ECE 6310 INTRO TO COMPUTER VISION

Lab 8 Range Image Segmentation

MOHAMMAD ANAS IMAM KHAN
C17566828

DECEMBER 3, 2020

Lab 8 Range Image Segmentation

In this project, we had to segment a range image based upon surface normals. A range image of a chair as shown below was provided. The segmentation process used the image grid for grouping pixels. 3D coordinates were used for calculating surface normals for region predicates.

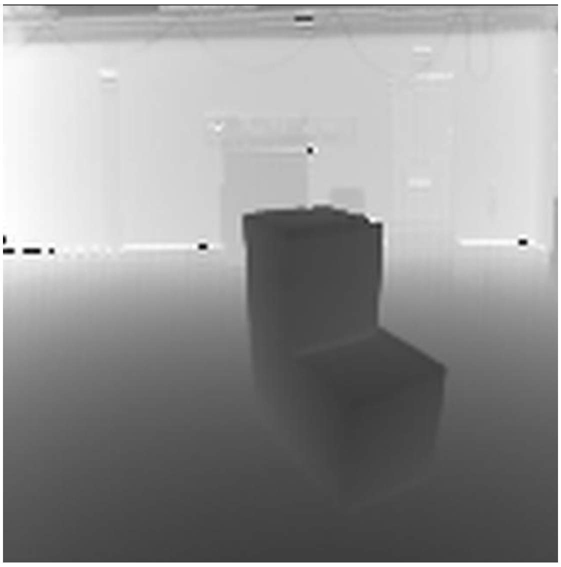


Figure 1 Chair Range Image

The image was first masked by thresholding at a distance that removed the background and left only the floor and the chair. A **threshold grayscale value of 125** was used to remove the background.

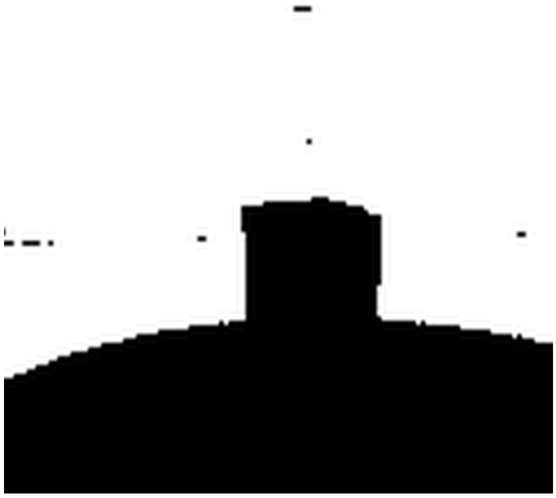
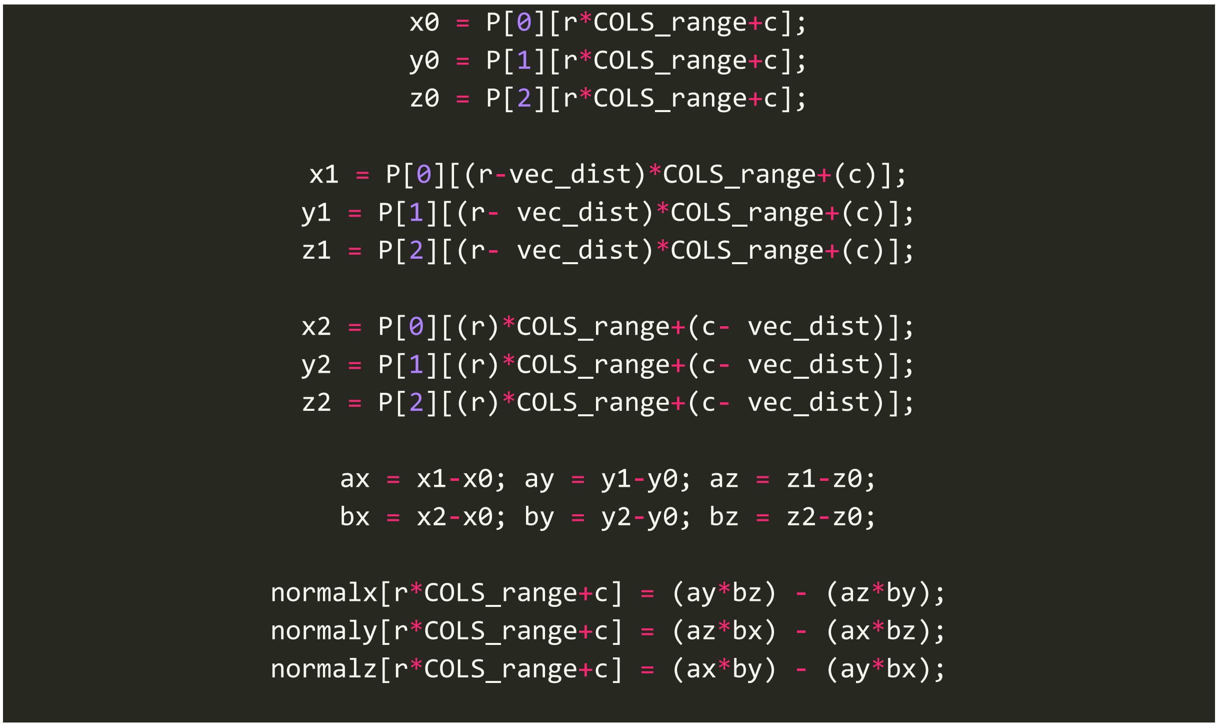


Figure 2 Threshold Image

The surface normal were calculated using the vector cross product formula. **The distance chosen between the pixels was 3.** Only those pixel's cross-product was calculated which had a value of 0 or lied within the threshold in the above image. The code stub for calculation of cross product is shown below.



```
x0 = P[0][r*COLS_range+c];
y0 = P[1][r*COLS_range+c];
z0 = P[2][r*COLS_range+c];

x1 = P[0][(r-vec_dist)*COLS_range+(c)];
y1 = P[1][(r- vec_dist)*COLS_range+(c)];
z1 = P[2][(r- vec_dist)*COLS_range+(c)];

x2 = P[0][(r)*COLS_range+(c- vec_dist)];
y2 = P[1][(r)*COLS_range+(c- vec_dist)];
z2 = P[2][(r)*COLS_range+(c- vec_dist)];

ax = x1-x0; ay = y1-y0; az = z1-z0;
bx = x2-x0; by = y2-y0; bz = z2-z0;

normalx[r*COLS_range+c] = (ay*bz) - (az*by);
normaly[r*COLS_range+c] = (az*bx) - (ax*bz);
normalz[r*COLS_range+c] = (ax*by) - (ay*bx);
```

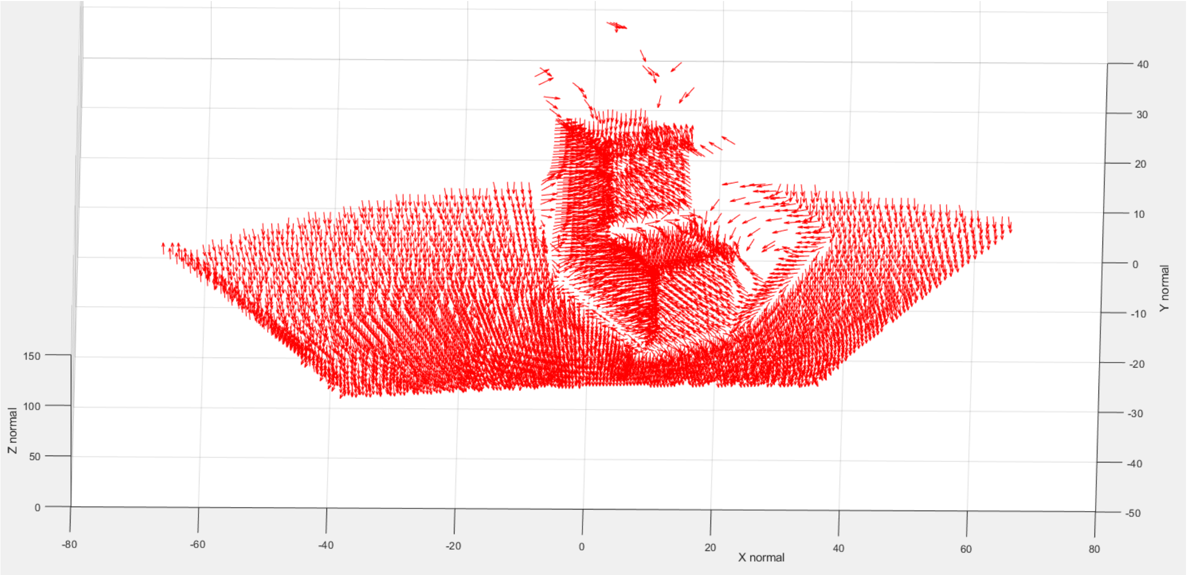


Figure 3 Surface Normals

The region predicate was such that a pixel can join the region if its orientation is within a threshold of the average orientation of pixels already in the region. The angular difference was calculated using the dot product. The orientation of only those pixels was computed which was withing the threshold image shown above and also did not have zero surface normal. A running average of each pixel within the 5*5 window was computed. This dot product of the average normals $\left(a_x, a_y, a_z\right)$ was computed with the normal of the neighboring pixel $\left(b_x, b_y, b_z\right)$. The magnitude of the average surface normals (mag_a) and the neighboring pixel (mag_b) was computed. Using the dot product and the magnitude of the surface normals the average orientation was computed as:

$$\cos^{-1}(\boldsymbol{dot}/(\boldsymbol{mag_a} * \boldsymbol{mag_b})$$

$$\boldsymbol{mag} = \sqrt{x_{normal}^2 + y_{normal}^2 + z_{normal}^2}$$

$$\boldsymbol{dot} = a_x * b_x + a_y * b_y + a_z * b_z$$

The angle threshold chosen was 67°.

The segmented image and the number of pixels along with the average surface normals have been shown below.

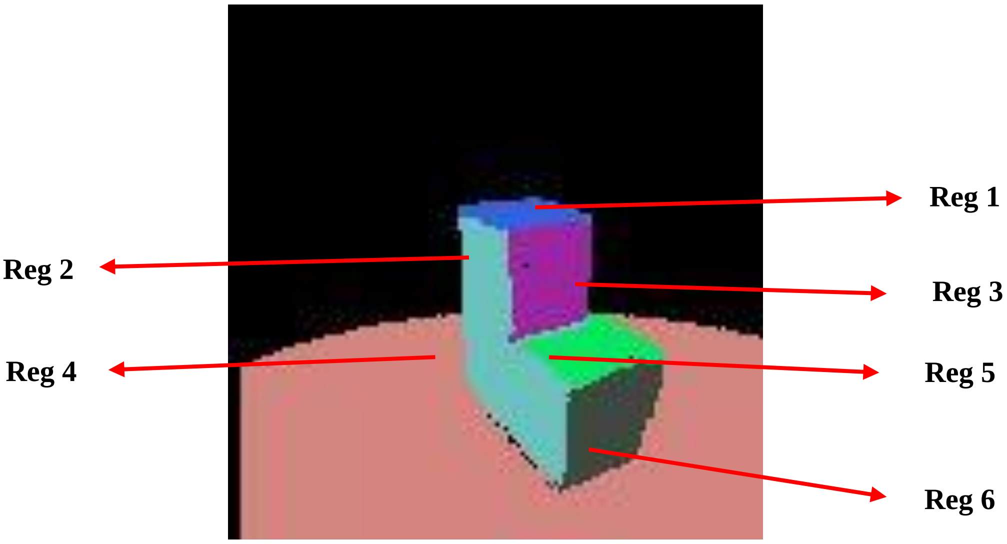


Figure 4 Segmented Image

Region no	Region Color	Image Location
1	Dark Blue	Top Surface of Chair
2	Light Blue	Left face of chair
3	Dark Pink	Top front face of chair
4	Peach	Ground
5	Light Green	Seating portion of chair
6	Grey	Front face at bottom of chair

Region number	**No of Pixels labeled**	**Average xNormal**	**Average yNormal**	**Average zNormal**
1	168	2.640505	−267.957343	47.840207
2	856	147.628572	3.878004	15.430773
3	475	−2.955761	0.412259	4.748544
4	4928	0.942606	−23.560746	7.730331
5	315	4.700335	−74.326480	−2.702088
6	478	−2.926740	4.092385	5.368893