Homework 4: Refocusing

CS 639, Fall 2020

Due on November 19

Total points: 7

Please follow the [homework guidelines](#) in your submission.

runHw4.m will be your main interface for running your code. Parameters for the different programs or unit tests can also be set in that file. Before submission, make sure that you can run all your programs with the command runHw4("all") with no errors.

Challenge 1: RefocusApp   (7 points)

Your task is to develop an application that allows a user to refocus a scene, much similar to the two cases linked here – [Lytro](#), [CAVE](#). We will call it a refocusing app.

In a conventional image, the focus of a scene is fixed at the time of capture. How can a refocusing app refocus a scene? The trick is that, instead of using a single image, a refocusing app uses a focal stack—a sequence of images captured at different focus settings – as an internal representation of a scene. When a user chooses a scene point to be "refocused", the app picks the "best focused" image from the focal stack, and displays the image. How does the app determine the "best focused" image? You have learned the answer in the Depth from Focus/Defocus lecture, and now it is your turn to apply the knowledge to refocus a real world scene. The task is divided into two parts, each corresponding to a program you need to write and submit.

a) **Index Map** (3 points):

Write a program named `generateIndexMap` that generates an index map from a focal stack. An index map is an image with each pixel corresponding to a scene point. The integer intensity of each pixel indicates the index of the best focused layer associated with the corresponding scene point.

Index map generation can be carried out in two steps, and the two steps need to be performed with caution to produce a quality result. The first step is to compute a focus measure for every pixel in every image in a focal stack. We will use the modified Laplacian (described in the lecture) as the focus measure. [*This step is already implemented for you.*]

The second step is to find the layer with the maximal focus measure for each scene point. Here we will not fit a Gaussian to the computed focus measures, as finding the precise depth is not the goal. Instead, we will simply choose the layer with the maximum focus measure as the best focused layer. Be careful, the computed focus measure could be noisy, thus you may need to smooth the data (e.g., with a moving average filter) before selecting the maximal focus measure.

Before calling `generateIndexMap`, write a program named `loadFocalStack` to load a focal stack into memory. Refer to `challenge1a` for the specifications. MATLAB has a collection of file operation functions, which can be useful for this task.

You may use the Image Processing Toolbox functions `fspecial` and `imfilter`.

b) **Refocus** (4 points):

Write a program named `refocusApp` that executes the following tasks in a loop:

i.   display an image in the focal stack

ii. ask a user to choose a scene point (with the MATLAB function `ginput`)

iii. refocus to the image such that the scene point is focused.

The program terminates when the user chooses a point outside of the displayed image frame. Use the index map computed in part (a) to facilitate refocusing. To animate the transition effect, simply call the MATLAB function `imshow` consecutively to display a sequence of images.