CIS4930 – Programming Assignment #2; Due Date: 02/27/2018

Submit your code via CANVAS. Also submit a TYPED REPORT that contains, concise discussion on your observations from the results and any reasoning necessary to explain your approach. The report should contain, results that are arranged to show the input image and the output images, if and when applicable. Try to minimize your page usage by displaying at least 3 images per row on a page. All your image displays must have a caption.

1. Phase Reconstruction: (a) [5 points] Write a MATLAB program that takes the FFT (use MATLAB FFT routine) of an input image (Century-Tower.jpg), hard limits (forces) the magnitude of the FT to 1 and reconstruct the the phase-only image. Note that every complex number can be represented in polar form as \( r \exp(i\theta) \), where, \( r = \sqrt{x^2 + y^2} \) is called the magnitude and \( \theta \) is called the phase. (b) [5 points] Repeat the reconstruction for magnitude only case i.e., zero out the phase in the FT and reconstruct the image. (c) Now perform the following tests: (i) [5 points] add uniform random noise to the magnitude \( \|I(u,v)\| \) but do not change the phase and reconstruct the image, (ii) [5 points] add uniform random noise to the phase but do not change the magnitude and then reconstruct. (d) [10 points] Compare the contents of the reconstructed images in c-(i) and c-(ii) in terms of perceptual clarity of meaningful image content. Display all the 3 images (from c-(i), c-(ii) and the phase only reconstruction with no noise in (a) ) side by side for comparison. Discuss concisely your observations about reconstructions.

2. For this problem, you will implement the seam carving algorithm based on dynamic programming described in the paper titled, “Seam carving for content aware resizing,” by Shai Avidan and Ariel Shamir in SIGGRAPH 2007. The paper has been posted on the class web site. Read the paper with emphasis on sections 3, 4.1 and 4.3. You will write MATLAB code for performing the following tasks: (i) compute the gradient based energy function at each pixel defined in the paper. (ii) Compute the optimal vertical seam in the given input image. (iii) Compute the optimal horizontal seam in the given input image. (iv) Reduce the image size by the specified amount in one-dimension (height or width). (v) Overlay the optimal seam on the original image. (vi) You should write the functions with the following interfaces: [output] = reduceWidth(im, numPixels) and [output] = reduceHeight(im, numPixels). These functions have two input parameters, im: the input image and numPixels: number of pixels to reduce the width or height by. put these functions in reduceWidth.m and reduceHeight.m.

Set up scripts so that you can remove different combinations of horizontal and vertical seam removals. Test your code on the supplied images and display the results in color. Note that the image gradients have to be computed using the grayscale equivalent (converted) image.

- MATLAB functions for possible use in this problem: imfilter, im2double, fspecial, imread, imresize, rgb2gray, imagesc, imshow, subplot;
- To plot points on top of a displayed image, use imshow(im); followed by hold on; followed by plot().
- Exercise caution with double and uint8 conversions as you go between computations with the images and displaying them filtering should be done with doubles.

Answer each of the following questions and include image displays where appropriate.

(a) (15 points) Run your reduceWidth and reduceHeight functions on Century-Tower.jpg and alligator-crossing.jpg respectively, with numpixels=100 i.e., shrink the width by 100 pixels in the first case and shrink the height by 100 pixels in the second case respectively.
(b) (15 points) Display (a) the energy function output (total gradient magnitudes $e_1(I)$) for the image Century-Tower.jpg, and (b) the two corresponding cumulative minimum energy maps (M) for the seams in each direction (use the `imagesc` function). Explain why these outputs look the way they do given the original images content.

(c) (40 points) This is the real thing!! Use your system with different kinds of input images and seam combinations, and see what kind of interesting results can be produced. The goal here is to make some visually pleasing outputs where the resizing better preserves content than a blind resizing would, as well as some examples where the output looks unrealistic or has artifacts.

Include results for at least three images of your own choosing. Include an example or two of a bad outcomes. Be creative in the images you choose, and in the amount of combined vertical and horizontal carvings you apply. Try to predict types of images where you might see something interesting happen. Its ok to fiddle with the parameters (seam sequence, number of seams, etc) to look for interesting and explainable outcomes.

For each result, include the following items, clearly labeled (the title function may be useful here): (a) the original input image, (b) your systems resized image, (c) the result one would get if instead a simple resampling were used (via MATLABs `imresize`), (d) the input and output image dimensions, (e) the sequence of enlargements and removals that were used, and (f) a brief qualitative explanation of what were seeing in the output.