CS376: Computer Vision: Assignment 1
Due: Feb. 15th, 11:59 PM

Format for writeup: You may use any tool for preparing the assignment write-up that you like, as long as it is organized and clear, and figures are embedded in an easy to find way alongside your descriptive text.
Submission: See the end of this document for submission instructions.
Assignment questions: Please see Piazza for questions and discussion from the class.

1 Short answer problems [30 points]

1. What are the sources of image noise?

2. Consider the filter $f = [1, 2, 1]$ and the 1D image $I = [0, 3, 2, 3, 4, 5, 6, 1, 0]$. What is the result of $f * I$? Pad the image with zeros at the boundaries if necessary.

3. Please specify two 2D filters that can effectively compute the image gradients along the x-direction.

4. When computing image gradients, we first smooth the image and then compute the image gradients. What will happen if we compute the image gradient first and then smooth the resulting image gradients?

5. In non-maximum suppressing, we detect the maximum pixel along the image gradient direction. Provide examples where this approach is sub-optimal. You can draw illustrations or provide results on real examples. Please provide a short justification (2-3 sentences) on why this is the case.

Extra credit (5points). So far we have covered filtering and edge detection for images. Please mention how to extend the idea to videos. Please discuss how to de-noise in both the spatial and/or temporal domain, how to compute gradients in the spatial and/or temporal domain, and how to detect "edges" in the spatial and/or temporal domain.

2 II. Programming problem: content-aware image resizing [70 points]

For this exercise, you are asked to implement a version of the content-aware image resizing technique described in Shai Avidan and Ariel Shamir's SIGGRAPH 2007 paper, "Seam Carving for Content-Aware Image Resizing" (See Figure 1 for one result). The goal is to implement the method, and then examine and explain its performance on different kinds of input images.
As we discussed in this class, the key idea of this approach is to use dynamic programming to select the optimal seams to insert/delete to modify an input image.
Write Matlab code with functions that can do the following tasks:

Figure 1: A typical result of seam carving. (Left) Input image. (Right) Output image.

- Compute the energy function at each pixel using the magnitude of the x and y gradients (equation 1 in the paper)
- Compute the optimal vertical seam given an image
- Compute the optimal horizontal seam given an image
- Reduce the image size by a specified amount in one dimension (width or height decrease)
- Display the selected seam on top of an image
- Functions with the following interface:

  $$[\text{output}] = \text{removeVertical}(\text{im}, \text{numPixels})$$

  $$[\text{output}] = \text{removeHorizontal}(\text{im}, \text{numPixels})$$

  These functions take an input image im, and a parameter specifying how many seams to carve, from the width or height,respectively. The image im will be a $h \times w \times 3$ uint8 matrix, which is what imread returns for a color image.Put these functions in file named removeVertical.m and removeHorizontal.m

Set up scripts so that you can play with the seam removal and specify different combinations of horizontal and vertical removals. Apply your system to the provided images. View the results in color, but note that the gradients should be computed with the grayscale converted image.
Matlab hints:

- Useful functions: 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(...)".
- Be careful with double and uint8 conversions as you go between computations with the images and displaying them–filtering should be done with doubles.
- Use toy examples to debug! For example, create an image with all Gaussian noise, while leaving three vertical seams right. Your result shall return that vertical seam. As another example,

Answer each of the following, and include image displays where appropriate:

- (10 points) Run your reduceVertical function on the provided ut.jpg with numPixels = 100 (in other words, shrink the width by 100 pixels). Run your reduceHorizontal function on the provided river.jpg with numPixels = 100 (in other words, shrink the height by 100 pixels). Display the outputs.

- (10 points) Display (a) the energy function output (total gradient magnitudes $e_1(I)$) for the provided image ut.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 image's content.
- (10 points) For the same image ut.jpg, display the original image together with (a) the first selected horizontal seam and (b) the first selected vertical seam. Ex plain why these are the optimal seams for this image.
- (10 points) Make some change to the way the energy function is computed (i.e.,filter used, its parameters, or incorporating some other a priori knowledge). Display the result and explain the impact on the results for some example.
- (30 points) Now, for the real results! Use your system with different kinds of images and seam combinations, and see what kind of interesting results it can produce. The goal is to form some perceptually 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. Be creative when generating the results. Look into the original seam carving for inspiration. Note that you are allowed to hand-craft the feature map, e.g., by removing an object or prevent pixels of an object from removing.

For each result, include the following things, clearly labeled (the title function may be useful here):

- (a) the original input image,
- (b) your system's resized image,
- (c) the result one would get if instead a simple resampling were used (this is considered the baseline comparison),
- (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 we're seeing in the output.