



# ECE 6310 – INTRODUCTION TO COMPUTER VISION

Lab 8 – Range Image Segmentation

C14109603

Rahil Modi  
rmodi@g.clemson.edu

## Objective:

In this lab we were expected to segment an image with a chair based on the grayscale image of the same provided to us for different surfaces in the image.

## Implementation:

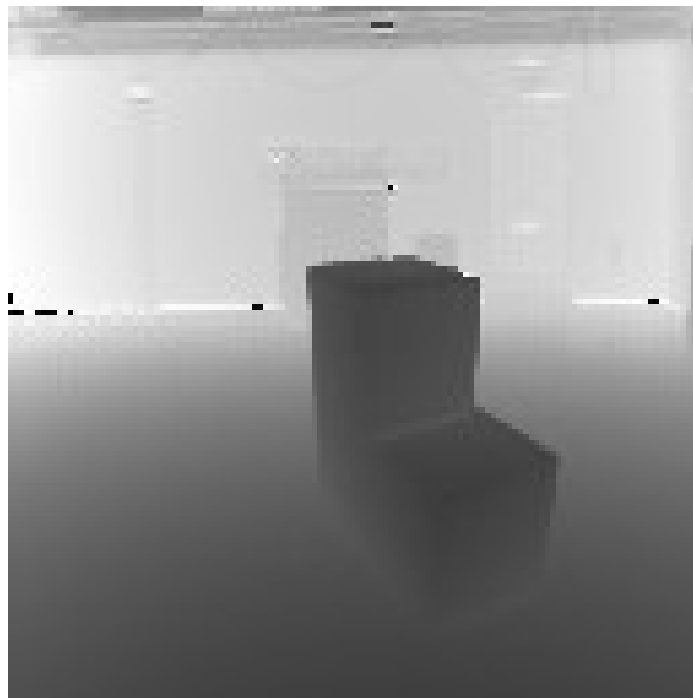
To begin with, the grayscale image was first threshold to a particular value to remove the background and only keep the surfaces which were required like the ground and the chair. The threshold value I used was **137**, in which any pixel value above 137 was set to 255 and rest of the pixel were left as they were this way the background of the image was removed and only ground and the chair were visible.

Now the X, Y, and Z coordinates were calculated with the help of the base code provided to us and using the grayscale image as the input to it and it was stored in the array of structs. The code had details specific to the camera used to click the picture through which it was possible to calculate the X, Y, and Z coordinates. The surface normal calculation was done after this with the help of the X, Y, and Z values. It was done by looping through every pixel and looking at the pixel to its right and below the current pixel. The surface normal was calculated by subtracting the right pixel from the current pixel and the pixel below from the current pixel and the cross product of the two vectors was done to get the surface normal. The distance was chosen as **3 pixels in the right direction and 3 pixels below the current pixel**.

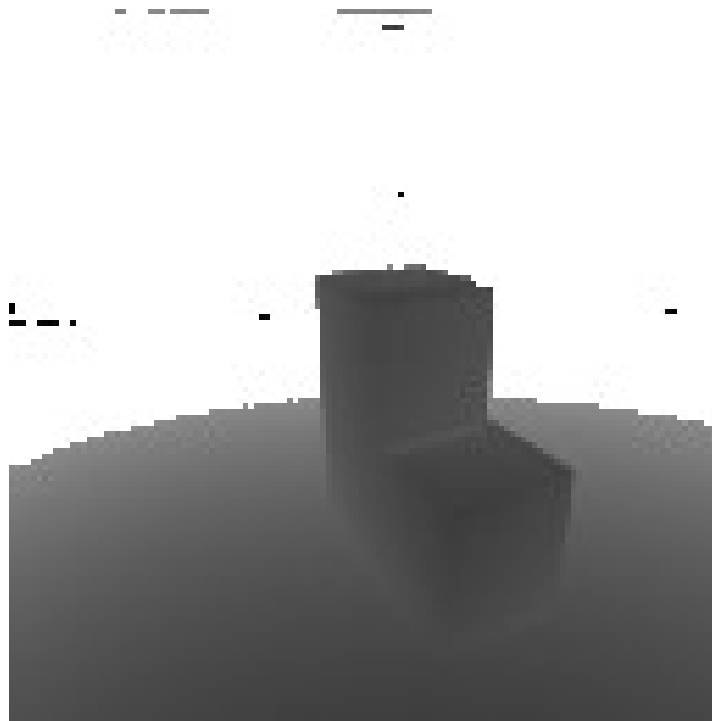
Now the region growing process was done. In a 5x5 window every pixel was checked whether if the pixel were removed in thresholding or already there is no pixel written over it then the region growing would start. The code was the same which we used in the earlier lab the part that was modified was the predicate to join the region. The predicate was to join the region only if the angle between the current surface normal and average surface normal was within a threshold. The threshold I selected for this is **0.72**. The angle value was calculated by taking the magnitude of the average vector and the magnitude of the current vector and taking the dot product between both the products. The value is in radians that is why the value is not in degrees. If the value was within the threshold then the region was grown. If the region was lesser than 100 pixels I have omitted those regions. The grayscale and the colored image of the region grown is shown in the results section.

## Results:

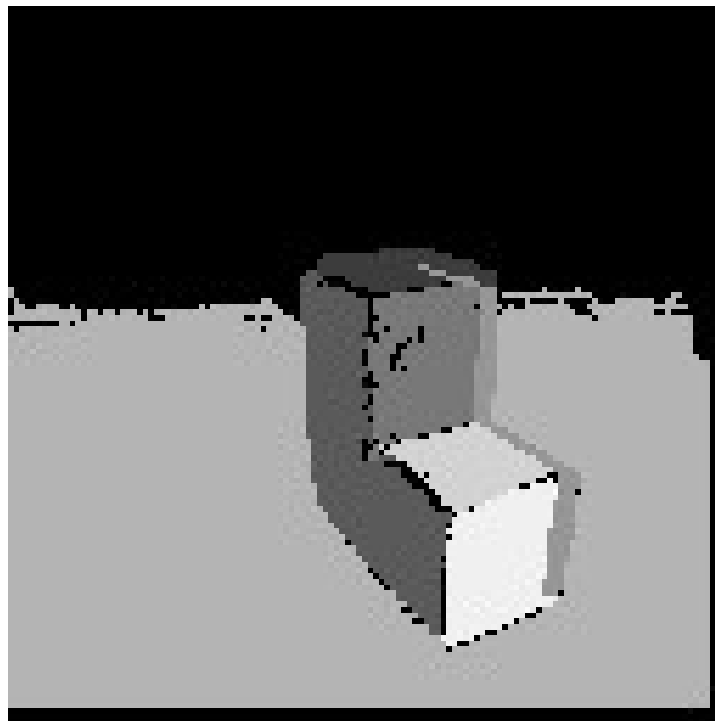
<i>Region Number</i>	<i>X Average</i>	<i>Y Average</i>	<i>Z Average</i>	<i>Number of Pixels</i>
<i>Region Removed</i>	0.112017	1.193703	-4.314375	3
1	-6.487198	-331.474135	-58.264375	165
2	-51.327779	-0.914586	-8.544396	763
3	2.662467	2.420322	-4.482366	466
4	103.44437	2.337080	-27.563593	207
5	-1.618351	-28.782052	-8.864264	6801
6	-1.050807	-8.514511	-2.389239	254
7	2.750688	1.627939	-4.829699	423



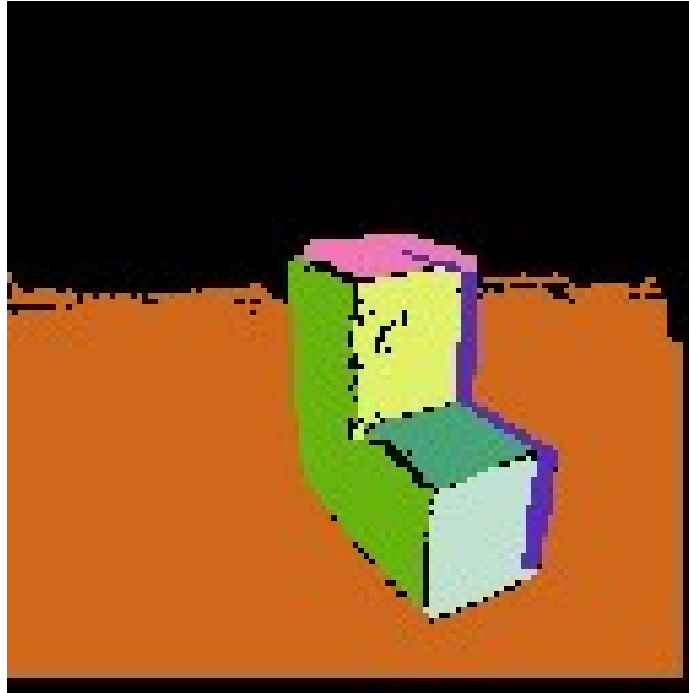
*Figure 1 The input range Image*



*Figure 2 Threshold Image at 137*



*Figure 3 Grayscale output*



*Figure 4 Colored output*

The output result was not the best as it missed pixels in between the surfaces but still it was able to identify all the surfaces. After trying different thresholds, the one I have mentioned above were the final ones I used as those showed the best output.