Programming Assignment 3

Ran Tian

4970588

**Section 1:** Histogram and Thresholding

The general process for this part is:

1: get the histogram array based on the pixel intensity.

2: smooth the histogram array with its 2 neighbors, take the average of these 3 values .

3: compute the threshold value and convert it to binary image.

Here is the strategy to get the threshold value based on the histogram array:

1: Select the initial threshold T;

2: Segment the image. We group pixels into 2 parts, the first is the ones with pixel intensity greater than T, the other is the ones with pixel intensity less than T.

3: Get the statistics for these 2 groups. I tried to get the mean, median and common values. And it shows that the mean has good effect. Corresponding to mean0 and mean1.

4:Compute the new threshold value:

T = (mean0+mean1)/2;

5: Continue step2-step4, and terminate the process when the difference in T is less than a value(h). In my trial, if h is in range(5,20), I get better result.

The threshold values for both images from my algorithm are both around 79.

**Section 2:** Region Extraction

The general idea for this section is to implement the raster scan method to extract the connected component.

For my implementation, I use 2-pass method both from column to column and then row to row.

The main idea for first pass is:

1: Ignore the background pixel

2: If west neighbor of current position is labeled, label it for the current position

3: else If north neighbor of current position is labeled, label it for the current position

4: else we start a new label for the current position and increase the label value prepare for the next region

The main idea for second pass is:

Given the union set from the first pass, if the west neighbor and north neighbor have different labels, we union the two sets in union set.

After the two passes, we get the labeled image and union set, and we can plot the labeled image in different colors.

But there are problems for this part:

1: We may have redundant regions which have less elements and probably are noises. My strategy here is to count the number of pixels for each region and set the threshold value to determine whether it is a plausible region. The threshold value based on my trial is 150.

2: For image 4, I get some misclassifications from different regions. In other words, In one case, we regard two regions as one because they are close to each other or even overlap a little bit. In another case, we regard one region as two regions because we lose some connectnesses within a region. For both of these two cases, it is directly depends on the binary image we have from section 1(or related to the threshold value from section 1).

This is actually the shortage of simple raster scan method, we can’t avoid them if we have image of “bad quality”. But once we have “good” image, we can have good result like image 3.

**Section 3:** Blob Statistics:

For this task, we compute the blob statistics for each region. The type of statistics I am interested in are: MBR, centroids, holes, holes\_areas, perimeter, elongation.

For image 3:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | MBR | centroids | holes | Hole\_areas | perimeter | areas | elongation |
| Region 1(red) | (15,165,95,214) | (54.26,  190.08) | 0 | 0 | 205 | 2583 | 16.27 |
| Region 2(green) | (157,185,208,236) | (181.96,  210.56) | 0 | 0 | 184 | 2024 | 16.72 |
| Region 3(blue) | (182,102,197,118) | (189.38,  110.22) | 0 | 0 | 17 | 212 | 1.36 |
| Region 4(yellow) | (46,113,99,154) | (72.70,  135.71) | 0 | 0 | 74 | 892 | 6.14 |
| Region 5(cyan) | (72,74,168,108) | (120.99,  91.50) | 0 | 0 | 140 | 2259 | 8.68 |
| Region 6(purple) | (118,168,148,216) | (133.47,  194.08) | 0 | 0 | 137 | 868 | 21.62 |
| Region 7(black) | (135,109,217,182) | (175.45,  145.8) | 1 | 384 | 221 | 3339 | 14.63 |

For image 4:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | MBR | centroids | holes | Hole\_areas | perimeter | areas | elongation |
| Region 1(red) | (24,161,106,217) | (64.27,  189.04) | 0 | 0 | 130 | 2513 | 6.72 |
| Region 2(green) | (119,165,198,227) | (160.14,  198.30) | 1 | 246 | 132 | 2999 | 5.80 |
| Region 3(blue) | (128,109,209,178) | (168.18,  144.12) | 1 | 344 | 204 | 3424 | 12.15 |
| Region 4(yellow) | (61,123,88,151) | (73.64,  135.97) | 1 | 276 | 72 | 325 | 15.95 |
| Region 5(cyan) | (69,64,166,105) | (119.86,  84.15) | 0 | 0 | 120 | 2522 | 5.71 |
| Region 6(purple) | (78,102,113,132) | (94.80,  119.91) | 0 | 0 | 122 | 416 | 35.77 |
| Region 7(black) | (103,140,120,157) | (111.47,  148.14) | 0 | 0 | 26 | 263 | 2.91 |

Notice that to find the holes, holes\_areas and perimeters, elongations are pretty challenging and complicated. I only implement simple method and there are some errors when I compare my results with the results from section 2.

To find the holes:

1: Construct new images for each labels, then invert each image.

2: Use raster scan method as section 2, find the number of connected regions and labels for each image.

3: Number of holes for each region is just the number form step 2.

4:Then use the same method as calculating the area of connected regions to calculate the area of holes.

**Notice**: I set a thresholding value for determine the connected component region in raster scan method => 100, which means if the area is less than 100, we don’t regard it as a hole!

To find the perimeter, I use the border followers:

1:first scan the label matrix from left to right and top to down, find the first position labeled as region.

2:Staring from the position in step 1, follow the direction in clockwise direction =>right, down,left,up. Each time we are blocked in one direction, we change to the next direction and continue the sweep.

3: The terminal case is tricky, Here I use the condition when we go back to the starting position. But obviously, this may fail to count the perimeter sometimes.

Again, the statistics here are directly related to the result from section 2. In other words, we misclassify the regions for image 4, then the statistics may make no sense like the number of holes, perimeter. But for image 3, since we have good region extraction, the blob statistic is making sense in some degree.