SAR Exploration Notebooks

SAR Exploration Notebooks

There will be a Google Colab Notebook showing examples of data visualization and interpretation for each SAR application.

Information included in the notebooks:

- Qualitative Browse Variables (How does the user know there is something in the image related to this purpose *before* they develop/implement any algorithms?)
- Useful SAR Wavelength(s)
- Useful Products (polarization, InSAR, coherence, etc.)
- Relevant SAR Sensors
- Instructions to View SAR data(RGB combo/gray scale)
- Interpretation

GitHub Repo: https://github.jpl.nasa.gov/nspinto/rgb_fun/tree/development

- 15 Folders – one for each SAR Application and one Template Folder
- Template Folder:
  - Jupyter Notebook template

In each application folder:

- *.ipynb
  - Jupyter Notebook that reads in and creates RGB or performs other simple analyses
  - Includes descriptions of SAR products and interpretations
- uavsar_data folder
  - Folder containing UAVSAR geotiffs cropped to areas of interest
- Pictures folder
  - Pictures included in the notebooks
- Tiles folder
  - Image tiles for the interactive maps are stored in these folders

#	SAR Application	UAVSAR Example	Products
1	**RGB**	Creating RGB Composite and Image Stretching	HH, HV, VV
2	**Biomass**	San Gabriel Mountains Station Fire	HV Ratio, Radar Vegetation Index (RVI), RGB Multi-Temporal Composite
3	**Crop Classification**	Central Valley-San Joaquin Agriculture	HV Backscatter Time Series
4	**Fire**	Los Angeles La Tuna Fire visualization, Thomas Fire Coherence	RGB, Coherence
5	**Fire Data**	Los Angeles La Tuna Fire	HV Transects, HV Time Series
6	**Flood**	Hurricane Florence	RGB, Coherence
7	**Forest Disturbance**	Amazon Gold Mining, Napa Valley Deforestation, Mammoth Tree Die Off	RGB, Coherence
8	**Inundation (Wetland Delineation)**	Balona Wetland, Marsh Island, Fay Slough Wildlife Area, Sierpe Mangroves	RGB
9	**Landcover (Urban Areas)**	Munich, Quebec, San Diego, New Orleans	RGB
10	**Landslide**	Thomas Fire's Montecito Debris Flows	HH Coherence
11	**Oil Spill**	Gulf Coast, North Sea	HH/VV Backscatter Ratio, HH and VV in dB
12	**Open Water (Surface Water Extent)**	Panama Canal	RGB, HH Thresholding, HH/VV Ratio
13	**Sea Ice**	Beaufort Sea	Intensity Images VV, HH
14	**Soil Moisture**	Tonzi Ranch Research Site	HV/HV Ratio, VV Power
15	**Template**	Jupyter Notebook Layout	

Google Colab Notebook Template

Part 1: Title, Map of Example Location, Table of Contents

Template.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Title [Location and Year of SAR Example]

Map highlighting location of example(s)

This Jupyter Notebook explores UAVSAR data over the [date] [Example]. The content of this notebook is as follows:

Site Map

Table of Contents

Part 2: Instructions on running the notebook, upload UAVSAR files, import and install the python packages

Running the Notebook

- To run the content, **select the play button** in the upper left corner of each code cell or **press shift-enter**.
- To access the data for this notebook, download the data at this link. Click this link to a shared Google Drive data folder and "Download All." (This link is currently in my @jpl.caltech.edu Google Account and has sharing limitations. Moving [illegible] this would be a publically accessible folder/data download link).
  - Link to data folder: https://drive.google.com/drive/folders/1fXENic894ElRJgACE40Lx5Jfpg_olvac?usp=sharing
  - Upload these downloaded files to Google Colab by running the cell below. A "Choose Files" button should appear. Select all the files in the downloaded folder. (Note: these will upload to a temporary "content" folder).
  - As a note, uploading larger files will take longer. If an upload error is encountered, try using Chrome instead of Safari. Another (faster and less data volume restricted) upload option is to expand the folder system on the left (select the folder icon) and click the upload button (icon of an upward arrow over a sheet of paper) and select all the files in the downloaded folder to upload to your "content" folder.

```
[ ] from google.colab import files
    uploaded = files.upload()
```

Choose Files no files selected Cancel upload

Python Packages:

This routine uses the following python libraries. Some are already included in the Google Colab environment and others are installed in the cell below before imported.

1. matplotlib
2. numpy
3. rasterio
4. etc.

```
[1] ## The packages will take around a minute to install
    !pip install rasterio --quiet

    import rasterio as rio
    import numpy as np

        |████████████████████████████████| 19.3MB 1.2MB/s
```

Instructions on using Google Colab

Import UAVSAR data

Import and download necessary python packages

Part 3: Section walking through SAR examples using python code and text descriptions

1. Section 1 of Example

[Insert Description and Introduction]

```
[ ] ## Code Cell
```

2. Section 2 of Example

[Insert Description]

[New text cell to describe code below]

```
[2] def read_file(path):
      with rio.open(path) as ds:
        band = ds.read(1)
        return band

    #var = read_file('*.tif')
    #var2 = read_file('*.tif')
```

Python code reading in, displaying, and exploring SAR data. Each code cell accompanied by descriptive text.

Google Colab Notebook: Fire Application Example

This is an example Google Colab Notebook exploring SAR data over the La Tuna Fire. The notebook can be accessed through a shareable link and begins with text describing the notebook's contents and how to run the notebok/upload data. The screenshot to the right shows how to import and install new python packages to Google Colab's environment.

UAVSAR_FireData.ipynb

File Edit View Insert Runtime Tools Help Last saved at July 8

+ Code + Text

2017 La Tuna Fire Los Angeles, California

This Jupyter Notebook explores UAVSAR data over the 2017 La Tuna Fire in Los Angeles, California. The content of this notebook is as follows:

0. Import Python Packages and Authenticate Notebook and data
1. UAVSAR RGB Composite of La Tuna Fire
2. Create and Plot HV Transects
3. Time Series (dB)

Running the Notebook

- To run the content, **select the play button** in the upper left corner of each code cell or **press shift-enter**.
- To access the data for this notebook, download the data at this link. Click this link to a shared Google Drive data folder and "Download All." (This link is currently in my @jpl.caltech.edu Google Account and has sharing limitations. Moving forward, this would be a publically accessible folder/data download link).
  - Link to data folder: https://drive.google.com/drive/folders/1fXENic894ElRJgACE40Lx5Jfpg_olvac?usp=sharing
  - Upload these downloaded files to Google Colab by running the cell below. A "Choose Files" button should appear. Select all the files in the downloaded folder. (Note: these will upload to a temporary "content" folder).
  - As a note, uploading larger files will take longer. If an upload error is encountered, try using Chrome instead of Safari. Another (faster and less data volume restricted) upload option is to expand the folder system on the left (select the folder icon) and click the upload button (icon of an upward arrow over a sheet of paper) and select all the files in the downloaded folder to upload to your "content" folder.

```
[ ] from google.colab import files
    uploaded = files.upload()
```

Choose Files no files selected Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

UAVSAR_FireData.ipynb

File Edit View Insert Runtime Tools Help Last saved at July 8

+ Code + Text

Python Packages:

This routine uses the following python libraries. Some are already included in the Google Colab environment and others are installed in the cell below before imported.

1. rasterio
2. rasterstas
3. geopandas
4. varname
5. rioxarray
6. matplotlib
7. numpy
8. plotly
9. skimage
10. pathlib

```
## The packages will take around a minute to install
!pip install rasterio --quiet
!pip install rasterstats --quiet
!pip install geopandas --quiet
!pip install rioxarray --quiet

import rasterio as rio
import rasterio.plot
from rasterio.plot import show
import rasterstats as rs
import rioxarray as rxr ## gdal Readasarray equivalent
import geopandas as gpd
import pandas as pd
from geopandas import GeoDataFrame
from shapely.geometry import Point
from shapely.geometry import Polygon
import matplotlib.pyplot as plt
import plotly.graph_objects as go
import numpy as np
from pathlib import Path
from skimage.measure import profile_line
import math
```

```
     |████████████████████████████████| 19.3MB 1.3MB/s
     |████████████████████████████████| 133kB 27.0MB/s
     |████████████████████████████████| 15.4MB 205kB/s
     |████████████████████████████████| 1.0MB 29.5MB/s
     |████████████████████████████████| 6.6MB 41.8MB/s
     |████████████████████████████████| 51kB 6.2MB/s
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
    Preparing wheel metadata ... done
  Building wheel for rioxarray (PEP 517) ... done
```

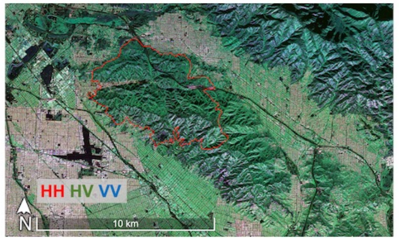
Section 1 and Section 2 of the Fire Application Notebook are below. The first section introduces the SAR example and shows the ability to embed data links and UAVSAR images. The second section reads in and plots transects of the UAVSAR data. The Google Colab environment allows for a mixture of descriptive text, python code, image display, and data plotting.

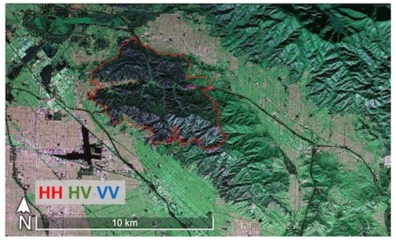
1. UAVSAR RGB Composite of La Tuna Fire

The RGB UAVSAR images below show the September 2017 La Tuna Fire in the Verdugo Mountains, Los / UAVSAR flightline SanAnd_08525 was acquired October 23, 2014 (Pre-fire) and November 02, 2017 (Pos perimeter extent is overlaid in red.

(Pre-Fire Top) 2014 Product Page: https://uavsar.jpl.nasa.gov/cgi-bin/product.pl?jobName=SanAnd_08525_14158_003_141023_L090_CX_01#data

(Post-Fire Bottom) 2017 Product Page: https://uavsar.jpl.nasa.gov/cgi-bin/product.pl?jobName=SanAnd_08525_17122_003_171102_L090_CX_01#data





2. Create and Plot HV Transects (Linear Power)

Below is a transect over the Verdugo mountains, spanning burned and unburned regions of the mountain range after the September 2017 La Tuna Fire. UAVSAR's HV polarization is used due to HV's sensitivity to vegetation change and transects from images pre and post fire can be uesd compare changes in HV backscatter values between areas burnt in the fire and unburnt areas. Burnt areas should have decreased HV backscatter.

Read in UAVSAR Data

```
def read_file(path):
  with rio.open(path) as ds:
    band = ds.read(1)
    return band

HH2014 = read_file('SanAnd_08525_14158_003_141023_L090HHHH_CX_01_cropped.grd.tif')
HV2014 = read_file('SanAnd_08525_14158_003_141023_L090HVHV_CX_01_cropped.grd.tif')
VV2014 = read_file('SanAnd_08525_14158_003_141023_L090VVVV_CX_01_cropped.grd.tif')

HH2017 = read_file('SanAnd_08525_17122_003_171102_L090HHHH_CX_01_cropped.grd.tif')
HV2017 = read_file('SanAnd_08525_17122_003_171102_L090HVHV_CX_01_cropped.grd.tif')
VV2017 = read_file('SanAnd_08525_17122_003_171102_L090VVVV_CX_01_cropped.grd.tif')

print(HH2014)
```

```
[[0.00456402 0.00456402 0.00731359 ... 0.00337009 0.00337009 0.00323988]
 [0.00771518 0.00771518 0.00538406 ... 0.00509634 0.00509634 0.0024167 ]
 [0.01003053 0.01003053 0.00976056 ... 0.00391764 0.00391764 0.00345487]
 ...
 [0.67873293 0.32198873 0.4658051  ... 0.05899433 0.04811405 0.05239956]
 [0.539634   0.32198873 0.4658051  ... 0.14835425 0.05932159 0.059321591
 [0.16789311 0.48786396 0.48786396 ... 0.03596146 0.16470343 0.16470343]]
```

Specify Pixel Coordinates

```
x0, y0 = 1300, 700
x1, y1 = 2200, 1300
```

Specify transect width with "linewidth" and transect output values with "reduce_func"

```
transect_2014 = profile_line(HV2014, (y0, x0), (y1, x1), linewidth=10, mode='nearest', reduce_func=np.median)
transect_2017 = profile_line(HV2017, (y0, x0), (y1, x1), linewidth=10, mode='nearest', reduce_func=np.median)
```

Plot 2014 and 2017 Data

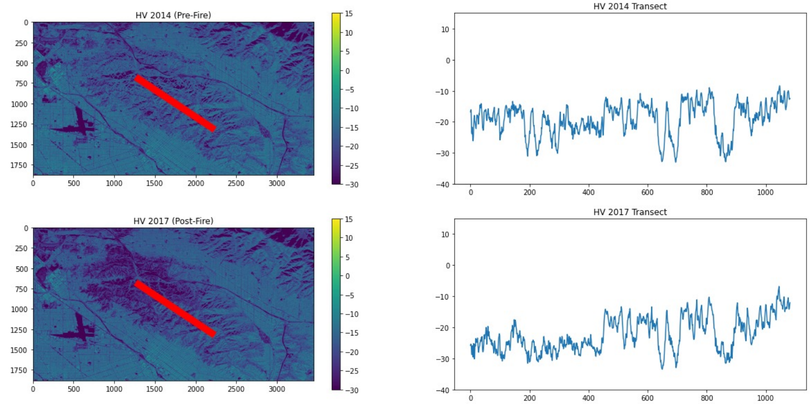
```
############## 2014 ##############
cm = ['viridis']
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(20, 10))
color = axes[0,0].imshow(HV2014, vmin=-30, vmax=15, cmap=cm[0])
axes[0,0].plot([x0, x1], [y0, y1], 'ro-', lw=10)
axes[0,0].set_title('HV 2014 (Pre-Fire)')
fig.colorbar(color, ax=axes[0,0])

axes[0,1].plot(transect_2014)
axes[0,1].set_ylim([-40, 15])
axes[0,1].set_title('HV 2014 Transect')

############## 2017 ##############
color = axes[1,0].imshow(HV2017, vmin=-30, vmax=15, cmap=cm[0])
axes[1,0].plot([x0, x1], [y0, y1], 'ro-', lw=10)
axes[1,0].set_title('HV 2017 (Post-Fire)')
fig.colorbar(color, ax=axes[1,0])

axes[1,1].plot(transect_2017)
axes[1,1].set_ylim([-40, 15])
axes[1,1].set_title('HV 2017 Transect')

plt.show()
```



Section 3 of the Fire Application Notebook is below. This section demonstrates another way to explore SAR data over a fire. The steps below are creating a time series from UAVSAR data extracted over burnt and unburnt areas.

3. Time Series (dB)

Below are subsets from UAVSAR data collection extent over the Verdugo Mountains - half over the burned area and the other half dispersed over unburnt areas. In the HV and HH polarizations, variations in values are visible between burned and unburned subsets.

The UAVSAR stack read in below was converted from linear power to decibels (dB). Formula: 10*log10(DN)

First, read in stack of UAVSAR PolSAR data from 2009 and 2020.

```
[ ] uavsar_stack = rio.open('uavsar_bands.tif')
```

Then, read in shapefile of pre-selected buffered points around burnt and unburnt areas.

```
[ ] gdf_plots = gpd.read_file(data_path + "gdf_buffer.shp")
    gdf_burnt = gdf_plots[gdf_plots['fire']=='burnt']
    gdf_notburnt = gdf_plots[gdf_plots['fire']=='notburnt']
```

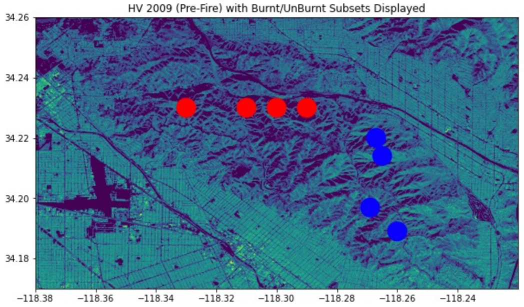
Plot the buffered points on top of the UAVSAR image to see distribution of points in relation to the study area.

```
fig, ax = plt.subplots(figsize=(10, 10))
cm = ['viridis']

ax = gdf_burnt.plot(ax=ax, facecolor='red', edgecolor='red', linewidth=2)
ax = gdf_notburnt.plot(ax=ax, facecolor='blue', edgecolor='blue', linewidth=2)
ax.set_title('HV 2009 (Pre-Fire) with Burnt/UnBurnt Subsets Displayed')

show((uavsar_stack, 1), ax=ax, vmin=-25, vmax=0, cmap=cm[0])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f704fbe4710>
```



Print out data frame of Burnt vs. Non-Burnt mean, min, and max

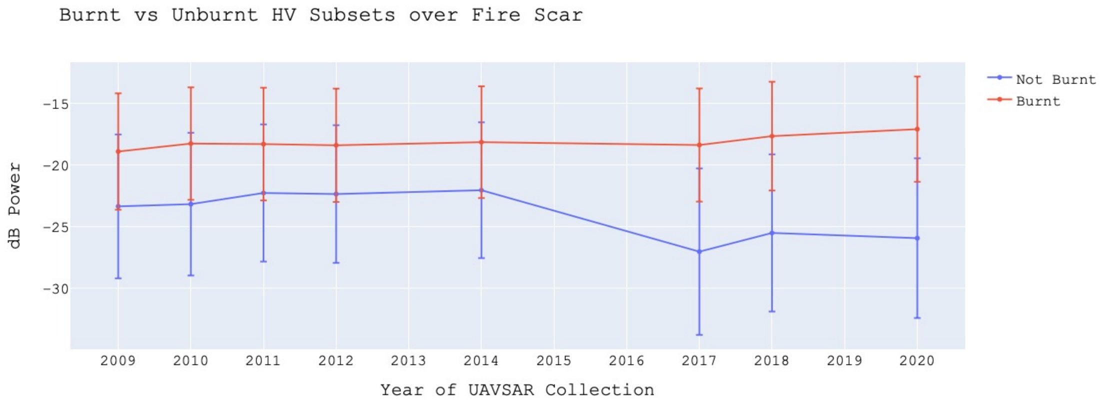
```
[ ] years = pd.Series(years, name="Year")
    pd.to_datetime(years)
    burnt_stats_df = pd.merge(years, burnt_stats_df, right_index=True, left_index=True)
    burnt_stats_df = pd.merge(burnt_stats_df, notburnt_stats_df, right_index=True, left_index=True)

    burnt_stats_df
```

	Year	burnt mean	burnt max	burnt min	not burnt mean	not burnt max	not burnt min
0	2009	-23.355330	-5.932755	-41.715858	-18.902123	-6.153824	-35.490726
1	2010	-23.172589	-6.953382	-41.617771	-18.258309	-5.647788	-35.282635
2	2011	-22.266615	-7.859430	-36.966049	-18.298684	-5.038753	-33.973343
3	2012	-22.353706	-6.629800	-36.821026	-18.395449	-4.563482	-34.874081
4	2014	-22.042698	-4.263805	-37.411942	-18.142646	-5.040717	-34.185871
5	2017	-27.028996	-9.001712	-37.978092	-18.375389	-6.055192	-33.777374
6	2018	-25.509915	-7.198468	-35.078541	-17.653546	-5.203991	-33.696453
7	2020	-25.932999	-7.830916	-36.895172	-17.088209	-5.383451	-31.470165

This time series plots the mean of the burnt and unburnt subsets from UAVSAR collections 2009 - 2020. At each acquisition point, the error bars extend plus and minus 25% of the data's spread.

Time Series Plot



Jupyter Notebook Template Content

SAR Application: {Insert Application}

I. How to create SAR quicklook images of {application}?

- Useful Products:
- Useful Wavelengths:
- Instructions to View:
- Qualitative Browse Products: How does the user know there is something in the image related to flooding before they develop/implement any algorithms?

II. How to interpret SAR images of {application}?

Section 1: Describe general processing and visualization guidelines for the application.

Example Section 1

Introduce a SAR example

In [1]:
```
## To insert regular pictures. Uncomment html below, and add in filename, desired width and align
ment.
## <img src="filename" width="400" align="right"/>
```

In [2]:
```
## Import Packages. For example:
from ipyleaflet import (Map, basemaps, basemap_to_tiles, SplitMapControl, ImageOverlay, ScaleCont
rol,
                        projections, LayersControl, FullScreenControl, LocalTileLayer, LegendCont
rol)
import ipywidgets as widgets
from ipywidgets import AppLayout, Button, Layout, HTML
```

Section 2: Introduce a SAR example and describe how to interpret and display. Here I can include different code cells to fill in depending on desired visualization.
Visualizations could include: interactive maps/graphs, histograms, etc. of different data products

Example Section 2

Elaborate on the example

Section 3: Provide another example or elaborate on another method/visualization

Other Resources

References

Section 4: Provide links to relevant tutorials and resources. List references.

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004)

Example Notebook Layout

SAR Application: Inundation Temporal Dyanimcs

L-Band SAR Flooding and Water State Change

I. How to create SAR quicklook images of flooding?

PolSAR HH and HV (and VV if possible) are useful products for flood area extent. For viewing, Pauli decomposition can be used for quad-pol, or HH/HV and VV/VH for dual-pol images.

NISAR
NASA - ISRO SAR Mission

II. How to interpret SAR images of flooding?

To interpret SAR images of flooding, compare the RGB image before and after the flood. L-band brightness increases over flooded forests and tall vegetation, while short vegetation and open water are dark. Co-pol signal is relatively stronger in inundated areas owing to double-bounce, and Co-pol signal component increases in areas of flood expansion.

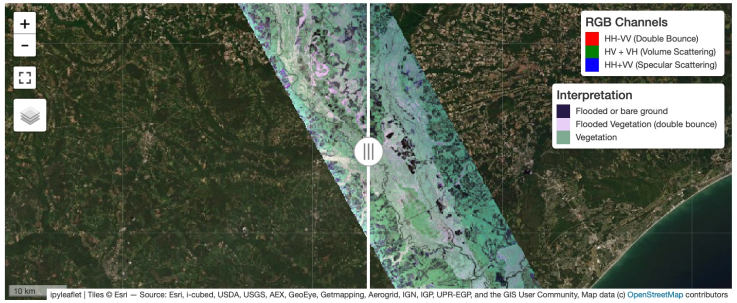
Section 1: Describe general processing and visualization guidelines for the application. Provide links to relevant tutorials and resources.

Hurricane Florence UAVSAR Example

Hurricane Florence was a category 4 hurricane (August 31 - September 18, 2018) that caused significant freshwater and storm surge flooding along the southeastern coast of the Carolinas. From September 17 through September 23, UAVSAR (L-Band airborne radar) flew five times over the Pee Dee River in South Carolina to collect data during and after the hurricane.

This interactive map displays two Pauli RGB UAVSAR images of line peedee_15100, acquired September 17 and September 23. Between these six days, noticeable differences occurred in inundation and open water extent along the river and in nearby fields.

The overlay of the intensities of the different polarization channels, allows users to visually classify a scene by its backscattering mechanism, such as surface scattering (strong HH and VV return), volume scattering (strong HV return) and double-bounce scattering (strong HH return). Thus in RGB images, areas dominated by green (HV) intensity are typically vegetated areas. Areas dominated by shades of pink (HH+HV) intensity are typically inundated forests or vegetated fields. Black and dark grey areas are usually smooth surfaces (roads, open water, smooth bare ground) where there is very little radar backscatter.

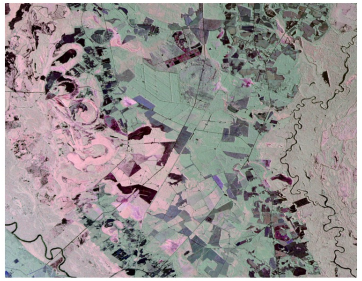


Section 2: Introduce a SAR example and describe how to interpret and display. This shows an interactive map of UAVSAR acquistions during and post Hurricane Florence flooding.

UAVSAR Image Time Series

Changes in flooding are evident in the RGB images, even between adjacent UAVSAR data acquisitions. This slideshow shows a of the image located near the Pee Dee River in an area of changing flood extent along the river and neighboring open fields.

Prev



Next

Section 3: Elaborate on the example. For this flooding example, we show a time series slideshow of the RGB images over a subset of the image.