9.1

Monday, 12 October 2020 3:57 PM

Testing of python sample code 04_keypoints_from_images

```
Body keypoints:
[[[3.1544412e+02 4.8411846e+01 8.0357540e-01]
  [3.0963919e+02 9.6343147e+01 8.1904769e-01]
  [2.7242642e+02 9.7292473e+01 7.3504728e-01]
  [2.1570950e+02 9.9286285e+01 7.7710027e-01]
  [1.5895428e+02 1.0420146e+02 7.7955765e-01]
  [3.4776492e+02 9.5388687e+01 7.2341627e-01]
  [4.0351672e+02 9.8325829e+01 8.3219570e-01]
  [4.7199103e+02 1.0420013e+02 5.8732933e-01]
  [3.0863235e+02 2.0983218e+02 5.6048787e-01]
  [2.8516928e+02 2.0788515e+02 5.3471798e-01]
  [2.8511249e+02 2.9298840e+02 8.1189322e-01]
  [2.7836523e+02 3.5758081e+02 6.2487118e-02]
  [3.3210388e+02 2.0984306e+02 5.2222937e-01]
  [3.3895288e+02 2.9392432e+02 8.4350091e-01]
  [3.4484348e+02 3.5757697e+02 2.2904444e-01]
  [3.0569620e+02 4.1600113e+01 8.1091797e-01]
  [3.2134830e+02 4.1569492e+01 8.1277883e-01]
  [2.9199545e+02 4.6476624e+01 8.4299642e-01]
  [3.2820767e+02 4.5497185e+01 4.5145312e-01]
  [0.0000000e+00 0.0000000e+00 0.0000000e+00]
  [0.0000000e+00 0.0000000e+00 0.0000000e+00]
  [3.4773383e+02 3.5758664e+02 9.5163584e-02]
  [0.0000000e+00 0.0000000e+00 0.0000000e+00]
  [0.0000000e+00 0.0000000e+00 0.0000000e+00]
  [0.0000000e+00 0.0000000e+00 0.0000000e+00]]]
```

```
"part_candidates":[{
    "0":[315.444,48.4118,0.803575,527.773,324.249,0.0575425],
    "1":[309.639,96.3431,0.819048],
    "2":[272.426,97.2925,0.735047],
    "3":[215.71,99.2863,0.7771],
    "4":[158.954,104.201,0.779558],
    "5":[347.765,95.3887,0.723416],
    "6":[403.517,98.3258,0.832196],
    "7":[471.991,104.2,0.587329],
    "8":[308.632,209.832,0.560488],
    "9":[285.169,207.885,0.534718],
    "10":[285.112,292.988,0.811893],
    "11":[278.365,357.581,0.0624871],
    "12":[332.104,209.843,0.522229],
    "13":[338.953,293.924,0.843501],
    "14":[344.843,357.577,0.229044],
```

Doing this since this sample code seems to be able to run it continuously, but I wasnt able to integrate it due to openpose running in synchronous mode or sth. Tried searching around on how to make it asynchronous but nothing works as of now :(

9.2

Tuesday, 13 October 2020 2:08 PM

Meeting CA3:

- Week 11 upload spec sheet

Include:

- Characterisation of the system. (eg fps, accuracy, resolution)
- GUI seems to be quite important to show how user might want to use it. Might want to allocate more time?
- Compiling python code into exec?
- Sampling each frame at longer intervals?

9.3

Friday, 16 October 2020 9:10 PM

```
# run through deprojection (2D to 3D)
depth_point = rs.rs2_deproject_pixel_to_point(depth_intrin, [j, i], depth)
text = "%.5lf, %.5lf, %.5lf\n" % (depth_point[0], depth_point[1], depth_point[2])
print("text:", text)
```

```
D:\STUDIES\Python\python.exe
text: 0.00306, 0.11777, 1.01600

depth: 1.0190000534057617
text: 0.00307, 0.11812, 1.01900

depth: 1.0140000581741333
text: 0.00306, 0.11754, 1.01400

depth: 1.0080000162124634
text: 0.00304, 0.11684, 1.00800

depth: 1.0080000162124634
text: 0.00304, 0.11684, 1.00800

depth: 1.0120000839233398
text: 0.00305, 0.11730, 1.01200

depth: 1.0099999904632568
text: 0.00305, 0.11707, 1.01000

depth: 1.0099999904632568
text: 0.00305, 0.11707, 1.01000

depth: 1.0099999904632568
text: 0.00305, 0.11707, 1.01000

depth: 1.0080000162124634
text: 0.00304, 0.11684, 1.00800
```

realsense_depth_test.py realsense_openpose_noOP.py T_pose_end_angle

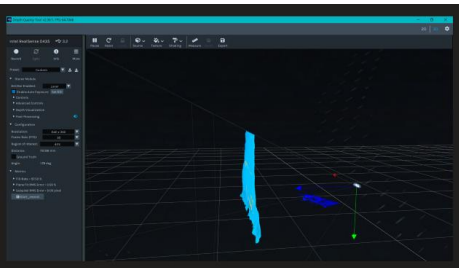
Still not sure what does depth_point[0] and [1] means but doesnt seems to matter. Will focus on using the depth part only and try to understand how is it gotten and how to make use of it.

3 of 4

9.4

Saturday, 17 October 2020 12:11 PM

Me + WP
- So depth is a plane projection



	average	stdev
2 tiles	0.8874	0.002159
5 tiles	2.37869	0.040418
10 tiles	5.404533	0.141619

Each tile is about ~0.45m.

So it seems like the further the point, the more inaccurate it gets...

=> Maybe take more points and make a regression curve to map perceived distance to real distance?

Need to find a way to add depth into the conditions.

4 of 4