Signal Filtering Methods for InSAR - Instruction to the codes.

Here I made two codes for processing profiles (1D) and 2D images. Both codes are almost the same except dealing with different dimensions.

filt_COR.m

I already cut several profiles of InSAR and DEM (Digital Elevation Data) beforehand. I used *GMT* (General Mapping Tool, an open source software) to cut the profiles (my GMT code, profile.sh, is also attached in the archive, but the original InSAR files are not included).

In this matlab file it firstly loads DEM and InSAR profiles. Not every InSAR pixel has a corresponding value, because the coherence of every pixel between two SAR images is not always high enough to return a reasonable value. Hence, some of the pixels have values like "NaN". I firstly remove these NaNs by using *isnan* in Matlab. Besides, due to the different resolution of InSAR and DEM, I have to resample both into the same data points by using interp1.

Then I apply *fft* as the Fast Fourier Transform for both signals from spatial domain into frequency domain. The complex cross-correlation and the coherence are computed from line 43 to 48 in the codes. The decorrelation is computed by subtracting coherence from 1. Once this is done, the codes would multiply the decorrelation and the signal, and then do inverse Fast Fourier Transform back to the spatial domain.

The rest of the codes are some comparison of the signals (InSAR & DEM) before and after this filtering method. The first figure shows both InSAR & DEM before/after the filtering. The second figure shows the different before and after the filtering. The third figure shows the distribution of every pixel in InSAR and DEM before and after the filtering. The forth figure shows the real and imaginary parts of the correlation coefficient (top plot) and the correlation in spatial domain [bottom, the values are real, so no imaginary number (red = 0)].

filt_COR2.m

Very similar to *filt_COR.m*, but is applied to 2D image. In other word, *fft* becomes *fft2*; *ifft* becomes *ifft2*, etc. I already resample and interpolate InSAR & DEM by using GMT, so I don't need to do this part in Matlab.

The computation of the complex cross-correlation is the same as *filt_COR.m* but is applied in 2D. The figure shown in this code compares two images before and after the filtering, and the difference before and after the filtering. The input signals can be InSAR and DEM or InSAR and InSAR.