Assignment 4 - Computer Vision (CSc8830)

Instructor: Dr. Ashwin Ashok - Spring 2024

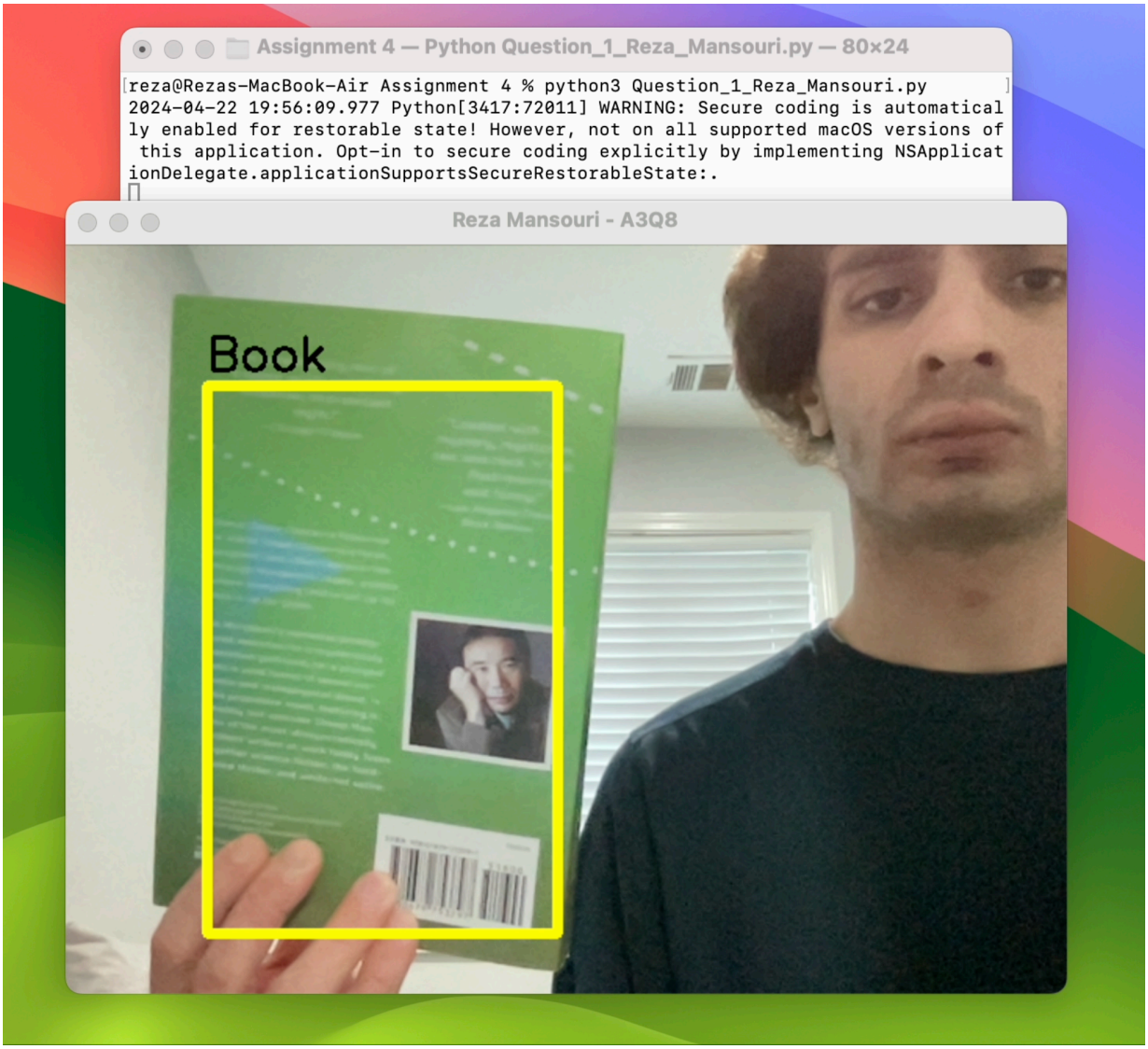
Student name: Reza Mansouri

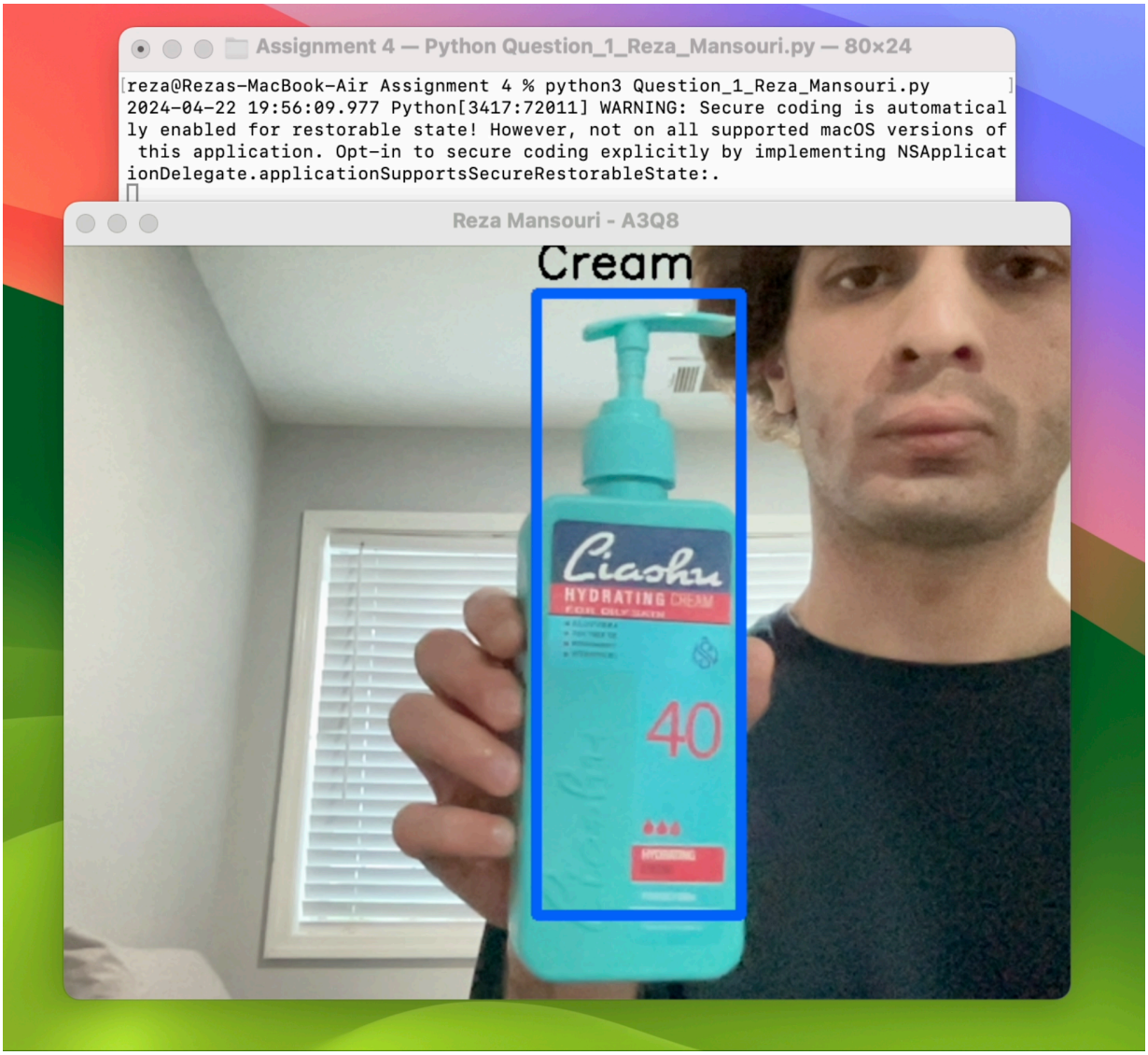
Repository address:
https://github.com/rezmansouri/csc8830/tree/main/Assignment%204

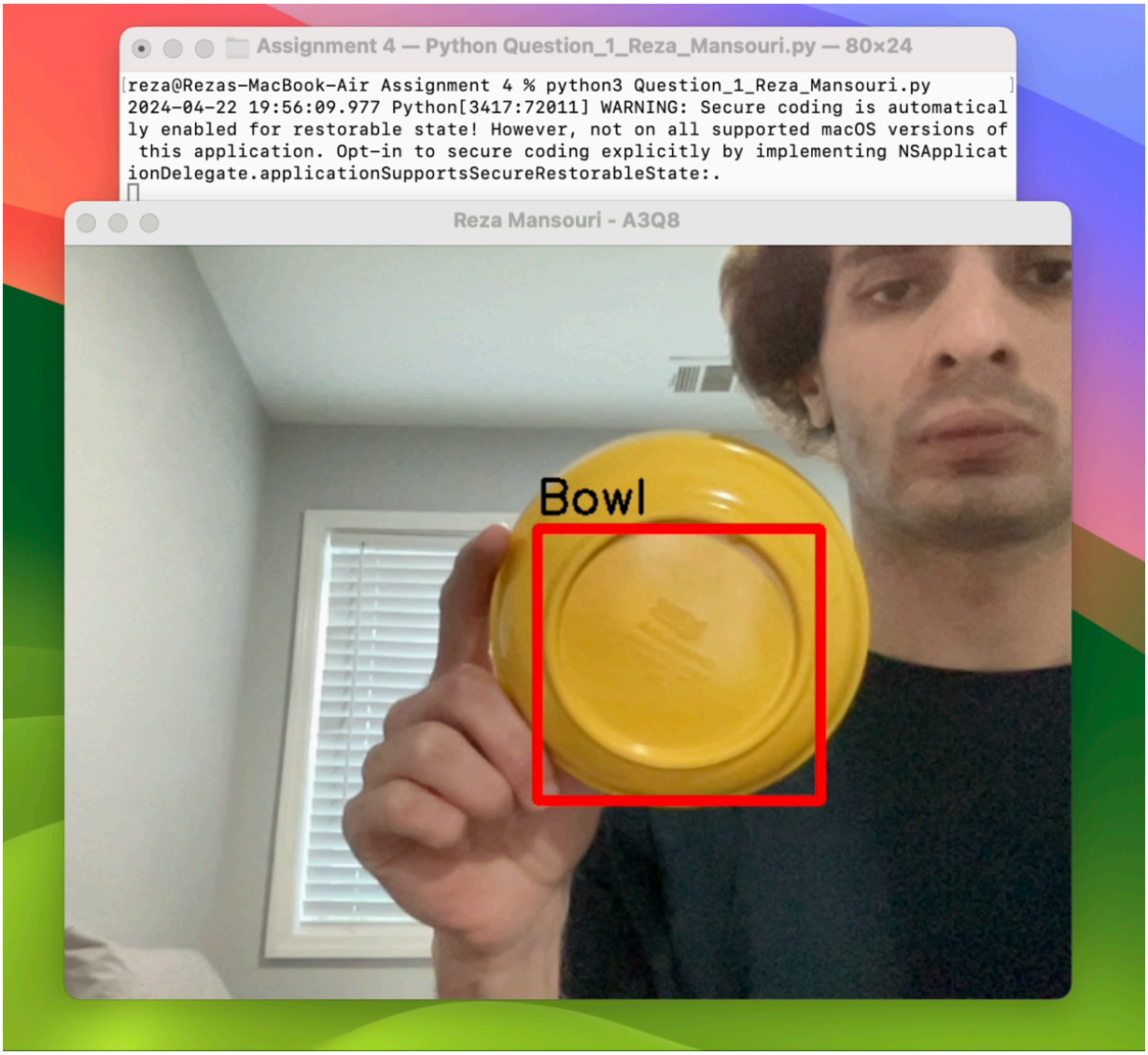
Question 1

Implement an application (must run on web or as an app on mobile device) using the stereo camera where it will recognize, track and estimate dimensions (at least 2D) of any object within 3m distance and inside field-of-view to the camera. You can use barcodes or text recognition tools for identification. However, the entire object must be tracked (not just the barcode or text). Machine/Deep learning tools are NOT allowed.

I implemented the recognition and tracking part by feature matching. Please find the attached screen recording and the code in the attachments. Here's also the code and some footage.


Assignment 4 — Python Question_1_Reza_Mansouri.py — 80×24
reza@Rezas-MacBook-Air Assignment 4 % python3 Question_1_Reza_Mansouri.py
2024-04-22 19:56:09.977 Python[3417:72011] WARNING: Secure coding is automatically enabled for restorable state! However, not on all supported macOS versions of this application. Opt-in to secure coding explicitly by implementing NSApplicationDelegate.applicationSupportsSecureRestorableState:.
Reza Mansouri - A3Q8
Book


Assignment 4 — Python Question_1_Reza_Mansouri.py — 80×24
reza@Rezas-MacBook-Air Assignment 4 % python3 Question_1_Reza_Mansouri.py
2024-04-22 19:56:09.977 Python[3417:72011] WARNING: Secure coding is automatically enabled for restorable state! However, not on all supported macOS versions of this application. Opt-in to secure coding explicitly by implementing NSApplicationDelegate.applicationSupportsSecureRestorableState:.
Reza Mansouri - A3Q8
Cream
Qiaohu
HYDRATING CREAM
FOR OILY SKIN
40




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

object_1 = plt.imread('input/book.jpg')
object_2 = plt.imread('input/cream.jpg')
object_3 = plt.imread('input/bowl.jpg')

label_1 = 'Book'
label_2 = 'Cream'
label_3 = 'Bowl'

cv2.namedWindow("Reza Mansouri - A4Q1", cv2.WINDOW_FREERATIO)
vc = cv2.VideoCapture(0)
vc.set(cv2.CAP_PROP_FRAME_WIDTH, 360)
vc.set(cv2.CAP_PROP_FRAME_HEIGHT, 200)

if vc.isOpened():
    rval, frame = vc.read()
else:
```

```
    rval = False

while rval:
    ssds = []
    coords = []
    for i in range(0, frame.shape[0] - object_1.shape[0], 30):
        for j in range(0, frame.shape[1] - object_1.shape[1], 30):
            candidate = frame[i:i+object_1.shape[0],
j:j+object_1.shape[1]]
            ssd = np.sum(np.power(candidate-object_1, 2))
            ssds.append(ssd)
            coords.append((j, i))
    min_ix = np.argmin(ssds)
    ssd = ssds[min_ix]
    if ssd < np.mean(ssds[~min_ix]) * .95:
        left_top = coords[min_ix]
        right_bottom = (left_top[0] + object_1.shape[1],
left_top[1] + object_1.shape[0])
        frame = cv2.rectangle(frame, left_top, right_bottom,
color=(0, 255, 255), thickness=5)
        frame = cv2.putText(frame, label_1, (left_top[0],
left_top[1]-10), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 2)
    else:
        ssds = []
        coords = []
        for i in range(0, frame.shape[0] - object_2.shape[0], 30):
            for j in range(0, frame.shape[1] - object_2.shape[1],
30):
                candidate = frame[i:i+object_2.shape[0],
j:j+object_2.shape[1]]
                ssd = np.sum(np.power(candidate-object_2, 2))
                ssds.append(ssd)
                coords.append((j, i))
        min_ix = np.argmin(ssds)
        ssd = ssds[min_ix]
        if ssd < np.mean(ssds[~min_ix]) * .94:
            left_top = coords[min_ix]
            right_bottom = (left_top[0] + object_2.shape[1],
left_top[1] + object_2.shape[0])
            frame = cv2.rectangle(frame, left_top, right_bottom,
color=(255, 100, 0), thickness=5)
            frame = cv2.putText(frame, label_2, (left_top[0],
left_top[1]-10), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 2)
        else:
            ssds = []
            coords = []
            for i in range(0, frame.shape[0] - object_3.shape[0],
30):
                for j in range(0, frame.shape[1] -
object_3.shape[1], 30):
```

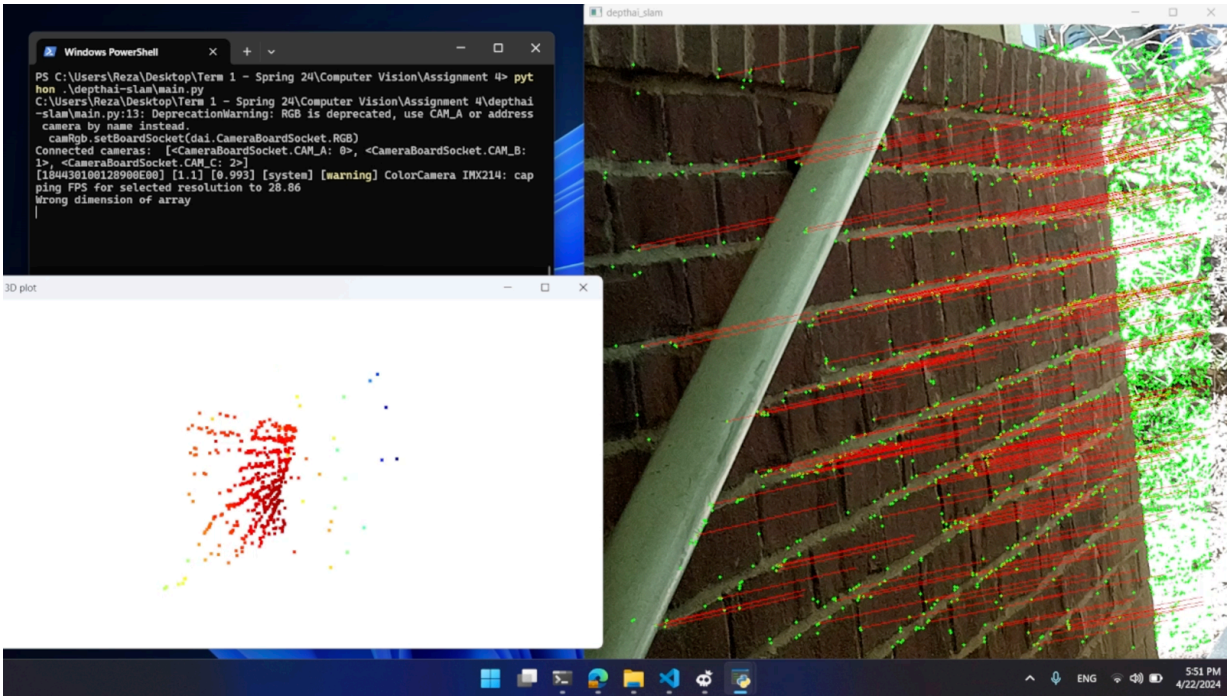
```
                    candidate = frame[i:i+object_3.shape[0],
j:j+object_3.shape[1]]
                    ssd = np.sum(np.power(candidate-object_3, 2))
                    ssds.append(ssd)
                    coords.append((j, i))
            min_ix = np.argmin(ssds)
            ssd = ssds[min_ix]
            if ssd < np.mean(ssds[~min_ix]) * .98:
                left_top = coords[min_ix]
                right_bottom = (left_top[0] + object_3.shape[1],
left_top[1] + object_3.shape[0])
                frame = cv2.rectangle(frame, left_top,
right_bottom, color=(0, 0, 255), thickness=5)
                frame = cv2.putText(frame, label_3, (left_top[0],
left_top[1]-10), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 2)
    cv2.imshow("Reza Mansouri - A4Q1", frame)
    rval, frame = vc.read()
    key = cv2.waitKey(100)
    if key == 27:
        break

vc.release()
cv2.destroyWindow("Reza Mansouri - A4Q1")
```

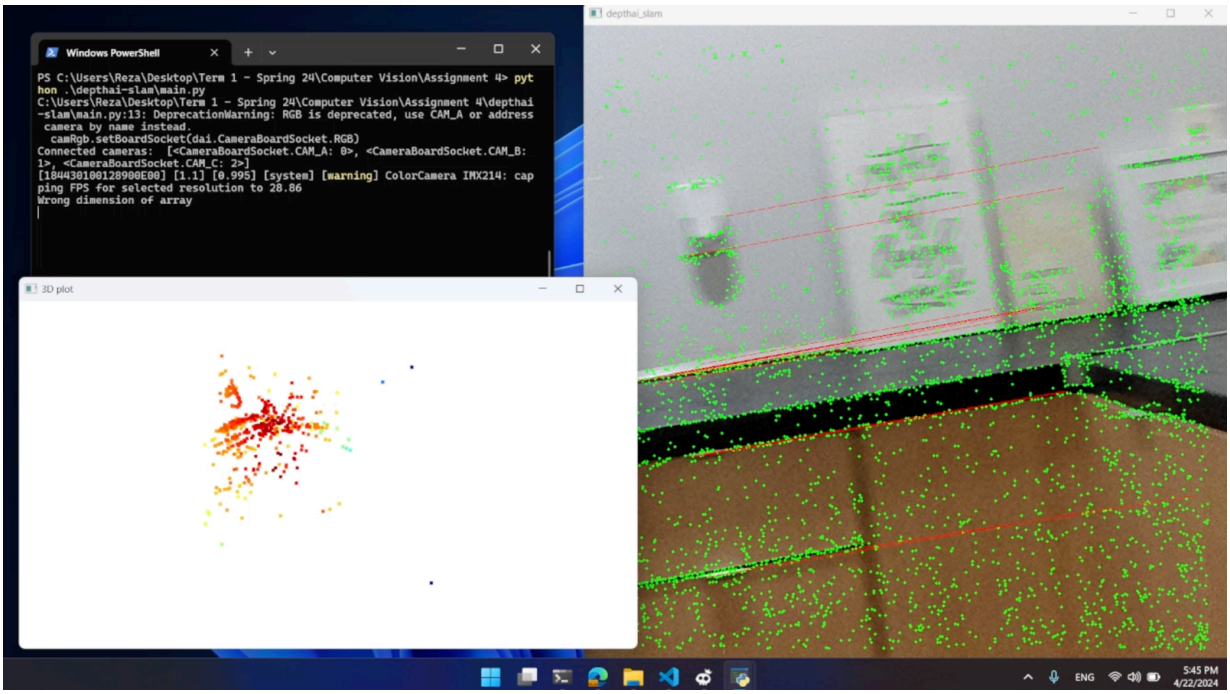
Question 2

Use the DepthAI SDK or use ORB3-Visual SLAM (https://github.com/UZ-SLAMLab/ORB_SLAM3) to execute the scripts on your depth camera and run experiments in two different locations. Provide snapshots of your SLAM output and what limitations/corner cases do you observe.

I used the depthai-slam repository: https://github.com/bharath5673/depthai-slam

The screen recordings have been uploaded and here is some footage:


Windows PowerShell
PS C:\Users\Reza\Desktop\Term 1 - Spring 24\Computer Vision\Assignment 4> pyt
hon .\depthai-slam\main.py
C:\Users\Reza\Desktop\Term 1 - Spring 24\Computer Vision\Assignment 4\depthai
-slam\main.py:13: DeprecationWarning: RGB is deprecated, use CAM_A or address
camera by name instead.
camRgb.setBoardSocket(dai.CameraBoardSocket.RGB)
Connected cameras: [<CameraBoardSocket.CAM_A: 0>, <CameraBoardSocket.CAM_B:
1>, <CameraBoardSocket.CAM_C: 2>]
[18443010012890OE00] [1.1] [0.993] [system] [warning] ColorCamera IMX214: cap
ping FPS for selected resolution to 28.86
Wrong dimension of array
3D plot
depthai_slam
ENG
5:51 PM
4/22/2024


Windows PowerShell
PS C:\Users\Reza\Desktop\Term 1 - Spring 24\Computer Vision\Assignment 4> pyt
hon .\depthai-slam\main.py
C:\Users\Reza\Desktop\Term 1 - Spring 24\Computer Vision\Assignment 4\depthai
-slam\main.py:13: DeprecationWarning: RGB is deprecated, use CAM_A or address
camera by name instead.
camRgb.setBoardSocket(dai.CameraBoardSocket.RGB)
Connected cameras: [<CameraBoardSocket.CAM_A: 0>, <CameraBoardSocket.CAM_B:
1>, <CameraBoardSocket.CAM_C: 2>]
[18443010012890OE00] [1.1] [0.995] [system] [warning] ColorCamera IMX214: cap
ping FPS for selected resolution to 28.86
Wrong dimension of array
3D plot
depthai_slam
ENG
5:45 PM
4/22/2024