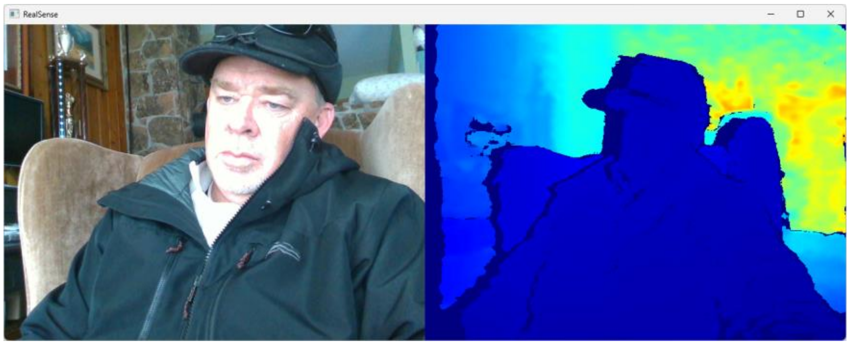
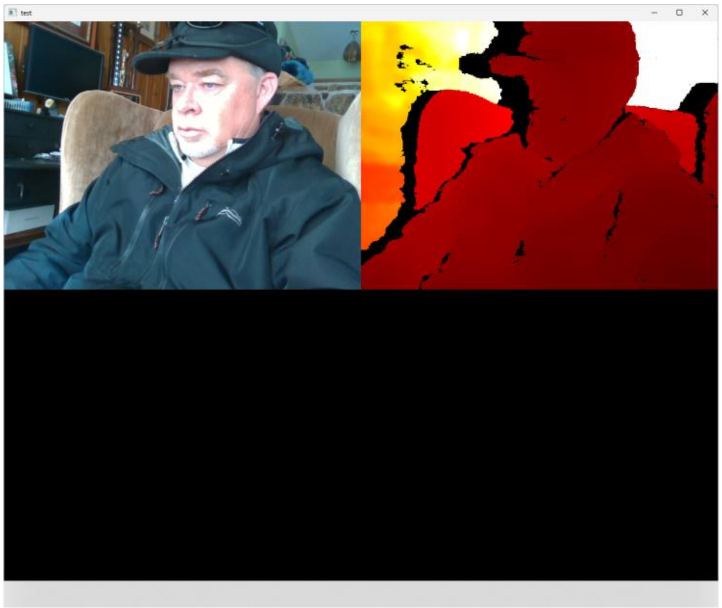
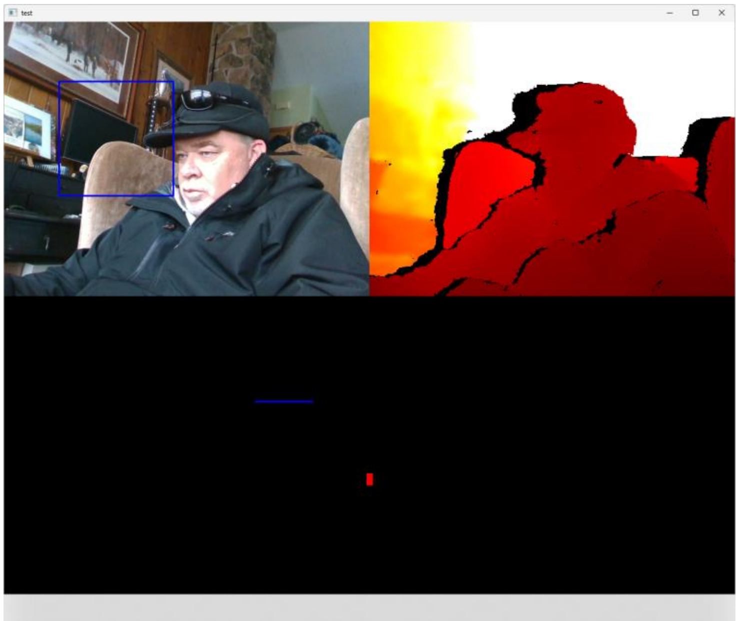
- Setup Python 3.10 (Python 3.11 will not work with the RealSense Libraries)
- Setup OPENCV with opencv-contrib-python
    - If you already have opencv installed on Python then you will need to do the following
        - uninstall both ("opencv-python" and "opencv-contrib-python") and then install "opencv-contrib-python"
        - to uninstall use this command:
            - pip uninstall opencv
        - to install use this command:
            - pip install opencv-contrib-python
- Install the pyrealsense2 libraries
    - pip install pyrealsense2
- Use the link to find the sample realsense code:
    - https://dev.intelrealsense.com/docs/python2
    - Use the demo source opencv_viewer_example.py code to get depth and color image working
- Display the color image and depth camera to look like this, the code should do all this for you:
- 

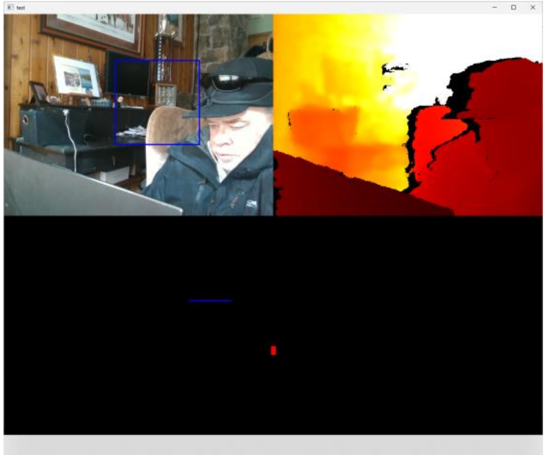
- Next you will add a second blank, black image below these two images in the same window so it looks like this, the size of this black image is the same as the color_image and depth image combined:

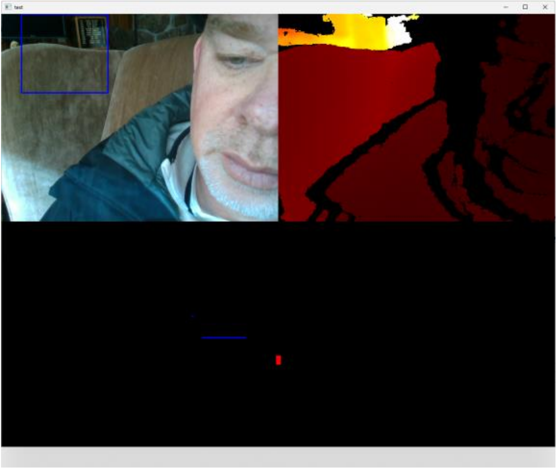


- The color_image and depth image were combined using np.hstack, you will also need to use np.vstack to add the third image.
- Now you are all set to do the actual assignment.
- First you need to get a tracker tracking something.
    - Here is a good explanation I used in class
    - https://learnopencv.com/object-tracking-using-opencv-cpp-python/
    - The source code on this page is in C++ so you have to interpret it back to python to use in your program.
    - Steps I did, I got this code working in Python, then I tried all the trackers.
        - I only used the KCF tracker after testing, it was by far the best for what we are doing.
    - Get the KCF tracker going with your realsense code/camera.
    - You can start with using the drawing a box around the object you want to track and just tracking the one item.
    - Once you have the rectangle on your object make sure the tracker is tracking it, this part is pretty automatic if you get everything set up correctly.
    - Once you have your tracked object find the depth of that object. A good example of finding the depth is the next demo back on the RealSense site called align-depth2color (you can find this example code on the first link above).
    - Now that you have a distance of the object being tracked you can start drawing the results in the bottom black drawing panel like this:



- The corner of the chair is being tracked, in my drawing panel you can see the red rectangle represents the camera and the blue line is representing the corner of the chair being tracked.
- As I move the camera farther away from the chair (my arms are only so long) the blue line moves farther away from the red rectangle, picture on left…… and in the picture on right as I move the camera closer to the tracked object the blue line is being drawn closer to the robot/camera/red rectangle.





Get all this working first, I will be adding a little bit more to this and make a video demo as soon as I get some time.

- Setup camera view
    - Get colorImage
    - Get depthImage
- Turn color to HSV
- Track a mouse click color in HSV
- Find the depth of the white colors in the color image
- Second image draw where the image is and where the camera is in a x, y plane
- Move forward and test the depth of same colors.
- Redraw the x, y plane updated with movement
- Track more than one HSV object at a time.
-
-