CSC320 Assignment 2 Report

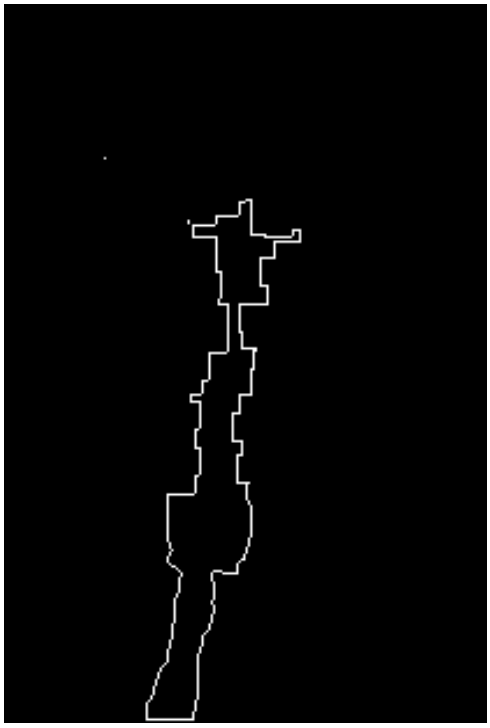
Zili Xie

Feb 19th

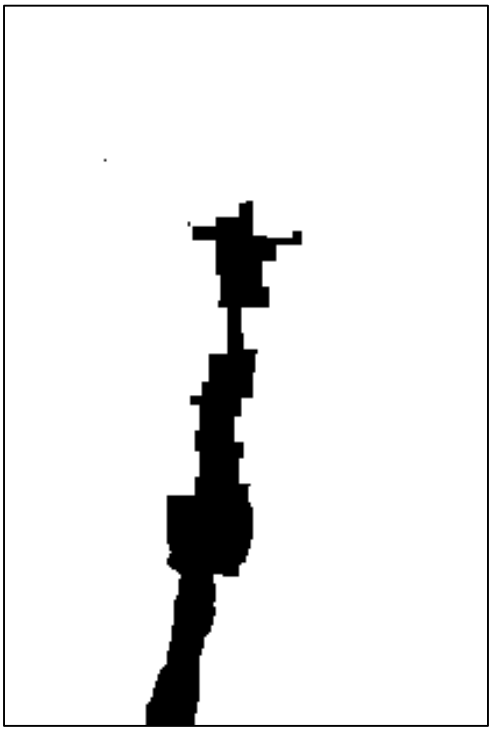
Part B.2.1 and Part B.2.2

1. Running the inpainting for 100 iterations on the test image pairs will produce the following results.

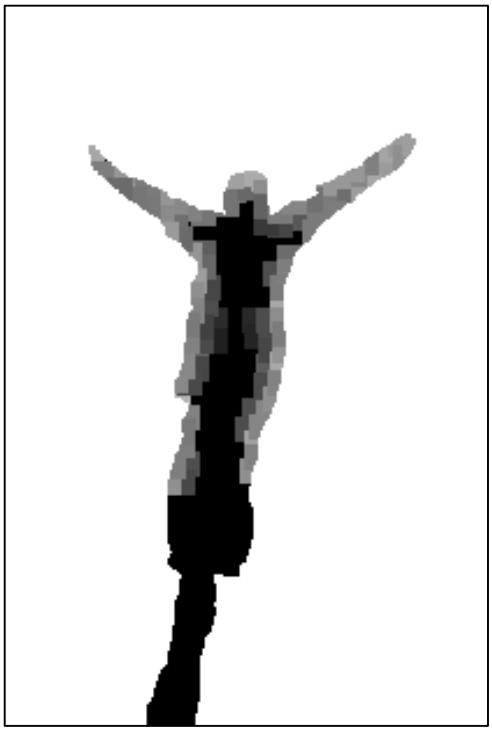
input.inpainted.png



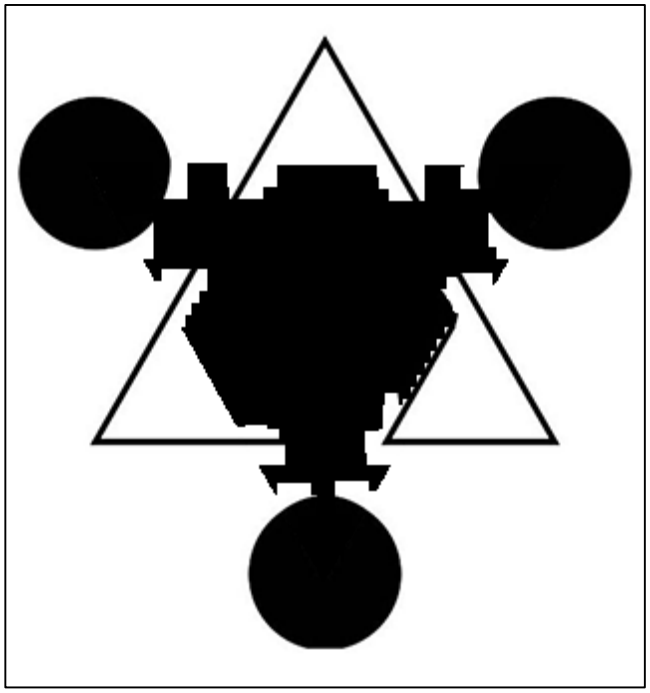
input.fillFront.png



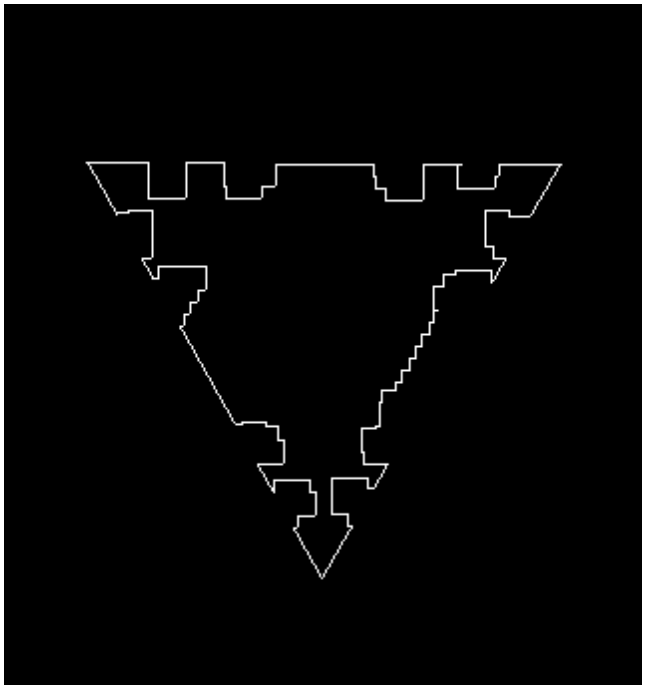
input.filled.png



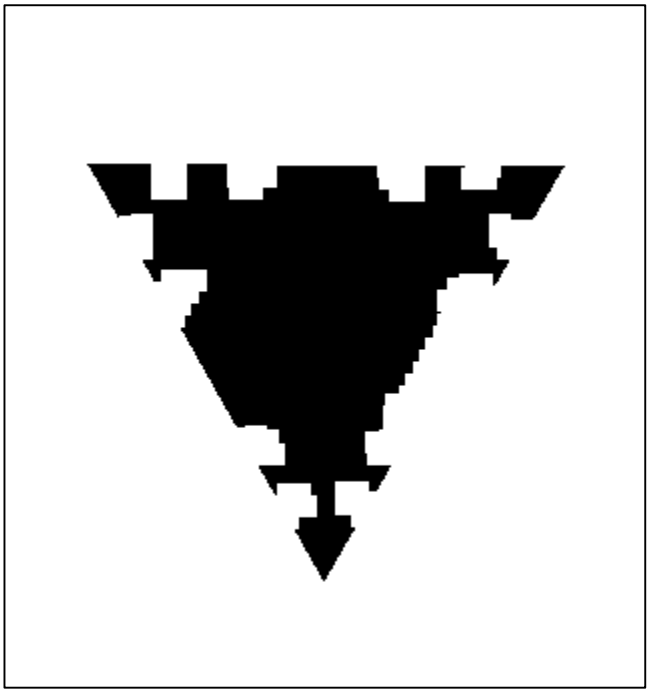
input.confidence.png



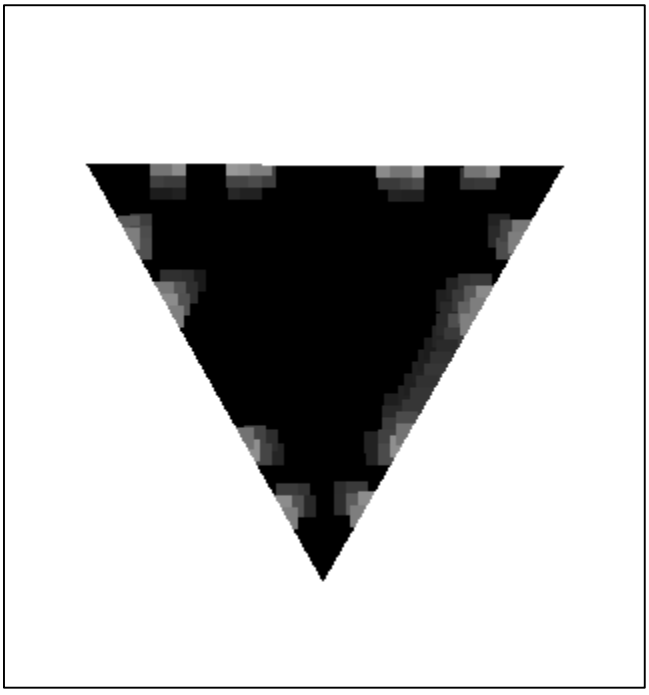
Kanizsa.inpainted.png



Kanizsa.fillFront.png



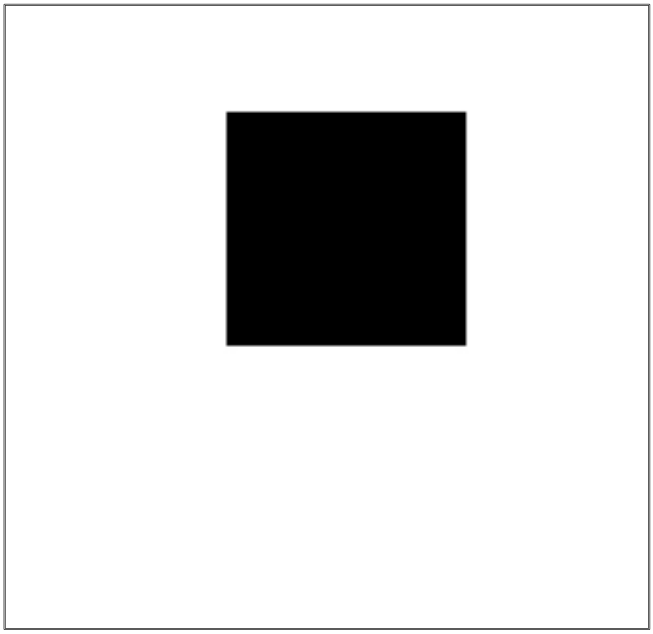
Kanizsa.filled.png



Kanizsa.confidence.png

2-5 Two source photos captured by my own camera are shown below. Source1 represents a good case for inpainting while Source2 demonstrates a bad case.

Source1.png

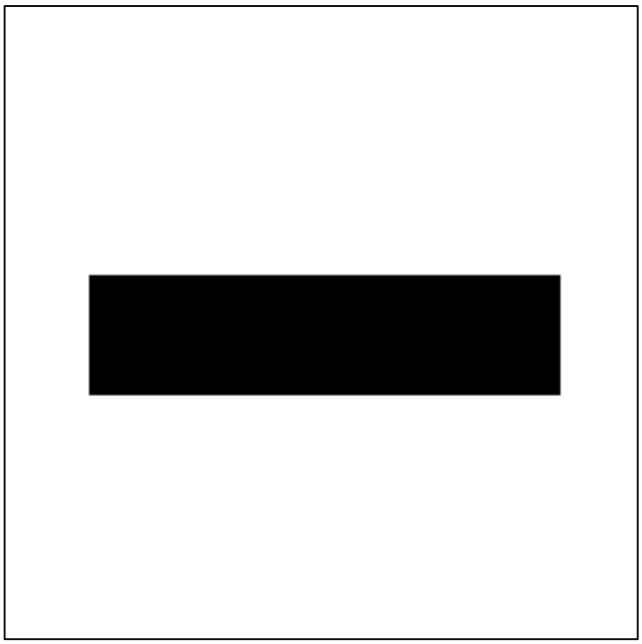


Mask1.png

Source1.inpainted.png

(1)
The first pair (Source1.png and Mask1.png) represents a good case for inpainting because it successfully regenerate the texture and pattern at the background. The blue and white stripes are inpainted in a correct order and arrangement.

Source2.png



Mask2.png

Source1.inpainted.png

(2)

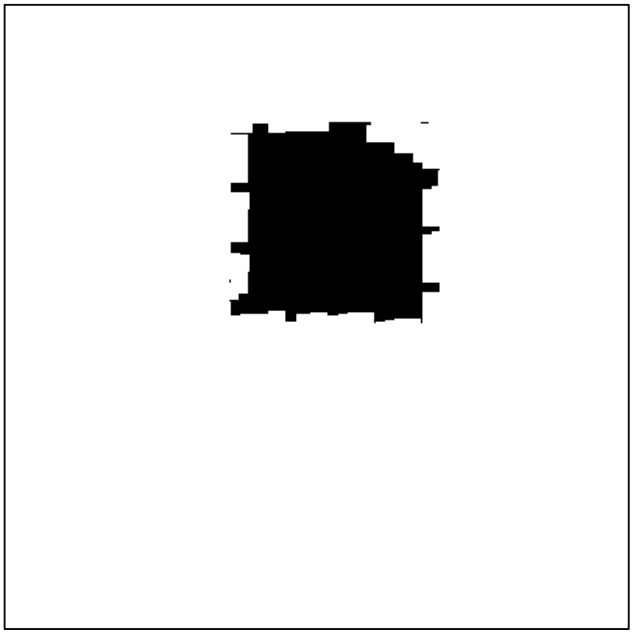
The second example is considered as a “bad” example because the inpainting program cannot correctly restore the pattern (stem and leaves) covered by mask image. One reason of this is the complexity of background area covered by alpha mask. The image pattern is so intricate that it’s hard for the program to find a similar patch and make the elongation through max gradient.

(3)
Some visible artifacts in result images from the two datasets: In “good” example, the inpainted result image has a slight displacement. See the zoom in image below.

Source1.inpainted.png (detail)

One obvious reason is that the image inpainting actually starts from both left and right at the same time. At the time when both sides are extended to the middle of mask, the stripes might not be aligned with each other. The below is the fillFront image after 200 iterations.

Source1.inpainted.png (200)



Source1.filled.png (200)

The patches in the final iterations will stitch the stripes that are extended from both sides together. Within one patch, if the program finds that the stripes

are not aligned, it will force them together using unnatural lines, which causes displacement.

Source1.inpainted.png (300)

Also, in image detail, you can find that the blue stripes are not in pure color, the blue color is actually mixed with some small white dots. This might cause some chaos on finding the max gradient, so the algorithm may choose to extend the white stripe from a white dot which lies inside of blue.

In the “bad” example, the result looks relatively “good” in the first 300 iterations.

Source2.inpainted.png (100)

Source2.inpainted.png (300)

But because the pattern is too complex that sometimes it’s hard to find a patch that actually looks similar to the origin image, one wrong patch may cause all the following patch to be wrong (filled in an incorrect order).

A patch in the above image introduce vertical pattern (gradient changes in x direction) which affects the following patch around this area.

Also, here is another example, the direction of extension was originally horizontal, but this new patch introduces

a vertical pattern, which makes all the following inpainting to be wrong.

That is why this L shape artifact show in output image.