For my region growing algorithm, I found that different values of maxColorDiff worked better depending on which of the effects was being called. The original value of 20, however, did not work for Baker-Berry when combined with the original value for minRegion (50). Either the color difference needed to be larger so that slightly different bricks could be combined into one region or the minRegion needed to be smaller so that the tinier (but more exact) brick regions would be counted as regions. I ended up leaving minRegion at 50 and moving maxColorDiff from 20 to 200 for the regionTest snapshot that I submitted, and I also liked the way the recolored webcam worked at 200. 200 seemed to be a happy medium where the difference was high enough that distinct regions showed up noticeably on Baker-Berry or on my webcam but also low enough that random regions wouldn’t start showing up along the curb or on tree trunks in front of Baker-Berry. The smaller the color difference is, the spottier the regions look, because the region finder is getting pickier, and as it gets larger, the regions become more filled-in, and eventually start morphing together into larger regions. For the painting, having a maxColorDiff of 200 made the paint strokes look more like chalk than paint (if that makes sense) and it was harder to draw because if I tilted the “paintbrush” in the wrong direction then the light would hit it differently and it would fall outside of the max difference and suddenly no longer register as a region with which I could paint. So for the painting I preferred a maxColorDiff of about 500 (and that is the maxColorDiff I used in for the painting in the screenshot I submitted). It made drawing easier since the region finder could continue to track an object even if the lighting changed its color slightly, and it allowed the pixels of the object that were slightly different in color to be included in the region so that the brush strokes looked darker/less spotty. Also, I’m not sure that this is considered a valid limitation or if it was just me being dumb, but I realized relatively early on in my testing that it is far easier to test webcam-related region growing on matte objects. The shinier an object is, the higher maxColorDiff has to be to compensate, because otherwise as soon as the object moves it reflects the light and/or colors from surrounding objects differently and no longer registers as the same region (oops). Additionally, this is definitely not a limitation to region growing in general, as it was just a bug, but I haven’t been able to figure out why it was occurring, and I’m still intrigued by it—in the two for loops that iterate through all 8 neighbors of a given point, when I (incorrectly) wrote my range as less than or equal to either point.getX() + 1 or image.getWidth() (without the “-1” after image.getWidth), I noticed an error where my recolor webcam only worked on specific colors. For example, it would accurately recolor regions on my face, but not my hair. I inserted a print statement in my HandleMousePress method to see if I could notice a numerical pattern between the colors that worked and the colors that didn’t, but whatever the pattern is, I couldn’t figure it out. I’m just not quite clear on why the absence of the “-1” affected which colors could be used as the tracking color.