Function to Convert InSAR Data into `.mat` Files

- Author: Miranda Holloway
- Date: Created 3/19/2023, Last Edited 3/20/2024

This code is originally adapted from post_process_data_ascending.m written by Dr. Roger Michaelides and my InSAR_to_MATLAB.m.

The purpose of this function is to convert InSAR data so that we can use it in MATLAB, specifically for `.amp`, `.cc`, `.int`, and `.unw` file types.

Before you can use this function, you must go through and edit the `filepath` variable (line 34). This variable represents the general folder where your InSAR data is located in sub-directories (NOT a specific folder containing InSAR data). This `filepath` variable should be a directory of subdirectories (for me, it is the directory where I scp my data files to on my local machine).

Arguments:

1. `dir` - The directory containing the InSAR data you want to be processed. Should be a char data type (use single quotes when specifying - for example, 'subdir', not "subdir").
2. `amp` - A boolean value that indicates if you want to process any `.amp` files in the specified sub-directory.
3. `cc` - A boolean value that indicates if you want to process any `.cc` files in the specified sub-directory.
4. `int` - A boolean value that indicates if you want to process any `.int` files in the specified sub-directory.
5. `unw` - A boolean value that indicates if you want to process any `.unw` files in the specified sub-directory.
6. `saving` - A boolean value that indicates if you want the processed data to be saved as .mat files in the specified subdirectory.

```
function [amps_out, coh_out, ints_out, phase_out, unw_phase_out] =
 insar2mat(dir,amp,cc,int,unw,saving)
filepath = strcat('C:\Users\mmpho\sent_test\',dir,'\');
addpath(filepath);

% Read dem.rsc to get image size
dat = split(fileread(strcat(filepath,'dem.rsc')));
dem_rsc = cell(((length(dat)-1)/2),2);
idx = 1;
for i = 1:(length(dat) - 1)
    if (mod(i,2) == 0)
        % Index is even
        % Value itself
        dem_rsc{idx,2} = str2double(dat{i});
        idx = idx + 1;
    else
        % Index is odd
        % Value title
```

```
        dem_rsc{idx,1} = dat{i};
    end
end

% Define image size
% Values taken from dem.rsc
nr = dem_rsc{1,2}; % number of x (range) pixels (WIDTH in dem.rsc)
naz = dem_rsc{2,2}; % number of y (azimuth) pixels (FILE_LENGTH in dem.rsc)

% Import and read intlist
cells = importdata('intlist');
N = length(cells);

% Preallocate arrays
if (amp)
    amps = zeros(nr,naz,N);
end
if (cc)
    coh = zeros(nr,naz,N);
end
if (int)
    phase = zeros(nr,naz,N);
    ints = zeros(nr,naz,N);
end
if (unw)
    unw_phase = zeros(nr,naz,N);
end

date_pair = cell(2,N);
doy_pair = cell(2,N);

% Read in the unwrapped phase (unw), coherence (coh), amplitude (amp) and
% unimodally-corrected unwrapped phase (uni)
disp('Processing data')

for i = 1:N
    disp(i)
    strint = cells{i};

    % Correlations
    if (cc)
        strcc1 = strrep(strint,'.int','.cc');
        filename_c = sprintf('%s',strcc1);
        fid = fopen(filename_c);
        dat = fread(fid,[2*nr,inf],'float','ieee-le');
        temp = dat((nr+1):end,:);
        coh(:,:,i) = temp;
        fclose(fid);
    end

    % Unwrapped phase
    if (unw)
        strunw1 = strrep(strint,'.int','.unw');
        filename = sprintf('%s',strunw1);
```

```
    fid = fopen(filename);
    dat = fread(fid,[2*nr,inf],'float','ieee-le');
    temp = dat(nr+1:end,:);
    unw_phase(:,:,i) = temp;
    fclose(fid);
end

% Interferograms
if (int)
    filename = sprintf('%s',strint);
    fid = fopen(filename);
    dat = fread(fid,[2*nr,inf],'float','ieee-le');
    temp = dat(1:2:end,1:naz)+1i*dat(2:2:end,1:naz);
    phase(:,:,i) = temp;
    fclose(fid);

    % Amplitude
    filename = sprintf('%s',strint);
    fid = fopen(filename);
    dat = fread(fid,[2*nr,inf],'float','ieee-le');
    temp = dat(1:2:2*nr-1,:)+1i*dat(2:2:2*nr,:);
    ints(:,:,i) = temp;
    fclose(fid);
end

% Amplitude
if (amp)
    stramp1 = strrep(strint,'.int','.amp');
    filename = sprintf('%s',stramp1);
    fid = fopen(filename);
    dat = fread(fid,[2*nr,inf],'float','ieee-le');
    temp = dat(1:2:2*nr-1,:)+1i*dat(2:2:2*nr,:);
    amps(:,:,i) = temp;
    fclose(fid);
end

% Date information
split1 = strsplit(strint,'_');
strint2 = split1{2};
split2 = strsplit(strint2,'.');
d1 = split1{1};
d2 = split2{1};

date1 = strcat(d1(5:6),'/',d1(7:8),'/',d1(1:4));
date2 = strcat(d2(5:6),'/',d2(7:8),'/',d2(1:4));

date1_vec = datetime(date1,'InputFormat','MM/dd/yyyy');
date2_vec = datetime(date2,'InputFormat','MM/dd/yyyy');

doy1 = day(date1_vec,'dayofyear');
doy2 = day(date2_vec,'dayofyear');

date_pair{1,i} = date1;
date_pair{2,i} = date2;
```

```
    doy_pair{1,i} = doy1;
    doy_pair{2,i} = doy2;
end

disp('Done with processing')

if (saving)
    disp(strcat("Saving variable(s) as .mat files to filepath ",filepath,'
 now'))

    if (amp)
        save(strcat(filepath,'amps_data.mat'),'amps',"-v7.3");
        disp('Saved amps_data.mat to folder')
    end
    if (cc)
        save(strcat(filepath,'coherence_data.mat'),'coh',"-v7.3");
        disp('Saved coherence_data.mat to folder')
    end
    if (int)
        save(strcat(filepath,'int_data.mat'),'ints',"-v7.3");
        disp('Saved int_data.mat to folder')

        save(strcat(filepath,'phase_data.mat'),'phase',"-v7.3");
        disp('Saved phase_data.mat to folder')
    end
    if (unw)
        save(strcat(filepath,'unw_phase_data.mat'),'unw_phase',"-v7.3");
        disp('Saved unw_phase_data.mat to folder')
    end
end

disp('Done with saving files')

% Assign outputs
if (amp)
    amps_out = amps;
else
    amps_out = 0;
end
if (cc)
    coh_out = coh;
else
    coh_out = 0;
end
if (int)
    ints_out = ints;
    phase_out = phase;
else
    ints_out = 0;
    phase_out = 0;
end
if (unw)
    unw_phase_out = unw_phase;
else
```

```
    unw_phase_out = 0;
end
end
```

```
Not enough input arguments.

Error in insar2mat (line 34)
filepath = strcat('C:\Users\mmpho\sent_test\',dir,'\');
```

Published with MATLAB® R2023a