

EECS6323 – Assignment 6: Auto focus

Assigned: Feb 15 (Tu)

Due: Mar 1 (Fri) by 11.59 pm

Percentage of total grade: 10%

Objective

1. To perform simple auto focus.
2. To perform manipulation of an image stack.

Provided to you.

1. Two focus stacks in *numpy* format. Use `np.load()` to read in.
2. Video of assignment.
3. Link to zip file (too big for eclass!) [click [here](#)]

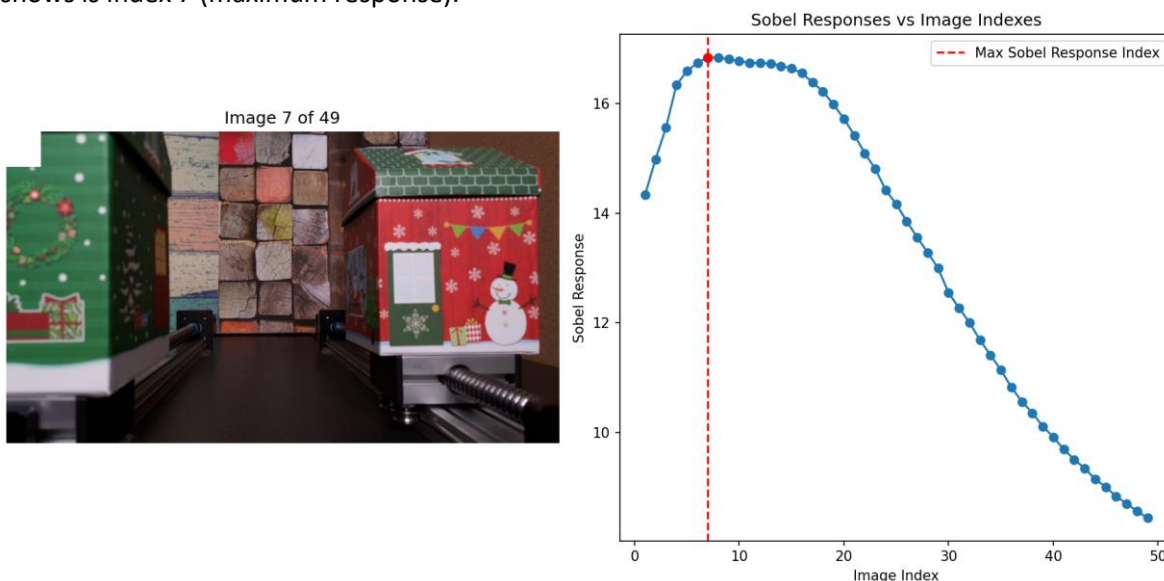
How the program should work.

1. Open a focal stack file (*numpy*).
 - > This will be a 4D array: 49, 2D RGB images [index, row, height, RGB].
2. Compute the sharpness response (Sobel “energy”) over all 49 frames.
 - > Note: you can convert the image to grayscale first.
3. Plot the sharpness response for all frames.
4. Show the image in the stack with the maximum response
 - > Set *current_frame* to this index.
5. If the user presses left/right keys, adjust the *current_frame* ± 1 and refresh the lot.
6. If the user clicks on the image, recompute the focus sharpness response, but only for a 100x100 window around the clicked position (simulating a “local” focus).
 - > Draw a box so we can see the local focal region.
7. Move from current position to the maximum position in the new “local response”.
 - > As you move, introduce a slight delay (condider: `plt.pause()`)
 - > Step 7 is to emulate an “animation” of real AF.
8. If user clicks in the upper left corner (shown in white), re-calculate the “global sharpness” and move to image in the stack with the maximum sharpness response.

Example 1.

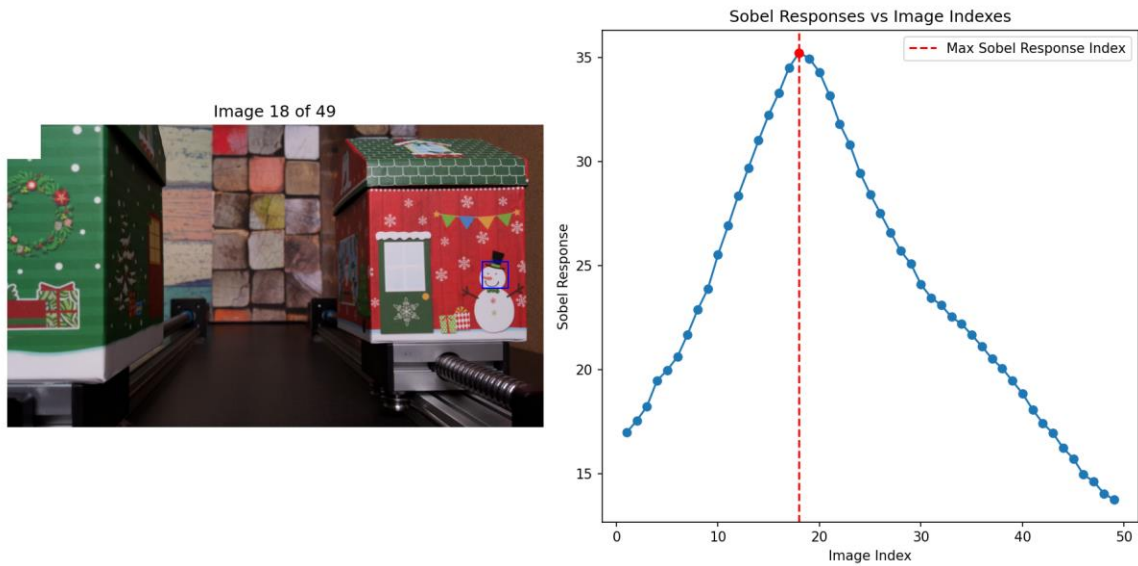
This shows the sharpness responses over the entire image frame.

The image shows is index 7 (maximum response).



See next page . . .

(B) Example of computing a “local” sharpness response. The user has clicked on the snowman’s face (see box).



What to submit via e-class:

Two files

1. Your Python code (zip) [don't include npy files!].
2. Video of your program running w/o audio (mp4).

Comments:

1. The assignment must be done in Python.
2. You are welcome to use ChatGPT.
3. Useful Python packages/functions:

numpy

matplotlib

Matplotlib.patches (Rectangle)

fig.canvas.mpl_connect() <- clicks and keypresses