Name: Hoon Yang

Date: Feb, 10th, 2025

The 3rd assignment is a continuation of the 2nd assignment where we not only make advanced DIP functions but also decompose the program into separate modules and link everything together to make one large c program file with an enable/disable Debug mode. The Advanced DIP operations include fisheye, posterize, rotate, and motion blur, all of which are written in a different file from the file that has previous DIP functions. For “fisheye” function, we just have to write the correct code based off of the pseudocode provided. It looks confusing at first glance but is, in fact, straightforward as long as you realize which datatype you use for each variable. For “posterize” function, we have to manipulate the image by converting the RGB values in bits to make a poster-like image. All that matters is just figuring out the right conversion from RGB value to desired value. It’s a little bit tricky to find out how but it just requires some math and know the concept of bitmask. For “rotate” function, it’s another math thinking required task similar to fisheye. You have to simplify each x and y coordinate by matrix multiplication and make sure the angle is initially converted to radian. I had a realization that since the image is rotating in a clockwise direction, it’s important to change the sign of sin and cos. The last function is “motion blur” function and it’s supposed to create a horizontal blur depending on the amount of blurAmount input value. I was stuck on how to approach in making this function but then I read through the lab slides and was able to figure out how to do it. For each pixel, its original value is halved, and other half value goes to the right neighboring pixels. Basically, we need to sum up all the neighboring pixel values within blurAmount and divide the sum by a count, in case the number of neighboring pixels is less than blurAmount. Finally, halve the sum of the original pixel and neighboring pixels and it will create a blur.