CS 435 - Computational Photography

Assignment 4 - Retargeting

Introduction

For this assignment we are going to implement some image retargeting approaches, allowing us to change the size and aspect ratio of an image. Techniques will include sub-sampling and seam carving.

Grading

Theory Questions	35pts
Image Resizing	15pts
Energy Matrix	10pts
Optimal Seam Discovery	20pts
Seam Removal	20pts
TOTAL	100pts

Table 1: Grading Rubric

1 (35pts) Theory Questions

1. Imagine that we want two blend two images such that in our target image there will be a circular region, centered at (a, b) with radius r, where the pixel values are taken solely from the first image, I_1, and outside of that, the contribution of the first image fall off according to a Gaussian with standard deviation σ. This Gaussian should have its maximum value of 1 at the edge of the radius. Or you can think of it as using the distance relative to the radius for the computation of the exponent of this function. What is the blending function f, such that given a pixel location (x, y), $f(x, y)$ will give us the blending factor according to the description above. Since there are essentially two parts to this function, your function may be piece-wise. Feel free to check your work in Matlab, but leave your answers in terms of fractions and exponents (10pts).

$$f(x,y) = \begin{cases} e^{-\dfrac{(x-a)^2+(y-b)^2-r^2}{2\sigma^2}} & (x-a)^2+(y-b)^2 > r^2 \\ 1 & (x-a)^2+(y-b)^2 \leq r^2 \end{cases}$$

2. Given the grayscale image, I, below, if the top left location in our image is $(x = 1, y = 1)$, what would be value of a pixel at location $x = 2.7, y = 3.1$, if we were to:

   (a) Use the nearest neighbor in I. (3pts)

   5

   (b) Perform bilinear interpolation (i.e a Manhattan distance as our distance function) using I. Leave your answer in terms of fractions (7pts)

$$\frac{\frac{2}{.8}}{\frac{1}{.8}+\frac{1}{.4}+\frac{1}{1.6}+\frac{1}{1.2}} + \frac{\frac{2}{1.6}}{\frac{1}{.8}+\frac{1}{.4}+\frac{1}{1.6}+\frac{1}{1.2}} + \frac{\frac{5}{.4}}{\frac{1}{.8}+\frac{1}{.4}+\frac{1}{1.6}+\frac{1}{1.2}} + \frac{\frac{1}{1.2}}{\frac{1}{.8}+\frac{1}{.4}+\frac{1}{1.6}+\frac{1}{1.2}}$$

$$\text{denominator} = \frac{1}{.8}+\frac{1}{.4}+\frac{1}{1.6}+\frac{1}{1.2} = \frac{5}{4}+\frac{5}{2}+\frac{5}{8}+\frac{5}{6} = \frac{125}{24}$$

$$\text{numerator} = \frac{2}{.8}+\frac{2}{1.6}+\frac{5}{.4}+\frac{1}{1.2} = \frac{205}{12}$$

$$f(p) = \frac{24}{125}\left(\frac{205}{12}\right) = \frac{410}{125} = \frac{82}{25}$$

$$I = \begin{bmatrix} 2 & 3 & 4 & 5 & 1 \\ 1 & 0 & 2 & 2 & 1 \\ 4 & 3 & 5 & 1 & 2 \\ 4 & 4 & 4 & 4 & 6 \\ 4 & 5 & 2 & 0 & 2 \\ 2 & 3 & 3 & 0 & 3 \end{bmatrix}$$

3. If the matrix below is the energy of image

$$\begin{bmatrix} 2 & 3 & 4 & 5 & 1 \\ 1 & 0 & 2 & 2 & 1 \\ 4 & 3 & 5 & 1 & 2 \\ 4 & 4 & 4 & 4 & 6 \\ 4 & 5 & 2 & 0 & 2 \\ 2 & 3 & 3 & 0 & 3 \end{bmatrix}$$

(a) Construct the optimal seam matrix if we assume vertical seams (10pts).

$$\begin{bmatrix} 2 & 3 & 4 & 5 & 1 \\ 3 & 2 & 5 & 3 & 2 \\ 6 & 5 & 7 & 3 & 4 \\ 9 & 9 & 7 & 7 & 9 \\ 13 & 12 & 9 & 7 & 9 \\ 14 & 12 & 10 & 7 & 10 \end{bmatrix}$$

(b) What is the optimal seam (5pts)?

$$\begin{bmatrix} 2 & 3 & 4 & 5 & \mathbf{1} \\ 3 & 2 & 5 & 3 & \mathbf{2} \\ 6 & 5 & 7 & \mathbf{3} & 4 \\ 9 & 9 & \mathbf{7} & 7 & 9 \\ 13 & 12 & 9 & \mathbf{7} & 9 \\ 14 & 12 & 10 & \mathbf{7} & 10 \end{bmatrix}$$

2 (15 points) Image Resizing

First grab two images of interest to you so that we can multiple example I/O of each approach. Next allow a user (either via prompt, parameter passing, or changing variable values) to provide a desired height and width.

We'll refer to the original height and width as h and w, respectively, and the new height and width as h' and w', respectively. Do this all in color! You may **not** use any Matlab functions (for instance, *imresize* or *interp2*) to do these changes for you.

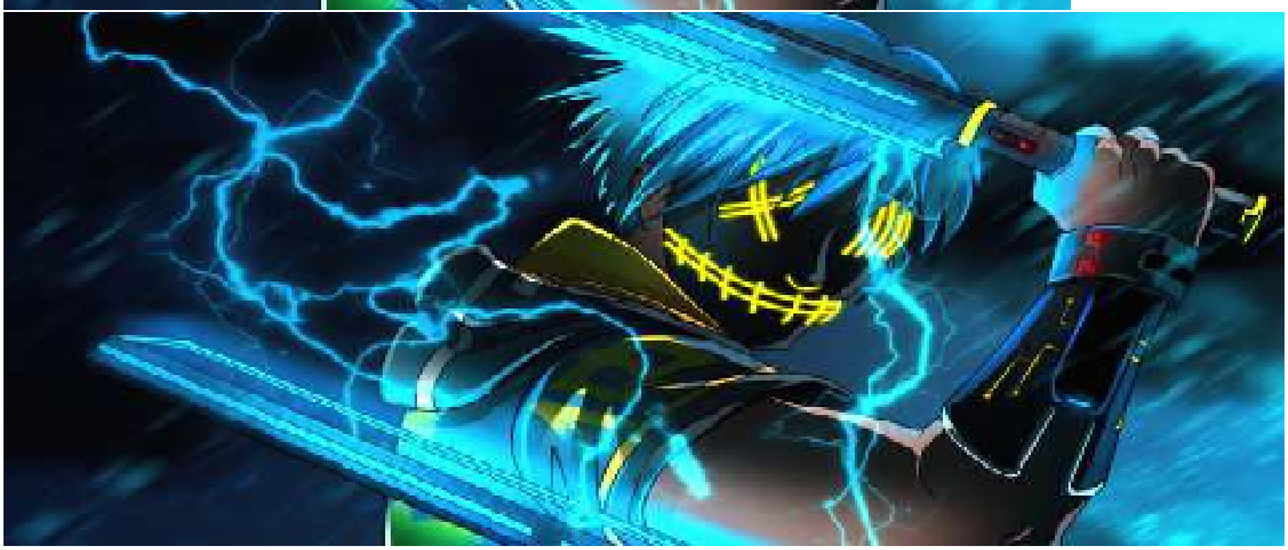
Implement the following resizing techniques:

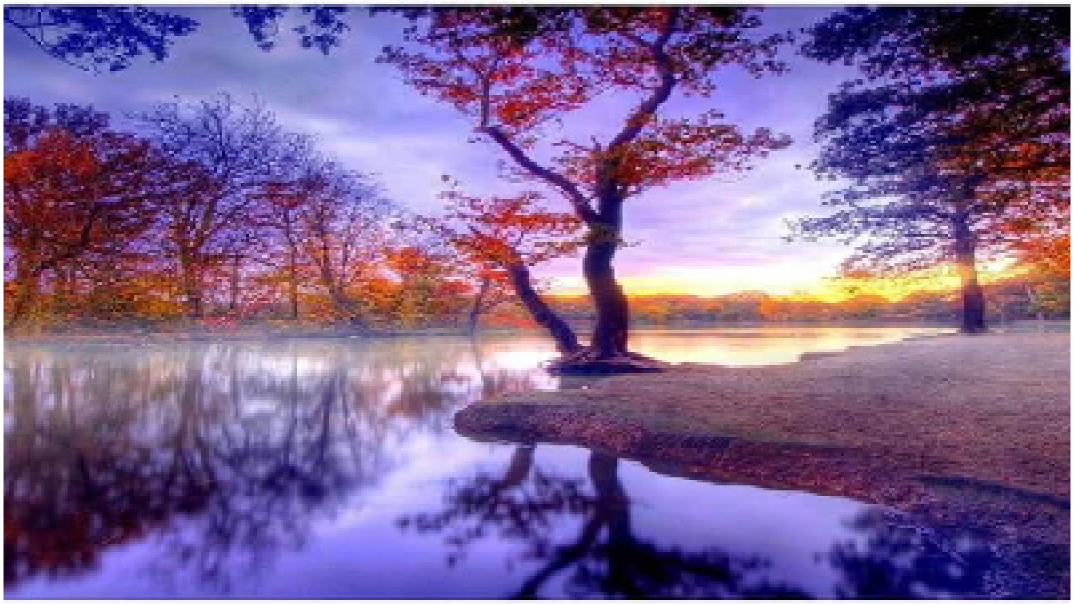
- Do nearest neighbor sampling. Go through each location in your new target image, (x', y'), and assign it the value of the nearest pixel, whose location is $(x, y) = round(x'\frac{w}{w'}, y'\frac{h}{h'})$.
- Do linear interpolation. Go through each location in your new target image, (x', y'), and compute the *ideal* floating point location as $(x, y) = (x'\frac{w}{w'}, y'\frac{h}{h'})$. Compute the values for location (x', y') by linearly interpolating the values of the four pixels nearest (x, y) (using Euclidean distance).

For your report provide sample I/O for both your test images, using both of these techniques, for two different values of (w', h') (8 total images).

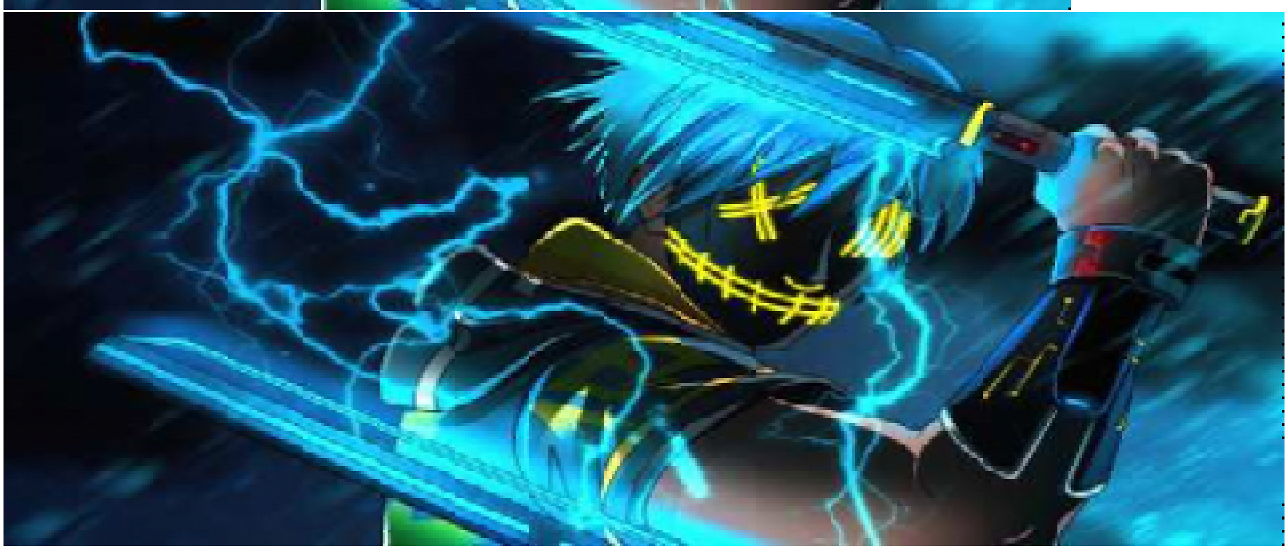
Original 1

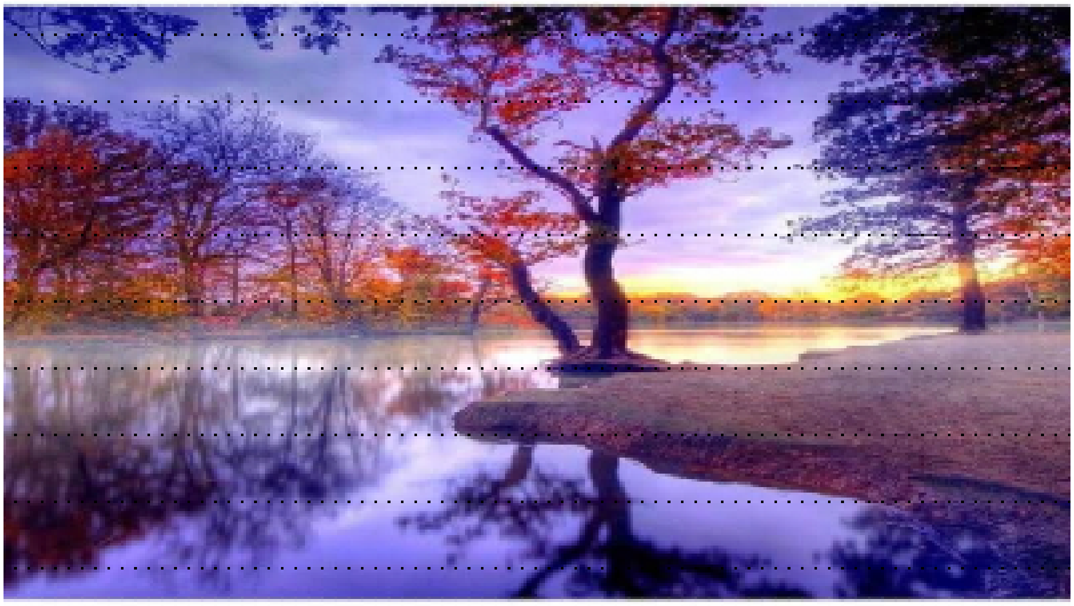
Original 2

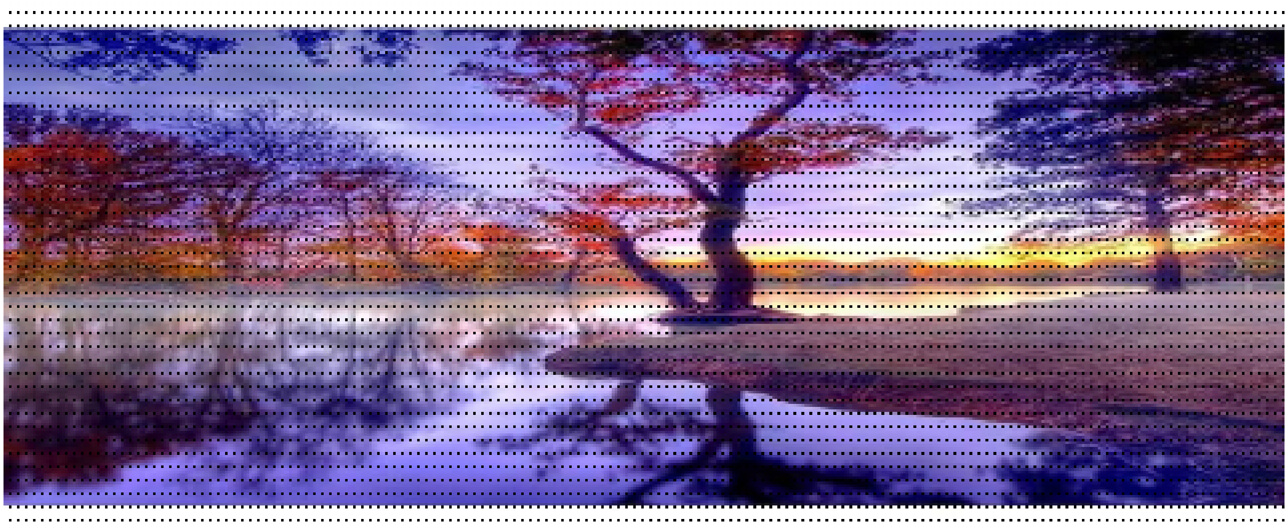




Nearest Neighbors using 250 x 400 and 200 x 480







Linear Interpolation using 250 x 400 and 200 x 480

3 (10 points) Energy Function

We're also going to want to explore seam carving. The first part of that requires computing the energy functions of your images.

For each of your images, compute its energy and visualize this as an image.

Some implementation details:

- Do this in grayscale
- First smooth your grayscale image using a Guassian filter prior to getting the gradients (or do this in one step). Choose parameters that make sense for you (and report them!).

Since you already demonstrated in prior assignments your to implement RGB to Gray, Gaussian and Gradient kernels and convolution, for this assignment **may** use Matlab functions like $conv2, rgb2gray$.

I used imgaussfilt(), which uses $\sigma = 0.5$ and automatically chooses the size of the matrix.

4 (20 points) Optimal Seam

Now that you have your energy images, we must find the optimal seam in it.

Using the technique discussed in class, for each of your images,

- Use its energy image to compute a seam matrix.
- Find the optimal seam in this seam matrix via backtracing.
- Superimpose on your color image the optimal seam in red.

Additional Details:

- We will do vertical seam carving, starting at the top of the image.
- You'll likely have to think about how to handle the edge cases.

5 (20 points) Seam Carving

Finally, let's use seam carving to reduce the aspect ratio. Since we found the optimal *vertical* seam in the previous part, we'll just reduce the width, from it's original width down to one pixel wide.

For each of your images, create a video showing the seam removal process. Each frame of the video should depict the current color image with the current optimal seam superimposed (like in the previous part).

Note:

- You may need to "pad" your frames so that they all have the same size in order to render as a movie.
- To create a movie in Matlab use the *VideoWriter* class. In addition, to keep the movies relatively small in file size, use the *MPEG-4* profile for your VideoWriter object.

Submission

For your submission, upload to Blackboard a single zip file containing:

1. PDF writeup that includes:
   (a) Your answer to the theory question(s).
   (b) Your two original images.
   (c) Eight resized images for Part 2
   (d) Your two energy function images for Part 3.
   (e) Your two optimal seam images for Part 4.
2. A README text file (not Word or PDF) that explains
   - Features of your program
   - Name of your entry-point script
   - Any useful instructions to run your script.
3. Your source files
4. The chosen images that you are processing.
5. The videos generated for Part 5.