Test

March 19, 2019

In [1]:
```
# Name:    simple.py
# Purpose: Simple example of SeaIceDrift application
# Authors:      Anton Korosov
# Created:      26.08.2016
# Copyright:    (c) NERSC 2016
# Licence:
# This file is part of SeaIceDrift.
# SeaIceDrift is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, version 3 of the License.
# http://www.gnu.org/licenses/gpl-3.0.html
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
import os
import sys
import glob
import unittest
import inspect

import numpy as np
import matplotlib.pyplot as plt
plt.switch_backend('Agg')
from nansat import Nansat, Domain, NSR

from sea_ice_drift import SeaIceDrift

# download Sentinel-1 data from https://scihub.copernicus.eu/dhus
# or get small size sample files here:
# wget https://github.com/nansencenter/sea_ice_drift_test_files/raw/master/S1A_EW_GRDM
# wget https://github.com/nansencenter/sea_ice_drift_test_files/raw/master/S1B_EW_GRDM
# ==== ICE DRIFT RETRIEVAL ====

# open files, read 'sigma0_HV' band and convert to UInt8 image
f1 = 'S1D_EW_GRDM_1SDH_20190311T080051.tif'
f2 = 'S1D_EW_GRDM_1SDH_20190311T093915.tif'
sid = SeaIceDrift(f1, f2)
```

```
# apply Feature Tracking algorithm and retrieve ice drift speed
# and starting/ending coordinates of matched keypoints
uft, vft, lon1ft, lat1ft, lon2ft, lat2ft = sid.get_drift_FT()

# user defined grid of points:
lon1pm, lat1pm = np.meshgrid(np.linspace(-3, 2, 50),
                     np.linspace(86.4, 86.8, 50))

# apply Pattern Matching and find sea ice drift speed
# for the given grid of points
upm, vpm, apm, rpm, hpm, lon2pm, lat2pm = sid.get_drift_PM(
                                    lon1pm, lat1pm,
                                    lon1ft, lat1ft,
                                    lon2ft, lat2ft)

# ==== PLOTTING ====
# get coordinates of SAR scene borders
lon1, lat1 = sid.n1.get_border()
lon2, lat2 = sid.n2.get_border()

# prepare projected images with sigma0_HV
sid.n1.reproject(Domain(NSR().wkt, '-te -3 86.4 2 86.8 -ts 500 500'))
s01 = sid.n1['sigma0_HV']
sid.n2.reproject(Domain(NSR().wkt, '-te -3 86.4 2 86.8 -ts 500 500'))
s02 = sid.n2['sigma0_HV']

# plot the projected image from the first SAR scene
plt.imshow(s01, extent=[-3, 2, 86.4, 86.8], cmap='gray', aspect=12)
# plot vectors of sea ice drift from Feature Tracking
plt.quiver(lon1ft, lat1ft, uft, vft, color='r',
           angles='xy', scale_units='xy', scale=0.5)
# plot border of the second SAR scene
plt.plot(lon2, lat2, '.-r')
# set X/Y limits of figure
plt.xlim([-3, 2])
plt.ylim([86.4, 86.8])
plt.savefig('sea_ice_drift_FT_img1.png', dpi=150, bbox_inches='tight', pad_inches=0)
plt.close('all')

# plot the projected image from the second SAR scene
plt.imshow(s02, extent=[-3, 2, 86.4, 86.8], cmap='gray', aspect=12)
# filter only high quality pixels
gpi = rpm > 0.4
# plot vectors of sea ice drift from Feature Tracking, color by MCC
plt.quiver(lon1pm[gpi], lat1pm[gpi], upm[gpi], vpm[gpi], rpm[gpi],
           angles='xy', scale_units='xy', scale=0.5)
# plot border of the first SAR scene
plt.plot(lon1, lat1, '.-r')
```

```
    # set X/Y limits of figure
    plt.xlim([-3, 2])
    plt.ylim([86.4, 86.8])
    plt.savefig('sea_ice_drift_PM_img2.png', dpi=150, bbox_inches='tight', pad_inches=0)
    plt.close('all')
```

```
    ---------------------------------------------------------------------------

    ValueError                                Traceback (most recent call last)

    <ipython-input-1-c05e7f8656cf> in <module>
     35 f1 = 'S1D_EW_GRDM_1SDH_20190311T080051.tif'
     36 f2 = 'S1D_EW_GRDM_1SDH_20190311T093915.tif'
---> 37 sid = SeaIceDrift(f1, f2)
     38
     39 # apply Feature Tracking algorithm and retrieve ice drift speed


    ~/anaconda3/envs/nansat/lib/python3.6/site-packages/sea_ice_drift/seaicedrift.py in __
     36
     37         # get Nansat
---> 38         self.n1 = get_n(self.filename1, **kwargs)
     39         self.n2 = get_n(self.filename2, **kwargs)
     40


    ~/anaconda3/envs/nansat/lib/python3.6/site-packages/sea_ice_drift/lib.py in get_n(filen
    212         n.resize(factor, resample_alg=-1)
    213     # get matrix with data
--> 214     img = n[bandName]
    215     # convert to dB
    216     if not denoise and dB:


    ~/anaconda3/envs/nansat/lib/python3.6/site-packages/nansat/nansat.py in __getitem__(se
    205         """
    206         # get band
--> 207         band = self.get_GDALRasterBand(band_id)
    208         # get expression from metadata
    209         expression = band.GetMetadata().get('expression', '')


    ~/anaconda3/envs/nansat/lib/python3.6/site-packages/nansat/nansat.py in get_GDALRasterB
    526
    527         # get band number
--> 528         bandNumber = self.get_band_number(band_id)
    529         # the GDAL RasterBand of the corresponding band is returned
```

```
    530         return self.vrt.dataset.GetRasterBand(bandNumber)


~/anaconda3/envs/nansat/lib/python3.6/site-packages/nansat/nansat.py in get_band_numbe
   1283             raise ValueError('Cannot find band %s! '
   1284                              'band_number is from 1 to %s'
-> 1285                              % (str(band_id), self.vrt.dataset.RasterCount))
   1286
   1287         return band_number


ValueError: Cannot find band {'name': 'sigma0_HV'}! band_number is from 1 to 1
```

In [ ]: