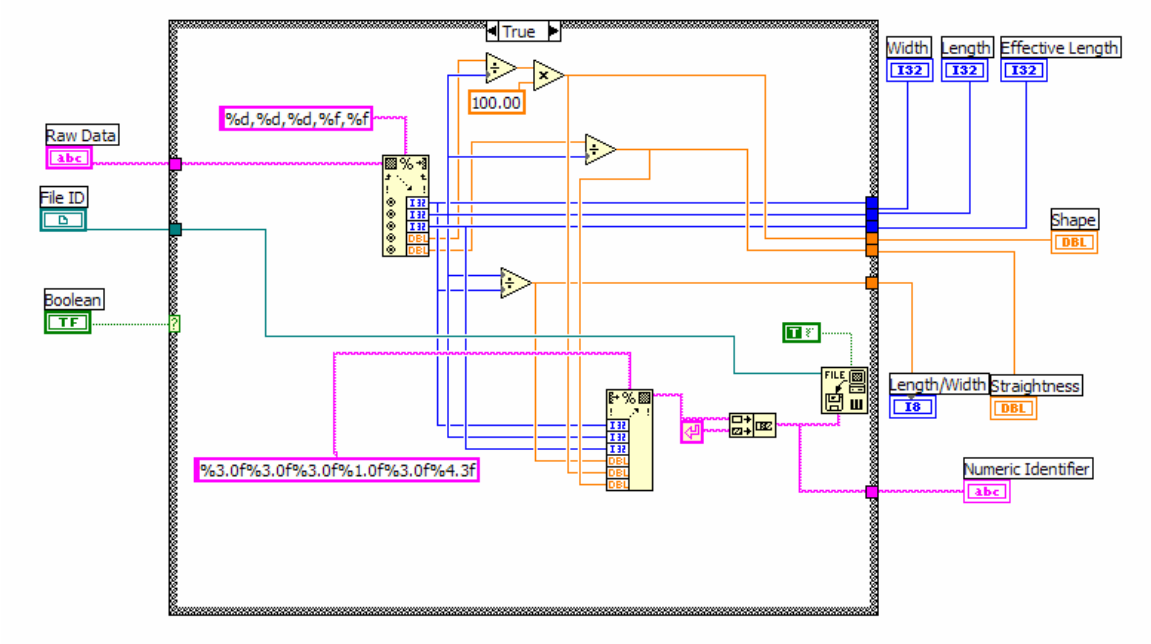
Labview Open Connection.vi



Labview Close Connection.vi



Labview Numerical Identifer.vi



Appendix B:
Raw data

Standard Objects

Wooden Ball	Ping Pong Ball	Tennis Ball
36 3110.065	61 5710.053	66 6010.167
35 3110.065	61 5710.053	66 6010.183
36 3010.067	61 5710.053	65 6010.167
36 3010.100	61 5710.053	65 6010.183
35 3010.100	61 5710.053	65 6010.183
35 3010.067	61 5510.073	66 6010.183
35 3010.067	61 5710.053	66 6010.150
35 3010.067	61 5710.053	65 6010.167
36 3010.033	61 5710.053	66 6010.133
35 3010.100	61 5510.053	66 6010.183
35 3010.067	61 5710.053	66 6010.167
35 3210.063	61 5710.053	65 6010.183
35 3010.067	61 5710.053	66 6010.167
35 3010.067	61 5710.053	65 6010.167
35 3010.067	61 5710.053	65 6010.167
35 3010.067	61 5610.053	66 6010.150
36 3210.063	61 5710.073	66 6010.167
35 3010.100	61 5710.053	66 6010.167
35 3010.067	61 5710.053	65 6010.167
35 3010.033	61 5710.053	66 6010.167
36 3010.100	61 5710.053	66 6010.150
35 3010.067	61 5710.053	65 6010.183
35 3010.067	61 5710.053	65 6010.183
35 3010.067	61 5710.053	65 6010.150
36 3010.100	61 5710.053	66 6010.167
35 3010.067	61 5710.053	66 6010.167
35 3010.067	61 5710.053	66 6010.167
35 3010.033	61 5610.053	66 6010.167
36 3010.067	61 5710.053	66 6010.167
35 3010.067	61 5710.053	66 6010.167

Mini Football 1	Mini Football 2
7512320.081	8912610.071
7512320.089	8912610.071
7512320.081	8912610.071
7512320.089	8912610.071
7512320.089	8812610.063
7512320.089	8812610.071
7512320.073	8912610.071
7512320.081	8912610.071
7512320.065	8712610.063
7512320.089	8912610.071
7512320.081	8912610.071
7612320.089	8912610.071
7612320.081	8912610.071
7512320.081	8812610.071
7612320.081	8912610.079
7512320.073	8712610.071
7512320.081	8912610.071
7512320.081	8912610.071
7612320.089	8912610.063
7512320.081	8912610.063
7512320.073	8912610.071
7512320.089	8912610.071
7512320.089	8912610.071
7512320.073	8912610.063
7612320.081	8912610.071
7612320.081	8912610.063
7512320.081	8912610.071
7512320.081	8812610.071
7512320.081	8912610.071
7512320.081	8912610.071

Numerical Identifiers

Ten Sweetpotatoes

Sweetpotato 1

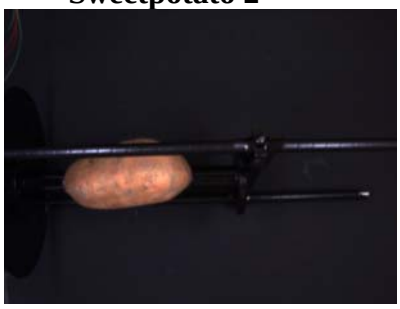


4417440.103
4417440.103
4317440.114
4417440.103
4317440.109
4417440.109
4417440.103
4417440.103
4417440.103
4417440.114
4317440.109
4317440.103
4317440.099
4417340.104
4417540.114
4317340.109
4417440.103
4317440.103
4417440.103
4317440.114
4317440.114
4417440.103
4317440.099
4417340.104
4417440.114
4317440.109
4417440.103
4317440.103
4417440.097
4417440.114

Sweetpotato 1 Inverted

4417340.115
4417340.234
4317240.093
4417340.109
4317340.115
4417340.109
4317240.988
4417340.104
4317340.115
4417340.115
4317340.104
4317240.098
4417340.104
4417340.115
4417240.110
4317240.098
4417340.109
4317340.115
4417240.116
4317240.098
4417340.104
4317340.115
4417340.109
4417340.109
4317340.098
4417340.104
4217240.116
4417340.104
4417340.104
4217340.098

Sweetpotato 2



6912020.125
6912020.133
6812020.125
6812220.131
6812420.129
7012220.123
7012220.123
6812220.114
6912420.121
6912220.131
6912320.122
6912320.122
6812220.123
7012420.121
6812220.114
6912320.130
7012320.122
6812220.123
7012320.122
7012220.123
7012320.130
6912220.123
6912320.130
6912220.123
6912320.130
7012220.123
6912320.130
6812220.114
6912320.122
6812220.123

Sweetpotato 2 Inverted	Sweetpotato 3	Sweetpotato 3 Inverted
6812220.133		5816130.192
6912220.123		5815930.239
6912220.123		5816230.240
6912320.130		5716130.217
6912220.123		5716130.223
6912220.123		5716130.211
6912220.131		5716130.217
6812220.123		5816130.223
6912220.131		5816130.198
6812220.131	5416130.136	5716130.198
7012220.123	5616130.236	5816130.267
6912320.122	5916130.186	5616130.205
6912220.123	5716130.173	5816130.235
6812220.123	5516130.223	5516130.254
6912220.131	5916130.180	5816130.242
6912220.131	5815930.232	5716130.242
6912220.123	5816030.200	5816130.198
6812220.123	5816130.211	5516130.248
6912220.131	5715930.220	5816130.248
6912220.123	5716130.223	5816130.205
6812220.131	5816130.161	5716130.198
6912220.123	6016130.155	5516230.240
6812220.139	6016130.180	5716130.244
6912220.123	5916030.237	5716130.205
6912220.131	5816130.223	5816130.248
6912220.114	6015930.157	5716230.234
6912220.123	5916030.231	5716130.248
6812220.123	5616130.173	5516130.242
6812220.131	5816230.222	5616130.267
6912220.123	5416230.259	5816130.223
	5716230.240	
	5916130.298	
	5816230.240	
	5816230.179	
	5416130.180	
	5616130.205	
	5816130.236	
	5816130.248	
	5716230.228	

]

Sweetpotato 4

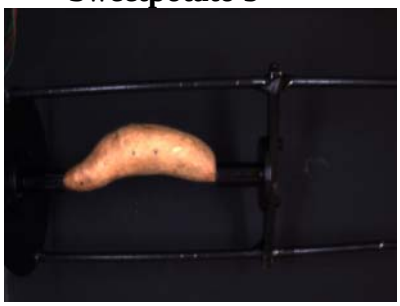


4711020.109
4710920.100
4710920.119
4710920.110
4710920.119
4710920.110
4710920.100
4711020.118
4710920.119
4711020.118
4710920.100
4711020.100
4711020.109
4710920.110
4711020.118
4710920.110
4711020.109
4710920.110
4711020.109
4710920.128
4710920.119
4710920.100
4710920.110
4711020.118
4711020.118
4711020.109
4710920.128
4710920.110
4810920.100
4710920.137

Sweetpotato 4 Inverted

4711020.118
4911020.090
4811020.127
4810920.119
4811020.109
4811020.109
4711020.090
4810920.110
4810920.119
4711020.109
4810920.174
4710920.082
4811020.163
4811020.109
4811020.109
4710920.091
4811020.154
4810920.128
4711020.118
4811020.200
4711020.090
4811020.154
4810920.119
4711020.109
4811020.109
4811020.136
4810920.128
4711020.109
4711020.109
4710920.100

Sweetpotato 5



5414830.243
5214930.201
5414930.187
5415030.266
5414930.221
5414930.194
5415030.233
5415030.273
5515030.213
5415130.231
5214930.181
5515030.253
5514930.255
5314930.201
5414830.229
5515030.240
5514930.241
5214830.209
5315030.220
5214930.174
5214830.223
5515030.226
5315030.226
5214830.189
5515030.233
5515030.213
5415030.180
5415030.186
5415030.246
5515030.273

Sweetpotato 5 Inverted

5415030.260
5515030.246
5414930.208
5414930.194
5414930.208
5414930.187
5414930.167
5615030.220
5415030.193
5115030.200
5515030.260
5415030.280
5414930.221
5514930.228
5214930.208
5314930.208
5414930.214
5414930.094
5414930.187
5414930.248
5414930.214
5414930.214
5414930.187
5414930.221
5114930.221
5414930.187
5414930.174
5414930.187
5415030.233
5214930.214

Sweetpotato 6



51 8920.179
51 9120.131
51 8920.146
51 9120.120
51 9120.153
51 8920.168
51 9120.131
51 9020.155
51 9120.120
51 9120.153
50 8920.168
51 9120.131
51 9020.155
51 9120.120
52 9120.142
50 8920.168
51 9120.131
51 8920.157
51 9120.120
51 9120.142
51 8920.168
51 8920.123
51 9020.155
51 9120.131
50 9120.153
51 8920.168
51 9120.131
52 9020.155
51 9120.120
51 9120.142

Sweetpotato 6 Inverted

52 9220.108
51 9220.119
52 9020.133
51 9220.097
52 9220.152
52 9220.097
52 9020.111
52 9220.108
51 9220.119
52 9020.133
51 9220.097
52 9020.133
52 9220.108
51 9220.108
52 9020.133
51 9220.108
52 9220.130
52 9220.097
51 9220.119
52 9020.133
51 9220.097
52 9220.130
52 9220.097
51 9220.119
52 9020.133
51 9220.108
52 9220.152
52 9220.108
51 9220.119
52 9220.130

Sweetpotato 7



5412020.200
5412120.223
5212120.281
5212120.214
5412120.214
5412020.250
5412120.190
5512120.181
5212120.223
5412120.214
5112120.214
5412120.181
5512020.183
5212020.216
5412220.213
5212120.214
5412120.157
5512120.214
5012020.233
5512120.214
5212020.208
5512020.183
5512120.206
5212020.200
5512120.214
5212120.206
5212020.200
5412120.214
5012020.225
5412120.223

Sweetpotato 7 Inverted

5312120.231
5412020.158
5412120.198
5112020.208
5412020.216
5412120.198
5412020.166
5412120.190
5112120.214
5412020.200
5312120.206
5412020.175
5412020.191
5112120.214
5412020.200
5312120.190
5412120.165
5412020.191
5112120.223
5412020.200
5412120.190
5412020.158
5412120.190
5112120.214
5412020.200
5412120.206
5412120.165
5412120.190
5112120.223
5412020.200

Sweetpotato 8



4714430.104
4814430.097
4814430.111
4714430.125
4714330.132
4714430.125
4714430.131
4714330.111
4714430.125
4714330.125
4714430.125
4714430.131
4714330.111
4714430.131
4714230.126
4714430.131
4714330.132
4714330.111
4714430.125
4714330.125
4714330.125
4714430.138
4714430.111
4714430.125
4714330.125
4714430.125
4714430.131
4714330.111
4714430.125
4714230.133

Sweetpotato 8 Inverted

4814330.125
4814330.132
4714330.139
4814330.125
4614330.139
4814330.118
4814330.132
4714330.125
4814330.132
4614330.139
4814330.132
4614330.125
4814330.139
4714330.139
4714330.125
4714330.125
4714330.132
4814330.139
4714330.139
4814330.125
4714330.139
4714330.111
4814330.132
4714330.111
4714330.132
4714330.139
4814230.105
4614330.139
4714330.104
4614330.118

Sweetpotato 9



4414430.131
4414430.145
4414230.133
4414430.138
4414330.146
4414430.125
4414430.166
4414230.133
4414430.125
4414330.153
4414430.138
4414430.145
4414230.133
4414430.131
4414330.146
4414430.159
4414430.138
4414430.138
4414430.131
4414230.126
4414430.131
4414330.146
4314430.131
4414430.125
4414230.140
4414430.138
4414430.138
4414430.131
4414430.159
4414430.118

Sweetpotato 9 Inverted

4414430.138
4414330.125
4414430.138
4414430.152
4414330.160
4414430.138
4414330.132
4414430.145
4414430.166
4414330.160
4414430.145
4414430.152
4414430.138
4414330.139
4414330.146
4414430.138
4414330.139
4414430.145
4414430.138
4414430.138
4414430.145
4414330.160
4414430.145
4414330.146
4414430.138
4414430.152
4414330.132
4414430.145
4414330.139
4414430.145

Sweetpotato 10



5218540.173
5118540.205
5418530.183
5418630.204
5118540.210
5418630.193
5318640.204
5118540.173
5418630.177
5118540.183
5418530.194
5318530.221
5418530.183
5418530.178
5118040.222
5418630.188
5418530.183
5118540.183
5318640.188
5418530.173
5418630.188
5418630.177
5318540.205
5418630.188
5418630.193
5318530.189
5418630.182
5118540.194
5418530.183
5418630.204

Sweetpotato 10 Inverted

5318530.189
5118540.189
5418530.189
5218540.178
5418530.178
5418530.200
5118540.189
5218640.183
5318530.216
5118540.189
5418530.173
5418530.205
5118540.178
5418530.205
5418530.189
5218540.184
5418530.211
5418430.185
5418530.178
5418530.216
5218440.190
5318530.178
5418530.195
5018540.189
5318530.184
5318530.189
5018540.189
5318640.177
5418530.184
5218440.185