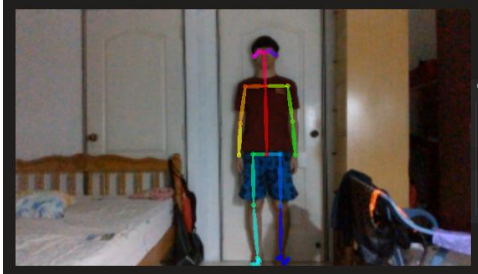
10.1

Monday, 19 October 2020 10:08 PM

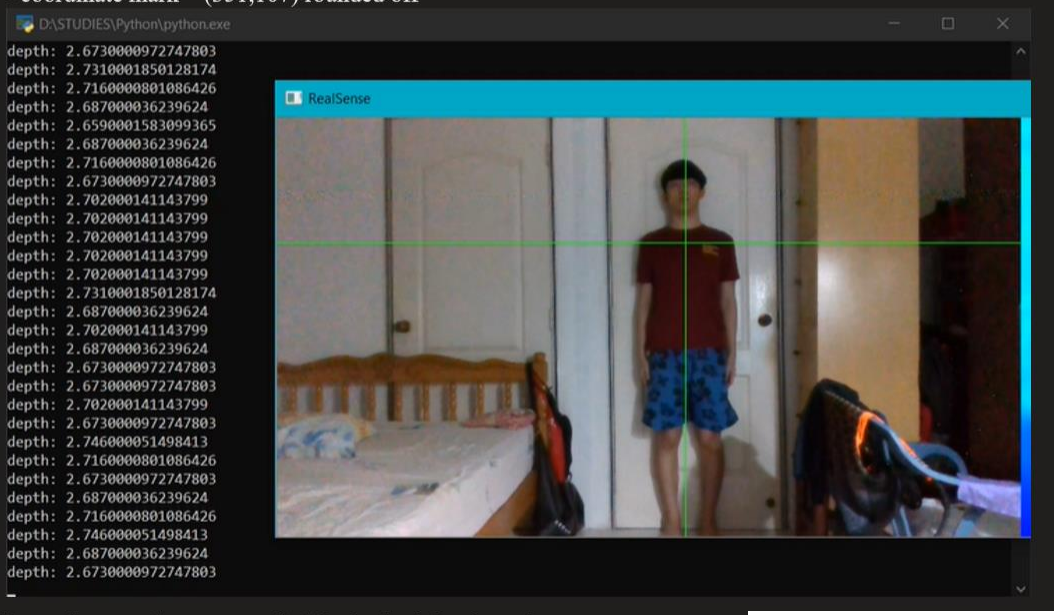
To help figure out the depth problem, I am going to experiment on using just realsense code to find depth, then using the same point on realsense + openpose.

1) Mark a sport
2) Use openpose to find the coordinate of my chest

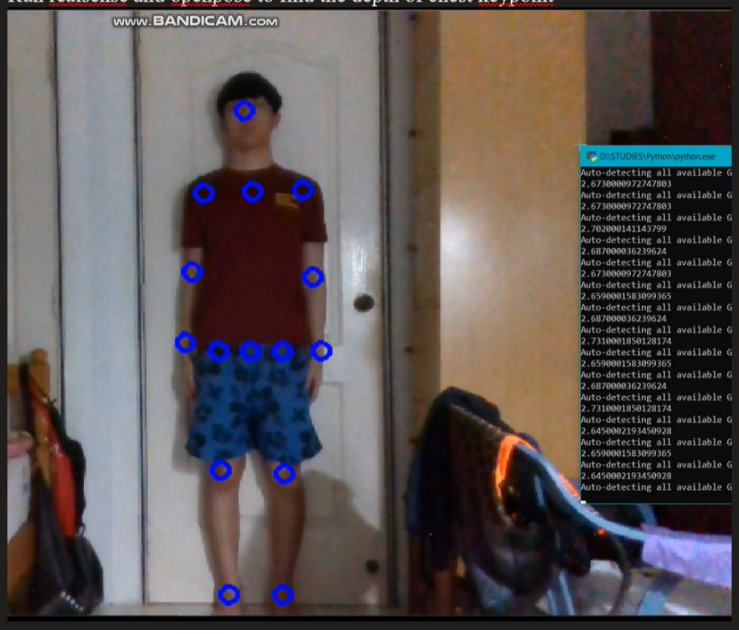



```
"part_candidates":[{
    "0":[348.741,61.1355,0.881971,228.402,213.739,0.11792,476.885,256.794,0.0705132],
    "1":[350.704,107.113,0.819585,222.559,233.327,0.0573557,488.638,269.47,0.394639],
    "2":[322.309,108.094,0.757561,227.443,235.262,0.0776108,511.125,261.684,0.134751,
    261,329.176,0.0959637],"9":[333.103,202.986,0.571967,490.594,334.07,0.0995396],"1
    705,0.093527,466.146,252.861,0.0556412],"18":[365.394,64.099,0.845031,228.414,215
```


3) Run realsense and find the depth of that coordinate
   - coordinate mark ~ (351,107) rounded off

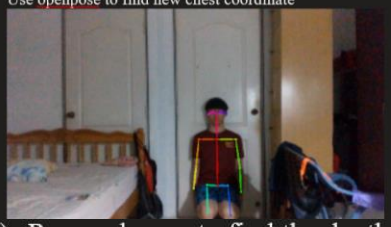


4) Run realsense and openpose to find the depth of chest keypoint



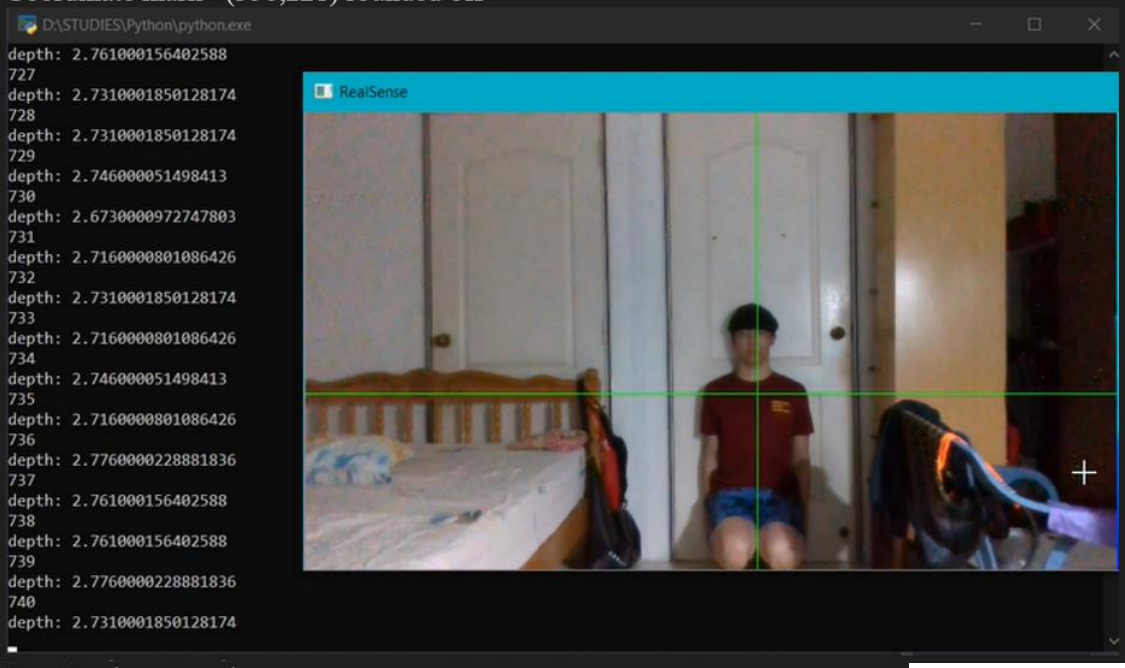
Next, i want to try if there is a different in depth if the chest is at a lower position, but other axis stays relatively constant.

1) Use back the same spot
2) Use openpose to find new chest coordinate

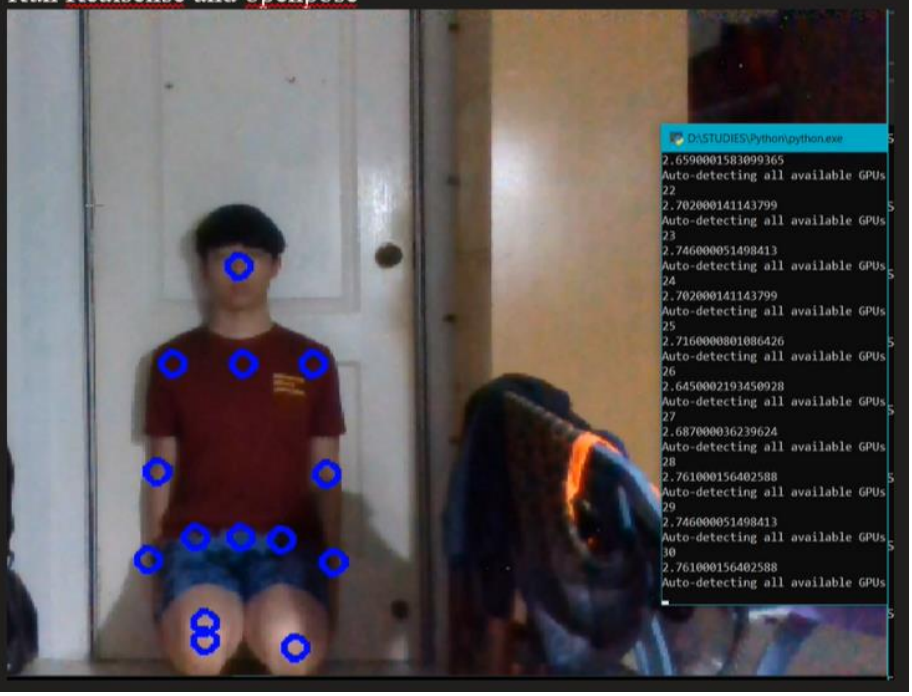



```
"part_candidates":[{
    "0":[355.606,178.524,0.872644,228.427,213.755,0.107086],
    "1":[355.628,220.58,0.854795,225.485,238.168,0.0868016,235.276,261.691,0.100731,489.59,261.663,0.0794093],
    "2":[325.248,220.6,0.790358,225.487,238.183,0.11845,243.086,262.639,0.0796858],"3":[320.355,266.555,0.73155
    .74969],"11":[],"12":[374.187,298.864,0.640103,238.215,305.705,0.0582235],"13":[379.078,349.71,0.657397],"1
```


3) Run realsense to find the depth of the new chest coordinate
   - Coordinate mark ~(356,221) rounded off



4) Run Realsense and openpose



Trying to fix the FPS problem:
- Changed to "opWrapper = op.WrapperPython(0)"
- Below is the datum.poseKeypoints array

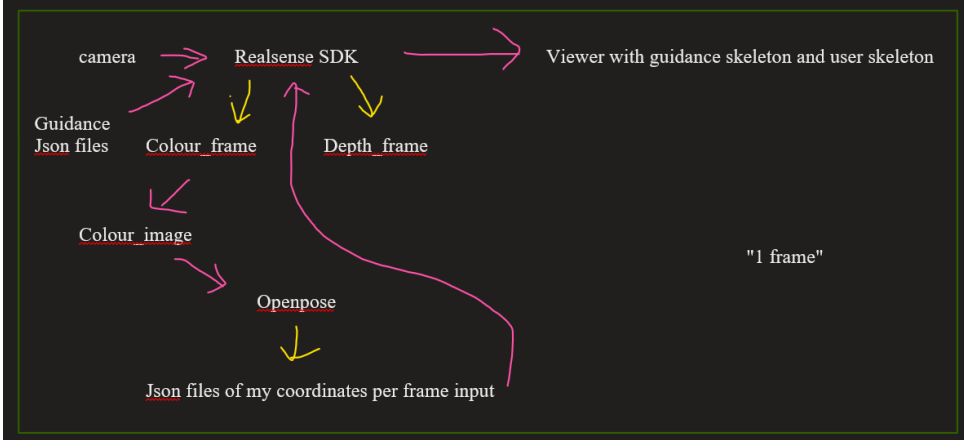
```
[[[2.30352875e+02 2.13763245e+02 3.96217048e-01]
  [2.23552017e+02 2.30349625e+02 3.20139408e-01]
  [2.22533173e+02 2.30349365e+02 3.01882744e-01]
  [2.25489456e+02 2.48958755e+02 1.12792544e-01]
  [0.00000000e+00 0.00000000e+00 0.00000000e+00]
  [2.28402893e+02 2.30403961e+02 2.85176396e-01]
  [2.27456451e+02 2.51904648e+02 1.39708638e-01]
  [2.16668762e+02 2.56804565e+02 1.60260096e-01]
  [2.30359802e+02 2.60708130e+02 1.38193980e-01]
  [2.27432129e+02 2.60687897e+02 1.39270425e-01]
  [0.00000000e+00 0.00000000e+00 0.00000000e+00]
  [0.00000000e+00 0.00000000e+00 0.00000000e+00]
  [2.32339493e+02 2.59720123e+02 1.16925165e-01]
  [0.00000000e+00 0.00000000e+00 0.00000000e+00]
  [0.00000000e+00 0.00000000e+00 0.00000000e+00]
  [2.26477325e+02 2.09850250e+02 4.24702346e-01]
  [2.30361694e+02 2.08891922e+02 2.67397791e-01]
  [2.22519287e+02 2.13758850e+02 4.09335166e-01]
  [0.00000000e+00 0.00000000e+00 0.00000000e+00]
  [0.00000000e+00 0.00000000e+00 0.00000000e+00]
  [0.00000000e+00 0.00000000e+00 0.00000000e+00]
  [0.00000000e+00 0.00000000e+00 0.00000000e+00]
  [0.00000000e+00 0.00000000e+00 0.00000000e+00]
  [0.00000000e+00 0.00000000e+00 0.00000000e+00]
  [0.00000000e+00 0.00000000e+00 0.00000000e+00]]]
```

- 25 rows, 3 columns
- <class 'numpy.ndarray'>
- 3 dimensional using .ndim

Error:
print(np.userskele_json_part([0]))
TypeError: 'numpy.ndarray' object is not callable

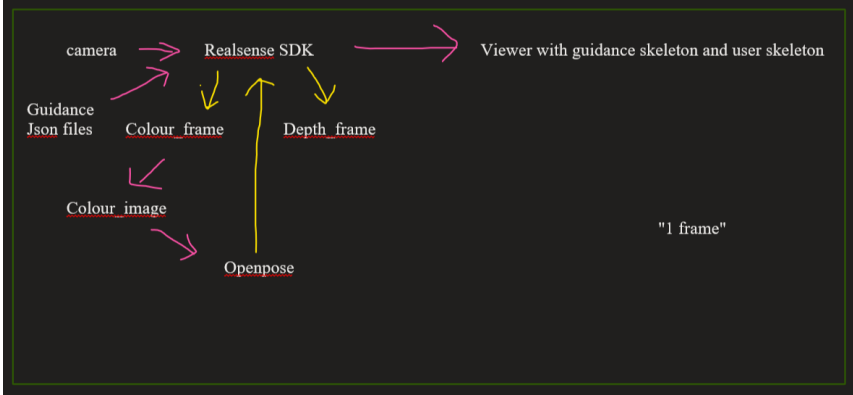
Fixed

10.2
Tuesday, 20 October 2020 3:35 PM



OLD

"1 frame"



"1 frame"

NEW (V3 onwards)
- FPS kinda improved to 7FPS

Update meeting:

- Try openpose with webcam get fps.
- Look into the callibration of colour and depth image
- Give more specification for Nab, and conditions.
- Maybe have the guidance follow the user. And normalised to the user

5 of 8

10.3

Wednesday, 21 October 2020 11:38 AM

Me + WP
- Retest depth calculation using V3 of our code (fps improved)
- FIXED DISTANCE BETWEEN 2 POINTS PROBLEM

GUI Meeting (Me + WP + Nab):
- Decide on functions that we want on the viewer, then let nab to do the rest of the code.
- We want a functional Help window and warning lines (both collapsible)
- For GUI, i need to add a button to show user step-by-step instructions on what to do for this exercise.

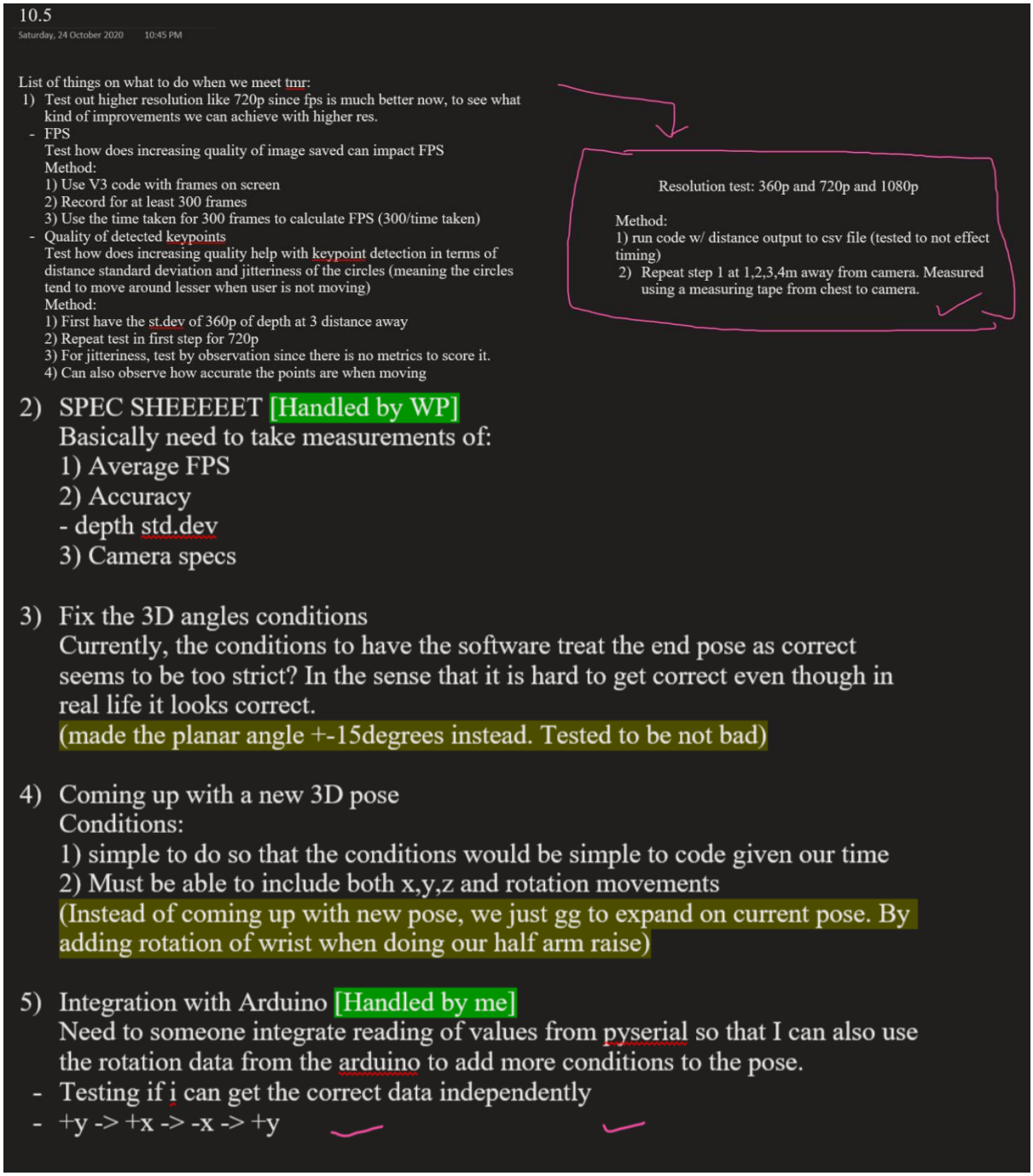
Realsense_openpose_half_arm_raise_2D Version:
- Add looping between start and end (for repetition)
- Added more user angle calc (123, 234, 813)
- Added more conditions

Realsense_openpose_half_arm_raise_3D Version:
- Add depth
- Added dot product between normal of chest to 1-4
- Added depth condition but too strict? Hard to invoke correct pose.
- Still got error for 234 :(((

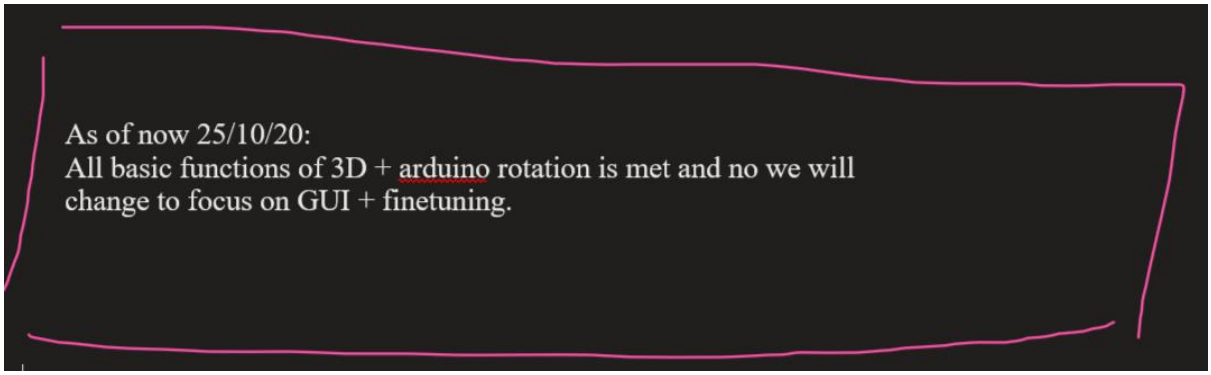
```
Traceback (most recent call last):
  File "D:\STUDIES\YEAR 3 - SEM 1\ESP3902\openpose-master\build\examples\tutorial_api_python\realsense_OP_Half_ar
_3D.py", line 217, in <module>
    user_angle234 = math.degrees(math.acos(np.divide(dot_prdt,(len_AB*len_AC))))
ValueError: math domain error

Process returned 1 (0x1)        execution time : 114.313 s
Press any key to continue . . .
```

6 of 8


10.5
Saturday, 24 October 2020 10:45 PM
List of things on what to do when we meet tmr:
1) Test out higher resolution like 720p since fps is much better now, to see what kind of improvements we can achieve with higher res.
- FPS
Test how does increasing quality of image saved can impact FPS
Method:
1) Use V3 code with frames on screen
2) Record for at least 300 frames
3) Use the time taken for 300 frames to calculate FPS (300/time taken)
- Quality of detected keypoints
Test how does increasing quality help with keypoint detection in terms of distance standard deviation and jitteriness of the circles (meaning the circles tend to move around lesser when user is not moving)
Method:
1) First have the st.dev of 360p of depth at 3 distance away
2) Repeat test in first step for 720p
3) For jitteriness, test by observation since there is no metrics to score it.
4) Can also observe how accurate the points are when moving
Resolution test: 360p and 720p and 1080p
Method:
1) run code w/ distance output to csv file (tested to not effect timing)
2) Repeat step 1 at 1,2,3,4m away from camera. Measured using a measuring tape from chest to camera.
2) SPEC SHEEEEET [Handled by WP]
Basically need to take measurements of:
1) Average FPS
2) Accuracy
- depth std.dev
3) Camera specs
3) Fix the 3D angles conditions
Currently, the conditions to have the software treat the end pose as correct seems to be too strict? In the sense that it is hard to get correct even though in real life it looks correct.
(made the planar angle +-15degrees instead. Tested to be not bad)
4) Coming up with a new 3D pose
Conditions:
1) simple to do so that the conditions would be simple to code given our time
2) Must be able to include both x,y,z and rotation movements
(Instead of coming up with new pose, we just gg to expand on current pose. By adding rotation of wrist when doing our half arm raise)
5) Integration with Arduino [Handled by me]
Need to someone integrate reading of values from pyserial so that I can also use the rotation data from the arduino to add more conditions to the pose.
- Testing if i can get the correct data independently
- +y -> +x -> -x -> +y

7 of 8


As of now 25/10/20:
All basic functions of 3D + arduino rotation is met and no we will change to focus on GUI + finetuning.

8 of 8