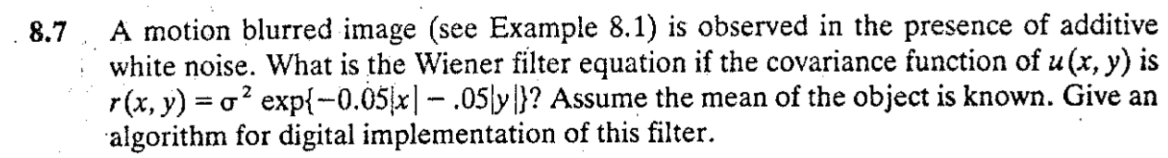
ECE 6364 Spring 2016 HW 10 Due 4/14

Problem 1.

Let $\vec{\varepsilon} = \underline{\mathbf{Q}}\,\vec{\mathbf{u}}$ where $\vec{\mathbf{u}}$ is a row-ordered unknown image, $\underline{\mathbf{Q}}$ is a known non-singular matrix with inverse $\underline{\mathbf{Q}}^{-1}$, and $\vec{\varepsilon}$ is a zero-mean vector of *i.i.d.* white noise with variance σ^2 so that covariance($\vec{\varepsilon}$) = $\sigma^2\,\underline{\mathbf{I}}$. You observe a row-ordered degraded image $\vec{v} = \underline{\mathbf{H}}\,\vec{\mathbf{u}} + \vec{\mathbf{n}}$ where $\vec{\mathbf{n}}$ is a zero-mean noise vector with covariance matrix $\gamma\,\underline{\mathbf{I}}$ and $\vec{\mathbf{u}}$ and $\vec{\mathbf{n}}$ are uncorrelated. Find the Discrete Wiener filter for restoring $\vec{\mathbf{u}}$ from $\vec{v}$.

Problem 2.

Problem 8.7 -Jain



8.7 A motion blurred image (see Example 8.1) is observed in the presence of additive white noise. What is the Wiener filter equation if the covariance function of $u(x, y)$ is $r(x, y) = \sigma^2 \exp\{-0.05|x| - .05|y|\}$? Assume the mean of the object is known. Give an algorithm for digital implementation of this filter.

Problem 3.

Find the periodic convolution of **X** and **Y**. 2-D periodic convolution is the 2-D extension of 1-D periodic convolution that you did in digital signal processing class. Note that, similar to the 1-D case, the 2-D periodic convolution result is the same as would be obtained by computing
IDFT{ DFT{ **X** } x DFT{ **Y** } } where x denotes element-by-element multiplication.

$$\underline{\mathbf{X}} = \begin{bmatrix} [2] & 1 & 0 \\ 3 & 1 & 1 \end{bmatrix} \qquad \underline{\mathbf{Y}} = \begin{bmatrix} [0] & 1 & 2 \\ 1 & 3 & 4 \end{bmatrix}$$

Problem 4.

You want to compute the linear convolution of two 1-D images $a\,[b]\,c \;\; ** \;\; d\,e\,[f]$ using DFTs. You load the two 1-D images into zero-padded vectors as $[0 \quad 0 \quad a \quad [b] \quad c]$ and $[0 \quad d \quad e \quad [f] \quad 0]$, compute the DFTs of the two vectors, multiply the two DFTs together term-by-term, and then take the inverse DFT. Your result is the vector $[1.1 \quad 2.2 \quad 3.0 \quad 4.2 \quad 5.9]$. Determine the numerical result of the **linear** convolution $a\,[b]\,c \;\; ** \;\; d\,e\,[f]$, marking the 0th pixel with a box.

Problem 5.

Using 2-D DFTs to compute linear 2-D convolutions is a straightforward extension of the use of 1-D DFTs to do 1-D linear convolution. Zero-padding in 2-D is the extension of zero-padding in 1-D. Again, the question is "where is your (0,0)th pixel in the output" and "how is the linear convolution result arranged in the output matrix obtained by the IDFT", the same as the question for the 1-D DFT case.

You want to do linear convolution of a 256x256 image f(i,j) with a 9x9 mask m(i,j) to form a 264x264 linear convolution c(i,j) image. Using image-indexing, you number the (0,0)th pixel of the image f(,) as the upper-left-hand-corner of f(,), so that the pixels of the image are numbered f(0,0) to f(255,255). Using image-indexing, you number the (0,0)th pixel of the mask m(,) as the pixel directly in the middle of the 9x9 pixel mask, so that the pixels of the mask are numbered m(-4,-4) to m(4,4).
The 264x264 linear convolution image is therefore correctly numbered c(-4,-4) to c(259,259).

To implement linear convolution you are to use DFTs with zero-padding. You load the image f(,) and mask m(,) into the upper-left-hand corners of separate 264x264 pixel matrices below, and zero-padding the right-hand columns and lower-rows of the matrix as necessary.. You take the DFT of both matrices, multiply the elements of the two matrices together term-by-term to form a 264x264 product matrix, and finally take the inverse DFT of the product matrix to obtain a 264x264 matrix **Q** with pixels numbered **Q**(0,0) to **Q**(263,263).

Which elements of this 264x264 matrix **Q** hold the central portion of the linear convolution; i.e. c(0,0) to c(255,255)?

$$\underset{264\text{x}264}{\begin{bmatrix} f_{0,0} & f_{1,0} & \bullet & f_{255,0} & 0 & \bullet & 0 \\ f_{0,1} & \bullet & & \bullet & \bullet & & \\ \bullet & & \bullet & \bullet & \bullet & & \bullet \\ f_{0,255} & \bullet & \bullet & f_{255,255} & 0 & & 0 \\ 0 & \bullet & \bullet & 0 & 0 & & 0 \\ \bullet & & & & & \bullet & \bullet \\ 0 & & \bullet & 0 & 0 & \bullet & 0 \end{bmatrix}} \quad \underset{264\text{x}264}{\begin{bmatrix} m_{-4,-4} & \bullet & m_{0,-4} & \bullet & m_{4,-4} & 0 & \bullet & 0 \\ \bullet & \bullet & \bullet & \bullet & \bullet & & & \\ m_{-4,0} & \bullet & m_{0,0} & \bullet & m_{4,0} & & & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & & & \\ m_{4,-4} & \bullet & m_{0,4} & \bullet & m_{4,4} & 0 & \bullet & 0 \\ 0 & & & & 0 & 0 & & \bullet \\ \bullet & & & & \bullet & & \bullet & \bullet \\ 0 & & \bullet & & 0 & \bullet & \bullet & 0 \end{bmatrix}}$$

Problem 6.
A portion of the video data from an endoscopic procedure was emailed to you.
Restore the video by removing blur and noise.
Your blur model might be spatially variant as described in class, or you might use a spatially invariant model.
You might use a method to estimate the blur based upon small features within the images, you might simply assume a Gaussian blur with a specific standard deviation. However you approach the project, right up your approach and results in a report and submit it by the last class meeting.