Goals

The purpose of this assignment is to experiment with the use of the Fourier transform for image registration and filtering and to build a small system that automatically builds a mosaic from a small set of overlapping input images.

1 Phase Correlation

Implement a Fourier based routine that computes the phase correlation between two images of the same size. This routine should return the result of the phase correlation in the spatial domain. I.e. for images $f$ and $g$, the phase correlation is

$$p(x, y) = \mathcal{F}^{-1}\left[ \frac{F^*(u, v)G(u, v)}{|F^*(u, v)G(u, v)|} \right]$$

In practice, you will need to extend your phase correlation to remove high-frequency components and noise. Thus, you will need a routine that computes

$$p(x, y) = \mathcal{F}^{-1}\left[ H(u, v)\frac{F^*(u, v)G(u, v)}{|F^*(u, v)G(u, v)|} \right],$$

where $H$ is a low-pass transform. Make a separate routine that builds $H$.

2 Peak Finding

You will need to find the peak in the phase correlation. For starters, you can simply look for the pixel location with maximal value. You will need to make some decisions. For instance, if the maximum value is below some threshold, you would say that two images do not overlap. For more advanced analysis, you would look for connected components in the correlation image above some threshold and do some averaging of magnitude and position within those components in order find the best “peak” to associate with the offset of the two image. Think about what would work best and come up with your own design and test it and describe it in the report.

3 Mosaic Building

You should build a routine that reads in a set of images (specified in a file) and builds and saves a mosaic. You can, perhaps, assume that the first image defines the target. You will, as in your last project, need to make a canvas and resample all images onto the canvas. Don’t worry about blending, etc., for now. Just build the mosaic using whatever scheme you want between the overlapping images and evaluate the effectiveness. To compute the phase correlation between two images, they need to be the same size. Thus, you will need to preprocess images by appending rows/columns so that they all have the size of the max row and column among all the images. I suggest padding with the average image value, or, possibly, by repeating the edge pixels or putting noise around the border. To make things simple you should pad on the right edge and bottom, keeping the origin of the image in place.

Because phase correlation in the image domain allows for wrap around, there is an ambiguity in the meaning of a phase correlation peak. There are two strategies to solve this. One strategy is to pad the image...
to be more than twice as big as the inputs in every direction. If you get peaks that are bigger than half the padded image, then you know that these result from the wrap around of the image in that direction.

Another approach is to pad as needed for the FFT and then resolve the four cases of ambiguity after you find the peek. To do this you would set up four separate loops to compute the correlation on the overlapping patches in the spatial domain, as shown in Figure 3.

4 Experiments

You should experiment with the set of six images attached (cellmosaicImages1 and cellmosaicImages2). You should also build a toy example to debug with. To do this, you can take a simple image and manually (using an image editing tool) break it into a couple of overlapping pieces. Experiment with different thresholds and different low-pass filters or parameters.

5 Code

You should not implement an FFT from scratch. In Matlab, you can use the routine \texttt{fft()}, which computes the 1D FFT of a signal, or \texttt{fft2()} which computes the FT from a 2D array. In Vispack, you can use the code attached (fourier), and have a quick look at the readme file.

6 Instructions, Requirements, and Restrictions

1. Write your project code in a single directory, called \texttt{project4}. 
2. For Matlab, each individual function (including functions you define) should be a “.m” file, and your functions should call one another as necessary. Likewise, in C++ clearly separate these different functions and document how you have organized the code.

3. You should use examples of images in your report.