Data Applications Notebooks with Synthetic Aperture Radar (DANSAR)

Annemarie Peacock, Naiara Pinto, Yunling Lou, and Ekaterina Tymofyeyeva
(Affiliation: Jet Propulsion Laboratory, California Institute of Technology)

Version 1.0 – October 2021

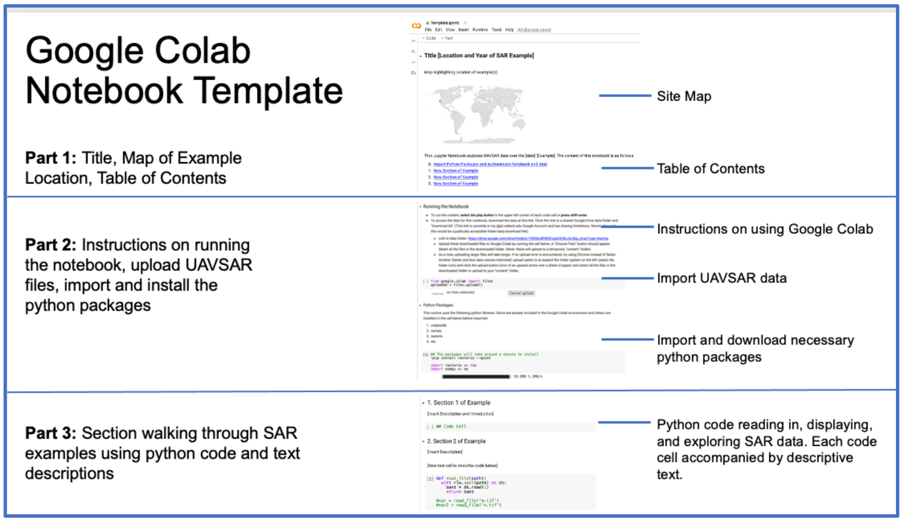
This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the National Aeronautics and Space Administration.

LEARNING OBJECTIVES AND TARGET USERS 1
LANGUAGE AND ROUTINES 1
ACCESSING NOTEBOOKS 2
INPUT DATA AND ORGANIZATION 2
EXAMPLE APPLICATION – FLOOD MAPPING 3
VERSION CONTROL 4

Learning objectives and target users

The Uninhabited Aerial Vehicle Synthetic Aperture Radar (UAVSAR) is a facility instrument suite built and maintained at JPL. The L-band instrument is a testbed for NISAR (NASA ISRO Synthetic Aperture Radar), a spaceborne instrument planned to launch in 2023. UAVSAR observations are operationally processed at JPL and can be downloaded by the public from the Alaska Satellite Facility. More information can be found here: https://uavsar.jpl.nasa.gov/

The last decade has experienced a significant increase in the demand and utilization of SAR imagery for various science applications. The Data Applications Notebooks with SAR (DANSAR) described here use concrete examples to expose the broader remote sensing community to SAR datasets. By analyzing real observations, users should be able to appreciate some of the main characteristics of SAR datasets, its potential uses, as well as limitations. DANSAR Notebooks were designed to overcome two bottlenecks experienced by beginner users. First, notebooks are run in a cloud environment, removing the need for downloading images, setting up files, or installing software. Second, notebooks are divided by discipline allowing users to get specific examples covering problems they are familiar with.


Google Colab Notebook Template
Part 1: Title, Map of Example Location, Table of Contents
Part 2: Instructions on running the notebook, upload UAVSAR files, import and install the python packages
Part 3: Section walking through SAR examples using python code and text descriptions
Site Map
Table of Contents
Instructions on using Google Colab
Import UAVSAR data
Import and download necessary python packages
Python code reading in, displaying, and exploring SAR data. Each code cell accompanied by descriptive text.
Template.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Cancel upload

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:

0. Import Python Packages and Authenticate Notebook and data
1. New Section of Example
2. New Section of Example
3. New Section of Example

Running the Notebook

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

Python Packages:

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
```

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')
```

Figure 1: Notebook organization.

Language and routines

The routines can be implemented in Google Colab: https://colab.research.google.com/notebooks/intro.ipynb?utm_source=scs-index Users can connect and view without the need for a google account or can run the notebooks locally using the repo's environment files. The Google Colab public interface provides access to processing and memory necessary to store and analyze UAVSAR images as well as ancillary files.

We use a python 3.8 kernel and include steps for installing python libraries via pip. The routines use open-source GIS libraries such as numpy, geopandas, and GDAL.

Accessing Notebooks

The public GitHub Repository “DANSAR” hosts the collection of Google Colab Notebooks. To view the Google Colab Notebooks for each SAR application, users may navigate to the Jupyter Notebook (*.ipynb) files in the main branch of the repository. Notebooks are named for their SAR application. At the top of the selected notebook, there will be an "Open in Colab" button such as this image here Open in Colab , which is a link replacing the web page URL "github.com" with the string "colab.research.google.com". Clicking this button at the top of the *.ipynb notebook will launch a new webpage with the Google Colab Notebook. A Google Account login is required to execute and upload data to the notebooks.

Input data and organization

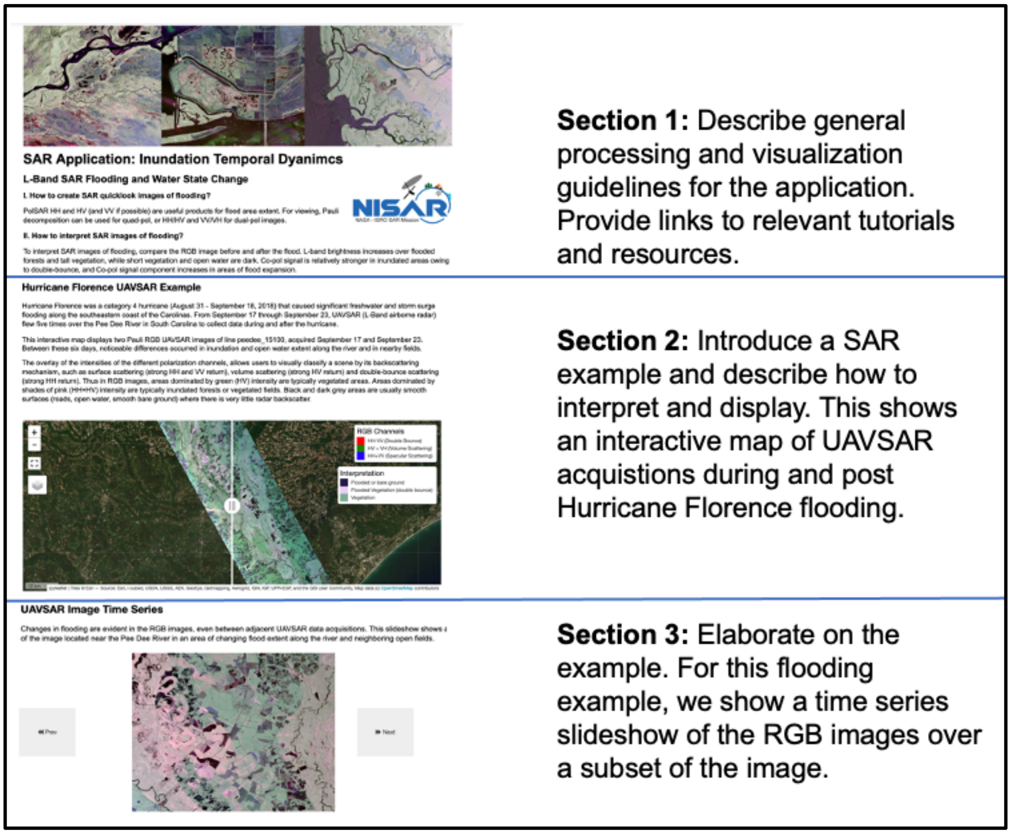
The main inputs for the notebooks will be UAVSAR imagery downloaded from the Alaska Satellite Facility and cropped to reduce file size. We will also consider ancillary datasets to help with interpretation. For example, maps of forest fire scars from the National Interagency Fire Center, basemaps showing administrative units and roads, and maps of known fault lines in California.

These data files are accessible from this JPL website: https://uavsar.jpl.nasa.gov/cgi-bin/sar-notebooks.pl. They can either be imported directly to the Google Colab Notebooks or first downloaded to run locally.

#	SAR Application	UAVSAR Example
1	**RGB**	Creating RGB Composite and Image Stretching
2	**Biomass**	San Gabriel Mountains Station Fire
3	**Crop Classification**	Central Valley-San Joaquin Agriculture
4	**Fire**	Los Angeles La Tuna Fire visualization, Thomas Fire Coherence
5	**Fire Data**	Los Angeles La Tuna Fire
6	**Flood**	Hurricane Florence
7	**Forest Disturbance**	Amazon Gold Mining, Napa Valley Deforestation, Mammoth Tree Die Off
8	**Inundation (Wetland Delineation)**	Balona Wetland, Marsh Island, Fay Slough Wildlife Area, Sierpe Mangroves

9	**Landcover (Urban Areas)**	Munich, Quebec, San Diego, New Orleans
10	**Landslide**	Thomas Fire’s Montecito Debris Flows
11	**Oil Spill**	Gulf Coast, North Sea
12	**Open Water (Surface Water Extent)**	Panama Canal
13	**Sea Ice**	Beaufort Sea
14	**Soil Moisture**	Tonzi Ranch Research Site
15	**Utils**	Folder for notebooks with general utility (i.e. image stretching and composites)

Example application – flood mapping



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.

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.

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_15130, acquired September 17 and September 23. Between these six days, noticeable differences occured 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.

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.

NISAR
NASA - ISRO SAR Mission
RGB Channels
🟥 HH-VV (Double Bounce)
🟩 HV + VH (Volume Scattering)
🟦 HH+VV (Specular Scattering)
Interpretation
Flooded or bare ground
Flooded Vegetation (double bounce)
Vegetation
+
-
Leaflet | Tiles © Esri — Source: Esri, i-cubed, USDA, USGS, AEX, GeoEye, Getmapping, Aerogrid, IGN, IGP, UPR-EGP, and the GIS User Community, Map data (c) OpenStreetMap contributors
◀◀ Prev
▶▶ Next

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

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.

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.

Figure 2: Describing the use of UAVSAR polarimetric images for mapping inundation extent using an example from areas impacted by Hurricane Florence. Many examples use false-color composites or RGBs that combine information from 3 bands.

Version control

Routines are maintained and stored from the public GitHub repository DANSAR: https://github.com/anniepeacock/DANSAR

DANSAR is distributed with an Apache license. Google Colab can pull routines from any public GIT repo and will also be used to update the code and generate new version files. Making changes in the browser is preferrable as it ensures correct setup of the python environment.

URS Clearance Number CL#21-5401