Progress Report

Supervised by Dr. J L Raheja

BY

ADITYA RAMACHANDRAN 2017A3PS0339P
NIRANJAN ASHOK JAHAGIRDAR 2017B3A70454P

MAY 2020

Introduction:

The aim is to find the dimensions of objects via their single images.

We first looked at several approaches.

We have been pursuing literature review, under the guidance of Dr. J. L. Raheja. During the course of our investigation into the feasibility of various approaches presented in several papers, we have come up with a few approaches one of which has been implemented below.

Methodology:

In this method, we make use of a reference object. The reference object should satisfy the following conditions:

1. From a practical perspective the reference object should be of a size similar to that of the object whose dimensions are being determined. The larger the size difference, the larger the chances of error.
2. The size of the reference object must be known (length and breadth) in a measurable unit.
3. The reference object should be easily identifiable in the image either via a distinctive shape or colour or via its position. As mentioned later, in the project we have taken the reference object to be the leftmost object in the image.

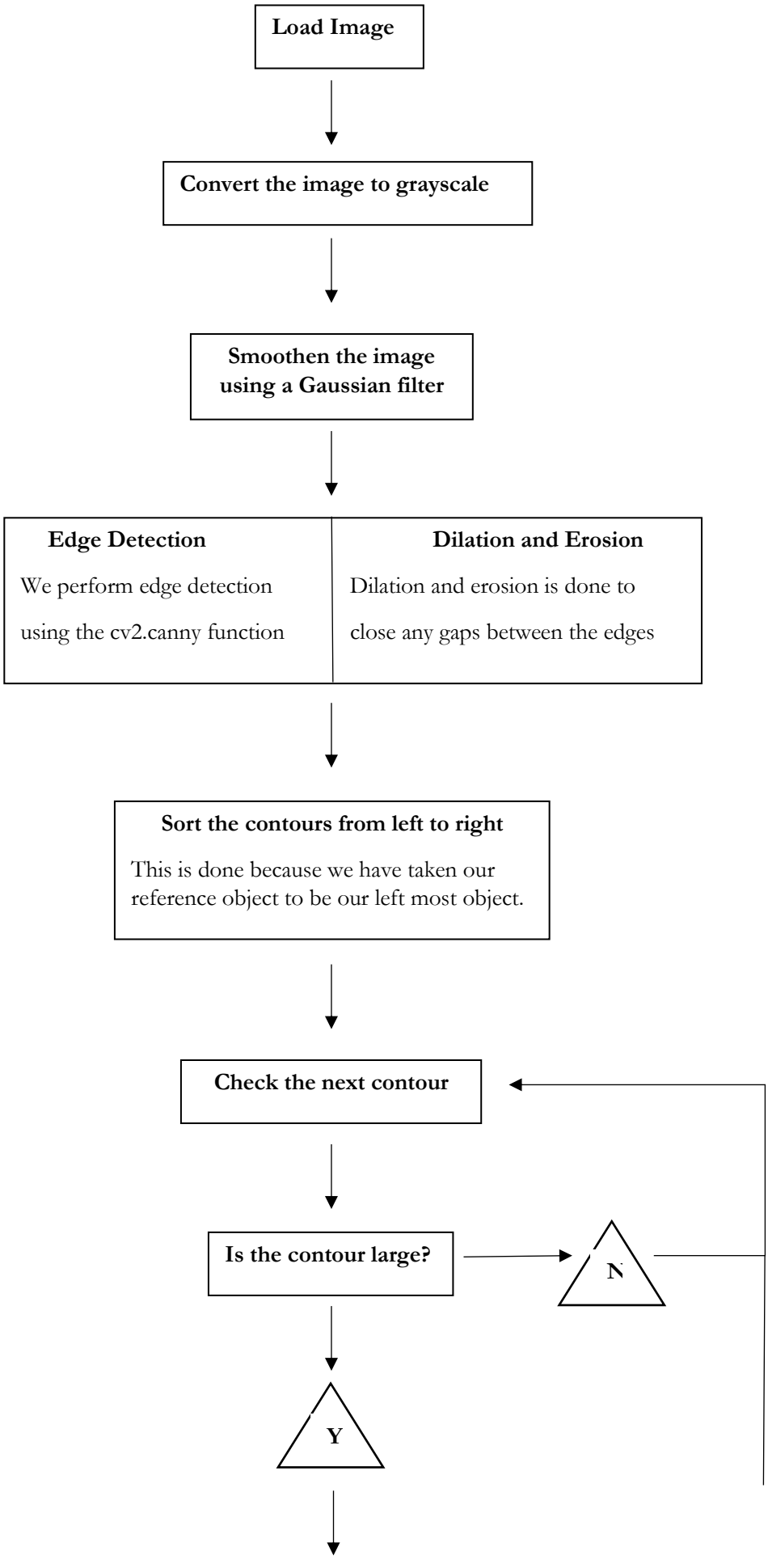
The reference object used in the testing is the Indian 2 rupee coin. We use the reference object in the image whose dimensions we know to calculate the *pixels_to_mm* ratio.

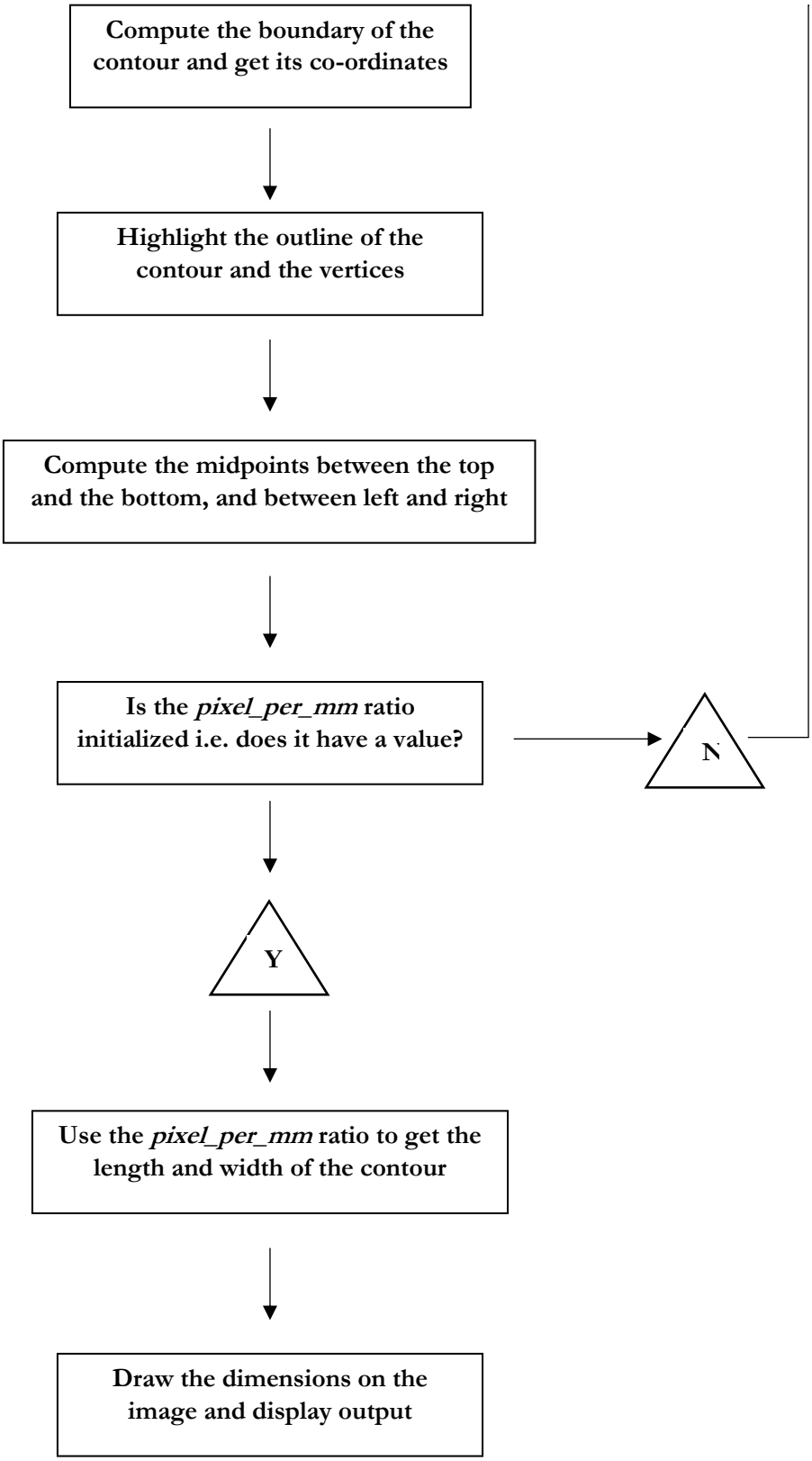
$$pixels\_to\_mm\ ratio = object\_length\ (measured\ in\ pixels) \ / \ real\_length\ (measured\ in\ mm)$$

This ratio will tell us that there are approximately x number of pixels per y mm. We can then use this ratio to compute the size of the other images after we retrieve their *object_length* from the images. From this ratio it is clear that the quality of the photo is a very important metric in this project. The higher the resolution of the image, the more accurately can the dimensions be determined.

The code is written in Python. The packages used are:

1. Numpy
2. Argparse
3. Imultils
4. Cv2


Load Image
Convert the image to grayscale
Smoothen the image using a Gaussian filter
Edge Detection
We perform edge detection using the cv2.canny function
Dilation and Erosion
Dilation and erosion is done to close any gaps between the edges
Sort the contours from left to right
This is done because we have taken our reference object to be our left most object.
Check the next contour
Is the contour large?
N
Y


Compute the boundary of the contour and get its co-ordinates
Highlight the outline of the contour and the vertices
Compute the midpoints between the top and the bottom, and between left and right
Is the *pixel_per_mm* ratio initialized i.e. does it have a value?
N
Y
Use the *pixel_per_mm* ratio to get the length and width of the contour
Draw the dimensions on the image and display output

Data Collection:

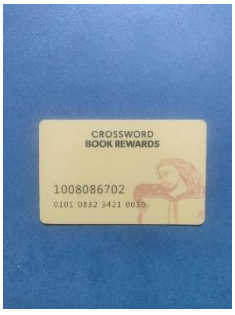
The data used consists of images of standard everyday objects. The pictures have been taken using a 16 MP camera of the phone, OnePlus 6. The reference object used is the Indian 2 rupee coin. The objects and their true dimensions are as follows:

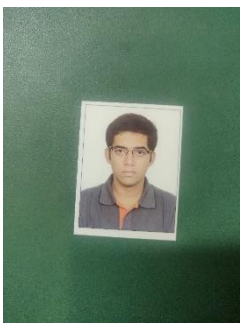
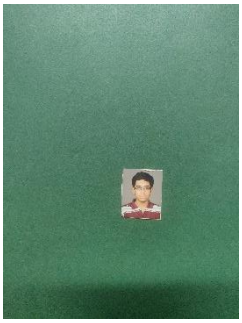
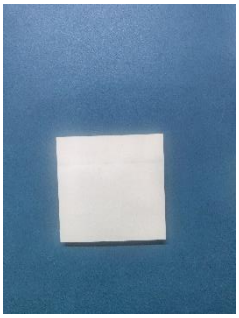
Object	Length (in mm)	Breadth (in mm)
2 Rupee Coin	26	26
Passport Picture	45	35
Stamp picture	25	20
Visiting Card	91.5	53.5
ATM card	85.6	53.9
Sticky Note	75	75
Staple	25	9.4
Playing cards	88.9	63.5

Images of the objects:



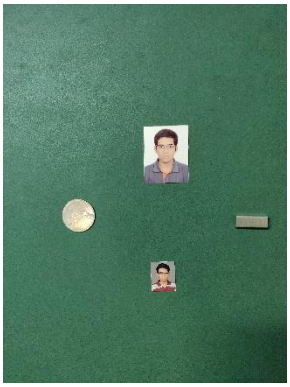




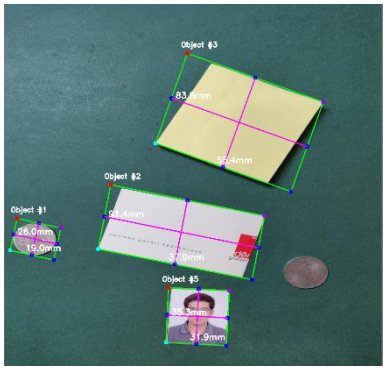




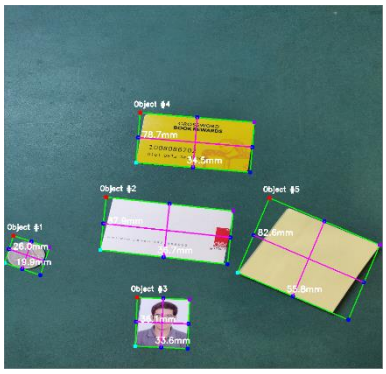
Using these objects, 100 photos have been taken. Each photo consists of three to four objects in addition to the reference object of the Indian 2 Rupee Coin. The objects have been placed on a dark background. The reference object has been placed in such a way that it is the left most object in the image or the first object from the left. This way we can sort our object contours from the left to the right, get the Indian 2 Rupee Coin and use it to define the pixels per metric ratio.

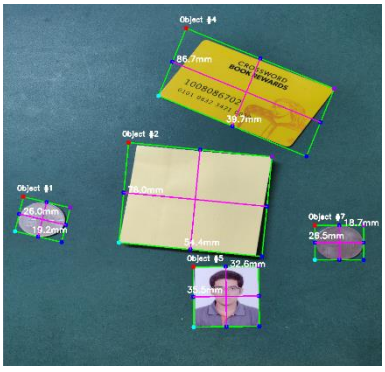
Examples of the photos:

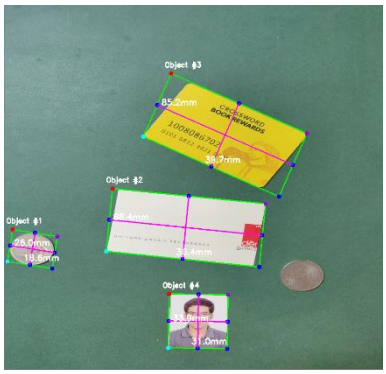


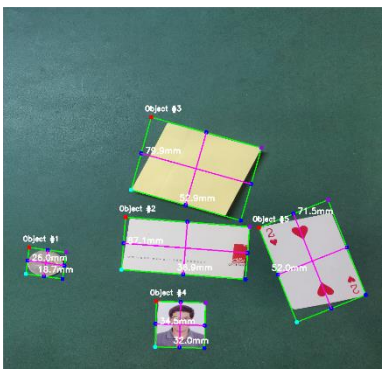
Result:











Error = (| Dimension Obtained – Actual Dimension |) / Actual Dimension

Sr. No.	Error	Sr. No.	Error
1	0	24	0.281
2	0.048	25	0.31
3	0.065	26	0.295
4	0.233	27	0.086
5	0.181	28	0.195
6	0	29	0.235
7	0.039	30	0.314
8	0.198	31	0.04
9	0.081	32	0.359
10	0.101	33	0.256
11	0	34	0.269
12	0.001	35	0.291
13	0.117	36	0.261
14	0.216	37	0.089
15	0	38	0.262
16	0.04	39	0.275
17	0.013	40	0.263
18	0.211	41	0.069
19	0.019	42	0.281
20	0	43	0.285
21	0.034	44	0.319
22	0.005	45	0.263
23	0.247	46	0.114

Average Error in Width: 0.80404904

Average Error in Height: 0.235365869

Average Error Overall: 0.157885387

Overall Accuracy: 0.842114613

Reasons for error:

1. The angle with which the pictures have been taken is not at a perfect ninety degrees or in other words is not a perfect *bird eye's view.* This is because the picture has not been taken with proper equipment ensuring angle stability. Deviation from the ninety degrees will cause distortion of the image.
2. Since a professional camera has not been used, the photos are prone to radial and tangential distortion. Radial distortion is caused by the spherical shape of the lens. Tangential distortion occurs when the lens and the image plane are not parallel. This is essentially due to the inability to calibrate the camera properly.