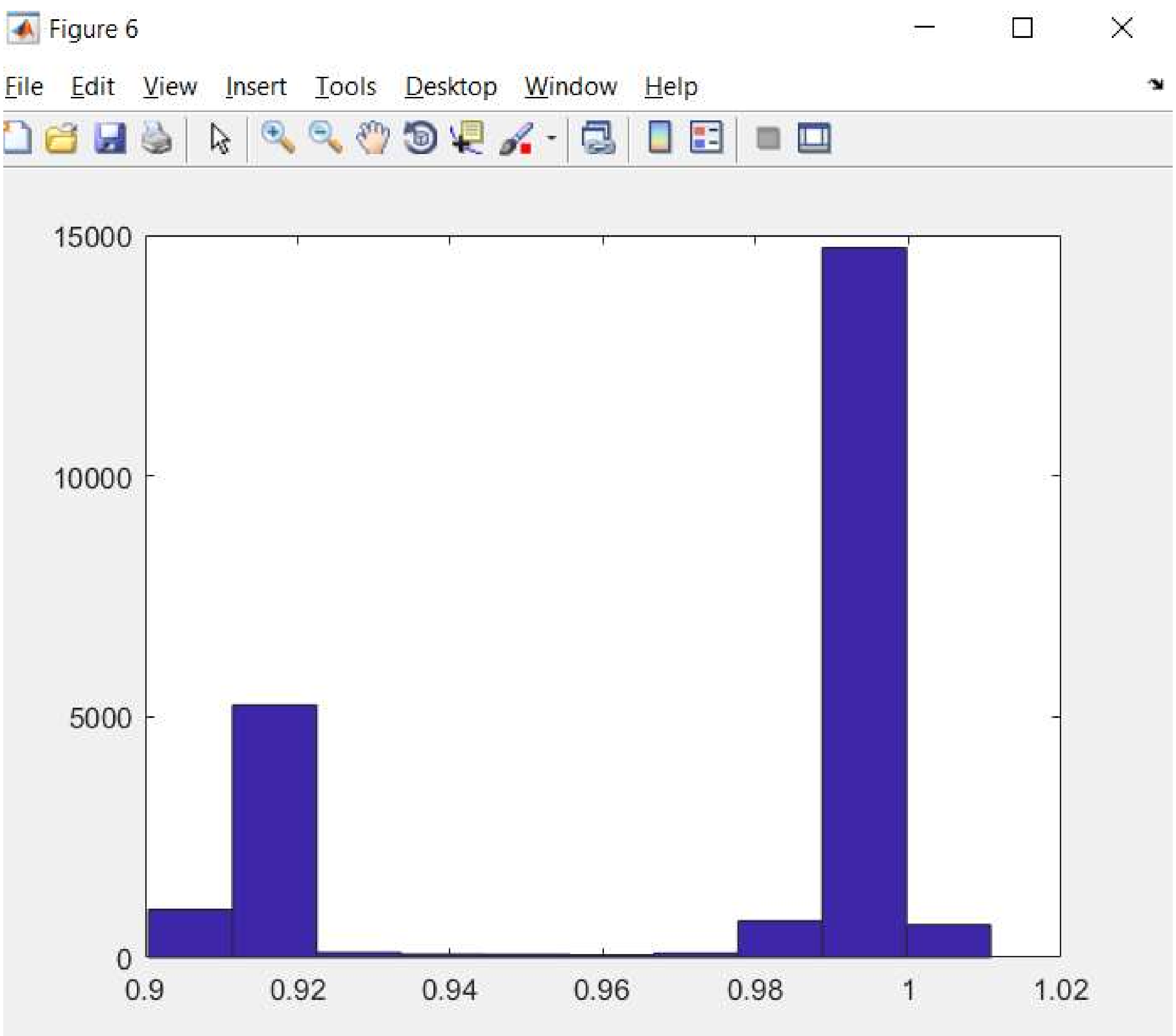
Questions

Explain why the projective model of Eq.(1) is representative of the imaging geometry of Figure-1?

Because It is x and y represent the digital pixel locations it is scaled by Zw which is the depth of the image per pixel the world coordinates when captured is also distorted after passing through the intrinsic matrix of the camera K. Thus the world coordinates needed to be scaled down by 1/Zw and the intrinsic camera matrix K.

How many peaks do you see in the histogram of depths within the region-of-interest?

I saw two main peaks below is an image of the histogram



Does the histogram represent a bimodal distribution?

Yes there is only two main distributions around 0.917 and 0.994

What do each of the peaks tell you about the depth Z_w of the box and the depth of the conveyor belt from the camera?

My histogram calculates the most often occurrence of certain values within the depth map. Each value at each pixel location represents the Z_W of that pixel. Therefore the two peaks in my histogram represents the distinction between two elevations in this case the depth of the box and the depth of the conveyer belt. More specifically the 0.994(meters) peak represents the depth of the conveyer belt, the 0.917(meters) peak represents the depth of the box these distances represents the distance away from the camera.

What is your estimate of Z_w for points on the box and points on the conveyor belt (express in centimeters)?

```
Zbelt =

    0.9942


Zbox =

    0.9170
```

These estimates are in meters their actual values would be 99.42cm for the belt and 91.70cm for the box.

What is your estimate of the height of the box (express in centimeters)?

7.772 cm

What is your estimate of the length of the box (express in centimeters)?

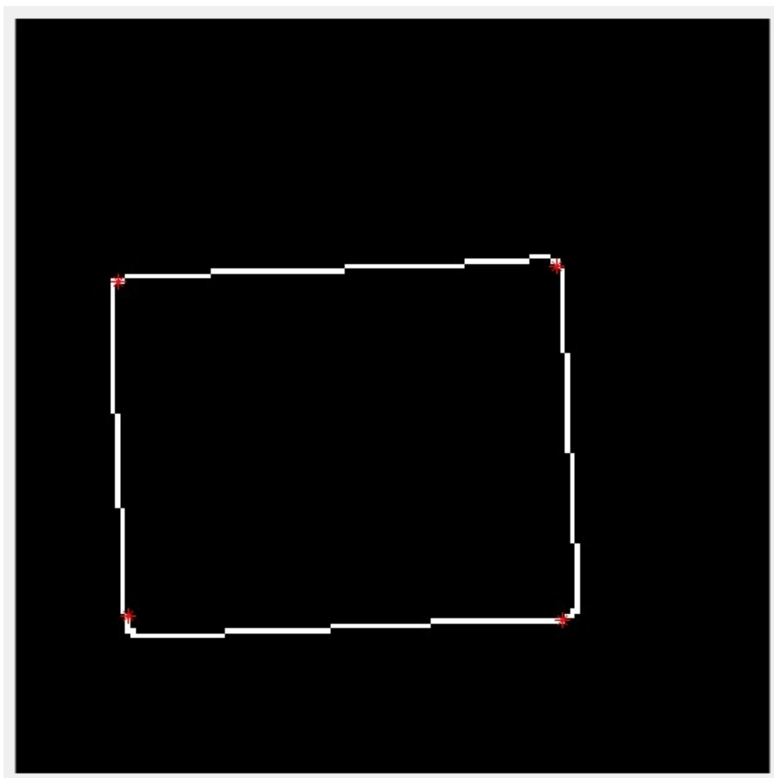
28.18 cm

What is your estimate of the width of the box (express in centimeters)?

35.58 cm

Explanation

My strategy for selecting the corners of the box is simple first I detected the edges of the box on the depth map using the edge function. I first had to pass this function a region of interest of the given depth map this was to weed out some outliers surrounding the edges of the depth map. Then I had to select which method I used for edge detection in which I selected a Canny edge detection method since it supported non maximum suppression. I ran the code and realized I was still getting too many extraneous edge fragments. What I wanted from this edge detector is simple but clear outline of the box so that's why needed to raise the threshold values to a point where I was only getting the outline of the box. I found that a threshold value set to 0.95 worked just fine for this given image, this threshold will need further tuning in the future but for now it works and the result from the edge map is shown below:



Things to note that this edge map only represents the region of interest the edge function was given which is why it's dimensions are much smaller. The red point markers indicate where the corners are I found these using the MATLAB function corners. I kept the parameters of this function to its default settings which it used a Harris corner detector. The only parameters I passed this second function was the output from the edge function which only has the outline of the box therefore it should only have 4 corners. This led me to input the N parameters which specified the maximum number of corners to return which should be 4 regardless in this case. Once I calculated the corner pixels, I overlaid the corner markers onto the edge map. Thus, this automates the process in which the corners are detected.

Now that I have the pixel index of the corners extracted, I converted all of them into world coordinates using equation (1) given in the assignment:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{Z_w} K \begin{bmatrix} X_w \\ Y_W \\ Z_w \end{bmatrix}$$

I simply deconstructed the matrices into two separate equations to solve for Xw and Yw the process starts by moving Zw to the left side:

$$Z_w \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K \begin{bmatrix} X_w \\ Y_W \\ Z_w \end{bmatrix}$$

The values of the intrinsic matrix are known so I substitute the values in and multiplied the values out:

$$\begin{bmatrix} Z_w * x \\ Z_w * y \\ Z_w * 1 \end{bmatrix} = \begin{matrix} 225.607 * X_w & 0 * Y_W & 158.205 * Z_w \\ 0 * X_w & 224.547 * Y_W & 118.488 * Z_w \\ 0 * X_w & 0 * Y_W & 1 * Z_w \end{matrix}$$

Taking the first two rows I solved for Xw and Yw:

$$Z_w * x = 225.607 * X_w + 158.205 * Z_w$$

$$Z_w * y = 224.547 * Y_W + 118.488 * Z_w$$

Apply simple algebra:

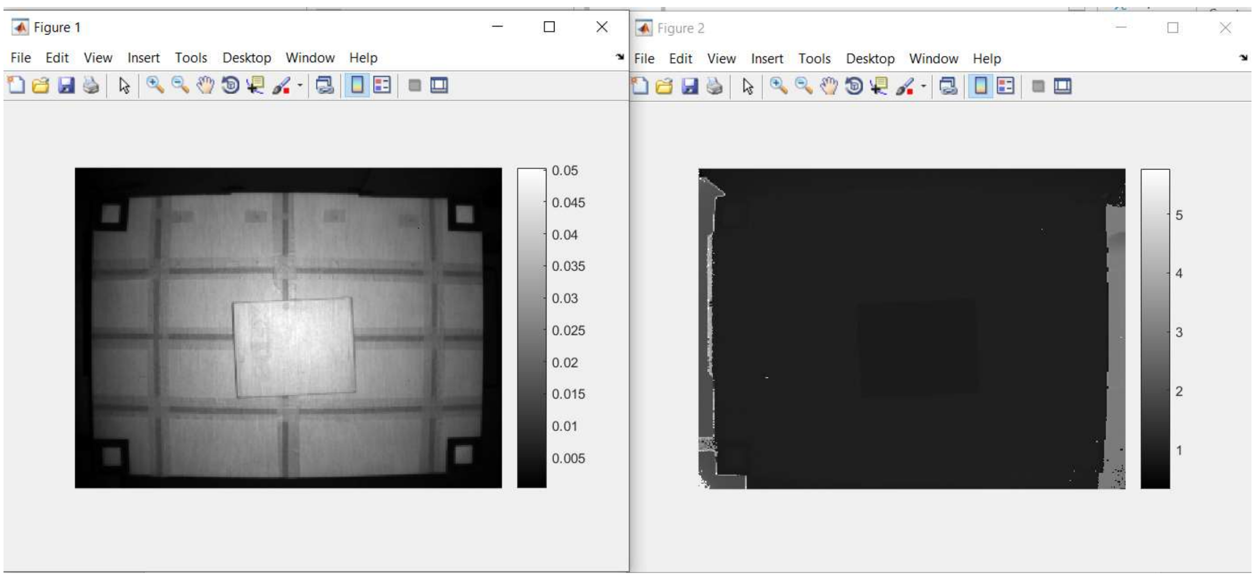
$$\frac{Z_w(x - 158.205)}{225.607} = X_w$$

$$\frac{Z_w(y - 118.488)}{224.547} = Y_w$$

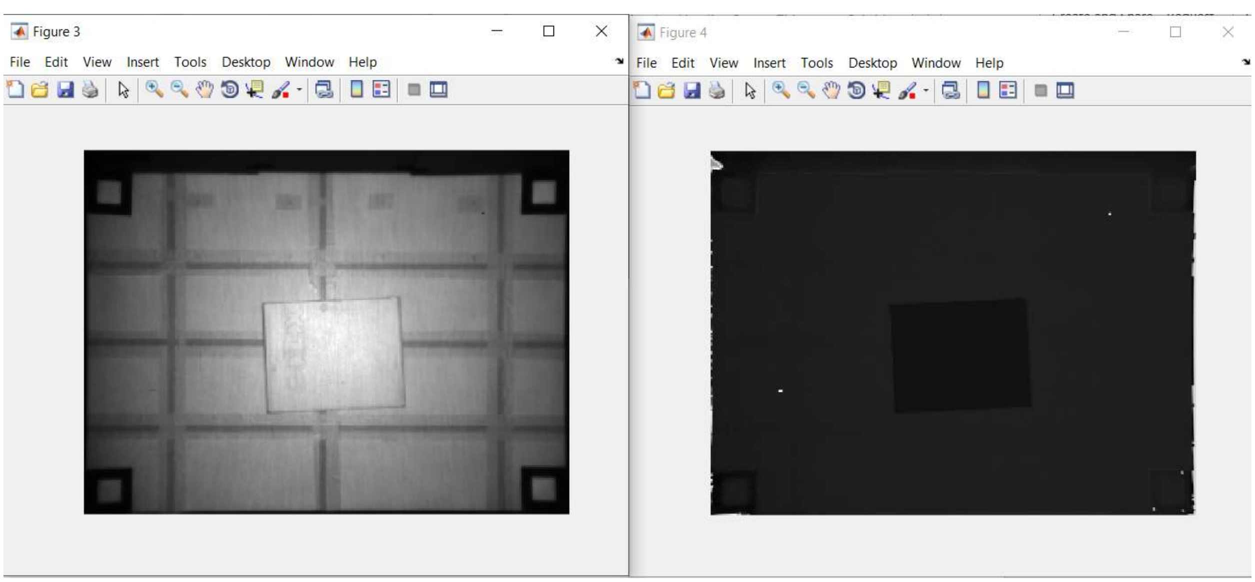
The Zw in these equations is the depth that these pixels are located which is the depth of the boxes as these pixels lie upon the corners of the box. These equations will then convert x and y in pixel coordinates to world coordinates I chose to just solve these matrices by hand since I did not wish to mess with potentially transposed intrinsic matrices stored in the camera parameters. After I converted the pixels to world coordinates, I calculated the Euclidean distance between the first two set of coordinates then I calculated the Euclidean distance between the last two set of coordinates and calculated the average of these two distances this became my horizontal dimension estimate. For the vertical dimension estimate I calculated the Euclidean distance between the 2nd and 4th sets of coordinates and the 1st and 3rd sets of coordinates. Then I averaged the distances together and named it the vertical dimensions. While they are labeled as left right top and bottom in my code that was only to help me keep track which how to type the Euclidean equations. For any other box orientation this method should still world since the corner detection function outputs the points from left to right and top to bottom and since I am calculating Euclidean distances negative values are squared summed and the square root is calculated. Finally, the height of the boxes is calculated by taking the difference between the depth of the conveyer belt and the depth of the box. These values were calculated from the histogram of the region of interest. The region of interest is an index section of the depth map. This was then flattened into a 1-dimensional vector and fed into a histogram function where the count and centers of each of the histogram is stored. The index where the counter was highest was found and stored and the center value at these indexes are the depth values of the box and conveyer belt.

Images:

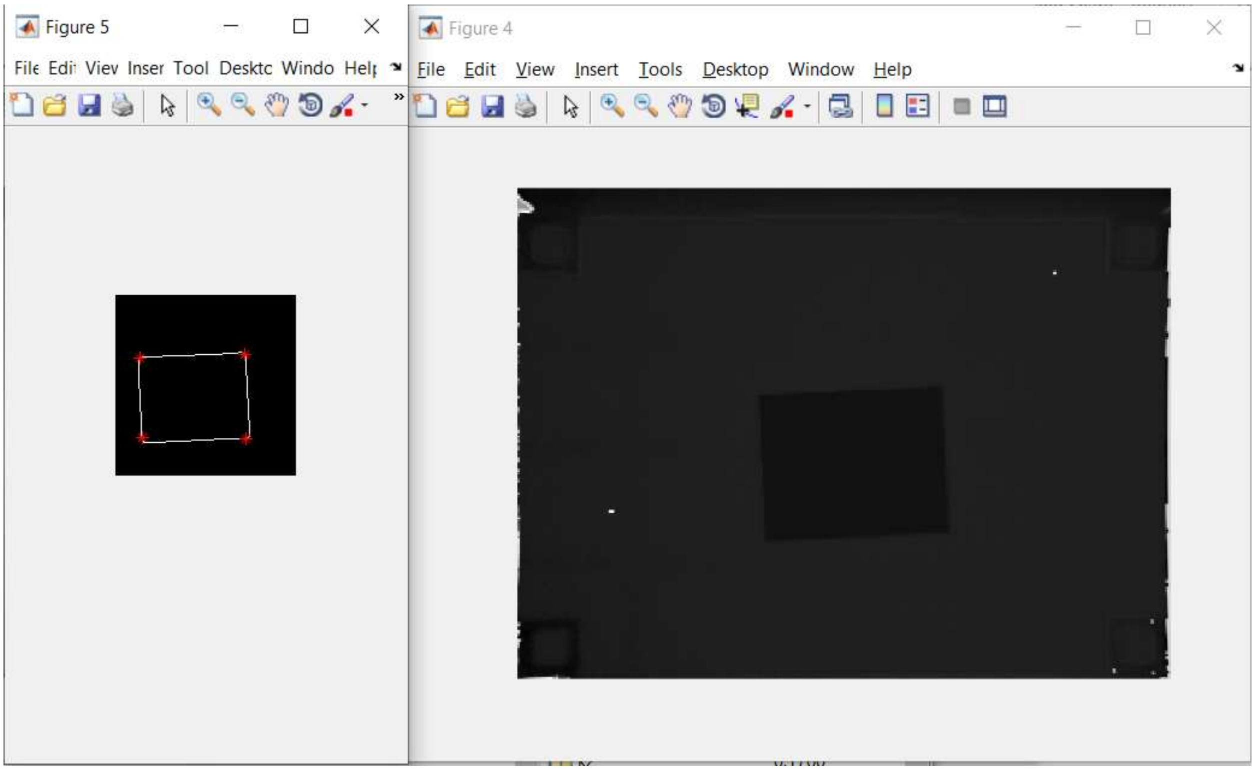
Distorted Intensity and depth maps



Undistorted Intensity and depth maps



Region of interest edge map with corners marked size comparison



Resources used:

Edge Function built in MATLAB Function

https://www.mathworks.com/help/images/ref/edge.html

Corner Function built in MATLAB Function

https://www.mathworks.com/help/images/ref/corner.html?s_tid=doc_ta#d117e39663

Histogram Function built in MATLAB Function

https://www.mathworks.com/help/matlab/ref/hist.html