Ali Asif & Raghav Sreeram

Short Answer Problems

1. In convolution, you flip filters in both dimensions and then apply. In the associative property of convolution states that the order in which multiple images have this process applied to them does not matter. An example of this is if we were to first blur then sharpen an image as opposed to sharpen and then blur an image. There is a computational advantage to using separability as fewer additions are required when the filters are applied in succession to one another than there is when the filters are combined to form one filter and then filter the image.
2. Dilation: [1 0 1 1 0 0 0 1] with [1,1,1]
   Answer: [1,1,1,1,1,0,1,1]
3. A possible flaw of using additive Gaussian noise is that the outcome may result in the blurring of lines that were originally meant in the image to sharper. The reason for this is because additive Gaussian noise includes a normal distribution in instances of poor illumination, high temperature, etc. The issue here is that because it is based on the normal distribution your image's white noise would be random.
4. The camera perched up upon the belt will collect images of the same part being mass produced repeatedly. Each image will look almost identical in the sense that they will all have the same image produced. This image will only show the assembly line product such as a door and the black color in the background from the conveyor belt. Any difference between images can be considered flaws in the assembly of the part. An assumption we can make is that there can be one 'master' image that which we can refer to that has what the assembly product should look like along with the belt in the background. Each image from then on can be uploaded to Matlab and the energy image can be found on each one. Next, the cumulative minimum energy map can be calculated for both vertical and horizontal seam directions. This will be done by taking the energy images calculated earlier. With these energy maps, we can calculate horizontal and vertical seams. We can transition from the lowest values in a column and then move on to the next column and find the lowest values of pixels that are either right on top of that pixel or one to the right or left of that. By repeating this process for each row/column of pixels in the image we can eventually form a seam for each image. One thing we know is that if the seams overlap with the master image, they do not have flaws. If we do happen to see a seam not in line with the others, we can deduce that there must be a flaw in this assembly part because the camera saw some sort of extra or missing feature in them.

Programming Answer Problems

1. **Decrease Width**

Output Seam Carving Prague

Output Seam Carving Mall

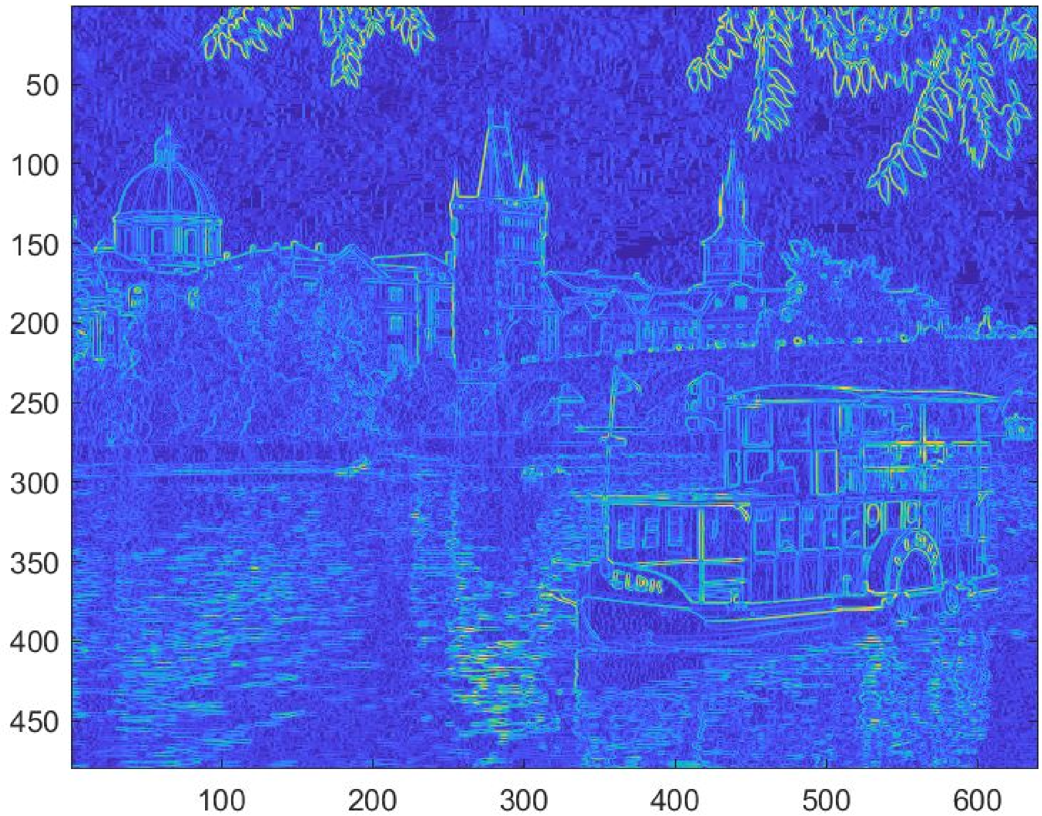
2. Decrease Height

Output reduce height prague

Output reduce height mall

3. Energy Output and Cumulative Energy Maps

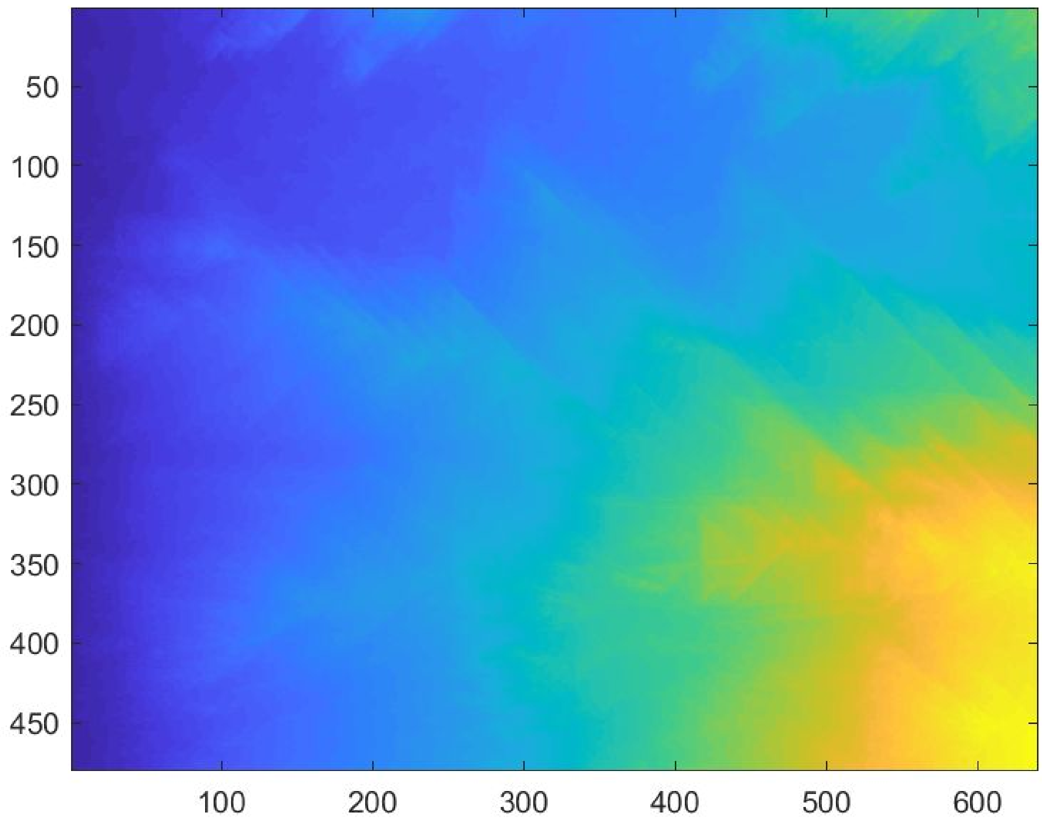
a) ENERGY OUTPUT



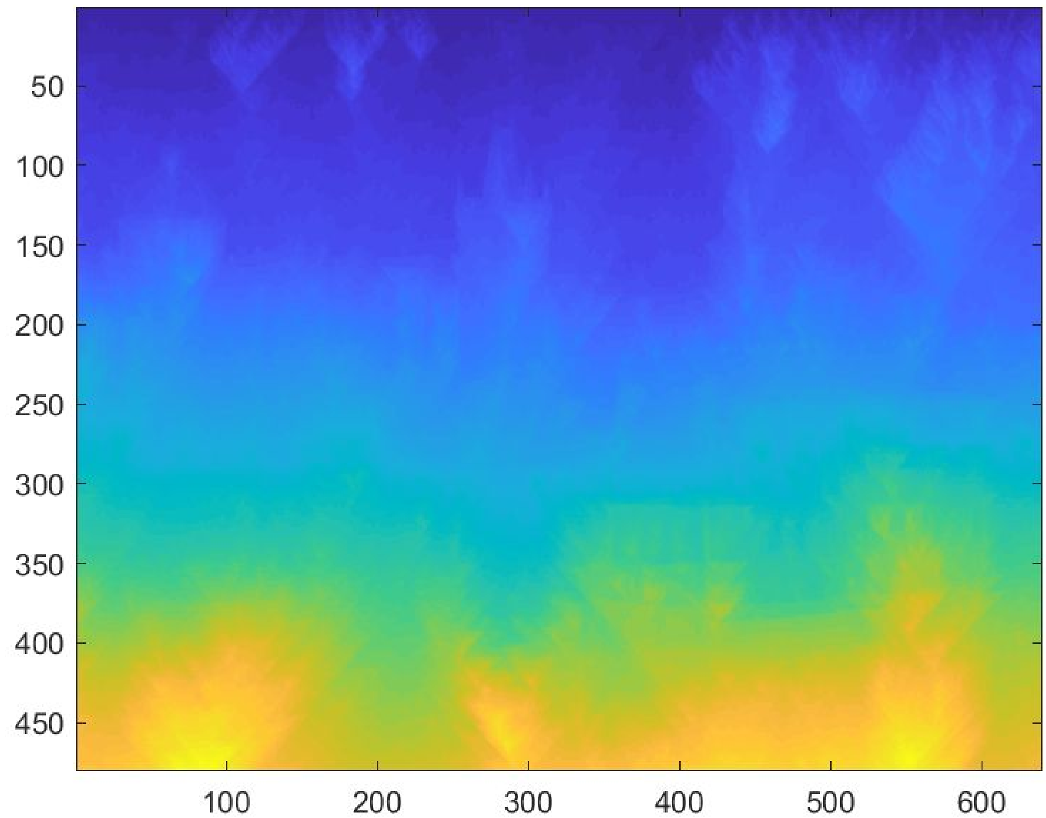
- EXPLANATION: This displays the output of the energy at each pixel using the magnitude of x and y gradients. The image above shows the energy values which is why we see the edges being highlighted here.

b) CUMULATIVE ENERGY MAPS

HORIZONTAL:



VERTICAL:



- EXPLANATIONS: The above figures show the horizontal and vertical cumulative energy maps. The maps show the cumulative energies upon which we can later do backtracking. This can tell us things like which seams are better for removal. As we can see in these examples, the horizontal map shows that the minimum energies are higher on the bottom right side and the vertical map shows the minimum energies are higher on the entire bottom area. This tells us that when we want to perform seam carving, we will likely carve areas in the top both horizontally and vertically.

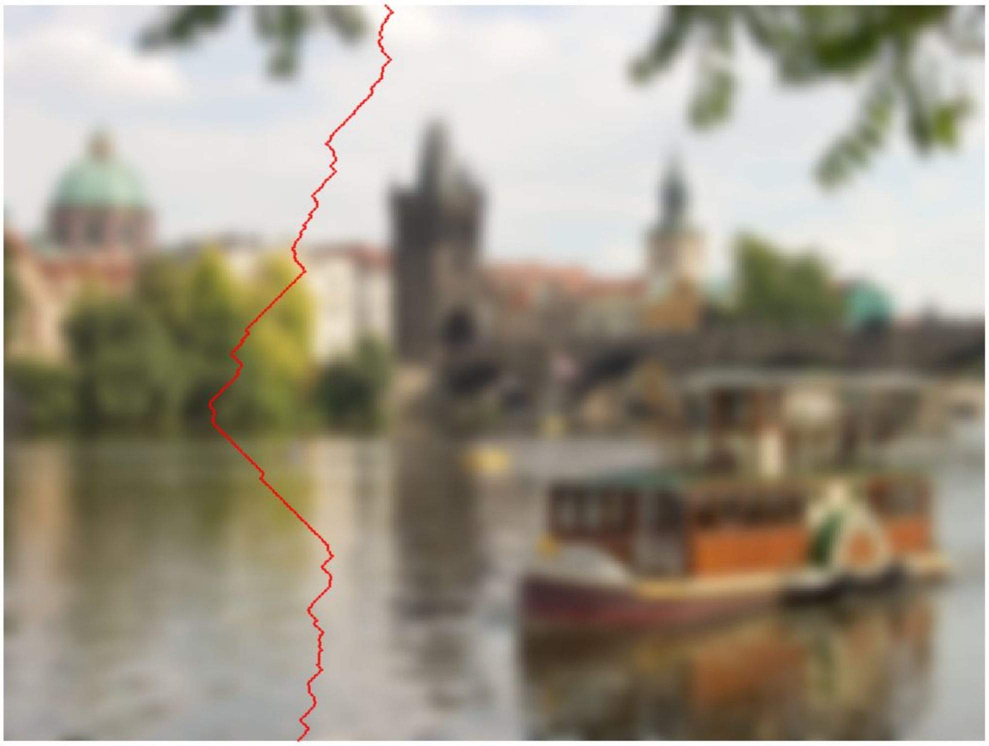
4. Horizontal and Vertical Seams

HORIZONTAL SEAM:

VERTICAL SEAM:

- EXPLANATION: The above figures show the first selected horizontal and vertical seams. The best way to see why these seams are optimal is with respect to the cumulative energy maps shown above. We see in the horizontal map that the optimal horizontal line we can draw (the area with the lowest values, represented in blue) is the same area in which the optimal horizontal seam was marked here. Similarly, we can see from the vertical cumulative energy map that the column that has the lowest minimum cumulative energy values (depicted in blue) is exactly where we see the carving for the optimal vertical seam.

5. Change the Energy Function - Blur



- EXPLANATION: As we can see above, we have changed the way in which the energy function is computed by changing the filter used. Shown are the results of the first vertical and horizontal outputs after blurring the image by a factor of five using the imgaussfilt function. As gaussian blurring is a method to reduce image noise and increase smoothing, edges will no longer be preserved. The impact of this is actually beneficial as the smoothing causes the seams to not be aligned with edges. Here we see that the vertical seam has adjusted so that it is not longer aligned with the building. In normal situations, this is actually a good thing. The seam is marking the true area between the two main buildings that actually should be carved out.
- We get better results with blur in this image because it allows the seam to overlay the bush(less important) than the building(more important). It shows that blurring can sometimes be better for dynamic programming to catch the less important parts of an image.

6. Goal: Show content resizing is better than blind resizing

a) the original input image,
b) your system's resized image,
c) the result one would get if instead a simple resampling were used (via Matlab's imresize),
d) the input and output image dimensions,
e) the sequence of enlargements and removals that were used, and
f) a qualitative explanation of what we're seeing in the output.

Good Example
Image Source
(https://www.stuff.co.nz/entertainment/film/101132197/black-panther-director-ryan-coogler-on-his-black-superhero?fbclid=IwAR1mVjVD54VhhTGdGATL9Gwug9d6phtdvOzUl5Vc5rLLoMWlSRWHNmtCLUs)

original

our system

resize

Original (168x299) (h x w)
Our Image (158x145)
Resize Image (158x145)
The sequence we used: 150 vertical seams removal, 10 horizontal seams removal
Explanation: Our system clearly outperforms resize in this image. The resize condenses and distorts the facial structures of Chadwick Boseman whereas our system keeps his body defined and symetrical.

Good example
Image Source
(http://www.essexwolverine.com/sport/0/1.php?fbclid=IwAR0h_30YaAMW9toMl_DnyYJAoHFe3DEyFvx0t8fJrENsLcHEZp4qzaAmnw0)

Original Image (600x1140)
Our Image (590 x 990)
Resize Image (590 x 990)
The sequence we used: 150 vertical seams removal, 10 horizontal seams removal
Explanation: Our system does better than resizing in this because it preserves the shape of the main content. The ball is still mostly round in our image, displaying its efficacy over resizing image. The resizing image squishes the ball and makes the image look demented.

Bad Example: Letters and non-uniform images

Image Source
(https://www.ucdavis.edu/admissions/?fbclid=IwAR3xYNZejgoCmFT3s8gGfdse233VjmhABNHnHHW5qOZwfjrGo1xybRdkPEI)

original

our system

resize

Original Image: (460 x 2000)
Our Image: (450 x 1850)
Resize Image: (450 x 1850)
The sequence we used: 150 vertical seams removal, 10 horizontal seams removal
Explanation: We decided this picture was a bad example for our system because the ‘W’ and the ‘O’ get thinner and we typically don’t want to change important content of an image. This happens because our system removes 150 vertical seams and 10 horizontal seams. These seams are based on energy levels which is essentially the level of change between pixel values. In this case, because we are not changing the image size drastically, so resized image looks fine. But our system, deletes the lowest energy seams, resulting in letters being destroyed.

EXTRA CREDIT: Greedy
Look at red Line

EXPLANATION - For our extra credit we chose to implement the greedy version of this algorithm. This was implemented by going to the lowest value in the cumulative energy map and picking the lowest value from there and then moving on to the next value in the next row that is either adjacent or diagonal from it. This is a greedy solution because it is less efficient but prioritizes getting the lowest value.