

**PS-7
REPORT**

Submitted By:
Akshay Antony
akshayan@andrew.cmu.edu

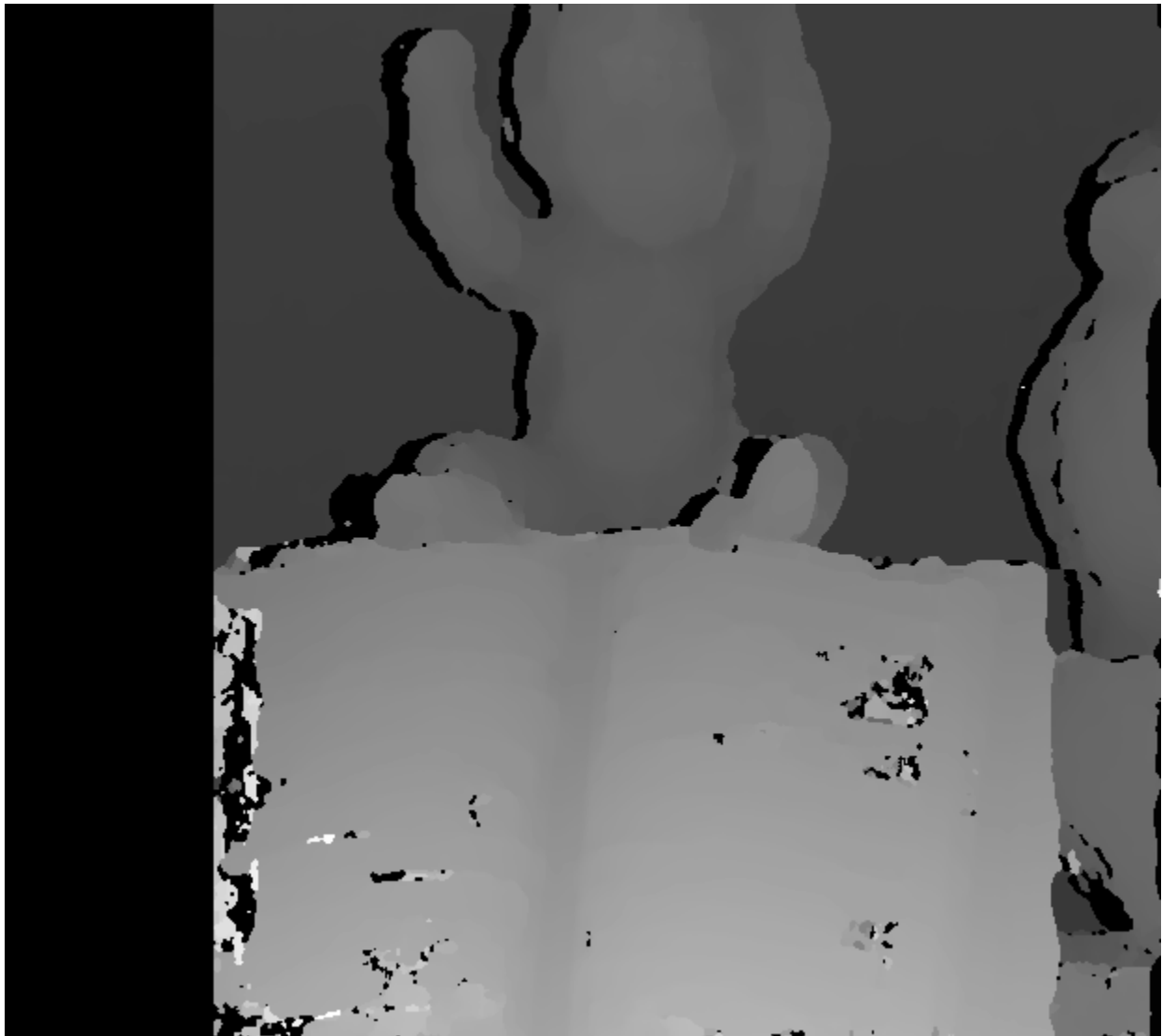
Q1:

Pseudo Code:

- Read corresponding left and right images of an object
- Use stereo SGBM from OpenCV to calculate the disparity map
- Saving the disparity grayscale image
- Distance less than zero in disparity map is made 0
- Create an np mesh grid with image width and height.
- This mesh grid will serve as the x&y coordinates
- Flatten the disparity map and stack with the mesh grid, thereby forming (x,y,z) values for each pixel
- Writing the x,y,z values to the ply file
- Render the ply in cloud compare

baby.png

Parameters used: numDisparities=112, blockSize=17



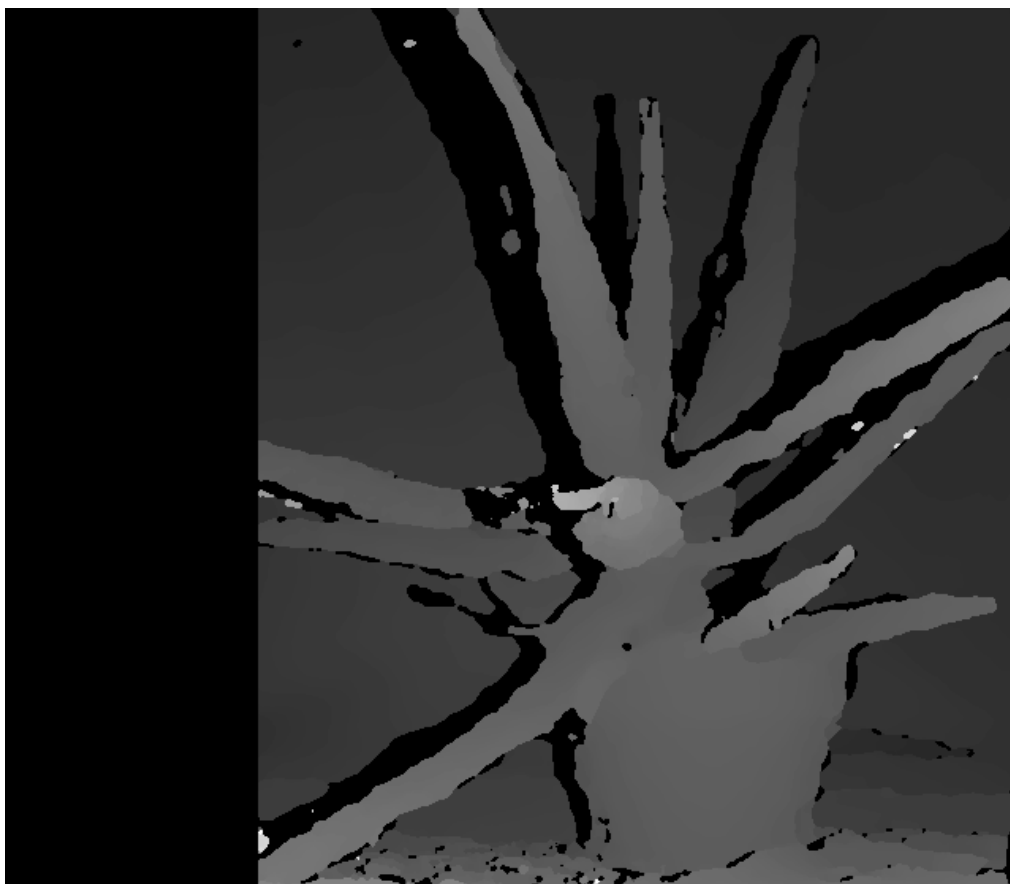
baby-disparity.png



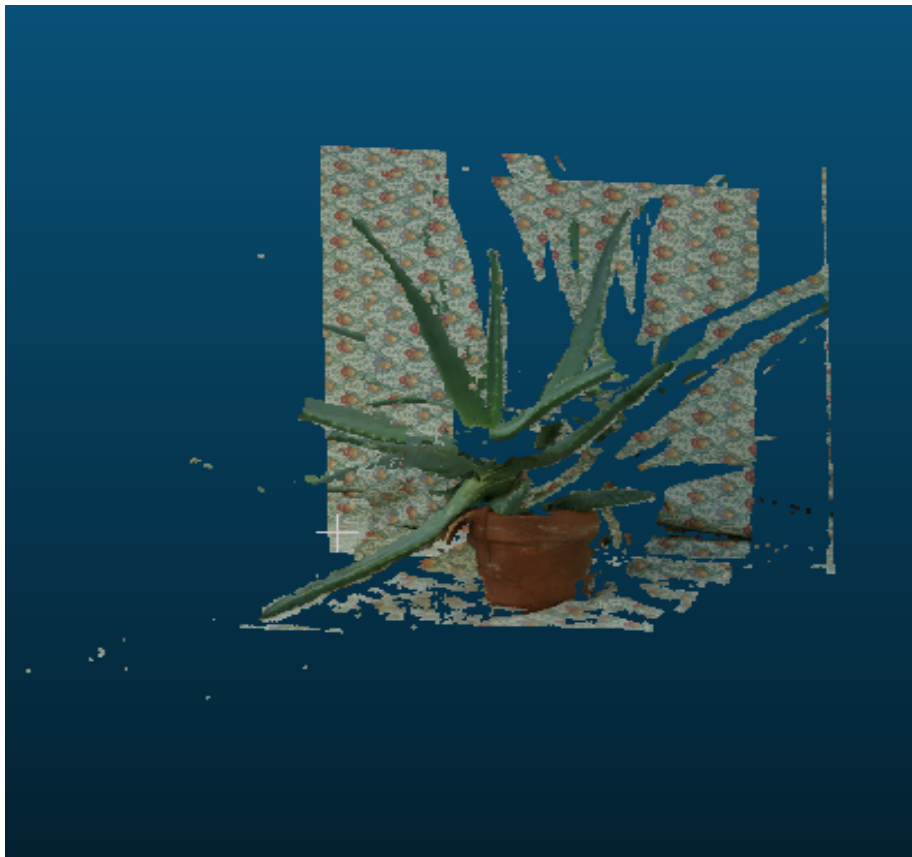
baby-ply

plant.png

Parameters used: numDisparities=160, blockSize=17



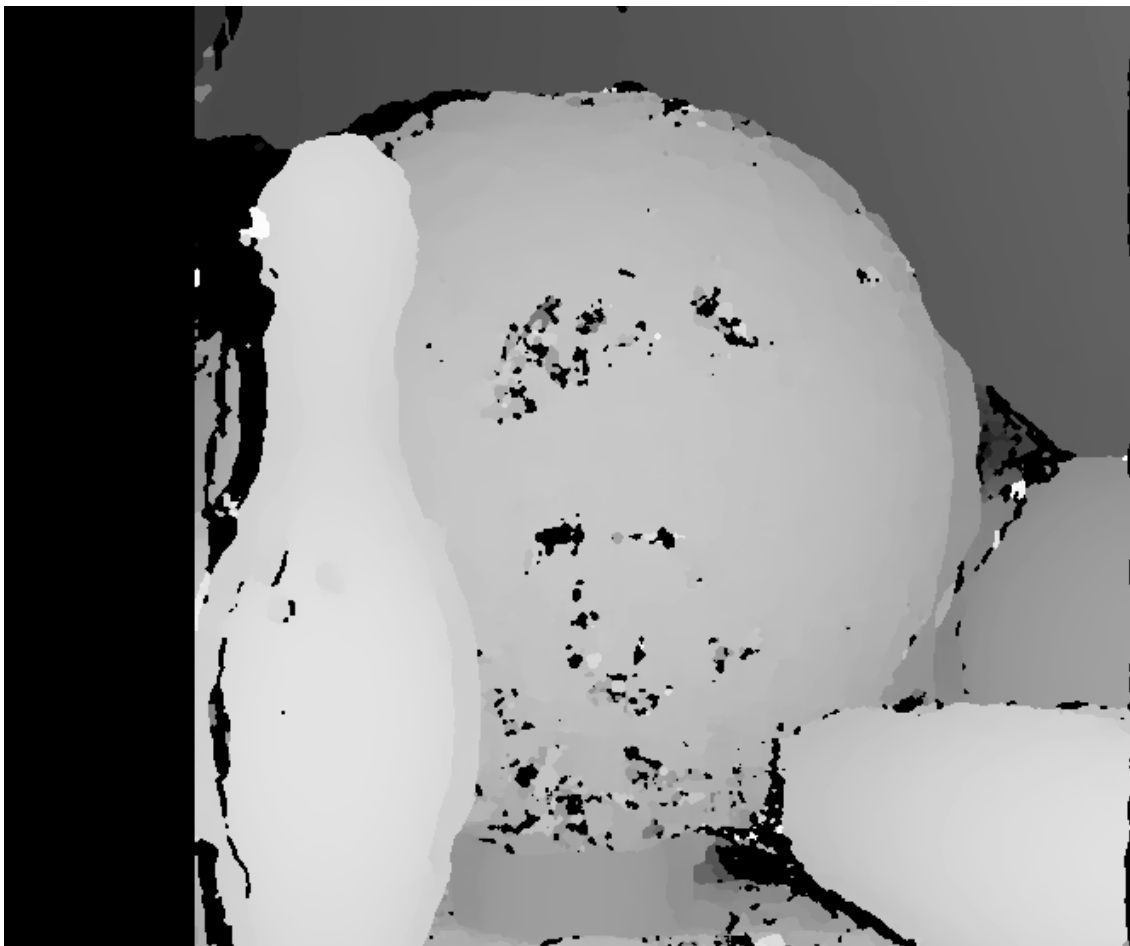
plant-disparity.png



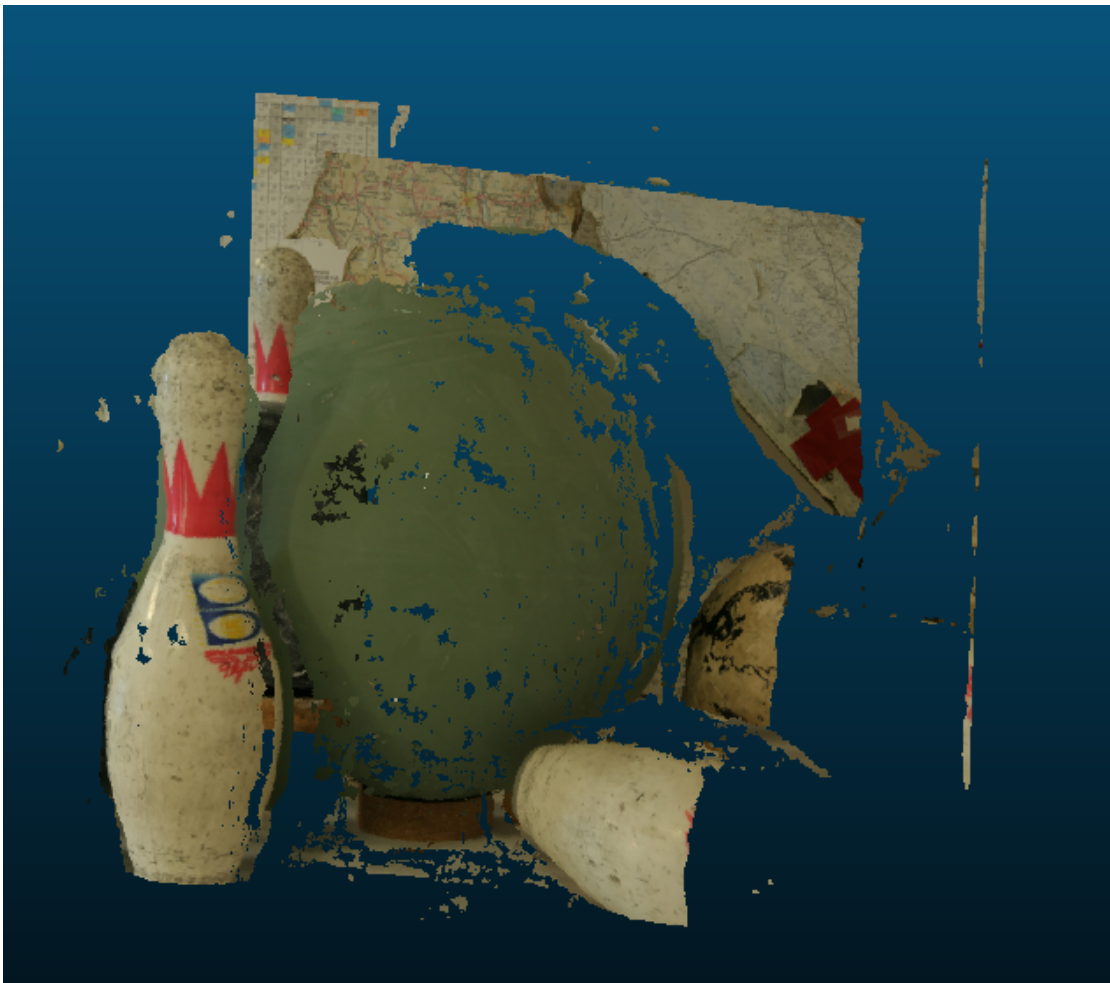
plant-ply

ball.png

Parameters used: numDisparities=112, blockSize=17



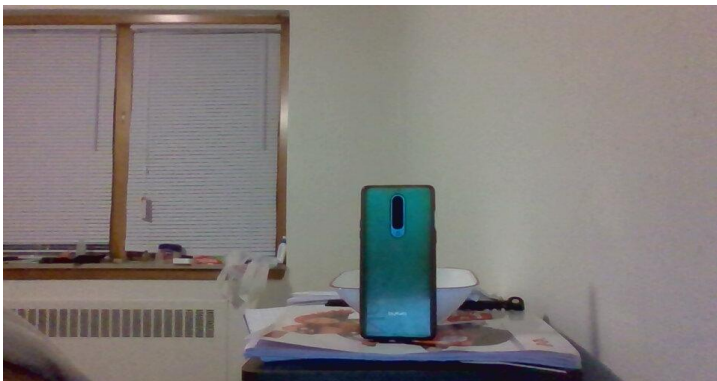
ball-disparity.png



ball-ply

andrew.png

Parameteres used: numDisparities=295, blockSize=5



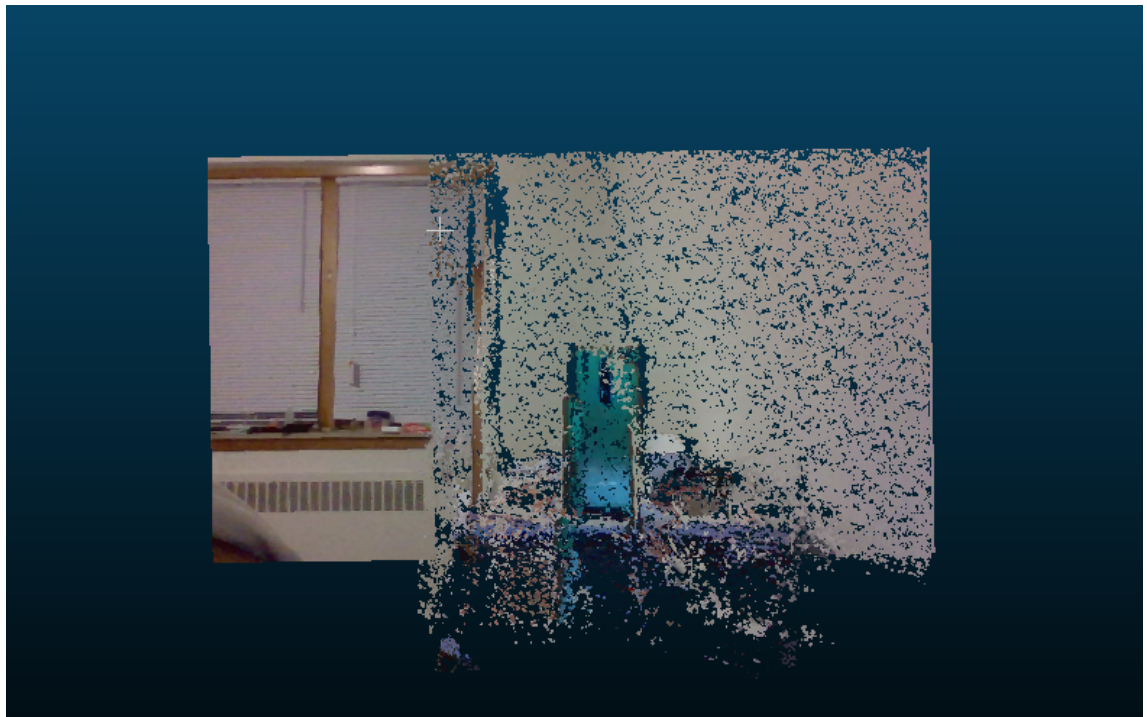
akshayan-left.png



akshayan-right.png



akshayan-disparity.png



Point cloud

Screenshot of codes:
Readme.txt

```
File Edit Selection Find View Goto Tools Project Preferences Help
< > q1.py x readme.txt x andrew.ply x baby.ply
1 Python version: Python 3.9.6
2 OpenCV Version: 4.5.2
3 Operating System: Linux 20.02
4 IDE: Sublime text, run via terminal
5 Almost spend 7 hours for this problem
```

```

import numpy as np
import cv2
import matplotlib.pyplot as plt

if __name__ == '__main__':
    #reading the images
    imL = cv2.imread("/home/akshay/Pictures/Webcam/left.jpg")
    imR = cv2.imread("/home/akshay/Pictures/Webcam/right.jpg")

    # calculating the disparity using SGBM
    stereo = cv2.StereoSGBM_create(numDisparities=295, blockSize=5)
    disparity = stereo.compute(imL, imR)

    #saving the disparity image
    plt.imsave("andrew-disparity.png", disparity, cmap='gray')
    im_shape = disparity.shape
    plt.imshow(disparity, 'gray')
    plt.show()

    #creating the x,y coordinates of point cloud using the meshgrid function
    # and limits as height and width of the image
    x = np.arange(0, im_shape[1], 1)
    y = np.arange(0, im_shape[0], 1)
    xx, yy = np.meshgrid(x, y)
    xx = xx.flatten()
    yy = yy.flatten()

    #Opening the ply file to write points
    f = open("andrew.ply", "a")
    f.write("\n")
    disparity_flat = disparity.flatten()

    #discarding negative values and making them 0, -ve implies no match found
    disparity_flat = np.where(disparity_flat < 0, 0, disparity_flat)
    print(disparity_flat.shape)

    #stacking the disparity values as the z coordinate and
    #writing the 3 coordinates after diving z by 2.
    point_cloud = np.vstack((yy, xx, disparity_flat, imL[:, :, 2].flatten(), imL[:, :, 1].flatten(), imL[:, :, 0].flatten()))
    point_cloud = np.transpose(point_cloud, (1,0))

    for pc in point_cloud:
        l = str(float(pc[0])) + " " + str(float(pc[1])) + " " + str(float(pc[2]/2)) + " " + str(pc[3]) + " " + str(pc[4]) + " " + str(pc[5]) + "\n"
        f.write(l)

```