PS1

Christopher Banks

September 11, 2019

1 Short Answer Problems

1. An individual can use the associative property of convolution to speed up the computation time for calculating the filtered image from N^2 operations per pixel to $2K$ operations. For instance, if you wanted to use a box filter you could find the corresponding 1D horizontal filter $h^T = \frac{1}{K}[1, 1, \ldots, 1]$, find the convolution of your image, f, with it such that $c_h = f * h^T$ is the horizontal convolution. This, in turn, can be convolved by the vertical box filter, v to return the image, g, after the full box filter has been applied where $g = v \star c_h$. This approach utilizes the associative property of the convolution operator and uses less operations than fully multiplying by the matrix form box filter.

2. The resulting image is [01111111].

3. $f'' = [1/4 \quad 0 \quad -1/2 \quad 0 \quad -1/4]$

4. Using the Canny edge detector one could apply hysteresis thresholding to find high and low threshold values to find edges that satisfy the high threshold value and are above the low value and are connected within an image. You could also, use non-maximal suppression to determine the local maximum pixel along the gradient direction on the edge.

5. If someone is using median filtering for an image, the use of Gaussian noise will detract from the efficiency of this filtering technique.

6. The following algorithm can be used for identifying faulty parts in an assembly line:

   (a) First, generate a sample set of ideal machine parts in order to generate a reference set of feature intensity distributions from a feature bank of filters for upcoming parts.

   (b) This feature bank will contain filters (e.g. Gaussian) of varying orientations and scales of x & y derivatives.

(c) In addition to this, the algorithm also takes advantage of k-means clustering for color variations and matches known machine parts to the data it has been trained on based on the sample data characteristics.

(d) Parts that are not within the bounds of the feature filter bank intensity distributions and cannot be matched to a proper cluster with some high threshold will be labeled as defective.

(e) Assumptions: This algorithm assumes enough "correct" machine parts are available to create an accurate k-means clustering pool and a representative distribution for the various intensities possibly represented in the feature bank. We also assume that ideal parts can be characterized by a bounded distribution and a k is known for the k-means portion.

2 Programming Problems

1. SeamCarving Width

(a) Prague, Width Reduced by 100 pix.

(b) Mall, Width Reduced by 100 pix.

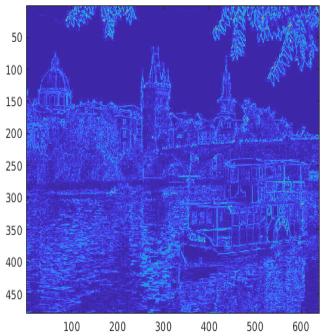
2. SeamCarving Height

3. Energy Function and Cumulative Images : The energy function image shows that there are high gradients at some of the boundaries with the water and surrounding distinctive objects. The cumulative energy (vertical) shows a high amount of desirable energy in the bottom of the image, most likely corresponding to the high gradient and irregular flow of the water. The cumulative energy (horizontal) image I generated appears incorrect and looks very similar to the energy function, which I'm not sure is right.

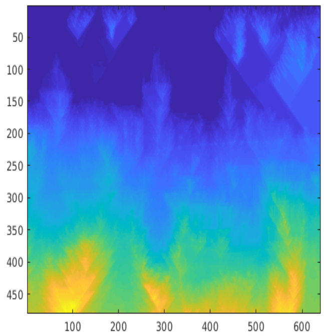
4. Seam Carving, First Iteration : Using the optimal subproblem search to find the lowest energy pixels and then recursing backwards we are looking for low energy segments in the image. The very top and the far left have little gradient change

(a) Prague, Height Reduced by 100 pix.

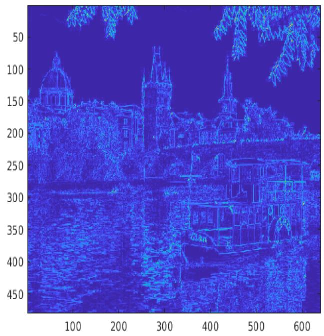
(b) Mall, Height Reduced by 100 pix.



(a) Energy Function



(b) Cumulative Energy (vertical).



(c) Cumulative Energy (horizontal).

and do not change as drastic as the boat on the water or the horizon, explaining why these are the first columns and rows to go.

(a) Initial Image

(b) First Seam Carves (horiz, vert)

5. I used a Sobel gradient instead of the traditional gradient for the Energy function calculation. This gave better selection of the buildings but more of the water was deemed "low energy" and cropped out.

Figure 1: By applying a Sobel gradient to the image we get better contours for buildings with sharp edges but the water seams are sacrificied.

6. Different Images w/ Seam Carvings

(a) input image (b) Seam Carving w/ Sobel (c) matlab imresize, scale=0.5

Figure 2: Here is a flower example with pretty bad Seam Carving Results, Fig. (a) shows the input image with size 266 x 190. With Fig. (b) I use the Sobel gradient for the energy function and use Seam carving (x 50) on width and Seam Carving (x 100) on height with a resulting image size of 216 x 90. The gradient information may not have been significant enough for the Seam Carving algorithm to properly identify good segmentation points.

(a) input image (b) Seam Carving w/ Sobel (c) matlab imresize, scale=0.8

Figure 3: This is an image of Seam carving done on a street view in Macau, China. The Fig. (a) is the input image with size 960 x 640. In Figure (b) I use the Sobel gradient for the energy function and use Seam carving (x 100) on width and Seam Carving (x 100) on height with a resulting image size of 860 x 540. The final image is a matlab resize with scale = 0.8. From the images, we can see that there is a bit more stretching of objects in the Seam Carved image, yet I keep more of the image in the Seam Carved image than that from Matlab's image.

(a) input image

(b) Seam Carving w/ Sobel

(c) matlab imresize, scale=0.8

Figure 4: This is an image of Seam carving done on a Google image face of an African Lady. The Fig. (a) is the input image with size 1200 x 676. In Figure (b) I use the Sobel gradient for the energy function and use Seam carving (x 100) on width and Seam Carving (x 100) on height with a resulting image size of 1100 x 576. The final image is a Matlab resize with scale = 0.8. This carving, while preserving the features of the lady, does a lot of stretching which doesn't look that natural.

3 Extra Credit