Food Volume Measurement

Concept and Prototype

Samson Azizyan

Report

Summer 2019

Engineer

Information and Communication Technology

Table of Contents

[1 Assignment 3](#_Toc16245722)

[2 Research 3](#_Toc16245723)

[2.1 Two-view 3D Reconstruction for Food Volume Estimation 3](#_Toc16245724)

[2.1.1 Conclusion 4](#_Toc16245725)

[2.2 Comparison of Known Food Weights with Image-Based Portion-Size Automated Estimation and Adolescents’ Self-Reported Portion Size 4](#_Toc16245726)

[2.2.1 Conclusion 4](#_Toc16245727)

[2.3 A New Approach to Image-Based Estimation of Food Volume 5](#_Toc16245728)

[2.3.1 Conclusion 6](#_Toc16245729)

[2.4 Food Weight Estimation using Smartphone and Cutlery (NAIST) 6](#_Toc16245730)

[2.4.1 Conclusion 7](#_Toc16245731)

[2.5 3D Shape Scanning with a Time-of-Flight Camera 7](#_Toc16245732)

[2.5.1 Conclusion 8](#_Toc16245733)

[3 Proposed Methods 8](#_Toc16245734)

[3.1 Food Volume Estimation with Smartphone and a Time-of-Flight Sensor 9](#_Toc16245735)

[3.2 Implementation Attempt 9](#_Toc16245736)

[3.3 Linear movement of the smartphone 11](#_Toc16245737)

[3.3.1 Getting X and Y coordinates 11](#_Toc16245738)

[3.3.2 Experiment 12](#_Toc16245739)

[3.3.3 Results 14](#_Toc16245740)

[4 Alternative Solutions 15](#_Toc16245741)

[4.1 Volume measurement using cutlery 15](#_Toc16245742)

[4.2 Volume measurement using a 3D scanner 15](#_Toc16245743)

[5 Conclusion 17](#_Toc16245744)

[6 Links 17](#_Toc16245745)

[References 18](#_Toc16245746)

Table of Figures

[Figure 1. A snapshot of the application. After the user takes a short video, six frames are selected automatically (a) and marked by the user to seed segmentation (b). 5](#_Toc16245616)

[Figure 2. Processing steps for container diameter and chopsticks length measurement 7](#_Toc16245617)

[Figure 3. Antique head (a); our algorithm computes a 3D model of reasonable quality (c) despite severe errors in the raw ToF data (b). Reconstruction error (e) compared to a laser scan (d) shows that no circumstance the error was larger than 2.5 cm, while for most of the surface it was below 1.0 cm. (Note: raw aligned scans, no hole ﬁlling done) 8](#_Toc16245618)

[Figure 4. Time of Flight sensor VL53L0X 9](#_Toc16245619)

[Figure 5. Arduino Nano 9](#_Toc16245620)

[Figure 6. ESP32 10](#_Toc16245621)

[Figure 7. The setup and the app 11](#_Toc16245622)

[Figure 8. The Linear Movement Measurement App 13](#_Toc16245623)

[Figure 9. Approximate movement of the phone 14](#_Toc16245624)

[Figure 10. The 3D scanner as seen from above 16](#_Toc16245625)

[Figure 11. The 3D scanner as seen from below 16](#_Toc16245626)

# Assignment

Being able to estimate food volume using a mobile phone camera is essential in estimating the quantity intake of that food. Such an endeavor is particularly helpful for caregivers of the diabetes mellitus (DM) or pre-DM patients to remotely monitor daily food intakes either so that insulin could be properly administered or as a preventative measure for the pre-DM patients. While algorithms exist, they have not yet taken full advantages of the current state of advance of the mobile technology. Besides none has been designed specifically for Thai food which could have different characteristics. In this project, the trainee will do a literature review on existing state of the art techniques for estimating food volume, summarizing the key findings as well as identifying those features that have now been supported by the mobile phone (i.e. Android and iOS). Based on such findings, the trainee recommends a conceptual design of a method for estimating Thai food volume using a mobile phone camera.

# Research

I have read through multiple researches and papers about food volume measurement and estimation. I picked four the most efficient out of all, I have summarized them and listed pros and cons of each method. The fifth method that I picked was about 3D shape scanning of an object, using a Time-of-Flight camera.

## Two-view 3D Reconstruction for Food Volume Estimation

This method is using stereovision to estimate the volume of multi-food meals with unconstrained 3D shape. The method requires two meal images with the food placed inside an elliptical plate, a credit card sized reference card next to the dish, and a segmentation of the food and dish available, possibly performed by automatic methods. The dish may have any elliptical shape and its bottom should be flat. The proposed system consists of three major stages: (i) extrinsic calibration, (ii) dense reconstruction, and (iii) volume extraction. (Anthimopoulos 2017.)

1. Extrinsic Calibration
   1. Salient point matching (SURF)
   2. Relative pose extraction (RANSAC)
   3. Scale estimation
2. Dense Reconstruction
   1. Rectification
   2. Stereo Matching
   3. Point cloud generation
3. Volume Estimation
   1. Food Surface Extraction
   2. Dish Surface Extraction
   3. Volume Calculation

### Conclusion

As this method was somewhat efficient with error rate varying from 9.4% to 8.2%, it still required two images from different angles and the reference card to measure the volume and weight of the food portion and the measured food was always on flat plates.

## Comparison of Known Food Weights with Image-Based Portion-Size Automated Estimation and Adolescents’ Self-Reported Portion Size

The system partitioned the space of the food objects into two geometric shapes, cylinders and squares, each with their own set of parameters. The spherical approximation models drew upon spheres and prismatic approximation models. For the foods and beverages served at meals, the automated volume was estimated as cubic centimeters and converted into weight (g) using density values derived from rapeseed volumeter measures of duplicate plates of each meal. (Boushey 2012.)

### Conclusion

The goal of this research was a comparison between manual and automatic food weight estimation, this was an important research. The research showed that the automated estimation is way more accurate than the manual one, so AI is better at estimating a food weight/volume than human. This research did not dive into the detailed description of the automated method of food weight estimation.

## A New Approach to Image-Based Estimation of Food Volume

The system has been designed as a client-server solution. The smartphone acts as the client and runs the semi-automatic part of the system. Building on our previous work, we complete the processing cycle addressing all the main practical and theoretical issues related with food volume estimation. In the system described in this paper, as a first step, the user is asked to take a short video of the food. Then, a subset of the frames is automatically extracted from the video to provide relevant views of the food from all sides. Next, the user marks some food and non-food parts in one of the selected frames of the video. A mobile application that we developed facilitates this part (see Figure 1). The segmentation process, which is performed on the server side, is seeded by the user’s marks. The segmented images are then fed into a customized image-based process that builds a 3D model of the food items. This model is finally used to calculate the volume of the food. A small checkerboard is used as the size and ground reference for the modeling process. (Cagnoni 2017.)

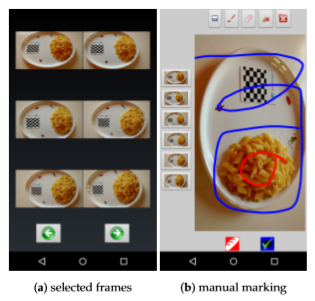


Figure 1. A snapshot of the application. After the user takes a short video, six frames are selected automatically (a) and marked by the user to seed segmentation (b).

### Conclusion

As the method was very efficient with average error rate for solid shapes being 8.27% and for non-solid shapes 6.7%, it still had some problems with it. It took a lot of effort from the user, as he or she had to take a vide and manually mark the food and non-food areas of the video. Also, this method required a reference card and the processing times varied from 21.5 s to 24.01 s, which is very long time.

## Food Weight Estimation using Smartphone and Cutlery (NAIST)

In this paper, researchers propose the use of cutlery objects, especially Japanese chopsticks, to estimate the weight of the food in an image, taken with a smartphone. Their system exploits image-processing techniques and the EXIF metadata of the food image (camera focal length, sensor size) to measure the food container size, determine the food volume, and get the food weight by using density information of that particular food. An important aspect of this process is the use of the cutlery present in the food image. Cutlery are the ﬁrst things that we need to start eating. In the proposed system, cutlery such as chopsticks, spoon, fork, and knife, which are found in any eating environment, are used as measurement references. The use of these cutlery makes the system ubiquitous and suppress the restriction of always carrying and using calibration objects utilized in existing systems. (Arakawa 2016.)

Although research focused on the chopsticks, the proposed system can use any other eating tools (cutlery) such as spoon and fork, as measurement references, as long as their real length and color are provided to the system. The user of the system is asked to take a single picture from the top with the chopsticks in the picture. Then via the network connectivity of the smartphone, the image is sent to a server where the system estimates the diameter and the height of the food container. Once the container sizes are known, the systems estimates the food volume and the food weight by specifying the food type. (Arakawa 2016.)

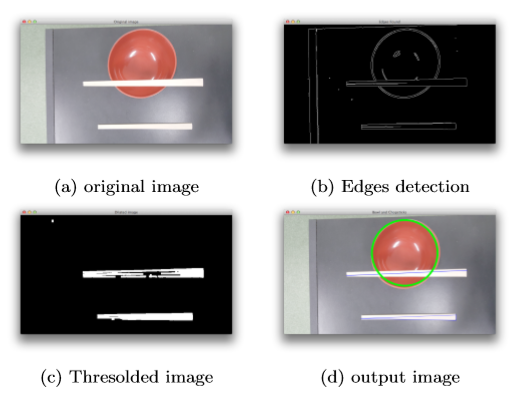


Figure 2. Processing steps for container diameter and chopsticks length measurement

### Conclusion

I liked this method the most, because it did not require a lot of effort from the user and it could measure food volume in deeper food containers (bowls). Also, user took only one picture of the food. This method had the lowest error rates of all of the researches, that I have read through. The downside to this method was that user had to know the length of the cutlery, that he or she were using to take an image. The user had to carry his or hers own cutlery or some type of a measurement tool, which could be less convenient for a regular user of an application, but for a user with diabetes, who’s life depends on food estimation, it might be not that big of a deal.

## 3D Shape Scanning with a Time-of-Flight Camera

This research used a Time-of-Flight camera to scan and reconstruct the 3D shaped objects. their goal was to build the, to their knowledge, ﬁrst 3D shape scanner based on a Time-of-Flight camera that can be used in hand-held and turntable scanning mode. They used a MESA Swiss ranger SR4000 as ToF sensor. In a nutshell, it emits infrared light into the scene and at each pixel measures the return time of the reﬂected light from which it determines the depth of the pixel. The raw depth data taken from ToF sensor had a lot of noise in it, they used the LidarBoost ToF superresolution approach to minimize the noise. (Chan 2010.)

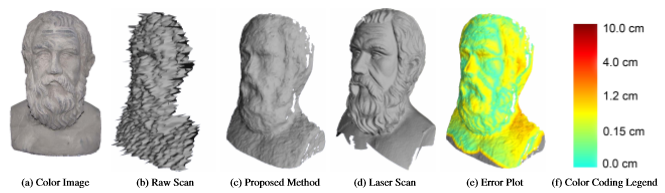


Figure 3. Antique head (a); our algorithm computes a 3D model of reasonable quality (c) despite severe errors in the raw ToF data (b). Reconstruction error (e) compared to a laser scan (d) shows that no circumstance the error was larger than 2.5 cm, while for most of the surface it was below 1.0 cm. (Note: raw aligned scans, no hole ﬁlling done)

### Conclusion

The error rate was very low, when you take into consideration the distance between the sensor and the object. This might be the most important research in 3d object scanning with handheld devices.

# Proposed Methods

After researching for some time, I had my first idea for the concept, where I used the cutlery research as the blueprint. This concept got shut down, because it seemed inconvenient to always carry around the chopsticks or some other cutlery with you, which is understandable. After that I was little bit stuck with this project, but the student from KMUTT, who is helping me, suggested that we use external Time-of-Flight sensor and Arduino Nano to make a 3D map of an object, which in this case is a food portion.

## Food Volume Estimation with Smartphone and a Time-of-Flight Sensor

I have decided to try and build a prototype using a Time-of-Flight sensor VL53L0X, Arduino Nano and an Android smartphone. There is a library for Arduino to read the data from VL53L0X and convert it into millimeters. The library is written with C programming language. We will use Android Studio to develop an app, Android Studio uses either java or kotlin. It is possible to use JNI (Java Native Interface) to use other programming languages than Java within the Java code. The challenge here is to learn java, JNI, Arduino, ToF sensor and 3D mapping as we go.



Figure 4. Time of Flight sensor VL53L0X

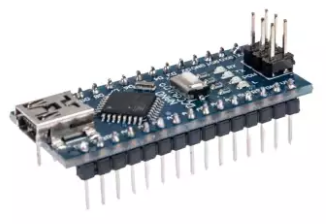


Figure 5. Arduino Nano

## Implementation Attempt

I was trying to develop an Android app in Android Studio, that can combine linear coordinates (X, Y) of the Android device and the distance reading from the Time of Flight sensor. That way we would get one point of the scan. To get the 3D map of the food portion, you would need thousands of those points. I ran into a problem as I have no previous experience with Java, Android Studio or microcontrollers. I couldn’t connect the Arduino board to my app via usb cable. While researching I could only find libraries for desktop java applications to read the Serial outputs from Arduino.

After multiple failed attempts to connect my app to Arduino using a usb cable I switched the microcontroller board from Arduino Nano to ESP32. I moved from Arduino Nano to ESP32, because ESP32 is more powerful and has Wifi and Bluetooth capabilities. This change gave me a lot of extra work as I had to learn how to use Bluetooth Low Energy and how get reading from ESP32 through it. I was able to connect to the board, but couldn’t get the readings on my own app. However, I did manage to get the reading through the third-party app, but it did not help me as I need to save the readings into my app.



Figure 6. ESP32

I felt like I was hitting the wall until the local student, who is helping me with the project found the Physicaloid library for Android Studio, which does exactly what we need. Now we can save the ToF readings into the app that we developed.

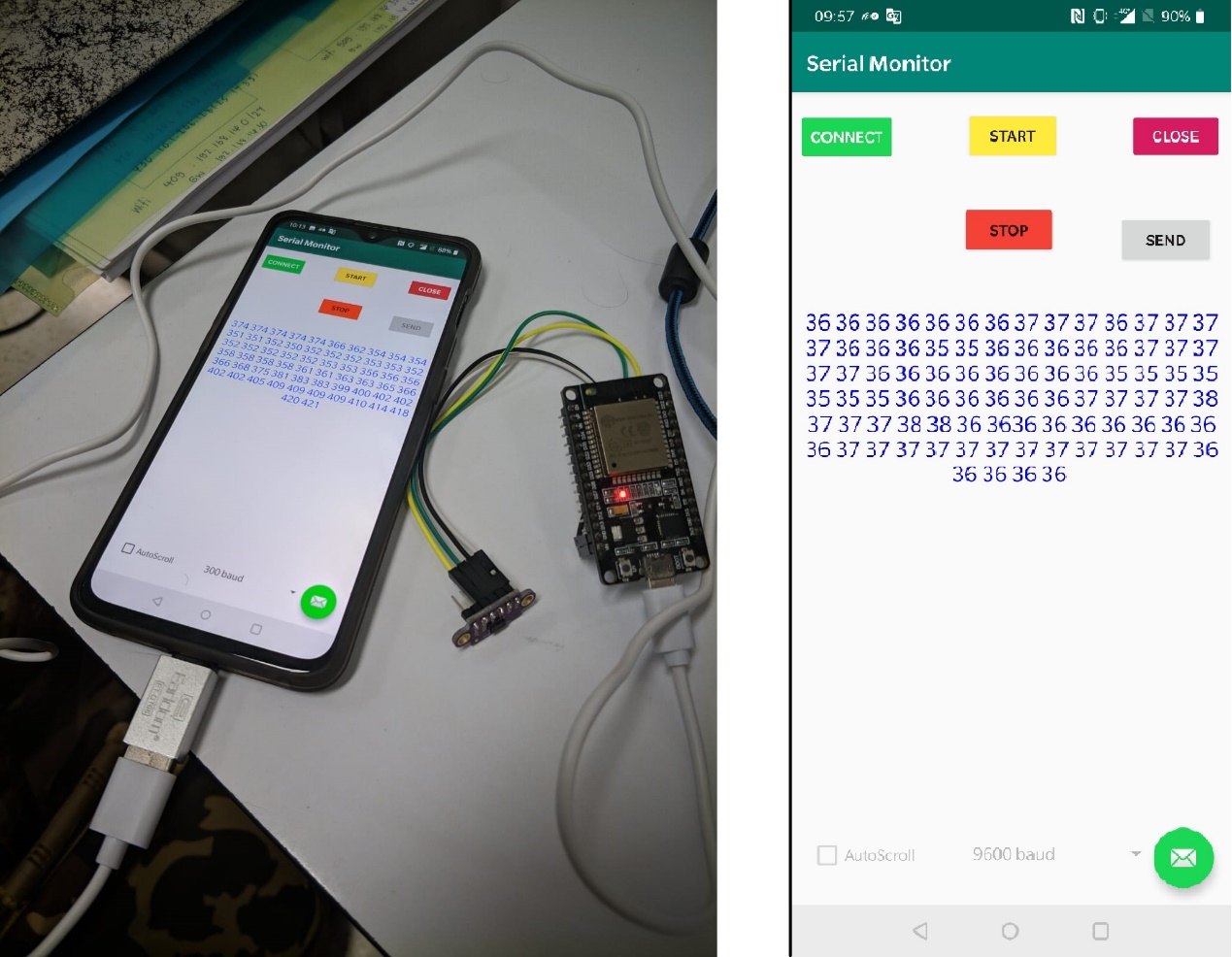


Figure 7. The setup and the app

## Linear movement of the smartphone

Next part might be even harder as the double integration of the accelerometer data creates a horrible drift.

### Getting X and Y coordinates

I have researched the internet and tried to find a solution or an algorithm for getting the XY coordinates while moving the android device, everything that I found was explaining how inaccurate the measurement would be. Almost all the sources were pointing to one video of the Google presentation: “Sensor Fusion on Android Devices: A Revolution in Motion Processing” by David Sachs. One part of the presentation was dedicated to linear movement of the android device and how to measure it. He explained that it is impossible to get result, that is even close to being precise. The reason for the inaccuracy is that accelerometer of any smartphone out there provides us with the raw accelerometer data. For measuring a linear movement of the device, we need Linear Acceleration of the device, that we can get from a “Sensor Fusion”, which is Accelerometer Data – Gravity. Accelerometer Data in its self is little noisy, when you fuse the sensors, subtract gravity from accelerometer data and get a Linear Accuracy, the noise level rises. After that we need to integrate a Linear Acceleration twice to get a linear movement. The first integration is to get a velocity and the second integration is to get a linear movement. When you integrate a linear acceleration once the error rate becomes quite substantial. The second time you integrate, the reading could be off by tens centimeters.

### Experiment

After getting the conclusion from my research, that the linear movement measurement with smartphone sensors is terribly inaccurate, I have decided to do my own experiment. So, I have created an app, that calculates the X and Y coordinates of the android device.

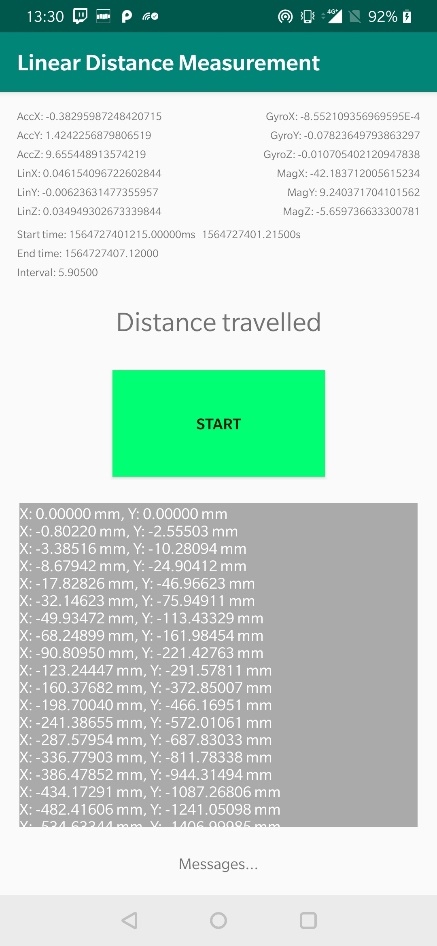


Figure 8. The Linear Movement Measurement App

The app uses “The Linear Movement Based on The Constant Linear Acceleration” equation.

Equation 1. Linear movement on the X axis

Equation 2. Linear movement on the Y axis

Equation 3. Velocity

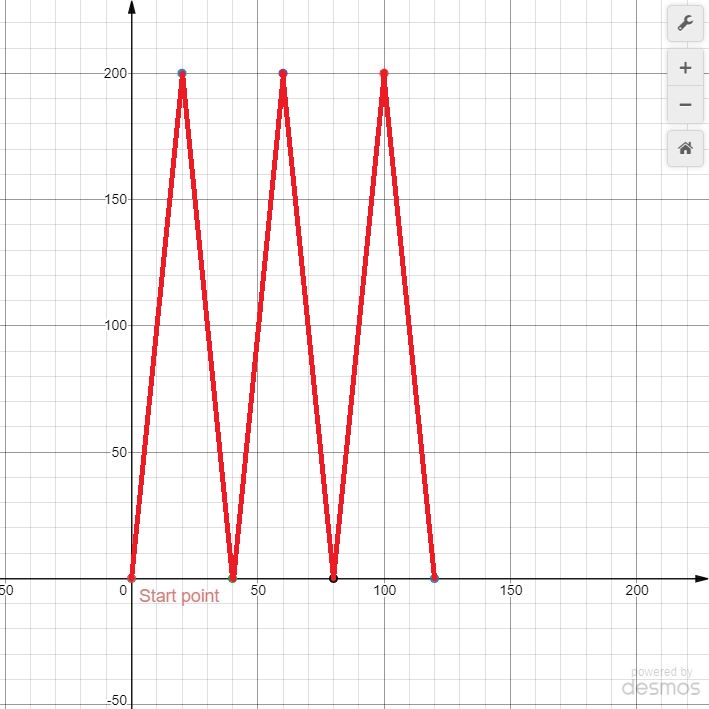
The program did the calculations every 0.1s, total measurement time was 6s. I started moving the phone from resting position (phone is placed flat on the table in the portrait position) (0,0) in a zigzag motion.

Figure 9. Approximate movement of the phone

### Results

First three readings:  
(0.00000,0.00000)  
(0.02503,0.05014)  
(0.15031,-0.00522)

Three readings from the middle:  
(164.02384,-2571.27175)  
(211.13584,-2373.65505)  
(188.72183,-2313.06858)

Last three readings:  
(-758.40884,-27582.96197)  
(-752.05957,-28615.18822)  
(-799.53970,-29814.19805)

From these results we can see, that the measurement is not even close to being accurate. The errors stack up each time the program calculates the coordinates and the end of the process X is about 0.8 meters and Y is almost 30 meters. (Full reading is provided in the link at the end of the report).

# Alternative Solutions

There is a reason why object volume measurement with just a smartphone has not been done before. It is impossible to get a precise result with the current smartphone hardware. Let’s look at the alternative solutions.

## Volume measurement using cutlery

As I have stated earlier, this proposed method has been shut down. I still think this is a legit method for people, that are struggling with diabetes. If your life depends on knowing the food estimation, I am sure it is not a problem to carry some chopsticks on you.

With the use of image processing and reference object, which is this case are chopstick or some other cutlery, the raw estimation can be achieved. The algorithm is public.

## Volume measurement using a 3D scanner

This method is an updated version of a method, that uses the Time of Flight sensor. The method might be inconvenient, because it uses more external devices and one extra sensor. The extra sensor is an optical sensor, that is in gaming mice. The inconvenience comes with the optical sensor, because it needs to be on a surface for it to work.

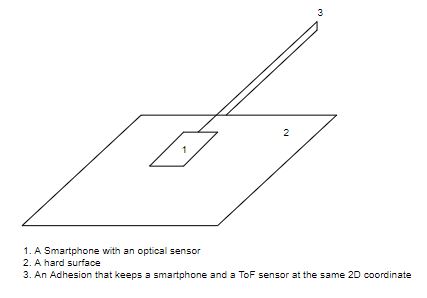


Figure 10. The 3D scanner as seen from above

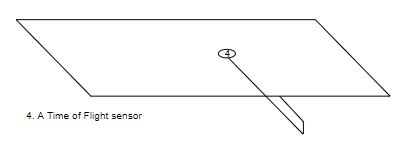


Figure 11. The 3D scanner as seen from below

The Time of Flight sensor and the smartphone must be on the same 2D coordinate, the adhesion makes sure that the coordinate is the same for both. The user must hold “the scanner” above the food portion, then by moving the smartphone on the surface in a zigzag motion the program creates the point cloud by scanning XY coordinates with an optical sensor and the Z coordinate with the time of light sensor. After we create the 3D map of the food portion, we can estimate its volume. With the help of the density database we will be able do the raw weight estimation of the food portion.

# Conclusion

As interesting as this project was, I am little disappointed that I was not able to create a working prototype, even thought it was not the mandatory part of the project. Other than that, I am satisfied with everything I have learned while working on this project. I had to learn Java, Android Studio and microcontrollers. Ten weeks is not enough time for the project of this magnitude, but I am happy with what I have been able to achieve in that time period. Even thought I have proved that linear movement cannot be measured with the hardware that we have in current smartphones, now we have a good foundation to continue building on when the new kinds of sensors will be included in the future smartphones.

# Links

<https://gitlab.labranet.jamk.fi/M3156/food-volume-estimation-kmutt-summer-2019>

References

Anthimopoulos, M., Dehais, J., Mougiakakou, S. & Shevchik, S. 2017. Two-view 3D Reconstruction for Food Volume Estimation. EEE Transactions on Multimedia 19-5, 1090-1099.

Arakawa, Y., Hippocrate, E., Suwa, H. & Yasumoto, K. 2016. Food Weight Estimation using Smartphone and Cutlery. Nara Institute of Science and Technology.

Boushey, C., Chae, J., Delp, E., Ebert, D., Lee, C., Kerr, D. & Schap, T. 2012. Comparison of Known Food Weights with Image-Based Portion-Size Automated Estimation and Adolescents’ Self-Reported Portion Size. Journal of Diabetes Science and Technology 6-2.

Cagnoni, S., Ciampolini, P., De Munari, I., Hassannejad, H., Matrella, G. & Mordonini, M. 2017. A New Approach to Image-Based Estimation of Food Volume. Algorithms.

Chan, D., Cui, Y., Schuon, S., Theobalt, C. & Thrun, S. 2010. 3D Shape Scanning with a Time-of-Flight Camera. Stanford University.