Kyle Qian

Connected Component Labeling

This project tasked us with finding and labelling connected components in an image. There are several different ways of going about the CCL algorithm, for example recursively or iteratively, but both of these methods are inefficient. For my project I used a sequential CCL algorithm, which has a much better time complexity than both the recursive and iterative methods— as the sequential algorithm only has to make two passes through the image, while recursive and iterative methods require searching from every pixel.

The basic idea of the algorithm is as follows:
For the first pass, because we are traveling through the image from the upper left corner to the bottom right corner, scanning from left to right, we only need to check two neighbors per pixel: the upper and left pixels. Depending on the label of each of these neighboring pixels we can choose what we should label the pixel we are currently viewing. This will require us to keep track of label equivalencies— which is essentially tracking disjoint sets of labels. We have to do this because the way we scan the image will cause certain objects to contain more than one label in certain areas.

<u>First Pass</u>

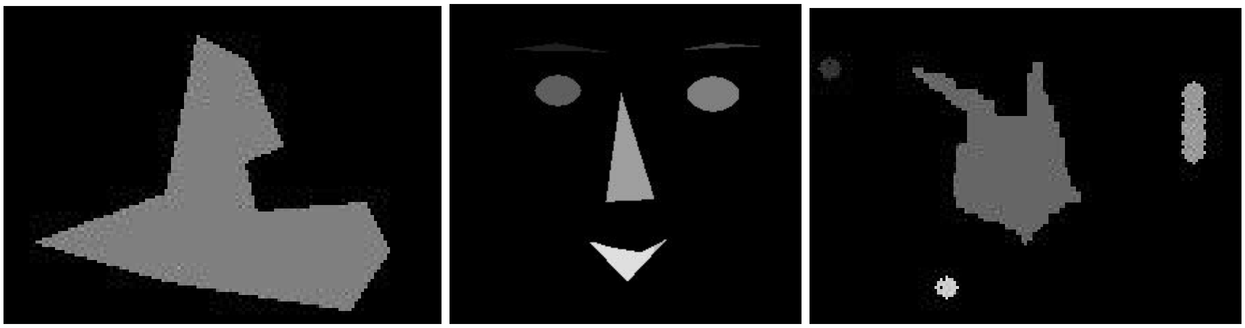
1. Find labels of the left pixel and upper pixel from current pixel
    a. Note: if there is not left or upper pixel because we are on a border, we can assume that the label of that left/upper pixel is 0
2. Follow the table:
    a. If L == U and they are both 0: make a new label for the pixel
    b. If L != U and one of them is 0: label the pixel the non-zero number
    c. If L != U and neither of them are 0: label the pixel the smaller number and track the new label equivalency between the upper and left labels
    d. If L == U and they are not 0: label the pixel the label of its neighbors

For the second pass, we simply need to use the equivalencies we had discovered during our first pass and replace each equivalent label with its root label. The root labels will be unique for each object and will allow us to properly label each object.

<u>Second Pass</u>

1. Find root labels for each label that has an equivalency
2. Color each object based on its root label

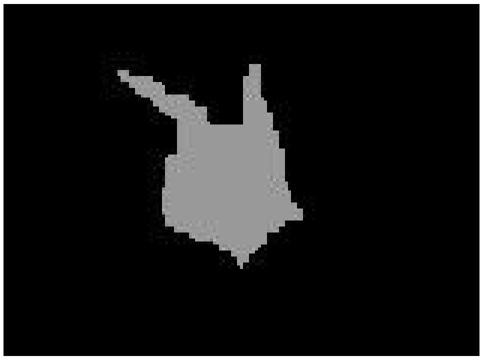
Here are the results of the CCL algorithm on a few test images:



From these images we can see the different objects based on their shading of gray. From left to right, the first image only has one object, while the second image contains 6 objects and the third contains 4 objects. Because we know the main focus of our third image should be the finger gun, we can apply a size filter to remove noise.

The size filter can be accomplished by simply counting the number of pixels for each root label and comparing that sum to a minimum area. If the sum is smaller than the minimum area then we know not to include that object in our result.

Here is gun.bmp with a size filter of min_area = 1000 pixels:



As you can see, we have removed the noise from the image and only the finger gun remains.