Take-Home Exam #2

- You may have 48 hours to work on this exam. You must enter your name and sign-out and sign-in times on the log sheet in Gerstacker 1107.
- You may use your books, your own notes, web resources, computers and calculators.
- You may not seek assistance, share or borrow notes from any current and former students or any other individual (instructors' notes available on the web are OK).
- This exam is governed by the Engineering Honor Code, which requires that you do not seek assistance on this exam and that you report any violations of the Honor Code. (For more info see http://www.engin.umich.edu/org/ehc/index.html).
- On the first page of your solutions, write "I have neither given nor received aid on this exam," and sign your name below it.
- Show your work and hand in all Matlab code.
- If parts of the exam are not clear and you cannot find the instructor to ask for clarification, please write your assumptions and proceed with the question. If you cannot solve a part that is needed later in the solution, define a parameter to represent the answer of that part and continue.
- Write legibly (yes I know I don't, but…)
- Please remember to put your name on your exam solutions.
- Good luck!

1. (50 pts) In this problem you will investigate a practical approach to determine space-invariant, compact, image-domain representations of the Wiener filter. We will use the picture of the house again (hw7image.mat). Download `e2p1_template.m`. This template creates a noisy and blurred versions of the image: $y(n,m) = s(n,m) + w(n,m)$ and $y2(n,m) = s(n,m) ** b(n,m) + w(n,m)$.
   a. Working first with *y*, choose a background (e.g. sky) area and estimate the noise variance $\hat{s}_w^2$. Is this close to the theoretical variance derived from the code?
   b. We hope to derive a 7x7 Wiener filter, so we will start by calculating a low-resolution (7x7) estimate of the power spectrum of the noisy image, $\hat{P}_y(w_x, w_y)$. Do this by averaging the periodograms of all non-overlapping 7x7 blocks of the image.
   c. Calculate the Fourier domain Wiener filter, $H(w_x, w_y)$, from the above estimates. Rather than using the affine form of the Wiener filter, make an adjustment to this filter so that it will not change the image mean. Display *H* using `imagesc` and include a colorbar.
   d. Derive the 7x7 image space Wiener filter, $h(n,m)$. Display *h* using `imagesc` and include a colorbar. What symmetries does this filter have? Is it real-valued?
   e. Apply filter to *y* and display. Calculate the MSE relative to *s* before and after filtering.
   f. Examine the blur function leading to *y2* (see template). Calculate the low-resolution (7x7) representation of $B(w_x, w_y)$.
   g. Using the same estimates from parts a. and b., repeat parts c., d., and e. (but apply to *y2*).

2. (40 pts) Here, we will examine a special case of image spatial transformations: rotations. Download `e2p2_template.m`. This template creates two different kinds of squares, each in an x-y orientations and a 30 degree rotated orientation. We will examine several different interpolation methods for bringing the rotated squares (`im1r, im2r`) into alignment with the x-y squares (`im1, im2`).
   a. Use nearest neighbor, bilinear and bicubic interpolation to bring rotated squares into alignment with the x-y squares. Calculate the MSE between original images (`im1, im2`) and realigned images. For this part you may either create your own code or use Matlab's `imrotate` or `interp2` functions (note, this cannot be done with `conv2`) – if you choose to use Matlab's `imrotate` function, please be cautious regarding nuances such as center of rotation and that the angles are specified in degrees.
   b. We have at times discussed that sinc interpolation is optimal, but can be computationally intensive. An alterative has been suggested that uses FFT's. This takes advantage of the property that the rotation matrix $\begin{bmatrix} \cos q & -\sin q \\ \sin q & \cos q \end{bmatrix} = \begin{bmatrix} 1 & -\tan\frac{q}{2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \sin q & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tan\frac{q}{2} \\ 0 & 1 \end{bmatrix}$, where $\begin{bmatrix} 1 & c \\ 0 & 1 \end{bmatrix}$ is a "x-shear" matrix – that is the x coordinate is shifted a different amount for every y location (the transpose is a y-shear). Show that this shear coordinate transformation leads to each row being shifted by an amount cy. Describe how you might use 1D FFT's to shift individual rows by cy (a circular shift is ok). Argue similarly for how to use 1D FFT's to implement the $\begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix}$ coordinate transformation.
   c. Implement part b. using Matlab's 1D FFT functions. Be careful with `fftshift`. Use this method to bring the rotated squares into alignment with the x-y squares. Calculate the MSE between original images (`im1, im2`) and realigned images.
   d. Which methods are best for `im1`? Which for `im2`? Is this what you expected? If the best methods are different for `im1` and `im2` explain why?

3. (10 pts) Let $\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ be jointly distributed according to $Normal\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}\right)$ and w_1 and w_2 are both white noise processes. Let the desired image be $f(n,m) = w_1(n,m) ** h_1(n,m)$, the noise $v(n,m) = w_2(n,m) ** h_2(n,m)$, and the measurement $g(n,m) = f(n,m) + v(n,m)$.

   a. Derive the autocorrelation functions, power spectra, cross correlation functions, and cross power spectral functions: R_f, R_v, R_g, R_{fv}, R_{fg}, P_f, P_v, P_g, P_{fv}, P_{fg}. Derive, also, expressions for the variances s_f^2, s_v^2 and s_g^2.
   b. Derive the L.S.I. filter, $H(w_x, w_y)$, that you would apply to $g(n,m)$ to produce an estimate of f, $\hat{f}(n,m)$ that minimizes the MSE.