*Dimensions and Density Measurement of an Object inside an Image using MATLAB*

*Rahul Pant1, Vrinda Gupta2, Rajni Bala3 Gaurav Bansal4*

Graduate Sudent1, Associate Professor & HOD2, Assistant Professor3 Admin Officer Faculty Affairs4

NIT Kurukshetra1-3, IIT Dhanbad4

rahul\_12015037@nitkkr.ac.in[1](mailto:1), vrindag16@nitkkr.ac.in[2](mailto:2), [ece13@nitkkr.ac.in3](mailto:ece13@nitkkr.ac.in3), gauravbansal@iitism.ac.in4

**Abstract—** ***Measuring object dimensions within an image can be conveniently executed using MATLAB instead of making physical measurements. This paper covers steps for isolating a desired object and measuring its diameter in centimeters and then calculating its volume and density.***

Keywords—MATLAB, Dimensions, Density, Volume, Image Processing Techniques

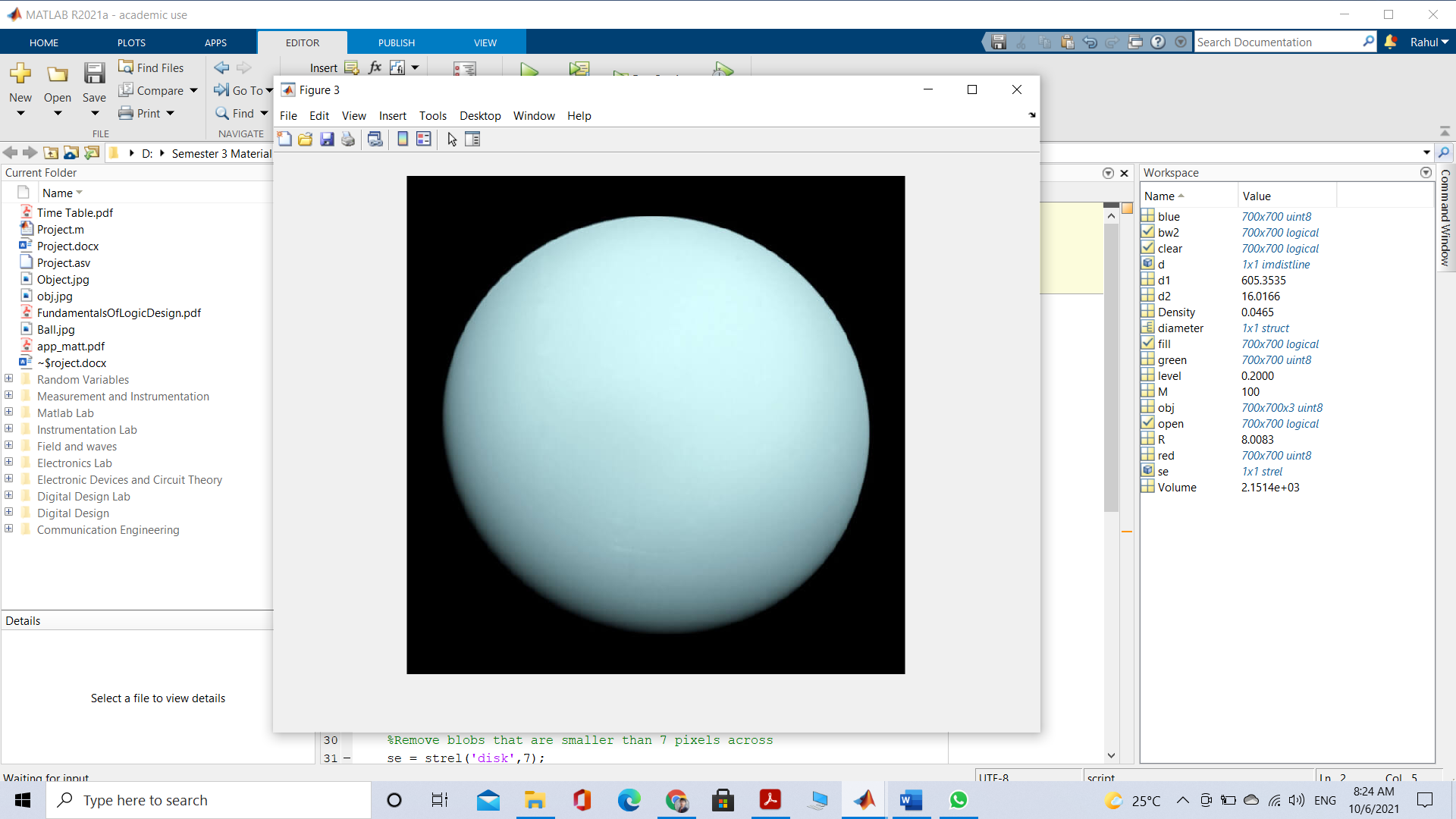
# Introduction

Matrix Laboratory MATLAB R2021a is a high-level language and interactive environment for computer computation, visualization, and programming. In MATLAB Image Processing Toolbox provides a comprehensive set of reference-standard algorithms, functions, and applications for image processing, visualization, analysis, and algorithm development. These tools provide a fast and convenient way to process and analyze images without the need of knowledge of a complex coding language. As discussed in this paper [1-4] by Matthew Wesolowski (2014) measured the diameter of object in pixels but in real life centimeter (cm) scale is more convenient to analyze. Moreover, there was no way mention to measure the volume and density. Moreover, Dots per inch (DPI), is a measure of resolution of a digital scan. The higher the dot density the higher the resolution of the print or scan. In this paper the diameter is displayed in cm and the volume and density of circular objects can be calculated by entering the resolution of image in dots per inch (dpi). Muhammad Ahmad Mahmoud Mayhoub (2020) produced a picture handling system to quantify the volume and size of orange fruits. Imdistline function was not used to physically measure the object [5-8]. In this the diameter of tomato is measured using image processing techniques using MATLAB for identifying the quality and size of tomatoes. No way to measure the volume is mentioned.

# Research Methodology

## How to open the Image

Open the MATLAB R2021a software. Download the Image Processing Tool Box. Create a new MATLAB script file. It is important that the Current Folder that you are working out of be the folder that contains both the script file and image. The command *imread* reads an image as shown in Figure 1(a) and Figure1(b) and converts it into a “3-dimensional” matrix in the RGB color space. The final dimension (RGB) corresponds to a red, green and blue intensity level. Use *imshow* to view the produced image in a new window.



**Figure 1(a): Original Image Figure 1(b): Original Image**

**Code:**

%% Import Image

clc

close all

obj = imread('obj.jpg');

imshow(obj)

## Segment Image

The given image is segmented into a binary image to differentiate the background from the desired object. Firstly, the image is divided into three images based on the intensities of each red, green and blue component within the image. This is Color Based Image Segmentation. The blue plane is the best choice to use for Image Thresholding because it provides the most contrast between the desired object and the background. Image thresholding takes an intensity image and converts it into a binary image based on the level desired. A value between 0 and 1 determines which pixels will be set to a 1 (white) or 0 (black). Figure 2(a) and 2(b) shows the result of the image thresholding at 0.2 and 0.25 respectively. The image Figure 3(a) and 3(b) have been segmented between the object we desire to measure and the background.

**Code:**

%% Segment Image

% Divide Image "obj" into its respective RGB intensities

red = obj (:, : ,1);

green = obj (: ,: ,2);

blue = obj (: ,: ,3);

Figure (1)

subplot (2,2,1); imshow(obj); title ('Original Image');

subplot (2,2,2); imshow(red); title ('Red Plane');

subplot (2,2,3); imshow(green); title ('Green Plane');

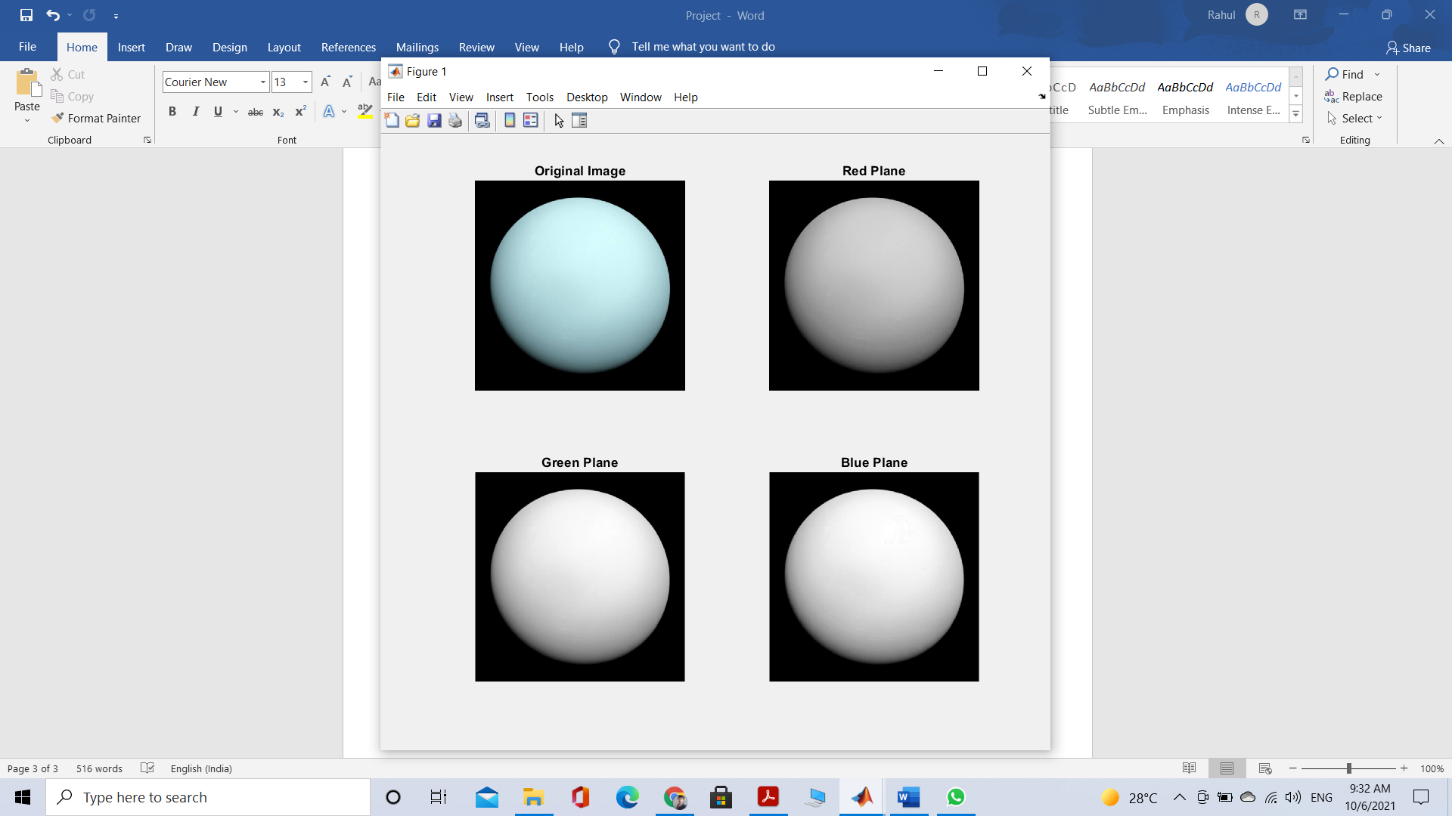
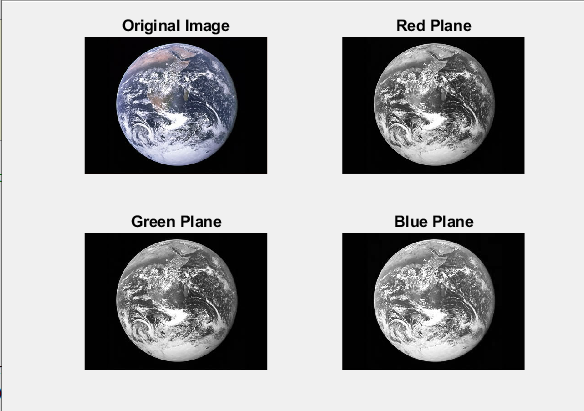
subplot (2,2,4); imshow(blue); title ('Blue Plane');

%Threshold the blue plane

Figure (2)

level = 0.2;

bw = imbinarize (blue, level);

imshow(bw); title ('Blue plane thresholded');

**Figure 2(a): Color Thresholding Figure 2(b): Color Thresholding**

## Removing Noise

The top-left image in Figure 3(a) and 3(b) have a bit of “noise” and the image need to be cleaned up significantly to improve the accuracy of our diameter measurement. Steps are taken to clean up the image and provide a more uniform blob to analyze. Blobs in this document are any collection of white pixels that touch to create a cohesive and distinct object.

**Code:**

%% Remove Noise

% Fill any holes

fill = imfill (bw, ‘holes');

subplot (2,2,2); imshow(fill); title ('Holes filled');

% Remove any blobs on the border of the image

clear = imclearborder (fill);

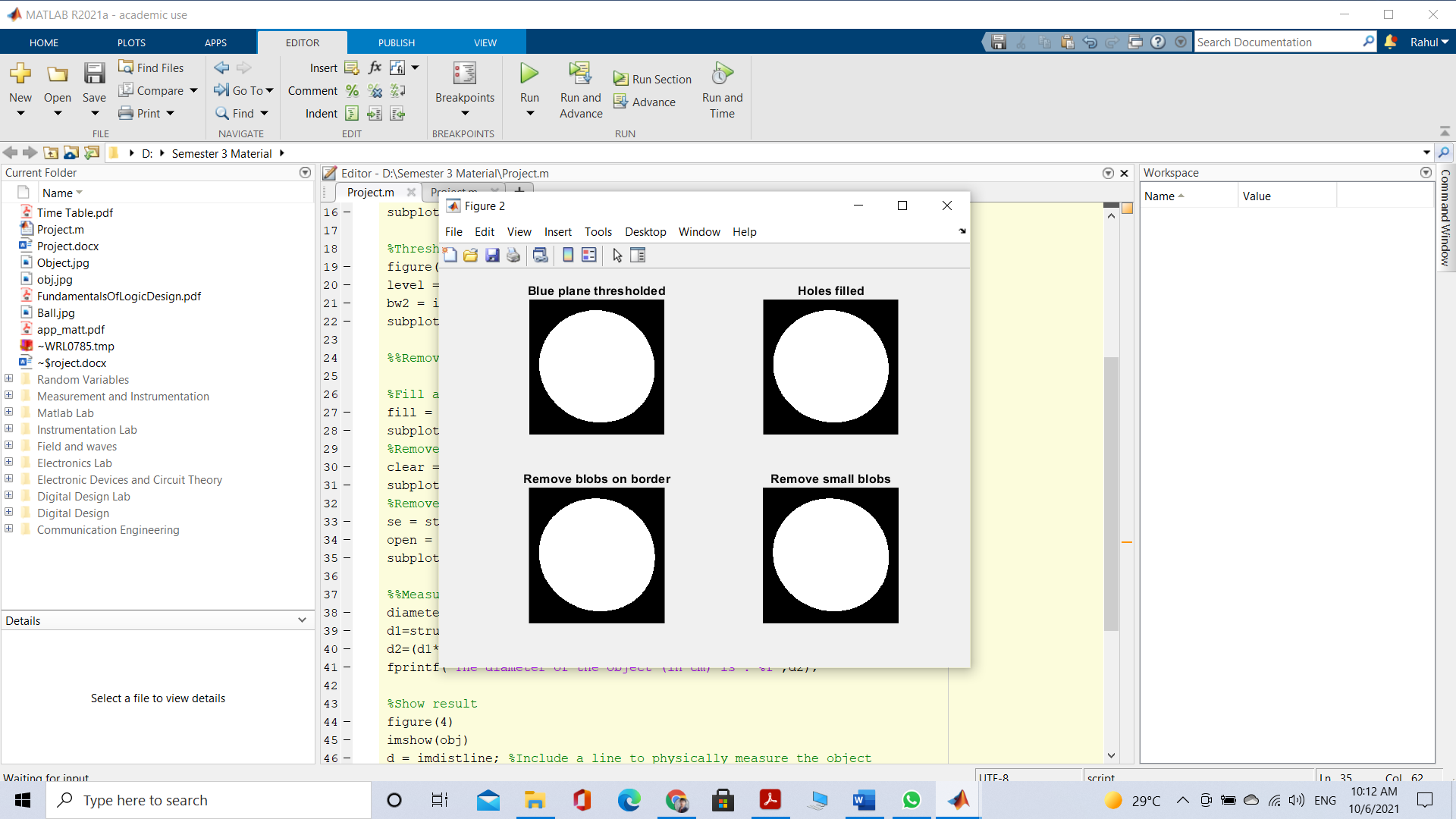
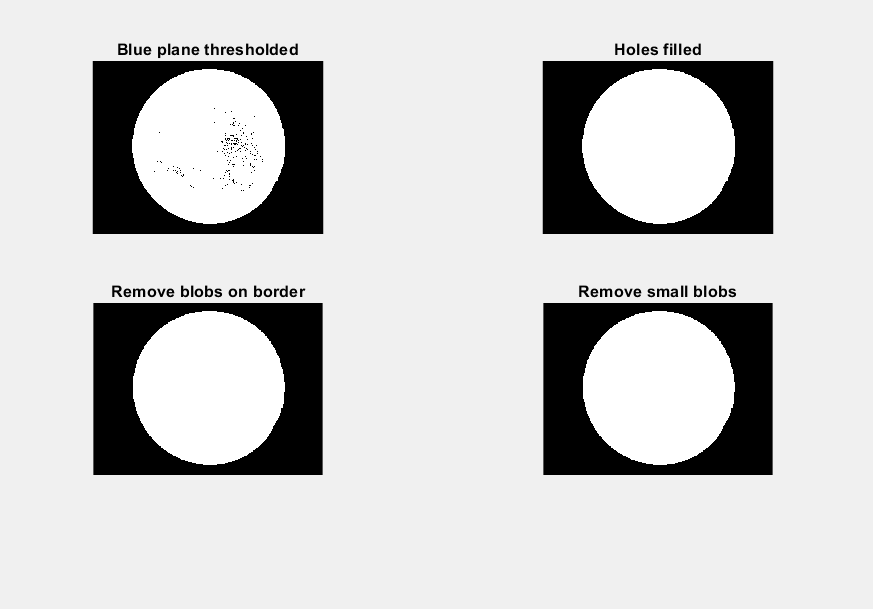
subplot (2,2,3); imshow(clear); title ('Remove blobs on border');

% Remove blobs that are smaller than 7 pixels across

se = strel ('disk',7);

open = imopen (fill, se);

subplot (2,2,4); imshow(open); title ('Remove small blobs');

****

**Figure 3(a): Complete Segmentation and Cleanup Image Figure 3(a): Complete Segmentation and Cleanup Image**

## Measuring Image

In Figure 3(a) and 3(b) the image at bottom right corner is the result of all image segmentation and clean up procedures to provide one distinct and cohesive blob, which represents the ball in the original image. Having the original image in a binary form will make it easy for functions built into MATLAB to quickly analyze the image. The regionprops function tool will provide the MajorAxisLength of the blob in the image.

**Code:**

%%Measure Object Diameter

diameter = regionprops (open, ‘MajorAxisLength')

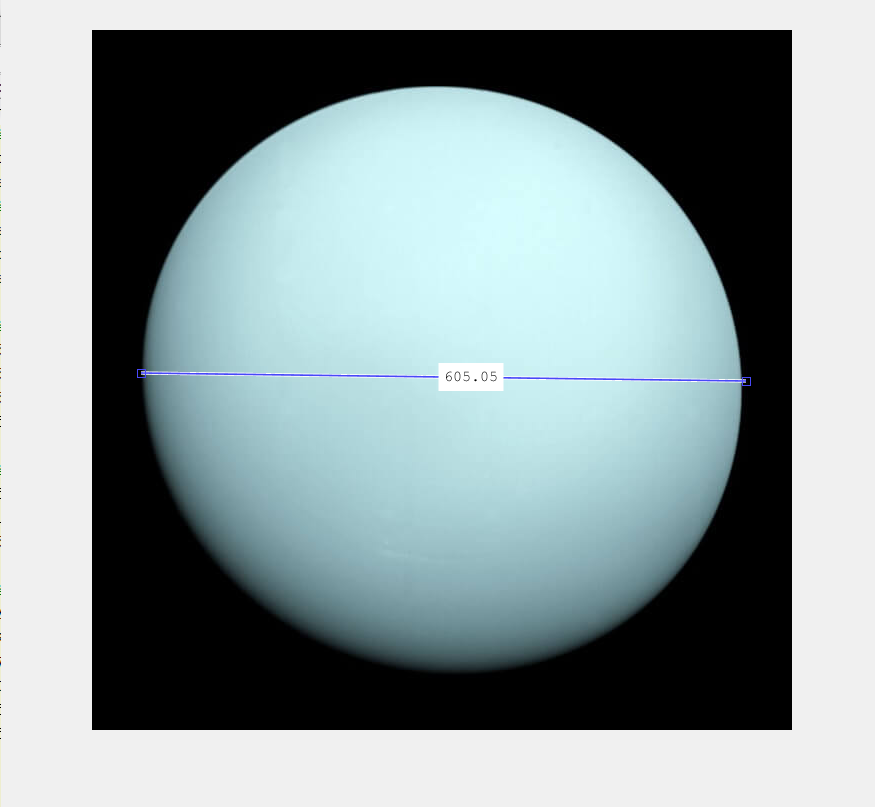
d1=struct2array(diameter);

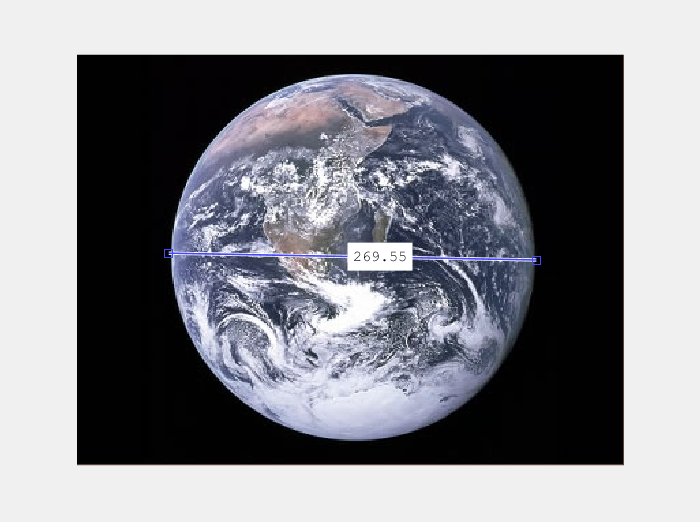
res = input ('\n Enter the resolution of image in dpi: ');

d2=(d1\*2.54)/res;

fprintf ('The diameter of the object (in cm) is: %f', d2);

d=imdistline;



****

**Figure 4(a): Manually Measuring Diameter Figure 4(b): Manually Measuring Diameter**

## Calculating Diameter and Volume

**Code:**

%Calculating density of object

M = input ('\n Enter the mass of the object (in g):');

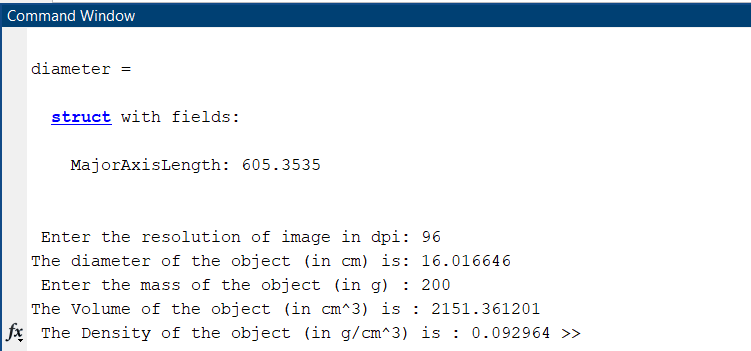
R=(d1\*2.54)/ (2\*96);

Volume= ((4/3) \*pi\*(R^3));

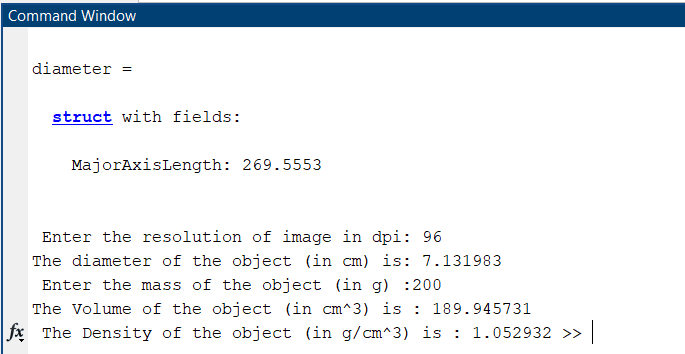
Density=M/Volume;

fprintf ('The Volume of the object (in cm^3) is: %f ', Volume);

fprintf ('\n The Density of the object (in g/cm^3) is: %f ', Density);



**Figure 5(a): Calculated Diameter and Volume**

****

**Figure 5(b): Calculated Diameter and Volume**

# Results and Discusion

The measured diameter of the objects is displayed in the Command Window as 605 and 269 pixels across respectively which in centimeters scale is equal to 16.01 and 7.13. This is verified in Figure 4(a) and 4(b) by using the imdistline function. The value calculated using code is close to the manual measurement in Figure 4(a) and 4(b).

As compared to other papers this program could be used for images of different resolutions. The volume and density of spherical objects can be easily measured. Imdistline function is also used to confirm the measurements physically and could also be used to measure other dimensions if needed.

It can be used to measure very small objects precisely which are difficult to measure physically like it can be used to measure medicine (circular pills) and can also be used to measure football, basketball, cricket balls etc.

# Conclusions

This project is designed to measure the diameter and hence calculate the volume and density of spherical objects. This project is advantageous than others as in this the diameter is converted into cm from pixels scale and hence data could be easily interpreted. The volume of object is also displayed and hence density can be calculated by giving mass of object as input.

##### Acknowledgment

This work is supported by Department of Electronics and Communication Engineering of National Institute of Technology, Kurukshetra, Haryana, by providing software facilities such as, MATLAB and other computational material.

##### References

1. Image Processing Made Easy. Perf. Andy Thé. Mathworks. N.p., 15 Oct. 2014. Web. 07 Nov. 2014
2. C. M. Sabliov, D. Boldor, K. M. Keener & B. E. Farkas (2002) , International Journal of Food Properties, 5:3, 641-653, DOI:10.1081/JFP-120015498.

1. Fonseca PC, Scherer GW. An image analysis procedure to quantify the air void system of mortar and concrete. Materials and Structures. 2015 Oct;48(10):3087-98.
2. Beggan C, Hamilton CW. New image processing software for analyzing object size-frequency distributions, geometry, orientation, and spatial distribution. Computers & Geosciences. 2010 Apr 1;36(4):539-49.
3. Bala R, Marwaha A, Marwaha S. Mathematical formulation of surface conductivity for graphene material. Journal of Engineering Science and Technology. 2017 Jun 1;12(6):1677-84.
4. Garrioch R, Langlo C, Dubis AM, Cooper RF, Dubra A, Carroll J. The repeatability of in vivo parafoveal cone density and spacing measurements. Optometry and Vision Science. 2012 May;89(5):632.
5. Bansal G, Bala R. Iris Pattern Detection Using Vision Assistant & Labview 2009. InInternational Conference on Computing 2012 Jun 27 (p. 76).
6. Rosell JR, Llorens J, Sanz R, Arno J, Ribes-Dasi M, Masip J, Escolà A, Camp F, Solanelles F, Gràcia F, Gil E. Obtaining the three-dimensional structure of tree orchards from remote 2D terrestrial LIDAR scanning. Agricultural and Forest Meteorology. 2009 Sep 1;149(9):1505-15.