

The DMSAT-1 mission: primary instrument - Polarimeter characteristics and its earth observation applications

Alya AlMaazmi – Remote Sensing Applications Engineer
Mohammed Bin Rashid Space Center (MBRSC)



MOTIVATION

UAE DUSTY ATMOSPHERE

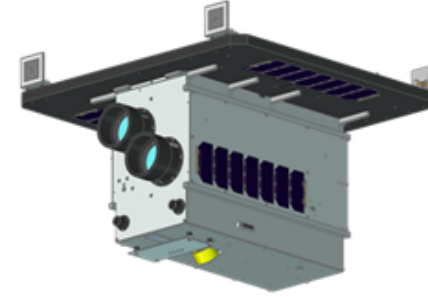
- Dust storms occur frequently in UAE, they most often hit during the summer and times of turbulent weather.
- Sandstorms typically contain silica crystals as well as viruses, bacteria, dust mites, fungi and even plant.
- Can cause vital problems linked to health, economy and environment.
- Space-based monitoring can supplement ground-based weather stations that are currently used for dust storm research.
- Map atmospheric aerosols, including their sources and transport, and study their influence within UAE.

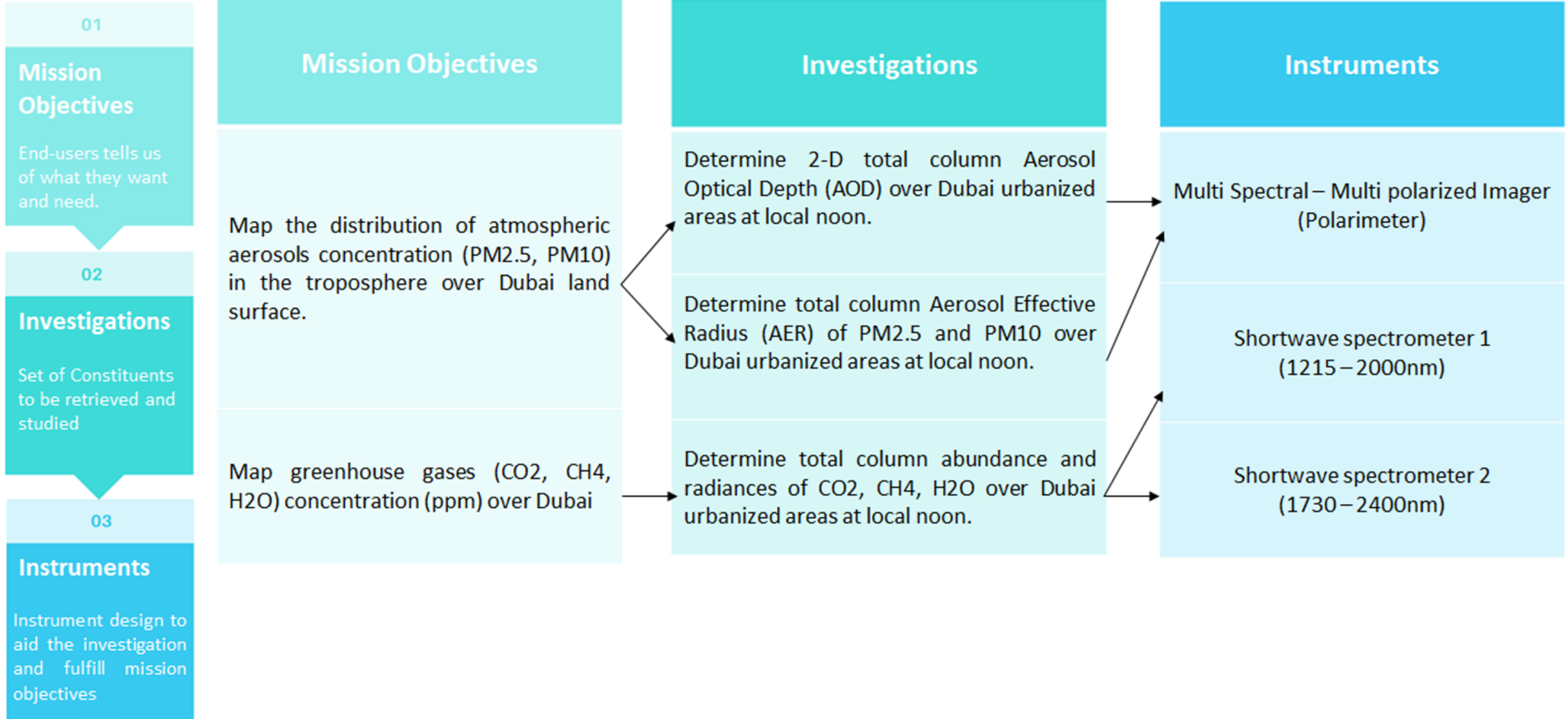


DMSAT-1

Dubai Municipality Satellite

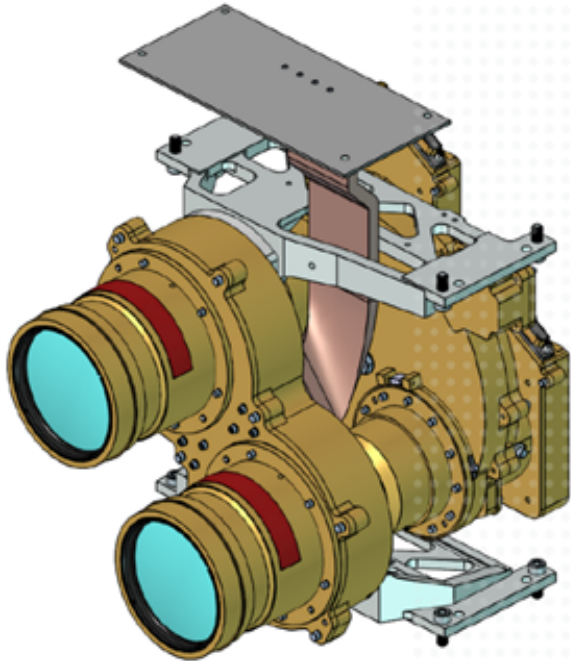
- Is a high-performance small microsatellite designed to perform multi-spectral multi-polarization observations in visual and near-infrared bands, in addition to shortwave spectrum, for aerosol and greenhouse gas monitoring.





Instrument Design

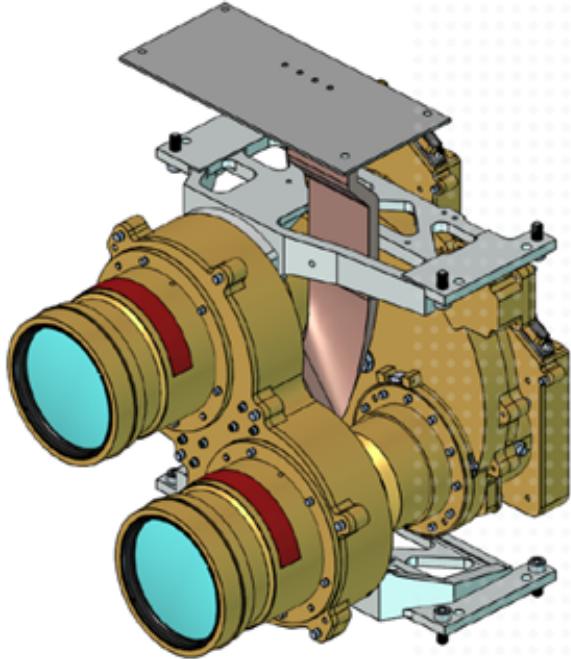
Polarimeter instrument



- High transitivity telescope with a focal length of 150 mm and focal-length-to-aperture ratio of 2.8.
- Filter wheel used to divide the incoming light into three bands: 480-500 nm, 660-680 nm, and 860-880 nm.
- A “p-s” polarizing beam splitter, generates two identical images at two different polarizations 0° and 90°
- Kodak CCD detector at the focal plane.

Instrument Design

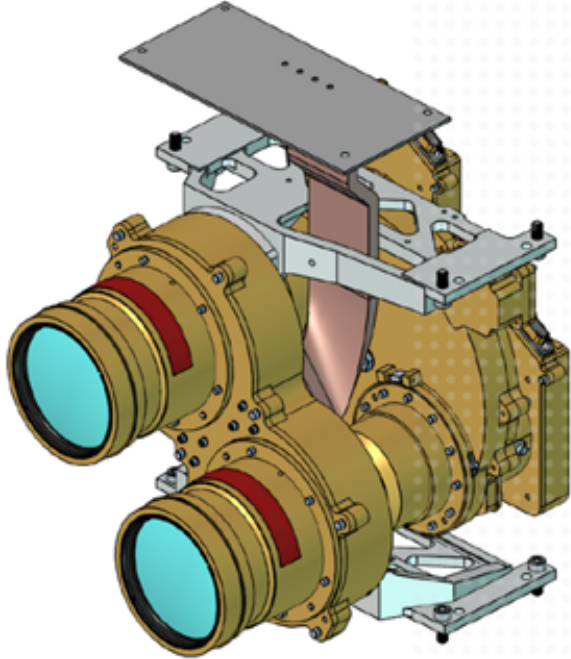
Spectral Bands and Polarization Measurements



- Chosen based on the mission's scientific requirement of retrieving aerosols properties.
- 3 polarized bands with central wavelengths 490, 670 and 870 nm.
- Each band has a polarization of 0° and 90°.
- The DMSAT-1 Polarimeter instrument is designed to measure the linearly polarized Earth-reflected radiance only.

Instrument Design

Geometry and spatial resolution

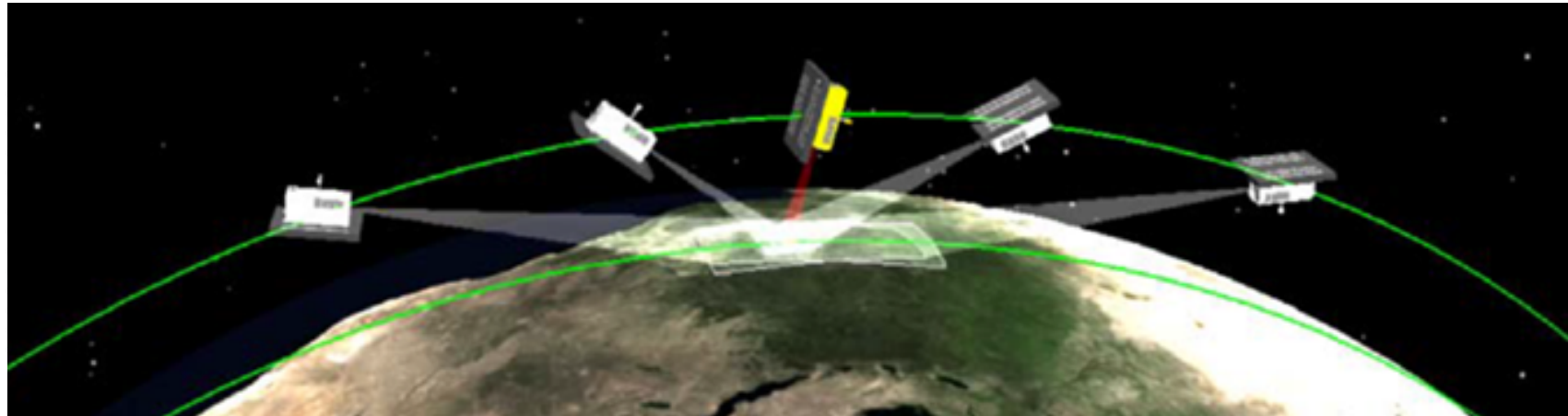


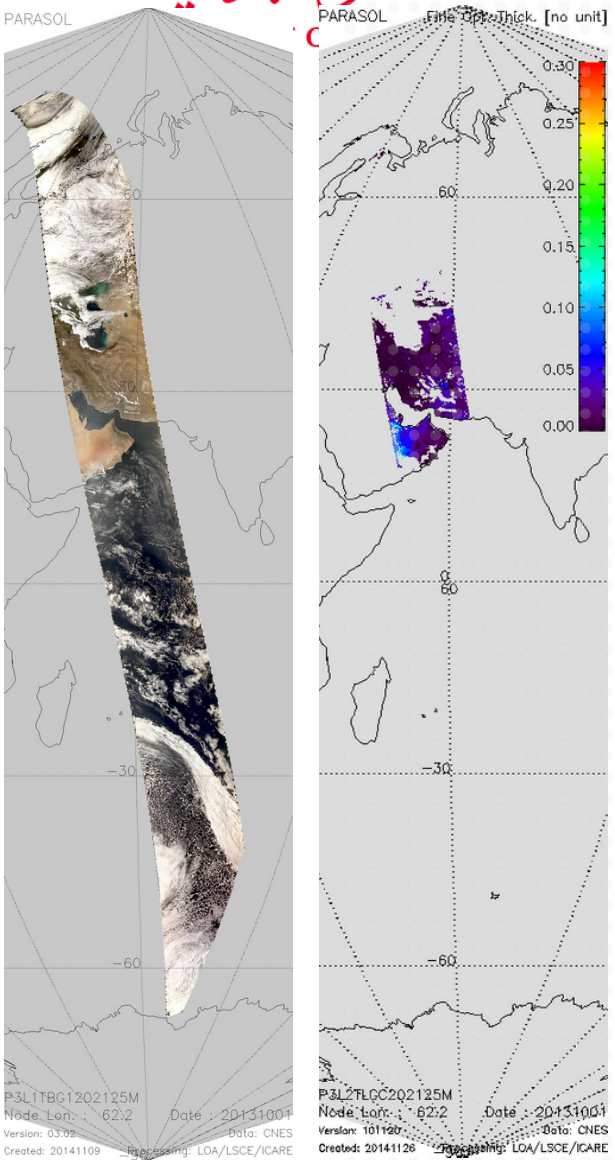
- Spatial resolution of the Polarimeter image is 43.8 m.
- Swath width of 107x38 km.
- Field of View (FOV) is 4.46 deg half diagonal, with an in-track look capability of less than 90° and an off-track look capability of 30°.

Operational Scenario

730 km Altitude, SSO 12:00 LTDN

- Slewing maneuver at 7 different angles.
- Target observation at different reluctance and scattering angles.





Polarimeter Applications

Primary Polarimeter Instrument Applications

- Aerosol Optical Depth.
- Aerosol effective radius.
- Aerosol type PM 2.5.
- Aerosol type PM 10.

Secondary Polarimeter Instrument Applications

- Surface ALBEDO.
- Normalized Difference Vegetation Index (NDVI).
- Aerosol mass mixing ratio.

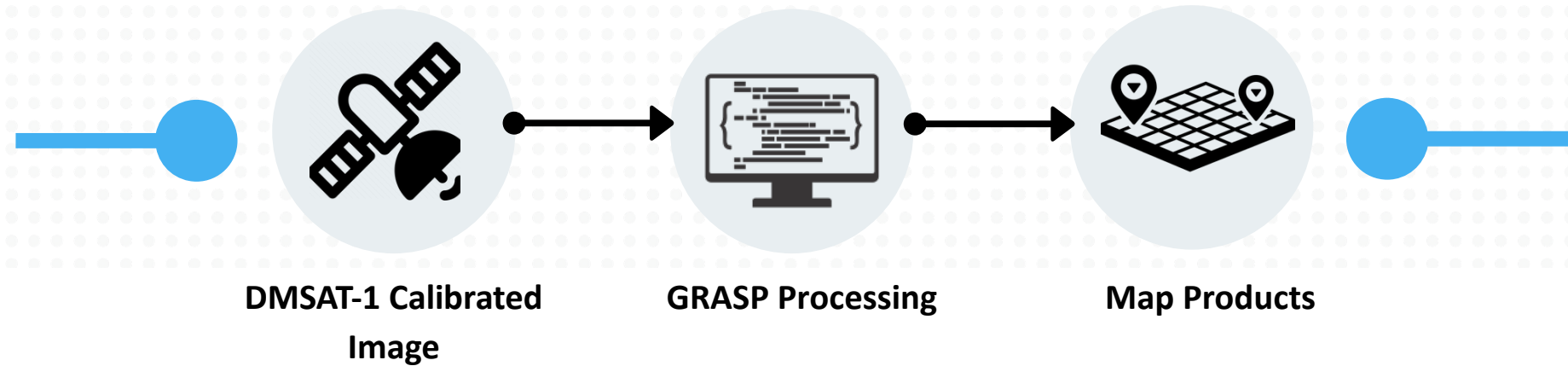
Polarimeter Applications

GRASP Algorithm for DMSAT-1 retrievals



GRASP
OPEN

 GRASP



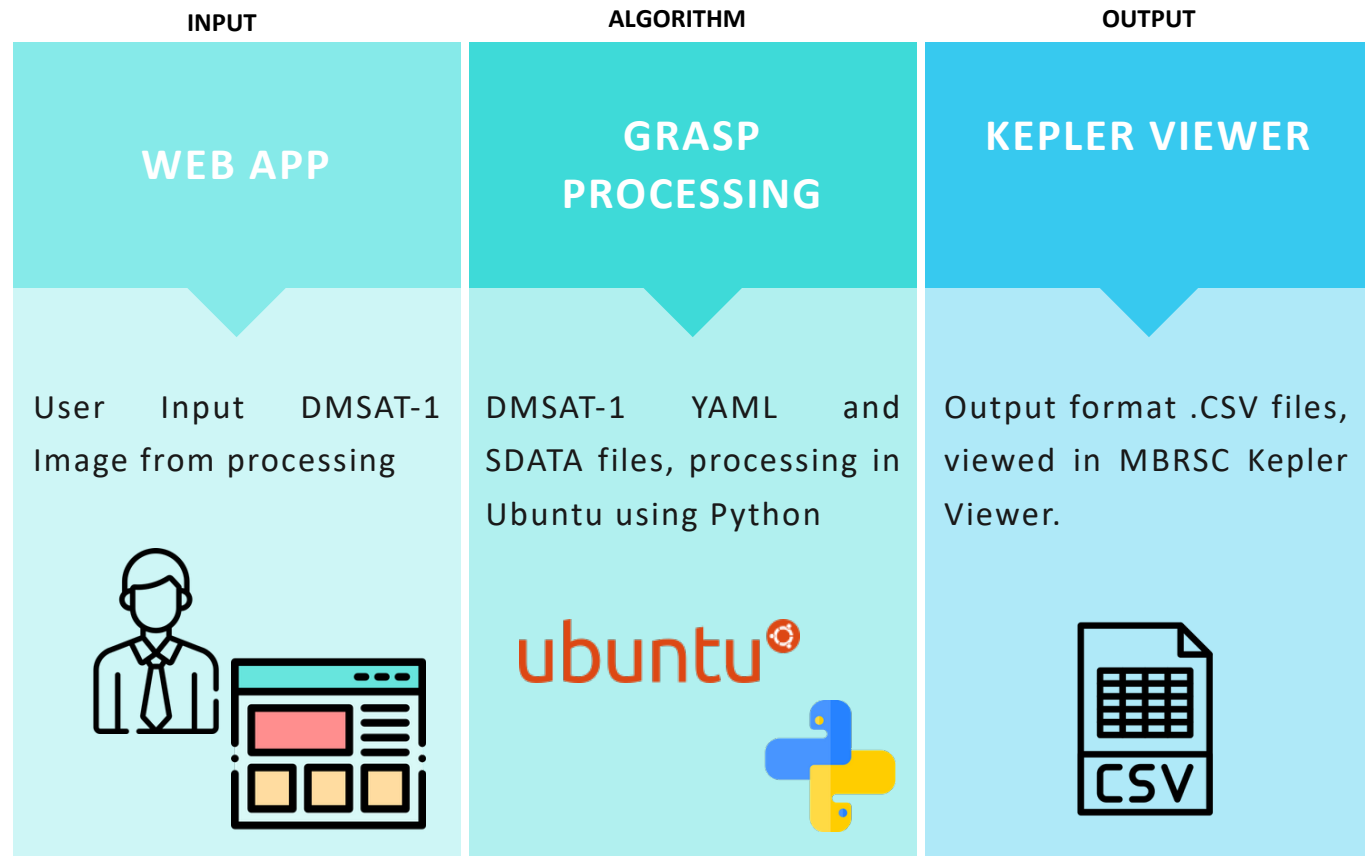
Polarimeter Applications

GRASP Algorithm for DMSAT-1 retrievals



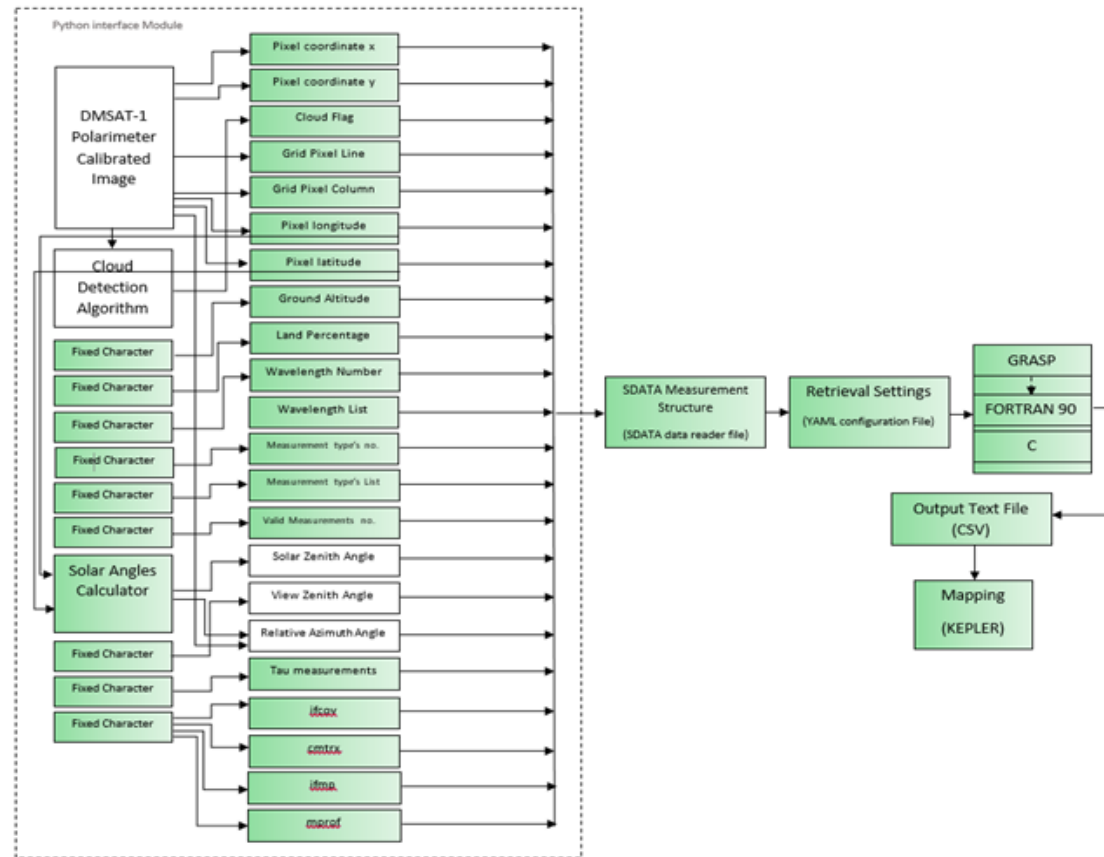
GRASP
OPEN

 GRASP



Polarimeter Applications

GRASP Algorithm for DMSAT-1 retrievals





GRASP
OPEN



Polarimeter Applications

GRASP Algorithm for DMSAT-1 retrievals

The screenshot displays a Jupyter Notebook interface with two main sections: a code editor on the left and an output area on the right.

Code Editor (Left): Shows Python code for the GRASP algorithm. Key lines include:


```

import sys
sys.path += ["grasp-dir/src/python"]
import grasp_code as pygrasp
from PySolar import solar # PySolar not pysolar, use solar as function
import datetime

print(pygrasp.version()) # to test if the interface works correctly

date = datetime.datetime(2013,1,2,11,8) #
S_Altitude = solar.GetAltitude(24.416666, 54.640678, date)
S_Azimuth = solar.GetAzimuth(24.416666, 54.640678, date)
S_Zenith = 90 - S_Altitude
T = print(S_Zenith, S_Azimuth, S_Altitude)

results = pygrasp.run("examples/DMSAT/settings_example_DMSAT_inversion.yml",
sdata = pygrasp.SDATA("examples/DMSAT/DMSAT_MASDAR_4pixels.sdat")

# colors = ["red", "green", "blue", "purple"]
# # = 0
# while i < len(colors):
#     print(colors[i])
#     i += 1

# coordinate x in the current cell, starting at 1 (in the direction EW)
# coordinate y in the current cell, starting at 1 (in the direction NS)
# cloud flag: 0 = cloud, 1 = clear (a)
# line of the pixel in its original grid or database
# column of the pixel in its original grid or database
# longitude of the pixel, in decimal degrees, in the range [-180, 180]
# latitude
# solar zenith angle in decimal degr
# solar zenith angle in decimal degr
# solar zenith angle in decimal degr

# colors = ["red", "green", "blue", "purple"]
# # = 0
# while i < len(colors):
#     print(colors[i])
#     i += 1

# pixel_number = np.asarray([0,1,2,3])
# lx_x = np.asarray([1,1,2,2])
# ly_y = np.asarray([1,2,1,2])
# lrow_row = np.asarray([1,1,2,2])
# lcol_col = np.asarray([1,2,1,2])
# sx_x = np.asarray([54.64067840576172, 54.64067840576172, 53.64067840576172, 53.64067840576172])
# sy_y = np.asarray([24.416666, 23.416666, 24.416666, 23.416666])

# # = 0
# while i < len(pixel_number):
    
```

Output Area (Right): Shows the output of the code execution, including a table of variables and their values:

Name	Type	Size	Value
lx_x	Int64	(4,)	[1 1 2 2]
ly_y	Int64	(4,)	[1 2 1 2]
l	Int	1	1
pixel_number	Int64	(4,)	[0 1 2 3]
s_x_x	Float64	(4,)	[54.64067841 54.64067841 53.64067841 53.64067841]
s_x2	Float64	(4,)	[54.64067841 54.64067841 53.64067841 53.64067841]
s_y_y	Float64	(4,)	[24.416666 23.416666 24.416666 23.416666]

The bottom part of the output area shows the execution of a cell with the following code:


```

In [*]: # -*- coding: utf-8 -*-
# -*-
Spyder Editor
This is a temporary script file.
# dont forget to work at grasp directory ( change the path to grasp directory)
import sys
from PySolar import solar # PySolar not pysolar, use solar as function
import datetime
sys.path = ["grasp-dir/src/python"]
import grasp_code as pygrasp

# solar angles calculator
date = datetime.datetime(2017,1,30,8,9) # year, month, day, hour, minute # UTC time BAH +12 PM
#S_Altitude = solar.GetAltitude(24.416666, 54.640678, date)
#S_Azimuth = solar.GetAzimuth(24.416666, 54.640678, date)
#S_Zenith = 90 - S_Altitude

print(pygrasp.version()) # to test if the interface works correctly
results = pygrasp.run("examples/DMSAT/settings_example_DMSAT_inversion.yml", print_screen=True)
# access SDATA parameters
# changing variables
sdata = pygrasp.SDATA("examples/DMSAT/DMSAT_MASDAR_4pixels.sdat")
# sdata.pixel[0]
# coordinate x in the current cell, starting at 1 (in the direction EW)
# coordinate y in the current cell, starting at 1 (in the direction NS)
# cloud flag: 0 = cloud, 1 = clear (a)
# line of the pixel in its original grid or database
# column of the pixel in its original grid or database
# longitude of the pixel, in decimal degrees, in the range [-180, 180]
# latitude of the pixel, in decimal degrees, in the range [-90, 90]

sdata.dump("sdata_save2.sdat")

# fixed variables
# altitude of the ground, in metres (MSL: metres above sea level)
# percentage of land, in the range [0 (sea) .. 100 (land)]. Intermediate values correspond
# number of available wavelengths (nwl)
# wavelength 1, in micrometers
# wavelength 2, in micrometers
# wavelength 3, in micrometers
# number of types of measurements for wavelength 1 (nlp)
# number of types of measurements for wavelength 1 (nlp)
# number of types of measurements for wavelength 1 (nlp)
    
```

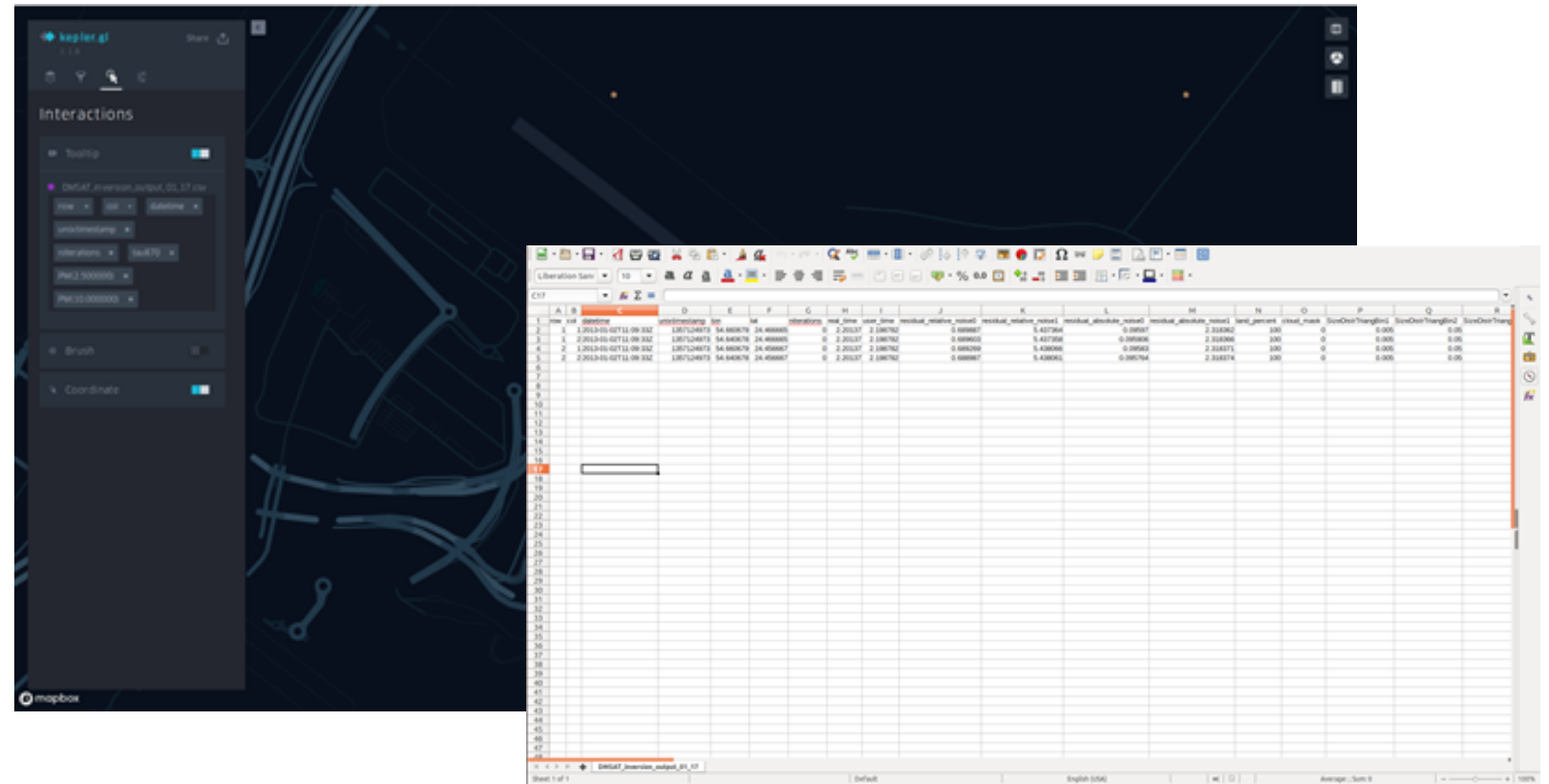
Polarimeter Applications

GRASP Algorithm for DMSAT-1 retrievals



GRASP
OPEN

GRASP



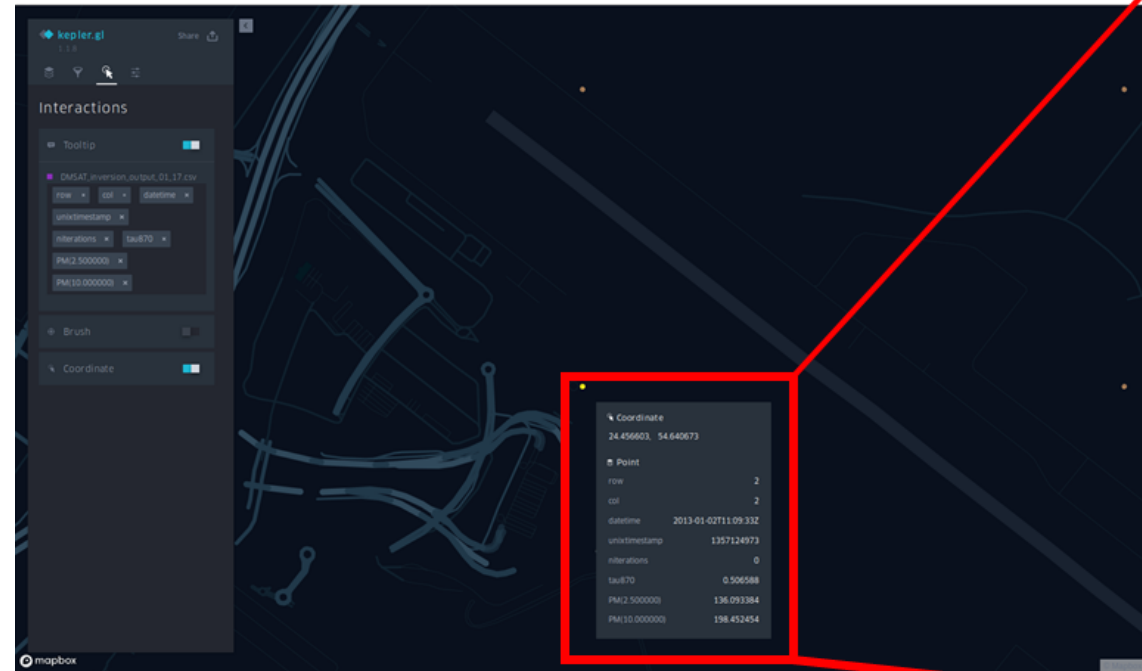
Polarimeter Applications

GRASP Algorithm for DMSAT-1 retrievals



GRASP
OPEN

GRASP



Coordinate	
24.456603, 54.640673	
Point	
row	2
col	2
datetime	2013-01-02T11:09:33Z
unixtimestamp	1357124973
iterations	0
tau870	0.506588
PM(2.500000)	136.093384
PM(10.000000)	198.452454

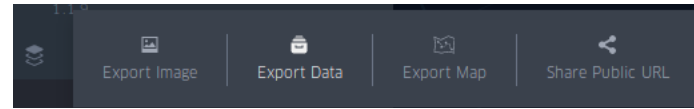
Polarimeter Applications

GRASP Algorithm for DMSAT-1 retrievals



GRASP
OPEN

 GRASP



Export Data

Dataset: Choose the datasets you want to export.

Data Type: Choose the type of data you want to export.

Filter Data: You can choose exporting original data or filtered data.

Cancel

Export

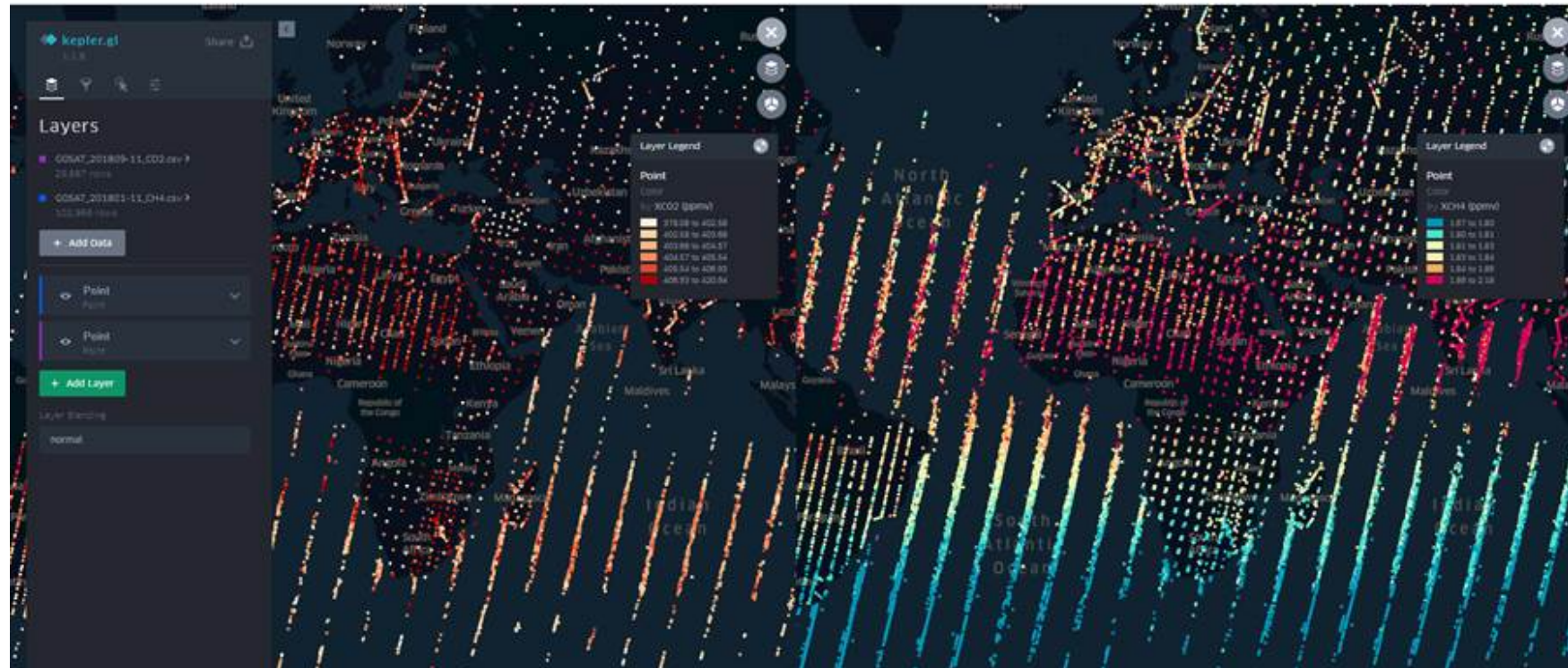
Polarimeter Applications

GRASP Algorithm for DMSAT-1 retrievals



GRASP
OPEN

 GRASP



Thank You

alya.almaazmi@mbrsc.ae