

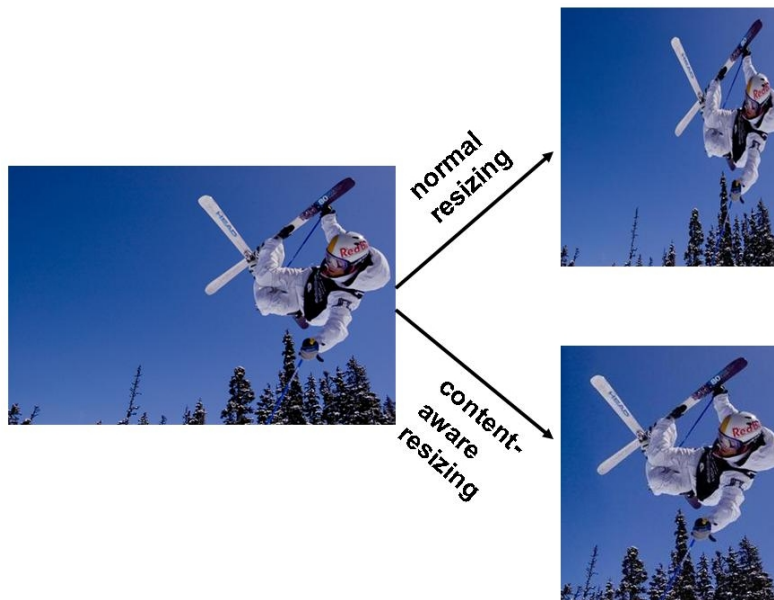
CS 378 Computer Vision
Problem set 1
Out: Thursday Sept 4
Due: Thursday Sept 18, by 11:59 PM

See the end of this document for submission instructions.

I. Short answer problems [5 points each]

1. Consider the filter $f = [1,2,1]$ and the 1D image $I = [0,1,2,3,3,3,1,3,6]$. What is the result of $f * I$? Pad the image with zeros at the boundaries if necessary.
2. Filter f is a 10×10 mean filter, and filter g is a 3×3 mean filter. Image I is a grayscale image. Will the images produced by $f * I$ and $g * I$ look any different? If so, explain how they would differ and why. If not, explain why $f * I$ and $g * I$ would give the same output.
3. Show that scaling the projection matrix does not change the transformation; that is, the non-homogeneous image coordinates are unchanged by a scaled projection matrix.

II. Programming problem: content-aware image resizing [85 points]



For this exercise, you will 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". The paper is available here: <http://www.seamcarving.com/arik/imret.pdf>. The goal is to implement the method, and then examine and explain its performance on different kinds of input images.

First read through the paper, with emphasis on sections 3, 4.1, and 4.3.

Then write Matlab code with functions that can do the following tasks:

- 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)
- Enlarge an image by a specified amount in one dimension (width or height increase)
- Display the selected seam on top of an image
- A function that has the following interface:

```
[output] = resize(im, newWidth)
```

This function takes an input image `im`, and a parameter specifying the new width that the output image should have after resizing. The image `im` will be a $h \times w \times 3$ `uint8` matrix, which is what `imread` returns for a color image. The `newWidth` may be larger or smaller than `im`'s width. Put this function in a file named `resize.m`

Set up scripts so that you can play with the seam removal and specify different combinations of horizontal and vertical removals or enlargements. 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`, `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.

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

1. [10 points] Run your `resize` function on the provided `ocean.jpg` with `newWidth = 540`. Display the output.
2. [10 points] Display (a) the energy function output (total gradient magnitudes $e_1(I)$) for the provided image `ocean.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.
3. [10 points] For the same image `ocean.jpg`, display the original image together with (a) the first selected horizontal seam and (b) the first selected vertical seam. Explain why these are the optimal seams for this image.
4. [10 points] What if we were to modify the energy function to include only gradients in one direction, i.e., only one of the addends in equation 1; how would the output (selected seams) qualitatively change? Choose any image where we might expect a difference, and show an illustrative example using the original and modified energy function, and explain the difference.

5. [10 points] What would be the impact of dilating the energy function output before computing seams? (You can describe the impact without writing code to do it.)
6. [35 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 the three provided images, as well as five images of your own choosing. Include at least two examples of “bad” outcomes. Be creative in the images you choose! Try to predict types of images where you might see something interesting happen. It’s ok to fiddle with the parameters (seam sequence, number of seams, etc) to look for interesting and explainable outcomes.

For each result, include (a) the original input image, (b) your system’s resized image, (c) the result one would get if instead a simple resampling were used (via Matlab’s `imresize`), (d) the input and output image dimensions, (e) the sequence of enlargements and removals that were used, and (f) a qualitative explanation of what we’re seeing in the output.

III. [OPTIONAL] Extra credit [up to 10 points each, max 20 points extra credit]

Below are ways to expand on the system you built above. If you choose to do any of these (or design your own extension) include in your writeup an explanation of the extension as well as images displaying the results and a short explanation of the outcomes. Also include a line or two of instructions telling what needs to be done to execute that part of your code.

1. Allow a user to mark an object to be removed, and then remove seams until all pixels on that object are gone (as suggested in section 4.6 of the paper).
2. Similarly, allow a user to manually specify a region that should be treated as having high energy (see Matlab’s `ginput` function for an easy way to get mouse click input).
3. Experiment with an alternate energy function, instead of the gradient magnitude.
4. To avoid warping regions containing people’s faces, have the system try to detect skin-colored pixels, and let that affect the energy map. Try using the hue (H) channel of HSV color space (see Matlab’s `rgb2hsv` function to map to HSV color space).
5. Optimize for the order of horizontal and vertical seams, as described in Section 4.2 of the Avidan & Shamir paper. Show an example where this compares favorably to hard-coding the sequence.
6. Design your own addition drawing on techniques covered in class. Be sure to explain in the report.

Submission instructions: what to hand in

Create a tar file with

- Your documented Matlab code. Include your name at the top of each file in a comment. Make sure the main program is in a file named `resize.m` as described above.
- A **single pdf** file containing the following
 - Your name at the top
 - Your answers to Part I, numbered.
 - An explanation of your implementation strategy: a short paragraph or two describing in English what you have computed.
 - Your responses and image results for each question in Part II, numbered. Insert image figures in the appropriate places for these questions.
 - (optional): any results and descriptions for extra credit portions.

1. Name the tar file with `<username>_pset1.tar` and submit via the turnin program.

2. Also submit a hardcopy of the pdf file.

Tip: How to save as pdf? If you have a pdf printer installed on your computer, you can convert a document prepared in Word to pdf. The CS machines have `openoffice` installed, which will also allow you to Save as... a pdf file. Or, if you work in Latex, you can use `pdflatex`, or compile to a ps and then convert.