MIT CSAIL

6.869 Advances in Computer Vision

Fall 2017

Problem Set 2

Posted: Thursday, September 14, 2017 **Due:** Thursday 23:59, September 21, 2017

Please submit **two files:** 1) a report named ⟨your_kerberos⟩.pdf, including your answers to all required questions with images and/or plots showing your results, and 2) a file named ⟨your_kerberos⟩.zip, containing relevant source code.

Late Submission Policy: We do not accept late submissions. The submission deadline has a 50-minute soft cut-off; after midnight Thursday, submissions are penalized 2% per minute late.

Problem 1 *Hybrid images*

In this problem you will create hybrid images as described in [3]. You may find the tools [4] useful.

Take two images, A and B, that you'll want to have blend from one to the other. Try to make the objects in the two images occupy more or less the same region. Construct a hybrid image from A (to be seen close-up) and B (to be seen far away) as follows:

$$\text{out} = \texttt{blur}(\text{B}) + (\text{A-}\texttt{blur}(\text{A}))$$

Where `blur` is a function that low-pass filters the image. You should write your own blur function. You can use a Gaussian filter or try other blur filters, such as the box filter. Which one works best? Try different sigmas for the Gaussian. How does the amount of blurring affect your perception of the results?

In your report, please specify the type of kernel you used and its parameters. Also, attach your final result.

Problem 2 *De-hybridizing*

Examine the image `einsteinandwho.jpg` included with the problem set. Using the method of your choice, remove the individual represented in the low spatial frequency range to create two images: one of Einstein, and one of the other person. Please intensity scale the images using the provided code to make them easier to see. Include both images and your code in the report. You may want to try different methods to achieve the best two images. For fun: can you guess who is in the low spatial frequency image?

Problem 3 *Depth-based Defocusing*

In this problem you will write code to simulate the depth of field effect in photography. In short, parts of the scene that are closer to your focal plane are convolved with a smaller blur kernel, and vice versa; more details can be found at [1].

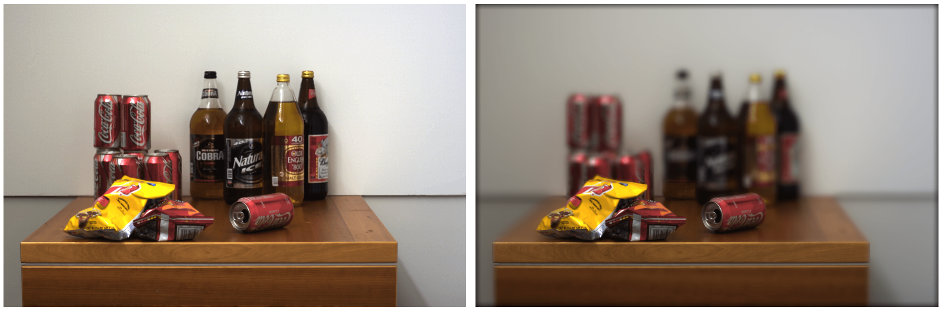

Figure 1: Example of defocusing: the picture on the left is all-in-focus; the one on the right is defocused based on depth.

Particularly, you are given an all-in-focus image $I(x, y)$ and its depth map $D(x, y)$. Depth-of-field blurring would blur the image non-uniformly based on its depth map, i.e:

$$B(x, y) = \sum_{w,v} I(x - w, y - v)K(w, v, D(x, y)),$$

where B is the blurred image and K is the blurring kernel.

We first assume the blur kernel is Gaussian and different depths correspond to different variances σ of the blur kernel. Specifically, we assume:

$$\sigma = \alpha|D(x, y) - \texttt{focal\_plane\_depth}|,$$

where α is a scaling factor. In our base code, we have determined all the coefficients for you, but you need to implement the convolution part.

We then assume our blur kernel is binomial. We assume the level of blurring is given by:

$$round(\alpha|D(x, y) - \texttt{focal\_plane\_depth}| * 10);$$

You need to implement this from ground up. Hint: the level of blurring is discrete, you can iterate through all the levels and fill each pixel with its correct level of blurring.

Please post your defocused result in your report.

Problem 4 *Content-aware Resizing; Grad students only.*

Inspired by a `6.009` lab, in this problem you'll implement a simple variant of the *seam carving* algorithm [2]. The goal is to resize an image of height h and width w to a smaller image of height h and width $w' < w$ such that the actual content of the image is not too distorted. Your code should iteratively remove one column at a time until you reach the target width. Use the combined output of horizontal and vertical Sobel filters to represent the energy of a pixel. At each iteration, remove the column of minimum total energy. Resize the provided image `camber_castle.jpeg` to `300 x 213` and reduce the width of one other image of your choosing by 60%. Include all the images in your report. Note: your resized image will have artifacts. That's okay. A better algorithm would find the minimum energy *path* from top to bottom and remove that path, but you do not need to implement this.

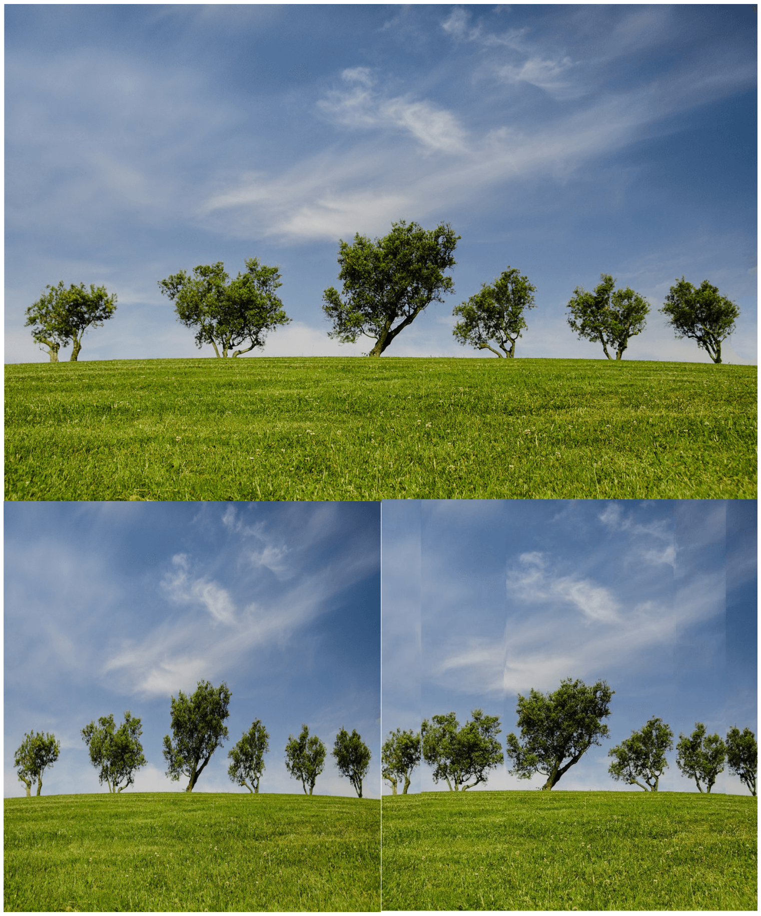

Figure 2: Top: original image. Left: resized image. Right: Content-aware resizing.

References

[1] `https://en.wikipedia.org/wiki/Depth_of_field`.

[2] `https://en.wikipedia.org/wiki/Seam_carving`.

[3] Aude Oliva, Antonio Torralba, and Philippe G Schyns. Hybrid images. *ACM Transactions on Graphics (TOG)*, 2006.

[4] Eero Simoncelli. Matlab pyrtools. `http://www.cns.nyu.edu/~eero/software.php`.