

Machine Learning and Computer Vision Assignment 1

Huihuang Zheng, huihuang@utexas.edu

hz4674 Fall 2015

1 Part I, Short Answer Problems

1. Suppose we have filter $f_1 \in \mathbb{R}^{1 \times n}$, $f_2 \in \mathbb{R}^{n \times 1}$. We have convolution $f_1 * f_2 \in \mathbb{R}^{n \times n}$. Suppose we have a picture g of $k \times m$ pixels. Time complexity of performing $(f_1 * f_2) * g$ is $O(kmn^2)$. That's because $f_1 * f_2 \in \mathbb{R}^{n \times n}$. But the time complexity of performing $f_1 * (f_2 * g)$ is $O(kmn)$. That's because size of $(f_2 * g)$ is still $O(k \times m)$ and performing the $(f_2 * g)$ needs $O(kmn)$. Time complexity of perform $f_1 * (f_2 * g)$ is $O(kmn)$.

A simple example: $f_1 = [1, 2, 1]$, $f_2 = [1, 2, 1]^T$, $g = \begin{pmatrix} 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 5 & 5 & 5 & 5 \\ 6 & 6 & 6 & 6 & 6 \end{pmatrix}$. $f_1 * f_2 =$

$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$. And we let output size equals g 's size. Performing $(f_1 * f_2) * g$ needs $9 * 25$ multiplications. Performing $h = f_2 * g$ needs $3 * 25$ multiplications and $f_1 * h$ needs $3 * 25$ multiplication. $f_1 * (f_2 * g)$ needs total $6 * 25$ multiplications.

2. $[0, 1, 1, 1, 1, 1, 1, 1]$
3. From filter f' , $[x_1, x_2, x_3, x_4, x_5]$'s first derivative is $(x_4 - x_2)/2$. So second derivative is $(\frac{x_5 - x_3}{2} - \frac{x_3 - x_1}{2})/2$. So $f'' = [\frac{1}{4}, 0, -\frac{1}{2}, 0, \frac{1}{4}]$.
4. 1. Using higher threshold. 2. When smoothing with Gaussian. Using larger σ value.
5. The Gaussian noise is hard to generate some contiguous pixels of noise. For example, some times we need $3 * 3$ pixels noise with all values are same. It's hard to be generate by Gaussian noise.
6. Assumption: the objects will be transferred periodically on conveyor belt during work time. The camera will be fixed, it doesn't move around the factory. My camera will work as following:

- (a) Perform edge detect on the video the camera takes.
- (b) Using filter on the edges to recognize conveyor belt and objects on it.
- (c) If objects out of conveyor, or don't remove on the conveyor, it reports error: object falls or the conveyor belt stops.
- (d) If camera hasn't seen objects on conveyor for a long time, it reports error: the conveyor belt may be stopped.

2 Part II, Programming problem: content-aware image re-sizing

Usage: just run *main.m* in matlab. It will run all sections of part 2 and generate codes in *./output_images/*. If you would like to run just some questions of part 2, just use *Part2([the question numbers in part 2])*. For example, you want to see my code result for question 2.1, 2.5, just run *Part2([1, 5])* and see results in *./output_images/2.1* and *./output_images/2.5*.

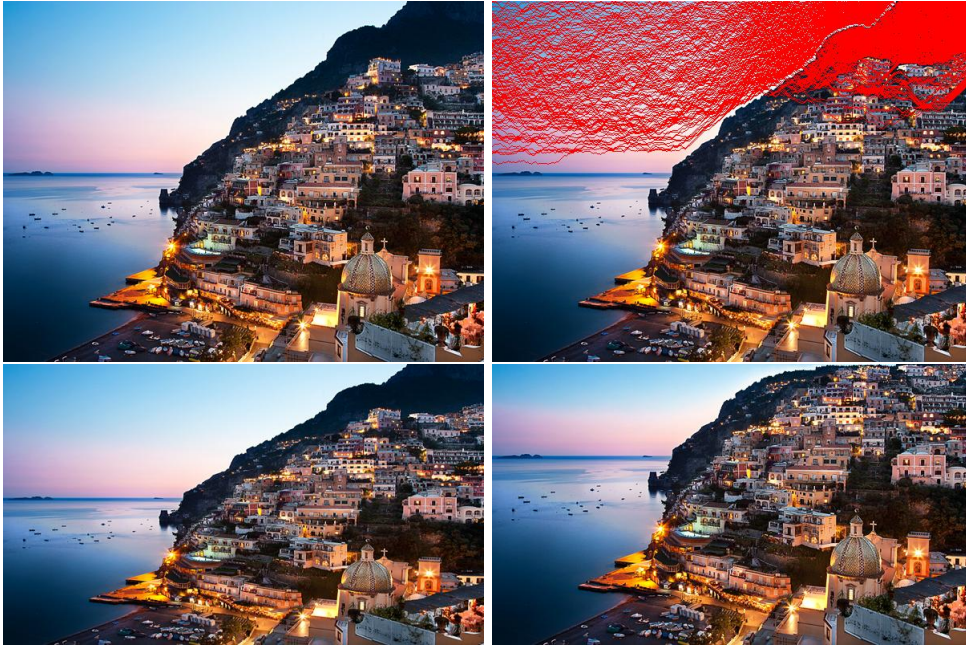
Implement Summary: the *main.m* call *Part2()* and *extra()* for questions in Part2 and extra functions. The main work in this project was done in *SeamCarvingImage.m*. It defines a class *SeamCarvingImage*, which compute derivatives, energy, seams and offer functions for decrease/increase height/width and remove object selected by GUI. If you want to test these functions using your images. There are several basic functions:

1. `[output] = reduceHeight(im, numPixels), [output] = reduceWidth(im, numPixels);`
2. `[output] = increaseHeight(im, numPixels), [output] = increaseWidth(im, numPixels);`
3. `[output] = removeObject(input_img);`

`im` and `input_img` mean the image you input in matlab and `numPixels` means number of pixels you want to increase/reduce.

2.1

For *im1.jpg*



Top left: original image1. 600 * 450

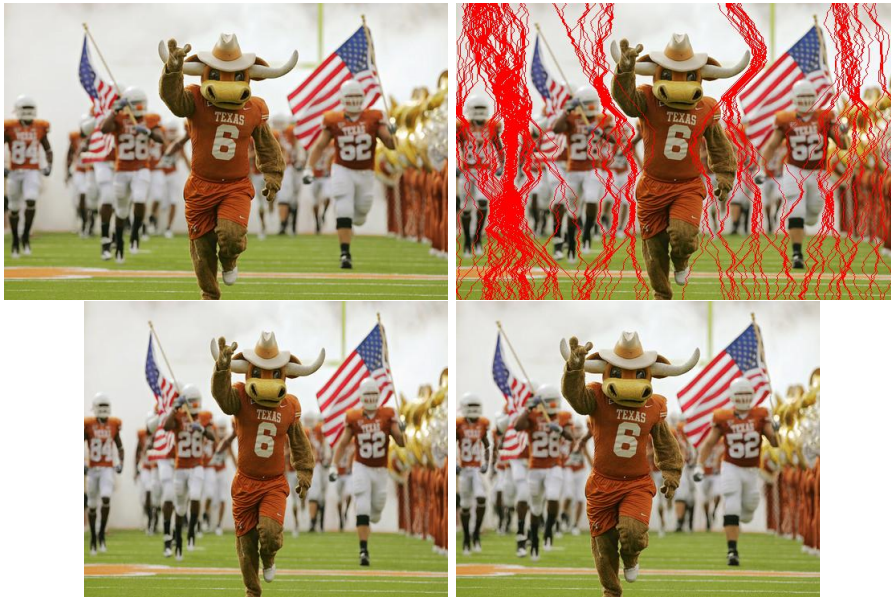
Top right: seams chosen.

Bottom left: resize to 600 * 350

Bottom right: seam carving reduce height to 600 * 350

We could see the seams chosen are mostly pixels of sky and mountain top. And seams miss the houses in the mountain. The seam reducing keep more information than resize but it changed the shape of mountain, causing some unrealistic.

For im2.jpg



Top left: original image2. 553 * 369

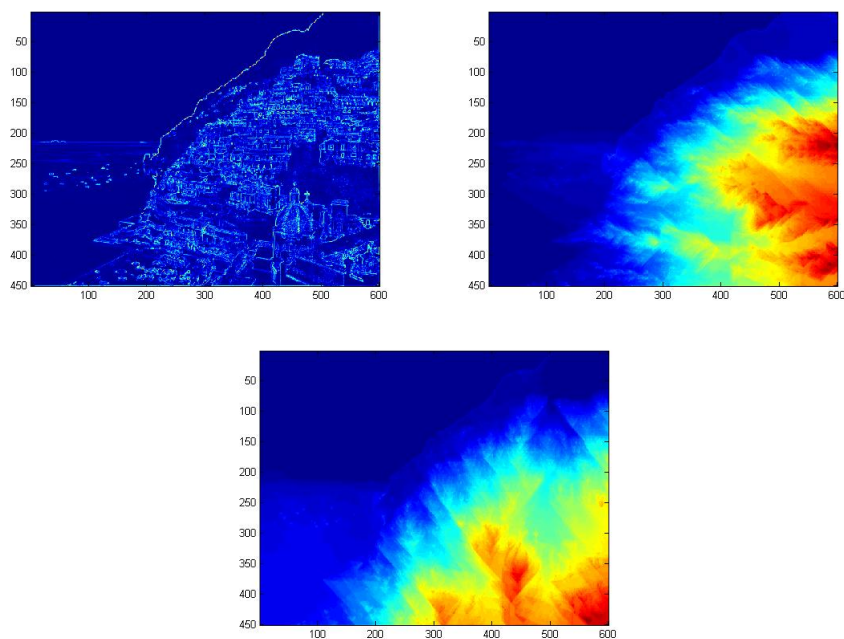
Top right: seams chosen.

Bottom left: resize to 453 * 369

Bottom right: seam carving reduce height to 453 * 369

We could see that most of seams are in the position of interspace between people. In this example we saw seam reducing keeps the sportsman in middle of the picture more really.

2.2



Top left: energy for every pixl.

Top right: cumulative energy for horizontal.

Bottom: cumulative energy for vertical.

Above 3 images show us the houses have more energy so the cumulative energy map has larger values in house pixels than values in sea or sky.

2.3



Above image shows the first horizontal seam and vertical seam my system chosen. It passes positions without large derivative.

2.4



Left: energy function as $energy = |\frac{\partial I}{\partial x}| + |\frac{\partial I}{\partial y^2}|$ (im2_rw_diff_energy_1.jpg)

Right: energy function as $energy = |\frac{\partial^2 I}{\partial x^2}| + |\frac{\partial^2 I}{\partial y^2}|$ (im2_rw_diff_energy_2.jpg)

I used second derivative as a test energy function. The reduced images look slightly different. The people in the right image are a litter fatter.