Introduction to Robotics
Lab 8 - Completing the Perception Pipeline

Names: Hamad Abdul Razzaq & Muhammad Ali Siddiqui
IDs: hr06899 & ms06875

Lab Report

Task 8.1

Following is the required modified code:

```
function determineIntrinsics()
    % Make Pipeline object to manage streaming
    pipe = realsense.pipeline();

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

    % Extract the color stream
    color_stream = profile.get_stream(realsense.stream.color).as('video_stream_profile');
    depth_stream = profile.get_stream(realsense.stream.depth).as('video_stream_profile');

    % Get and display the intrinsics
    color_intrinsics = color_stream.get_intrinsics()
    depth_intrinsics = depth_stream.get_intrinsics()
    save lab_8
end
```

The Output is shown below:

```
color_intrinsics =

  struct with fields:

      width: 1920
     height: 1080
        ppx: 944.4568
        ppy: 533.8895
         fx: 1.3991e+03
         fy: 1.3991e+03
      model: 0
     coeffs: [0 0 0 0 0]

>> depth_intrinsics

depth_intrinsics =

  struct with fields:

      width: 640
     height: 480
        ppx: 313.6514
        ppy: 244.2748
         fx: 479.4813
         fy: 479.4813
      model: 2
     coeffs: [0.1295 0.1781 0.0055 0.0048 -0.1084]
```

The width and height determines the number of pixels horizontally and vertically in an image. ppx and ppy are the u_0 and v_0 respectively. and fx and fy corresponds to $m_x f$ and $m_y f$ respectively.

Task 8.2

The following code writes the transformation in standard format:

```
function determineExtrinsics()
    % Make Pipeline object to manage streaming
    pipe = realsense.pipeline();

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

    % Extract the color and depth streams
    color_stream = profile.get_stream(realsense.stream.color).as('video_stream_profile');
    depth_stream = profile.get_stream(realsense.stream.depth).as('video_stream_profile');

    % Get and display the intrinsics
    Tdc = depth_stream.get_extrinsics_to(color_stream);
    T = [Tdc.rotation(1:3) Tdc.translation(1); Tdc.rotation(4:6) Tdc.translation(2); ...
        Tdc.rotation(7:9) Tdc.translation(3); 0 0 0 1]
    Tdc = depth_stream.get_extrinsics_to(depth_stream);
    T = [Tdc.rotation(1:3) Tdc.translation(1); Tdc.rotation(4:6) Tdc.translation(2); ...
        Tdc.rotation(7:9) Tdc.translation(3); 0 0 0 1]

end
```

Task 8.3

Part (a)

Resolution of Color Camera: 1920×1080
Resolution of IR Camera: 640×480

Part (b)

Frame Rate for both the cameras: 10,30,60

Part (c)

Depth Field of View for IR Camera:
Horizontal: 69
Vertical: 54

Part (d)

Depth Start Point: 0.9mm from the front of lens

Task 8.4

Using the given formula and the values from the DataSheet, the height came out to be 82 cm. It was verified that the whole workspace was indeed visible when camera was placed at this height.

Task 8.5

The pixel coordinates were transformed to World coordinates with the helpd of the following code:

```
Z = 820;
K = [
    1399.1 1 944.4568 ;
    0 1399.1 533.8895;
    0 0 1
    ]
T_ = [
    1 0 0 -20;
    0 cos(pi) -sin(pi) 0;
    0 sin(pi) cos(pi) 688;
    ];
X = inv(K * T_) * [u;v;1]*Z;
```

The Link to the demo can be found here.

Performance of the System

Although the system did worked for one of the tests, there are still a lot of issues in picking up the Object specifically. The correctness of our system can be determined by the fact that the Manipulator did got closer to object every time we tried picking it up, but not exactly above the center of the cube. There could be many reasons for this error of which some are as follows:

- Center Determiniation: The Center pixel determination may not always be correct from Lab 7 and can contain errors
- Transformation from Pixel coordinates to World Frame. Due to human error in determining the distance from camera to world frame, some error can also occur while performing transformation from Camera to World Frame