Jack Cardwell

Professor Bailey-Kellogg

Computer Science 10

April 15, 2015

Problem Set 1

In my RegionFinder class, findRegions is the main method that takes an image, goes through all of the x and y points, so long as they haven’t been visited and are of the right targetColor, and adds them to a region of points of the same color. Points are marked as visited through an initially black image that turns each pixel red once it has been added to a region. I chose to make a point red to mark it because I figured that it would be unlikely for a color to be entirely (1,0,0) in an image, so it would increase the probability that each pixel of an image is visited at least once. Each region that is discovered is stored as newRegion in the while loop, but only some are stored. Arbitrarily, we only care about regions that contain 50 pixels or more. After exiting the loop, once a no points are left to visit, a region is saved in an array list called regions if it meets this requirement. I determined which points met the targetColor using a method entitled colorMatch. I isolated colors into their three channels and compared the difference in each channel. As long as the difference in each channel didn’t exceed 20, I allowed the colors to be considered the same. The value 20 accounts for color error that could be attributed to lighting conditions, such as shadowing. Increasing this value would’ve led to bigger regions. Toggling this value would’ve been interesting for the webcam application, but it suited the purpose well for the stationary image.

CamPaint was the other portion of this short assignment. It contained three “screens” that implemented different methods. By pressing the “w” key, the user would pull up the normal webcam window. The next screen was a recolored image window that implemented the recolorRegions function of the findRegions class. This application was practical because it allows the user to see what color they selected. Finally, the painting screen draws in a certain color on a blank white screen. While region growing does find a color in a webcam, lighting conditions greatly diminish its utility. Because most places aren’t uniformly lit, movement could change the tracking color significantly, which would change the region. This inaccuracy within the regions led to some difficulty when trying to paint. For instance, the webcam picked up regions that were outliers from the original selection and used them as paintbrushes because their color appeared to change in the light. I’d also imagine that if regionsFinder were applied on a bigger scale or to a bigger image, it would take a lot of computing power to make it work because the runtime for the method is fairly large. Therefore it would have to cipher through lots of pixels very quickly before the image changed. While region growing worked for our smaller webcam and stationary images, its poor runtime and changing lighting conditions limit its practicality.