Jack Hanke
MSAI 495 Computer Vision
MP #1

Connected Component Labeling Report

In this assignment we implement a sequential connected component labeling algorithm. This algorithm was designed to improve on the naive recursive approach to connected component labeling. The faster algorithm is described as follows.

A bitmap image is input, and is parsed in raster order twice. In the first pass, the image is initially labelled. The initial label for a pixel is calculated by examining the pixels one to the left and one above. If these locations are outside the bounds of the image, the pixel value is considered 0, i.e. not part of a connected component. Let us call these values *up* and *left*. If these values are equal, the current pixel has this value. If only *up* is 0, then the current pixel's value is *left*, and vice versa. If *up* and *left* are both non-zero and different, assign the current pixel the *up* value. Then note that *up* and *left* are equivalent in the equivalence table, and update these equivalences if necessary.

In my implementation, the equivalence table is a list of sets, where each set represents all labels that are equal to each other. Updating equivalences entails popping sets that contain the *up* and *left* values, and appending the union of the two sets. If *up* and *left* are different values but equivalent, no update is required.

In the second pass, the image is again parsed, relabelling the initial value with the index of the . In my implementation, I further linearly scale this value to obtain equal shading differences between 0 and 255. I also chose for these labellings to only be shown in the red channel of the image to better distinguish between the black background. This leads to a more clearly segmented image. These adjustments to coloring can best be seen in labelling of the "face" image below.