1. Welcome

Welcome to the **Hands-on Lab on Polarimetric UAVSAR Data Processing for Land Cover / Land Use Change Applications**. Synthetic aperture radar (SAR) is a powerful tool for mapping and monitoring the characteristics of terrestrial landscapes. SAR polarimetry allows us to extract additional information from SAR data as compared to conventional radar backscatter, by separating the scattering mechanisms mixed in the radar return.

The purpose of this tutorial is to introduce basic polarimetric tools that can be used in the context of land cover / land use applications. In the next sessions we will walk you through the steps to:

(1) Obtain the necessary inputs to perform polarimetric analyses with UAVSAR GRD products.
(2) Familiarize yourself with polarimetric decompositions and classification techniques.
(3) Apply free, open source software package PolSARPro to analyze a forest fire dataset.
(4) Understand the advantages, challenges and limitations of working with airborne SAR datasets.

Let’s get started and learn more about the data and software used in this practice.

2. The PolSARPro Tool

PolSARPro is a freely available, open-source software package funded by the European Space Agency (ESA) to process polarimetric radar data (http://earth.eo.esa.int/polsarpro/). Although PolSARPro has a basic graphical interface, in this tutorial we will call its tools from the linux command line, which is a highly flexible and scalable approach to process UAVSAR data. We will use the PolSARPro version currently available on the web (v4). A new version of PolSARPro (v5) is being developed by ESA and will be distributed in the near future.

Note:

- ➔ Commands are case sensitive.
- ➔ All relevant commands are preceded by numbers in square brackets (e.g. [23]) and can be copied and pasted into your terminal.
- ➔ Steps to be typed into the Remote Desktop are followed by “RD” (e.g. [23-RD])
- ➔ Steps in blue color are short “do it yourself” assignments.

PolSARPro routines are located in this directory:
`/home/ubuntu/install/polsarpro/Soft`
And PDFs with detailed instructions for each routine are here:
`/home/ubuntu/install/polsarpro/TechDoc/C_Routines`

For this tutorial, all PolSARPro routines are available in your current path. To see a function’s usage, just type its name with no arguments.

For example, if you type:
[1] `UTM_LatLong.exe`
You should see:

```
> A processing error occured !
> UTM_LatLong x_coord y_coord UTM_zone
```

The error message can be disregarded as it is just telling us we are calling the routine with no input parameters.

The function `UTM_LatLong` is run by typing its name followed by its 3 arguments separated by spaces: the UTM Easting, UTM Northing, and the UTM zone. Type:
[2] `UTM_LatLong.exe 500000 0  23`
You should see:

```
> longitude = -45.000000    latitude = 0.000000
```

More detailed documentation can be found in the Documentation files.

Next, we will go over a few basic commands as we review the input files for this exercise.

3. Input Files

In the first exercise we will compare pre-fire and post-fire polarimetric UAVSAR images acquired for the 2009 Station Fire (CA). The input UAVSAR data are located in `/home/ubuntu/data/lab10/fire`. Type `cd` to move to the data directory.

[3] `cd /home/ubuntu/data/lab10/fire`

At any time, use the command `pwd` to see your current directory.

[4] `pwd`

`> home/ubuntu/data/lab10/fire`

Use the command `ls` to list the files in a directory.

[5] `ls`

As you can see, the `fire` directory contains 2 directories corresponding to 2 UAVSAR flights: flight_09010 (pre fire: February 26, 2009) and flight_09072 (post fire: September 18, 2009). The other directories will be used to store outputs.

See the files available for the pre-fire flight:

[6] `cd flight_09010`

[7] `ls`

Go up one directory

[8] `cd ../`

Take a look at the post-fire files

[9] `ls flight_09072`

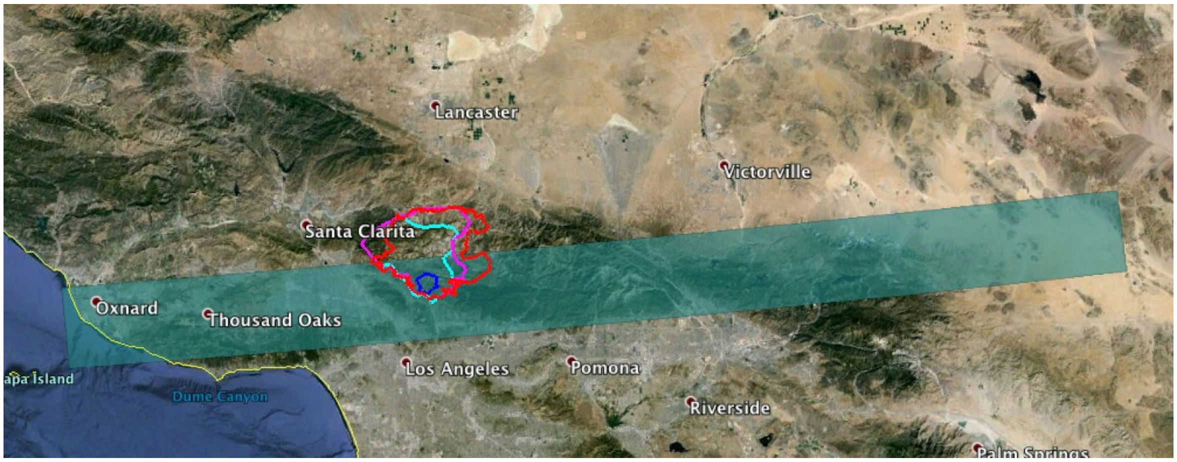
Each flight directory contains:

- ❏ Six ground-projected, polarimetric data files (*.grd).
- ❏ One file containing the local incidence angle in radians for each image pixel (*.inc). We'll use this file to mask out severe topography.
- ❏ One text file containing metadata information such as image dimensions, resolution, and geographic information (*.ann).

More details on file formats:
http://uavsar.jpl.nasa.gov/science/documents/polsar-format.html

In the next page we will see how to read these files within PolSARPro.

The transparent green rectangle represents the area imaged by UAVSAR (297 km long), whereas polygons show the fire progression between August 29 2009 and Sept 01 2009.



Lancaster
Victorville
Santa Clarita
Oxnard
Thousand Oaks
apa Island
Dume Canyon
Los Angeles
Pomona
Riverside
Palm Springs

4. Importing, Cropping, and Multi-looking UAVSAR GRD Images

We will now import the polarimetric UAVSAR GRD images in PolSARPro. As we are interested in a limited region (ROI) around the fire, we will crop the images and also take looks to reduce speckle noise. Importing, cropping and multi-looking can be done in PolSARPro with a single routine, `uavsar_convert_grd_MLK_T3.exe`

Let's look at the routine's usage by typing
[10] `uavsar_convert_grd_MLK_T3.exe`

The arguments are:

- `HeaderFile` = UAVSAR annotation file (*.ann)
- `in_dir` = Input directory
- `out_dir` = Output directory
- `Off_lig` = Offset rows
- `Off_col` = Offset columns
- `Nligfin` = Number of rows in user-defined ROI
- `Ncolfin` = Number of columns in user-defined ROI
- `Nlook_col` = Multilook factor for columns
- `Nlook_lig` = Multilook factor for rows

Two directories have been already created to save the outputs for this step:
`/home/ubuntu/data/lab10/fire/output_09010/T3`
`/home/ubuntu/data/lab10/fire/output_09072/T3`

The command below imports, crops and multi-looks the pre-fire images.
[11] `cd /home/ubuntu/data/lab10/fire`
[12] `uavsar_convert_grd_MLK_T3.exe flight_09010/*ann flight_09010 output_09010/T3 2500 21000 1800 4000 2 2`

This means that starting at pixel position (2500, 21000), we define an area that is 1800 rows x 4000 columns and and average 2x2 pixels in 1 pixel. PolSARPro will crop all polarimetric images contained in the input directory, provided that their names match the provided annotation file. This includes all polarimetric bands and the local incidence angle image.

List output files by typing:
[13] `ls output_09010/T3`
You should see:
```
> T11.bin  T12_imag.bin  T12_real.bin  T13_imag.bin  T13_real.bin
T22.bin  T23_imag.bin  T23_real.bin  T33.bin  config.txt  dem.bin
```

Do it yourself: call `uavsar_convert_grd_MLK.exe` once again to crop the post-fire images from flight 09072. Use the same Region Of Interest as the example above.

The routine `uavsar_convert_grd_MLK_T3.exe` creates the file `config.txt` as well as 9 cropped data (*bin) files. File names reflect the convention used in the polarimetry literature.

File	Description
T11.bin	$\vert HH+VV\vert^2/2$
T22.bin	$\vert HH-VV\vert^2/2$
T33.bin	$2*HV^2$
T12_real.bin T12_imag.bin	Complex correlation (HH+VV)(HH-VV)*/2
T13_real.bin T13_imag.bin	Complex correlation (HH+VV)HV*
T23_real.bin T23_imag.bin	Complex correlation (HH-VV)HV*

The "T3" term refers to the coherency matrix used to represent polarimetric data. We could have also used the covariance "C3" matrix.

The output file dem.bin is actually the cropped incidence angle file (this will be fixed in future releases of PolSARPro), so let's rename it:

[14] `cd /home/ubuntu/data/lab10/fire/output_09010/T3/`
[15] `mv dem.bin inc.bin`
[16] `cd /home/ubuntu/data/lab10/fire/output_09072/T3/`
[17] `mv dem.bin inc.bin`

The size of the new images is recorded in the file `config.txt`. Type:

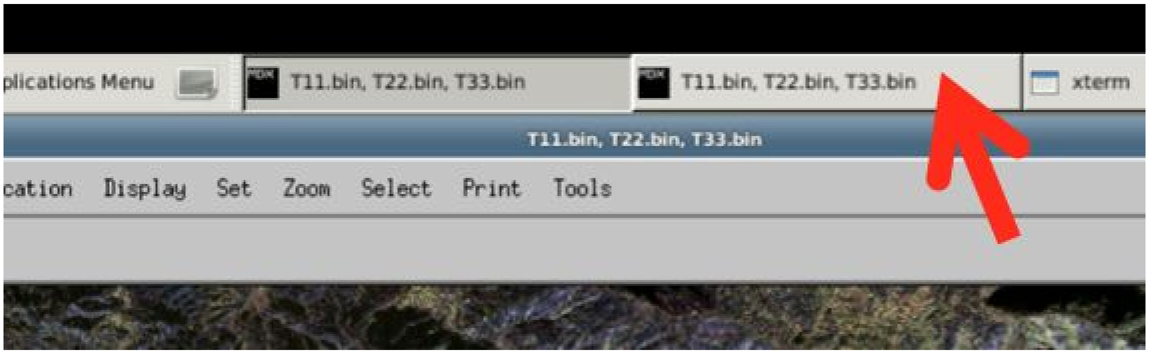
[18] `cat /home/ubuntu/data/lab10/fire/output_09010/T3/config.txt`
[19] `cat /home/ubuntu/data/lab10/fire/output_09072/T3/config.txt`

Make a note of the number of columns and rows (you will need them for the next steps). They should match for the pre-fire and post-fire images.

Let's look at the cropped backscatter images by launching the Remote Desktop interface. From the Remote Desktop, the command below displays a false color composite **HH+VV** **2HV** **HH-VV** of the pre-fire image:

[20-RD] `cd /data/lab10`
[21-RD] `sh mdx_polsar.sh fire_cropped`

You can switch between pre-fire and post-fire by clicking on the mdx window tab.


plications Menu
T11.bin, T22.bin, T33.bin
T11.bin, T22.bin, T33.bin
xterm
T11.bin, T22.bin, T33.bin
cation
Display
Set
Zoom
Select
Print
Tools

5.Compensation for Polarimetric Orientation Angle

Now we will correct for possible polarimetric distortions induced by azimuth terrain slopes.

We start from this directory:
[22] `cd /home/ubuntu/data/lab10/fire/`

We already have two directories where we'll save the corrected images:
```
output_09010/T3_poa
output_09072/T3_poa
```

Copy the config.txt file into the output `T3_poa` directories:
[23] `cp output_09010/T3/config.txt output_09010/T3_poa`
[24] `cp output_09072/T3/config.txt output_09072/T3_poa`

Check the usage of the orientation angle correction routine:
[25] `orientation_estimation_T3.exe`
```
> orientation_estimation_T3 in_dir out_dir offset_lig offset_col
sub_nlig sub_ncol
```

Apply orientation angle correction to pre-fire image:
[26] `orientation_estimation_T3.exe output_09010/T3 output_09010/T3_poa 0 0 900 2000`

Do it yourself: call `orientation_estimation_T3.exe` to apply the polarimetric orientation angle compensation to post-fire images from flight 09072.

Go to the Remote Desktop and display the pre-fire image before and after the compensation for azimuth distortion.
[27-RD] `sh mdx_polsar.sh fire_poa`

6.Eigenvalue-based Decomposition: H/A/Alpha Decomposition

Let’s now apply the H/A/alpha decomposition to the cropped data. First check out the usage:
`> h_a_alpha_decomposition_T3.exe`

These are the input arguments:

- `in_dir` = Input directory containing the T3 matrix produced with `orientation_estimation_T3.exe`
- `out_dir` = Output directory
- `Nwin` = Window size in pixels
- `offset_lig` = Offset rows
- `offset_col` = Offset columns
- `sub_nlig` = Number of rows in input images from the config.txt file
- `sub_ncol` = Number of columns in input images from the config.txt file
- `alpbetdelgam` = Alpha, Beta, Delta, Gamma, Lambda [1=yes, 0=no]
- `lambda` = Lambda component [1=yes, 0=no]
- `alpha` = Alpha component [1=yes, 0=no]
- `entropy` = Entropy estimate [1=yes, 0=no]
- `anisotropy` = Anisotropy [1=yes, 0=no]

The options below refer to combinations of Entropy and Anisotropy components:

- `CombHA` = HA [1=yes, 0=no]
- `CombH1mA` = H(1-A) [1=yes, 0=no]
- `Comb1mHA` = (1-H)A [1=yes, 0=no]
- `Comb1mH1mA` = (1-H)(1-A) [1=yes, 0=no]

Then make sure you are in the Station Fire directory:
[28] `cd /home/ubuntu/data/lab10/fire/`

Call the function of the H/A/alpha decomposition on the pre-fire dataset. Note that we only set three output options to 1 in order to get the alpha, entropy, and anisotropy components.
[29] `h_a_alpha_decomposition_T3.exe output_09010/T3_poa output_09010/T3_poa 1 0 0 900 2000 0 0 1 1 1 0 0 0 0`

Do it yourself: apply the H/A/Alpha decomposition to the post-fire dataset.

Go in the Remote Desktop. Let’s see what the entropy images look like. We can mask out extreme local incidence angle values < 0.2 radians (11.5 degrees) and > 1.4 radians (80 degrees). Entropy values range between 0 and 1.
[30-RD] `sh mdx_polsar.sh fire_entropy`

Close the entropy mdx images and display the alpha angle images. What differences do you

expect to see when comparing pre and post fire conditions?
[31-RD] `sh mdx_polsar.sh fire_alpha`

Notice results are not influenced by terrain slope, although we still see geometric topographic effects. Alpha values range between 0 deg and 90 deg:

If alpha is close to..	The dominant scattering mechanism is..
0 deg	surface
45 deg	volume
90 deg	double bounce

7.Model-based Decomposition: Van Zyl Decomposition

We will now run the Van Zyl polarimetric decomposition to separate the scattering contribution from the surface, the double bounce and the volume.

Pre-fire:

[32] `cd /home/ubuntu/data/lab10/fire/output_09010`

[33] `vanzyl_3components_decomposition_T3.exe T3_poa/ T3_poa/ 1 0 0 900 2000`

Post-fire:

[34] `cd /home/ubuntu/data/lab10/fire/output_09072`

[35] `vanzyl_3components_decomposition_T3.exe T3_poa/ T3_poa/ 1 0 0 900 2000`

Instead of displaying the RGB images with mdx we compare the histograms of volume component for the fre-fire and the post fire (values in decibels using 100 bins):

Pre-fire:

[36] `cd /home/ubuntu/data/lab10/fire/output_09010/T3_poa`

[37] `echo $[2000*900] > npts.txt`

[38] `statistics_histogram.exe VanZyl3_Vol.bin VanZyl3_Vol_hist.txt npts.txt float db10 100 1 0 0`

Post-fire:

[39] `cd /home/ubuntu/data/lab10/fire/output_09072/T3_poa`

[40] `echo $[2000*900] > npts.txt`

[41] `statistics_histogram.exe VanZyl3_Vol.bin VanZyl3_Vol_hist.txt npts.txt float db10 100 1 0 0`

Now let’s plot the histograms of the volume Van Zyl scattering component:

[42-RD] `sh mdx_polsar.sh fire_volhist`

8. Polarimetric Classification

For this exercise we will use an image acquired in Monterey Bay, California. We'll repeat many steps from previous sessions to make an H/A/Alpha decomposition, which will be used here as input to a polarimetric image partitioner.

Go to data directory and create two new directories to save outputs:

[43] `cd /home/ubuntu/data/lab10/monterey`

Let's read in the image and multilook it by a factor of 2. This time we'll read the entire image with no cropping.

[44] `uavsar_convert_grd_MLK_T3.exe input_23025/SanAnd_23025_12030_010_120521_L090_CX_03.ann input_23025/ T3 0 0 0 0 2 2`

Compensate for polarimetric orientation angle. The values 7282 and 10033 correspond to the number of rows and columns taken from the config.txt file. Copy the config file to the T3_poa directory.

[45] `orientation_estimation_T3.exe /home/ubuntu/data/lab10/monterey/T3 /home/ubuntu/data/lab10/monterey/T3_poa/ 0 0 7282 10033`

[46] `cp /home/ubuntu/data/lab10/monterey/T3/config.txt /home/ubuntu/data/lab10/monterey/T3_poa`

The H/A/Alpha decomposition serves as input for the classification:

[47] `h_a_alpha_decomposition_T3.exe T3_poa T3_poa 1 0 0 7282 10033 0 0 1 1 1 0 0 0 0`

To partition the image according to the H/Alpha plane, type:

[48] `h_a_alpha_planes_classifier.exe T3_poa T3_poa 0 0 7282 10033 1 0 0 /home/ubuntu/install/polsarpro/Config/Planes_H_A_Alpha_ColorMap9.pal`

This produces a binary classified image H_alpha_class.bmp containing 9 classes, and two quicklook *bmp files: H_alpha_class.bmp and H_alpha_occurence_plane.bmp. Let's open the two *bmp files.

[49-RD] `sh mdx_polsar.sh monterey_class`