Lab 10: Fourier Shift Theorem: Image Registration and Change Detection

due TBD

Goals:

1. To learn about the Fourier transform the Fourier shift theorem.
2. To write a program to apply an arbitrary, uniform pixel shift to an image.
3. To write a program to detect an arbitrary, uniform pixel shift in a pair of images.
4. To design an algorithm to detect non-uniform pixel shifts in a pair of images.

Introduction:

Assume two functions defined in the domain x are related in the following fashion:

$$h(x) = f(x - a) \qquad (1),$$

where a is a constant "shift" between the two functions. The Fourier Shift Theorem states that the following holds in the frequency domain:

$$H(\omega) = F(\omega)e^{-i(\omega a)} \qquad (2),$$

such that,

$$\frac{F(\omega)H^*(\omega)}{\left|F(\omega)H^*(\omega)\right|} = e^{i\omega a} \qquad (3),$$

where $F(\omega)$ and $H(\omega)$ are the Fourier transforms of $f(x)$ and $h(x)$, respectively, and ω is the frequency corresponding to the spatial dimension x. The * denotes the complex conjugate.

Equation (3) defines the "phase component of the cross-correlation power spectrum" and yields a phase term that is independent of the spatial frequency ω and only dependent on the shift a. The exponential term on right-hand side of Equation (2) can be transformed back into the spatial domain using the inverse Fourier transform to yield an impulse function with a maximum value at the shift, a.

The above holds for 1-D, continuous functions. For 2-D, discrete, bounded functions such as images, the mathematics are similar (see readings), with the 2-D

discrete IFFT of Equation (2) yielding a 2-D impulse function with a maximum value at spatial coordinates corresponding to the amount of shift (in the row and column directions) between the two images.

Note that the two functions/images in question so far are assumed to be shifted by a uniform and constant amount. In other words, every pixel in image $f(x,y)$ would be shifted by the exact same amount (and in the same direction) with respect to its counterpart in image $h(x,y)$. It is important to realize that, for geodetic purposes, this is rarely the case. More often, the amount of shift between corresponding pixels in an image pair will vary as a function of position, with some pixels, in fact, having experienced no shift between image *f* and *h*. In these cases, the mathematics remain the same, but some scheme for subsetting the images and systematically processing the images in a piecewise fashion is necessary to map out spatial variations in the amount of shift (i.e. to derive a displacement field such as those illustrated in the readings).

Instructions:

The assignment is to write two MATLAB codes:

A. The first should take an input image and apply a (arbitrary) constant and uniform amount of shift in the row and/or column directions. You do not need to use the Fourier Shift Theorem for this part, but you can.

B. The second should take a pair of images that are identical to one another except for some amount of (unknown) constant and uniform shift in the row and/or column directions, and compute the shift using the Fourier Shift Theorem.
Use an image of your own choosing as input for part A. Use the input and output image pair for part A as the input for part B.

C. You are also to design an algorithm. No working MATLAB code is required for this – you are to write "psuedocode" outlining the problem and your proposed solution to it. The problem your algorithm is to address is the case of two satellite images taken before and after an earthquake. The earthquake produced non-uniform surface change, specifically horizontal displacement of the Earth's surface (and, therefore, horizontal displacement between equivalent pixels in the before and after images). In particular, those pixels closest to the fault have a

maximum amount of shift and those further away having minimal to no shift. How can you adapt the mathematics of Equations (1) and (2) and your code for question B to map out a non-uniform/spatially-variable displacement field? The readings should help you here…

What To Turn In and How:

For parts A and B, you need to turn in your two MATLAB codes, your input image for part A, and your input images for part B. Also indicate to me (for instance in the comments to your code for part A) what amount of shift you used in part A (which is also the amount of shift between the input images for part B that your code is supposed to figure out).

For part C, you need to turn in your pseudocode. Since this will not necessarily be executable MATLAB code, it is especially important that you make sure that I can understand and follow your algorithm outline. Keep in mind that sketches can usually help illustrate how the code would work.