ECE 6310: Introduction to Computer Vision - Lab 8 Report
Range Image Segmentation

Objective: To segment a range image based upon surface normals

Implementation and steps:

1. Original range image of chair and reflectance image was provided for the lab. The code for odetics to 3D coordinates conversion was also given.

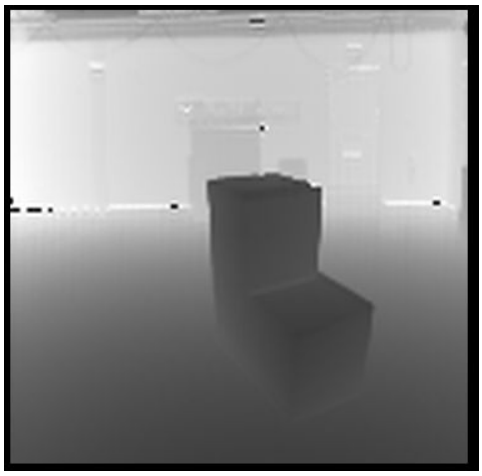

Fig. 1 chair-range.ppm

Fig. 2 chair-reflectance.ppm

2. The range image was first masked by thresholding at a distance that removes the background and leaves only the floor and the chair. The intensity threshold I chose was 126.

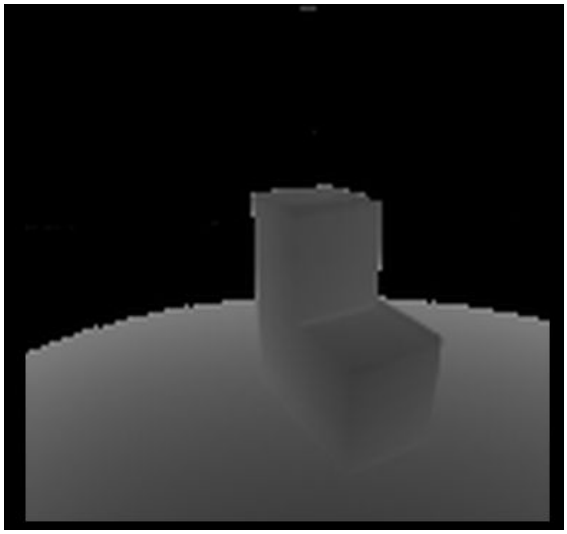

Fig 3. Thresholded image

3. Used the provided C-code to calculate 3D coordinates for the pixels. The slant type was assumed to be scan-direction downward.

4. Surface normals were calculated by taking the cross product method discussed in class. The distance between pixels for cross product was 3 pixels. The surface normal at pixel X was calculated by taking the cross product of (B-X) x (A-X), where A, B and X were the 3D coordinates of those pixels. The distances B-X and A-X were selected as 3.

4. Region growing was used to segment regions, using the queue-based C code previously provided. The region predicate was 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 region growing code was modified to recalculate the average after every new pixel joins the region. The orientation threshold chosen was 40 degrees.

5. Seed pixels for region growing were found by identifying a complete 5x5 window of unlabeled (and not masked out in the first step) of still-unlabeled region. If any pixel within the 5x5 window was masked out or already labeled in a region, then the pixel was not considered as a seed for a new region. Region growing ends when there are no more possible seed pixels.

Region Grow Predicate Formula:

```
/* Orientation test criteria to join region */
/*Pixel norm value*/
pixel_norm[0] = norms[0][(queue[qt]/COLS+r2)*COLS+queue[qt]%COLS+c2];
pixel_norm[1] = norms[1][(queue[qt]/COLS+r2)*COLS+queue[qt]%COLS+c2];
pixel_norm[2] = norms[2][(queue[qt]/COLS+r2)*COLS+queue[qt]%COLS+c2];

/*Calculate the dot product and find the orientation*/
dot_ab=seed_norm[0]*pixel_norm[0]+seed_norm[1]*pixel_norm[1]+
        seed_norm[2]*pixel_norm[2];
mag_a=sqrt(SQR(seed_norm[0])+SQR(seed_norm[1])+SQR(seed_norm[2]));
mag_b=sqrt(SQR(pixel_norm[0])+SQR(pixel_norm[1])+SQR(pixel_norm[2]));
cos_theta = dot_ab/(mag_a*mag_b);
theta = CONV_2_DEGREE(acos(cos_theta));
/*Predicate*/
if (abs(theta-average) > Orientation_Predicate)
        continue;
```

Results and Observations:

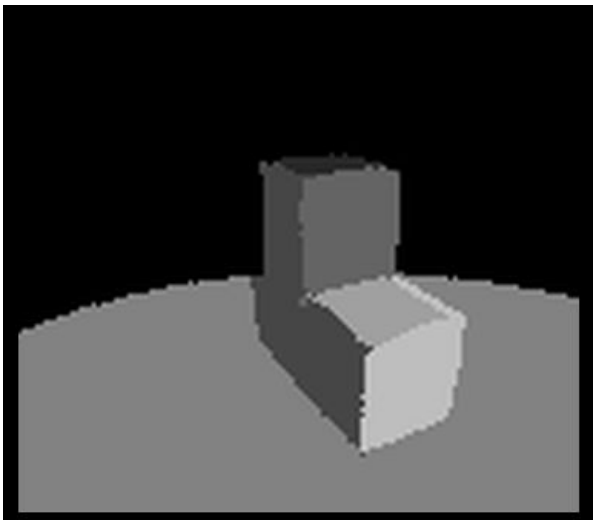

Fig 4. Segmented image

Region Label	#Pixels	Greyscale	Avg surface normal X	Avg surface normal Y	Avg surface normal Z
1	61	40	2.3536	-18.8632	6.4265
2	762	70	12.5854	0.5730	4.7127
3	494	100	-3.3766	2.2472	4.6300
4	5166	130	0.6706	-9.6898	3.1082
5	259	160	0.9101	-8.4452	2.4057
6	532	190	-10.2393	1.5054	7.2498

Table 1. Number of pixels for each region and average surface normal X,Y,Z