

3D Surface Reconstruction Using Infrared (IR) Signal

by

Nasuha binti Mohd Shaber

18248

Dissertation submitted in partial fulfilment of
the requirements for the
Bachelor of Engineering (Hons)
(Electrical and Electronic)

JANUARY 2017

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
32610 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

3D SURFACE RECONSTRUCTION USING INFRARED (IR) SIGNAL

by

**Nasuha binti Mohd Shaber
18248**

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(ELECTRICAL & ELECTRONICS)

Approved by,

Dr Siti Asmah Binti Daud

UNIVERSITI TEKNOLOGI PETRONAS

SERI ISKANDAR PERAK

January 2017

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

NASUHA BINTI MOHD SHABER

ABSTRACT

Three-Dimension (3D) surface reconstruction from object using an infrared sensor is becoming one of the widely used in research today. In this project data collected from infrared sensor will be used to reconstruct a 3D surface of an object. The sensor value data need to be converted into distance between infrared sensor to the surface of an object. One-Dimensional (1D) to two-dimensional (2D) conversion is carried out by using the angle between collection of each data and the distance between sensor and an object. The method to be used to reconstruct a 2D image by using conversion between Polar (angle-distance-axis) to Cartesian coordinate (x-y-axis) plane. The 2D images that were obtained will be used to reconstruct three-dimensional (3D) surface using a mesh grid algorithm and by acquiring the z-axis of the object. The accuracy of the reconstructed surface is carried out by measuring the dimensions of the reconstructed surface and reference object. The results from the experiment shown that by using infrared sensor value, a 3D reconstructed surface can be recreated and has an accuracy of 60% and above.

ACKNOWLEDGEMENT

I wish to thank various people for their contribution and guidance that they gave to me throughout this project. Firstly, I would like to express my deep gratitude to my project supervisor Dr Siti Asmah binti Daud, by taking me under her supervision and for her patient guidance, enthusiastic encouragement and useful critiques for this research work. I am also thankful to my friends that manage to share and teach their knowledge on the MATLAB software during the simulation process of my project. Finally, I wish to thank my parents for their support and encouragement during the project. Through all this people my project managed to be completed successfully.

TABLE OF CONTENTS

CERTIFICATION OF APPROVAL	ii
CERTIFICATION OF ORIGINALITY	iii
ABSTRACT	iv
ACKNOWLEDGEMENT	v
CHAPTER 1: INTRODUCTION	1
1.1 Background	1
1.2 Problem Statement	2
1.3 Objectives	3
1.4 Scope of Study	3
CHAPTER 2: LITERATURE REVIEW	4
2.1 Infrared Sensor Overview	4
2.2 Noise Filter Overview	5
2.3 Radial Range Sensors	6
2.4 1D and 2D Signal Overview and Conversion Method	7
2.5 3D Image Reconstruction Overview and Method	8
CHAPTER 3: METHODOLOGY	10
3.1 Project Methodology	10
3.2 Simulation of the 3D Surface Reconstruction	10
3.3 Simulation Process Flow Chart	14
CHAPTER 4: RESULTS AND DISCUSSION	15
4.1 Conversion of Sensor Value to Distance	15
4.2 Noise Filter in 1D Data	16
4.3 1D to 2D Conversion	24
4.4 2D to 3D Reconstruction	32
4.5 Dimensions Calculation	37
CHAPTER 5: CONCLUSION & RECOMMENDATION	42
REFERENCES	43

LIST OF FIGURES

FIGURE 1	Conversion of polar to Cartesian coordinates and vice versa in the first quadrant.	8
FIGURE 2	Simulation Process Flow Chart	14
FIGURE 3	Sensor Value vs Number of Data	15
FIGURE 4	Distance vs Number of Data	15
FIGURE 5	Filtered Cylinder 1D Data using Median Filter	16
FIGURE 6	Filtered Cylinder 1D Data using Butterworth Filter	16
FIGURE 7	Filtered Cylinder 1D Data using Mean Filter	17
FIGURE 8	Filtered Ellipse 1D Data using Median Filter	18
FIGURE 9	Filtered Ellipse 1D Data using Butterworth Filter	18
FIGURE 10	Filtered Ellipse 1D Data using Mean Filter	19
FIGURE 11	Filtered Hemisphere 1D Data using Median Filter	20
FIGURE 12	Filtered Hemisphere 1D Data using Butterworth Filter	20
FIGURE 13	Filtered Hemisphere 1D Data using Mean Filter	21
FIGURE 14	Filtered Rectangle 1D Data using Median Filter	22
FIGURE 15	Filtered Rectangle 1D Data using Butterworth Filter	22
FIGURE 16	Filtered Rectangle 1D Data using Mean Filter	23
FIGURE 17	Polar to Cartesian Coordination Graph for a Cylinder using Median Filter	24
FIGURE 18	Polar to Cartesian Coordination Graph for a Cylinder using Butterworth Filter	24
FIGURE 19	Polar to Cartesian Coordination Graph for a Cylinder using Mean Filter	25
FIGURE 20	Polar to Cartesian Coordination Graph for an Ellipse using Median Filter	26
FIGURE 21	Polar to Cartesian Coordination Graph for an Ellipse using Butterworth Filter	26
FIGURE 22	Polar to Cartesian Coordination Graph for an Ellipse using Mean Filter	27
FIGURE 23	Polar to Cartesian Coordination Graph for a Hemisphere using Median Filter	28

FIGURE 24	Polar to Cartesian Coordination Graph for a Hemisphere using Butterworth Filter	28
FIGURE 25	Polar to Cartesian Coordination Graph for a Hemisphere using Mean Filter	29
FIGURE 26	Polar to Cartesian Coordination Graph for a Rectangle using Median Filter	30
FIGURE 27	Polar to Cartesian Coordination Graph for a Rectangle using Butterworth Filter	30
FIGURE 28	Polar to Cartesian Coordination Graph for a Rectangle using Mean Filter	31
FIGURE 29	3D Reconstructed Cylinder using Median Filter	32
FIGURE 30	3D Reconstructed Cylinder using Butterworth Filter	32
FIGURE 31	3D Reconstructed Cylinder using Mean Filter	33
FIGURE 32	3D Reconstructed Ellipse using Median Filter	33
FIGURE 33	3D Reconstructed Ellipse using Butterworth Filter	33
FIGURE 34	3D Reconstructed Ellipse using Mean Filter	34
FIGURE 35	3D Reconstructed Hemisphere using Median Filter	34
FIGURE 36	3D Reconstructed Hemisphere using Butterworth Filter	34
FIGURE 37	3D Reconstructed Hemisphere using Mean Filter	35
FIGURE 38	3D Reconstructed Rectangle using Median Filter	35
FIGURE 39	3D Reconstructed Rectangle using Butterworth Filter	35
FIGURE 40	3D Reconstructed Rectangle using Mean Filter	36

LIST OF TABLES

TABLE 1	Comparison Table on Different Radial Range Sensors	6
TABLE 2	Dimension Value for Cylinder Diameter 1	37
TABLE 3	Dimension Value for Cylinder Diameter 2	37
TABLE 4	Dimension Value for Ellipse Major Axis 1	38
TABLE 5	Dimension Value for Ellipse Major Axis 2	38
TABLE 6	Dimension Value for Ellipse Minor Axis 1	38
TABLE 7	Dimension Value for Ellipse Minor Axis 2	38
TABLE 8	Dimension Value for Hemisphere Radius 1	39
TABLE 9	Dimension Value for Hemisphere Radius 2	39
TABLE 10	Dimension Value for Hemisphere Diameter 1	39
TABLE 11	Dimension Value for Hemisphere Diameter 2	39
TABLE 12	Dimension Value for Rectangle Width 1	40
TABLE 13	Dimension Value for Rectangle Width 2	40
TABLE 14	Dimension Value for Rectangle Length 1	40
TABLE 15	Dimension Value for Rectangle Length 2	40

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

Three-dimensional (3D) surface reconstruction is the process of recreating the shape of an object from multiple two-dimensional (2D) images. Major applications of the 3D surface reconstruction major applications are in prosthetic design particularly in the Magnetic Resonance Imaging (MRI) and Computerized Tomography (CT) scan. CT Scan is a generation from a combination of X-ray images while MRI are generated from magnetic field and radio wave energy. Even though these applications were widely used in the image reconstruction field, the cost of these applications are very expensive due to the sensors used [1-2].

There are many researches who tried to find an alternative way to create a low-cost 3D surface reconstruction application using low-cost infrared sensor [1-3]. Infrared sensor is widely used as a proximity sensor and obstacle avoidance rather than using it for image reconstruction. Nevertheless, infrared sensor recently is becoming an alternative option in 3D reconstruction for an object compared to ultrasonic or visual sensor due to its fast response and low cost [1].

Infrared sensor comes with a few downsides such as high sensitivity of the sensor which will creates noises in the signal. The noises will usually affect the acquired depth in images and causes inaccuracies and alterations along the borders of the objects that were scanned [4]. MATLAB software will be used to filtered out the infrared signal. The infrared sensor signal will be in one-dimensional (1D) array and converted into 2D array is required to reconstruct a 3D surface. The 3D surface reconstruction will then be measured and compared with a reference object to calculate its accuracy.

1.2 PROBLEM STATEMENT

The implementation of the infrared sensor (IR) attracted a lot of interest as it has a low cost compared to ultrasonic and visual sensors. Besides having a low cost, infrared sensor also has a faster response than the other sensor particularly the ultrasonic sensor. Blind spot can be reduced by using infrared sensor when the sensor was put in a rotating position with multiple sensors. Other sensor such as ultrasonic sensor cannot detect these blind spots as they have a narrower angular sensitivity [3].

However, one of the concern when using infrared sensor was the high sensitivity. They are unable to distinguish between objects that irradiate the same thermal energy level. The data that were collected from infrared sensor also contains noise that will pose a major obstacle in developing a 3D reconstruction surface. Noise from the signal will affect the reconstructed surface as it will not be smooth and accurate as the reference object. Hence, the signal need to be filtered out before converting the sensor value into distance.

On the other hand, the distance between the measured object to the infrared sensor need to be calculated based on the graph in the datasheet. However, as the graph provided are not linear, few derivations need to be done based on some equations to create an equation that can calculate the distance. Based on the equations an algorithm can be created to convert sensor value into distance easily.

Not only that but, another problem that can be encountered when using infrared sensor signal is that the sensor will be in 1D value. To create a 3D reconstruction surface one or multiple value of 2D images are needed. Therefore, the 1D signal need to be converted to 2D signal beforehand so that a 3D reconstruction surface can be reconstructed. Multiple algorithm need to be explored and tested to view which algorithm is the most suitable to convert 1D to 2D and from 2D to 3D.

1.3 OBJECTIVE

From the problem statement, three objectives were analysed and picked to overcome the problems arises:

- 1) To convert infrared sensor value into distance.
- 2) To reconstruct a 3D surface of an object by using data collected from infrared sensor
- 3) To measure the accuracy between the reconstructed surface with the reference object.

1.4 SCOPE OF WORK

The scope of work for this project is to reconstruct a 3D surface image from the infrared signal and compare the accuracy of the 3D surface image with the reference object. This experiment will only be conducted using the computer simulation software, the MATLAB. The infrared sensor signal needs to be converted into distance and any noise that produced with it need to be filtered out. Besides that, the sensor signal which is in 1D array index need to be converted into 2D array index, then a 3D array index can be reconstructed. In the end, the reconstructed 3D surface dimension needs to be measured, so that the accuracy could be calculated and compared between the 3D reconstructed surface with the reference object.

CHAPTER 2

LITERATURE REVIEW

2.1 INFRARED SENSOR OVERVIEW

Infrared sensor is a distance measurement components that are usually used in robot mapping and object reconstruction. Researchers are prone to use infrared sensor due its low cost and fast response time. Infrared sensors utilize the reflected light intensity to estimate the distance from an object but they depend on the reflectance properties of the object surfaces. The accuracy and range of information for an infrared sensor can be improved by increasing the number of infrared sensor. This is because it decreases the time to scan the surface image than using one sensor and on a more crucial part it reduces the blind spot of the sensor [2].

Currently, there are many research conducted on the application of infrared sensor but mostly in a mobile robot mapping application. Infrared sensor is the preferred sensor to be used in these research as it is a low-cost sensor and has a fast response. In a research done by P. Hyungwoon et. al, the SHARP GP2 series were used because it can measure the offset distance of point on photo-resistive strip where the reflected light hits [2]. On the mobile robot, infrared sensors ranging from five to twelve sensors were put in a rotating position to cover the blind spot. The higher the number of the sensor the higher the coverage area of the scanned object. These infrared sensors will have an input of 5 V and an analogue voltage output. These analogue voltage output are non-linear depending on its distance and need to be converted into digital number using an Analog Digital Converter (ADC). After converting the voltage output into distance, the mapping of the environment can be established [1-3]. From these research, almost 90% of the environmental area are correctly mapped in the computer screen [1-3].

2.2 RADIAL RANGE SENSOR

There are many radial range sensors that have the same functions as infrared sensor but they came with a higher cost. Some of the commonly used radial range sensors are the ultrasonic sensor, laser scanner and visual sensor. A laser scanner creates a high-resolution data but the pricing cost are very much higher than an ultrasonic or infrared sensor [2]. Meanwhile a visual sensor also provides a high-resolution information but they are very slow in computing its data with the addition of its high cost [1]. A conventional ultrasonic sensor on the other hand can measure the distance by measuring the time of flight (ToF) very accurately in its depth but not in its angle measurement (azimuth). The ToF is the elapsed time between the emission and subsequent arrival after ultrasound pulse train reflection travels. The transducers used for ultrasound researchers are usually big in angular sensitivity lobe but has a poor angular resolution [3].

Infrared sensor has more advantages compared to radial range sensors. One of the benefits are the faster response times compared to the other sensor. It is very striking due to the importance of enhancing the real-time operation in mobile robots and image processing [3]. The infrared sensor is different from ultrasound sensor as it has a narrower angular sensitivity and the distance are measured from the offset of the two elements which are the intensity of the reflected light or beam. The infrared sensor however has a limit in maximum distance. The maximum distance is smaller than the ultrasonic sensor maximum distance but it can be overcome by rotating the infrared sensor in a small amount of angle in each step [2-3]. There is also infrared sensor that are based on phase shift measurement which has a medium distance detection but are usually expensive [3].

However, infrared sensor also has disadvantages which the other radial range sensor does not have which is the sensitivity of the sensor. The infrared sensor is highly sensitive and will create unwanted noises in the signal that were collected. The noises will usually affect the acquired depth in images and causes inaccuracies and alterations along the borders of the objects that were scanned. One of the sources are natural illumination which happens whenever the light sources in the scene is uncontrollable. Due to this event, the infrared sensor will collect a significant quantity of light radiation that are not provided from the surface itself. Besides that, the source of this

noise come from the diffraction of light which leads to difficulties in analysing the radiated structured pattern throughout the object boundaries. This will eventually lead to irregular object contours and affect the accuracy of the 3D surface reconstruction. Other factors that creates noise is the shot noise which is related to the radiation, thermal noise and A/D conversion quantization noise [4].

Based on the radial range sensors that functions the same as infrared sensor, a comparison table is done.

TABLE 1: Comparison Table on Different Radial Range Sensors

Sensor	Infrared	Ultrasound	Laser Scanner	Visual
Range	Short distance (under 25cm) [3]	Long distance (1m) [3]	-	-
Cost	Low cost [1-3]	High cost [2]	High cost [2]	High cost [1]
Data Quality	Low quality due to noise [4]		High quality [2]	High quality [1]
Reaction Time	Fast [3]	Slower than infrared [3]	-	Slow [1]

2.3 NOISE FILTERS OVERVIEW

Noise reduction is the process of reducing or eliminating noise in a signal. Infrared sensor signal is usually affected with noise due to its sensitivity. To overcome this problem, filters are used to smoothen the data so that the reconstructed surface will be smooth. The noise will be filtered after conversion of sensor value to distance are calculated. Simple 1D filters were analysed to view its advantages and disadvantages.

Mean filter is one of the filter used to filter one dimension data. Mean filter can also be called moving average filter and it smoothed image data. The filter is done by performing spatial filtering on each data using a rectangular window. The window length is picked beforehand. The sum of pixel within the window will be divided by the window length to calculate the average value of the pixel within a window. The centre pixel in the window will then be replaced with the average value. One example is when the pixel value within the window is [1 , 4 , 3 , 2 , 5 , 6 , 7], the sum of the pixel 28 is divided by the window length which is 7 to get the filtered pixel. The centre pixel will be replaced with the new value which is 4. The window will move across all

the pixel to filter the data and considered as a sliding-window spatial filter. The mean filter also has common names such as box filter, averaging filter and smoothing filter. [5]

Besides that, median filter is also a sliding-window spatial filter but it has a better reliability than mean filter. The difference between median filter and mean filter is that the median filter replaces the centre value in the window with the median of the data within the window. One example as below are if the value data within the window are [0 , 10 , 19 , 3 , 2 , 4 , 6 , 3 , 97] the data will be rearranged in an increasing value like [0 , 2 , 3 , 3 , 4 , 6 , 10 , 19 , 97]. Therefore, the value 4 is picked as the new value for the window as it is the median value in the window. In comparison with mean filter if the values above are used the calculation is $144/9 = 16$. The value 16 is returned. This proves that median filter can remove 'impulse' noise and does a better work in decreasing the blurring in images. [5]

2.4 1D AND 2D SIGNAL OVERVIEW AND CONVERSION METHOD

1D signal is the measurement of one value and will create a line when plotted in a graph. One of the 1D example is the measurement of distance or length. Meanwhile a 2D signal is plotted in flat planes which have two values. One of the examples are the length and width of an object. Sensors that use the Time-of-Flight (ToF) measurement can measure 1D data only and conversion is needed to obtain 2D data. [6]

1D sensor signal can be converted to 2D image by using a computer software. One of the methods is by making use of the angle differences during data acquisition. From this angle, polar coordinates can be plotted with the distance creating an angle versus distance graph. As it is well known the Time-of-Flight (ToF) sensors can produce 1D and polar coordinates, many researches have been done to implement the method of converting Polar coordinates to Cartesian coordinates to get a 2D image. [6-8]

The most common method used in previous research to convert a polar to Cartesian coordinates is done based on the Pythagoras theorem. The distance is considered as r and the angle needs to be converted to theta value θ . Two equations are used to convert Polar coordinates (r, θ) to Cartesian Coordinates (x, y) . [9]

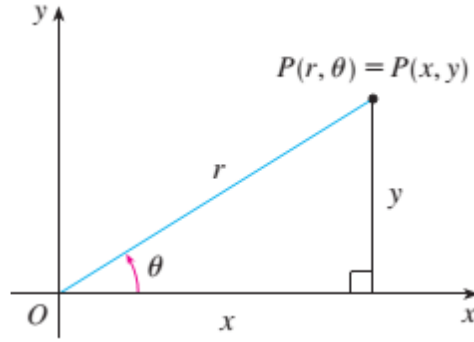


FIGURE 1: Conversion of polar to Cartesian coordinates and vice versa in the first quadrant. [9]

$$x = r \times \cos(\theta) \quad (1)$$

$$y = r \times \sin(\theta) \quad (2)$$

2.5 3D IMAGE RECONSTRUCTION OVERVIEW AND METHOD

3D image is an image that has three measurements which are the length, width and depth of an object. 3D surface reconstruction is the process of reconstructing the shape of an object from multiple two-dimensional (2D) images. 3D image reconstructions are widely used in the medical imaging field, remote sensing, object detection and avoidance, industrial inspection and many others.

Some of the 3D surface reconstruction major applications are in prosthetic design particularly in the Magnetic Resonance Imaging (MRI) and Computerized Tomography (CT) scan. CT Scan are generated from combination of X-ray images while MRI are generated from magnetic field and radio wave energy. [10].

Commonly, 3D image reconstruction methods will be divided into two categories which are passive and active techniques. The passive methods depend on 2D images to extract surface information from images while the active method uses the calibrated light sources which are commonly from lasers or coded light. Most of the researchers uses the shape that come from the structured light method for the application of the active method [11].

For a passive method, the technique is done by using cameras from two or more points of view and it imaged a scene which will help to found the correspondence between all the images. The correspondence will then be triangulated and a 3D position

will be obtained from it [12]. Some strategies were used in passive techniques to achieved a better result such as having more than two cameras and taking images in a short time interval which will make it easier to find correspondences and create a prototype that are more accurate [13]. However, there are disadvantages in using a passive technique which is constraint in a texture less scenes and makes it harder to find correspondence [12].

In contrast, active method is done by replacing one of the cameras in passive method to a projection device which will results in a combination of a camera and an illumination source. The light source is a form of temporary module or an illumination which will be helpful in this method and simplifies the steps in capturing 3D surface [13]. Active method solves the limitation in texture less scenes by projecting an artificial feature on the object's surface which will be used as a correspondence. Therefore, the 3D reconstruction can be done by comparing the differences between the imaged patterns and the projected patterns in which case it is the same as passive method [12]. However, the active method also has its downsides which are the constraint in time when taking the image. This is because, only one image can be taken at a time when a projected light is projected at a spot. To solve this problem, researchers usually use a super-fast imager but it comes with a high cost. Other recommendations are by using a cylindrical lens in front of the light source to form a wider projected image therefore reducing time in taking an image [13].

CHAPTER 3

METHODOLOGY/PROJECT WORK

3.1 PROJECT METHODOLOGY

The objective of this project are to reconstruct a 3D surface from the infrared signal and measured the dimension of the reconstructed images. The research on suitable 3D surface reconstruction algorithm and infrared sensor are carried out in the beginning of the project. The simulation part of the project is done in the second half of the project.

3.2 SIMULATION OF 3D SURFACE RECONSTRUCTION

The acquired infrared sensor signal will be uploaded in the MATLAB. The distance can be acquired based on the voltage versus distance graph provided by the Sharp IR datasheet that were used during data collection. However, the graph that were provided are not linear an algorithm is required in order to convert infrared sensor value into distance.

Fortunately, Sharp has introduced a method to convert the sensor data based on the linear graph of voltage output versus the inverse number of distance that are provided in the datasheet. A few derivations need to be done based on some equations to achieved a straight equation that can convert sensor value to distance. The first equation is the voltage equation 3 which were used to find voltage based on the range of the infrared sensor which are 4 to 30 cm. This equation will also produce a linear graph that can be used to calculated distance. Based on the calibration point in the datasheet for Sharp GP2D120 the value 0.42 is used as a constant value in this equation 3.

$$Voltage = \frac{1}{(Range + 0.42)} \quad (3)$$

$$y = m \cdot x + b \quad (4)$$

In a linear equation, as (4) the y value is set up as the inverse number of distance while the x value is the voltage value. By substituting the equation (3) into (4) this will yield:

$$\frac{1}{(\text{Range} + 0.42)} = m \cdot \text{Voltage} + b \quad (5)$$

The equation (5) is rearranged so that range will be the function of voltage. The range will act as a distance and a new equation will be obtained as below.

$$\text{Distance} = \frac{1}{m \cdot \text{Voltage} + b} - 0.42 \quad (6)$$

$$\text{Distance} = \frac{m'}{\text{Voltage} + b'} - 0.42 \quad (7)$$

$$\text{where } m' = \frac{1}{m} \text{ and } b' = \frac{b}{m}$$

The m and b values are determined from a calibration procedure that has been done during the data gathering process. The calibration procedure yields constant values for m and were substituted in the equation. The following formula were derived for the sensor [14-15]:

$$\text{Distance} = \frac{2914}{(\text{Voltage} + 5)} - 1 \quad (8)$$

After obtaining the distance from the sensor value, the data will be filtered out first using three different filter which are the mean, median and Butterworth filter. The three filter is tested to view which is more reliable. The mean filter functions by calculating the average of pixel in a window. Equation 9 is used to calculate the average of pixel. In this equation, $x[]$ is the input data, $y[]$ is the output after data is filtered, N is the window size and k is the current pixel number that is filtered.

$$y(k) = \frac{x(k - N) \dots + x(k - 1) + x(k) + x(k + 1) \dots x(k + N)}{N} \quad (9)$$

Meanwhile the median filter is carried out by replacing the centre pixel in a window with the median value within the pixels in the window. The equation 10

illustrates how the median pixel is picked among all the pixel where $x[]$, $y[]$, N and k are representing the same value as in mean equation.

$$y(k) = \text{med}\{x(k - N), \dots, x(k - 1), x(k), x(k + 1), \dots, x(k + N)\} \quad (10)$$

On the other hand, the Butterworth filter are designed to ensure a flat magnitude response possible in the passband. The magnitude response of the low-pass Butterworth filter is written as equation 10 where n is an order of the filter and ω_0 is a 3dB-limit frequency.

$$|G(j\omega)|^2 = \frac{1}{1 + (\frac{\omega}{\omega_0})^{2n}} \quad (11)$$

After that, the 1D sensor signal will be converted into 2D array index. The method to be used in converting 1D to 2D is by using polar to Cartesian coordinates conversion. The angle data that were taken during the acquiring of the distance data is used as Polar coordinates. The angle will be in theta, θ and the distance data is the radius, r based on the equation provided. The Cartesian coordinates are calculated using the Pythagoras theorem calculation using the theta and radius value. The two equations below are used to acquire the Cartesian coordinates:

$$x = r \times \cos(\theta) \quad (12)$$

$$y = r \times \sin(\theta) \quad (13)$$

During the process of converting 1D to 2D, a problem was encountered. A gap was formed between the first and last data. To overcome this complication a duplicate data same as the first data is added in the ending of the sets of data to fill in the gap. The first sensor value data is considered as the sensor value data scanned on 0° of rotation during data gathering and the last data is the data scanned on 360° of rotation. Hence, there will be 181 sensor data and 181 angle starting from 0° to 360° . This is done after the 1D data is filtered and before the conversion of 1D to 2D. By modifying this data, the 2D image of a cylinder can be plotted smoothly without having any gap between the first and last data.

The conversion of 2D to 3D array index is carried out using the mesh grid function to the x-axis (length), y-axis(width) and z-axis(height) value of the object. The height of the objects that is 10cm and the length and width of the objects scanned

for each height are all the same. Therefore, the value for x-axis and y-axis are duplicated along the z-axis and were plotted in a 3D graph.

In the end, the dimension of the reconstructed 3D surface will be compared with the original measurement of the reference object to view its accuracy. To measure the dimension of the reconstructed 3D surface, a distance measurement tool is used in the Matlab toolbox called *imdstline*. The distance tool is a draggable line on an axis, that measures the distance between the two endpoints of the line. The distance tool displays the distance in a text label superimposed over the line.

Based on the outcome, a conclusion will be made to justify whether the infrared sensor can successfully reconstruct a 3D surface.

3.3 SIMULATION PROCESS FLOW CHART

Based on the flow chart for simulation process, the first step is to upload data signal into the MATLAB. After that, the infrared sensor signal needs to be converted into distance and noise will be filtered out. The sensor signal which is in 1D array index then need to be converted into 2D array index and 3D array index. In the end, the reconstructed 3D surface dimension needs to be measured, so that the accuracy could be calculated and compared between the 3D reconstructed surface with the reference object.

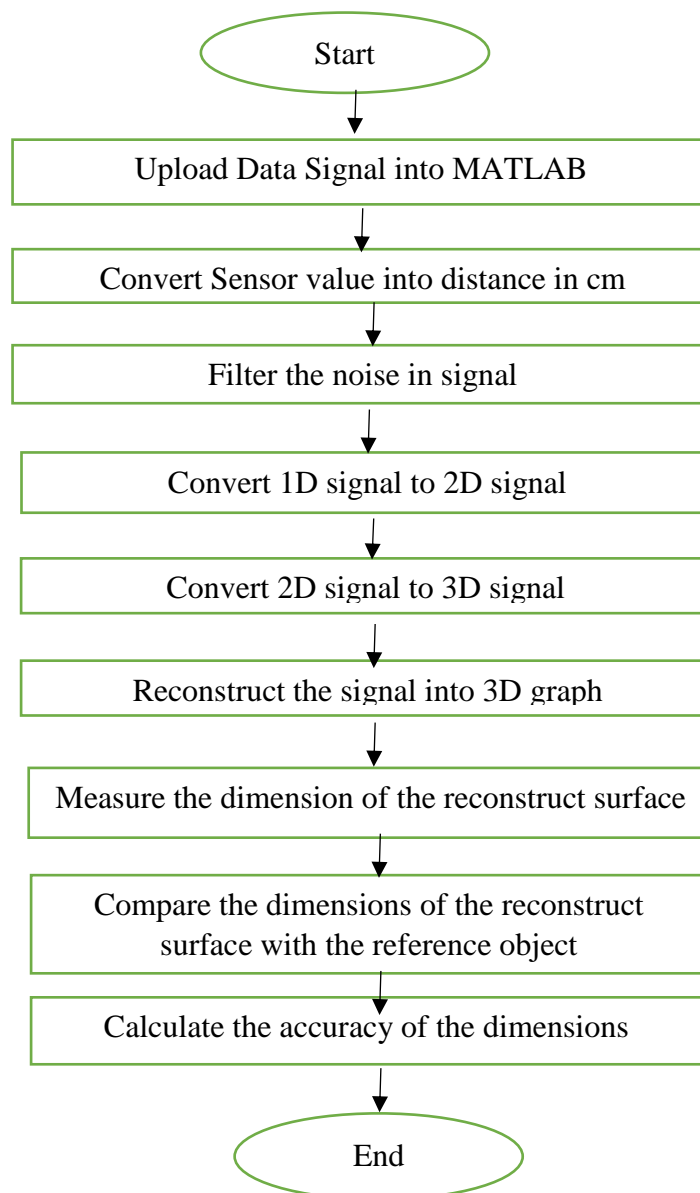


FIGURE 2: Simulation Process Flow Chart

CHAPTER 4

RESULTS AND DISCUSSION

4.1 CONVERSION OF SENSOR VALUE TO DISTANCE

The conversion was done in MATLAB by keying in the equation to convert sensor value to distance. The sensor value range are around 400 above while the distance range should be around 3cm and above. Figure 3 shows the sensor value versus number of data graph and Figure 4 shows the converted sensor value into distance versus number of data graph. Based on Figure 4 the range of the distance are around 4cm which proves that the sensor value is successfully converted.

$$Distance = \frac{2914}{(Voltage + 5)} - 1 \quad (8)$$

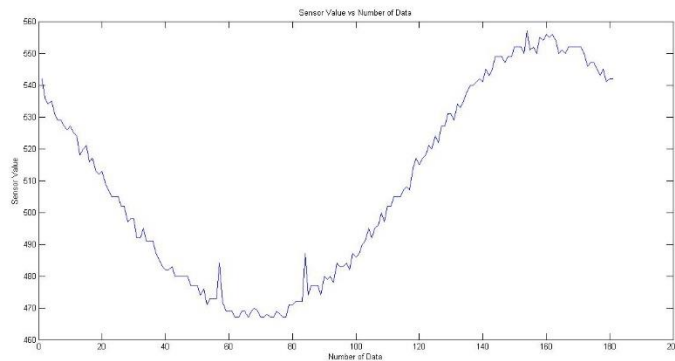


FIGURE 3: Sensor Value vs Number of Data

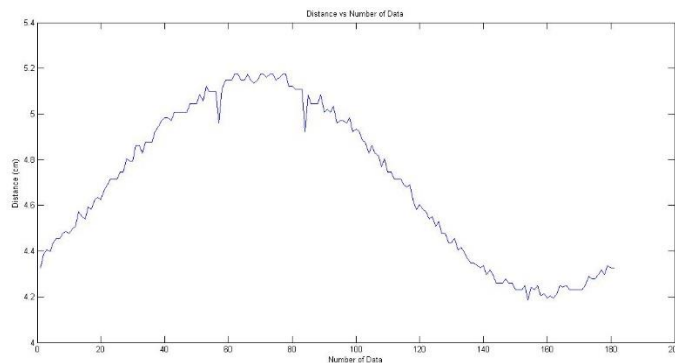


FIGURE 4: Distance vs Number of Data

4.2 NOISE FILTER IN 1D DATA

Three different filter were used to eliminate noise which are Median, Butterworth and Mean filter. In this project, four different polygon shape are used to reconstruct a 3D surface. The data become smooth after using the Median filter but there are a few that are smoother after using the Butterworth or Mean filter.

4.2.1 Cylinder

Figure 5, 6 and 7 shows the filtered 1D signal for a cylinder shape using Median, Butterworth and Mean filter. Based on these results, the Butterworth filter managed to smoothen the data as flat as possible compared to Median and Mean filter. However, the data between 170 to 180 (4cm to 4.2cm) deviates a little from the original value. This may affect the reconstructed 2D image and will be discussed more in the 1D to 2D conversion process result.

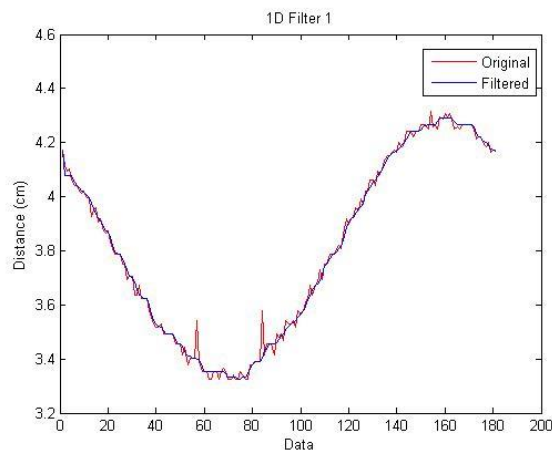


FIGURE 5: Filtered Cylinder 1D Data using Median Filter

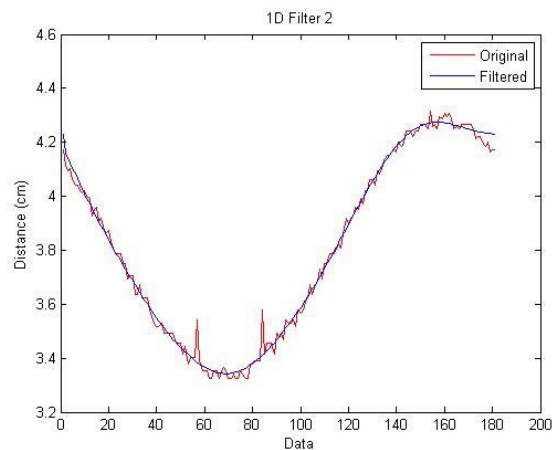


FIGURE 6: Filtered Cylinder 1D Data using Butterworth Filter

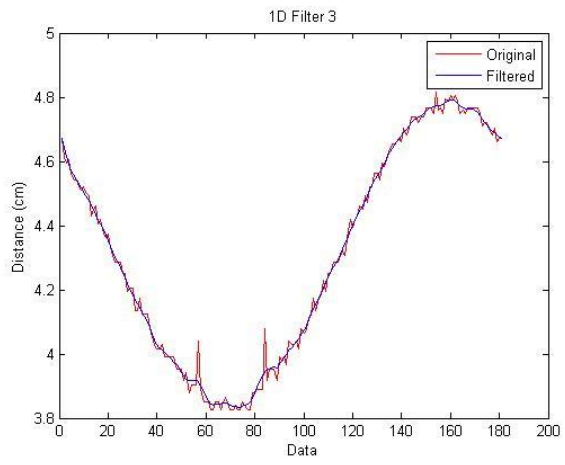


FIGURE 7: Filtered Cylinder 1D Data using Mean Filter

4.2.2 Ellipse

Figure 8, 9 and 10 shows the filtered 1D signal for an ellipse shape using Median, Butterworth and Mean filter. Based on these results we can conclude that all the filter managed to smoothen data in the same amount. However, the Median and Butterworth filtered data value between 170 to 180 (4cm to 4.2cm) deviates a little from the original value while the Mean filtered data are still in line with the original value.

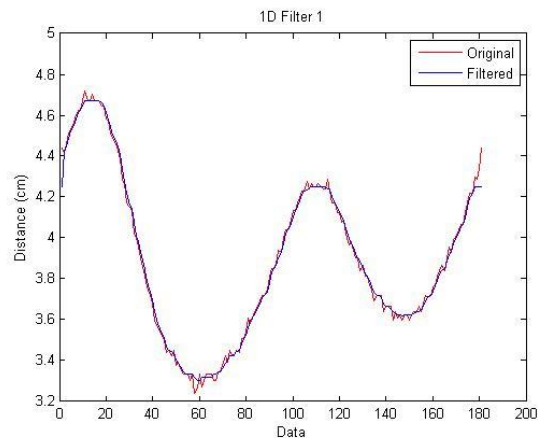


FIGURE 8: Filtered Ellipse 1D Data using Median Filter

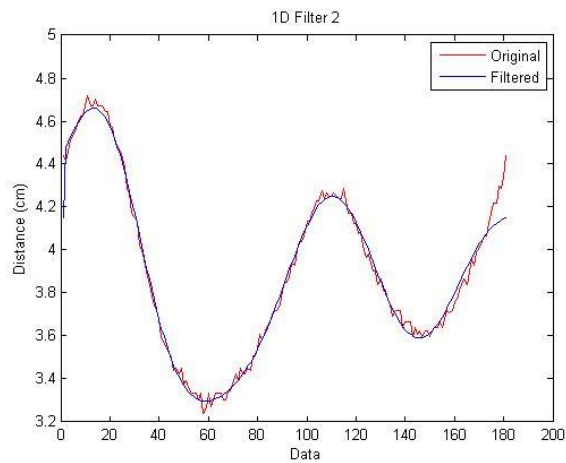


FIGURE 9: Filtered Ellipse 1D Data using Butterworth Filter

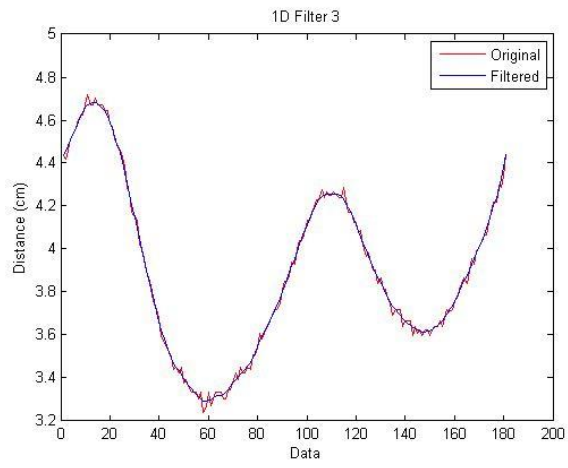


FIGURE 10: Filtered Ellipse 1D Data using Mean Filter

4.2.3 Hemisphere

Figure 11, 12 and 13 shows the filtered 1D signal for a hemisphere shape using Median, Butterworth and Mean filter. Based on these results, the filter managed to smoothen data in the same amount same as the ellipse data. However, the Butterworth filtered data value between 50 to 180 (4.4cm to 5.4cm) deviates from the original value while the Median and Mean filtered data are still in line with the original value.

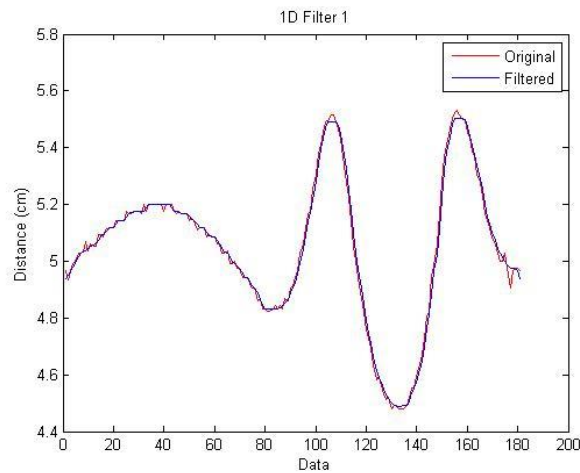


FIGURE 11: Filtered Hemisphere 1D Data using Median Filter

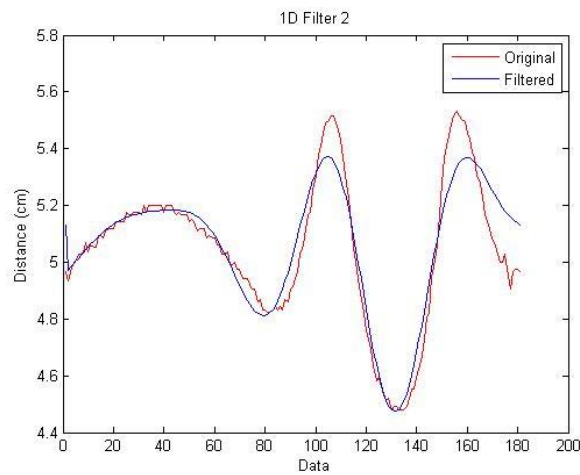


FIGURE 12: Filtered Hemisphere 1D Data using Butterworth Filter

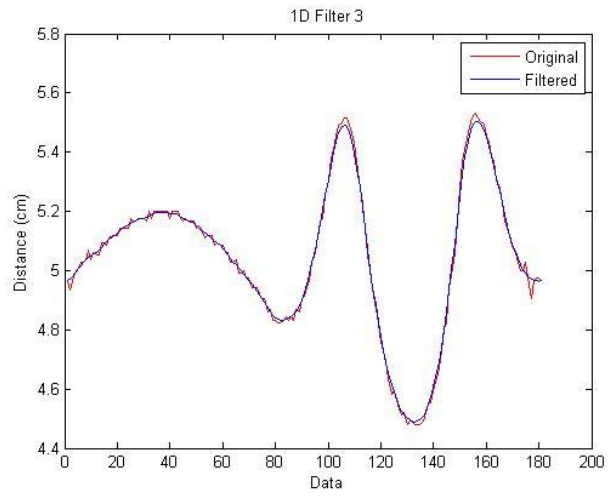


FIGURE 13: Filtered Hemisphere 1D Data using Mean Filter

4.2.1 Rectangle

Figure 14, 15 and 16 shows the filtered 1D signal for a rectangle shape using Median, Butterworth and Mean filter. Based on these results, Butterworth filter managed to smoothen data while the median and mean filtered data manage to remove the noise but not as smooth as Butterworth filter. However, the Butterworth filtered data value between 150 to 180 (4.1cm to 4.5cm) deviates from the original value while the Median and Mean filtered data are still in line with the original value.

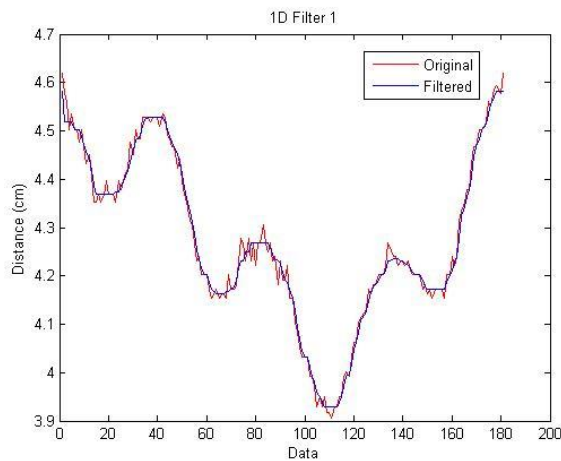


FIGURE 14: Filtered Rectangle 1D Data using Median Filter

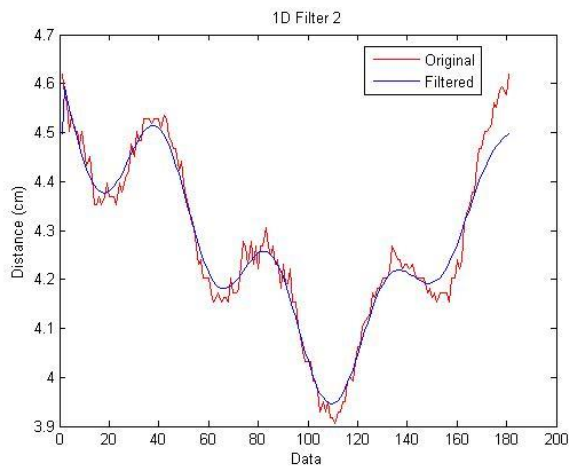


FIGURE 15: Filtered Rectangle 1D Data using Butterworth Filter

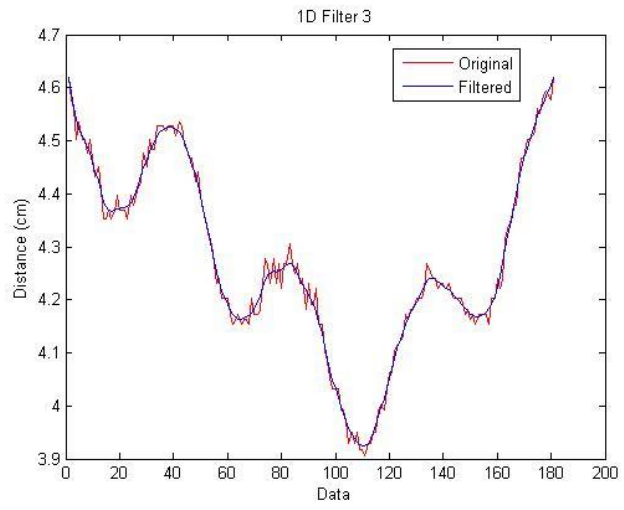


FIGURE 16: Filtered Rectangle 1D Data using Mean Filter

4.3 1D TO 2D CONVERSION

4.3.1 Cylinder

Based on the cylinder 1D data, a 2D graph is plotted by using the Polar to Cartesian coordinates conversion. Figure 17,18 and 19 shows the plotted 2D image of cylinder filtered by three different filter. For Figure 17 and 18 we can see that there are a few rigid parts between the 4 to 5 cm in the x-axis and 0 to 1 cm in the y-axis. Meanwhile, in Figure 19, the reconstructed cylinder 2D image filtered using mean filter is the smoothest among the three image.

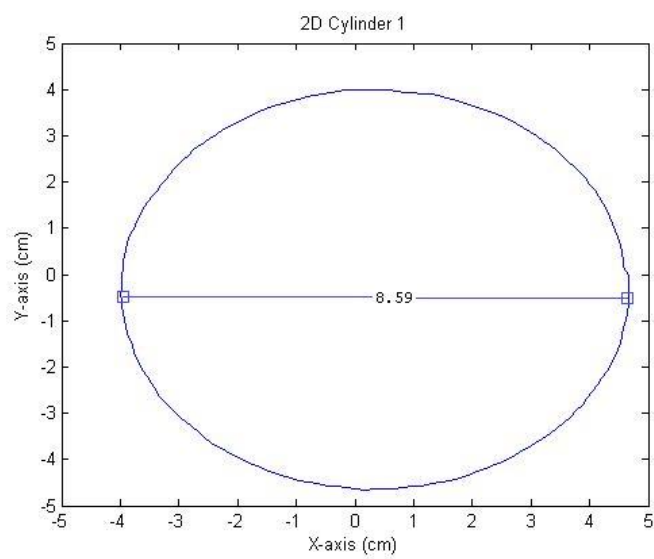


FIGURE 17: Polar to Cartesian Coordination Graph for a Cylinder using Median Filter

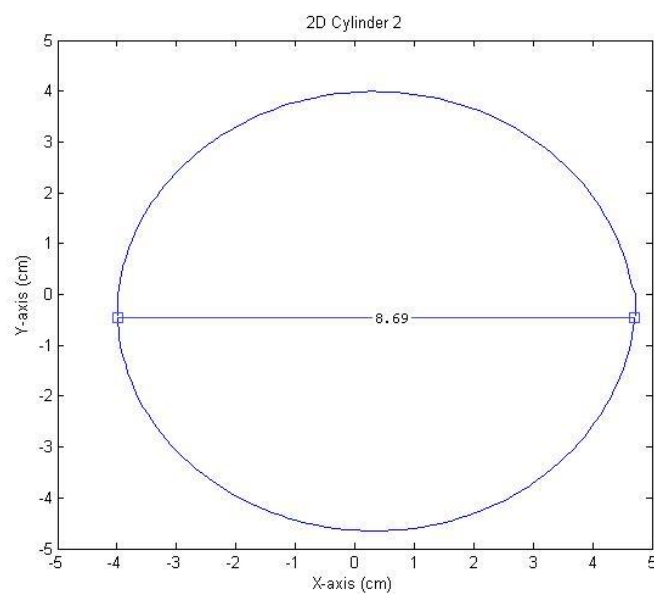


FIGURE 18: Polar to Cartesian Coordination Graph for a Cylinder using Butterworth Filter

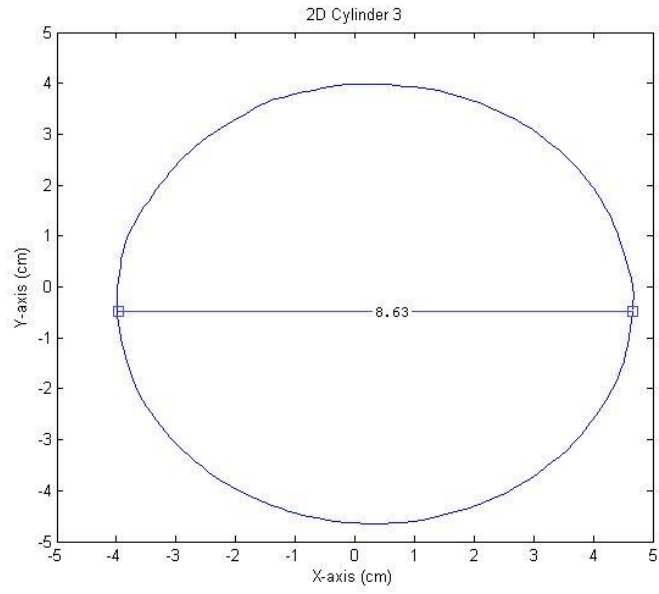


FIGURE 19: Polar to Cartesian Coordination Graph for a Cylinder using Mean Filter

4.3.2 Ellipse

Figure 20, 21 and 22 shows the plotted 2D image of an ellipse filtered by three different filter. The 2D image is considered as an ellipse because the length and width of the reconstructed image are not the same making it looks like an ellipse. The value of the length and width are recorded in the dimension measurement result part. Likewise, with the cylinder result, for Figure 21 and 22 there are a few rigid parts between the 4 to 5 cm in the x-axis and -1 to 0 cm in the y-axis. Meanwhile, in Figure 23, the reconstructed ellipse 2D image filtered using mean filter is the smoothest among the three image.

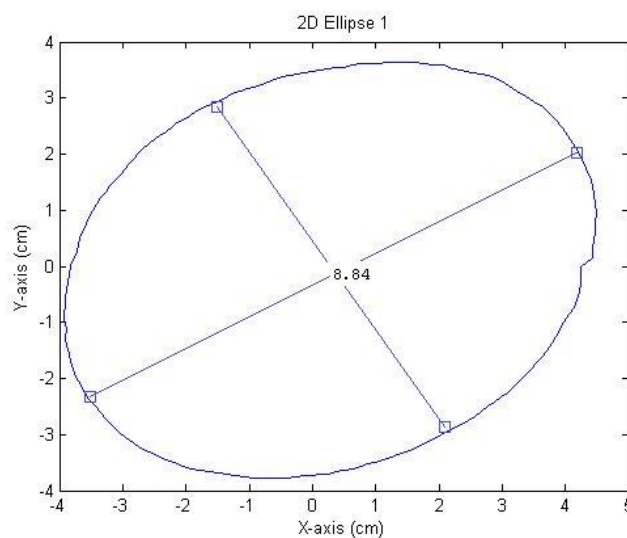


FIGURE 20: Polar to Cartesian Coordination Graph for an Ellipse using Median Filter

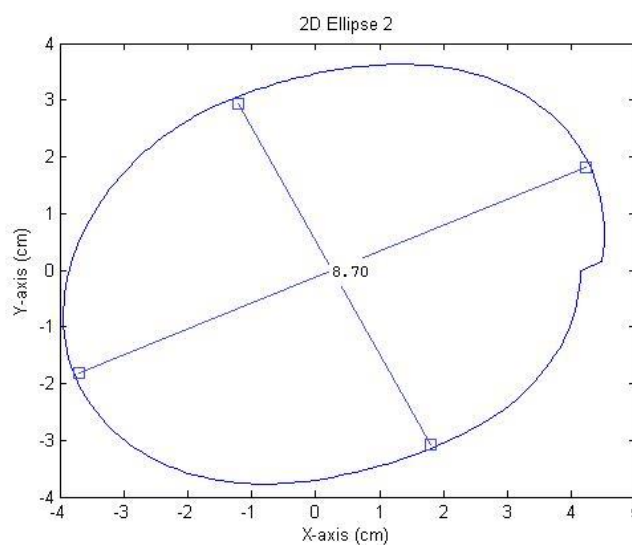


FIGURE 21: Polar to Cartesian Coordination Graph for an Ellipse using Butterworth Filter

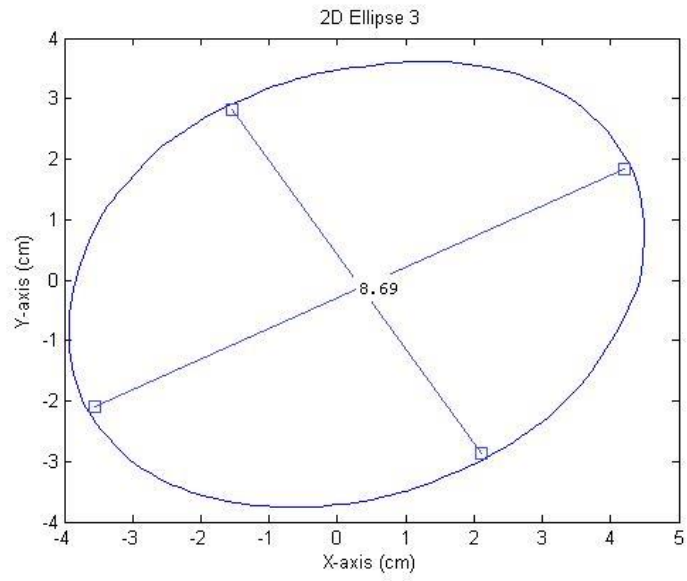


FIGURE 22: Polar to Cartesian Coordination Graph for an Ellipse using Mean Filter

4.3.3 Hemisphere

Figure 23, 24 and 25 shows the plotted 2D image of hemisphere filtered by three different filter. For Figure 24, the reconstructed 2D image does not resemble a hemisphere because the Butterworth filter has smoothed the edges making the edges look rounder. Meanwhile in Figure 23 and 24, the reconstructed hemisphere 2D image filtered using mean and median filter managed to produce a hemisphere shape same as the reference object shape.

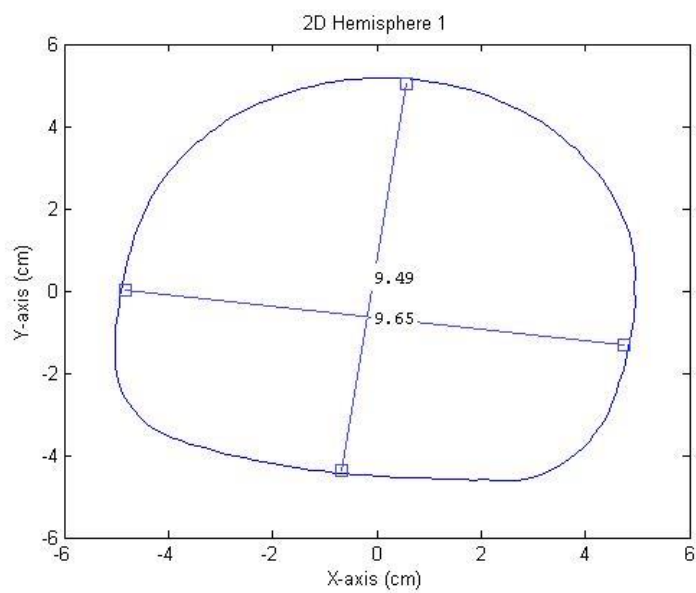


FIGURE 23: Polar to Cartesian Coordination Graph for a Hemisphere using Median Filter

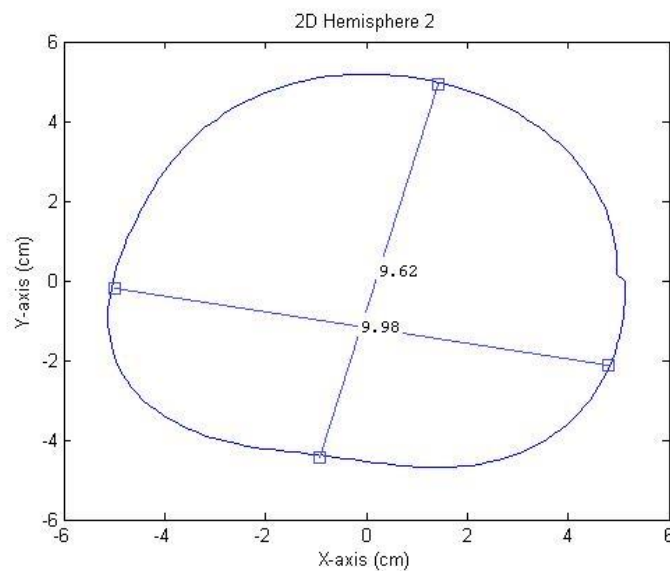


FIGURE 24: Polar to Cartesian Coordination Graph for a Hemisphere using Butterworth Filter

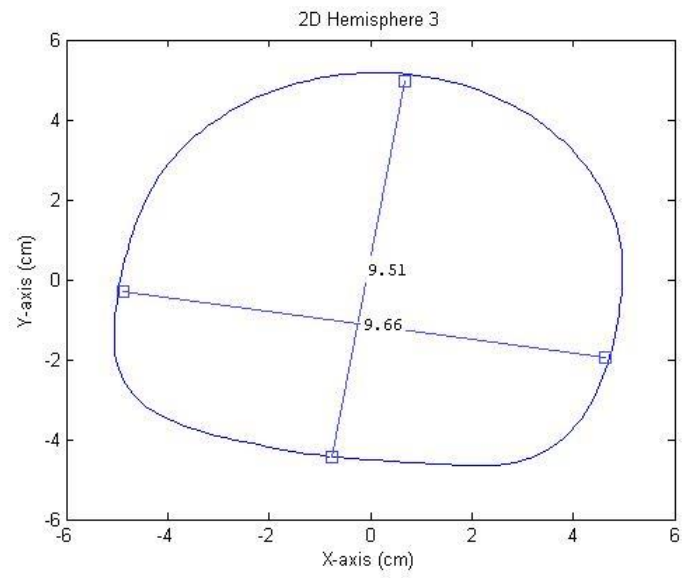


FIGURE 25: Polar to Cartesian Coordination Graph for a Hemisphere using Mean Filter

4.3.4 Rectangle

Figure 26, 27 and 28 shows the plotted 2D image of rectangle filtered by three different filter. Likewise, with the hemisphere result, the reconstructed 2D image in Figure 27 does not resemble a rectangle because the Butterworth filter has smoothed the edges making the edges rounder. Meanwhile in Figure 26 and 28, the reconstructed hemisphere 2D image filtered using mean and median filter managed to produce a rectangle shape but the edges are still rounder than the reference object.

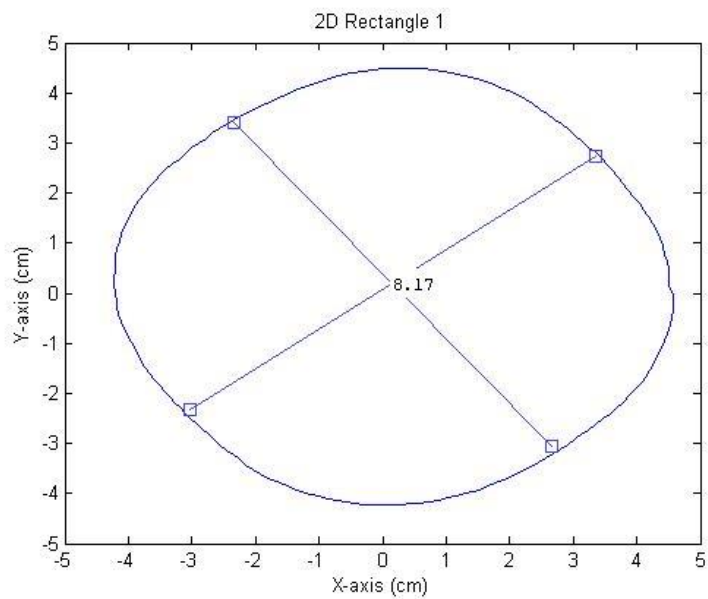


FIGURE 26: Polar to Cartesian Coordination Graph for a Rectangle using Mean Filter

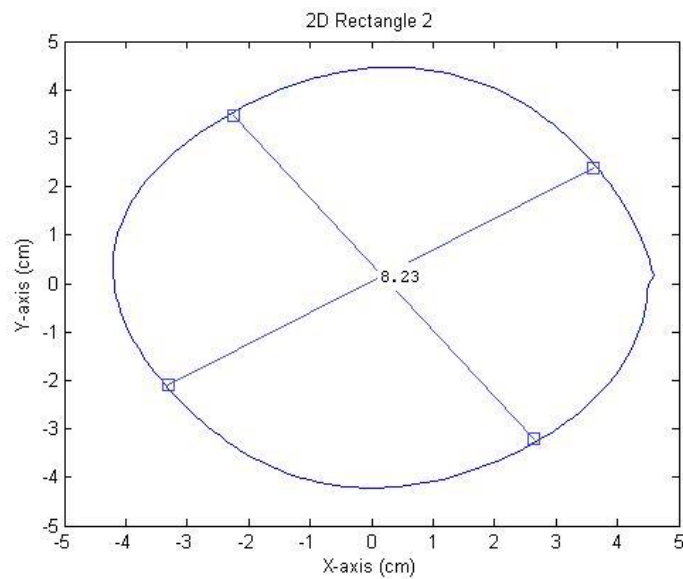


FIGURE 27: Polar to Cartesian Coordination Graph for a Rectangle using Mean Filter

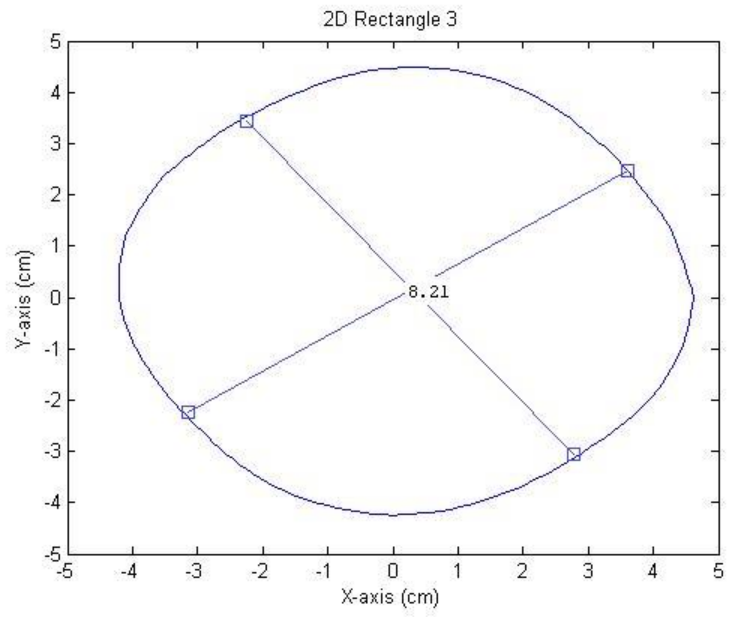


FIGURE 28: Polar to Cartesian Coordination Graph for a Rectangle using Mean Filter

4.4 2D TO 3D RECONSTRUCTION

The 3D reconstruction is produced using the x-axis, y-axis and z-axis of the object and applying a meshgrid to the axis. Figure 29 until 37 shows the reconstructed 3D surface shape matches with the reference object which is a cylinder, ellipse and hemisphere. However, for the rectangle reconstructed surface in Figure 38, 39 and 40, the edges are smooth and rounder compared to the reference object. This is due to high noise that makes the value inaccurate when an infrared sensor infrared sensor capture the borders of the rectangle.

4.4.1 Cylinder

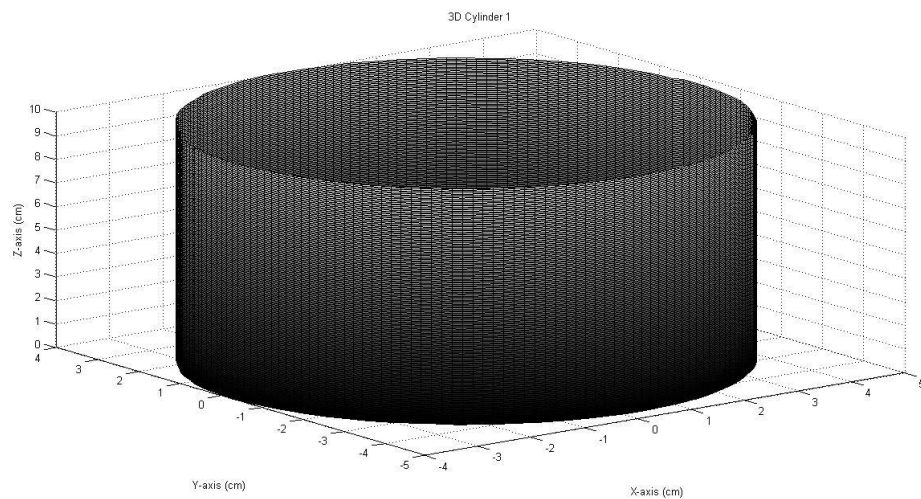


FIGURE 29: 3D Reconstructed Cylinder using Median Filter

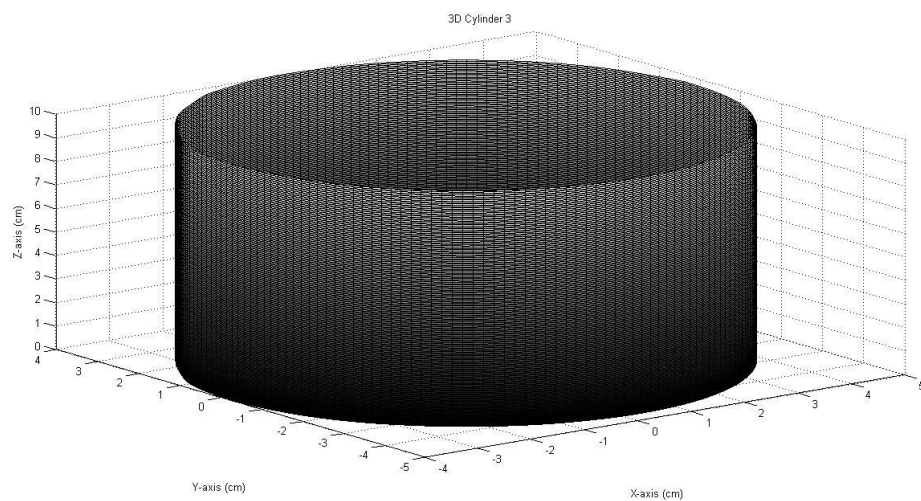


FIGURE 30: 3D Reconstructed Cylinder using Butterworth Filter

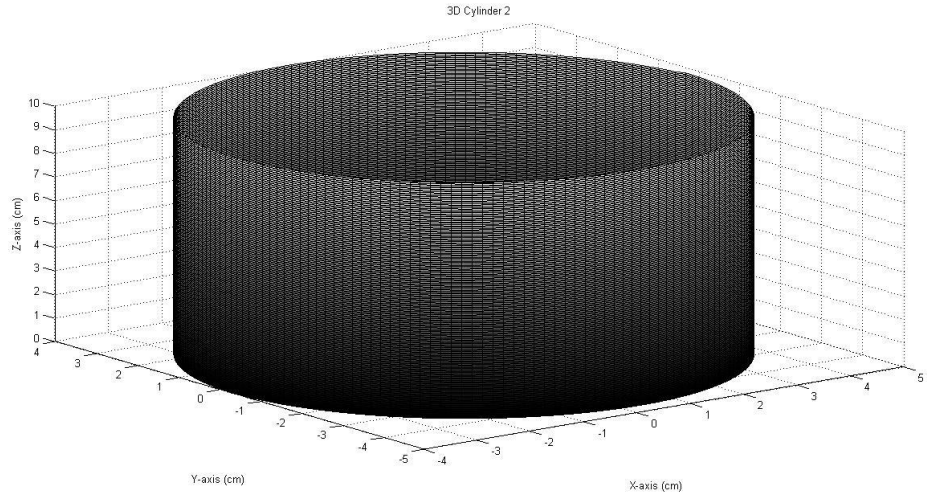


FIGURE 31: 3D Reconstructed Cylinder using Mean Filter

4.4.2 Ellipse

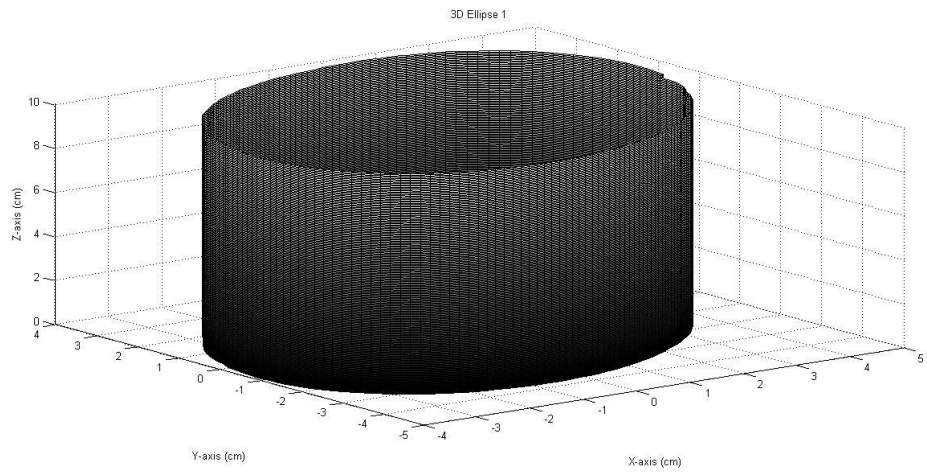


FIGURE 32: 3D Reconstructed Ellipse using Median Filter

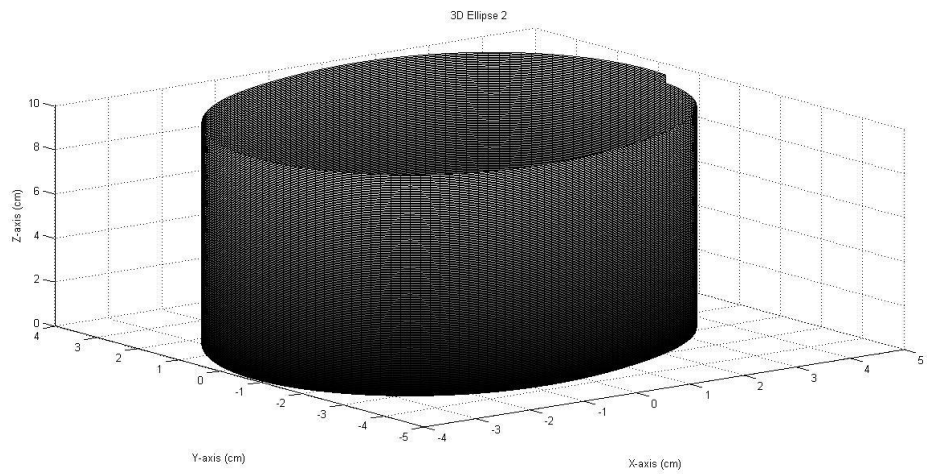


FIGURE 33: 3D Reconstructed Ellipse using Butterworth Filter

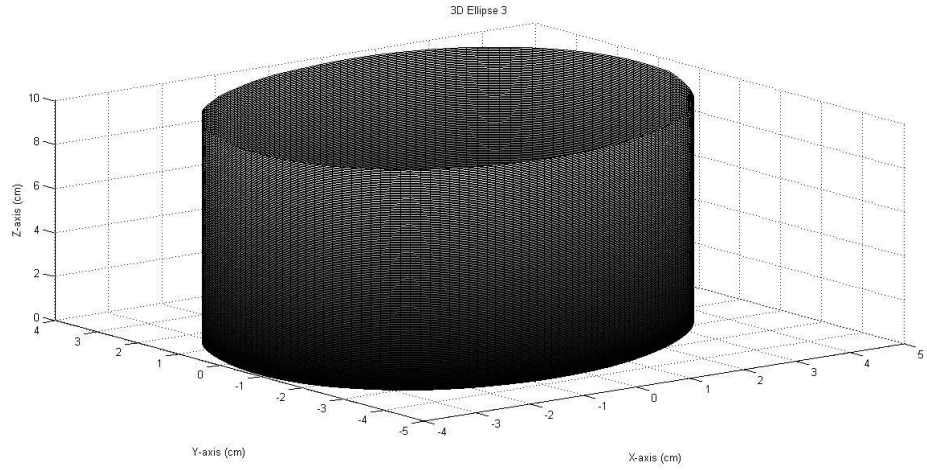


FIGURE 34: 3D Reconstructed Ellipse using Mean Filter

4.4.3 Hemisphere

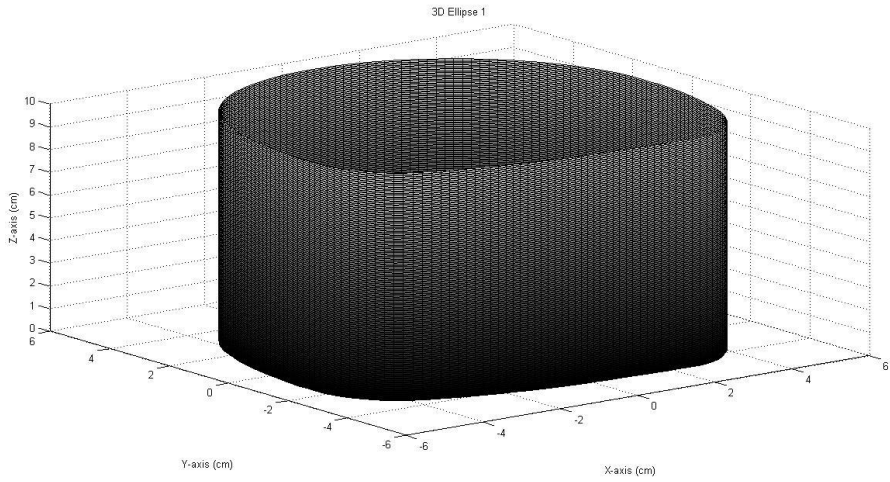


FIGURE 35: 3D Reconstructed Hemisphere using Median Filter

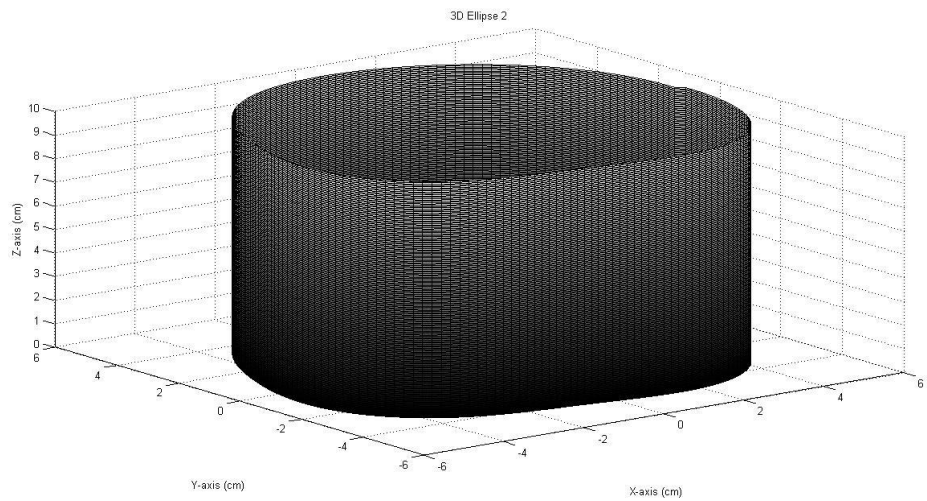


FIGURE 36: 3D Reconstructed Hemisphere using Butterworth Filter

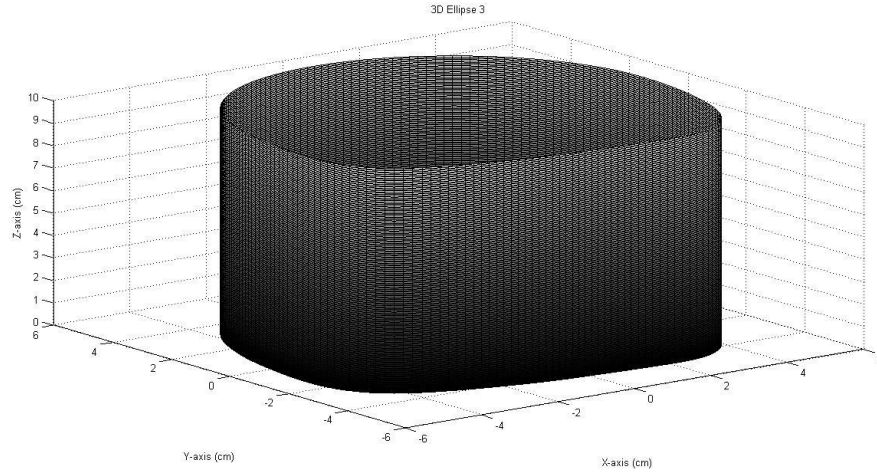


FIGURE 37: 3D Reconstructed Hemisphere using Mean Filter

4.4.4 Rectangle

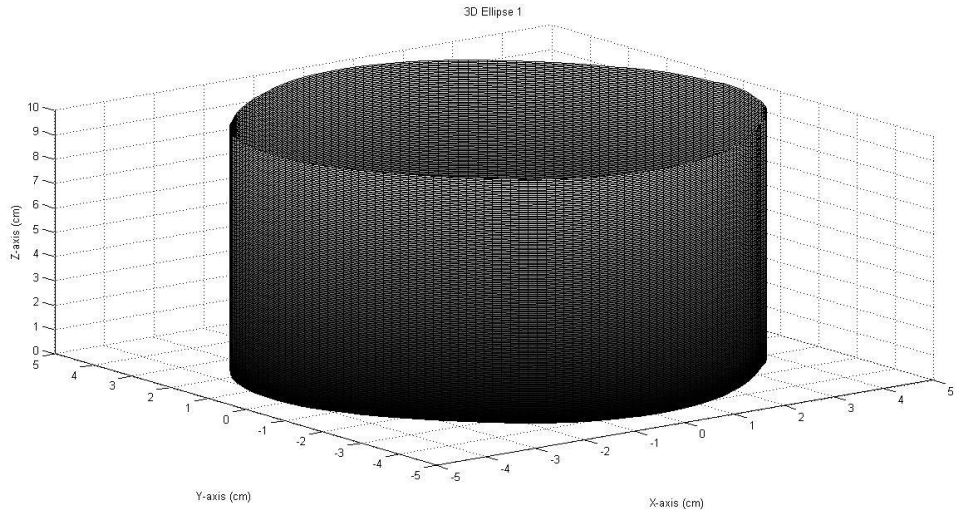


FIGURE 38: 3D Reconstructed Rectangle using Median Filter

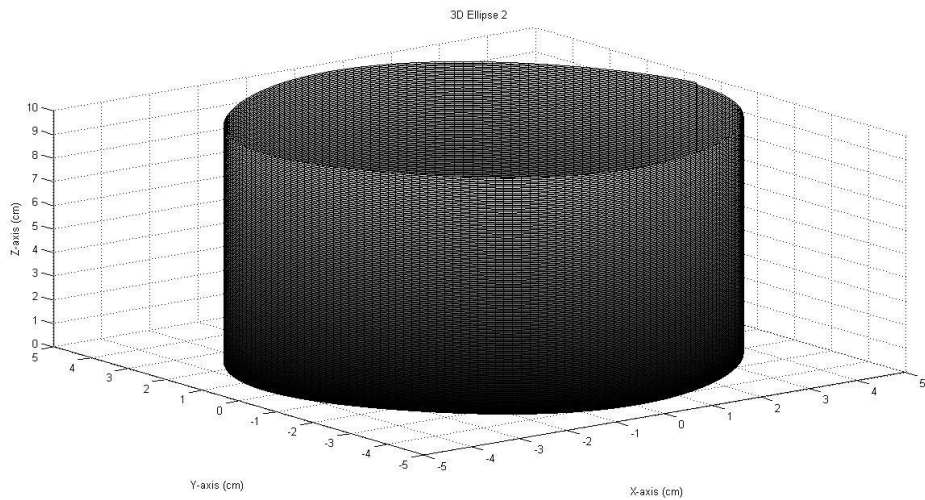


FIGURE 39: 3D Reconstructed Rectangle using Butterworth Filter

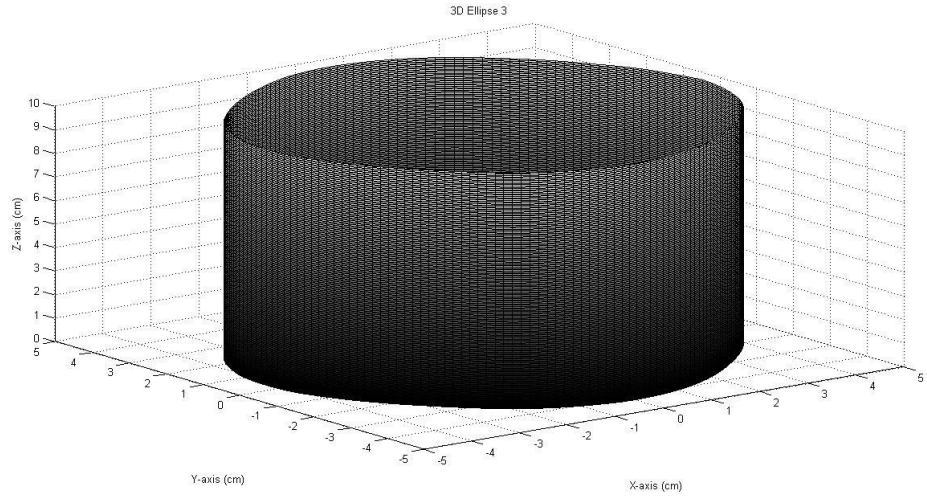


FIGURE 40: 3D Reconstructed Rectangle using Mean Filter

4.5 DIMENSIONS CALCULATION

The dimensions for the reconstructed surface are measured using a measurement tool in the Matlab toolbox called *imdstline*. The toolbox pops a cursor that can be dragged on the 2D image to measure distance between two points. For each of the measured dimensions it will be compared with the reference object dimensions and the accuracy are calculated.

4.5.1 Cylinder

Based on Table 2 and 3, the highest accuracy for the cylinder shape dimension measurement are for the Median filtered data. The accuracy for all three of the shape have a ± 0.80 gap and the accuracy are approximately 74%.

TABLE 2: Dimension Value for Cylinder Diameter 1

Dimension	Cylinder	1 st	2nd	3rd	4 th	5th
Diameter	Median	8.65	8.62	8.59	8.99	8.96
	Butterworth	8.62	8.72	8.69	9.01	9.03
	Mean	8.67	8.60	8.63	9.01	8.99

TABLE 3: Dimension Value for Cylinder Diameter 2

Dimension	Cylinder	Average	Reference Object	% Accuracy	% Error
Diameter	Median	8.762	7	74.83	25.17
	Butterworth	8.814	7	74.09	25.91
	Mean	8.798	7	74.31	25.69

4.5.2 Ellipse

The dimension for an ellipse are divided into two which are for the major axis and minor axis. For the major axis dimension, the highest accuracy for dimension measurement are for the Mean filtered data as seen in Table 5. The accuracy for all three of the shape have a ± 1.40 gap and the accuracy are approximately 59%.

TABLE 4: Dimension Value for Ellipse Major Axis 1

Dimension	Ellipse	1 st	2nd	3rd	4 th	5th
Major Axis	Median	9.80	10.47	8.84	10.42	10.12
	Butterworth	9.82	10.31	8.69	10.40	9.99
	Mean	9.79	10.43	8.68	10.32	9.95

TABLE 5: Dimension Value for Ellipse Major Axis 2

Dimension	Ellipse	Average	Reference Object	% Accuracy	% Error
Major Axis	Median	9.93	7	58.15	41.85
	Butterworth	9.84	7	59.43	40.57
	Mean	9.83	7	59.58	40.42

For the minor axis dimension, the highest accuracy for dimension measurement are for the Mean filtered data similar as the major axis as seen in Table 7. The accuracy for all three of the shape have a ± 0.40 gap and the accuracy are approximately 33%.

TABLE 6: Dimension Value for Ellipse Minor Axis 1

Dimension	Cylinder	1 st	2nd	3rd	4 th	5th
Minor Axis	Median	8.78	8.73	6.83	8.93	8.53
	Butterworth	8.66	8.99	6.83	8.87	8.46
	Mean	8.68	8.72	6.78	8.80	8.73

TABLE 7: Dimension Value for Ellipse Minor Axis 2

Dimension	Cylinder	Average	Reference Object	% Accuracy	% Error
Minor Axis	Median	8.36	5	32.8	67.2
	Butterworth	8.36	5	32.8	67.2
	Mean	8.34	5	33.2	66.8

4.5.3 Hemisphere

The dimension for a hemisphere are divided into two which are the radius and diameter. For the radius dimension, the highest accuracy for dimension measurement are for the Median filtered data as seen in Table 9. The accuracy for all three of the shape have a ± 1.80 gap and the accuracy are approximately 62%.

TABLE 8: Dimension Value for Hemisphere Radius 1

Dimension	Hemisphere	1 st	2nd	3rd	4th	5th
Radius	Median	9.51	9.59	9.49	9.53	10.14
	Butterworth	9.60	9.92	9.61	9.51	10.18
	Mean	9.55	9.54	9.51	9.50	10.19

TABLE 9: Dimension Value for Hemisphere Radius 2

Dimension	Hemisphere	Average	Reference Object	% Accuracy	% Error
Radius	Median	9.65	7	62.14	37.86
	Butterworth	9.76	7	60.57	39.43
	Mean	9.66	7	62.00	38.00

For the diameter dimension, the highest accuracy for dimension measurement are for the Median filtered data similar as the radius as seen in Table 11. The accuracy for all three of the shape have a ± 2.50 gap and the accuracy are approximately 60%.

TABLE 10: Dimension Value for Hemisphere Diameter 1

Dimension	Hemisphere	1 st	2nd	3rd	4 th	5th
Diameter	Median	9.68	9.67	9.64	9.67	10.25
	Butterworth	9.80	9.63	9.98	9.94	10.47
	Mean	9.69	9.66	9.67	9.59	10.48

TABLE 11: Dimension Value for Hemisphere Diameter 2

Dimension	Hemisphere	Average	Reference Object	% Accuracy	% Error
Diameter	Median	9.78	7	60.29	39.71
	Butterworth	9.96	7	57.71	42.29
	Mean	9.82	7	59.72	40.26

4.5.4 Rectangle

The dimension for a rectangle are divided into two which are the width and length. For the radius dimension, the highest accuracy for dimension measurement are for the Mean filtered data as seen in Table 13. The accuracy for all three of the shape have a ± 1.30 gap and the accuracy are approximately 74%.

TABLE 12: Dimension Value for Rectangle Width 1

Dimension	Rectangle	1 st	2nd	3rd	4th	5th
Width	Median	8.81	9.48	8.17	8.35	9.29
	Butterworth	8.87	9.42	8.23	8.42	9.29
	Mean	8.85	9.37	8.21	8.31	9.33

TABLE 13: Dimension Value for Rectangle Width 2

Dimension	Rectangle	Average	Reference Object	% Accuracy	% Error
Width	Median	8.82	7	74.00	26.00
	Butterworth	8.85	7	73.57	26.43
	Mean	8.81	7	74.14	25.86

For the diameter dimension, the highest accuracy for dimension measurement are for the Mean filtered data similar as the radius as seen in Table 15. The accuracy for all three of the shape have a ± 0.40 gap and the accuracy are approximately 74%.

TABLE 14: Dimension Value for Rectangle Length 1

Dimension	Rectangle	1 st	2nd	3rd	4 th	5th
Length	Median	8.81	9.44	8.18	8.26	9.30
	Butterworth	8.84	9.40	8.22	8.33	9.30
	Mean	8.85	9.32	8.20	8.26	9.32

TABLE 15: Dimension Value for Rectangle Length 2

Dimension	Rectangle	Average	Reference Object	% Accuracy	% Error
Length	Median	8.80	7	74.29	25.71
	Butterworth	8.82	7	74.00	26.00
	Mean	8.79	7	74.43	25.57

In conclusion, based on the result that were tabulated for Table 2 until Table 15, median and mean filter has the highest accuracy for all the shapes. Therefore, both of this filter is recommended to be used in filtering 1D signal. However, the accuracy between the reconstructed surface and reference object dimension are mediocre and varies depending on different shapes. The cylinder and rectangle shape has accuracy

above 70% while the ellipse and hemisphere are the opposite. This are due to the noise that causes inaccuracies and alterations along the borders of the objects that were scanned. Even after the data is filtered the reconstructed surface still has a low accuracy. Besides that, the reconstructed surface of shapes that have edges are also not perfectly reconstructed. The edges become more softer and smoother due to the filter.

CHAPTER 5

CONCLUSION & RECOMMENDATIONS

In conclusion, the objectives of this experiment are successfully achieved. A 3D reconstructed surface was created by using a low cost and fast response infrared sensor. The noise inside the signal are also successfully filtered using three different filters. Besides that, the sensor value is converted to distance value using an equation derived from the linear graph of voltage output versus the inverse number of distance. Not only that but, the sensor value which are in 1D are also successfully converted into 2D and 3D using an algorithm that were created which are the polar to cartesian conversion for 1D to 2D conversion and mesh algorithm for 2D to 3D conversion. Lastly, the accuracy of all the 3D reconstructed surface has an average of 60%. This are due to the noise that causes alterations along the borders of the objects that were scanned even after the signal are filtered.

For future recommendations, different type of filter need to be tested and used other than Median, Butterworth and Mean filter to view which one is the best to remove the noise in the infrared signal. Besides that, other types of algorithm for conversion between 1D to 2D also need to be tested as this project only focuses on one algorithm. This is because other algorithm may have a higher dimension accuracy compared to polar to cartesian conversion method used in this project.

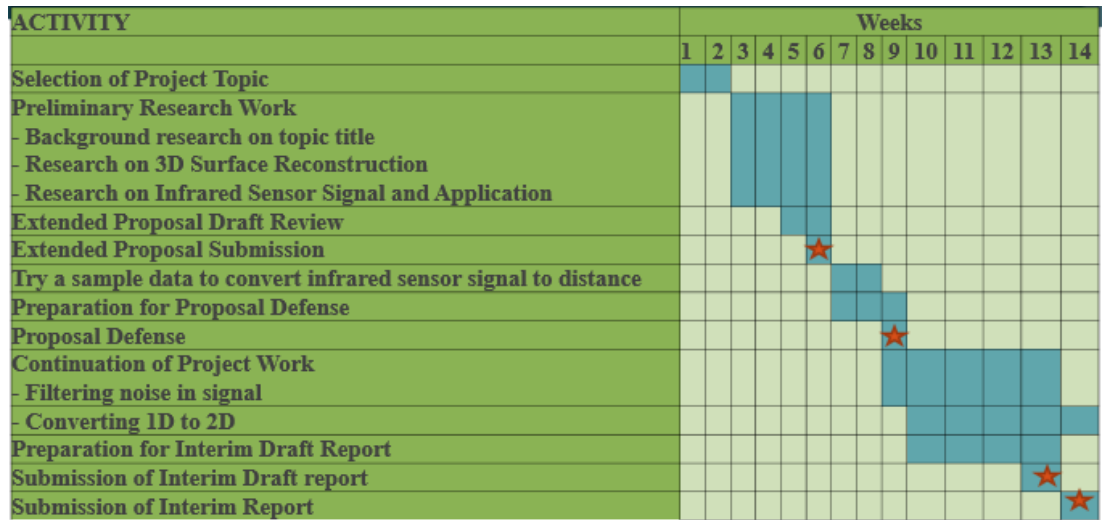
REFERENCES

1. S. Nurmaini, "Intelligent Low Cost Mobile Robot.", *International Journal of Computer Applications (0975 – 8887)*, 2011, vol. 35, (12), pp. 1-7.
2. P. Hyungwoong, B. Sungjin, and L. Sooyoung. "IR Sensor Array for a Mobile Robot.", in *International Conference on Advanced Intelligent Mechatronics. Monterey, California, USA, 24th – 28th July 2005*, pp. 928-933.
3. G. Benet*, F. Blanes, J.E. Simo and P. Perez, "Using Infrared Sensors for Distance Measurement in Mobile Robots.", *Robotics and Autonomous Systems*, 2002, vol. 40, pp. 255-266.
4. S. Milani, E. Frigerio, M. Marcon and S. Tubaro, "Denoising infrared structured light DIBR signals using 3D morphological operators.", in *3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, Zurich, 2012, pp. 1-4.
5. N. Sakthivel and L. Prabhu, "Mean – Median Filtering For Impulsive Noise Removal.", in *International Journal of Basics and Applied Sciences*, 2014, vol. 2, (4), pp. 47-57.
6. S. Hussmann, T. Holtorf and F. Knoll, "Investigation of different polar to Cartesian coordinate system conversion methods for ToF-cameras in the close-up range," in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, Pisa, 2015, pp. 1072-1077.
7. J. Alakuijala, J. Oikarinen, Y. Louhisalmi, X. Ying and J. Koivukangas, "Image transformation from polar to Cartesian coordinates simplifies the segmentation of brain images," in *1992 14th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Paris, France, 1992, pp. 1918-1919.
8. V. Borji and M.G. Voskoglou, "Applying the APOS Theory to Study the Student Understanding of Polar Coordinates", in *American Journal of Educational Research*, 2016, vol. 4, (16), pp. 1149-1156.
9. T. Luginbuhl and C. Hempel, "Converting Bearings-Only Measurements to Cartesian Coordinates," in *IEEE Transactions on Aerospace and Electronic Systems*, 2009, vol. 45, no. 1, pp. 393-404.

10. W. Chanapai, P. Fuangfa, S. Jaovisidha, A. Narttharung and P. Ritthipravat, "3D reconstruction from multiple imaging planes: A pilot study of bone tumor MR images.", in *2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Abu Dhabi, 2015, pp. 354-359.
11. N. H. Mahmood and T. Tjahjadi. "3D Reconstruction for Prosthetic Design.", in *Computer Engineering and Applications (ICCEA). Second International Conference*, Bali Island, 2010, pp. 431-435.
12. H. Jaewook, K. Soon, J. Jaekyo and K. Hyeonwoo, "3D Surface Reconstruction Method Using a Coded Binary Pattern.", in *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 2013, vol. 6, (6), pp. 59-68.
13. T. Moons, L. Van Gool and M. Vergauwen, "Reconstruction from Multiple Images.", *Foundations and Trends in Computer Graphics and Vision*, 2008, vol. 4, (4), pp. 291-299.
14. SHARP, "Device Specification for Analog Output Type Distance Measuring Sensor MODEL No. GP2D120XJ00F", 2005, pp. 1-10.
15. S.A. Daud, N. H. Mahmood, L. P. Ling, F. K. C. Harun, "The Used of Infrared Sensor for 3D Image Reconstruction" in *Jurnal Teknologi*, 2015, vol. 73, (3), pp. 127-132.
16. A. Torralba and A. Oliva, "Depth estimation from image structure," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002, vol. 24, (9), pp. 1226-1238.

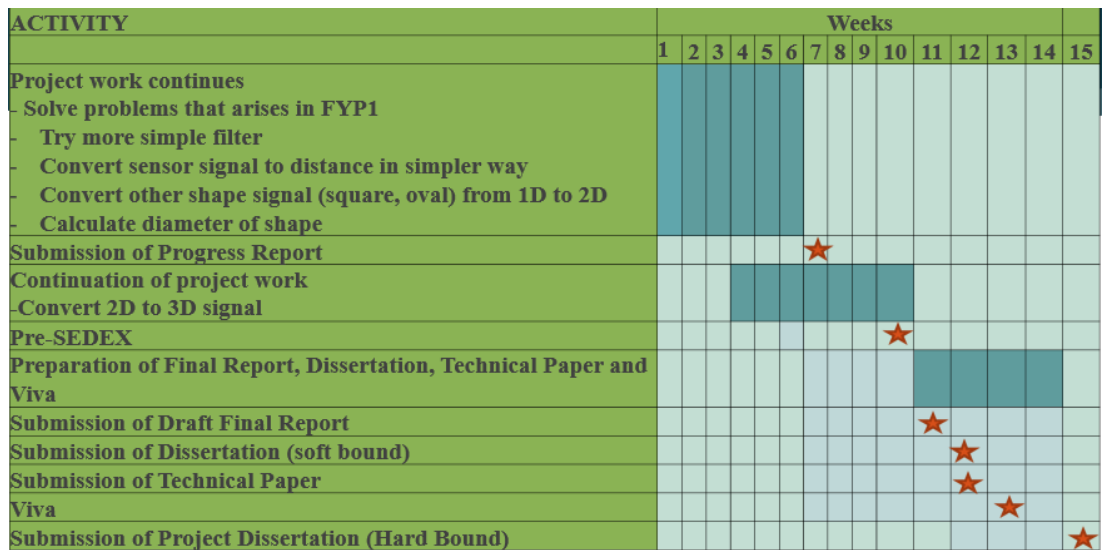
APPENDICES

1. Final Year Project 1 Gantt Chart



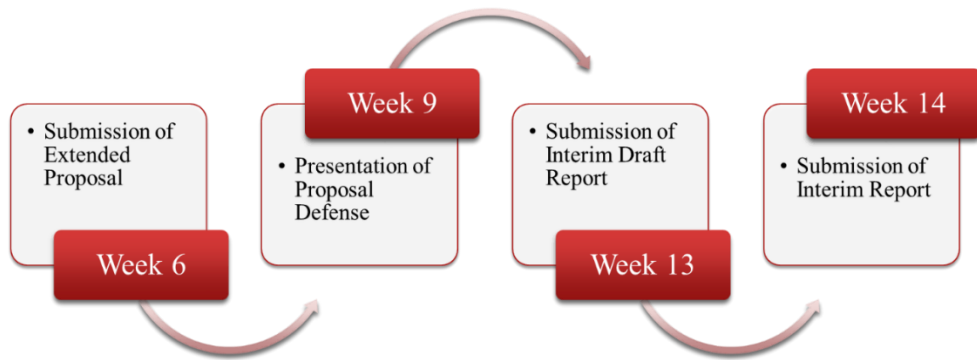
- ★ Project Milestone
- █ Process

2. Final Year Project 2 Gantt Chart

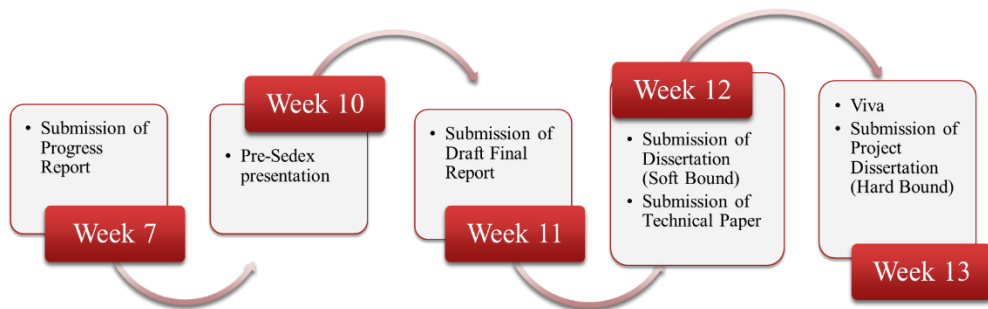


- ★ Project Milestone
- █ Process

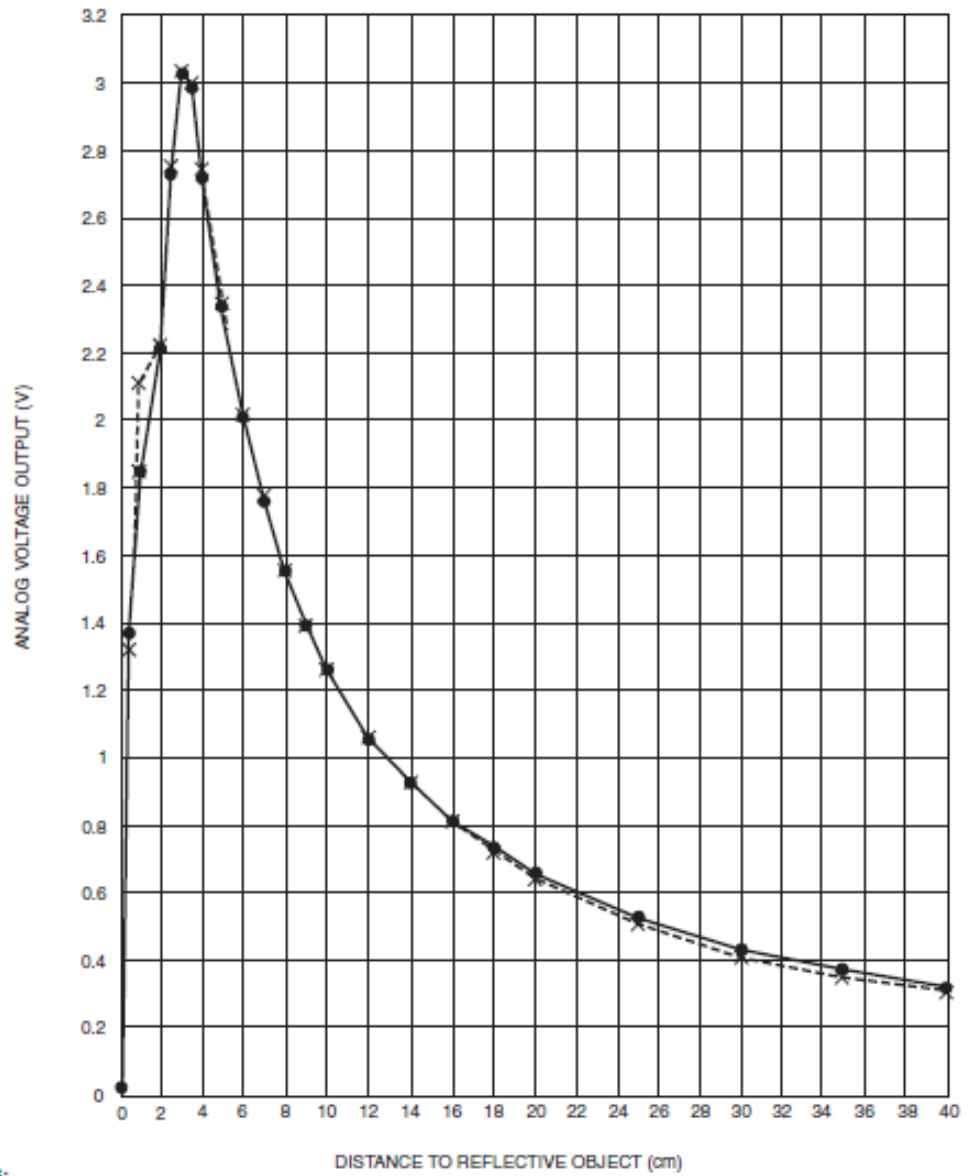
3. Final Year Project 1 Project Milestone



4. Final Year Project 2 Project Milestone

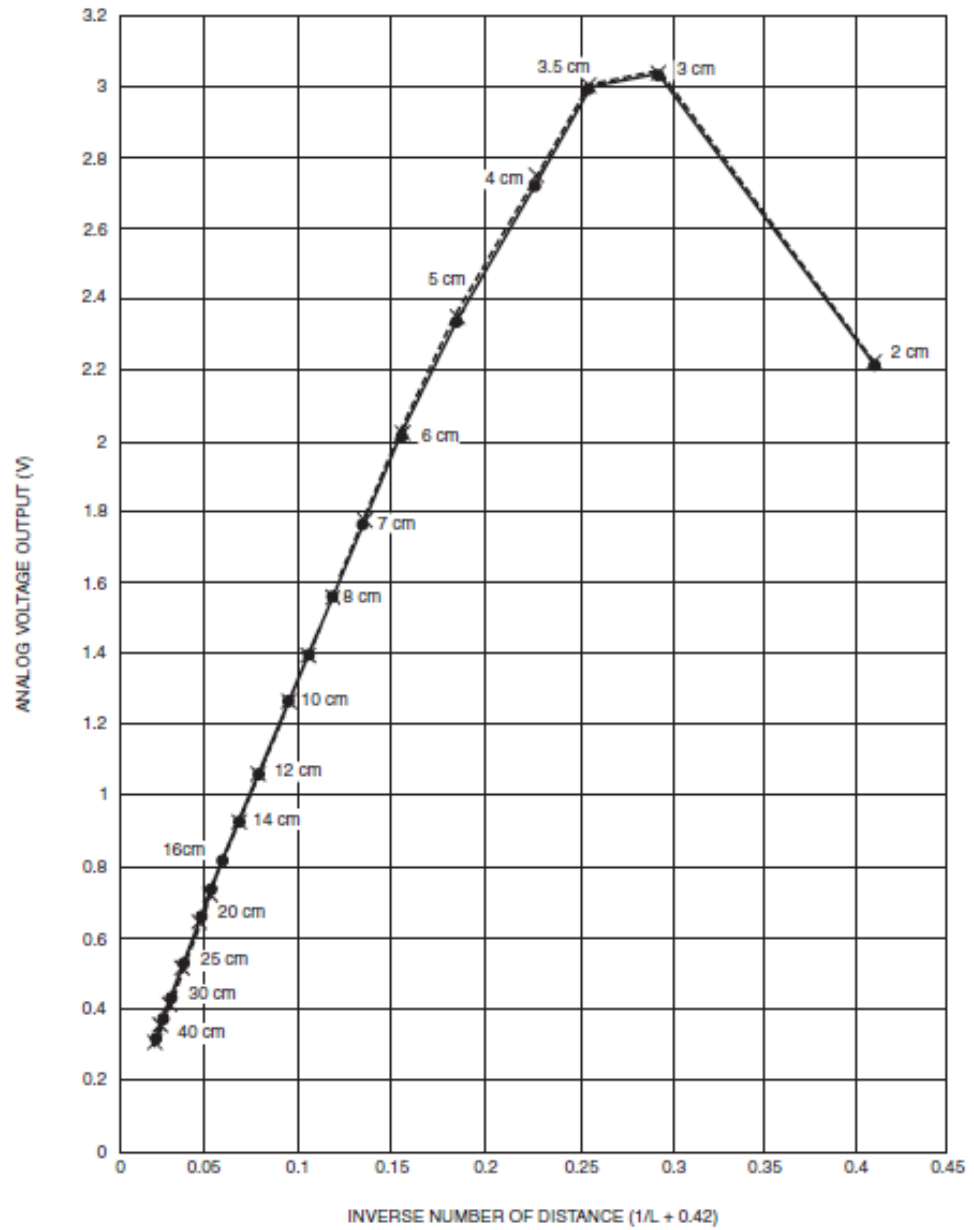


5. Voltage vs Distance Graph on GP2D120 Datasheet



FR-

6. Voltage vs 1/Distance Graph on GP2D120 Datasheet



7. 3D Reconstructed Algorithm Matlab Code

```
1. % Full code on converting 1D vector data to 3D
2. % Add a 181 data that contains the same value as the 1st data
3. Ellipse3(181) = Ellipse3(1)
4.
5. % Convert Sensor to Distance
6. Dist1 = (2914./(Ellipse3 + 5))-1
7.
8. % Filter Median
9. D = 9 - Dist1;
10. D0 = medfilt1(D,8);
11. D0(1) = D0(end);
12. figure; plot (D, 'r')
13. hold on;
14. plot (D0);
15. title('1D Filter 1')
16. xlabel('Data') % x-axis label
17. ylabel('Distance (cm)') % y-axis label
18.
19. % Filter Butterworth
20. fs = 100;
21. T = 2;
22. t = 0:1/fs:T;
23. f = 2;
24. [B,A]=butter(6, 5/fs,'low');
25. D1=filtfilt(B, A, D);
26. D1(1) = D1(end);
27. figure; plot (D, 'r')
28. hold on;
29. plot (D1);
30. title('1D Filter 2')
31. xlabel('Data') % x-axis label
32. ylabel('Distance (cm)') % y-axis label
33.
34. % Filter Mean
35. windowSize = 5;
36. b = (1/windowSize)*ones(1,windowSize);
37. a = 1;
38. D2 = filtfilt(b,a,D);
39. D2(1) = D2(end)
40. figure; plot (D, 'r')
41. hold on;
42. plot (D2);
43. title('1D Filter 3')
44. xlabel('Data') % x-axis label
45. ylabel('Distance (cm)') % y-axis label
46.
47. %Conversion Angle to radian
48. theta = deg2rad(Angle);
49.
50. %Conversion 1D to 2D (polar to cartesian)
51. %Data1
52. [x1,y1] = pol2cart(theta, D0);
53. figure();
54. plot (x1,y1);
```

```

55. h = imdistline(gca);
56. h = imdistline(gca);
57. title('2D Ellipse 1')
58. xlabel('X-axis (cm)') % x-axis label
59. ylabel('Y-axis (cm)') % y-axis label
60.
61. %Data2
62. [x2,y2] = pol2cart(theta, D1);
63. figure();
64. plot (x2,y2);
65. h = imdistline(gca);
66. h = imdistline(gca);
67. title('2D Ellipse 2')
68. xlabel('X-axis (cm)') % x-axis label
69. ylabel('Y-axis (cm)') % y-axis label
70.
71. %Data3
72. [x3,y3] = pol2cart(theta, D2);
73. figure();
74. plot (x3,y3);
75. h = imdistline(gca);
76. h = imdistline(gca);
77. title('2D Ellipse 3')
78. xlabel('X-axis (cm)') % x-axis label
79. ylabel('Y-axis (cm)') % y-axis label
80.
81. % Plot 3D
82. [xx,yy] = meshgrid (x1,y1)
83. z=meshgrid (linspace (0,10,181), linspace (0,10,181))
84. xx1 = xx.'
85. figure;
86. surf(xx1,yy,z)
87. colormap(gray)
88. title('3D Ellipse')
89. xlabel('X-axis (cm)') % x-axis label
90. ylabel('Y-axis (cm)') % y-axis label
91. zlabel('Z-axis (cm)') % x-axis label

```