Image Processing and Computer Vision
CSCI 55700
Spring 2019
Project 4
Due: Wednesday, April 24, 11:59pm

1 Introduction

In this project you will implement a simple segmentation algorithm and identify connected components in an image. This involves the following steps:

1. Do the segmentation of the image based on an analysis of the intensities in the image (i.e. create a histogram, decide on a threshold and do the segmentation). Use a simple thresholding strategy of finding the valley between two peaks. You may want to smooth your histogram before trying to find the valley. Remember that the histogram is a 1D array of frequencies and not a 2D image (although this is what we use to view it). So, smoothing the histogram is done with a 1D filter on the 1D array. Use the following 1D averaging mask to accomplish this task: (1/9,2/9,3/9,2/9,1/9). How you will define a valley is up to you.

2. Perform a connected component analysis on the resulting binary image of step 1. Use 4-connected neighbor definition and the algorithm described in class for this.

3. For each component identified compute some features that will be useful in describing its shape. The features you are to compute are the following:

   (a) Area, A.

   (b) The centroid, $(\bar{x}, \bar{y})$.

   (c) Second order moments around the centroid (i.e., central moments).

   (d) Perimeter, P

   (e) Compactness measure as defined by $P^2/4\pi A$.

   (f) An elongation measure computed from second order central moments given further below.

Note that (a)–(c) are zeroth, first, and second order moments of area. As we saw in class, the (p+q)th moment of a region S is given by the expression:

$$m_{pq} = \sum_{(x,y)\in S} x^p y^q$$

According to this, $m_{00} = A$, $\bar{x} = m_{10}/m_{00}$, and $\bar{y} = m_{01}/m_{00}$. Once you have the centroid $(\bar{x}, \bar{y})$, then the second order central moments are computed as

$$\mu_{pq} = \sum_{(x,y)\in S} (x-\bar{x})^p (y-\bar{y})^q, \quad \text{for } p+q=2$$

The scale invariant second order central moments are obtained from this by:

$$\mu'_{pq} = \mu_{pq}/m_{00}^{\gamma}$$

where $\gamma = \frac{p+q}{2} + 1$, which in the case of second order moments (i.e., $p+q=2$) would be $\gamma = 2$.

The elongation measure from μ_{pq} are computed as follows:

$$elong = \sqrt{\frac{\sqrt{(\mu'_{20} - \mu'_{02})^2 + 4(\mu'_{11})^2}}{\mu'_{20} + \mu'_{02} + \sqrt{(\mu'_{20} - \mu'_{02})^2 + 4(\mu'_{11})^2}}}$$

Note that (elong) can also be considered compactness (circles are compact and thin ellipses are not compact). How does this value compare with the compactness measure given by (e) above?

2 Data

Run your program on at least the following images:

- hinge.pnm
- hinges.pnm
- shapes1.pnm
- shapes2.pnm
- shapes3.pnm
- shapes4.pnm
- keys.pnm
- pillsetc.pnm
- coins.pnm

If you can find some other images, run your program on them also.

3 Output

Your output should consist of the following items:

(1) A histogram of the image intensities (do this as a bar graph on a regular printer output. No need for fancy graphics needed).

(2) The value of the threshold intensity picked by your program.

(3) The resulting binary image.

(4) A display of the connected components. Show the same components as one gray level. A list indicating what gray level stands for what component so that we know how to interpret this image. Alternatively, you can use pseudo-color to encode the segment id.

(5) The list of features computed for each component. This could be a simple listing of component id and the features computed.

4 What to hand in

You will hand in a report for this project and the source code. Submit your source code as a zipped folder separately from your report. Make sure you have the proper makefiles or shell scripts that we can compile your program and run and test it on tesla. Then zip the entire folder and submit.

The report will contain the following:

1. A short introduction and a brief description of the theory and the algorithms you are implementing.

2. The results of running your implementations on the images given above. You are welcome to test other images as well. The results in your report should include the following clearly labeled.

    - Input image.
    - Outputs of your program run on test images as described above.

3. A short discussion and any conclusions you may draw. How good is the segmentation algorithm doing? What problems arise? How you might improve the performance? Are the measures you computed reflecting the shapes properly?

5 Grading Rubric

Report	25%
Segmentation	25%
Connected Component Analysis	25%
Feature Computations	25%
Total	100%