Kavin Krishnan
PS1

1. SHORT ANSWER PROBLEMS

1. The associative property of convolution states that (f * g) * h = f * (g * h). You have to apply a gaussian to an image before applying a filter to smoothen it first. However, the formula shows that we can just apply the gaussian g to the filter h, and then use the produced matrix on the image.

2. [0 1 1 1 1 1 1 1]

3. [-0.25   0   0.5   0   0.25]

4.

> i. One way to minimize the amount fine, detailed edges is to have higher threshold values for the intensity capture. By doing so, this maximizes the amount of major edges who have intensity values the fall between the thresholds during hysteresis.
>
> ii. Another way is to increase the sigma of the gaussian used to smoothen the image. The larger the sigma, the smoother the image and fine-grained edges are less like likely to get preserved as much as major edges.

5. A flaw with additive gaussian noise is it does not simulate real world factors that influence real noise. Simply using the normal distribution to add noise does not completely simulate actual noise.

6.

> i. First you must define an interval to sample a frame from the video feed. This interval should be based around the average rate at which parts are placed on the assembly line
>
> ii. Next a canny edge detector is used to highlight the major edges of a part. This will help in identifying a part in the next step.
>
> ii. The part must be identified based on templates of the different parts the factory produces. These templates were taken using canny edge detection. A distance transform is done on the image using the chamfer distance on the feed sample and part templates. The feed sample is assigned to the part who's template best matches using the chamfer distance.
>
> iii. After identifying which part is being analyzed in the feed sample, Binary image analysis is done to make sure features of the part were properly completed. This includes identifying connected components and extracting feature blobs.

2. PROGRAMMING PROBLEM: CONTENT-AWARE IMAGE RESIZING

1.
outputReduceWidthPrague.png


outputReduceWidthMall.png

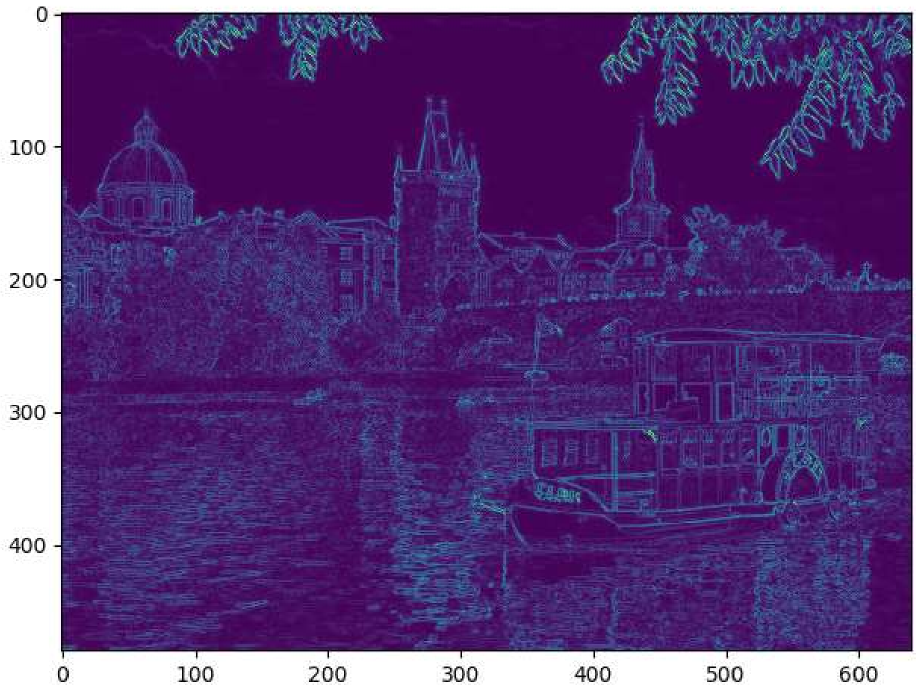

2.

outputReduceHeightPrague.png

outputReduceHeightMall.png

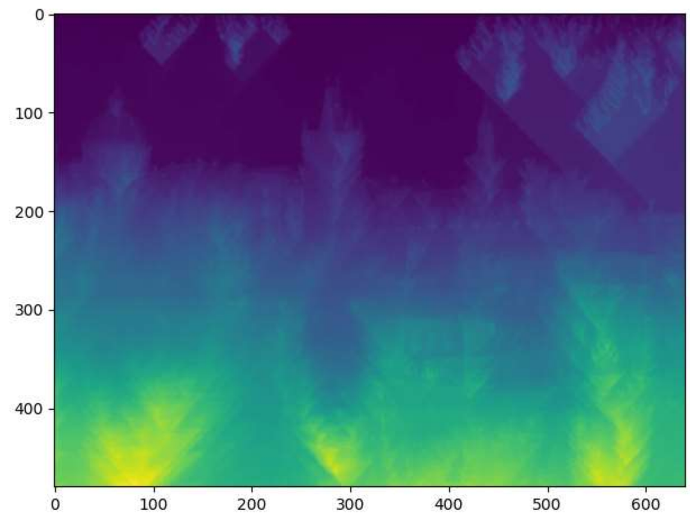
3.

Energy Function



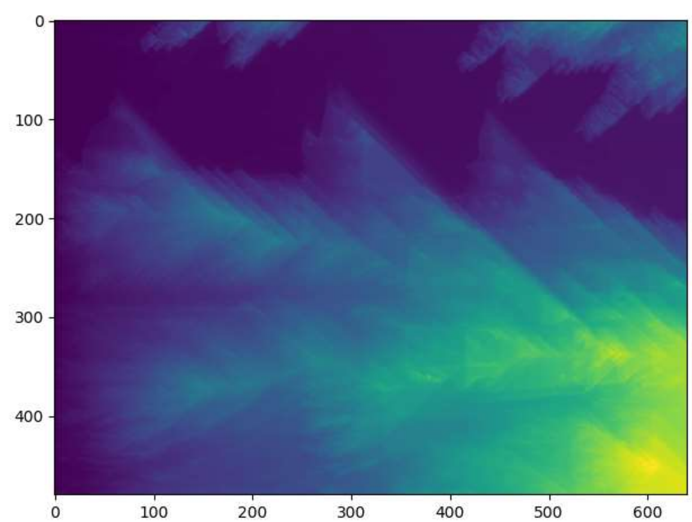
The energy function appears this way as it takes the sum of the absolute gradients in the x and y direction. This enables to the capture of edges (as highlighted above) by detecting pixels where the intensity values make the largest change. For example, the boat is composed of very distinct contrasting colors and this allows for good detection edges.

Cumulative Energy in the Vertical Direction



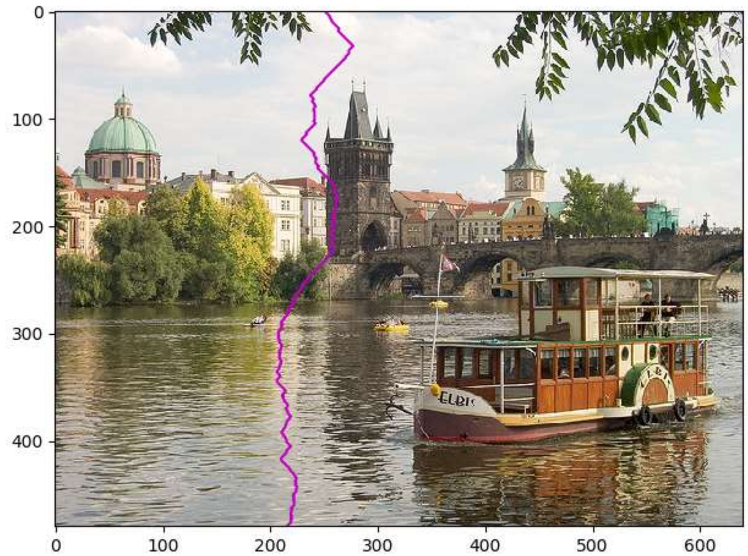
This graph shows that the largest seams (largest cumulative energies) occur when x is around 100, 300 and 600. At these areas, we can see vertically that there is a lot of objects and contrasting colors. For example if were to trace to the top of the image vertically from around x=600, we can see that there is a bridge as well as a boat. We can see in the energy function plot that there are a lot of edges in this area. We can see that around x=200 is our smallest seam as vertically there is essentially just an empty area of water and the sky. There are very few edges in this area.

Cumulative Energy in the Horizontal Direction



This graph shows that there the largest seams (largest cumulative energies) occur when y is greater than 200. This is a result from the boat as well as the crowded skyline of buildings which have a lot edges. The smallest seam is around x=100 because horizontally across the image there is just the sky with very little edges.

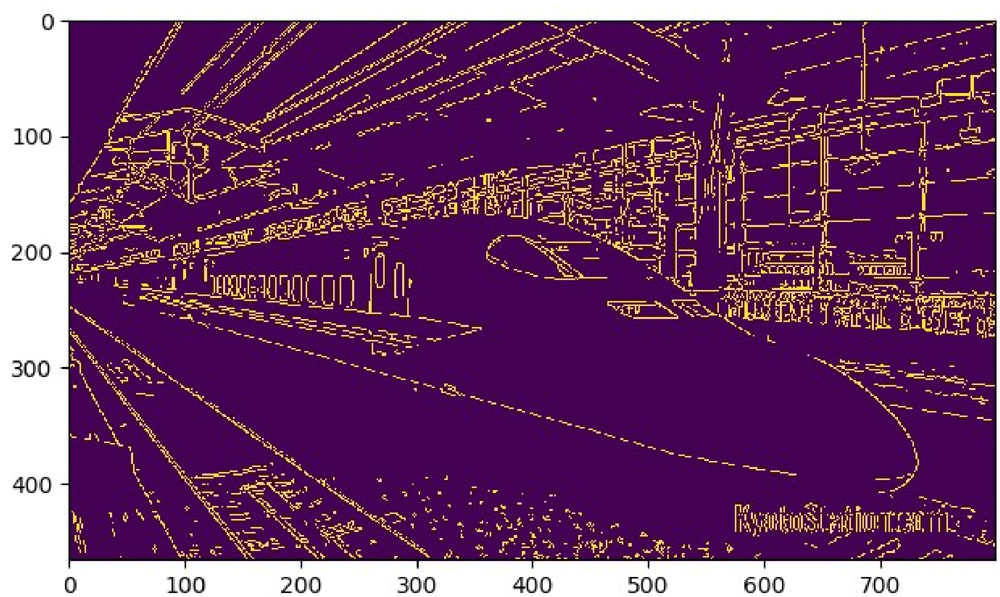
4.



This is the optimal seam in the vertical direction as it has the smallest cumulative energy vertically. We can see that it merely contains the sky, a nearly uniform colored façade of a tower, and an empty area of water. We can see along this seam that are not too many edges compared to other areas of the image. Other vertical seams have a higher energy due to buildings with more complex features, the bridge, as well as the boat. Essentially these seams have a lot more edges.

This is the optimal seam in the horizontal direction as it merely traces over the sky. This seam almost has no edges as shown by our energy function. It can be clearly seen that the sky has a nearly uniform intensity and therefore the seam with smallest cumulative energy is found tracing over it. In other horizontal seams, we can see that it runs across the buildings, bridge, and boat which have a lot of edges (and therefore adds a lot of energy).

5. Canny Edge Detector



I modified the energy function to use canny edge detection. By doing so, major features are emphasized through hysteresis. This can benefit seam carving as it focusses on the major features of the image and assigns high energy values in these areas. Looking at the original energy map, we see that it detects more fine-grained edges as well which may or may not add noise to determining the most optimal seam.

6.

Image #1

a)

b)

c)

d) Input: 768x513
 Output: 668x413

e) Reduced the width and height each by 100 pixels through vertical and horizontal same carving

f) Using the horizontal and vertical seam reduction, the green background behind the red panda gets minimized and more focus is put on the red panda as shown in output b. We see that when we use cv2.resize() in output c the whole image simply gets scaled to a smaller matrix rather than remove areas with the least major features.

Image #2


a)


b)


c)

d) Input: 240x360
Output: 140x260

e) Reduced the width and height each by 100 pixels through vertical and horizontal same carving

f) Using horizontal and vertical seam reduction, the white background behind Wall-E the robot gets minimized. However parts of Wall-E's arms get removed because a lot of the optimal vertical seams get the lowest cumulative energy through the uniform white background but they also run through his arms. Therefore, an image of a squished robot is produced. Using cv2, an arguably better image is produced by simply scaling the entire image into a smaller matrix.

Image #3

a) 

b) 

c) 

d) Input: 1024x768

Output: 924x668

e) Reduced the width and height each by 100 pixels through vertical and horizontal same carving

e) In output b, Spiderman gets more pronounced. However in output b with vertical and horizontal seam carving, the building towards the right of the picture becomes deformed. A lot of seams run through the building as it is a large of area where there is a little contrast in intensity. As a result, the geometry of the building is not preserved. In output C however, cv2.resize simply refits the image into a smaller matrix and preserves the geometry.