Robotics Lab 03

Syeda Manahil Wasti, Syed Mujtaba Hassan, Aatiqa Khalid

30^{th} January 2023

Task 3.1: Understanding functioning of system

(a) Identify and list all the sensors and actuators in our complete robotic system illustrated in Figure 3.1.

Solution:
<u>Sensors</u>

1. Position sensor (Potentiometer)

<u>Actuators</u>

1. Motors

(b) Figure 3.3 suggests that a potentiometer is being used as the shaft position sensor. Research and describe how can a potentiometer be used for this purpose.

Solution: Potentiometric sensors measure the resistance of a conductor track between a reference point and a cursor attached to a moving element (or its support). It uses a wiper contact linked to a mechanical shaft that can be either angular or linear in its movement along a track.
This movement causes the resistance value between the wiper/slider and the two end connections to change giving an electrical signal output that has a proportional relationship between the actual wiper position on the resistive track and its resistance value. In other words, resistance is proportional to the physical position. [1]

Task 3.2: Motor Specifications

1. This task is about familiarizing ourselves with the Dynamixel reference manual and some relevant specifications included in it.
   (a) Find the angle rotation limits, resolution, speed limit, and torque limit of AX-12A servo.

   Solution:

   1. **Angle rotation limits:** CCW= 150° - 300°
      CW= 0°- 150°
   2. **Resolution:** 0.29°
   3. **Speed limit:** In Joint mode, max limit is 114 rpm.
      No Load Speed: 33.2 rpm

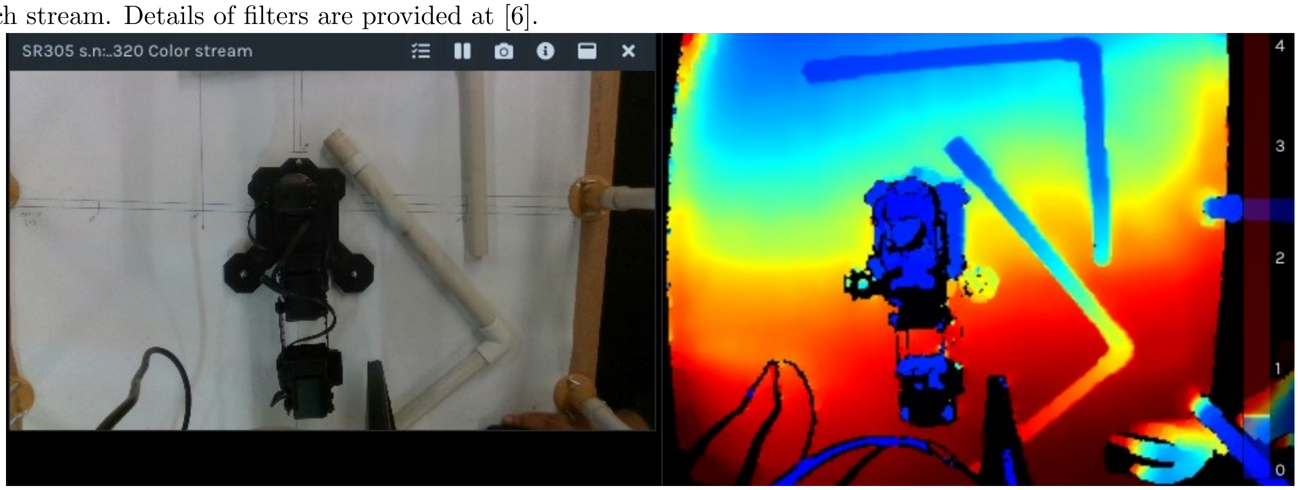
4. **Torque limit:** Stall torque: 1.5 [N.m] (at 12 [V], 1.5 [A])

(b) Will this motor resolution limit the possible Cartesian resolution of the end-effector? If yes, why?

Solution: Yes, the motor's resolution will limit the possible Cartesian resolution of the end-effector because if motor has limited resolution then it can move only in a limited number of positions and the robot will not have a greater volume of space to operate in. Thus, the Cartesian resolution will also be limited.

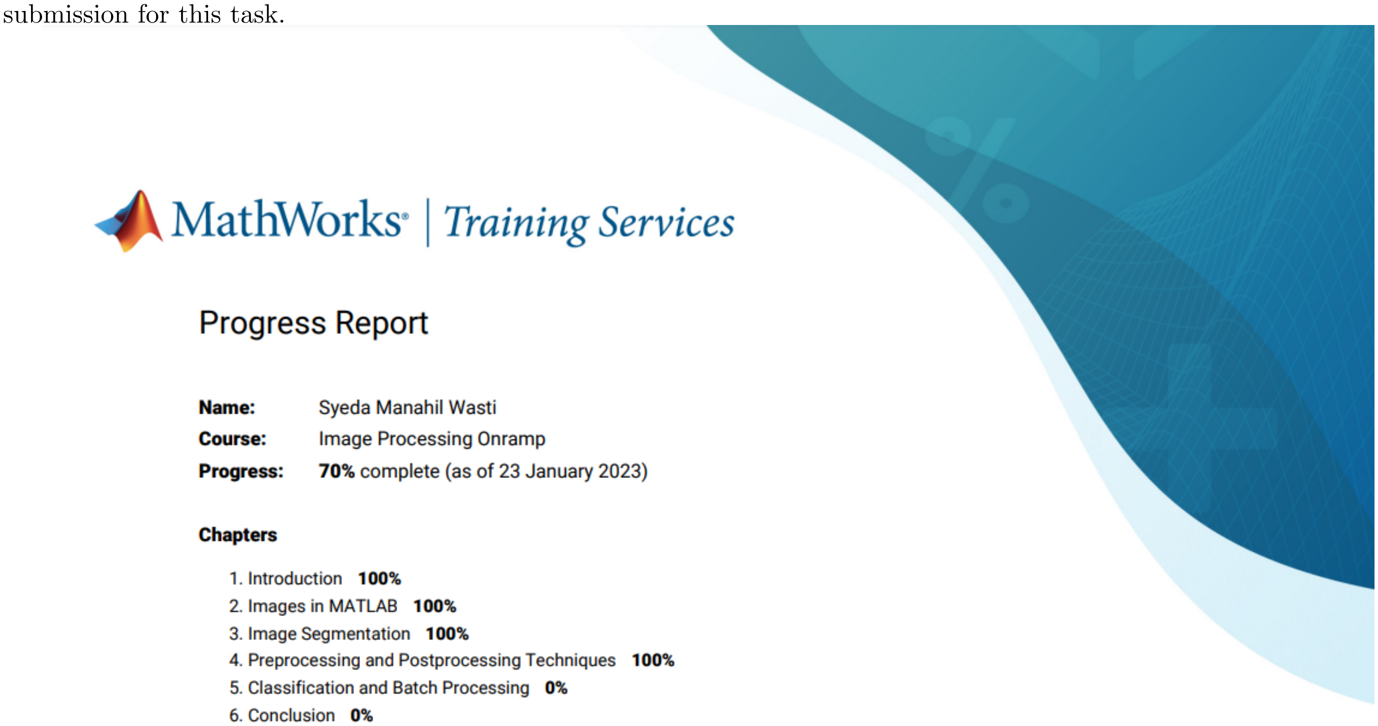
Task 3.3: Getting to know the camera

Download Intel RealSense Viewer tool from Canvas to verify that your camera is working and to explore the various parameters. If you enable both the RGB and depth streams, you shall see live videos for both where the depth stream represents different depths in different colors. Hover over any pixel in the depth image and you shall see the depth value in meters at the bottom. Explore the different processing filters available for each stream. Details of filters are provided at [6].



Task 3.4: Image Manipulation in MATLAB

Complete at least modules 1-4 of the 'Image Processing OnRamp' and provide your 'Progress Report' as submission for this task.



MathWorks | Training Services

Progress Report

Name: Syeda Manahil Wasti
Course: Image Processing Onramp
Progress: **70%** complete (as of 23 January 2023)

Chapters

1. Introduction **100%**
2. Images in MATLAB **100%**
3. Image Segmentation **100%**
4. Preprocessing and Postprocessing Techniques **100%**
5. Classification and Batch Processing **0%**
6. Conclusion **0%**

Task 3.5: Extract color and depth images

1. The MATLAB function depth_example(), located in the folder where the SDK is installed, provides the code to extract a depth image. Verify that the code works and make sense of the provided code.

   Modify this code so that it also extracts a color frame and displays it. You can do so by using the function get_color_frame(). Don't forget to reference the appropriate class instance and format the received frame before displaying.

Solution:
Modified code to extract colour frame:

```
function depth_example()
    % Make Pipeline object to manage streaming
    pipe = realsense.pipeline();
    % Make Colorizer object to prettify depth output
    colorizer = realsense.colorizer();

    % Start streaming on an arbitrary camera with default settings
    profile = pipe.start();

    % Get streaming device's name
    dev = profile.get_device();
    name = dev.get_info(realsense.camera_info.name);
```

```
    % Get frames. We discard the first couple to allow the camera
       time to settle
    for i = 1:5
        fs = pipe.wait_for_frames();
    end

    % Stop streaming
    pipe.stop();

    % Select depth frame
    color = fs.get_color_frame();

    % Get actual data and convert into a format imshow can use
    % (Color data arrives as [R, G, B, R, G, B, ...] vector)
    data = color.get_data();
    img = permute(reshape(data',[3,color.get_width(),color.
       get_height()]),[3 2 1]);

    % Display image
    imshow(img);
    title(sprintf("Colorized depth frame from %s", name));
end
```

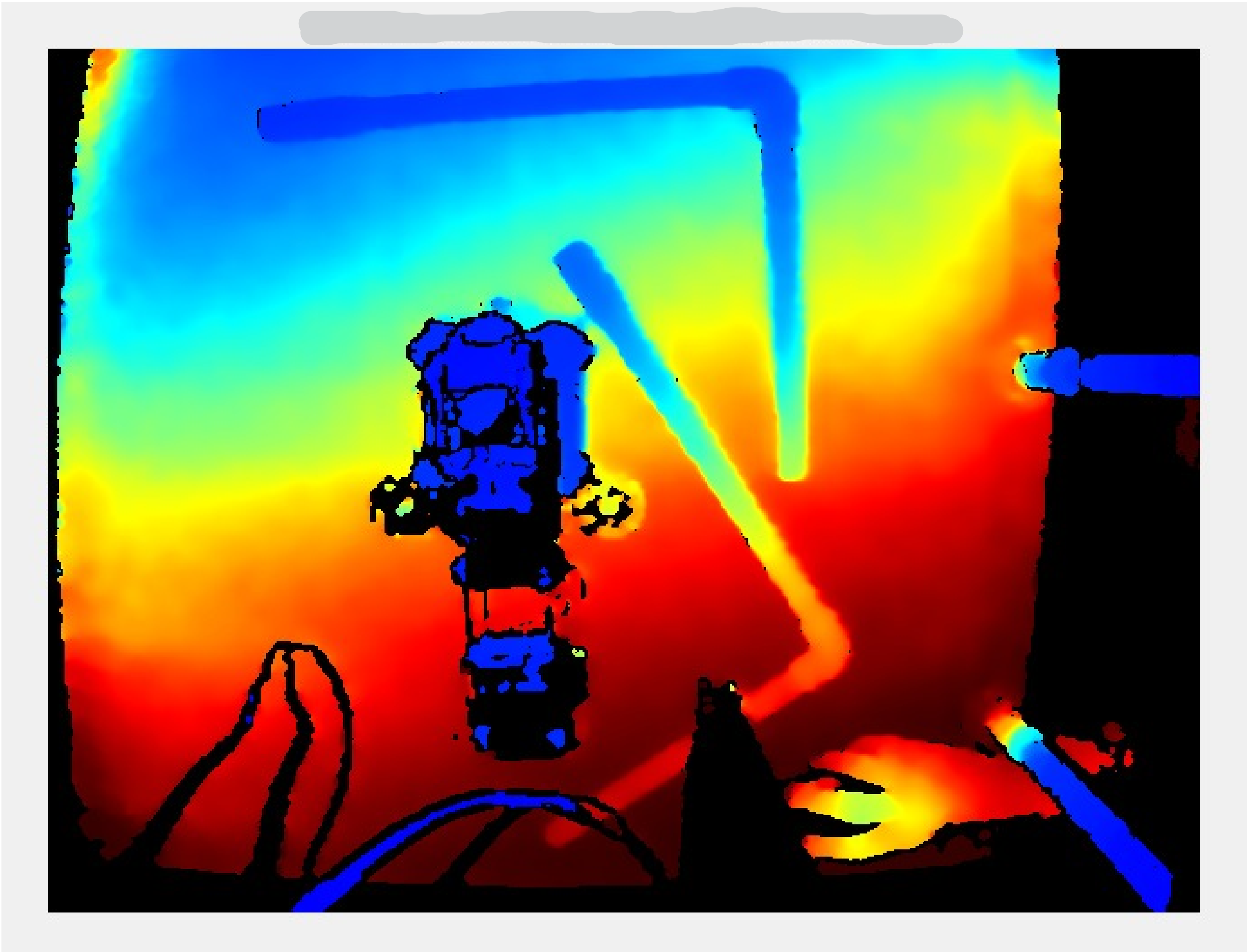


Figure 1: extracted depth image

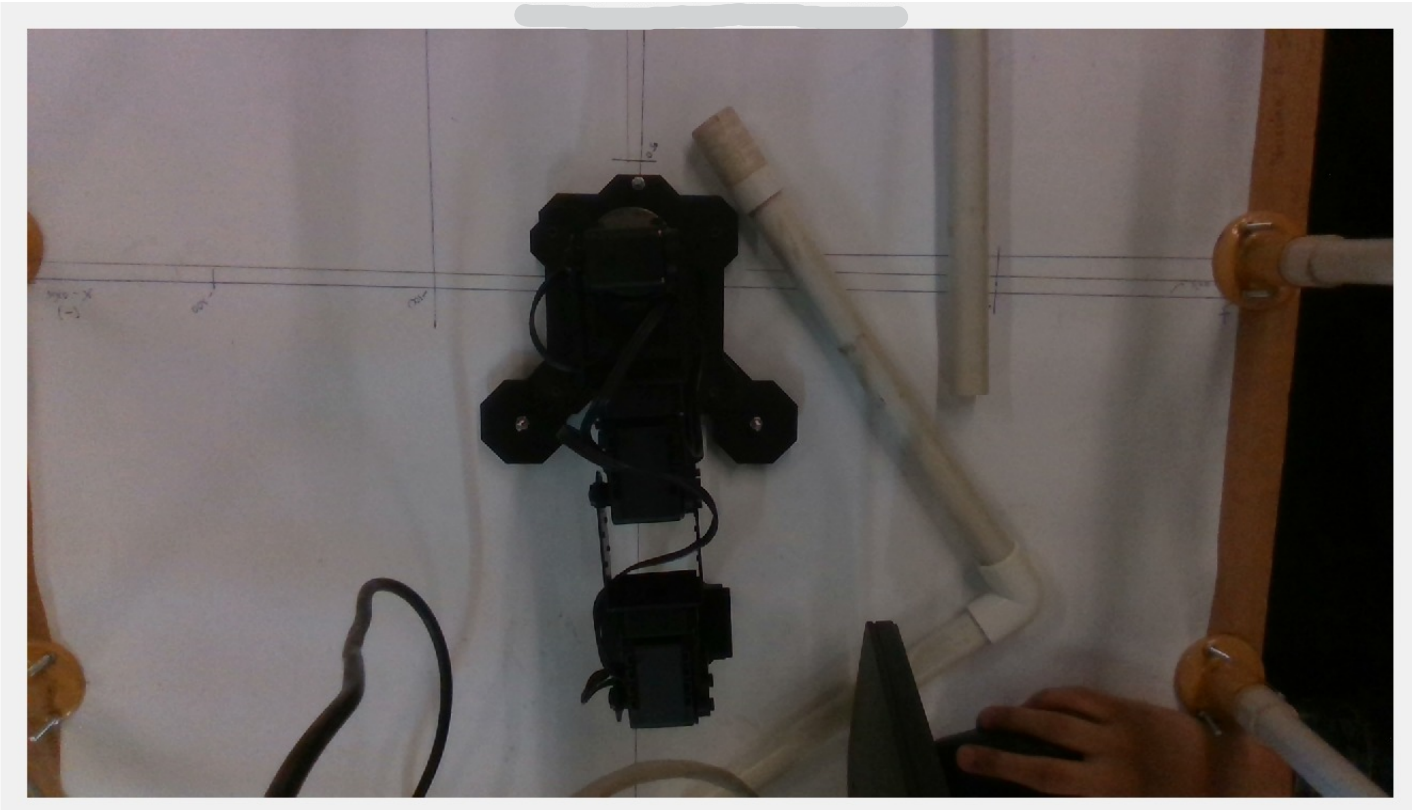


Figure 2: extracted color frame

Task 3.6: Reflection

1. Based on your experiences with the robot arm in the current and previous sessions, write a brief note outlining your takeaways and any unanswered questions about the vision-based pick-and-place robotic system that we intend to develop as our first project.

> **Solution:** Highly repeatable processes are used by vision-based pick-and-place robotic systems. In many pick-and-place applications, the arm must not only determine the object's location and orientation, but also its structure, as well as the optimum way to choose or place it.
> We observed that in our case, the arm had good repeatability. The sensor and the vision system will be the camera in our case which will identify the position of the object.

References

[1] W. Storr, "Position Sensor and Linear Positional Sensors," Basic Electronics Tutorials, Aug. 15, 2013.