High Accuracy Mobile 3D Scanning Using Structured Laser Beam Patterning

Sepehr Makhsous

A dissertation

submitted in partial fulfillment of the

requirements for the degree of

Master of Science

University of Washington
2016

Committee:

Alexander V. Mamishev, Chair

Payman Arabshahi

Shwetak N. Patel

Program Authorized to Offer Degree:

Department of Electrical Engineering

University of Washington

Abstract

High Accuracy Mobile 3D Scanning Using Structured Laser Beam Patterning

Sepehr Makhsous

Chair of the Supervisory Committee:

Professor Alexander V. Mamishev
Department of Electrical Engineering

A person's diet affects their weight, lifespan, and chances of occurrence of such medical problems as diabetes, obesity, and cancer, to name a few. Enhanced dietary assessment techniques are critical for epidemiological studies that target diet-related problems. Currently, nutritional research is considerably hindered by the low accuracy in estimating individual dietary intake, and, more specifically, portion size. Dietary assessment plays an increasingly significant role in modern medical research. While inadequate diets can increase the risk of diabetes, obesity, and cancer. It is essential to design accurate, cost-effective tools that measure dietary data to advance nutritional research. This dissertation describes a low-cost and efficient method of calculating nutritional information by using 3D reconstruction and image processing. This system is called Dietary Data Recorder System (DDRS), which consists of a smartphone, a laser projector, and the main

algorithm, which extracts the data from the DDRS for volume and nutritional calculations. The DDRS software consists of four main algorithms: Automatic Laser Detection algorithm, Segmentation algorithm, 3D Mapping algorithm, and Nutritional Estimation algorithm. In particular, this dissertation focuses on the first two functions: Automatic Laser Detection algorithm and Segmentation algorithm.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

Over the past five years, I have been blessed to have had the opportunity to conduct part of my undergraduate and graduate research under the direction of Professor Alexander Mamishev, Director of the Sensors Energy and Automation Laboratory (SEAL) in the Department of Electrical Engineering at the University of Washington. I would like to thank Professor Mamishev for always pushing and demanding the best in many different aspects of my work. I would also like to thank him for supporting me to pursue my interest in entrepreneurship by completing a concurrent master's degree in business and association (MBA). In addition, I would like to thank him for enabling me to work with other research and development groups within the academic world and outside of it.

I would also like to express my gratitude to my thesis committee members, Dr. Payman Arabshahi and Dr. Shwetak Patel for their contributions of thoughts, advice, and time as committee members. I would also like to thank Professor Arabshahi for all his support, guidance, and advice in both my Engineering and Entrepreneurship studies.

Similarly, I would like to thank all members at Fred Hutchison Cancer Center, especially to Dr. Alan Kristal and Dr. Jeannette Schenk, for all their support, time, advice, and help throughout the past four years.

Chapter 1. INTRODUCTION

The work presented in this thesis is motivated by growing concerns regarding health problems related to diet, such as obesity and cancer. Despite the need, there is no efficient and accurate method of estimating the calories of the food, because people usually cannot accurately estimate food volume. This dissertation is a study in the design, modeling, and integration of a small scale smartphone add-on for measuring the nutrients of a plate of food with possibly more accuracy and less user burden in dietary measurements. This work is based on publications [1] and [2] by the author.

1.1 Introduction to Dietary Measurement For Nutritional Assessment

A person's diet affects their weight, lifespan, and chances of increased occurrence of such medical problems such as diabetes. Over the past ten years, the occurrence of diabetes has been growing rapidly in all ages but more drastically in younger age groups [3]. Enhanced dietary assessment techniques are critical for maintaining a healthy living for a diabetic patient. Moreover, it is crucial for hospitals to monitor their diabetic patients' food intake to prescribe a certain amount of insulin. Malnutrition is also a long-standing issue for hospitalized patients, and nationwide 30-50% of patients are diagnosed with it. Malnutrition greatly increases patient mortality, the duration of the hospital stay, and ultimately medical costs [4]. Currently, it is possible to measure the nutrition of a meal before sending to the patients if the meal is prepared by the medical facility, but the precise consumed amount is impossible to be calculated accurately; hence, it is estimated how much insulin needs to be prescribed. Additionally, nursing and nutrition staff are only able to assess approximately 50% of meal trays due to the staffing levels and logistics involved in meal

delivery and pickup (Harborview Medical Center internal data). This is a common problem in hospitals nationwide.

Nutritional research is considerably hindered by the low accuracy in estimating individual dietary intake, as it relies on visual estimations This proposal presents a new method that uses 3D reconstruction to determine the volume of a scanned food item accurately. After achieving the accurate measurement of the volume, the nutrient content of the food item can then be estimated using existing nutritional databases.

Current dietary assessment methods are largely inaccurate. Lack of precision in estimating energy intake and, more specifically, portion size, has hindered the nutritional research [5]. This dissertation describes an innovative approach to calculating the amount of the food using a laser dot matrix emitted from a smartphone hardware add-on.

1.2 Advantages and Disadvantageous of Using Dietary Measurements For Nutritional Assessment

Understanding nutritional info has been a keen interest in the research industry; however, in the recent decade, individuals, such as professional athletes, health enthusiasts, and patients with related dietary needs, have shown a great interest in the significance of their dietary intake. Hence, there has been much work done in the field of measuring and calculating dietary intake. Any solution to nutritional assessment hopes to provide an accurate, detailed, and easy measurement of one's food intake.; however, paper-based methods are more established in the existing research fields. 24-hour dietary recall, which we will talk about in the future sections is one of the more commonly used methods amongst nutritional researchers [6]. One of the major criticisms of existing food recording methods is that users often change the way they eat when they are recording, and then they no longer reflect "typical diet.[7]" It requires the user, after a training

process, to measure his/her dietary intake manually and write down all the information in a food diary. In addition to a food diary, the user is typically given multiple specialized measuring tools and a serving size booklet. Once the recording period is over, the food record is analyzed by a nutrition analyst or research dietitian.

Recently, digital dietary measurement has been able to compete in both research and consumer industries with the traditional approaches by proposing advanced and sophisticated methods using sensors, high-quality cameras, and image processing to reduce the limitations and the user burden of calculating while increasing the accuracy. However, using a variety of sensors and high quality has increased the cost and complexity of the operation.

1.3 Scientific and Engineering Challenges

Manually logged food diaries have proven to have a significant percentage of error when tracking nutritional intake [8]. In this dissertation we introduce a new method that can be used to create a 3D reconstruction of an object, using a structured light system, which will be discussed in the following chapters. This method can be utilized to estimate the volume of any food item, allowing us to accurately and affordably calculate the portion size and nutritional intake using the United States Department of Agriculture (USDA) [9].

Using the theory behind the structured light system, we can create a mathematical model, which can be utilized to estimate the depth and the distance from the camera to the specific object. However, measuring the volume of an entire plate introduces more variables such as, segmenting out the plate, recognizing individual items on the plate, understanding the ingredients, scanning a full 360 view, integrating all components, and communicating with the user [10].

Over the past decades, image processing techniques in 3D reconstruction as improved drastically, which allows us to calculate an object's volume with a smartphone and a laser module.

However, analyzing this data and estimating the nutritional information of a meal has its challenges. To find the nutritional info of a meal, other than the portion size we require to know the ingredients as well as the density of ingredient to calculate a detailed nutritional report [11].

1.4 Scope of the Dissertation

Figure 1.1 shows the scope of this dissertation in a visual format. This dissertation is a study of the design theory, modeling, and implementing a small-scale 3D scanner used as a nutritional assessment tool. The entire system is called Digital Dietary Recorder System (DDRS), which consists of, image processing algorithm for detecting laser pointer, segmenting food items, and calculating the volume. Also, DDRS includes a smartphone and a hardware add-on integrated into one housing for capturing the required images from the meal.

Theoretical design: In this dissertation, we attempt to design and develop an electronic dietary measurement tool using volume estimation via structured light system theory. Using a hardware add-on, we can connect a laser module to a smartphone and create a small-scale 3D scanner by accurately placing the laser pointer within a specific distance to the smartphone's camera lens. In addition to the hardware piece, there is an extensive design and development in the image processing algorithm for calculating the volume.

Functional design: The dissertation attempts to develop such a device by designing and developing individual components such as, a smartphone for capturing sensor data and footage of the meal; a Printed Circuit Board (PCB) for communicating between the laser module and the smartphone; a laser module with a customized diffraction lens for projecting a matrix plane onto the food item; a calibration algorithm to calibrate a laser module with a smartphone for depth calculation; image segmentation algorithm for cropping out the targeted food item from everything

else in the background; and laser dot detection algorithm for detecting each laser dot and calculating the distance to the device.

Future work and commercialization: There is a relatively large demand for an easy to use, accurate and reliable dietary measurement device. The road to commercialization for DDRS is not too far away, though, finding the right industry is the key to success. Currently, due to the cost and size of DDRS, it does not seem reasonable to enter a consumer market; however, knowing the demand for medical facilities for treating patients such diabetic patients, gives us a clearer path to commercialization by working solely with medical facilities.

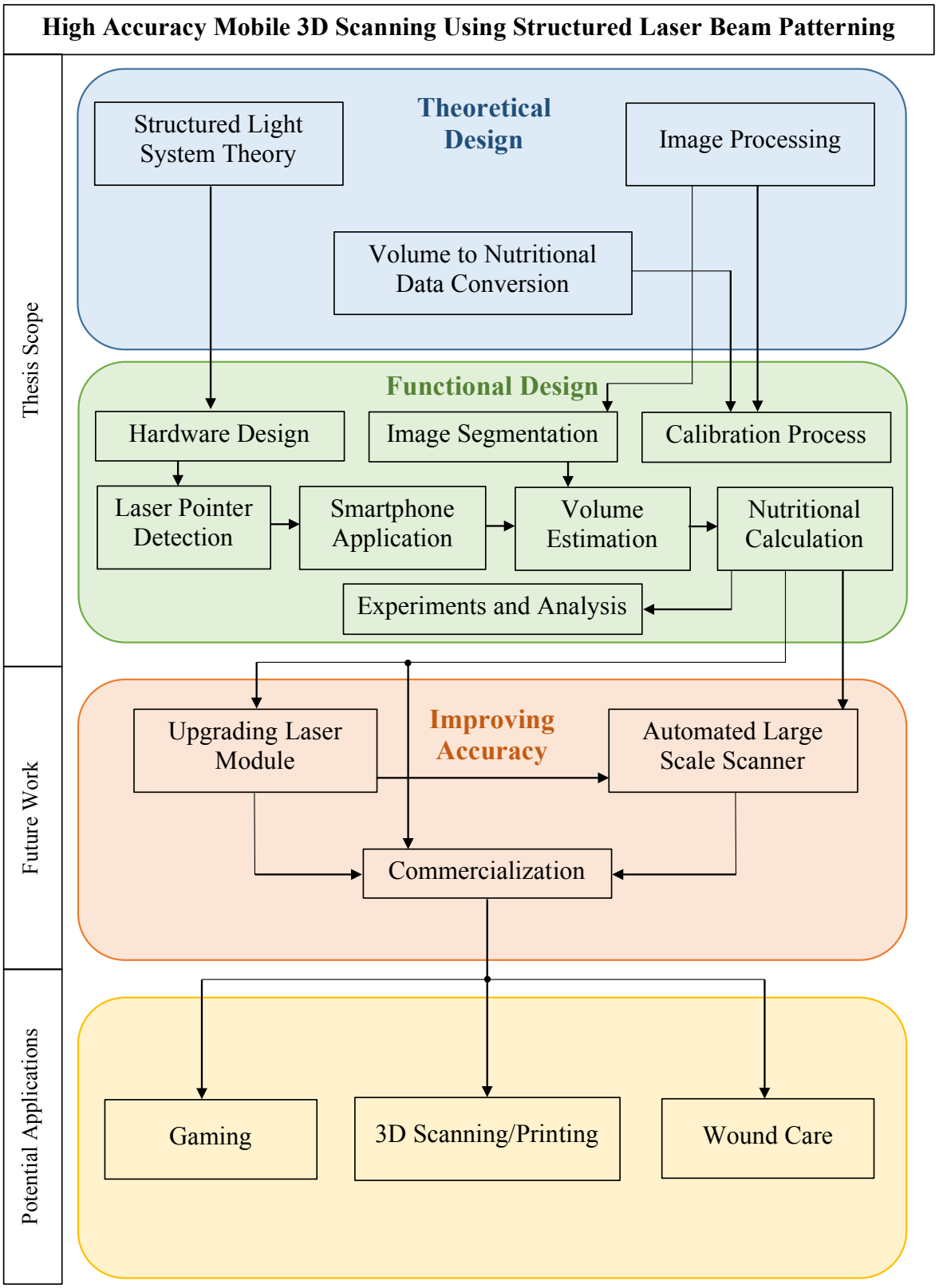

Figure 1.1. The visual presentation of the scope.

1.5 Contributions of the Dissertation

This dissertation makes three principal contributions. First, it designs a portable yet accurate 3D scanner using the structured light system by combining a laser module and a smartphone. DDRS does not require to have any external components for measuring the volume, no index cards or other referencing tools.

Second, DDRS is capable of communicating with both user and the nutritionists by taking advantage of the smartphone and cloud computing. This allows a quick turnaround of the results to both user and dietician.

Third, DDRS is one of the only portable 3D scanner which can produce exact dimensions of the scanned object. Other 3D scanners cannot be used in the field of dietary measurements even if they become portable they still cannot find the exact dimensions, which is required for finding the volume.

Chapter 2. DIETARY INTAKE ASSESSMENT BACKGROUND AND MOTIVATION

2.1 Existing and Traditional Dietary Assessment Methods

There currently are several existing ways to track dietary consumption. One is called paper method which is tracking the food that user consumes throughout the day. The 24-hour dietary recall is a well-known paper method, which allows users to log their dietary intake manually [12, 13].

There are many limitations and lack of accuracy with any manual survey-like method. This approach has been shown to be difficult for both the user and the dietician. The user must record everything eaten, and know the nutritional content of the food as well.

To overcome these issues, developers have come up with means of tracking diet with different sensors, cameras, and signal and image processing technologies. Many of these methods can even overcome the drawbacks of the 24-hour log method, and often are more accurate.

2.2 Review of Paper-based Dietary Intake Measurement

There currently are several paper-based methods to track dietary consumption. The 24-hour log uses multiple measuring tools and booklets to ensure the accuracy of tracking the food that user consumes throughout the day. However, manually logging diet is complicated: the user must record everything eaten, and know the nutritional content of the food as well. It is also inaccurate, with typical users of this method reporting caloric intakes that are off by as many as 400 calories per day [14].

2.3 Review of Digital Dietary Intake Measurement

In the past years, with improvements in image processing and variety of sensors, we have seen new digital dietary intake measurement methods become more popular. One of the most recent methods, developed by a team at the University of Ottawa, uses high-quality images taken by a handheld camera (smartphone or any other digital camera) to calculate a user's dietary intake. The proposed method uses the user's thumb to calibrate the system and to give the ability to the user to find the correct dimensions, hence, the overall portion size [15]. Of course, there are some limitations in using a human thumb as a reference point for measuring portion size: such as having to repeat calibrations, having to take each picture with a thumb in it, etc. Also, the system uses

images taken of food items and a graph-cut segmentation method to isolate the food portion of the image from the rest of the image, allowing for more precise volume estimations. To do this, they establish an undirected graph to create a coordinate system, which in turn allows them to isolate the food in the image better. With this, they aim to minimize the amount of the food that is segmented out through a system that is similar to labeling pixels as either 1 or 0, determining whether that pixel of the coordinate system is food or background. Once these pixels are identified, they can then make a digital "cut" to isolate the background from the food. This process is demonstrated briefly in Figure 2.1. The Ottawa researchers also have a method for segmenting the texture to retain this information within their calculations [16].

Although the graph-cut segmentation method implemented by the Ottawa researchers is an improvement over the previous color-texture segmentation system they had developed, it is limited in its accuracy of scanning mixed food items—such as curry or soup. With single food items, volume measurement accuracy was 90%, but for mixed food items, this same accuracy dropped to 75% [17]. One problem here is that the Ottawa researchers have to classify food items *after* they have been segmented. For complex mixed foods, this process is more complicated because the algorithm has to correctly predict which food item it is analyzing to ensure an accurate measurement. This is done by recognizing a variety of features within the segmented image. If these features are not found, then the resulting calculations will be inaccurate. Thus when food is mixed together on the plate, the system has difficulties determining features peculiar to food and is unable to determine the food item.

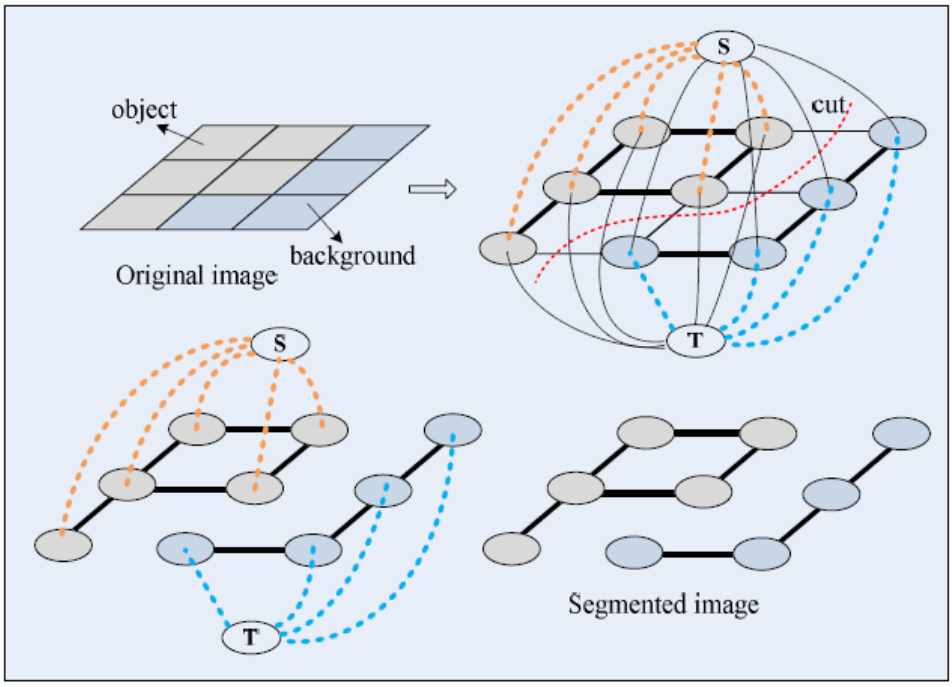

Figure 2.1. Demonstration of the graph-cut process for segmentation of an image.

However, the graph-cut method excels in reducing the boundary around the food, allowing for better predictions. A main limitation of the system is the necessity for an extensive database used to predict the type of food being analyzed. The inaccuracy of this system, especially towards mixed food items, suggests the limitations of the graph-cut method, showing possible room for improvement from a better segmentation system.

2.4 Review of 3D Reconstruction from Food Image

Based on our research, most of the digital dietary measurement methods use some 3D reconstruction to find the volume of the scanned item. However, to reconstruct a 3D map in space, one would need to have the true dimensions of the object, which requires knowing the distance from the camera to the object in multiple designated points. As mentioned in previous sections, one of the methods used for capturing the distance between the camera and the object is to use the

user's thumb in the picture as a reference. Although this provides a reference for calculating the true dimensions, it lacks the rotational and multi-point reference. Other methods include an index card which enough information for capturing the rotational data and a multipoint reference. However, the index card method does not give enough referencing points, and the ones that are given are on the card not on the food item, this causes a decrease in accuracy of the calculated volume.

The algorithm used for the Dietary Data Recorder System (DDRS) implements laser dots as the main reference to reconstruct the 3D image. The laser dots are extracted from the image using either the AutoDot detection algorithm or through manual user selection. Both methods map the coordinates of the dots with respect to a 2D plane. A center dot is taken as a reference, and all the other dots will be indexed according to their distances from this center dot. Calibration data is then used with the laser dots indices to calculate the depth of the image. The calculated depth values allow for the final 3D reconstruction, as the planar image (pixels) is transformed into a three-dimensional representation.

2.5 CHAPTER SUMMARY

There are many methods for calculating dietary intake; however, in this chapter we discussed the two most common categories, paper method, and digital method. In each category, we discussed the advantages, and limitations of each category, as well as, popular techniques and approaches for each one in depth. Overall, the majority of the focus was dedicated towards digital dietary assessment techniques, which has allowed us to briefly discussed how DDRS plans on addressing some of these limitations.

Chapter 3. DIGITAL DIETARY RECORDER SYSTEM (DDRS) DESIGN

3.1 DESIGN OVERVIEW

The design of DDRS has two main pieces, the physical, and the software (the algorithm). The physical portion consists of a laser module, diffraction lens, a printed circuit board, and a smartphone, which are all mounted on a 3D printed housing out of ABS plastic. The housing is designed using a 3D computer-aided design (CAD) software, SolidWorks. As shown Figure 3.1, the housing provides an appropriate fit for all the components in a portable size.

The smartphone is the only component that connects the physical portion to the software piece. By having a customized smartphone app, DDRS can upload the captured video in addition to all the sensor data from the cell phone (i.e. accelerometer data, gyroscope data, etc.) on the cloud to be processed. The software starts by selecting the best six frames from the captured video and then processes each frame by segmenting and 3D mapping the individual items on the plate. While this process is completing, the user is inputting the names of each item on the plate. Once all the data is processed and input by the user, the software converts the volume data to nutritional information for each item.

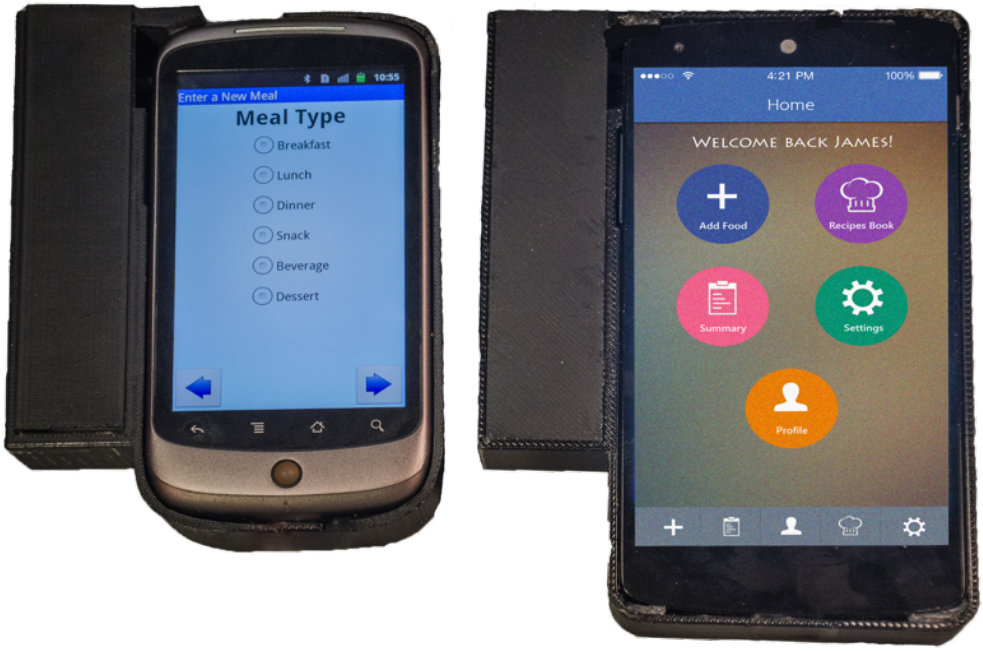

Figure 3.1. The Dietary Data Recorder System device, prototype 1 on the left and prototype 2 on the right.

3.2 HARDWARE DESIGN

To communicate between the laser module and the smartphone, we designed and developed a printed circuit board (PCB). The PCB consists of a Bluetooth module, a lithium battery to act as a power source for the laser module, a micro USB charger to charge the battery and other miscellaneous electronic components. Figure 3.2 shows the schematic of the DDRS's printed circuit board. The Bluetooth enables the DDRS smartphone app to communicate to the laser module, which allows the DDRS smartphone app to turn the laser off and on when needed to optimize the power consumption.

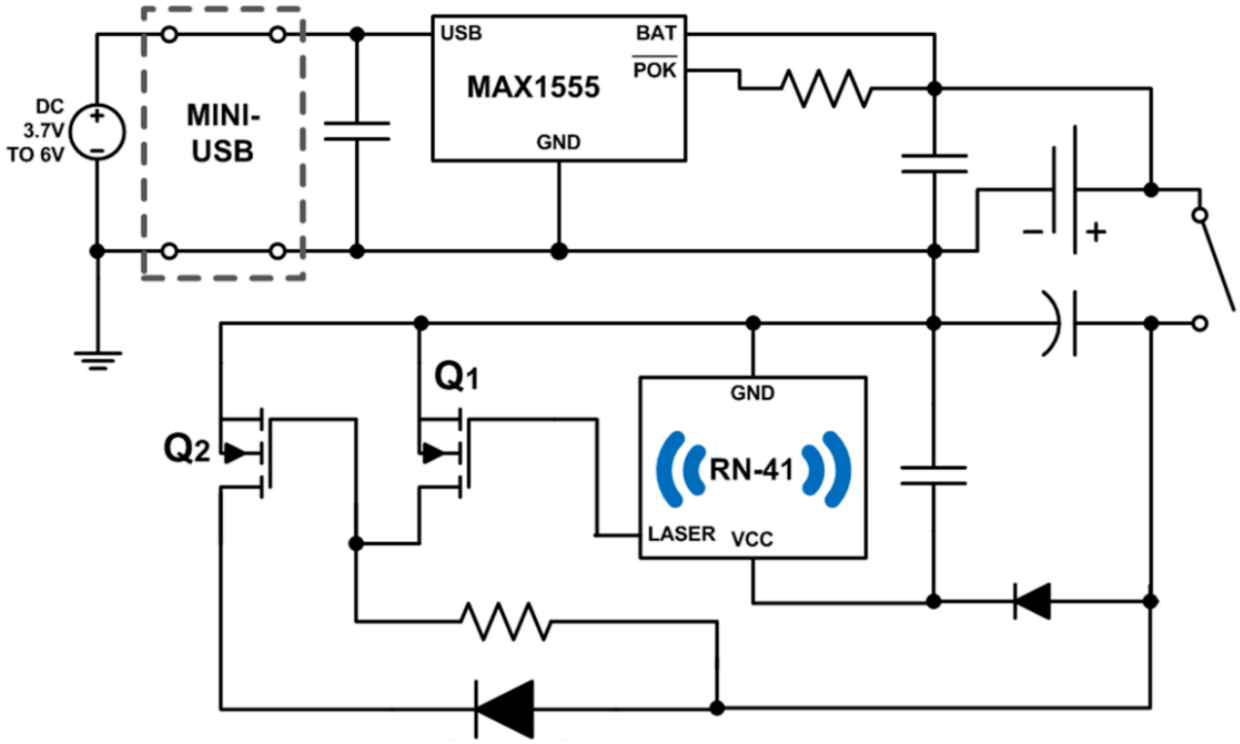

Figure 3.2. Schematic design of the printed circuit board.

3.3 Algorithm Design

3.3.1 *Overall Algorithm*

The process of recording with our current system, DDRS (Digital Dietary Recording System) could be much simpler, faster, and more accurate than the existing dietary measuring tools. The user takes a 360-degree video of their food on the plate using the DDRS device, which is then uploaded onto DDRS database cloud, allowing nutrition analysts instant access to the data or processing using DDRS software and NDSR (Nutritional Data System for Research) [18]. Figure 3.3 shows the overall process of dietary intake: recording and analysis using DDRS device, and DDRS web based software.

Figure 3.3. Digital Dietary Recording System (DDRS) overall system overview.

The core of the DDRS system is the software, which includes three main algorithms: Image Segmentation, Automatic Dot Detection, and 3D Volume Calculation. Figure 3.4, shows the flowchart of DDRS software and how all the pieces connect to calculate the volume of the food item.

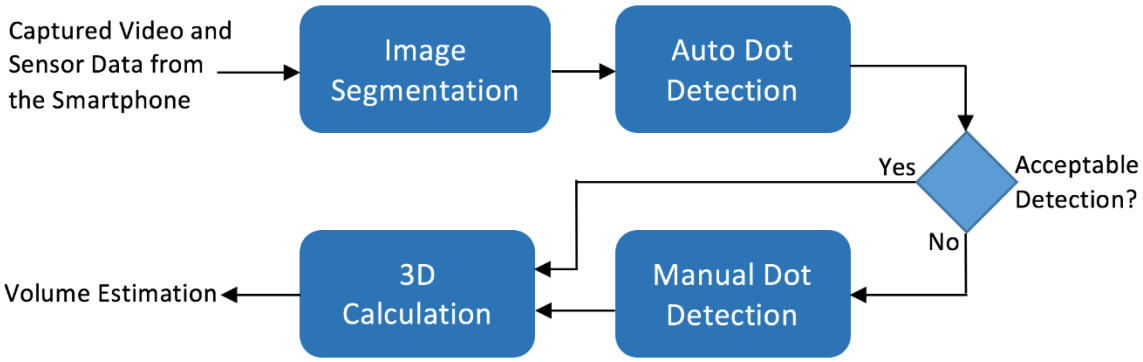

Figure 3.4. DDRS volume calculation software flowchart.

3.4 Smartphone User Interface Design

Once the scanning process is complete, the application processes the captured video to estimate the exact amount of food on the plate using the cloud servers. Additional user input is required to estimate the dietary nutrients for a specific food or dish. The prompts have been simplified and designed for a 5th-grade education level. The user can identify food items through a variety of methods, including typing the names of the foods, using voice recognition to speak the names, or, if the food is known in the DDRS cloud library, then the user can simply click on the graphic icon. The user is prompted with the ability to scan relevant recipes or other information to assist with data collection; otherwise, the user can identify the components of the meal, such as the name of the food item or dish, and, in some cases, the ingredients of a dish/recipe. The smartphone application stores, shares, and scans recipes; after scanning the plate of food, the user also has the opportunity to link the scan to preexisting recipes on the shared and local cloud database to get the ingredients of the scanned dish. The user can save new entries to the cloud and share her recipes with other users and nutritionists. Like similar technologies, the DDRS uses a cloud library of food items and has been designed to accommodate new foods through user input

and crowdsourcing easily. The user is also able to manage all entries and organize them as different graphs focusing on chosen nutritional categories by the user. Representative screenshots of this process are shown in Figure 3.5.

Figure 3.5. Demonstration of the process of entering food items and recipes using DDRS smartphone app.

3.5 CHAPTER SUMMARY

This chapter reviewed the overall design of the DDRS system. The high-level topics discussed in this chapter are:

- Overall functionality of the DDRS device
- The relativity of all the DDRS pieces, connection, and functionality.
- Specific design and development of the housing, hardware, circuit, volume estimation algorithm, and smartphone application.

Chapter 4. VOLUME ESTIMATION USING DDRS ALGORITHM

4.1 OVERVIEW OF VOLUME ESTIMATION USING DDRS ALGORITHM

Images of food samples on a plate are first captured using a smartphone, then the images with the smartphone's sensors information are imported to collect the needed data for 3D reconstruction. The food from the image is segmented with saliency mapping auto-cropping, and then a source pixel is located to construct a 3D line segment to a range of depth guesses. Then, the Automatic Dot Detection algorithm locates the laser dots in a segmented image using an HSV filtered cosine similarity mask and a luminance mask. The dot coordinates are then passed to the 3D rotation algorithm discussed in this paper to model the object. The 3D points are first calculated, the 3D image is reconstructed, and finally, the volume is calculated (see Figure 4.1). The following sections will describe the calibration process and the 3D reconstruction algorithm.

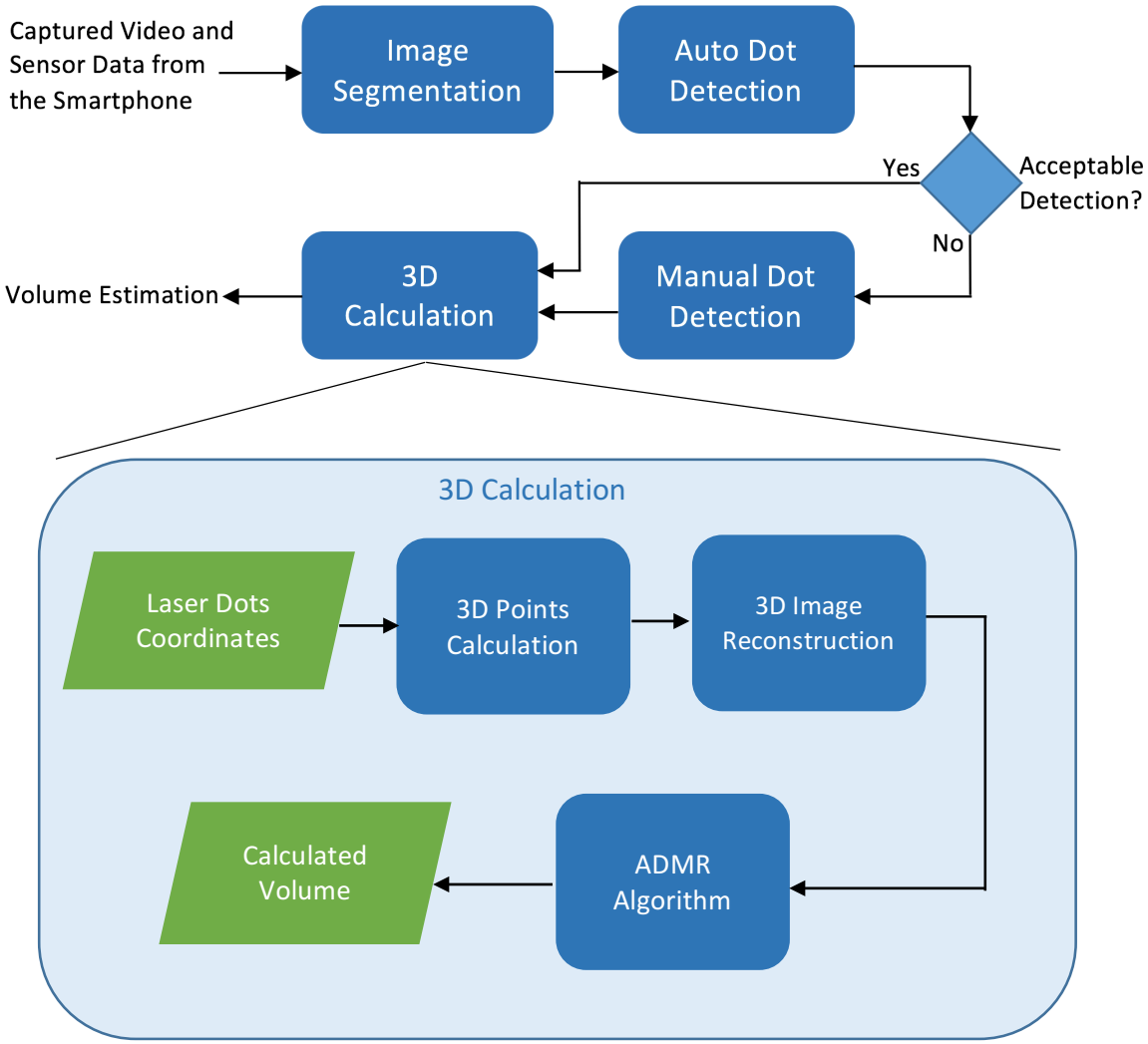

Figure 4.1. 3D Calculation flowchart.

4.2 Laser Module Calibration

The calibration process plays an essential role in the accuracy of the 3D reconstruction. Unlike other calibration methods that consist of finding intrinsic and extrinsic parameters of the camera, this calibration method uses SLS in coordination with lasers to reconstruct the 3D object. The calibration process aims to determine the depth map of a white plane at a known distance. This reference depth map will aid in the construction of the depth map of any object scanned.

To calibrate a DDRS unit, we need to know how the laser dot pattern changes on a flat surface as we move the phone away from that surface. The simplest way to do this is by taking pictures of the laser grid at different distances from the phone. A white planar surface is used to project the laser grid and determine the pixel values of each dot in the image. Figure 4.2 shows the calibration device or DDRS calibrator, which operates automatically and can eliminate any human error. DDRS calibrator consists of robotic mapping system algorithm, which is based on the model of a single axis computer attached to a controlled cutting machine (CNC). A holder on a pair of rails driven by an Arduino-controlled stepper motor.

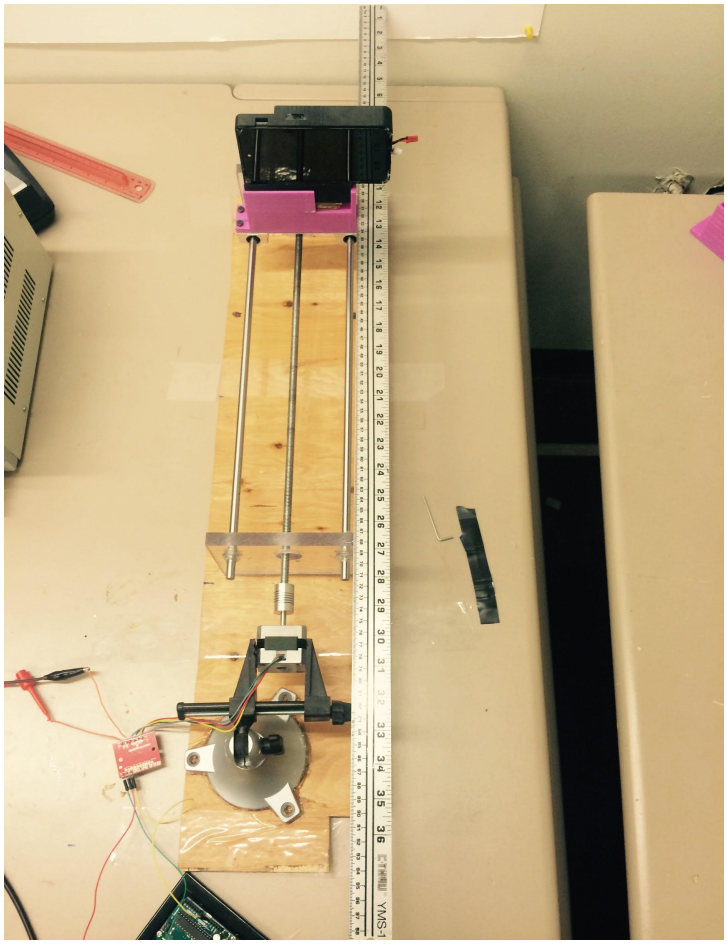

Figure 4.2. DDRS calibrator device.

Since accelerometer data is needed for later processing in the algorithm, a picture, taken by the phone camera cannot be used. However, a picture can be easily extracted from a recorded video. To begin calibration, the phone must start far enough away from the white wall that all the laser dots on the plane can be seen on the screen, shown in Figure 4.3. Then the video recording can begin. While recording, the phone has moved 20 mm away from the plane, and the camera is covered to make the screen dark while the phone is moving. The black screen will serve as a delimiter for each video section related to a distance. This process is repeated by moving away from the plane in 20 mm increments up to 500 mm. After setting up the hardware and recording a video of the white plane at different distances, the algorithm is now ready to be used to process the image and generate calibration data.

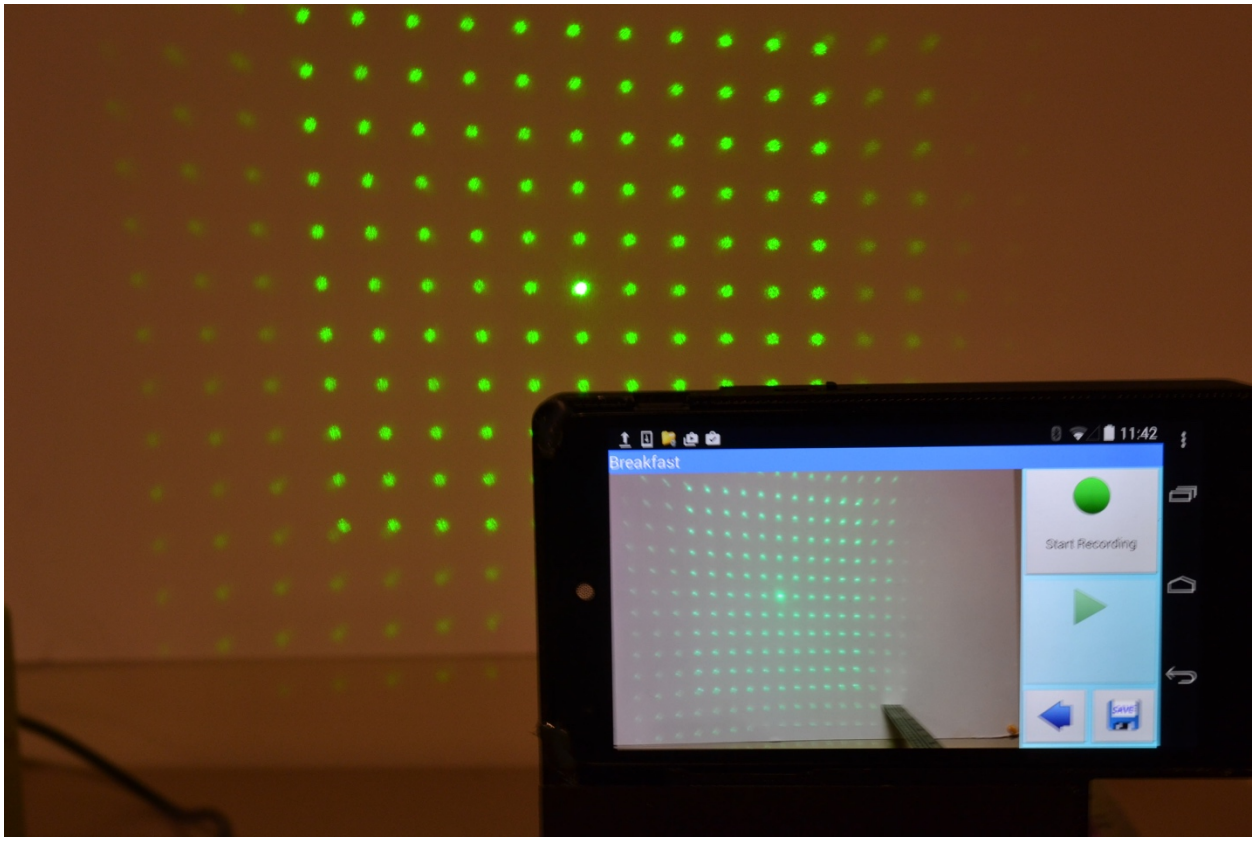

Figure 4.3. A DDRS device in the process of calibration.

The average computational power of the smartphone affords a suitable resource for data collection, but more is required for data processing. Therefore, acquired videos must be transferred to a cloud server for processing. The video sequence will be first extracted to frames, and each frame is used to generate calibration data. Through calibration, a series of parallel and planar images within a valid range of depths are acquired from the camera, each containing a grid of dots. The laser dots' pixel positions in the image change as a function of distance from the image plane, forming a line segment that can be mapped in 3D space. The steps required to generate the calibration data are as follows:

1. Extract frames from the video
2. Locate the center of the laser grid, and index the values of neighboring points
3. Generate calibration data.

4.2.1 *Frames Extraction*

The purpose of this algorithm is to extract frames from the video at varying distances. This process consists of converting video to frames and determines which frames contain the entire laser pattern before the camera was covered. Next, only one image per delimiter is selected, and the results are saved with a name corresponding to the distance values (200.jpg, 220.jpg…. 500.jpg). The selected images will be processed by an algorithm to generate the calibration data.

4.2.2 *Locating the Laser Coordinates and Index Positions*

The rest of the calibration process requires locating the pixel coordinates of all the laser dots in each frame and mapping each laser dot to an index position. This will make it easier to keep track of each dot as the camera moves away from the plane. An algorithm was designed

specifically for locating the coordinates of green laser dots, which processes a single image containing the laser grid. It isolates dots in the image by converting the image to grayscale and using intensity of light to distinguish dots from the image shown in Figure 4.4.

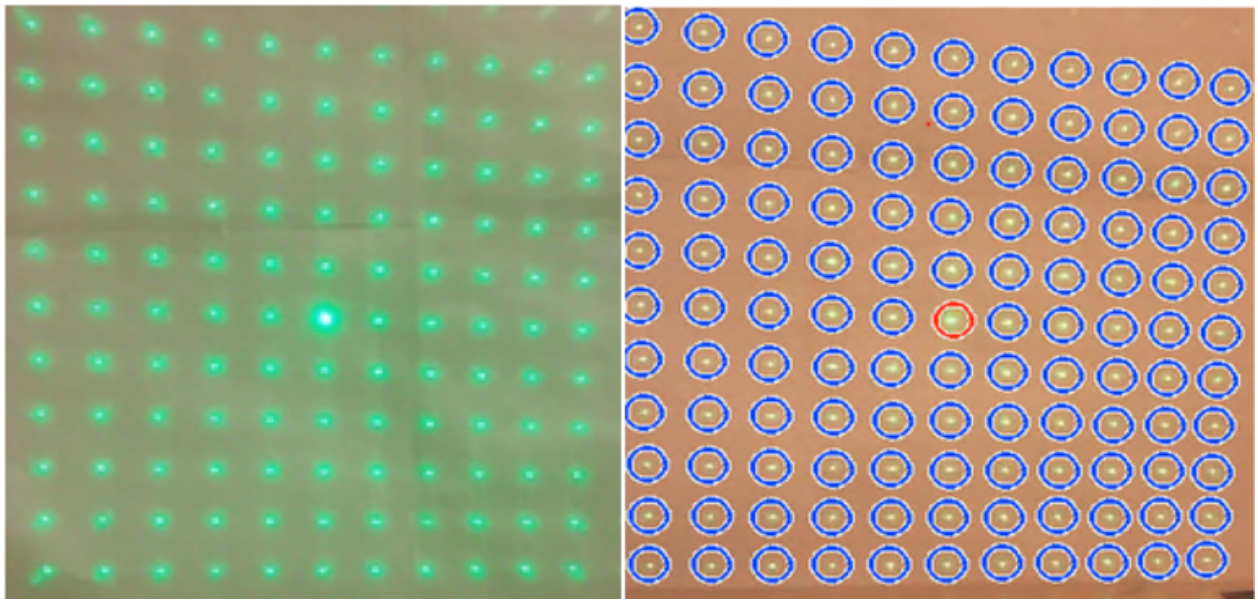

Figure 4.4. A snapshot of the laser point on the left, and a snapshot of the AutoDot detection algorithm processing the laser points for the calibration process on the right.

After extracting all the dots from the images, the pixel coordinates are processed to generate a map that will represent each dot as a field of a two-dimensional array. The idea is to find the distance from each point to the center. The smallest distance will be at the center point, so its assigned index is (0,0). The next four closest points are used to define the four quadrants (-1,0), (1,0), (0,-1), (0,1). Using these quadrants, we can map each point to an appropriate index, which will represent how far each point is located from the center (see Figure 4.5).

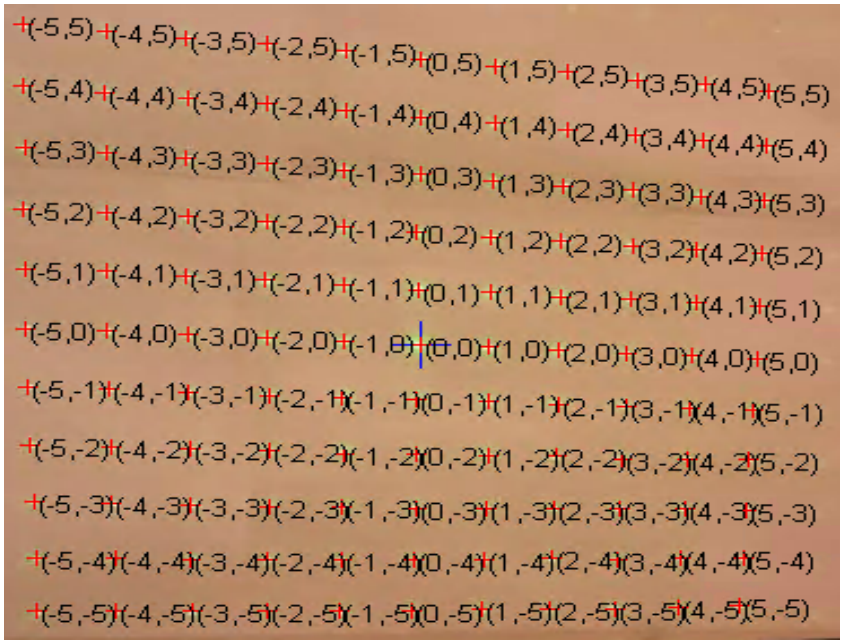

Figure 4.5. All coordinates assigned automatically for the calibration process.

4.2.3 *Generating Calibration Data*

The calibration data is a fundamental component for accurately calculating the position of the 3D points. The precision of the calibration data, especially the linear and nonlinear regression parameters, directly affect the accuracy of the 3D reconstructed shape. Assuming n images are extracted at equally spaced distances from the calibrating plane, the input data for generating calibration data are the sets of distances where frames are extracted, also explained further below in (1, 2, 3).

$$y = xm + b \tag{1}$$

$$Z = [Z_1, Z_2, Z_3, \dots, Z_n] \tag{2}$$

$$Z = \frac{G}{H+x} \tag{3}$$

where y, x, m, and b are the variables for the regression line; Z is the depth coordinate; G and H are the sensor data from the extracted frame.

There are then n structures, each of which has the 121 pairs of x and y values for each laser dot in the image. A single set of 121 coordinates can be visualized in Figure 4.6. The accumulated data is then used to calculate the calibration data.

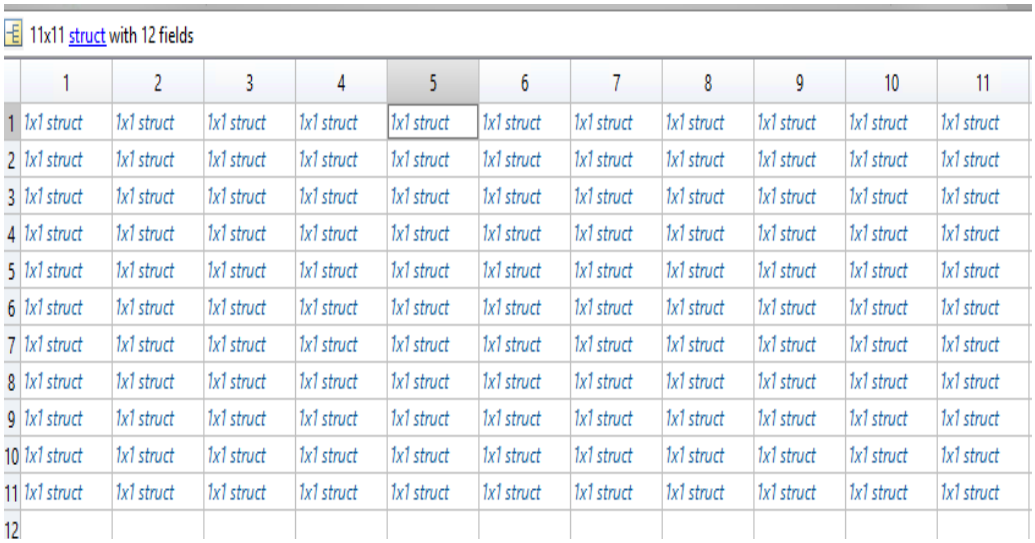

Figure 4.6. Complete set of calibration data.

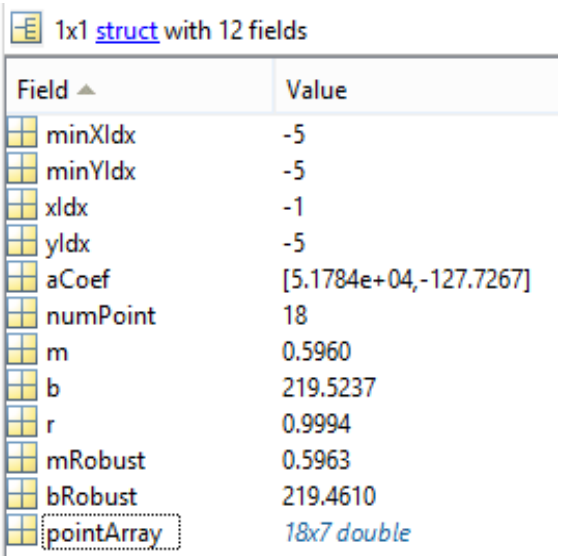

Figure 4.7. Calibration data fields for each laser dot.

4.3 Automating Dot Detection

Our approach of automatic laser dot detection on a segmented image adapts computations of several successive image filtering and masking operations. The primary objective of the algorithm is to find the pixel location of each laser dot on the food in a picture. Given that all of the laser dots used for this application are bright green, we use color properties and mathematical concepts to create filtering and detection methods that work best with green laser dots." The three primary filters implemented for the algorithm are the HSV controlled filter that intensifies the green dots, the cosine similarity masking of two image pairs, and the luminance erosion of the object image. Figure 4.8 shows the high-level block diagram overview of the Automatic Dot Detection, which consists of four main functions: HSV filter, Cosine Similarity Luminance Erosion, and Mask Merge.

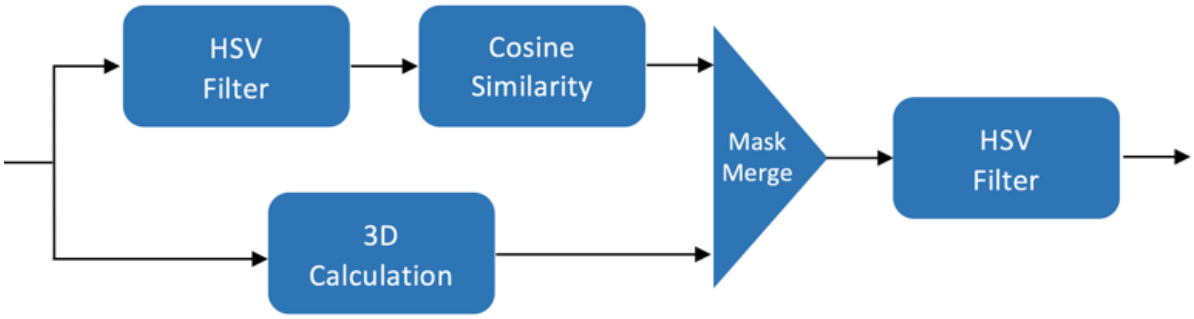

Figure 4.8. The automatic dot detection algorithm flowchart.

4.3.1 *HSV Filter*

The first filtering technique applied to the segmented image is an HSV filter that intensifies the brightness of green pixels in the image. This is a necessary step since the cosine similarity mask (detailed in the next section) relies on the difference between the green pixels and all other pixels in the image.

To accomplish this, we first use the image to convert from the RGB to the HSV domain. In the HSV domain, two separate filters are created: one filter mask that uses the hue component of the image to identify colors of the pixels, and another filter mask that uses the saturation component to distinguish color intensities of the pixels. As the hue component provides distinguishable color information, saturation provides comparable color intensity between each pixel; the two components are combined to create an HSV filtering mask that intensifies the color of the bright green pixels, isolating them from the rest of the image. The image below illustrates the relationship between color information and intensity in the HSV domain. Figure 4.9 shows a scanned croissant after HSV filtration, which enables the green laser pixels to be visible for detection—the noise will be addressed in the next section.

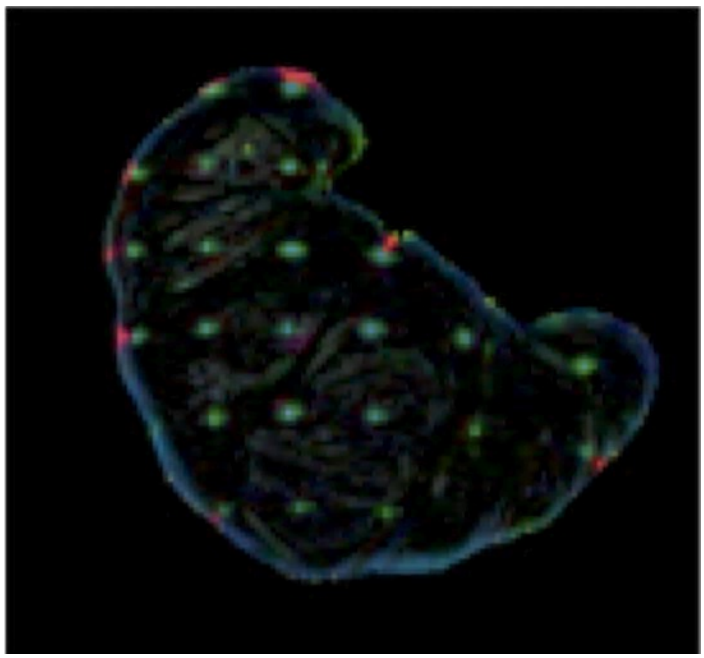

Figure 4.9. Processed image of a croissant after HSV filtration.

The hue filter is applied to mark only pixels that are identifiable as green pixels from the image. This filter is then combined with a saturation filter that marks only the pixels with relatively

high color intensities, which are the green laser dots in this particular application The combination of these two filters identifies regions of the image where the green pixels are located. These areas are highly intensified, whereas the rest of the image remains the same; this mainly highlights the laser dots placed on the food while the remainder of the image does not change. The computations used for calculating the hue and saturation values utilized for this filter are shown in equations (4) and (5).

$$GH = 90° \leq PH \leq 180° \tag{4}$$

$$LCI = PS \geq 90\% \tag{5}$$

where GH is Green Hue, PH is Pixel Hue, LCI is Laser Color Intensity, and PS is Pixel Saturation.

Since laser dots tend to be the most saturated color in the image, the green image pixels are defined by the hue values between 90 and 180 degrees, and with a color intensity greater than 90%. Once the HSV filter is applied, the image is converted from the HSV domain back to the RGB domain to be processed by further filtering and masking techniques.

4.3.2 *Cosine Similarity Mask*

After the HSV filter has been applied, the image shows the laser dots with more intensity. At this stage, cosine similarity masking is used to isolate the laser dots further in the image.

The approach begins by removing most of the background of the image (non-laser dots). This process is mathematically computed by determining the similarity of every pixel in the image to the determined color of the laser dots (filtering mask image). The equation for this similarity is known as cosine similarity, denoted by the equation (6):

$$S = \cos(\theta) = \frac{A \times B}{[[A] \times [B]]} \tag{6}$$

where S is Cosine Similarity, A is, B is, and θ is the angle of the device held at that moment.

As a starting point, the similarity of each pixel with respect to the color of pure green (RGB value of 0/255/0) is determined using (6). This can be visualized as a 3D projection of two vectors in RGB domain on a color plane, which can be illustrated as shown in Figure 4.10.

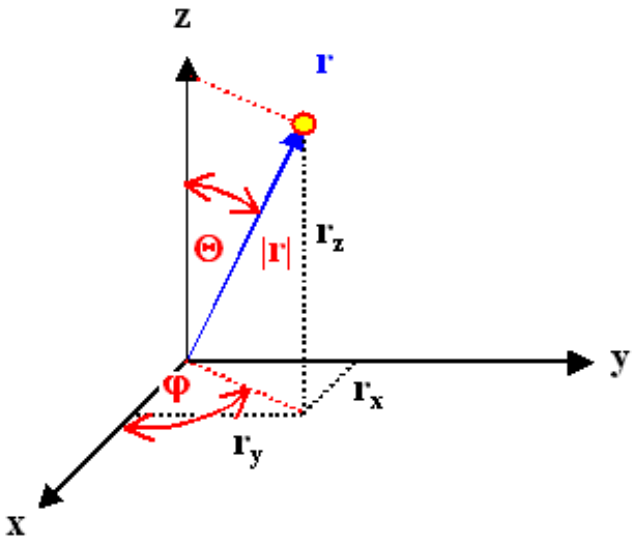

Figure 4.10. Projection of a vector *r* onto the color plane with *x, y,* and, *z* components.

The computation and thresholding of similarities between two images create a binary mask that only marks pixels that are similar in both pictures (the bright laser dots). This mask is then applied to the original image to eliminate all pixels outside a range of similarity to the color of pure green (0, 255, 0). The average RGB values of the remaining pixels are calculated (the average of green laser dots) and the similarity mask is readjusted with a color similar to the average laser dots on the image. Depending on the brightness of the image and the reflection of the dots off the object, an iterative process may be executed to determine the color of the dots dynamically. Figure 4.11 shows two iterations: one with a value of pure green, and the second with the adjusted value.

Figure 4.11. (a) Original image (a croissant), (b), (c) two iterations of cosine similarity using thresholds.

For the iterative process to function, the result is required to converge to a particular value. Hence, the threshold value must be set to such a value that eliminates as much noise as possible while preserving the pixels of the actual laser dots.

The value that is selected for the threshold must be chosen carefully; if the wrong value is selected, the output does not converge. Therefore, a value with a low threshold should be selected to help filter out noise in the image.

Creating a dynamic threshold would be ideal here, but it does not guarantee that we will be able to isolate these dots in different image types. Hence, images, which contain objects with high contrast to the dots, would require additional maskings such as cosine similarity and luminance (which are both detailed further in the next section).

4.3.3 *Luminance Mask*

Since applying the cosine similarity mask is not enough to isolate the dots in the image, a combination of the cosine similarity mask and another mask, called a luminance mask, is used to achieve more accurate results. The two masks are applied to the original image separately and then combined (multiplied) to produce a final mask with highly accurate dot locations. The luminance mask takes into account the intensity of the light in the image to isolate the dots.

This luminance filter begins by first isolating the background of the image, which in this case includes everything that is not a laser dot. This is accomplished by performing a morphological opening operation using a structuring element of a disk. In this scenario, morphological opening means the dilation of the erosion of a set A by a structuring element B. Equation (7) shows this operation.

$$A \circ B = (A \ominus B) \oplus B \tag{7}$$

where A is the image, B is the structuring element of a disk; $\ominus$ represents erosion and $\oplus$ represents dilation.

This calculation leaves us with the image background, and with some noise of the original laser dots. This background is then subtracted from the original to isolate the foreground, which at this stage consists of the laser dots, the border of the object, and a small amount of noise.

With the foreground now somewhat isolated, the next step is to reduce the overall noise in the image. This is done by converting the image to a luminance matrix, and thresholding the acceptance value at a certain cutoff to retain the brightest parts of the image. The resulting image contains laser dots and some of the border but does not remove the noise that consists of darker pixels. Equation (8) shows how the luminance is calculated.

$$L = 0.2126R + 0.7152G + 0.0722B \tag{8}$$

where L is Luminance, R is red color value, G is green color value, and B is blue color value.

This equation represents the luminance using the BT.709 HDTV standard for luminance [19]. The resulting mask of this stage removes noise from the image while keeping the laser dots, and some points on the border. Although the image is mostly isolated at this stage, there are still values at the border that are detected as dots. To fix this, the border must be eliminated through further application of image erosion.

4.3.4 *Eliminating Border and Applying Masks*

At this point, the image left is an isolated image containing the laser dots as well as part of the border. To isolate the laser dots, image erosion is applied with a structuring element of a small size to remove a thin outside edge on the border. This removes the entire border and leaves just the laser dots in the image. Since the structuring element size is static, some dots on the border might be eroded and not located. Figure 4.12 shows the application of a luminance mask to the original image. The final result is shown in the right-most image (c) of Figure 4.12.

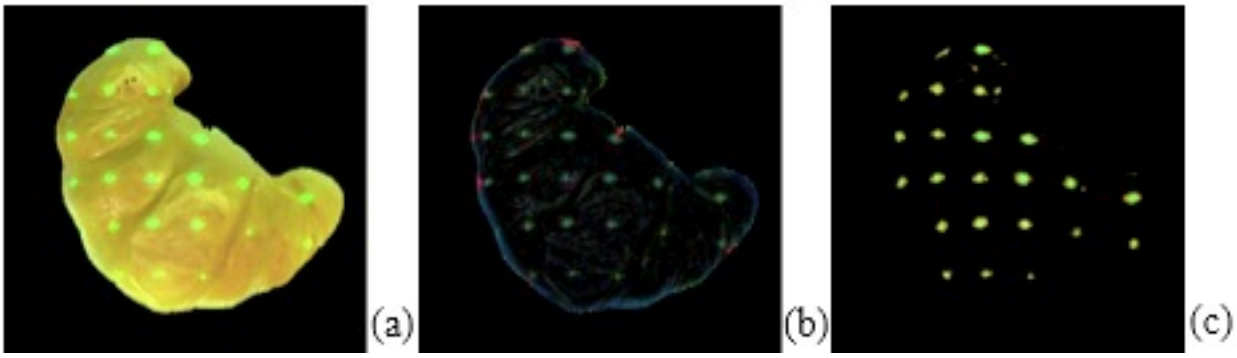

Figure 4.12. Figure 19. Original croissant picture (a), croissant picture with just the border (b), and croissant picture after the luminance mask is completely applied (c).

This step completes the luminance mask. At this point, both the cosine similarity and luminance masks can be combined. Each mask removes portions of the image that do not represent laser dots. Therefore, the masks can be multiplied by each other, and the overlapping area is a very close representation of just the laser dots. It is important to keep the threshold for each mask relatively low, as it can be expected that the other mask will remove extreme differences through an alternate method.

4.3.5 *Eliminating Reflections*

With the combination of the masks, there is now an accurate representation of the laser dots in the image. One additional thing that must be accounted for, however, is the imperfections of the hardware.

When using the laser, there are sometimes reflections from a single dot that appears on the image as three dots very close to each other. To achieve an accurate 3D reconstruction of the object from these points, it must be ensured that only the actual dot is recognized. The dots are stored for initial reflection analysis as objects with a size and an x and y position inside of an array.

To find these reflections in the image, the array is iterated through to allow comparisons of the sizes and locations of the dots. If the dots are less than a pre-defined distance from each other, it must be decided which dot is valid, and which is a reflection, by comparing the sizes and removing the smaller of the elements from the array. This leaves an array where no dots are within a certain distance from each other, ensuring no laser points in the array are reflections.

Another highly important aspect of this dot detection algorithm is locating the center dot, which is critical for the 3D reconstruction. The hardware being used for this particular project has a dot with stronger intensity, and that is larger in size at the center of the laser array. For the process to work properly, the center dot must appear on the object itself since all other points are relative to it. Given this assumption is met, the largest dot in the array can be selected as the center.

4.3.6 *Final Error Catching*

Although the algorithm by itself is highly accurate, since it is also critical that the 3D calculation algorithm gets the correct values, we added the Manual Dot Selection algorithm to help users manually add or remove laser dots in the image that might not have been captured. This feature includes a GUI that allows the user to add dots to the image not caught by the algorithm,

as well as removing false positives detected in the image. This final error-catching step can be seen in Figure 4.13.

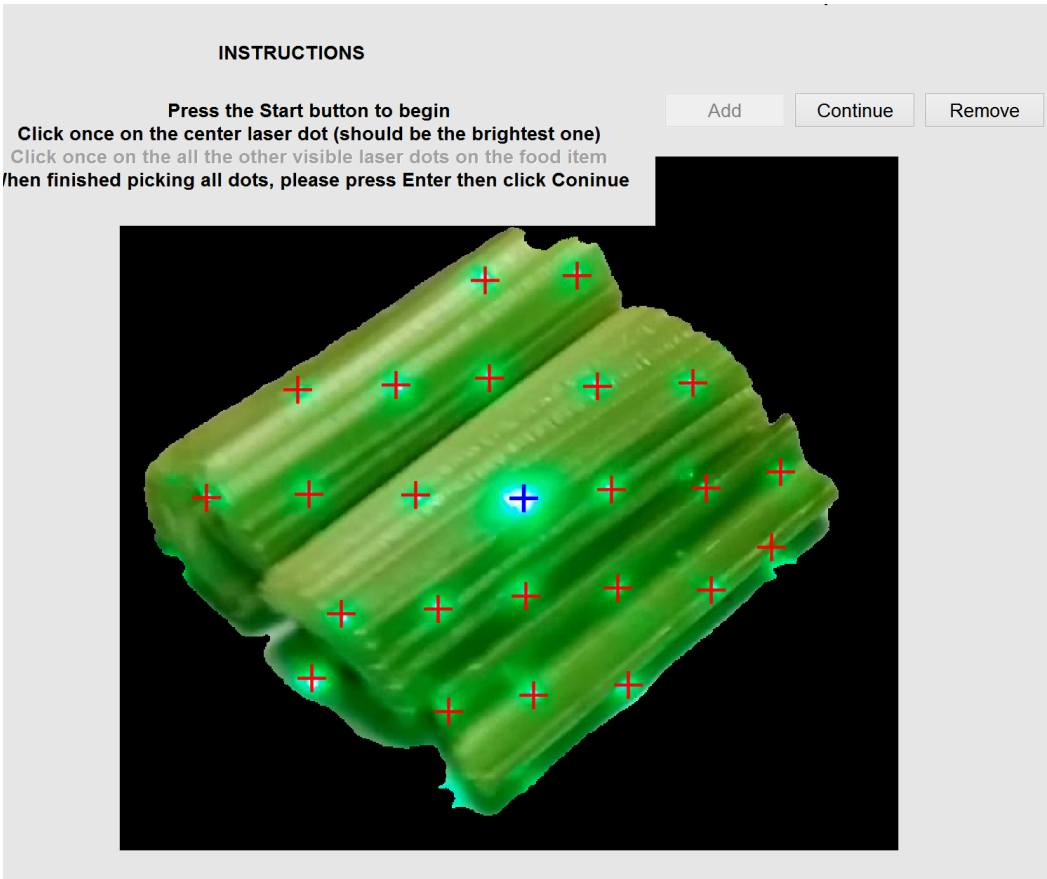

Figure 4.13. The final step, the Manual Dot Selection GUI (which in this case shows some celery), corrects the automation algorithm's errors.

Figure 4.13 shows the manual dot selection GUI with its add, remove, and continue buttons. The red cross points on the image are the regular laser dots, while the blue cross point represents the center dot.

4.4 Image Segmentation

Our current approach utilizes Saliency Detection via the Graph-Based Manifold Ranking Algorithm with the addition of automatic image cropping to increase the efficiency of the algorithm and to automate image segmentation by locating just food on a plate [20]. Since researchers at Fred Hutchinson Cancer Research Center are currently the primary users of this application, this algorithm is tailored to work best with foods placed on the white or brightly colored plates used at the facility. Since many household plates are also white, this application could have a commercial use as well. Figure 4.14 shows an overview of the stages of the algorithm.

Figure 4.14. The segmentation algorithm flowchart.

4.4.1 *Auto-Cropping*

To utilize saliency detection for image segmentation in a fast and accurate manner, the image of the food should ideally only contain the object of interest [21]. The goal of this stage is to reduce the image to be bounded by just the plate containing the food, eliminating the background of the image such as the table.

The first step is to take a picture of the entire plate of food, which can contain much more background than just the plate and the food itself. Once this is taken, plate cropping is made possible by the fact that the plate is brighter than its background.

The image is first converted into the grayscale domain by applying weighted sums. This is computed using the RGB weighted conversion formula, shown in (9),

$$GS = 0.2989R + 0.5870G + 0.1140B \quad (9)$$

where GS stands for grayscale, R is the value of the red pixel, G is the value of the green pixel, and B is the value of the blue pixel.

The conversion equation is based on the SDTV standard BT.601. A thresholding value is then set to separate pixels that are considered to be bright (the plate) from dim (background) pixels. This creates a binary image that works as a mask. At this stage, several noise-reduction steps are applied to improve the image. First, to ensure that the entire plate is captured, we fill in all of the holes in the binary image. Using the previously created mask, black pixels enclosed by connected borders of each object are filled in. This is done to mask the bright objects in the image completely. The mathematical computation of this process is shown in Figure 4.15.

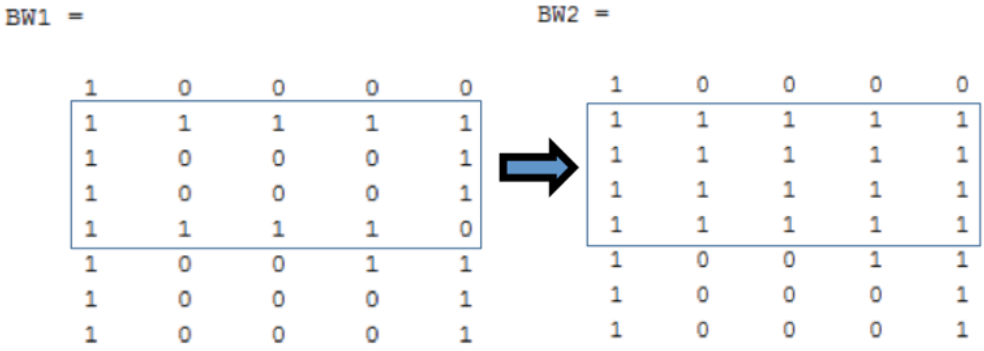

Figure 4.15. Filling all image binary holes to reduce noise (Binary Masking).

These existing masks represent all of the bright individual objects in the picture, whose areas are calculated by summing the connected pixels. The object with the largest area is isolated, and all others are removed. This isolated object is assumed to be the plate, as it is expected to be the main focus of the image.

With this information, the boundaries of the plate can be located and extracted from the image. This is done by finding the minimum and maximum indices in both the horizontal and vertical

dimensions, which we use to resize the image and bound the image size to the plate of food. Figure 4.16 shows the steps of this process.

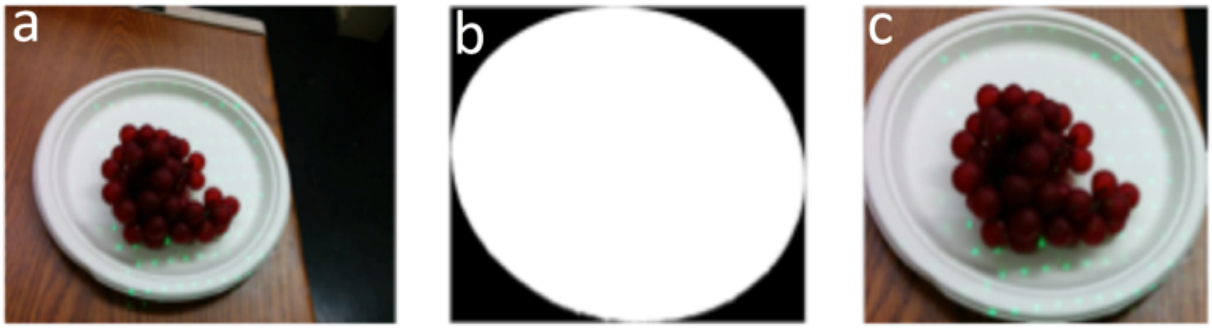

Figure 4.16. The transition of a plate of grapes from the original image (a), to binary mask image (b), to the minimized image bounded by the plate (c).

In step c, the size of the image has been significantly reduced. However, it still has more pixels than just the food itself. Since the salience mapping function requires the image to be tightly bounded by just the food to work, the image still needs to be cropped further.

4.4.2 *Secondary Cropping Bounded by Food*

With the first iteration of the image cropping completed, the image is reduced to be strictly bound to the plate as shown in Figure 4.16(c). We then perform a secondary cropping using image erosion, analyzing the image in the HSV domain to reduce the image to just contain the food [22]. The first step of this is to apply a binary erosion to the image in the grayscale domain. The binary erosion of an image A with a structuring element B is defined as the set operation shown in (10):

$$A \ominus B \equiv \{Z | BZ \subseteq A.\} \tag{10}$$

where A is the binary erosion of the image, B is the structuring element, and Z is the pixel location.

This is the set of pixel locations Z, where the structuring element translated to location Z overlaps only with the foreground pixels in A. The gray-scale erosion of $A(x, y)$ by $B(x, y)$ is defined in (11):

$$(A \ominus B)(x, y) = \min[\{A(x + x', y + y')] - B(x', y') | (x', y') \in D_E\} \quad (11)$$

where D_E is the domain of the structuring element B and $A(x, y)$, and is assumed to be $+\infty$ outside the domain of the image [23].

The structuring element does not need to be a specific shape, but it must contain the food and background to reduce the segmentation threshold value. The result of this erosion provides the backdrop of the image [24]. This background is then subtracted from the original image to remove as much of the background (the plate and table) as possible. We apply thresholding again to highlight just the significant borders of the object and to remove as much of the noise surrounding the food. The holes in the binary image from the thresholding are then filled in, and the remaining noise is removed from the image. This creates a rough segment that is not meant to bound the image to just the food but reduces the image's size yet again for further processing.

The next step in the algorithm is to apply a filter to the previously segmented mask, which is done to locate the food on the plate to enable further segmentation. To use this filter, the image is converted to the HSV domain. The process we use to convert from RGB to HSV is shown in (12) – (20)

$$R' = \frac{R}{255} \quad (12)$$

$$G' = \frac{G}{255} \quad (13)$$

$$B' = \frac{B}{255} \quad (14)$$

$$C_{max} = \max(R', G', B') \tag{15}$$

$$H = \begin{cases} 0°, \ \Delta = 0 \\ 60° \times \left(\frac{G'-B'}{\Delta} mod 6\right), \ C_{max} = R' \\ 60° \times \left(\frac{B'-R'}{\Delta} + 2\right), \ C_{max} = G' \\ 60° \times \left(\frac{R'-G'}{\Delta} + 4\right), \ C_{max} = B' \end{cases} \tag{16}$$

$$C_{min} = \min(R', G', B') \tag{17}$$

$$\Delta = C_{min} \tag{18}$$

$$V = C_{max} \tag{19}$$

$$S = \begin{cases} 0, \ C_{max} = 0 \\ \frac{\Delta}{C_{max}}, \ C_{max} \neq 0 \end{cases} \tag{20}$$

where H, S, and V are each the RGB value in HSV (Hue, Saturation, Value) domains. R', G', and B' are the pixel values calculated from the original RGB data. C_{max} and C_{min} are the thresholding values to distinguish the background from the image itself.

At this stage, the image can be decomposed and analyzed in each of the three domains (hue, saturation, value). The image is considered in the saturation domain if the food contrasts significantly with its background. In this case, we use a white background to fulfill this goal. By applying a thresholding value, the food itself can thus be separated from the plate and background that remains after segmentation.

This results in a binary image with the food outlined and significantly less background noise. At this stage, the first automatic-cropping process can be applied to this binary image. This is done to bind the image tightly to the object and reduce the image size before passing it to the Manifold Ranking Saliency Detection function. Figure 4.17 below shows the subsequent cropping process given the segmented images from the first crop. The image on the far right is the fully cropped image, which will be segmented by the saliency mapping—this is necessary because there is still a small, yet visible, amount of noise surrounding the food.

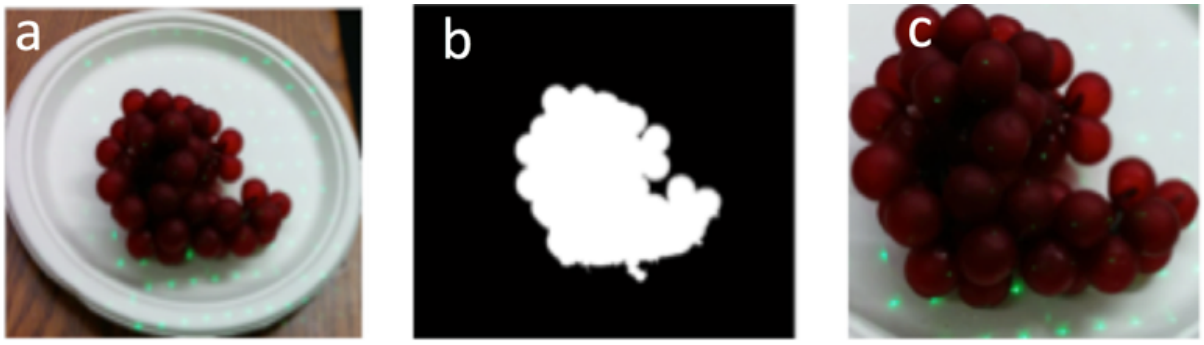

Figure 4.17. The transition of a plate of grapes from the segmented image (a), to binary mask image (b), to minimized image bounded only by the food (c).

4.4.3 *Manifold Ranking Saliency Detection Function*

The image is now fully cropped and ready to be mapped by the saliency algorithm. Since the image size has been minimized, the runtime of the saliency function will be improved significantly over running it on the original image.

We now apply the graph-based Manifold Ranking Saliency Detection function on the fully cropped image. The saliency is found by ranking the similarity of image elements with foreground and background cues. This is represented as a close-loop graph with super pixels as nodes [20].

This algorithm returns a saliency map image, which is used in the final step of the segmentation process.

The saliency map removes most of the additional noise the original cropping process may have missed. Since the saliency map is a grayscale image rather than a binary image, a thresholding value must be computed again to segment just the food on the plate [26]. The original image, the saliency map, and the final output result are shown in Figure 4.18.

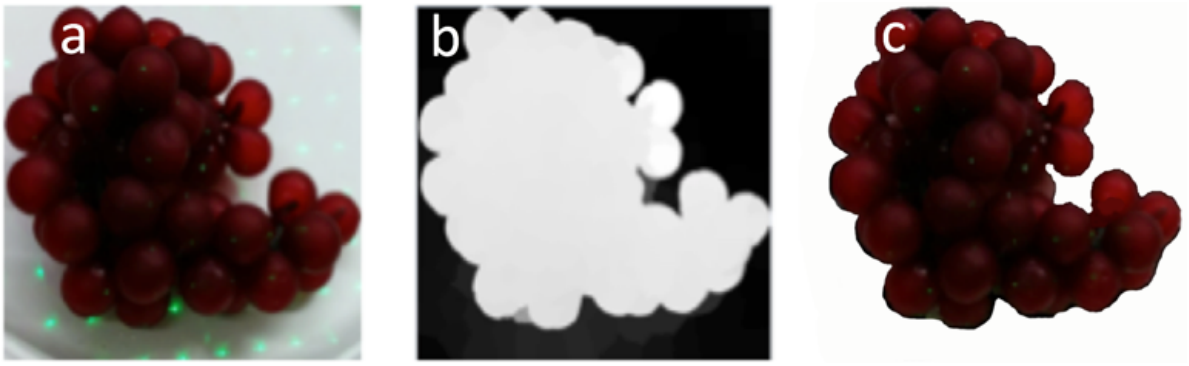

Figure 4.18. The transition of a plate of grapes from the fully cropped image (a), to a saliency map (b), to a completely segmented image (c).

4.5 Results and Analysis

4.5.1 *Automatic Dot Detection*

This section presents and analyzes the current results of running this algorithm as presented in the approach section. The images below show the results of dot locations on a small subset of sample images used for initial testing.

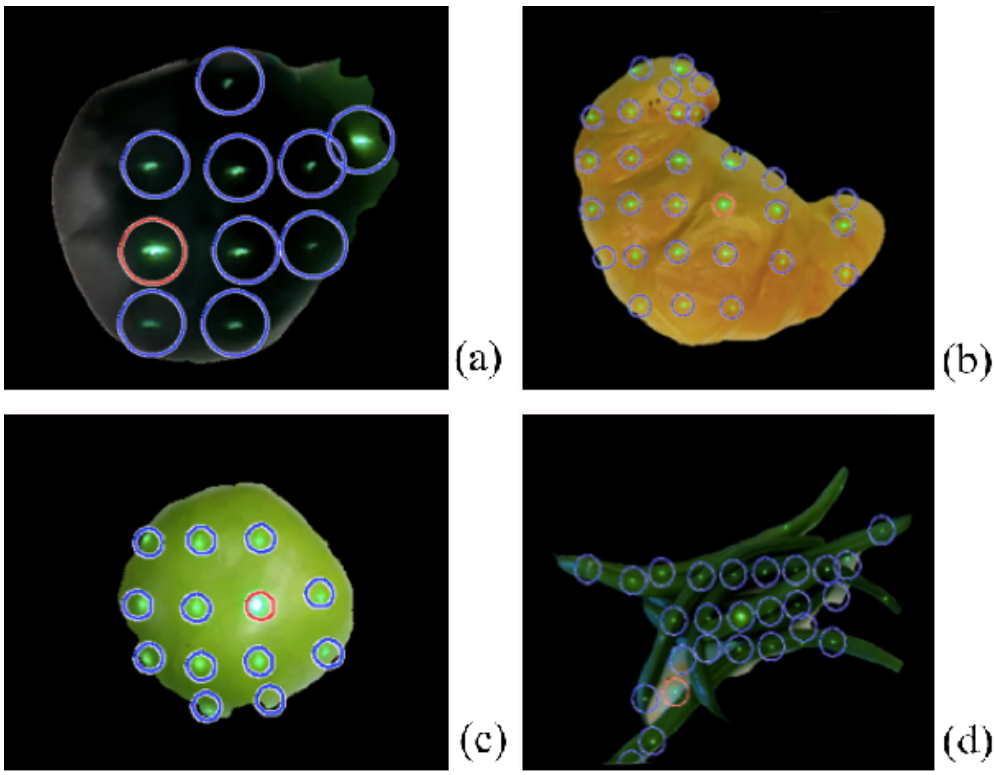

Figure 4.19. Preliminary results of the algorithm on (a) a pear, (b) a croissant, (c) an apple, and (d) green beans.

As can be seen in Figure 4.19, the dots are detected accurately when there is an apparent contrast between the dots and the color of the food. Some of the limitations are visible in (d), where several dots were not detected on the border.

The failure to detect dots on the border is due to the limitation of the static variables discussed in the analysis section. To confirm improvements in both speed and accuracy, several tests were run to compare the current algorithm to previous versions. The experiment was conducted on a group of objects using AutoDot detection, and then the number of dots correctly detected was compared against the actual number of dots on the object recognized by a user.

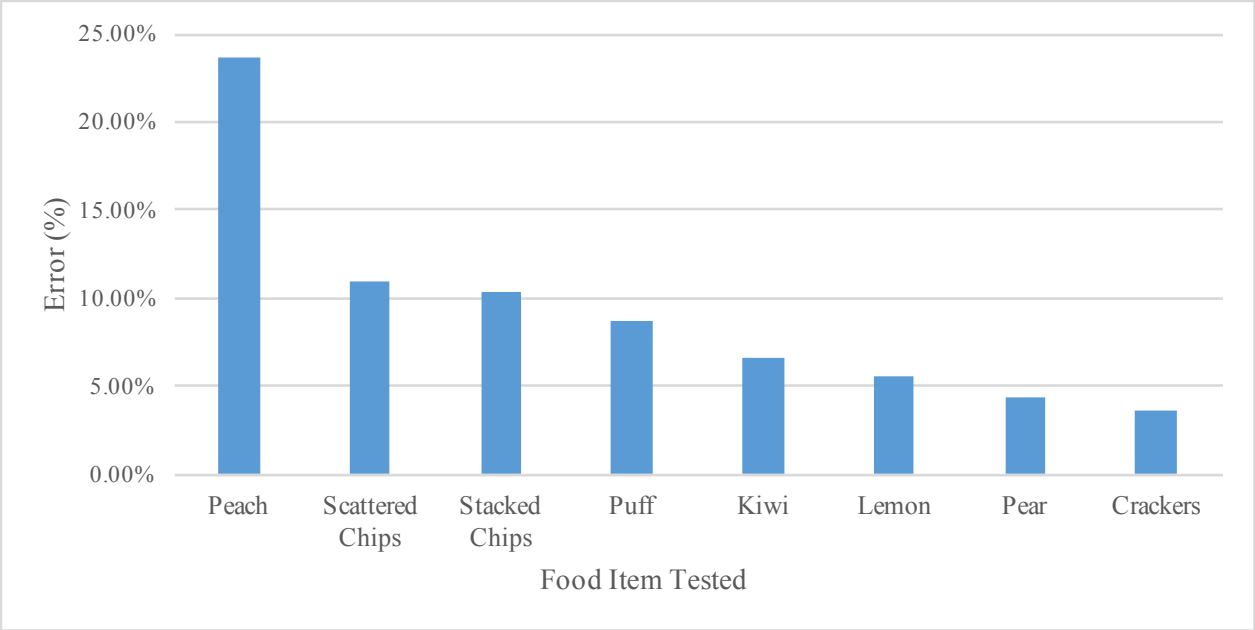

Figure 4.20. Percentage error of the number of dots detected compared to the manual dot detection.

According to Figure 4.20, peaches have the highest percentage error among all tested objects. This is a result of the color, which enhances the green color of the laser dots. Also, the error percentage for stacked chips and scattered chips are also relatively high. One possible reason is that the shape of the chips is very irregular compared to other measured foods. Many of the dots on the edge of the segmented image are not correctly recognized.

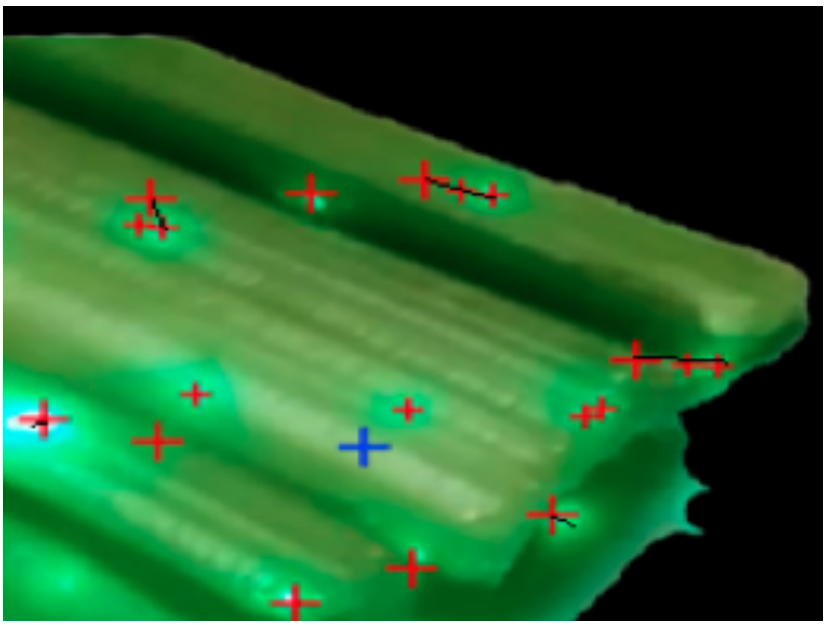

Figure 4.21. A bunch of celery; the black lines portray deviation of auto-detected laser points.

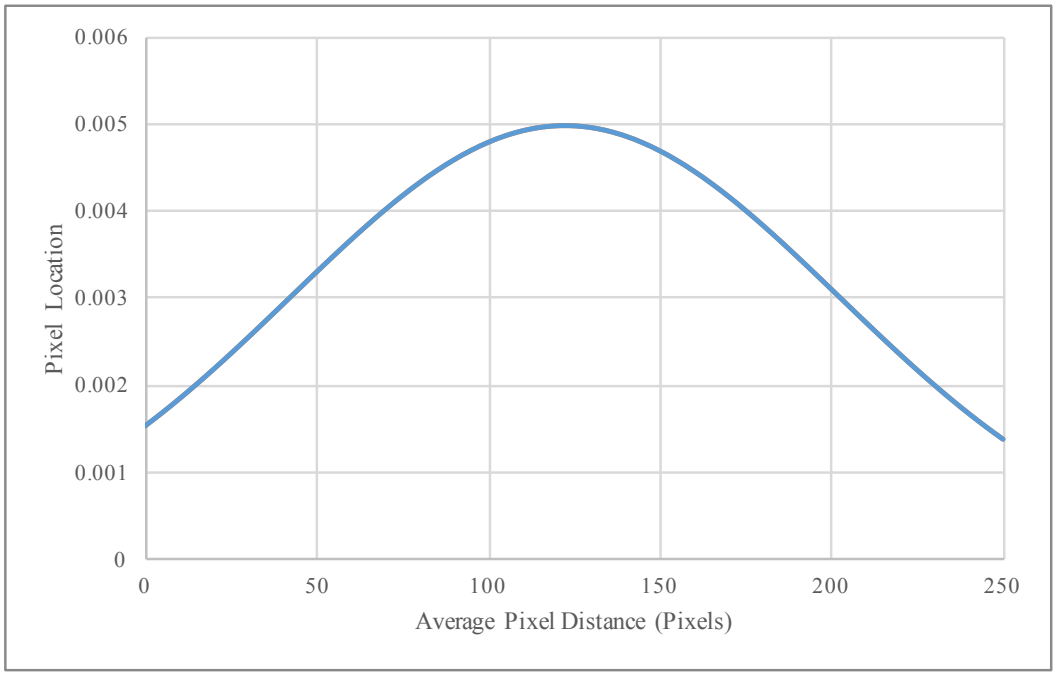

Figure 4.22. A sample of 12 different pictures presents the average pixel distance of the location calculated by the algorithm versus the actual location pinpointed by manual entry.

Figure 4.22 shows the average pixel distance of the location computed by the algorithm versus the actual location pinpointed by manual entry, which has a mean of 123 pixels distance over 12

different pictures of food. Figure 4.21 displays the definition of the “distance” plotted in Figure 4.22. Three images were accurate with 0 distance error. In images of 1920x080 resolution, a 123 pixels difference is a difference of 6.4% compared to the length of an image. Reasons for this 6.4% deviation might include that the surface of food scatters different amounts of light, or that the food’s color makes it hard to distinguish it from the green laser dot. However, to understand the performance of AutoDot algorithm we have to look at the final results (volumes of multiple food items) while comparing with the actual measured volumes.

Figure 4.23 shows the performance of the DDRS algorithm by comparing the calculated volume using AutoDot detection integration and measured volume using water displacement technique for high accuracy. All the items presented in Figure 4.23 have been computed using only the AutoDot algorithm. The overall percentage error for the items tested in Figure 4.23 is around 11%; however, these tests were conducted by ourselves in the lab using engineering students. To test the actual performance of the algorithm, we will need to conduct a larger study to evaluate all possible scenarios. The detailed analysis of the results presented in Figure 4.23 will be discussed in our next publication, Dietary Assessment Using 3D Reconstruction.

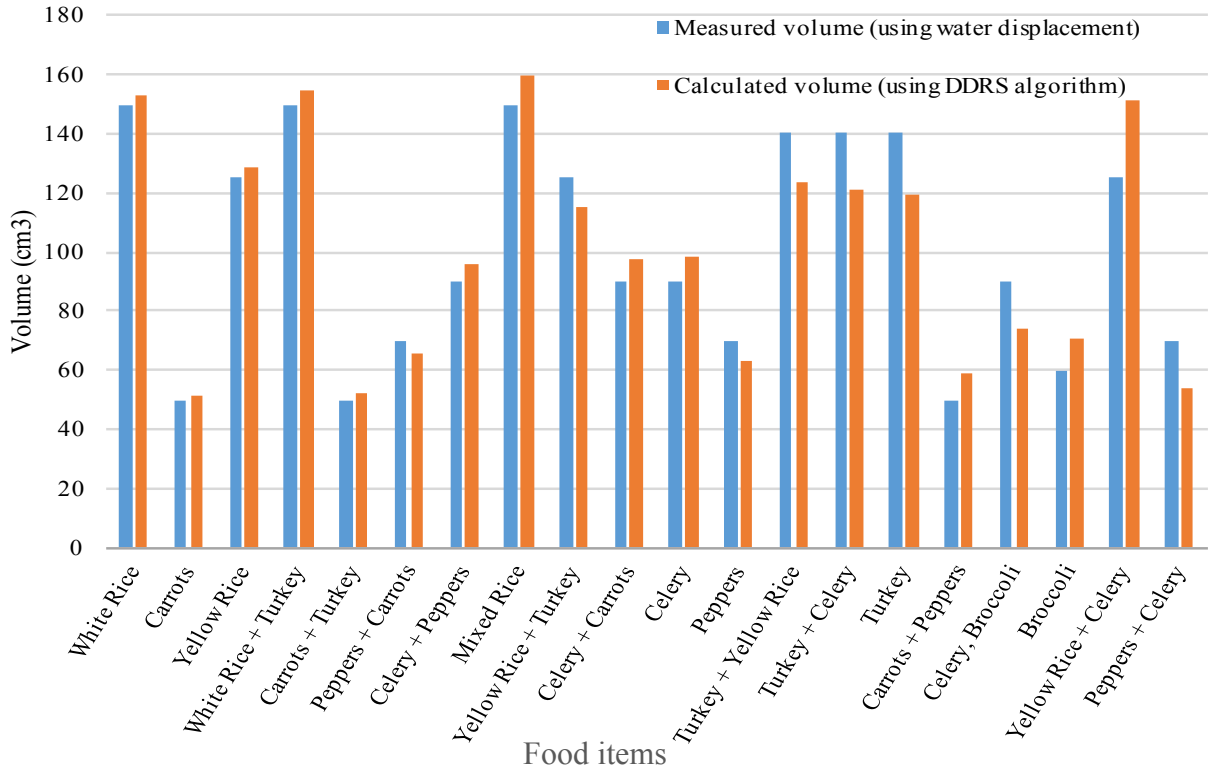

Figure 4.23. Measured volume using DDRS algorithm vs. calculated volume using water displacement.

4.5.2 *Image Segmentation*

In this section, the current status and results of the Segmentation algorithm will be described and analyzed. The algorithm is currently implemented as described throughout this chapter. The initial image crop bounds the image to the white plate, the second image crop bounds it to the object of interest, and we use saliency mapping to create an entirely segmented image on top of a black background [28]. The input and output pairs in Figure 4.24 demonstrate the results of running this algorithm on several test images.

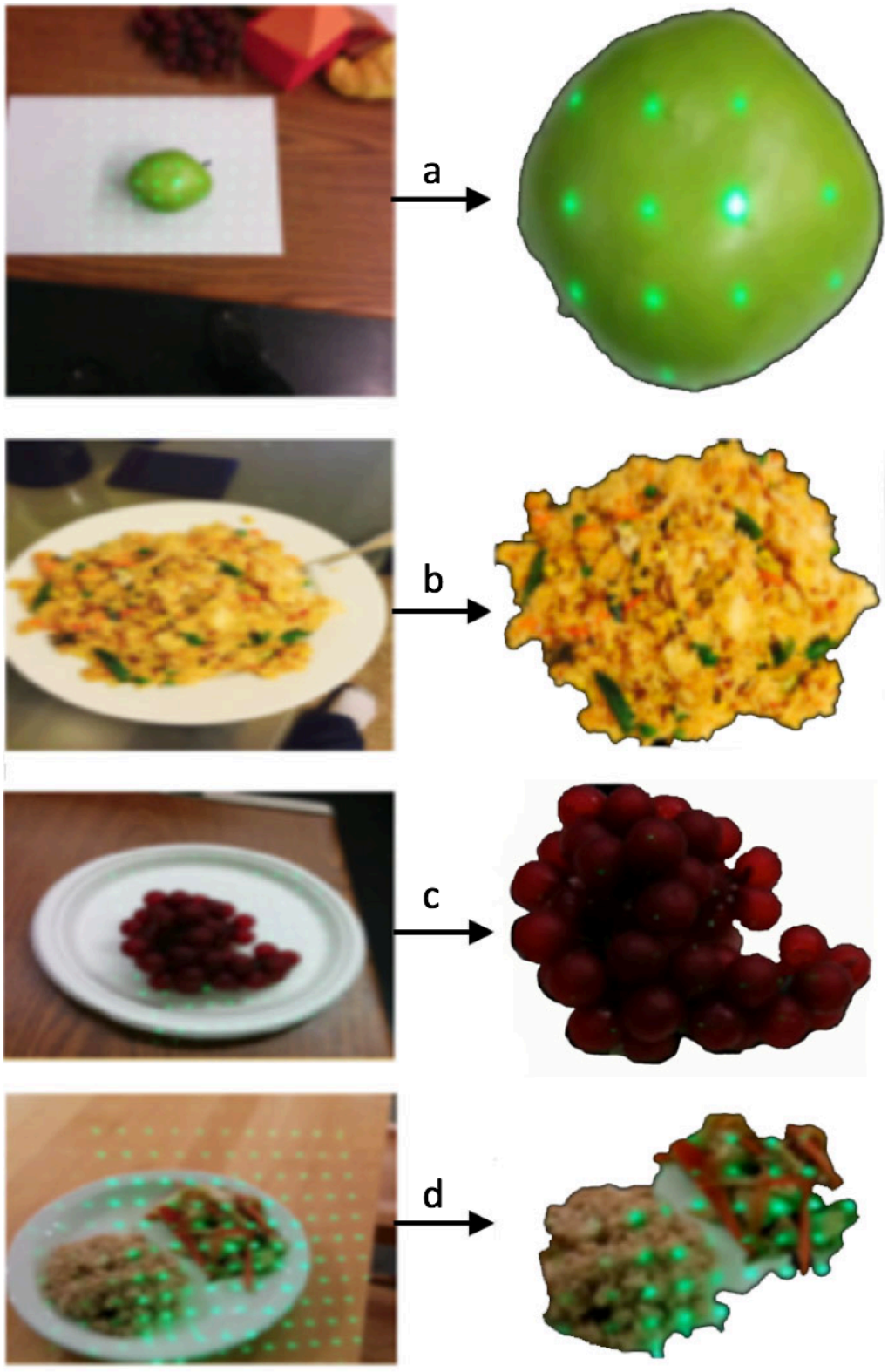

Figure 4.24. Demonstration of the algorithm's performance on multiple food items with different color and shapes: a) green apple, b) yellow rice, c) grapes, d) white rice and red peppers.

The images in the left column of Figure 4.24 show the original samples from a video taken by a user. These images contain no pre-processing work, and, as a result, contain a lot of background noise—including the plate itself [29]. The right column of Figure 4.24 represents the final output of the segmentation algorithm for each food item.

As described previously, the algorithm is highly accurate for images with single food items—as shown in (a, b, and c) of Figure 4.24. However, in the last item (d) of Figure 7, an error is clearly visible. Despite this, we have further optimized the accuracy of this algorithm using linear optimization to aid in segmentation cases with multiple objects.

To accurately measure how precise the saliency mapping approach is in the segmentation algorithm, we tested many different items, allowing comparisons of the calculated area to the area value yielded by an accomplished open source algorithm called Simple Linear Iterative Clustering (SLIC) Superpixel Segmentation [30]. This algorithm was chosen to test against the DDRS segmentation algorithm because it has been thoroughly evaluated, producing accurate results that we can use to determine the accuracy of the DDRS algorithm.

First, we will look at the results from testing single objects on a white plate. Figure 4.25 shows the accuracy percentage of the absolute difference between the calculated segmented area using SLIC Superpixel Segmentation and measured segmented area using the DDRS algorithm. The overall average error percentage for single items on a plate is 5%.

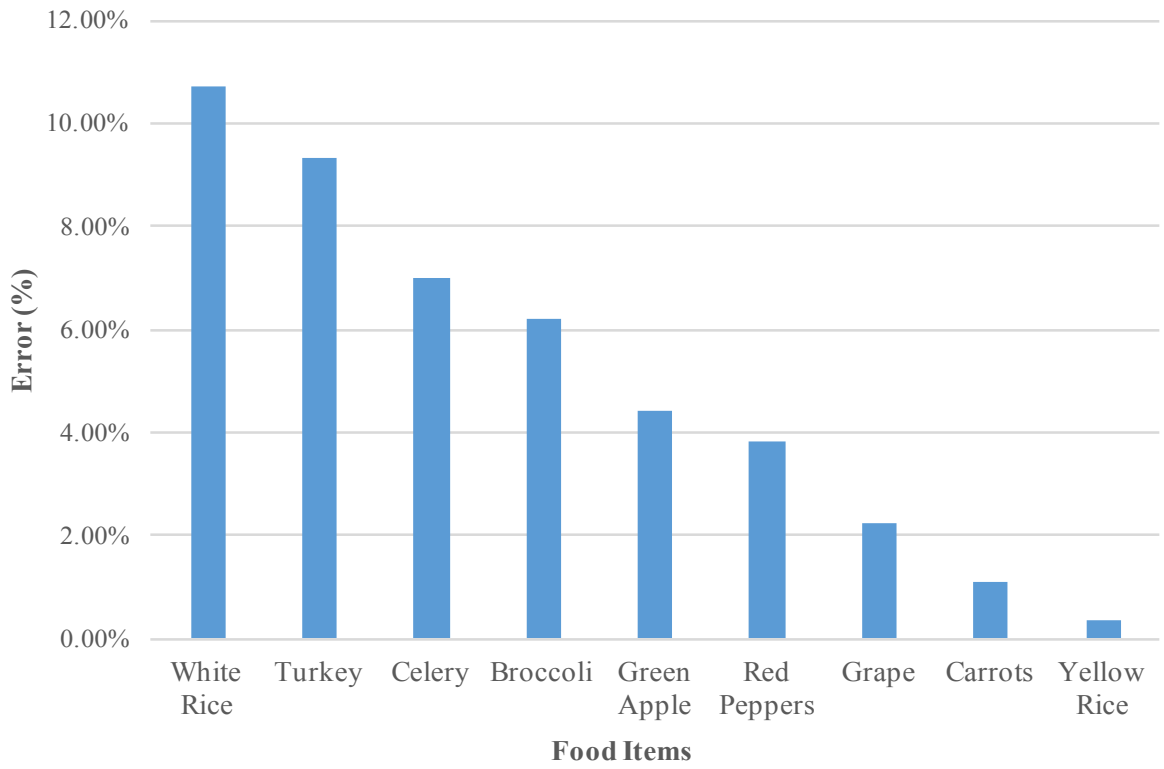

Figure 4.25. Percentage error of single item plates between calculated segmented area using SLIC Superpixel Segmentation and measured segmented area using the DDRS algorithm.

Next, we will look at the different tests conducted on the same food items as presented in Figure 4.25. This time, however, we combine all the food items with another food object on the plate, similar to Figure 4.24 (d). Using the same method as the single food plates, we calculate the absolute percentage error using SLIC Superpixel. Figure 4.26 shows the percentage error for multiple food plates. The overall average error percentage for multiple items on a plate is 7.16%.

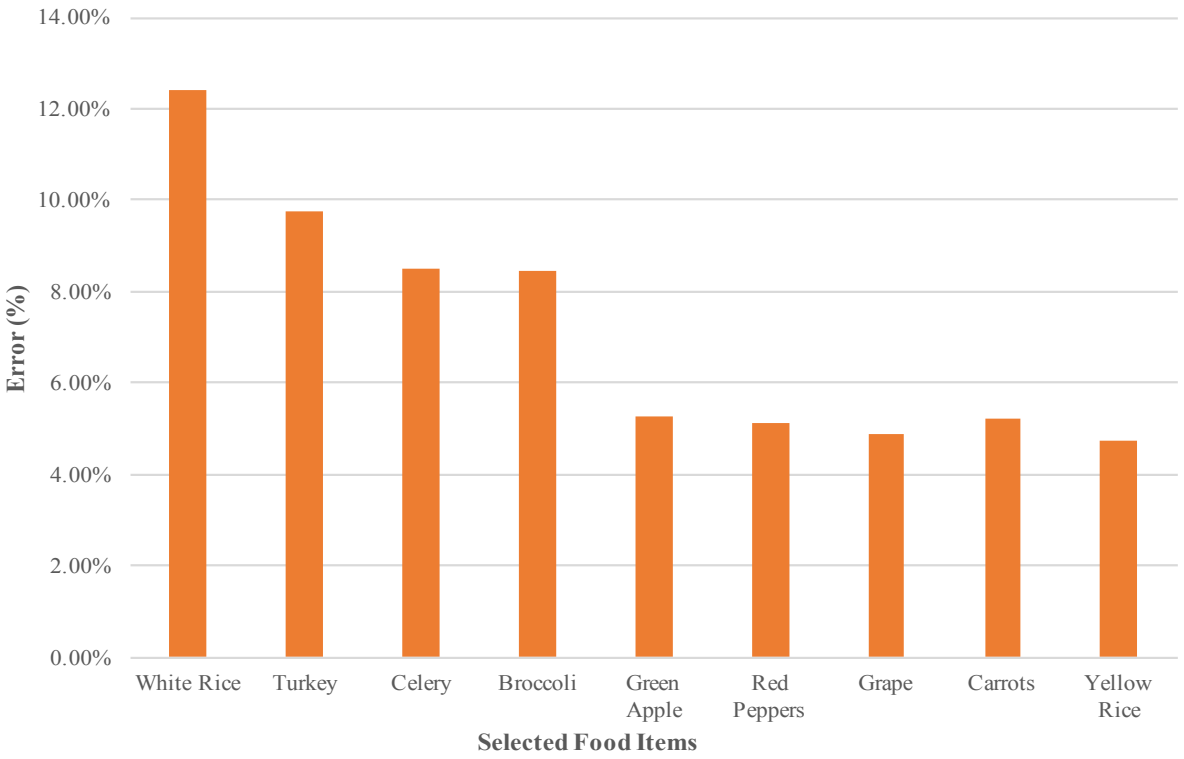

Figure 4.26. Percentage error of multiple food plates between calculated segmented area using SLIC Superpixel and measured segmented area using DDRS algorithm.

As seen in both Figure 4.26 and Figure 4.26, the percentage error is higher on multiple food plates compared to single food items. In both graphs, there is also a trend directly related to the color of the item being segmented. Figure 4.27 shows both single item plates and multiple food plates side by side to show this trend.

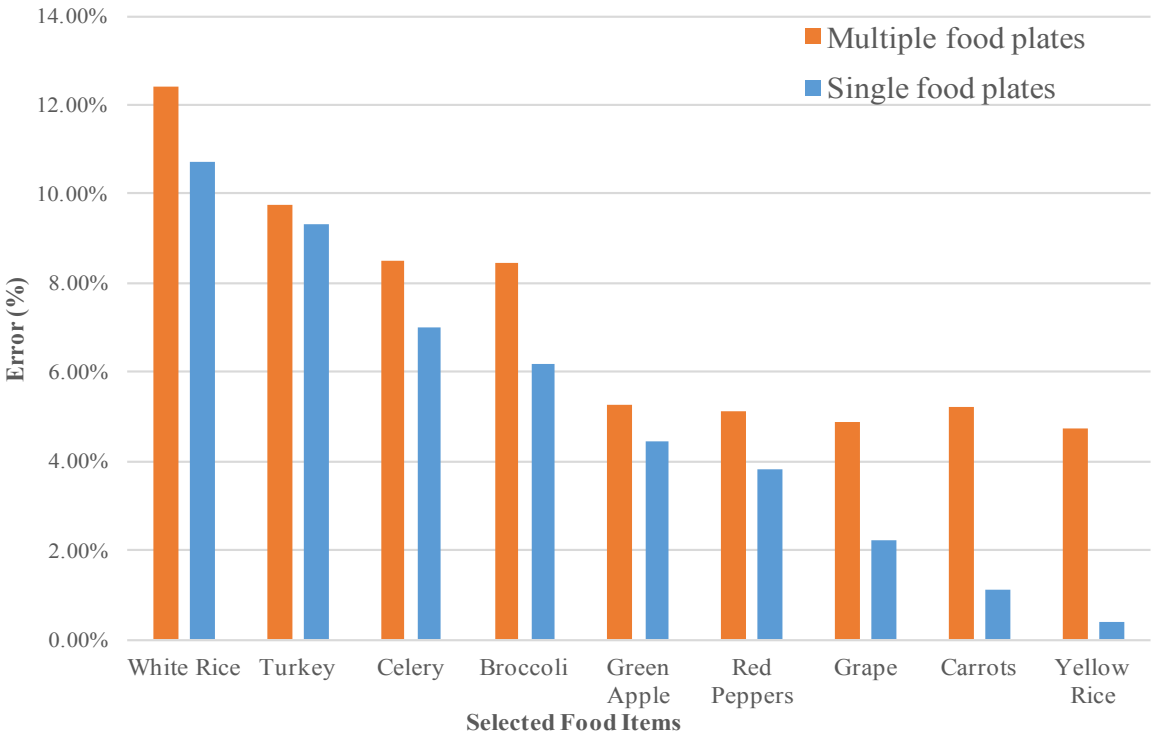

Figure 4.27. Percentage of error comparing a number of food items on a plate, where the error is determined by comparing the segmented area of DDRS and SLIC Superpixel Segmentation algorithms.

The sources of error come from the distinct color characteristics of the images, and the algorithm's static thresholding values. This can be attributed to the limitations discussed in Chapter 5. Let us look at a bigger sample to see the trend of the color characteristics of each food item. Figure 4.27 shows all of the food items scanned, so as to analyze the segmentation algorithm's performance.

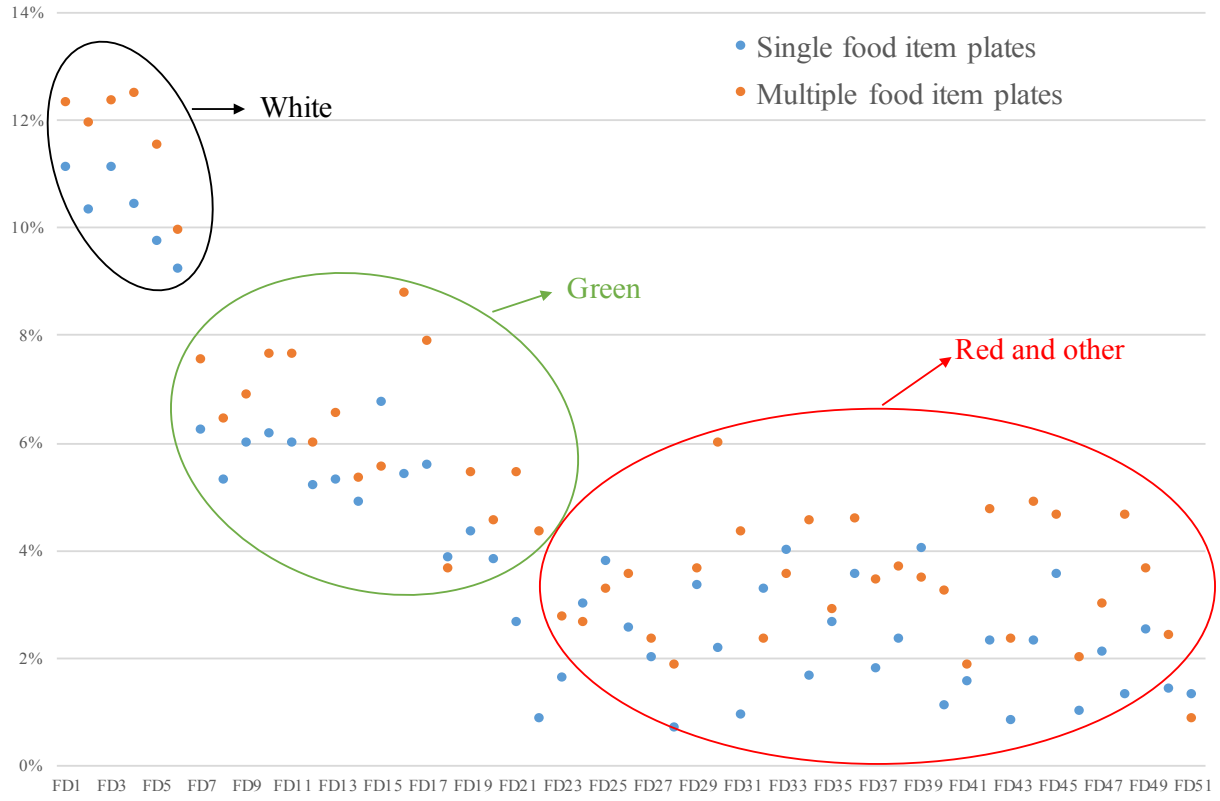

Figure 4.28. A larger sample size of percentage error calculated with both single and multiple food plates.

By analyzing the population of the data points and color characteristics of the scanned food items in Figure 4.28, we can observe that the food items in the near-white color category tended to have the highest errors in segmentation. Also, food items in the red category tended to have the lowest percentage of error.

We then analyzed the data to find patterns that might indicate the greatest sources of this error, discovering that when different objects were combined and tested, the error was higher than when the same objects were tested individually. We also noticed that similarly colored objects had similar percentages of error.

From these results, it is evident that the algorithm is highly accurate, capable of segmenting a multi-food plate in less than a minute—especially when analyzing single-object images. This accuracy dropped, however, when segmentation was used on multiple-object images. With further investigation, we noticed some reoccurring patterns in images with low accuracy results, based on two main categories: the single-object vs. multiple-object images, and their corresponding object colors. There was an overall error average of 9.34% for single objects, and 8.05% for multiple objects. As for the color-based analysis, segmenting white objects had the poorest volume measurement when compared to SLIC Superpixel Segmentation. Due to the similarity in color to the white serving plates, whiter foods can be misinterpreted as part of the background plate, and unnecessarily segmented from the image [31]. Thus, the color of the food is a greater contributor to the error in our algorithm than the number of food items in an image.

Another important result of this analysis was that the green objects were the second highest percentage of error. This is primarily because the projected laser dots were green, which caused there to be a lack of contrast between the dots and the food items being measured. This shows that the hue of the object clearly presents a limitation to the algorithm and that the static thresholds in the algorithm must be made dynamic to enhance the segmentation process.

4.6 CHAPTER SUMMARY

This chapter mainly talked about the two core algorithms for calculating the volume estimations; 1- Automatic Dot Detection, 2- Segmentation.

Automatic Dot Detection: Overall, the algorithm works well for a high percentage of images. The limitations of this algorithm are present in the color/brightness of the object, where the accuracy is limited due to the laser hardware interaction with bright objects. Currently, laser dots

on white objects are not located very well because the laser dots themselves do not stand out from such a bright background.

Segmentation: Overall, the current algorithm could function with a high percentage of accuracy for plates containing only one type of food, as is demonstrated in the previous section. The most critical limitations of this algorithm can be seen when multiple objects of different textures and colors appear on the plate, in which case, the area between the multiple food objects is not segmented as well by the saliency function.

Chapter 5. LIMITATIONS AND FUTURE WORK

5.1 LIMITATIONS IN THE CURRENT SYSTEM

Automatic Dot Detection: Two possible limitations of the algorithm occur with dark colored liquids such as wine, and foods with very high reflectivity. These objects do not show the dots very well, and it is very difficult to detect the dots with the current algorithm. However, these limitations depend on the hardware—for instance, on the luminance of the lasers—and we are still trying to improve this hardware.

Another limitation in this algorithm is the use of static thresholding variables throughout the software. Specifically, when the object is relatively small regarding pixels and the structuring element constant in the erosion is large. In this case, some of the dots near the edges are not captured. Characteristics of the images must be analyzed to determine how to make these threshold values dynamic.

Segmentation: This limitation can be explored in numerous ways. One way is to replace the static thresholding values of the algorithm with dynamic values. Thresholding is done multiple

times throughout the algorithm in both crops, as well as in the saliency map, and thus has a significant impact on the accuracy of the results. The current static thresholding values are set to values that were determined through testing on various images and resulted in consistently accurate and stable results. While we expect these fixed values to yield reasonable results on simple images, the algorithm is likely the cause for less reliable results on more complex ones.

Another way to bypass this limitation could be to enhance the auto-segmentation section of the algorithm. The current version of the algorithm is shown to yield highly accurate results for images with only a single object. The auto-segmentation algorithm could be expanded to detect the presence of multiple objects from the plate, and, as a result, to create multiple separately segmented images, each with a single food object in them. We expect this expansion of the auto-segmentation process to improve the current version of the algorithm immediately.

Additionally, as mentioned previously, the saliency map has very high accuracy when images are manually cropped to contain just the food [32]. This is because manual cropping by a user provides a tightly bound image around the food as compared to the original. Hence, one place where that the current auto-segmentation process could be improved is in the second stage. This stage performs a rough segment of the image with large structuring elements instead of a tight bound, before passing the segmented image to the HSV filter. A more accurate method could be designed to crop the image from the first crop of the plate to the second crop of the food using either edge detection or color segmentation. This might present less noise than the current HSV thresholding done in the second crop, thus passing on a more certain crop to the saliency mapping function [33].

5.2 FUTURE WORK

The limitation section points out the DDRS system's shortcomings. With the current hardware, we require modifying the algorithm to compromise the quality of the data coming from the physical portion of DDRS. However, by upgrading the hardware, especially the laser module to be able to project more than a 16x16 matrix; we would increase the accuracy, speed, and efficiency of the algorithm by over three times. Luckily, Occipital Inc. has recently produced an infrared 3D scanner hardware, which can produce over millions of dots on to an object. Shown in Figure 5.1 is the Structure Sensor which has been developed by Occipital Inc.

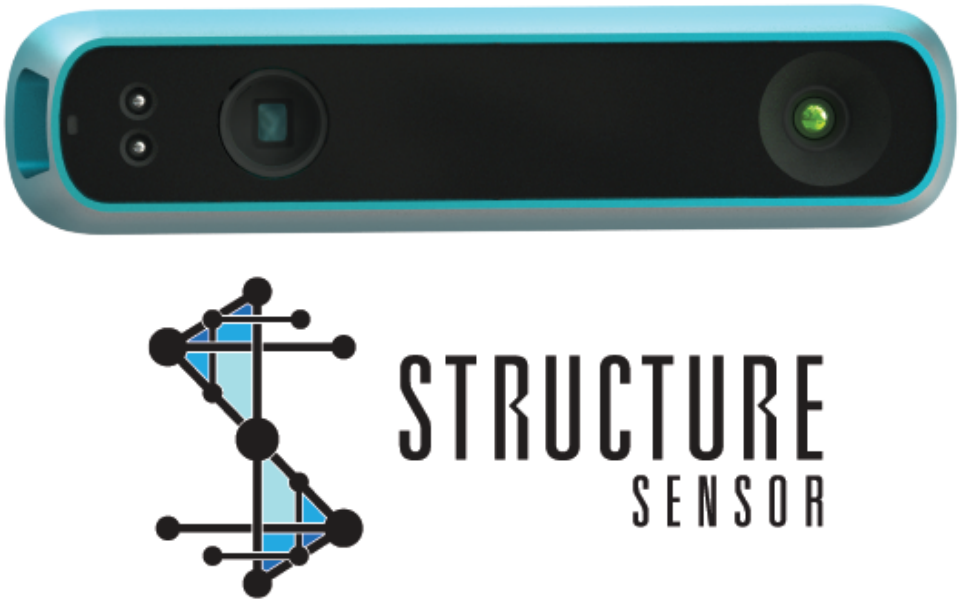

Figure 5.1. Structure Sensor device, developed by Occipital Inc.

The new developed DDRS with Structure Sensor would provide us with drastically high precision for volume estimation. This approach may increase the speed and overall completion rate of dietary intake assessment due to the automation of tasks. However, the Structure Sensor is enabled by connecting to an iPad and our custom built iOS app to capture the scan by interacting with the iPad's camera and Structure Sensor [34].

Future testing will be done by running the algorithm on sets of data provided by users from test studies to be conducted at the Harborview Medical Center. The results will be compared to empirical results found by manually selecting the dots. We expect that these improvements would further enhance the accuracy of the algorithm. Currently, the algorithm functions primarily as a proof of concept, with several areas remaining for possible improvement.

Chapter 6. 3D RECONSTRUCTION ALGORITHM IN OTHER APPLICATION

As a 3D scanner, DDRS has a potential to enter the market in multiple application, 3D printing, Gaming, wound care, etc.

3D reconstruction is a fast growing market. The new 3D printers can print flexible material, even human organs. This creates a significant opportunity for 3D scanners especially a portable one like DDRS.

New 3D goggles, such of Oculus, HoloLens, etc. has created a new market in Gaming. DDRS can act as a 3D camera and can be attached in Drones, Robots, etc. and be controlled via 3D goggles.

Using 3D scanners in wound care is a very new field added to the 3D reconstruction industry. Medical personnel is able to 3D map any patient's wound and create a care plan accordingly. This allows DDRS to be used in wound care market by medical professionals.

Chapter 7. CONCLUSION

In recent years, digital dietary measurement has been able to compete in both research and consumer industries with the traditional approaches by proposing advanced and sophisticated methods using sensors, high-quality cameras, and image processing to reduce the limitations and the user burden of calculating while increasing the accuracy.

In this dissertation, a new digital nutritional assessment method/tool design and development were presented. This dissertation has attempted to create a new method for dietary measurement using the structured light system. In addition to the hardware piece, we discuss the main algorithm of DDRS and emphasize mostly on the Automatic Laser Dot Detection and Segmentation algorithms.

The Automatic Laser Dot Detection algorithm consisted of filtering via two masks: an HSV filtered cosine similarity mask, as well as a luminance mask. The combination of the two masks, as well as threshold filtering, isolates the laser dots as can be seen in the results section. An additional manual error catching GUI was developed to ensure the correct values are being passed to the 3D reconstruction phase in the DDRS. Several suggestions were presented in the analysis section for possible improvements to the algorithm.

The Segmenting algorithm enabled us only to focus on the food item by cropping the rest of the image as background. The algorithm does this by developing an auto-cropping method that implements knowledge about the environment with saliency mapping techniques. This led to the creation of an algorithm that successfully isolates food in an image with over 85% similarity to the SLIC algorithm, and a fast processing speed that will not hinder the operator.

This algorithm was tested thoroughly on various food samples, showing that it works with percentages of error as low as one percent. We grouped foods with similar percentages of error to

see if any similarities between them might have caused these errors and found that using different food colors produces different results. Segmenting red and yellow food items produced the best results while processing white and green food items produced the highest percentages of error.

Future work consists of improving the 3D calculation as well as upgrading the physical portion by using Structure Sensor instead of a laser module. This will provide us with possibly faster, more accurate, nutritional intake information.

BIBLIOGRAPHY

[1] S. Makhsous, H. M. Mohammad, A. V. Mamishev, A. R. Kristal, and J. M. Schenk, "Dietary Assessment Using Automatic Laser Point Recognition," To be published.

[2] S. Makhsous, A. V. Mamishev, A. R. Kristal, and J. M. Schenk, "Dietary Assessment Using Image Segmentation," To be published.

[3] M. B. Schulze, S. Liu, E. B. Rimm, J. E. Manson, W. C. Willett, and F. B. Hu, "Glycemic index, glycemic load, and dietary fiber intake and incidence of type 2 diabetes in younger and middle-aged women," *Am. J. Clin. Nutr.*, vol. 80, no. 2, pp. 348–356, Aug. 2004.

[4] S. L. Lim, K. C. B. Ong, Y. H. Chan, W. C. Loke, M. Ferguson, and L. Daniels, "Malnutrition and its impact on cost of hospitalization, length of stay, readmission and 3-year mortality," *Clin. Nutr.*, vol. 31, no. 3, pp. 345–350, Jun. 2012.

[5] R. Almaghrabi, G. Villalobos, P. Pouladzadeh, and S. Shirmohammadi, "A novel method for measuring nutrition intake based on food image," in *Instrumentation and Measurement Technology Conference (I2MTC), 2012 IEEE International*, 2012, pp. 366–370.

[6] M. B. E. Livingstone, P. J. Robson, and J. M. W. Wallace, "Issues in dietary intake assessment of children and adolescents," *Br. J. Nutr.*, vol. 92, no. Supplement S2, pp. S213–S222, Oct. 2004.

[7] D. Mozaffarian, T. Hao, E. B. Rimm, W. C. Willett, and F. B. Hu, "Changes in Diet and Lifestyle and Long-Term Weight Gain in Women and Men," *N. Engl. J. Med.*, vol. 364, no. 25, pp. 2392–2404, Jun. 2011.

[8] P. Pouladzadeh, S. Shirmohammadi, and A. Yassine, "You are what you eat: So measure what you eat!," *IEEE Instrum. Meas. Mag.*, vol. 19, no. 1, pp. 9–15, Feb. 2016.

[9] "USDA Food Composition Databases." [Online]. Available: https://ndb.nal.usda.gov/ndb/search/list. [Accessed: 06-Oct-2016].

[10] J. Shang, M. Duong, E. Pepin, X. Zhang, K. Sandara-Rajan, A. Mamishev, and A. Kristal, "A mobile structured light system for food volume estimation," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 100–101.

[11] R. J. Carroll, D. Midthune, A. F. Subar, M. Shumakovich, L. S. Freedman, F. E. Thompson, and V. Kipnis, "Taking Advantage of the Strengths of 2 Different Dietary Assessment Instruments to Improve Intake Estimates for Nutritional Epidemiology," *Am. J. Epidemiol.*, p. kwr317, Jan. 2012.

[12] G. H. Beaton, J. Milner, P. Corey, V. McGuire, M. Cousins, E. Stewart, M. de Ramos, D. Hewitt, P. V. Grambsch, N. Kassim, and J. A. Little, "Sources of variance in 24-hour dietary recall data: Implications for nutrition study design and interpretation," *Am. J. Clin. Nutr. USA*, 1979.

[13] A. Schatzkin, V. Kipnis, R. J. Carroll, D. Midthune, A. F. Subar, S. Bingham, D. A. Schoeller, R. P. Troiano, and L. S. Freedman, "A comparison of a food frequency questionnaire with a 24-hour recall for use in an epidemiological cohort study: results from the biomarker-based Observing Protein and Energy Nutrition (OPEN) study," *Int. J. Epidemiol.*, vol. 32, no. 6, pp. 1054–1062, Dec. 2003.

[14] A. Meyers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarram, G. Papandreou, J. Huang, and K. P. Murphy, "Im2Calories: Towards an Automated Mobile Vision Food Diary," *IEEE Int. Conf. Comput. Vis. ICCV*, Dec. 2015.

[15] G. Villalobos, R. Almaghrabi, B. Hariri, and S. Shirmohammadi, “A Personal Assistive System for Nutrient Intake Monitoring,” in *Proceedings of the 2011 International ACM Workshop on Ubiquitous Meta User Interfaces*, New York, NY, USA, 2011, pp. 17–22.
[16] P. Pouladzadeh, S. V. B. Peddi, P. Kuhad, A. Yassine, and S. Shirmohammadi, “A virtualization mechanism for real-time multimedia-assisted mobile food recognition application in cloud computing,” *Clust. Comput.*, vol. 18, no. 3, pp. 1099–1110, Sep. 2015.
[17] P. Pouladzadeh, S. Shirmohammadi, and A. Yassine, “Using Graph Cut Segmentation for Food Calorie Measurement,” in *2014 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, 2014, pp. 1–6.
[18] “Nutrition Data System for Research – Nutritional Analysis Software - ndsr87072 - University of Minnesota Office for Technology Commercialization.” [Online]. Available: http://license.umn.edu/technologies/ndsr87072_nutrition-data-system-for-research-nutritional-analysis-software. [Accessed: 02-May-2016].
[19] G. Wang, Y. Wang, H. Li, X. Chen, H. Lu, Y. Ma, C. Peng, Y. Wang, and L. Tang, “Morphological Background Detection and Illumination Normalization of Text Image with Poor Lighting,” *PLOS One*, vol. 9, no. 11, p. e110991, Nov. 2014.
[20] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, “Saliency Detection via Graph-Based Manifold Ranking,” presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3166–3173.
[21] J. Hyyppa, O. Kelle, M. Lehikoinen, and M. Inkinen, “A segmentation-based method to retrieve stem volume estimates from 3-D tree height models produced by laser scanners,” *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 5, pp. 969–975, May 2001.
[22] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz, “Spacetime faces: High-resolution capture for~ modeling and animation,” in *Data-Driven 3D Facial Animation*, Springer, 2008, pp. 248–276.
[23] C. Y. Ren and I. Reid, “gSLIC: a real-time implementation of SLIC superpixel segmentation,” *Univ. Oxf. Dep. Eng. Tech. Rep.*, 2011.
[24] M. A. Tschopp, “Image Processing with MATLAB 1,” *Mississippi State University ICME*. 04-Aug-2014.
[25] L. Shapiro, “Stockman G: Computer Vision,” *Prentice Hall*, 2002.
[26] Y. Bok, Y. Jeong, D.-G. Choi, and I. S. Kweon, “Capturing Village-level Heritages with a Hand-held Camera-Laser Fusion Sensor,” *Int. J. Comput. Vis.*, vol. 94, no. 1, pp. 36–53, Oct. 2010.
[27] S. Makhsous, A. V. Mamishev, A. R. Kristal, and J. M. Schenk, “Dietary Assessment Using 3D Reconstruction,” To be published.
[28] J. C. Bezdek, L. O. Hall, and L. P. Clarke, “Review of MR image segmentation techniques using pattern recognition,” *Med. Phys.*, vol. 20, no. 4, p. 4, 1993.
[29] T. van Lankveld, M. van Kreveld, and R. Veltkamp, “Identifying rectangles in laser range data for urban scene reconstruction,” *Comput. Graph.*, vol. 35, no. 3, pp. 719–725, Jun. 2011.
[30] Y. J. Zhang, “A survey on evaluation methods for image segmentation,” *Pattern Recognit.*, vol. 29, no. 8, pp. 1335–1346, 1996.
[31] A. Banno, T. Masuda, T. Oishi, and K. Ikeuchi, “Flying Laser Range Sensor for Large-Scale Site-Modeling and Its Applications in Bayon Digital Archival Project,” *Int. J. Comput. Vis.*, vol. 78, no. 2–3, pp. 207–222, Nov. 2007.

[32] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient Graph-Based Image Segmentation,” *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.
[33] Y. Boykov and G. Funka-Lea, “Graph Cuts and Efficient N-D Image Segmentation,” *Int. J. Comput. Vis.*, vol. 70, no. 2, pp. 109–131, Nov. 2006.
[34] “Structure Sensor - 3D scanning, augmented reality, and more for mobile devices.” [Online]. Available: http://structure.io/. [Accessed: 10-May-2016].
[35] A. Banno, T. Masuda, T. Oishi, and K. Ikeuchi, “Flying Laser Range Sensor for Large-Scale Site-Modeling and Its Applications in Bayon Digital Archival Project,” Int. J. Comput. Vis., vol. 78, no. 2–3, pp. 207–222, Nov. 2007.
[36] E. A. Finkelstein, J. G. Trogdon, J. W. Cohen, and W. Dietz, “Annual Medical Spending Attributable To Obesity: Payer-And Service-Specific Estimates,” Health Aff. (Millwood), vol. 28, no. 5, pp. w822–w831, Sep. 2009.
[37] W. Tao, H. Jin, and Y. Zhang, “Color Image Segmentation Based on Mean Shift and Normalized Cuts,” IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 37, no. 5, pp. 1382–1389, Oct. 2007.
[38] F. B. Hu, J. E. Manson, M. J. Stampfer, G. Colditz, S. Liu, C. G. Solomon, and W. C. Willett, “Diet, Lifestyle, and the Risk of Type 2 Diabetes Mellitus in Women,” N. Engl. J. Med., vol. 345, no. 11, pp. 790–797, Sep. 2001.
[39] “Multi-Scale Analysis of Area Fractions Characterization GUI - File Exchange - MATLAB Central,” 04-Feb-2016. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/23425-multi-scale-analysis-of-area-fractions-characterization-gui. [Accessed: 04-Feb-2016].
[40] H. Møller, A. Mellemgaard, K. Lindvig, and J. H. Olsen, “Obesity and cancer risk: a danish record-linkage study,” Eur. J. Cancer, vol. 30, no. 3, pp. 344–350, 1994.
[41] H. B. Hubert, M. Feinleib, P. M. McNamara, and W. P. Castelli, “Obesity as an independent risk factor for cardiovascular disease: a 26-year follow-up of participants in the Framingham Heart Study.,” Circulation, vol. 67, no. 5, pp. 968–977, May 1983.
[42] R. T. Hurt, T. H. Frazier, S. A. McClave, and L. M. Kaplan, “Obesity Epidemic: Overview, Pathophysiology, and the Intensive Care Unit Conundrum,” JPEN J Parenter Enter. Nutr., vol. 35, no. 5 suppl, p. 4S–13S, Sep. 2011.
[43] K. M. Flegal, M. D. Carroll, C. L. Ogden, and L. R. Curtin, “Prevalence and Trends in Obesity Among US Adults, 1999-2000,” JAMA, vol. 303, no. 3, pp. 235–241, Jan. 2010.
[44] R. A. Hammond and R. Levine, “The economic impact of obesity in the United States,” Diabetes Metab Syndr Obes, vol. 3, no. 1, pp. 285–95, 2010.

VITA

Sepehr Makhsous received the B.S. degree in electrical engineering in 2012 from the University of Washington (UW), Seattle, where he is currently working toward the Ph.D. degree in electrical engineering. He is currently with the Sensors Energy and Automation Laboratory (SEAL), UW. His research interests include Visual Based Measurement (VBM), nutritional measurement, image processing, sensors, and classification.