Lab 11
Camera Pixels to Angles

Introduction

Now that we have learned how to acquire images and use color to detect and identify objects of interest, we are well on our way toward building image processing logic to automatically identify red and green buoys (and turtles and sharks!). In this lab we will learn how to convert the objects we identify on an image into a sensor output that our AUV controller can use to help guide its decision making.

We will focus our initial work on the picture below, which was taken by the Raspberry Pi V2 Camera. In this image, we can see two objects that are important for navigation in desert ocean regions like the Salton Sea: a cactus and a water bottle. With these two objects, we will learn to generate reports of angles in a coordinate convention of degrees clockwise from the look direction, or center of the camera. When our camera is mounted with a forward looking direction on the AUV, this choice of coordinates will align with maritime (and BWSI AUV course) convention of clockwise degrees from forward.

Once we have learned to convert detected objects in the image into angles, we will put it into practice with our own Raspberry Pis. At the same time, we will measure the true angles to the objects in order to calibrate our cameras. We will then be prepared to take imagery from our Pis and convert them into actionable sensor outputs for our AUV controller.

Calculate the position of calibration points in the sensor coordinate system

Take a look at the cactus and water bottle picture. What process do you go through to identify the two objects in the image? A key step in this process is to somehow make a determination that a group of pixels all go together as part of each object in the picture. These pixels can then be thought of as being on the same local coordinate frame - maybe you think of the object as a height off the ground or as spreading from a point of attachment. The choice of this reference "anchor" point is arbitrary, but having one is necessary for calculating relative distances on the object.

1) We first need to identify some anchor points that we will use for our calibration. Later in the course, we will cover methods to choose anchor points, but for now you can use the ones I picked: (1750, 950) and (2870, 1050). I plotted them as a blue and

red dot on top of the image:

2) Write python code to read in the cactus image with cv2 and plot the same anchor points that I did. Check that your dots are showing up in the same place. Also, note the location of the point (0,0) in the pixel array. Sadly, (0,0) is not in the center of the image. Remember that in our angle output, 0 degrees will be in the center of the array.
3) The camera imaging sensor, where the image is formed, is 3.68mm (x) x 2.76 mm (y). Write a function that converts from pixel position to measurement units (meters) on the camera imaging sensor. For convenience, also shift the origin of the camera coordinate system to the center of the sensor as opposed to the origin in the pixel frame that you observed in step 2. We will refer to this point as the camera coordinate system origin (CCSO).

def sensor_position(pix_x, pix_y, res_x, res_y)

....

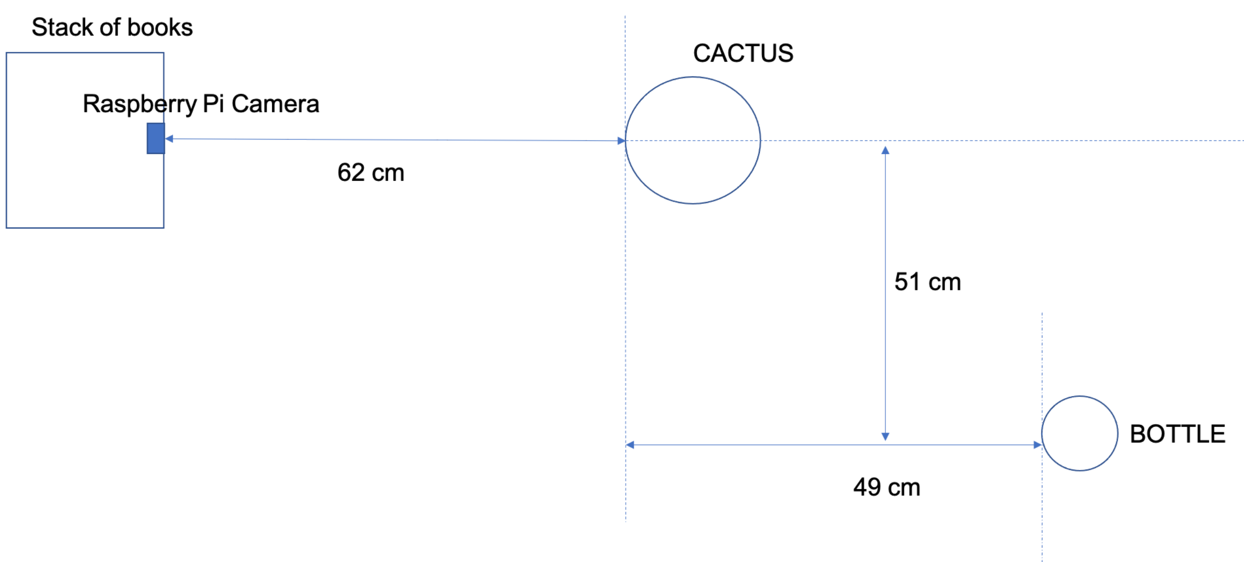
return (sensor_pos_x, sensor_pos_y)

4) Use your new function sensor_position to determine the (x, y) location of the blue anchor point and the red anchor point in the camera sensor plane. Estimate the size of the two objects on the camera imaging sensor.

Calculate angles from CCSO to points in image

5) Using the rectilinear mapping $r = f tan\theta$, write another function that returns both the vertical and horizontal angles to a point in the image relative to the CCSO.
6) With your new mapping function, calculate the vertical and horizontal angles from the CCSO to the blue dot and red dot. You should end up with 4 numbers. Do they make sense?
7) Now consider the diagram below, which is a bird's-eye view of the photographed scene.
    a) What is the relative horizontal angle between the blue dot and red dot as shown in this diagram?
    b) How does this angle compare to the one you calculated from the anchor points on the image?

Warning: Drawing not to scale



Calibration with objects of known size

Now we have determined the relative bearings of the two objects, what can we say about their relative size? Not much, as it turns out - the size of the object at the imaging sensor is a product of both its *actual* size and its distance from the camera. With knowledge of one of these pieces of information, we can infer the other. In our navigation challenge we will be able to assume that all of the buoys are of the same size, so we can use that knowledge to our advantage in determining the relative distance to the buoys in imagery. We will test this approach using our cactus and water bottle image.

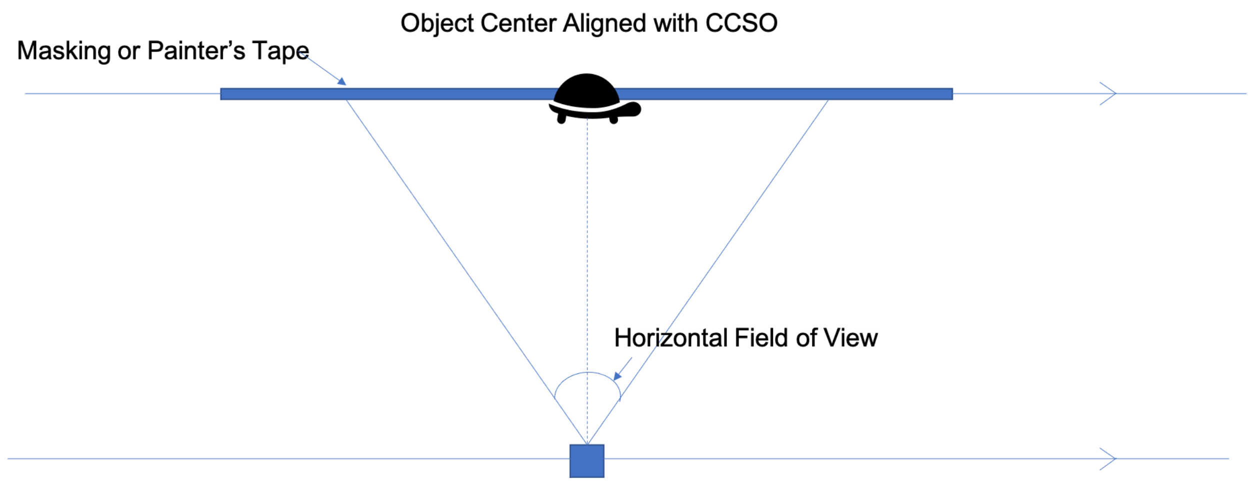
8) Using the information you have so far and the fact that the height of the blue dot is 18 cm, can you determine the height of the CCSO from the floor?
9) Which is taller, the cactus or the water bottle?

Finding Angles from the Raspberry Pi Camera

Now that we understand how to calculate angles and estimate distances on the image provided, let's transition this knowledge to imagery collected by your own Raspberry Pi camera. While we're at it we will develop an understanding of the field of view of the camera and estimate any distortions or aberrations along the horizontal axis.

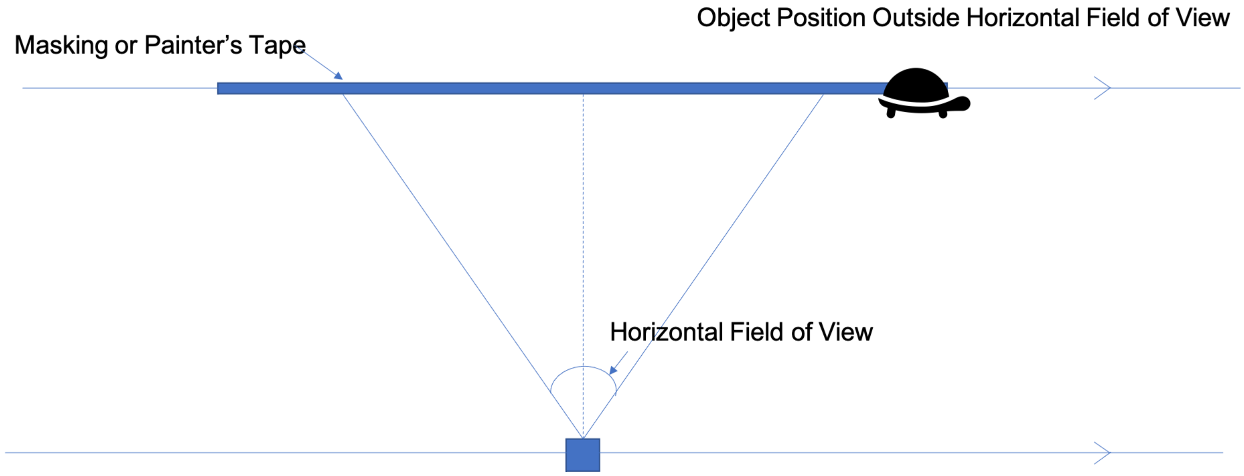
For this section, you will need one or more measuring apparatus (note that the plural form of "apparatus" is "apparatus"). A tape measure is a good choice, but you can be creative and use an object of known length, such as a foot, water bottle, pen or all of the above. It also might be helpful to use some masking or painter's tape to lay out a line of reference. If you don't have those, consider doing this against a wall or along a carpet edge.

10) Find an object and center it on the camera's origin as closely as possible, both in the vertical and horizontal planes



11) Move the object along a line parallel to the camera's horizontal axis and take a series of photos. Use a tape measure to measure the distance along the line and calculate the horizontal angle of your object in each photo. Compare the calculation done with your tape measure measurements to the angle you compute with your camera functions.

12) At what tape-measured horizontal angle can you no longer see the object in the picture? How does that compare to the camera's horizontal field of view specification?


Object Position Outside Horizontal Field of View
Masking or Painter's Tape
Horizontal Field of View