Lab 2
CSC 5930/9010 - Computer Vision
Grading: 50 points
Due Date: Sept. 28, 2016

Lab Description:

Seam-carving is a content-aware image resizing technique where the image is reduced in size by one pixel of height (or width) at a time. A vertical seam in an image is a path of pixels connected from the top to the bottom with one pixel in each row; a horizontal seam is a path of pixels connected from the left to the right with one pixel in each column. This lab is meant to reinforce the basic data structures, variables, and methods that you have learned thus far. You must have the image processing toolbox from Matlab installed.

(a) Original image. (b) Uniform scale. (c) Seam Carving.

Figure 1: Example of vertical seam carving (smart resizing) on an image.

The image shown above shows an example of seam carving vs a uniform scaling operation. Notice that with a uniform scale, the salient objects of the image become distorted whereas the seam carving resizing intelligently shrinks the image.

Step 1. Create a function called "seamcarve" that returns nothing and takes as input a variable "imname" that corresponds to the string filename of the image to resize and an input variable "iters" that represent the number of columns you would like to remove. Read in the image using imread and call it "im".

$>> im = imread(imname);$

Step 2. Energy calculation. The first step is to calculate the energy of a pixel, which is a measure of its importance - the higher the energy, the less likely that the pixel will be included as part of a seam. You can compute the gradient magnitude and gradient direction by the function "imgradient". Notice that I compute the magnitude on the grayscale (1 dimensional) image, not the RGB image.

$>> [gmag, gdir] = imgradient(rgb2gray(im));$

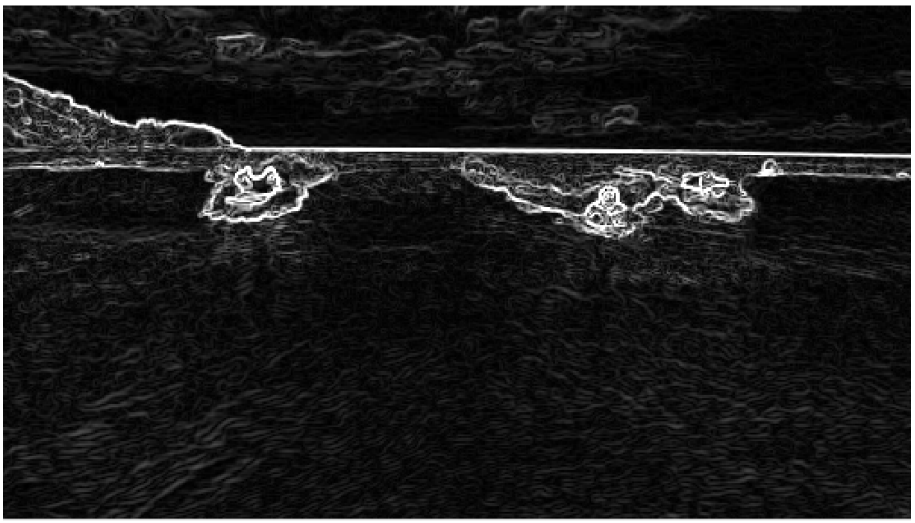

Figure 2: Example of the gradient computation on an image.

Step 3. Seam identification. The next step is to find a vertical seam of minimum total energy. This is similar to the classic shortest path problem in an edge-weighted digraph, but there are three important differences:

- The weights are on the vertices instead of the edges.
- The goal is to find the shortest path from any of the W pixels in the top row to any of the W pixels in the bottom row.
- The digraph is acyclic, where there is a downward edge from pixel (x, y) to pixels (x - 1, y + 1), (x, y + 1), and (x + 1, y + 1). assuming that the coordinates are in the prescribed ranges.

Seams cannot wrap around the image (e.g., a vertical seam cannot cross from the leftmost column of the image to the rightmost column).

Step 4. Compute the seam that has the global lowest energy. Use a dynamic programming approach to find the globally optimal seam. My suggestion is to find the current size of the image using the "size" command.

$>> [h, w, d] = size(im);$

Create 2 more 2-D matrices that store the total cost of the seam up to that point, and a backtracking matrix that stores which pixel is the next one in the optimal seam i.e. if i is your x position in the image and j is your y position, then the backtrack(i,j) would store either j-1, j, or j+1. Remember the edge cases where j=0 or j=width.

Step 5. Seam removal. Once you reach the top of the image, find the minimum energy from your total energy matrix. Use the backtrack matrix to follow the seam down. You may want to highlight the pixels in red by setting the R channel of these pixels to 255. Then, remove all of the pixels along the vertical seam.
You can begin but creating a 2D image that has 1 less column,

$>> newim = zeros(h, w-1, 3);$

You can remove a column element in a vector in the following way,

$>> tvec = im(1, :, :)$; grabs the first row, 3 channels
$>> tvec(:, somecolumn, :) = []$; deletes some column from the vector
$newim(1, :, :) = tvec$; sets the newim as the n-1 vector

finally show the original image, pause, then show the new image. I'm magnifying by 200 percent.

$>> imshow(im, 'InitialMagnification', 200);$
$>> pause(0.001);$
$>> newim = uint8(newim);$
$>> imshow(newim, [], 'InitialMagnification', 200);$

Deliverables: Submit on Blackboard.