Wave Parameter Extraction Pipeline

This MATLAB LiveScript serves as a live document to show the implementation of the SAR wave parameter extraction pipeline with additional documentation. Documentation notes are included before code blocks which explain the code which follows.

Table of Contents

Prerequisites

Software Prerequisites

SNAP ESA

SNAP is a tool developed by the European Space Agency (ESA) to process SAR data obtained from Sentinel Satellites. SNAP is used to pre-process data in this pipeline and is essential to install.

To install SNAP please do the following:

- Download SNAP here for your OS distribution
- Choose only the Sentinel Toolboxes installer
- Install SNAP and follow the onscreen instructions. It is only necessary to install the Sentinel-1 Toolbox

MATLAB Prerequisites

Python 3.8 (Need to check version in MATLAB)

```
pe = pyenv;
pyVersion = pe.Version;
```

```
if ~(strcmp(pyVersion, "3.8"))
    pe(Version="3.8")
end
```

CDS Toolbox

M_Map

M_Map is a mapping package for MATLAB which allows data plots on different world maps.

To use M_Map please do the following:

- Download the zipped package here and
- Add the extracted folder to your MATLAB path

Note: A copy of the M_Map package is included on this git repo in the following path:

wgrid

Obtain SAR Data

SAR data from Sentinel-1A can be downloaded from Copernicus SciHub. In order to download data, you need to register an account on the site.

Once you have registered your account, you can search for SAR data on the site by choosing the appropriate filters and region. The region you search is set by right-clicking and drawing a rectangle, otherwise you can left-click and draw the vertices of any polygon. Recomended filters are:

1. **Sensing Period:** As desired
2. **Satellite Platform:** S1A
3. **Product Type:** GRD
4. **Polarisation:** As desired

After searching with your filters, you will see all the available data within this region. Clicking on a red footprint on the map, shows a preview of these data, and the data can be downloaded from the pop-up.

Note: These data are in excess of 1GB, so large amounts of storage space is required. Use an external hard drive if possible.

Pre-Process SAR Data

After downloading SAR data from Copernicus SciHub, open SNAP and import the data. These data can be viewed as follows:

1. Click the '+' next to the dataset
2. Click the '+' next to the *Bands* folder
3. Open the desired band to preview the image

Thermal Noise Calibration

This needs to be done first

To remove thermal noise, click *Radar -> Radiometric -> S1 Thermal Noise Removal.* This will bring up a window where the following needs to be done.

1. Ensure the file will save as *BEAM-DIMAP*
2. Check that *Remove Thermal Noise* is checked in the *Processing Parameters* window
3. Select a specific polarisation (If desired)
4. Rename the output file (*Target Product)* if desired
5. Check the *Open in SNAP* checkbox

After clicking *Run,* SNAP will begin removing thermal noise from the selected data and once complete will open in the *Product Explorer* sidebar.

Radiometric Calibration

To radiometrically calibrate the SAR data, ensure that the **noise calibrated product is selected**, then click *Radar -> Radiometric -> Calibrate* This will bring up a window where the following needs to be done.

1. Ensure the file will save as *BEAM-DIMAP*
2. Check that **only** *Output sigma0 band* is checked in the *Processing Parameters* window
3. Select a specific polarisation (If desired)
4. Rename the output file (*Target Product)* if desired
5. Check the *Open in SNAP* checkbox

After clicking *Run,* SNAP will begin radiometric calibration from the selected data and once complete will open in the *Product Explorer* sidebar.

Export Incidence Angle

In order to access the incidence angle of each pixel in the the SAR data, the following process needs to be followed.

1. Click the '+' next to the *Tie-Point Grids* folder
2. Right-click on the *incident_angle* band
3. Select *Band Maths*
4. Rename the band to "Incidence_Angle" and click OK

Exporting as NetCDF

In order to export these pre-processed SAR data in a format supported by MATLAB, the following process needs to be followed.

1. Click *File*
2. Hover over the *Export* option, and select **NetCDF4-BEAM**
3. If your file explorer is opened, click on *Subset...* on the right hand side of the dialog box
4. This will bring up a new window with four different menus.
5. You can take a Spatial Subset of the image using either pixel or geographical coordinates in the *Spatial Subset* menu

6. Ensure that under the *Band Subset* menu, your desired bands are selected, as well as your created ***Incidence_Angle* band.**
7. Ensure that under the *Metadata Subset* menu, all options are selected
8. After checking all of these parameters, click OK, and name the file as desired and save it in your desired location

Optional

Take transects and subdivide

After exporting the SAR data as a NetCDF file, use the following commands to import the data and metadata into MATLAB.

```
filepath = "D:\UCT\EEE4022S\Data\CPT\smaller_subset_incidence.nc";
% Import data values
ncImport = ncinfo(filepath);
% Update the band to import as required
sarData = ncread(filepath,'Sigma0_VV');
metadata = ncinfo(filepath,'metadata');
incidenceAngle = ncread(filepath,'Incidence_Angle');
```

Define Transect Parameters

Enter the number of desired transects to take, the start point (in pixels), along with the angle at which to take the transects. This angle is calculated from the positive x-axis clockwise.

```
% Enter the number of transects
n = 3;
th = 10;
[transectData, positions] = get512Transects(sarData,1,1,th,n);
```

The original data, with shown transects can be plotted as follows.

```
figure(1)
imshow(sarData)
hold on;

for i = 1:n
    annotate512Transect(positions(i,1),positions(i,2),i,'red','black',1);
end

title('Sentinel-1A data with transects displayed')
hold off
```

The individual transects can be plotted as follows.

```
figure(2)
```

```
subplot(1,3,1)
imshow(transectData(:,:,1));
title(['Transect 1 at ',num2str(th),' degrees'])
subplot(1,3,2)
imshow(transectData(:,:,2));
title(['Transect 2 at ',num2str(th),' degrees'])
subplot(1,3,3)
imshow(transectData(:,:,3));
title(['Transect 3 at ',num2str(th),' degrees'])
```

Metadata can be imported using the `ncinfo` function. This large metadata structure can be slimmed down by filtering using the desired attributes for subsequent functions.

```
% Metadata
%metadata = ncinfo(filepath,'metadata');
req_atributes =
["first_near_lat","first_near_long","first_far_lat","first_far_long","last_near_lat"
,"last_near_long","last_far_lat","last_far_long","centre_lat","centre_lon",
"num_output_lines","num_samples_per_line"];
metadata_filtered = filterAttributesNetCDF(metadata.Attributes, req_atributes);
```

Metadata Extraction

Generate Wave Spectra

Before generating wave spectra, please source data from NOAA NCEP here. Ensure that you select data in the following manner.

1. Select data of the following form: **`gdas.date`**
2. Select the desired time (Either `00/`, `06/`, `12/`, or `18/`)
3. Select `wave/` data
4. Choose the `gridded/` data
5. Once completing all of the above, choose any data in the following form: `gdaswave.`**`timez`**`.`**`location`**`.0p25.f000.grib2,` where the **bold** values are user decided variables
6. After choosing a data set, right-click and copy the link address

Parse in your chosen dataset

```
noaaUrl = "https://nomads.ncep.noaa.gov/pub/data/nccf/com/gfs/prod/gdas.20231002/12/
wave/gridded/gdaswave.t12z.gsouth.0p25.f000.grib2";
```

Set the name for the saved downloaded `.grib2` file and the file will be downloaded and saved.

```
waveDownloadName = "wave_data.grib2";
waveFilePath = downloadNOAAWaveFile(noaaUrl,waveDownloadName);
```

Set the location of wgib and the downloaded `.grib2` filepath as well as choosing if you'd like to plot your downloaded data along with the type to plot

```
GribPath = "C:\Users\ryanj\OneDrive - University of Cape Town\4. Fourth Year\Second Semester\EEE4022S\repo\sar-parameter-extraction-pipeline\functions\";
wgrib2Path = "C:\Users\ryanj\Downloads";
waveStruct = getGribStruct(wgrib2Path,GribPath);
noaaPlot = true;
noaaValToPlot = 'direction';
if noaaPlot
    noaaDataPlot('miller',waveStruct,noaaValToPlot);
end
```

Set the location at which to generate wave spectra for

```
latitude = -34.2;
longitude = 18.6;
[gridLat, gridLon] = createLatLonGrid(latitude, longitude, resolution);
startLat = max(gridLat)
startLon = min(gridLon)
endLat = min(gridLat)
endLon = max(gridLon)
waveVals = getSubsetWaveVals(waveStruct,startLat,startLon,endLat,endLon);
```

Generate One-dimensional Wave Spectra

Define and instantiate variables for the one-dimensional wave spectra

```
imageSize = size(sarData,1);
w = linspace(0,2*pi,imageSize)';
f = linspace(0,1,imageSize)';
Hs = waveVals.significantWaveHeight(1,1);
T0 = waveVals.significantWavePeriod(1,1);
w0 = 2*pi./T0;
f0 = 1./T0;
gammaVal = 1.308;
multipleWaveSpectra = false;
if multipleWaveSpectra
    S = generateMultipleJONSWAP(waveVals,gammaVal,w,1);
else
    S = generateSingleJONSWAP(Hs,w0,gammaVal,w);
end
```

Generate Two-dimensional Wave Spectra

```
[D,theta] = generateDirectionalDistribution(waveVals,w,1);
E = generate2DWaveSpectrum(S,D);
```

SAR Spectrum Calculation

Hasselmann Procedure

Generate SAR spectrum of Ocean Waves

Inversion

Output Parameters