Computer Vision Assignment PS1

Darsh Thakkar
GT ID: 903438830

1 Short Answer Problem

1. To filter an image, one might need to apply multiple filters to the image. Let f1, f2 and f3 be 3 convolutions that need to be applied to image 'I'. Instead of applying each convolution individually to the image, one can create a convolution which is a combination of f1, f2 and f3 and directly apply it to image 'I' because convolutions follow associative property. This in turn is more efficient than applying each filter individually.

2. [0 0 1 1 0 0 1 1] $\oplus$ [1 1 1] = [0 1 1 1 1 1 1 1 1]

3. [0 0 1/4 0 -1/2 0 1/4 0 0]

4. (a) Controlling the sigma of the Gaussian filter (b) Thresholding (hysteresis) - high threshold

5. If Gaussian noise is used to represent an image noise than one has to realize that this image already has some noise which can be Gaussian and when two Gaussian signals are added with each other they do not return an arithmetic sum but instead return the sum in the following form:

$$\sigma_{total} = \sqrt{\sigma_1^2 + \sigma_2^2}$$

   This can be a possible flaw as the total noise turns out to be less than the arithmetic sum of the Gaussian signals which might affect the results of image processing.

6. In order to analyze the video, video itself will be divided into image captures such that we can analyze images rather than analyzing video as a whole.

   Next plan of action would be to figure out the noise that might be present in the camera such as schottkey noise which comes from the schottkey diode in the camera, once this is identified, I would apply filters to the input images which would reduce noise in the image.

After this, next thing that can be done is to take images of the background excluding the assembly parts and then take the image with the assembly part. Subtract the first image that is the image without assembly parts from the second image that is the image with assembly parts and we would then get an approximation of how the assembly parts look. We can do this several times and get true examples of when there is a flaw in assembly and also when there is no flaw in the assembly and store these examples.

After this, we can convert the images into a binary images and subsequently perform erosion to separate out the assembly parts and the finer details of the image, moving on dilation can be applied as erosion might make the image elements too small and thus dilation will help in getting good sizable blobs.

7. Since we have true examples we can compare the images with and without flaw to see how they differ, for instance if flaw can be detected by volume, we can use number of pixel difference in blobs or in some cases intensity of the blob can also be used to evaluate presence/absence of flaw in the image.

2 Programming Problem

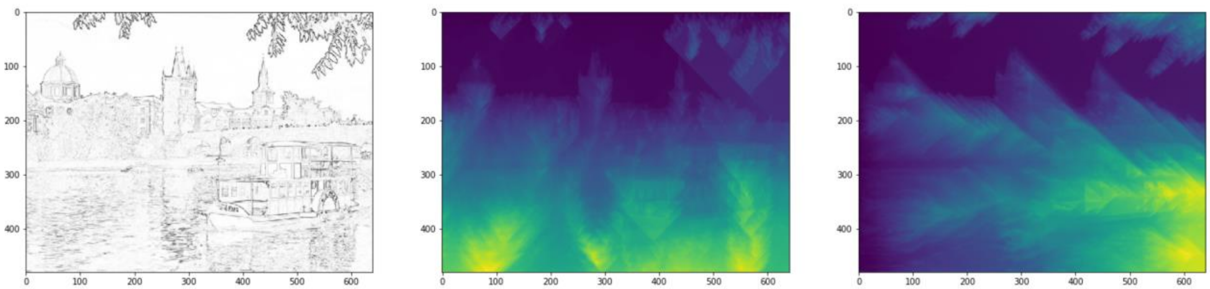
1.

(c)

(d)

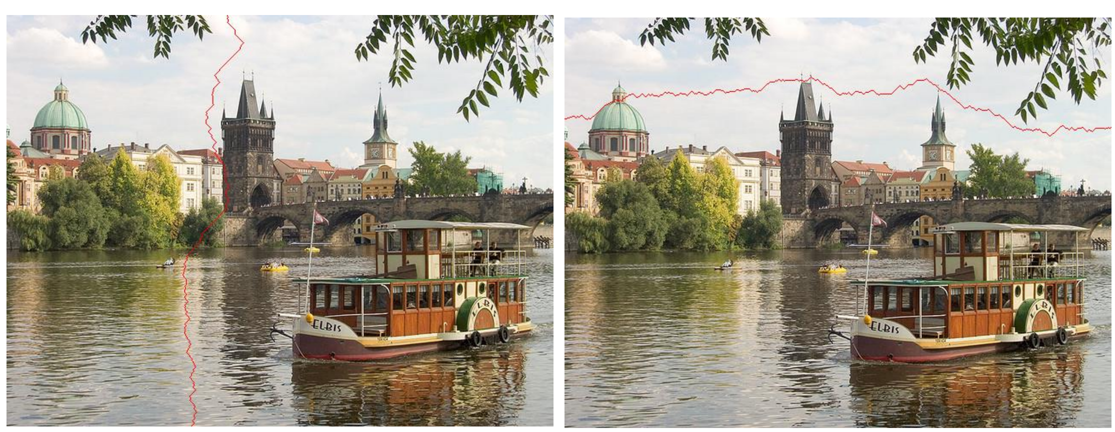
2.

3. Following is the (a) the energy function output for the provided image inputSeamCarvingPrague.jpg, and (b) the two corresponding cumulative minimum energy maps for the seams in each direction – vertical and horizontal respectively.



The output of the energy image tries to capture the gradient residing in the image after converting it into a grey scale image and calculating the energy based on the derivatives in x and y direction, thus the gradient of the image is clearly visible in this. For the second image, we see the color changing from top to bottom because the cumulative energy mapped onto the 2d matrix from the image increases in that order and lower the cumulative energy (dull shade), higher chance of that seam being removed in order to have a content aware resizing. Similar analogy is followed by the third image but instead of top to bottom, the pattern here is left to right.

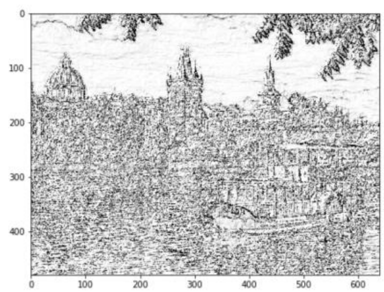
4. First image displays the optimal vertical seam and the second image displays the optimal horizontal seam.

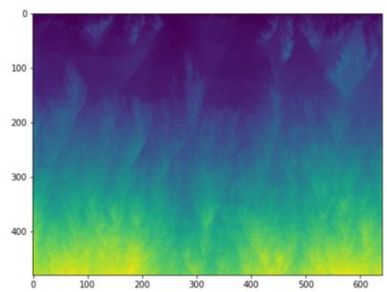


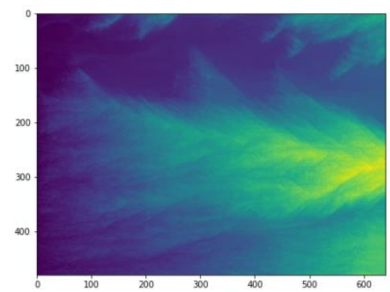
The seams showed in this image are optimal because they produce the lowest cumulative energy, following the 8-connectedness analogy among all other seams, horizontal or vertical. These semes are found using the cumulative minimum energy map.

5. Input Image – inputSeamCarvingPrague.jpg

Images below are the output of energy map function, vertical cumulative energy map and horizontal cumulative energy map respectively, for the modified energy function.







Energy map is calculated for inputSeamCarvingPrague.jpg using the sobel filter and without converting the image into grey scale. The difference here is that here all 3 that is RGB values of the image are not treated in form of weighted average but instead energy is calculated for each RGB value of a pixel and then added together. This leads to a more detailed gradient which ultimately only gives an idea about high level shapes as the finer details become so busy that it gets difficult to differentiate between objects. Results can also be seen in the cumulative minimum energy map as compared to previous function the shade of colors is a bit brighter, indicating that energy map had higher values than what the result was previously.

6.

I. (a) Original input image

(b) System's resized image

(c) Simple resized image

(d) Input: [638,343] Output: [588,293]

(e) reduceHeight, reducewidth (x50). Alternatingly removing total 50 pixels from height as well as width.

(f) The original image contains many geometrical line elements and thus when it is resized using reduceHeight function, which in turn removes horizontal seams, several lines straight lines are distorted which can be seen in image (b) on the middle left side. Thus, when the image has such elements which require geometric precision in resizing may not do well depending on the input. When resized using simple resizing such problem is not faced as seen in image (c).

II. a) Original input image

(b) System's resized image

(c) Simple resized image

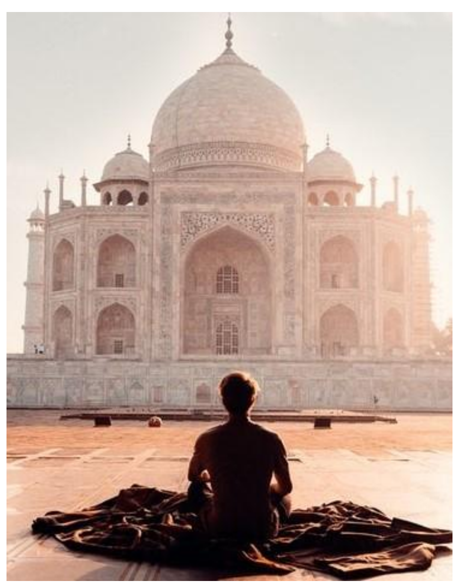
(d) Input: [640,873] Output: [540,623]

(e) reduceHeight (x250), reducewidth (x100). Removing total 250 pixels from height and 50 pixels from width.

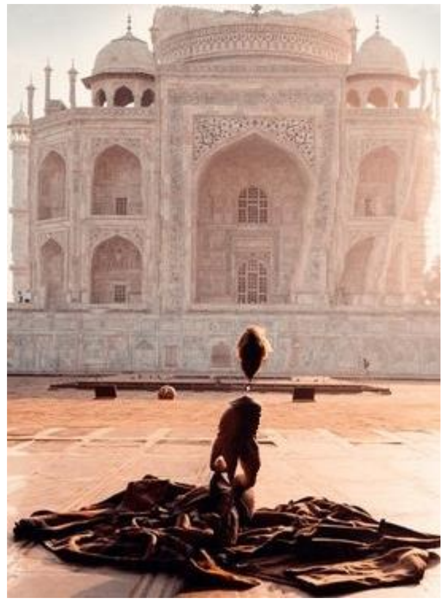
(f) This example is perfect to illustrate a good output from content aware image resizing. In image(b) we can see that the original image is resized by removing horizontal semes above the balloon and bellow the balloon but in image (c), the balloon is squashed in some way as well as the segment of hill and ground becomes lower in terms of image occupancy.

III.

(a) Original input image



(b) System's resized image



(c) Simple resized image

(d) Input: [371,473] Output: [271,373]

(e) reduceHeight, reducewidth (x100). Alternatingly removing total 100 pixels from height as well as width.

(f) In this example when the original image is resized using simple resizing, the image received seems like a good output and there aren't much irregularities or distortions observed. However, when the image is resized using content aware image resizing, the algorithm messes up and the resultant image is not correct with several important parts of image being removed. Here, the energy maps derived lead to removal of essential seams.