Rensselaer Polytechnic Institute
Department of Electrical, Computer, and Systems Engineering
ECSE 4540: Introduction to Image Processing, Spring 2018

Homework #4: due Thursday, Feb. 20^{th}, at the beginning of class.
Show all work for full credit!
Submit your work as a single PDF on Gradescope, labeling each page with a problem number.

For this homework, you'll need the image at `http://www.ecse.rpi.edu/~rjradke/4540/tiger.png`.

1. (20 points.) First, write a few functions that will help with the rest of the homework. Turn in your code listings for each function. Each function should only be a couple lines long.

   (a) (5 points.) A function called `fftshow` that displays the log of the magnitude of the 2-D DFT of an image and plots it so that the DC element is at the center of the frequency-domain image, and the result is scaled to use the full grayscale range. Use the functions `fft2` and `fftshift`.

   (b) (5 points.) A function called `boxfilt` that takes as input a grayscale image `im`, an odd box width w and an odd box heighth, and returns an ideal low-pass box frequency-domain filter the same size as `im` that that passes the `w` frequencies around DC along the columns of the image and the `h` frequencies around DC along the rows of the image. The DC element should be at the upper left corner of the output. You can assume the image also has odd dimensions.

   (c) (5 points.) A function called `freqflt` that applies a frequency-domain filter to an image by multiplying the filter with the image's 2D DFT, taking the inverse 2D DFT, and returning the real part of the result as a `uint8`. The function should assume that the frequency-domain filter has the same size as the input image. Note there is a function called `ifftshift` that moves the DC element back to the upper left corner, if you need it.

   (d) (5 points.) A function called `gaussfilt` that takes as input a grayscale image `im` and a 2×2 covariance matrix `C`. The filter should return a low-pass frequency-domain Gaussian filter the same size as `im` centered at DC with the given covariance matrix. This one's a little tricky. Remember the formula for a 2D Gaussian distribution (e.g., equation 2-123), and how you used `meshgrid` to construct a related image in HW 1. We want the maximum element of the filter to be 1, so you only need to worry about the $\exp(\cdots)$ part, not the scaling factor in front. As above, the DC element should be at the upper left corner of the output.

2. (30 points.) Consider the input image `tiger.png`. In each of the following subproblems, provide a result (either in the spatial domain or frequency domain as indicated) and explain why the result looks the way it does. For frequency-domain results, use `fftshift` to make sure DC is in the middle of the image for easier visualization.

   (a) (4 points.) Use `fftshow` to display the 2D DFT of the image.

   (b) (4 points.) Construct a box filter with `[w,h] = [31, 31]`. Look at it in the frequency domain with `fftshift`.

   (c) (4 points.) Apply the above box filter to the input image. What does the result look like in the spatial domain?

   (d) (4 points.) Construct a box filter with `[w,h] = [91, 91]`. Look at it in the frequency domain with `fftshift`, and apply it to the input image. How does the output image differ from part (c)?

   (e) (4 points.) Construct a box filter with `[w,h] = [91, 31]` and apply it to the input image. How does the result differ from parts (c) and (d)?

   (f) (5 points.) Construct a box filter with `[w,h] = [91, 399]` and apply it to the input image. What is the interpretation of this filter? How does the result differ from part (e)?

   (g) (5 points.) Construct a box filter with `[w,h] = [599, 31]` and apply it to the input image. What is the interpretation of this filter? How does the result differ from part (e)?

More fun on the next page ⟶

3. (30 points.) We'll again consider the input image `tiger.png`, but now look at the results of Gaussian filtering instead of box filtering. In each of the following subproblems, provide a result (either in the spatial domain or frequency domain as indicated) and explain why the result looks the way it does.

   (a) (4 points.) Construct a Gaussian filter using `gaussfilt` with $C = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}$ and look at it in the frequency domain with `fftshift`. How does it compare to the box filter in Problem 2b?

   (b) (4 points.) Apply the above Gaussian filter to the input image. What does the result look like in the spatial domain?

   (c) (4 points.) What does the above result look like in the frequency domain?

   (d) (4 points.) Construct a Gaussian filter using `gaussfilt` with $C = \begin{bmatrix} 900 & 0 \\ 0 & 900 \end{bmatrix}$ and apply it to the input image. How does the result differ from part (b)?

   (e) (4 points.) Construct a Gaussian filter using `gaussfilt` with $C = \begin{bmatrix} 900 & 0 \\ 0 & 100 \end{bmatrix}$ and apply it to the input image. Explain the results.

   (f) (5 points.) Now let $R = \begin{bmatrix} \cos 30° & -\sin 30° \\ \sin 30° & \cos 30° \end{bmatrix}$ and $D = \begin{bmatrix} 900 & 0 \\ 0 & 100 \end{bmatrix}$. Construct a Gaussian filter using `gaussfilt` with $C = R^\top DR$. What does this filter look like in the frequency domain?

   (g) (5 points.) What is the result of applying this filter to the input image? Carefully explain the results.

4. (20 points.) Consider the high-boost spatial-domain filter given by $\frac{1}{16}\begin{bmatrix} -1 & -2 & -1 \\ -2 & \underline{28} & -2 \\ -1 & -2 & -1 \end{bmatrix}$.

   (a) (5 points.) In Matlab, display the magnitude of the 2D 256×256 DFT of this filter, centering so the DC term is at the center of the image. Interpret what you see.

   (b) (5 points.) Apply the filter to the image `tiger.png` and verify that this is indeed an edge-enhancing filter.

   (c) (10 points.) With paper and pencil, explicitly compute the frequency response $H(u, v)$ of the high-boost filter. You should get a closed-form answer involving trig functions.