The Cooper Union Department of Electrical Engineering
Prof. Fred L. Fontaine
ECE310 Digital Signal Processing
Problem Set V: Multidimensional Signals & Systems
November 14, 2023

You will be using the following tools in this assignment: *conv2, filter2, freqz2, contour, surf, image, colormap.* You are provided with two gray-scale images, stored in *ImgLily.mat* and *ImgRodan.mat.*
If h is a matrix representing a 2-D FIR filter, then:
`[Hf,fx,fy]=freqz2(h);`
computes its frequency response on a 2-D grid given by fx, fy, where the frequencies are in the range -1 to 1; digital radian frequency would be $pi * fx$ and $pi * fy$, i.e., fx, fy can be viewed as frequency normalized to the Nyquist bandwidth.
When you load the images from the files, they will be 512×512 matrices with *uint8* (unsigned 8 bit integer) data types, representing images on a gray scale from 0=black to 255=white. For such an object x, then $image\,(x)$ displays the object; you should call *colormap(gray)* to get a gray scale for the images. Use this to display the original images and the result of *filter2*, which applies a 2-D FIR filter to the image. Make sure you check the documentations of *filter2* so that you use it properly!
When you look at the frequency response of a filter, use *freqz2* to compute it, but not plot it. Your filters will be zero-phase, so Hf will be real in all cases. Use $surf$ and *contour* to object surface and contour plots of the filters, with *colormap(jet).*
Regarding taking Fourier transforms, the function to use is *fft2*. If you apply *fftshift* to a matrix, it will perform the desired 2-D centering operation, just as in 1-D the operation *fftshift(fft(x))* places the "DC" value (DFT index 0) in the "center" of the vector.

1. In this problem, we develop a model for a signal incident on a sensor array. Assume the sensors are located on a rectangular grid in a plane. For simplicity, we assume unit distances, and hence the sensor coordinates are $\vec{n} = (n_x, n_y)$ with $n_x, n_y \in \mathbb{Z}$. In particular, we are assuming the array is large, so say both indices extending from $-\infty$ to ∞, at least for purposes of modeling the signal. The output of each sensor is measured in continuous time (though, in practice, the outputs would be digitized, though some preliminary analog processing is typically performed as well). The result is a signal of the form:
$$g\left(\vec{n}, t\right)$$
where $\vec{n} \in \mathbb{Z}^2$, $t \in \mathbb{R}$. Denote the spatial radian frequency vector as $\vec{k} = (k_x, k_y)$, and the temporal radian frequency as ω. In what follows, keep $\vec{n}, t, \vec{k}, \omega$ separate (i.e., do not create 3-D vectors). The individual coordinates $n_x, n_y.k_x.k_y$ should not appear in your formulas.

   (a) Specify the domains for $\vec{k}$ and ω.

   (b) Write formulas for the Fourier transform $g\left(\vec{r}, t\right) \rightarrow G\left(\vec{k}, \omega\right)$ and the inverse Fourier transform $G\left(\vec{k}, \omega\right) \rightarrow g\left(\vec{r}, t\right)$.

(c) Write the formulas for inner product in the $(\vec{r}, t)$ and $\left(\vec{k}, \omega\right)$ domains.

(d) Write the formula for convolution in the $(\vec{r}, t)$ domain. Identify whether convolution is linear or circular with respect to the $\vec{r}$ variable, and the ω variable.

(e) Write the formula for convolution in the $\left(\vec{k}, \omega\right)$ domain. Identify whether convolution is linear or circular with respect to the $\vec{k}$ variable, and the ω variable.

2. Here you will be looking at two discrete approximations to the 2-D Laplacian operator ∇^2. A simple discrete approximation to the two-dimensional Laplacian ∇^2 is:

$$h_{\text{Lap}} = \frac{1}{6}\begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix}$$

where the center point in the matrix corresponds to $h_{\text{Lap}}[0,0]$. Specifically, since ∇^2 is LTI, it is reasonable to approximate it with a discrete 2-D filter, and the prescribed h is the proposed impulse response of an FIR filter. In this problem, use *freqz2* to compute frequency response, as described above. The symmetry in h corresponds to zero-phase $H(\omega_1, \omega_2)$. There is an alternative way to arrive at an approximation to the Laplacian. Let h_x, h_y represent the Sobel filters, i.e., for a 2-D image g:

$$\frac{\partial g}{\partial x} \approx h_x * g \qquad \frac{\partial g}{\partial y} \approx h_y * g$$

Then from:

$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

this suggests a discrete approximation, say h_{LapSob}, to ∇^2, that can be expressed in terms of h_x, h_y.

(a) Obtain an expression for h_{LapSob} in terms of h_x, h_y. Then evaluate this, choosing from among the MATLAB tools described above. Confirm that the result is a 5×5 zero-phase FIR filter.

(b) Compute the frequency responses of both h_{Lap} and h_{LapSob}. Confirm that both are real, and specifically $H_{\text{Lap}} \leq 0$ and $H_{\text{LapSob}} \leq 0$ at all frequencies. **Remark:** The frequency response for ∇^2 is $-\left\|\vec{k}\right\|^2$, so this is consistent! **Note:** Because of numerical error, there may be a small imaginary part in $H_{\text{Lap}}, H_{\text{LapSob}}$. Check if this happens and, if so, discard it!

(c) Obtain surface plots, and contour plots (as described above) for H_{Lap} and H_{LapSob}, not their magnitude (i.e., the values are negative real). Would you call H_{Lap} highpass, bandpass, lowpass or bandstop? Would you call H_{LapSob} highpass, bandpass, lowpass or bandstop? Also, is it reasonable to say they are approximately isotropic?

(d) Load the Rodan and Lily images. Create gray-scale images for each, as well as for the results of applying both h_{Lap} and h_{LapSob} to each. This shows one typical use of the Laplacian.

3. Upsampling in $\mathbb{Z}^D$ is defined as follows. Let M be a $D \times D$ invertible integer matrix, and let $M\mathbb{Z}^D$ denote all integer vectors of the form:

$$M\mathbb{Z}^D = \left\{\vec{n} : \vec{n} = M\vec{m} \text{ for some } \vec{m} \in \mathbb{Z}^D\right\}$$

For example, with:

$$M = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

we have $M\mathbb{Z}^2 = \{(\text{even,even})\}$. Now define $v = (\uparrow M)\, u$ as:

$$\begin{aligned} v\left[M\vec{n}\right] &= u\left[\vec{n}\right] \\ v\left[M\vec{n} + \vec{m}\right] &= 0 \text{ for all } \vec{m} \notin M\mathbb{Z}^D \end{aligned}$$

For the same M as above, this means:

$$v\left[2n_x, 2n_y\right] = u\left[n_x, n_y\right]$$

but $v\left[\text{even,odd}\right] = v\left[\text{odd,even}\right] = v\left[\text{odd,odd}\right] = 0$. In the 2-D case, we can perform this operation on a matrix u by inserting alternating rows and columns of zeros between the original data.

(a) Write a function in MATLAB that, given u an $N_1 \times N_2$ matrix, will generate:

$$\left(\uparrow \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}\right) u$$

(b) Take both the Lily and Rodan images, and perform the upsampling operation on each. Plot each of the resulting images as grayscale. Vary the size of the image displayed on your screen. You will observe certain artifacts, especially at certain displayed image sizes. As a hint to what is happening, on your physical display, each "pixel" of the image (point in the image matrix) represents a block of points in the video memory; if this is not a fractional amount, say it is 2.5, then you may get patterns like 2-3-2-3-2$\cdots$ across different rows or columns; i.e., each point in your matrix is represented by an $K_1 \times K_2$ block of display pixels. One way to check you are getting the right answer is look at say the entries of the upper left 10×10 points in your image before and after upsampling.

(c) Compute the magnitude 2-D DFT of the images before and after upsampling, and display them as gray scale images (using *fftshift* to place DC at the center). What you are looking at is essentially the $\{-\pi \le \omega_x \le \pi, -\pi \le \omega_y \le \pi\}$ frequency square. Comment- what you are observing in the upsampled spectrum would be called _______ distortion.