CSC320 Assignment1 Lab report

Zhongliu Liu 1004706936 liuzho15

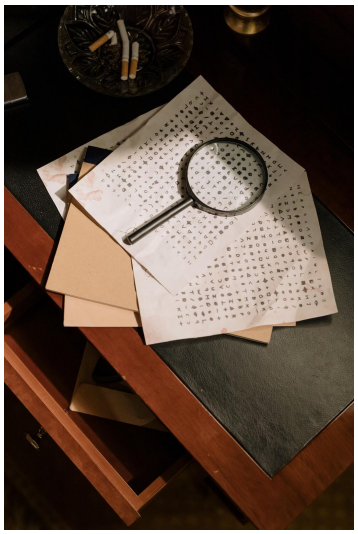
In this report, I will discuss the features of the results, the part needs improvement and how I could improve it.

Firstly, Here are some test images and the results of their transformations.

Building0

Building1

Document0



Phone0

Tokyo0

Document1

Then here are some images that I chose.

My_image1

My_image2

My_image3

My_image4

My_image5

Observation based on image and results

From both results of test images and images that I chose, we can see that the function works smoothly, it can perfectly projects the image of our input onto the plane that we want. The features includes windows, the clock, the document with a magnifier, the screen on phone and an ad board. However, we also see some flaws of the projections.

For instance, the window frames from building0 result is not a straight and the color is discontinuous. This might due to backward mapping. During the process, when we denormalize the coordinates of source image, we did some rounding/flooring calculations, to choose the nearest pixel of that coordinate, that may cause inaccuracy.

Take a look at building1, the edges of the the clock is not smooth, this also appears on the edge of the circle logo in image tokyo0. They has some jagged liked features. The reason might be that, the result image is enlarged compare to the parts it appeared in the source image. Some of the pixels in the destination image might refers to the same pixels in the source image, so this creates a pixel like, jagged edge.

Thoughts on algorithm of backward mapping

1. From the observations above, we can tell that when we want to project the detailed parts of the image, the edges may not be smooth.

2. The algorithm is really time-consuming while dealing the larger size image. During the backward mapping, to find the pixels within the range in the mask matrix, it needs to iterate through h,w where H,W is the height and width of an image. Therefore the time complexity should relates to O(HW)

Dedication on result improving

To solve the edge problem and improve the behavior of the result image, I prefer to use a Gaussian blurring filter on the result image, so that pixel color around the edge would be blended better and reduce the inconsistency.