Lecture Review and Quizzes (Due: Wednesday, February 5, 1:30pm)
Please review what you have learned in class and then complete the online quiz questions for the following section:

- Linear Image Processing and Filtering
- Template Matching

Homework #4
Released: Monday, January 27
Due: Wednesday, February 5, 1:30pm

1. Moire Pattern Suppression in Radiographs (Total of 9 points)

Radiographs of tissue more than 10 cm thick are typically acquired through a Bucky grid, a fine pattern of alternating lead and plastic strips that suppresses scattered radiation and thus improves the contrast of the image. Unfortunately, when the radiograph image is sampled, Moire patterns can result. This problem studies how to reduce Moire patterns while properly preserving the salient features for diagnosis. Please download the images hw4_radiograph_1.jpg and hw4_radiograph_2.jpg from the handouts webpage. Each of these images is corrupted by a clearly visible Moire pattern.

(a) For each image, apply an N×N median filter (function: medfilt2). Adjust the window size N so that the Moire pattern is removed as much as possible while salient features are properly preserved. Report your choice of N, submit the filtered image, and comment on the quality of the filtered image.

(2 points)
(b) For each image, compute its Discrete Fourier Transform (DFT) (functions: \texttt{fft2} and \texttt{fftshift}) and submit an image showing the DFT magnitude (function: \texttt{abs}). A log display may be most appropriate. Clearly identify and label the frequency components that correspond to the Moire pattern. \hfill (3 points)

(c) For each image, design a notch filter so that the frequency components for the Moire pattern are suppressed as much as possible while other frequency components are preserved. Apply your notch filter to the image’s DFT and submit an image showing the filtered DFT magnitude. Display and submit the filtered image in the spatial domain (functions: \texttt{ifftshift}, \texttt{ifft2}, and \texttt{real}). Compare the quality of the result to that of the filtered image from part (a). The function \texttt{meshgrid} is useful for creating an \((\omega_x, \omega_y)\) array. \hfill (4 points)

Note: Please include relevant MATLAB code.
2. Difference-of-Boxes Filtering (Total of 9 points)

The difference-of-boxes (DoB) filter can be used for fast keypoint detection in images. For a particular scale $S$, the DoB filter has the following impulse response $h_S[x, y]$: \[ h_S[x, y] = \frac{1}{(2S+1)^2} g_S[x, y] - \frac{1}{(4S+1)^2} g_{2S}[x, y] \] \[ g_w[x, y] = \begin{cases} 1 & -W \leq x \leq W, \ -W \leq y \leq W \\ 0 & \text{otherwise} \end{cases} \]

(a) Write an expression for the Fourier Transform of the DoB filter. Submit a plot of the frequency response for $S = 2$ (functions: meshgrid, mesh, surf). What type of frequency response (low-pass, band-pass, high-pass) does the filter have? You can use the following Fourier Transform pair, which is valid for any positive integer $M$. \[ f_M[x] = \begin{cases} 1 & 0 \leq x \leq M \\ 0 & \text{otherwise} \end{cases} \leftrightarrow F_M(e^{j\omega}) = \frac{\sin(\omega (M+1)/2)}{\sin(\omega / 2)} e^{-j\omega M/2} \quad (3 \text{ points}) \]

(b) Show that DoB filtering can be implemented by a set of 1-d filtering operations and some additions/subtractions/multiplications to combine intermediate results. Count and report the total minimum number of additions, subtractions, and multiplications required per pixel, neglecting boundary effects. \quad (2 \text{ points})

(c) Show that DoB filtering can be implemented with an integral image. Count and report the total minimum number of additions, subtractions, and multiplications required per pixel, including the cost of constructing the integral image but neglecting boundary effects. For a single scale $S$, is this implementation more efficient than the implementation in part (b)? What about for multiple scales $S = 1, 2, 3, \ldots$? \quad (2 \text{ points})

(d) Please download the images hw4_cd_cover.jpg and hw4_ornament.jpg from the handouts webpage. Convolve each image with a DoB filter of scale $S = 2$ (function: imfilter with options ‘replicate’ and ‘conv’). Display and submit the filtered images (show 0 as mid-gray in each image, positive values as bright, and negative values as dark). Comment on which features in the images are relatively accentuated by the DoB filter. \quad (2 \text{ points})

Note: Please include relevant MATLAB code.
3. Analysis of Sampling Effects in a Zoneplate (Total of 9 points)

This problem focuses on analyzing some of the effects of sampling a zoneplate image that we observed in the lecture covering “Zoneplate for Analysis of Filtering and Sampling”. In particular, we would like to investigate the effect of the sampling phase on the phase of the resulting spectral replica.

(a) The formula to generate the zoneplate is: \( f[x,y] = \hat{f} \cdot \cos(a_xx^2 + a_yy^2) + f_0 \). Using the parameters \( \hat{f} = -112 \), \( f_0 = 128 \), \( a_x = \pi / 512 \), \( a_y = \pi / 512 \), display and submit a zoneplate image \( f[x,y] \) where \(-256 \leq x \leq 256\), \(-256 \leq y \leq 256\).

(b) Display and submit the sampled images \( f_T[x,y] \) for \( T = 2,4 \) where

\[
    f_T[x,y] = \begin{cases} 
        f[x,y] & \text{if } y \text{ is a multiple of } T \\
        0 & \text{otherwise}
    \end{cases}
\]

(1 point)

(c) Display and submit the sampled images \( g_T[x,y] \) for \( T = 2,4 \) where

\[
    g_T[x,y] = \begin{cases} 
        f[x,y] & \text{if } (y-T/2) \text{ is a multiple of } T \\
        0 & \text{otherwise}
    \end{cases}
\]

What is the most noticeable difference between images \( g_T[x,y] \) and \( f_T[x,y] \) of part (b)?

(1 point)

(d) Since the sampling in parts (b) and (c) are in one dimension, we model the sampling process using 1-d space continuous signals. Let \( f_T(y) = f(y) \cdot \text{III}_T(y) \), where \( f_T(y) \) is the sampled signal, \( f(y) \) is the original signal, and \( \text{III}_T(y) \) is the Shah function with sampling period \( T \):

\[
    \text{III}_T(y) = \sum_{n=-\infty}^{\infty} \delta(y-nT) .
\]

Compute the continuous-time Fourier Transform (CTFT) \( F_T(\omega_y) = \text{CTFT}\{f_T(y)\} \) in terms of \( F(\omega_y) = \text{CTFT}\{f(y)\} \). You can use the following fact:

\[
    \text{CTFT}\{\text{III}_T(y)\} = \frac{1}{T} \sum_{n=-\infty}^{\infty} \delta(\omega_y - n \frac{2\pi}{T}) .
\]

Use the derived relationship between \( F_T(\omega_y) \) and \( F(\omega_y) \) to explain the results observed in part (b).

(3 points)

(e) Let \( g_T(y) = f(y) \cdot \text{III}_T(y-T/2) \), where the Shah function has been shifted by half of the sampling period \( T \). Compute the CTFT \( G_T(\omega_y) = \text{CTFT}\{g_T(y)\} \) in terms of \( F(\omega_y) = \text{CTFT}\{f(y)\} \). Use the derived relationship between \( G_T(\omega_y) \) and \( F(\omega_y) \) to explain the results observed in part (c).

(3 points)

Note: Please include relevant MATLAB code.
4. Wiener Filtering of Satellite Images (Total of 10 points)

Satellite images can be degraded by atmospheric turbulence, which we try to model as linear, shift-invariant blurring with an isotropic Gaussian impulse response (with unknown standard deviation) plus additive white noise (with unknown variance). Please download hw4_satellite_1_degraded.tif and hw4_satellite_2_degraded.tif from the handouts webpage.

(a) Design and implement a method to estimate the standard deviation of the additive white noise directly from the degraded image. For each image, report the estimated noise standard deviation, assuming image intensity values in the range [0,1].

(b) Design and implement a method to estimate the standard deviation of the Gaussian impulse response directly from the degraded image. For each image, report the estimated Gaussian standard deviation.

(c) Perform inverse filtering on each image, using the estimated Gaussian impulse response from Part B (function: deconvwnr with a noise estimate of 0). Submit the inverse filtered image. Comment on the quality of the restored image.

(d) Perform Wiener filtering on each image, using the estimated noise standard deviation from Part A and the estimated Gaussian impulse response from Part B (function: deconvwnr). Submit the Wiener filtered image. Comment on the quality of the restored image and compare to the result from Part C.

Note: Please include relevant MATLAB code.