

Prepared by Nicholas Tuffaro

NASA grant NNX17AG05G

Oregon State University

22 June 2020, Version 1.0

# Processing MOS-P and GeoTASO for Ocean Products from NASA KORUS-OC

Abstract:

Table of Contents

1. Introduction
2. Quick Start
3. Operational Processing Methods
4. Open Tasks
5. Concluding Comments
6. References

Appendices

- A. Data Set Summary Tables
- B. Data Set Map Overviews
- C. User Manual

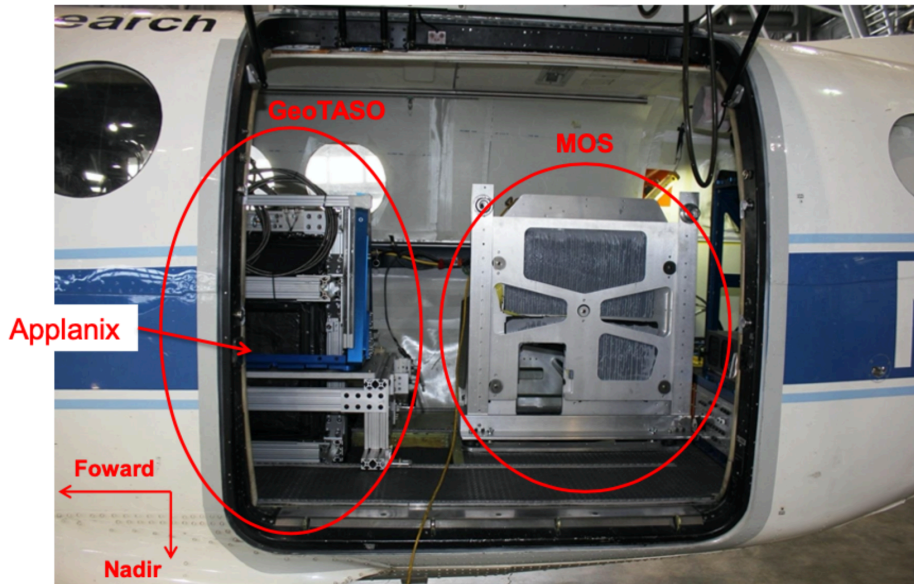


Figure 1: MOS-P and GeoTASO Spectrometers

## 1. Introduction

The KORUS-OC campaign (Korea-United States Ocean Color) was a joint NASA and KIOST (Korean Institute of Ocean Science and Technology) 18-day field campaign staged during May and June 2016 to characterize the daily changes of the seas surrounding South Korea [1]. In addition to extensive ship board measurements, NASA also undertook coordinated air campaigns that included two NASA ESTO - IIP instruments (Earth Science Technology Office - Instrument Incubator Program): MOS-P (Multi-Slit Optimized Spectrometer with Polarization [2]), and GeoTASO (Geostationary Trace gas and Aerosol Sensor Optimization [3]). MOS-P is designed for ocean color measurements and is a hyperspectral imager covering the visible spectrum that also captures simultaneous measurements of linear polarization. GeoTASO is primarily a hyperspectral atmospheric imager that covers the UV and visible spectrum, which can be also be used for ocean color retrieval's. Both instruments where flown on a King-Air B200 between 29th April 2016 till 10th June 2016, with the coordinated ship board ocean measurements taking place primarily between 20th May 2016 till 5th June 2016. Figure 1 shows a photograph of the instruments on the aircraft, and Table 1 provides a brief summary of typical instrument and flight characteristics, processing level conventions, and resolutions.

A total of 30 different flights where undertaken with the instruments during KORUS. The primary coordinated ocean color measurements total 13 flights. Level-1 calibrated radiance data

(L1b) were computed by Ball Aerospace (MOS-P) and NASA (GeoTASO). The L1b data consists of ~20 Terabytes (TB) and was provided to Oregon State University (OSU) for processing to Level-2 atmospherically corrected ocean color products including Remote sensing reflectance (Rrs), pigments (e.g. Chlorophyll-a), and Intrinsic Optical Properties (IOP's). The MOS-P data set is particularly interesting since it also includes linear polarization parameters which are not typically available for ocean measurements and opens up new applications ranging from novel water surface reflection corrections (Fresnel reflection correction [4]) to new multivariate ocean color retrieval's. All the data was processed for surface reflectance (over both land and water), and suitable water scenes were processed for ocean color. About half the data from all the flights is over water targets, and of those less than half have a stable flight pattern where the water scene is not obscured by atmospheric conditions (primarily cloud cover). Unrestricted L2 data will be made available by NASA.

Table 1: Flight and Ocean Color Product Summary Table

Sensor parameters			Instrument and aircraft parameters			
	MOS+P	GeoTASO				
Wavelengths (nm)	410-890	290-695	Aircraft Speed: 100 - 120 m/s			
Spectral sampling (nm)	2-10 (variable over range)	0.4 (UV); 0.8 nm (VIS)	Altitude: 7500 - 9000 m			
Swath width (km)	3.7	7	Instantaneous Field of View:			
Polarization	Open, 0°, 90°, 135°	Scrambled - time domain modulation	MOS+P: 0.013 deg			
Spatial resolution (m)	45	80	GeoTASO: 0.044 deg			

KORUS-OC Data Levels							
	L0 native		L1 binned		L2 binned		
	MOS-P	GeoTASO	MOS-P (Ball)	GeoTASO (NASA)	MOS-P (OSU)	GeoTASO (OSU)	
spectra #z (nm)	1344 (330-925)	1056 (410-699): ~0.28	256 (330-925: ~1-8	1056 (410-699): ~0.28	128 (350-900: ~5)	64 (420-690: ~5)	approximately square image
cross-track #x (m)	2016 (~2)	1033 (~7)	2016 (~2)	33 (~250)	100 (~50)	33 (~250)	
along track #y (m)	1 (~50)	1 (~25)	100 (~50)	# (~50)	100 (~50)	33 (~250)	
t (s)	1/4	1/3	~2 1/2	~3 1/3	~2 1/2	~3 1/3	

Figure 1: MOS-P and GeoTASO Spectrometers

## 2. Quick Start

To aid researchers in using the data, this report provides a 'quick start guide' for navigating the data. Appendices A and B provide further information on what data was suitable for ocean color processing, and provides additional details to help researchers identify scenes of interest. It

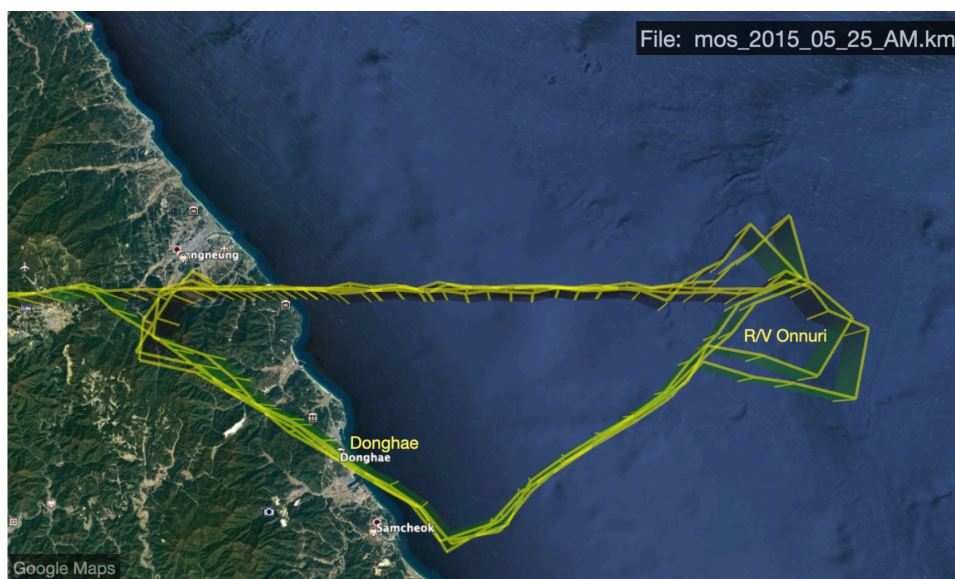


Figure 2: Example of KLM file containing flight Paths.

is expected that a typical researcher will only be interested in a handful of scenes leading to data transfers of a few gigabytes.

The data is organized in folders by 'flights' which have a 'day' associated with them, and a 'time-of-day' (eg. AM, PM, or MD - Mid-Day). Each spectra is collected in a line-scan, and these line-scans are bundled together in 'frames,' which are typically a collection of 100 line scans (see L2 binned column in Table 1 under MOS-P). To find a date and place of interest, first look at Appendix A to find the 'name' of the folder containing the flight data of possible interest, and then look at Appendix B to see the flight map to help identify the overflight regions. To find a specific 'data-cube' next identify the 'frame' number when the imager is over your target of interest. The frame number points to the specific data-cube containing the spectra (z) for pixels in the along-track (y) and cross-track (x), and in the case of MOS-P there is an additional dimension in the data cube specifying the polarization, indicated by the 'slit number' (s).

After locating the flight of interest there are a couple of things to check before grabbing a particular frame containing spectra for a specific scene. The spectra might have poor quality data or no data at all due to a number of reasons including: (1) there is cloud cover blocking the scene, (2) the flight is not 'even and steady' (when the aircraft is turning the 'pointing angle' is greatly distorted and not facing downward toward the target), (3) the imager doors might be closed which is typical for take off and landing before the aircraft reaches altitude, (4) the instrument operator could be resetting the instrument parameters (typically the exposure time is reset as the



aircraft goes from land to water to allow the imager to better capture a 'dark target, (5) high glint due to sun and view angle, (6) and so on.

To help check that the spectra wanted is going to be useful, movies, kml files, and google earth images, are available for each flight which help the researcher both identify the desired frame number, as well as checking the quality of the data. The highest 'resolution' (~50m pixel size) sensor is MOS-P, so RGB movies using the MOS-P sensor were produced and are available for viewing at the URL:

<http://aqua.homeunix.net/korus-oc/>

The images in the movies are of sufficient spatial resolution to identify many targets of interest (rivers, islands, larger ships, ports) for the researcher to confirm the target is viewable at the time and place specified. The movies also provide a quick indication of the quality of the data in terms of its view angle, atmospheric conditions, flight turbulence, or other obfuscations.

As an example, let's find when the imagers are above the R/V Onnuri on the 25th of May 2016. Shipboard records indicate a ship station at 'K22' at the digital lat/lon [37.7711, 129.6670] in the AM flight, which is above non-Territorial waters. Note that the time in a files name uses UTC, and the local time (KST) is UTC+9. For convenience, the folders containing flight data are indicated by local time, so the folder title should include the local date stamp '2016\_05\_25' (KST is UTC+9 so the UTC day is the 24th). Looking in Appendix A shows that the flight folder is '2016\_05\_25\_AM', and looking in Appendix B indicates the flight path covers multiple passes over the R/V Onnuri's stationed in the East Sea, off the coast of Donghae. Figure 2 shows a

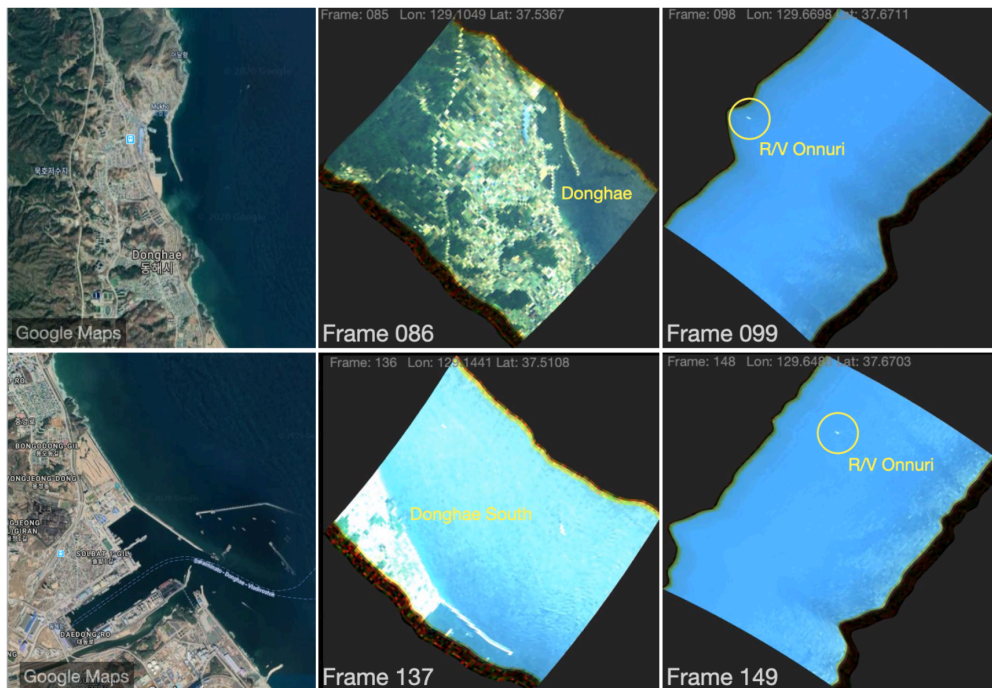


Figure 3: Image frames from 'MOS-P' movies used to identify scenes of interest.

google maps keynote markup language (kml) plot of the flight path, and Figure 3 shows frames from the movie. The flight path indicates multiple overpasses above the R/V Onnuri. To find the location of the R/V Onnuri in the MOS-P movies, first identify a frame with a land mark (the city of Donghae in Figure 3), and then view the movie frames until you are in the vicinity of the ship (the digital lat/lon of the center pixel of each frame is show in the movie at the top left). After spotting the ship, note the frame number, and retrieve that frame from the data folder. The first two overpasses identified in Figure 3 occur in frames 099 and 149.

The metadata for this frame is contained in the HDF File within the 'GeolocationFields' folder (Fig. 4). In particular UTC Time is the variable 'Time.UTC' and Altitude is the variable 'AircraftAltitude.' For frame 149 the UTC Time is 01:11 (10:11 AM KST), Altitude is 7579 m, and the center Lat/Lon is (37.671, 129.667).

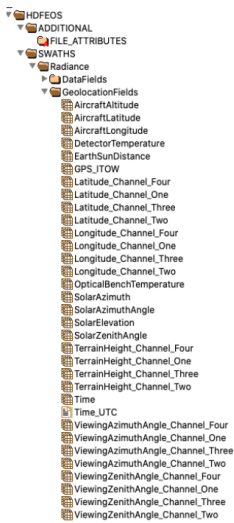


Figure 4: HDF File Example

Plotting the Top of Sensor (TOA) spectra for the center pixel results in the Figure 5, with recognizable features such as the O<sub>2</sub> absorption band at 760 nm. For L2 processing the spectra are resampled to a 5 nm grid between 400-900 nm. All scenes are operationally processed for atmospheric correction resulting in a Remote sensing reflectance (R<sub>rs</sub>). Atmospheric correction is done with Tafkaa - the ocean color atmospheric correction program created by the Naval Research Labs [5] that uses look up tables to

quickly estimate the path radiance between an airborne imager and the surface target. Initial operational products for R<sub>rs</sub> are generated with fixed aerosol (maritime), gas (NO<sub>2</sub> ~ 10<sup>15</sup> VCD), and water vapor parameters. Selected high value scenes are carefully processed by iterating through aerosol models and parameters such as water vapor to identify the most consistent products. Two HDF files are constructed for each frame: L2A which contains R<sub>rs</sub>, and L2B for water product maps (Chl-a, IOP's, ...). Additionally, the binned top-of-sensor data is contained in an L1c HDF container.

Plots of the L1A R<sub>rs</sub> spectra are shown in Fig. 6. The 'open slit' (slit-1), as expected has the largest signal, and the 'horizontally polarized' (slit-2) has the lowest. The polarized spectra are significantly lower in the 'blue' part, which is presumably due to filtering of (and possible overcorrection) for atmospheric scatter reaching the sensor in the channels using a polarizer.

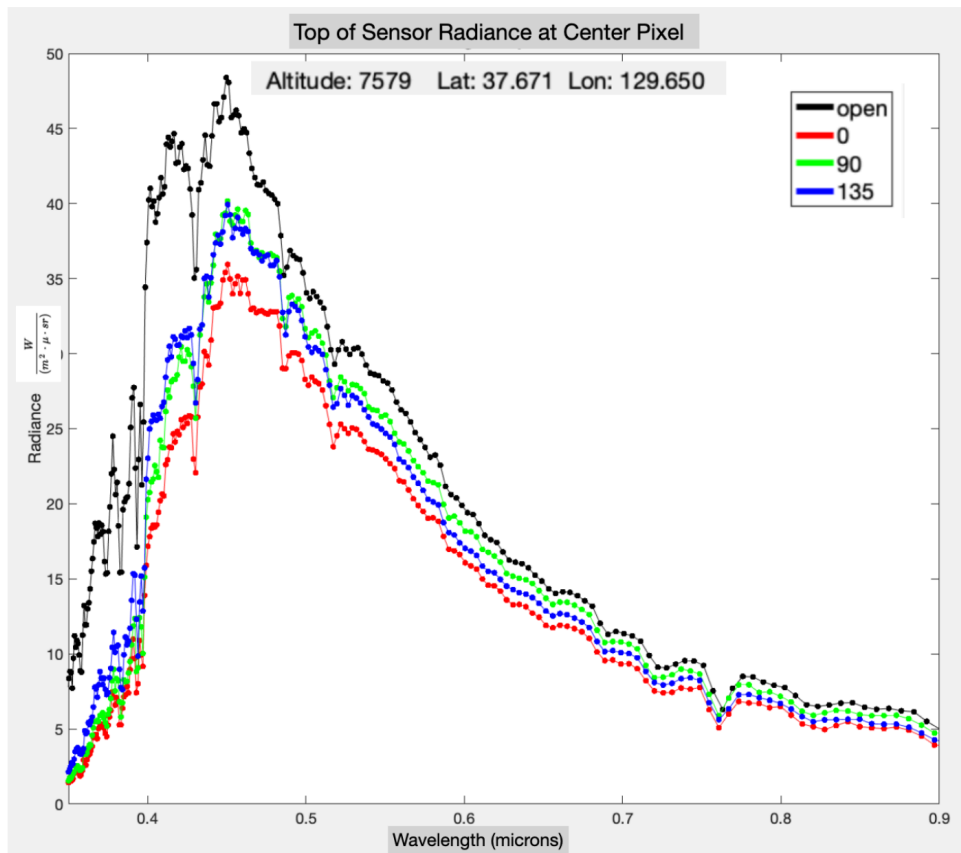


Figure 5: At sensor Radiance Spectra.

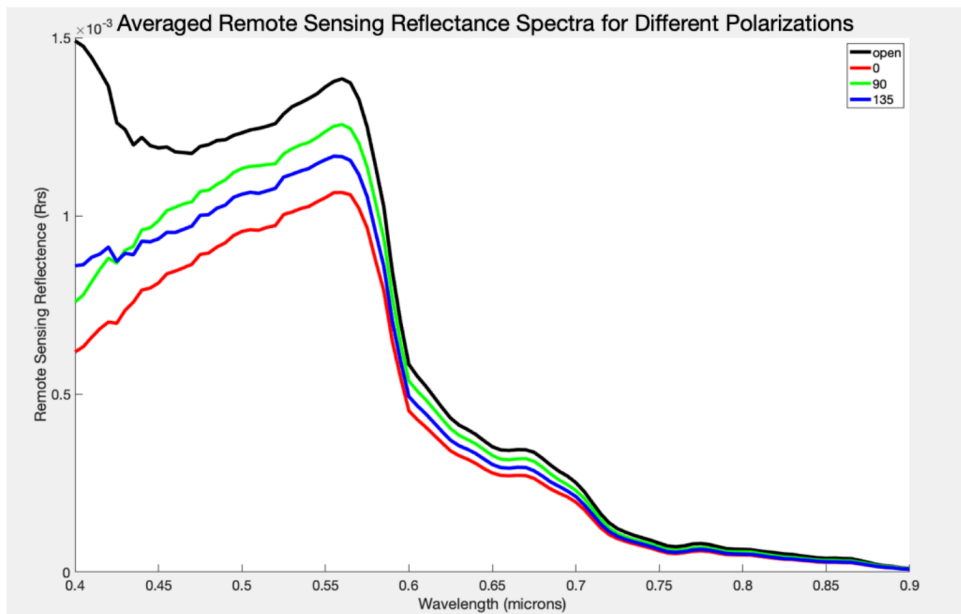


Figure 6: Examples of L2A Remote sensing reflectance spectra for each polarizer.

However, the linear polarizers also presumably removes much of the water glint, especially slit 2 (oriented horizontally) which can be a useful feature.

Examples of spectra from the 'open slit' are shown in Figure 7. A sampling of the spectra are collected in a line along the right side of the scene as illustrated in the inset image of Figure

7. The median value of the line of spectra is shown by the solid black line. The left side of the image appears to be contaminated with surface glint which is difficult to remove. Tafkaa takes into account sensor view angles and solar positions to estimate surface reflectance (Fresnel Correction [4]), but the glint is a more sticky problem, and glint removal from direct sunlight reflections from the water surface are not corrected for in the current L2 processing.

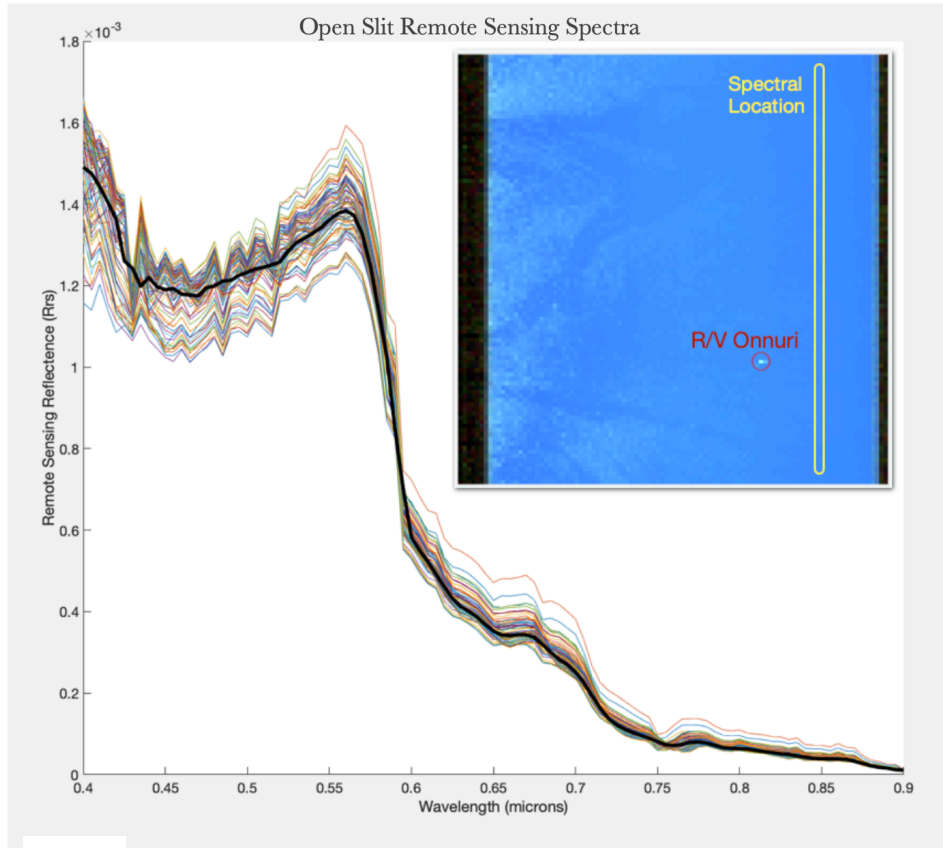


Figure 7: L2A Remote sensing reflectance spectra, variation along yellow strip shown in inset.

The L2B data sets contain a standard chlorophyll product (oc3v [6]) computed from the open slit. This is shown in Figure 8, which indicates a well mixed chlorophyll concentration between 2-3 mg/L over the  $\sim 5$ km square scene. However, we suspect the glint is also contaminating pixels on the left side of the image. To explore this, we also created a Fluorescence Line Height (FLH) map of the scene built on data from the horizontally polarized image (slit-2, red spectra in Figure 6) [7]. A close examination of the spectra appears to indicate a small rise around 675 nm which can indicate chlorophyll fluorescence. Since we have hyperspectral data, we choose the bands for the line height to be as close as practical, in this case they are chosen at 650, 675, and 700 nm. The resulting FLH product is shown in Figure 9, which does appear to

effectively filter the sea surface glint, and provide what we presume is a more accurate (relative) concentration field for the chlorophyll. Indeed, there is a coincident GOCI image and which also shows a similar distribution of chlorophyll in this region as illustrated in Figure 10.

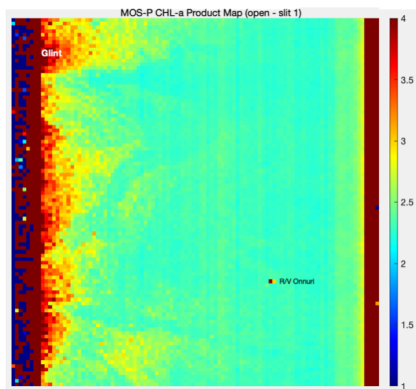


Figure 8: L2B Chl-a Product Map

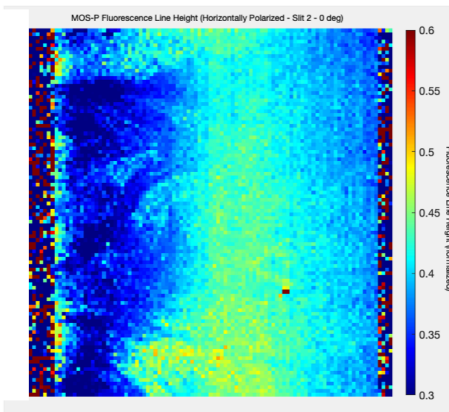


Fig 9: L2b FLH Product Map

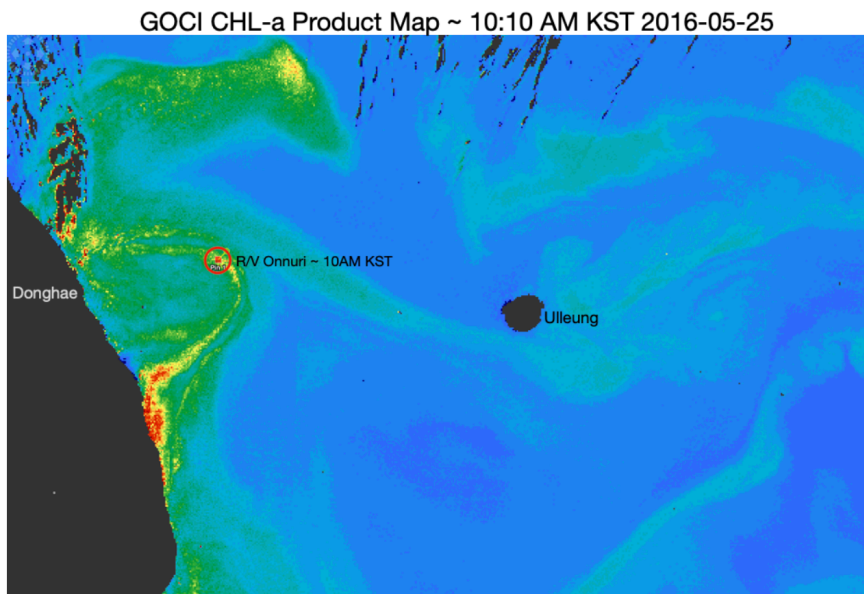


Figure 10: Coincident GOCI CHL-a Product Map

GeoTASO data is also available for this flight. GeoTASO data (UV & VIS) can be used for estimates of trace gases (e.g. NO<sub>2</sub>) and water vapor for retrievals. Results of joint retrievals will be reported in a separate paper. However, there is also interest in using visible GeoTASO spectra for ocean color retrievals directly. To do this, we first bin the sub-nanometer GeoTASO spectral data to ~ 5 nm (Fig. 11) between 400-690 nm, and also spatially bin to ~ 250 x 250 m. This



allows retrieval's of ocean color products with a cross track size of  $\sim 8$  km (Figure 12). Unlike the MOS-P data, the GeoTASO data for an entire flight is collected in one HDF folder for the L2A

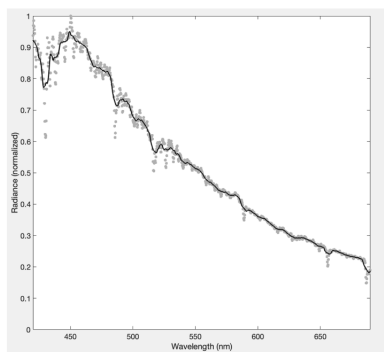


Fig 11. Example of resampled (5 nm) sensors radiance for GeoTASO. The full resolution spectra can be used for atmospheric parameter retrieval's.

and L2B product maps. To coordinate between the two data sets, both the latitude and longitude, and time stamps, can be used to collocated the two image streams.

One final note, some of the flight data sets are broken into segments which are indexed by a capital letter in the folder names (ex. 2015\_05\_25\_AM\_A, 2015\_05\_25\_AM\_B), this is because of instrument resets or adjustments mid-flight. The frame numbering counter is reset to 1 for each of these segments.

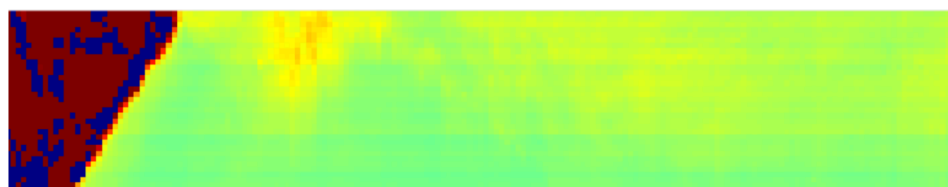


Fig 12. Chlorophyll retrieval (OCv) using GeoTASO for the AM flight on 2016-05-25. Donghae is on the left of the image. The cross track size is  $\sim 8$  km and the pixel size  $\sim 250 \times 250$  m.

### 3. Operational Processing Methods

All data were processed from L1b (calibrated at sensor radiances), and next to binned and resample spectral data cubes (L1c), then onto a ground/water reflectance (L2a), and finally to selected water products (L2b). Water products are only generated for frames containing coastal waters, and the land reflectance is really a water Rrs product, so it is not a standard land ground reflectance.

As mentioned, the water remote sensing reflectance products are generated with NRL's Tafkaa program [5]. In particular, the 'Tafkaa tabular' program was used which includes corrections for sea surface atmospheric reflections (sometimes called 'sky glint' [8]), but not sun glint. MATLAB codes are used to automatically extract, bin, resample, and create the relevant radiance data, along with the meta-data (byte-order, sensor altitude, location, etc) into ENVI standard file formats (\*.bsq, \*.hdr). An example ENVI header file looks like:

```
description    = {MOS-P KORUS-OC}
file type     = ENVI Standard
```

```

interleave      = bsq
sensor type     = MOS
samples        = 100
lines          = 100
bands          = 109
header offset   = 0
data type      = 2
byte order     = 0

image_center_date      = {2016, 05, 25}
image_center_time     = {01, 12, 30}
image_center_zenith_ang = { 6.4137664, 0., 0.}
image_center_azimuth_ang = {202.22554, 0., 0.}
image_center_lat      = { 37.671995, 0., 0.}
image_center_long     = {129.64693, 0., 0.}
sensor_altitude      = 7.578
tafkaa_ground_elevation = 0.0
image_center_lat_hem  = N
image_center_long_hem = E

wave length units = microns
wavelength        = {
... wavelength data
}
fwhm = {
... full width half max data
}

image scale factor = {500.0}
default bands = {87, 73, 49}
z plot titles = {Wavelength (micron), Scaled Radiance}

```

A Tafkaa processing control file is also generated containing estimates of aerosol, atmospheric gas, and other parameters as this example shows:

```

tafkaa_input_image_name = tafkaa_in/korus_2016_05_25_am_rad1.bsq
; Image parameters

; Environmental parameters
tafkaa_ground_elevation = 0.000000

; Gaseous (absorption feature) parameters
tafkaa_atmo_model = mid latitude summer
tafkaa_atmo_gasses = {O2, O3, NO2, H2O}
tafkaa_atmo_gas_scale_factors = {1.0, 1.0, 1.0, 1.0, 1.0}
tafkaa_h2o_enter_inputs = 1
tafkaa_h2o_wl_set1 = {0.705, 0.745, 0.725}
;tafkaa_h2o_wl_set1 = {0.714, 0.736, 0.725}
tafkaa_h2o_nb_set1 = {1, 1, 1}
tafkaa_h2o_wl_set2 = {0.805, 0.845, 0.825}
;tafkaa_h2o_wl_set2 = {0.813, 0.841, 0.827}
tafkaa_h2o_nb_set2 = {1, 1, 1}

tafkaa_atmo_ozone = 0.34
tafkaa_atmo_no2_scale = 1
tafkaa_use_prev_atmo_trans = 0

; Aerosol parameters
tafkaa_aerosol_model = maritime
tafkaa_aerosol_method = 0
tafkaa_aerosol_visibility = 50
tafkaa_aerosol_tau550 = 0.1

; Mask parameters
tafkaa_use_which_masks = {none}

; Lookup table location and output parameters
tafkaa_data_directory = Tafkaa/LSF/
tafkaa_output_type = rrs

```

```
tafkaa_output_root_name = tafkaa_out/mos_2016_05_25_am_149_1
tafkaa_output_scale_factor = 10000.000000
```

For selected frames the default parameters, particularly NO<sub>2</sub> concentrations and 'aerosol visibility' were iterated on until a suitable retrieval was achieved. This process was not automated and 'suitable' generally meant that the aerosol visibility resulted in dark pixels for the far 'red' spectrum, and NO<sub>2</sub> parameter settings were used to adjust high 'blue' regions of the Rrs spectra. The NO<sub>2</sub> settings and the water vapor estimation (using absorption maximums at 725 and 825 nm) can both be improved upon by using coincident atmospheric parameters retrieval's where available. In particular the NO<sub>2</sub> concentrations required were higher than usual for ocean scenes (between 1-10 \*10<sup>15</sup> SCD, and aerosol visibility was typically between 10-50 km).

The processing of the polarized data (slits 2,3,4) was similar to the non-polarized data (slit 1). This results in an under estimation (particularly in the blue) for the remote sensing reflectance from the polarized channels -- particularly slit 2 which is approximately horizontally polarized. Nevertheless, L2a polarized data was processed to L2b products with some success -- at least in displaying relative concentration changes.

Standard NASA and band algorithms were implemented in MATLAB and used to create product maps from the ENVI standard output data cubes containing remote sensing reflectance, and these are finally bundled into HDF containers [6,7,9]. In general, parameters setting from VIIRS were used to implement the L2b processing, but no attempt was made to convolve the instruments response spectra for a specific satellite, but rather the hyperspectral data was binned to the nominal bandwidth for the reported satellite multispectral channel. Also, the nearest available satellite spectral band was used for the L2b, so for example OC3v uses channels 443 nm and 550 nm, while for MOS-P and GeoTASO processing we used channels 445 nm and 550 nm instead. This will cause some bias in the reported product field concentrations, but should not greatly effect the relative change in concentration fields in the retrials. Lastly, for the polarized retrievals, it is expected that algorithms that utilize the red bands (such as red band chlorophyll retrieval algorithms [10]) would be more accurate than those methods that rely on blue bands (e.g. OCx algorithms).

SENSOR KST	MOS-P	GEO-TASO
KORUS-OC Flights		
05/20 AM	P	P
05/20 PM	P	P
05/22 AM	P	P
05/25 AM	P	P
05/25 PM	X	P
05/28 PM	P	P
05/30 AM	P	P
05/30 PM	P	P
06/01 PM	P	P
06/02 AM	X	P
06/02 PM	P	P
06/05 AM	X	P
06/05 PM	P	P
PRE KORUS-OC FLIGHTS		
04/29 AM	X	P
05/02 AM	X	P
05/02 PM	X	P
05/04 AM	X	P
05/04 PM	X	P
05/05 AM	X	P
05/07 AM	X	P
05/07 PM	X	P
05/11 AM	X	P
05/12 MP	X	P
05/17 AM	X	P
05/17 PM	X	P
05/18 PM	X	P
POST KORUS-OC FLIGHTS		
06/09 AM	X	P
06/09 PM	X	P
06/10 AM	X	P
06/10 PM	X	P

P - Processed; X - To be Done.

#### 4. Open tasks and next steps

Table 2 provides a summary of available flight data and the data processed to date. Note that the pre- and post-MOS-P KORUS-OC is still to be done. There are also three MOS-P flights that still need to be processed. For flights between 06-01 - 06-05 there were some issues with Applanix on board system, and we are currently reprocessing those scenes and working Ball Aerospace with to get additional details for the metadata revisions. An updated report will contain maps for all the data sets. The entire data set including pre and post KORUS-OC will be delivered to NASA by Q3 2020. However, quality controlled processed data sets can be used by researchers and PACE project team members (in particular researchers at NRL are already looking some of the MOS-P data) and

will be available directly from OSU servers with password control.

Also, at the requests of other researchers, in particular members of the PACE science team, additional processing can be undertaken to attempt to refine product retrievals. A recent method known as 'C3' looks promising to better estimate the surface (Fresnel) reflections both for 'sky glint' and 'sun glint' [11].

#### 5. Concluding comments

The data in this report supports the following observations. GeoTASO can be used to generate consistent ocean color products, though at a reduced spatial resolution. Its lack of near IR bands though does limit its application to more open ocean (instead of coastal applications).

The NO<sub>2</sub> levels as estimated by the ocean color retrievals are significantly higher than expected by typical ocean color retrievals. This is not surprising given the proximity to urban land masses, but does present difficulties for traditional blue/green band algorithms. This can be addressed both by greater use of red band algorithms, as well as new methods that use joint atmospheric and ocean color retrieval's [2]. If a GeoTASO like instrument could be extended

farther into the red, they would be able to address both retrials with a single instrument, though again at a reduced spatial resolution with could limit its utility to only the open ocean.

Sun glint, though always an issue, is particularly difficult to avoid with airborne measurements. Ocean color measurements through a horizontal linear polarizer can significantly reduce glint enhancing water leaving radiance recoveries. However, at present there are no operational procedures to compute an effective  $R_{rs}$  with polarization, or large data sets to calibrate and validate such a quantity. A obvious next set, then, is to estimate the atmospheric correction for each polarization, either though a full vector simulation and/or a combination of empirical methods using all the channels.

The look up table approach used by Tafkaa, though fast, does appear to be much noisier (errors due to interpolation) then a full vector simulation. We are currently working on a full vector simulation for such scenes both to properly estimate  $R_{rs}$  for polarized channels, but also to obtain more robust recoveries. After that we will look at retrial methods that use all the linear polarization channels and compare these to those using only unpolarized data. Though it is expected that this should lead to better retrieval's, I am not aware of any field data measurements that support this conclusion, and the current data set is well structured to address this issue.

MOS-P was originally designed to optimize a wide field of view. It was retrofitted to get a narrower field of view with overlapping swaths for each polarizer. Still, at the typical flight altitude (8 km) for this campaign, a target pixel for each slit is typically offset at least 0.5-1 km, presenting a different line of sight. Thus, because of its design (depending on ocean and atmospheric variability), the atmospheric correction for the same pixel (at different polarizations) can require separate atmospheric correction simulations for each polarized channel. In the future this could be minimized by design with better co-registration, either by a fore-optic change, or a design change which located the polarizer closer (or on) the focal plane array. Alternatively, we did not explore here how this information might be exploited to better estimate heterogeneity in atmospheric conditions (since we effectively have four different lines of sight to each target pixel. Given the flight data near urban centers (both over land and ocean) this could be very interesting to analyze by atmospheric scientists.



## 6. References

- [1]. Tzortziou, M.; Parker, O.; Lamb, B.; Herman, J.R.; Lamsal, L.; Stauffer, R.; Abuhassan, N. Atmospheric Trace Gas ( $\text{NO}_2$  and  $\text{O}_3$ ) Variability in South Korean Coastal Waters, and Implications for Remote Sensing of Coastal Ocean Color Dynamics. *Remote Sens.* 2018, *10*, 1587.
- [2]. Tuffillaro, N., and Good, W., Retrieval of Ocean and Atmospheric Products Along the Coast of South Korea using GeoTASO and MOS-P, CD44D-0219 Sciences 2018.
- [3]. Judd L. M. et. al., The Dawn of Geostationary Air Quality Monitoring: Case Studies From Seoul and Los Angeles, *Frontiers in Environmental Science*, 6, 2018.
- [4]. Mobley, C.D., 2015. Polarized reflectance and transmittance properties of wind-blown sea surfaces. *Appl. Optics* 54(15), 4828-4849.
- [5]. Montes, G., & Davis, C., NRL Atmospheric Correction Algorithms: Tafkaa User's Guide, NRL/MR/7230—04- 8760.
- [6]. Werdell, P.J., and C. R. McClain. 2019. "Satellite Remote Sensing: Ocean Color." *Encyclopedia of Ocean Sciences (Third Edition)*, 5: 443-455.
- [7]. Gower, J. and King, S., FLH and MCI products from fluorescence, red tides, sargassum and blooms in ice., Second Working Meeting on MERIS and AATSR Calibration and Geophysical Validation (MAVT-2006), 20-24 March 2006.
- [8]. Ruddick, K. G. et. al, A Review of Protocols for Fiducial Reference Measurements of Water-Leaving Radiance for Validation of Satellite Remote-Sensing Data over Water, *Remote Sensing*, 11, 2019.
- [9]. Lee, et al (2002) "Deriving inherent optical properties from water color: A multi-band quasi-analytical algorithm for optically deep water" *Applied Optics*, (41) 27, 5755-5772, 2002.
- [10]. Freitas, F. and Dierssen, H. M., Evaluating the seasonal and decadal performance of red band difference algorithms for chlorophyll in an optically complex estuary with winter and summer blooms, *Remote Sensing of Environment*, 231, 2019.
- [11]. J. Pitarch, M. Talone, G. Zibordi, and P. Groetsch, "Determination of the remote-sensing reflectance from above-water measurements with the 3C model: a further assessment," *Opt. Express* 28, 15885-15906 (2020).

[12]. Vanhellemont, Q., 2019. Adaptation of the dark spectrum fitting atmospheric correction for aquatic applications of the Landsat and Sentinel-2 archives. *Remote Sensing of Environment* 225, 175–192.

## Appendix A: Data Set Summary Tables

### KORUS-OC FLIGHTS FROM 20 MAY 2016 TO 05 JUNE 2016

Folder_Name	Time_GMT	Time_KST	Comments
2016_05_20_AM_A	T221906	07:19	Clear
2016_05_20_AM_B	T234712	08:47	
2016_05_20_AM_C	T234950	08:49	
2016_05_20_PM_A	T045550	13:55	Onnuri Over flight Busan Coast
2016_05_20_PM_B	T051601	14:16	
2016_05_22_AM	T233438	08:34	Cirrus Overhead, Pigments/IOP's/Rrs
2016_05_25_AM	T235235	08:52	Clear, 4 Onnuri Overpasses, Pigments/IOP's/Rrs
2016_05_25_PM			R/V KMA Kisang, Pigments/IOP's/Rrs
2016_05_28_PM	T032320	12:23	Pigments/IOP's/Rrs
2016_05_30_AM_A	T223827	07:38	Onnuri Overpass with DC8
2016_05_30_AM_B	T020200	11:02	
2016_05_30_PM_A	T042935	13:29	Pigments/IOP's/Rrs
2016_05_30_PM_B	T052126	14:21	
2016_05_30_PM_C	T065116	15:51	
2016_06_01_PM	T041519	13:15	West of Seoul, Onnuri, Turbid waters, Pigments/IOP's/Rrs, Clear over water
2016_06_02_AM			Clear, Pigments/IOP's/Rrs
2016_06_02_PM_A	T060810	15:08	Late in day, SZA ~ 70 deg,
2016_06_02_PM_B	T063602	15:36	Pigments/IOP's/Rrs
2016_06_02_PM_C	T065545	15:54	
2016_06_02_PM_D	T071652	16:16	
2016_06_02_PM_E	T081122	17:11	
2016_06_02_PM_F	T084052	17:40	
2016_06_02_PM_G	T085456	17:54	
2016_06_05_AM			Pigments/IOP's/Rrs
2016_06_05_PM_A	T035219	12:52	Pigments/IOP's/Rrs
2016_06_05_PM_B	T040731	13:07	
2016_06_05_PM_C	T044236	13:42	
2016_06_05_PM_D	T045754	13:57	
2016_06_05_PM_E	T051128	14:11	
2016_06_05_PM_F	T052152	14:21	
2016_06_05_PM_G	T053457	14:34	
2016_06_05_PM_H	T054838	14:48	
2016_06_05_PM_I	T055725	14:57	
2016_06_05_PM_J	T061852	15:18	
2016_06_05_PM_K	T062424	15:24	
2016_06_05_PM_L	T064030	15:40	
2016_06_05_PM_M	T064413	15:44	
2016_06_05_PM_N	T072558	16:25	
2016_06_05_PM_N	T073130	16:31	
2016_06_05_PM_O	T073242	16:32	
2016_06_05_PM_P	T075609	16:56	

### PRE-KORUS-OC FLIGHTS FROM 29 APRIL 2016 TO 18 MAY 2016

2016\_04\_20\_AM  
2016\_05\_02\_AM  
2016\_05\_02\_PM

2016\_05\_04\_AM  
2016\_05\_04\_PM  
2016\_05\_05\_AM  
2016\_05\_07\_AM  
2016\_05\_07\_PM  
2016\_05\_11\_AM  
2016\_05\_12\_MD  
2016\_05\_17\_AM  
2016\_05\_17\_PM  
2015\_05\_18\_PM

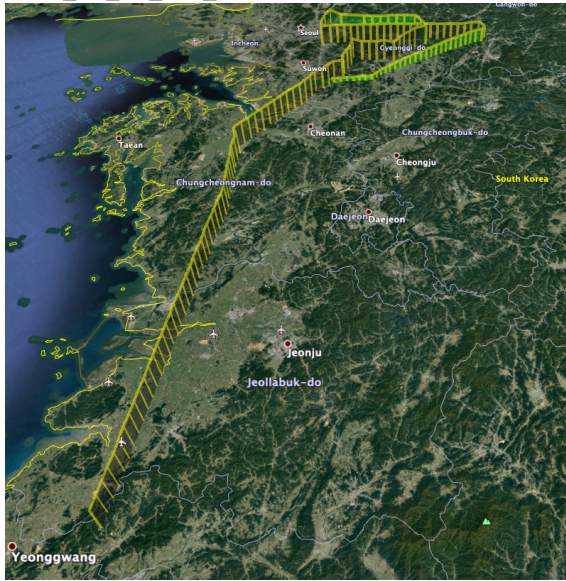
POST-KORUS-OC FLIGHTS FROM 09 JUNE 2016 TO 10 JUNE 2016

2016\_06\_09\_AM  
2016\_06\_09\_PM  
2016\_06\_10\_AM  
2016\_06\_10\_PM

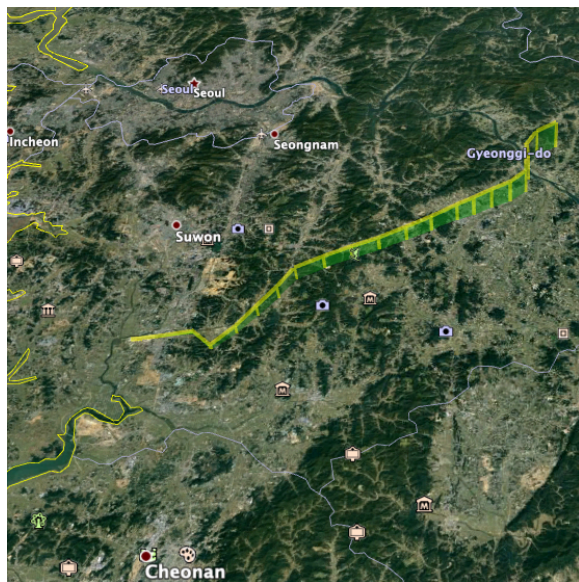


# Appendix B: Data Set Map Overviews

2016\_05\_20\_AM\_A T221906



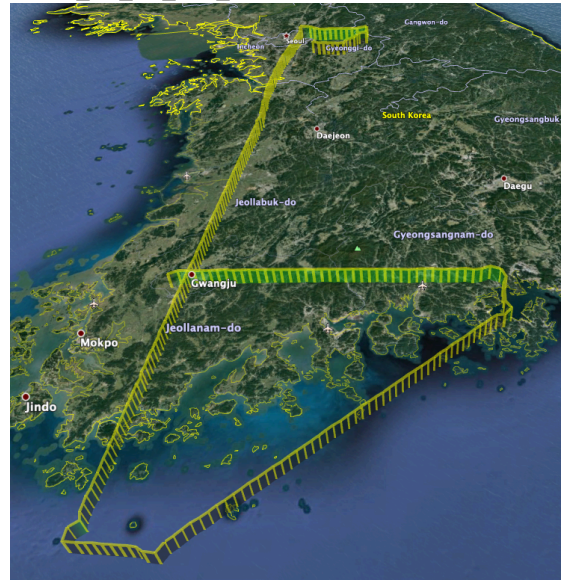
2016\_05\_20\_PM\_A



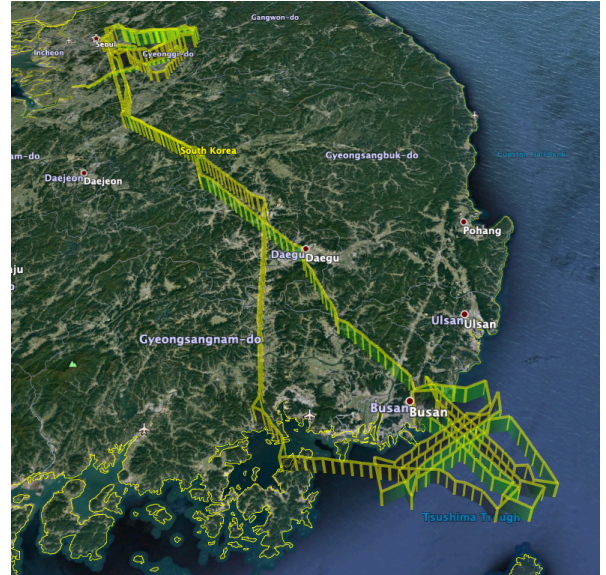
2016\_05\_22\_AM



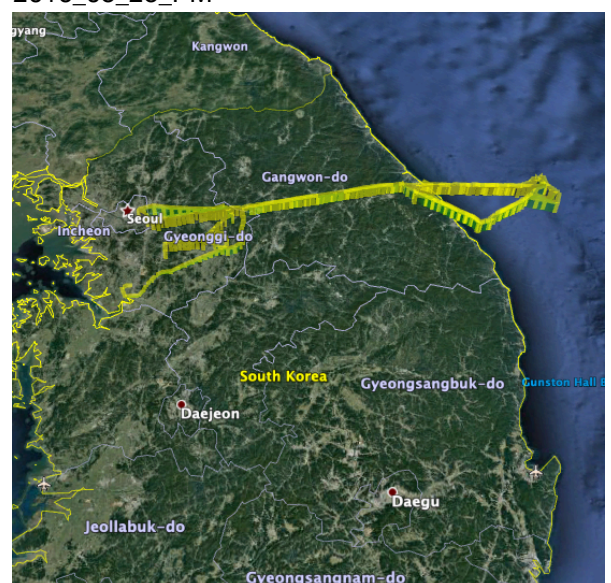
2016\_05\_20\_AM\_C



2016\_05\_20\_PM\_B



2016\_05\_25\_PM







**Image Indices (1-based)**

Spatial indices = 2016

Spectral indices = 1344

**MOS-P Wavelength Scales**

To derive the estimated wavelength for a given row:

First express row position relative to one of these four anchor rows for slits 1 through 4 (note that anchors are expressed as 1-based indices):

425 980 1532 2082

A valid resulting row will be in the range [-266 162] for any slit.

Then apply the polynomial below to derive estimated wavelength in micrometers:

$$wavelength = p1 * row^5 + p2 * row^4 + p3 * row^3 + p4 * row^2 + p5 * row + p6$$

Coefficients:

p1 = 49.817

p2 = 36.253

p3 = 12.932

p4 = 3.8628

p5 = 1.1866

p6 = 0.52033

Valid wavelength ranges are from 0.350 um to 0.900 um. Data are not valid outside these pixel ranges and wavelength ranges.

## MOS \*.HE5 Data Structure Information

The following shows the \*.he5 data structure.

```

S = HDF5.open(h5filename,'write');
S.HDFEOS = HDF5.group;
S.HDFEOS.ADDITIONAL = HDF5.group;
S.HDFEOS.ADDITIONAL.FILE_ATTRIBUTES = HDF5.group;
S.HDFEOS.ADDITIONAL.FILE_ATTRIBUTES.InstrumentName = HDF5.attribute(char(attr.InstrumentName));
S.HDFEOS.ADDITIONAL.FILE_ATTRIBUTES.ProcessLevel = HDF5.attribute(char(attr.ProcessLevel));
S.HDFEOS.ADDITIONAL.FILE_ATTRIBUTES.GranuleMonth = HDF5.attribute(int32(attr.GranuleMonth));
S.HDFEOS.ADDITIONAL.FILE_ATTRIBUTES.GranuleDay = HDF5.attribute(int32(attr.GranuleDay));
S.HDFEOS.ADDITIONAL.FILE_ATTRIBUTES.GranuleYear = HDF5.attribute(int32(attr.GranuleYear));

S.HDFEOS.SWATHS = HDF5.group;
S.HDFEOS.SWATHS.Radiance = HDF5.group;
S.HDFEOS.SWATHS.Radiance.DataFields = HDF5.group;
S.HDFEOS.SWATHS.Radiance.DataFields.RadianceMantissa = data.RadianceMantissa;
S.HDFEOS.SWATHS.Radiance.DataFields.RadianceExponent = data.RadianceExponent;
S.HDFEOS.SWATHS.Radiance.DataFields.QualityFlag = data.QualityFlag;

S.HDFEOS.SWATHS.Radiance.GeolocationFields.Latitude_Channel_One = geo.Latitude_Channel_One;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.Latitude_Channel_Two = geo.Latitude_Channel_Two;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.Latitude_Channel_Three = geo.Latitude_Channel_Three;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.Latitude_Channel_Four = geo.Latitude_Channel_Four;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.Longitude_Channel_One = geo.Longitude_Channel_One;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.Longitude_Channel_Two = geo.Longitude_Channel_Two;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.Longitude_Channel_Three = geo.Longitude_Channel_Three;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.Longitude_Channel_Four = geo.Longitude_Channel_Four;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.TerrainHeight_Channel_One = TerrainHeight(:,1);
S.HDFEOS.SWATHS.Radiance.GeolocationFields.TerrainHeight_Channel_Two = TerrainHeight(:,2);
S.HDFEOS.SWATHS.Radiance.GeolocationFields.TerrainHeight_Channel_Three = TerrainHeight(:,3);
S.HDFEOS.SWATHS.Radiance.GeolocationFields.TerrainHeight_Channel_Four = TerrainHeight(:,4);
S.HDFEOS.SWATHS.Radiance.GeolocationFields.SolarAzimuthAngle = geo.SolarAzimuthAngle;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.SolarZenithAngle = geo.SolarZenithAngle;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.AircraftAltitude = geo.AircraftAltitude;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.AircraftLatitude = geo.AircraftLatitude;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.AircraftLongitude = geo.AircraftLongitude;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.Time = geo.Time;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.Time_UTC = geo.Time_UTC;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.GPS_ITOW = geo.GPS_ITOW;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.ViewingAzimuthAngle_Channel_One = geo.ViewingAzimuthAngle_Channel_One;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.ViewingAzimuthAngle_Channel_Two = geo.ViewingAzimuthAngle_Channel_Two;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.ViewingAzimuthAngle_Channel_Three = geo.ViewingAzimuthAngle_Channel_Three;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.ViewingAzimuthAngle_Channel_Four = geo.ViewingAzimuthAngle_Channel_Four;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.ViewingZenithAngle_Channel_One = geo.ViewingZenithAngle_Channel_One;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.ViewingZenithAngle_Channel_Two = geo.ViewingZenithAngle_Channel_Two;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.ViewingZenithAngle_Channel_Three = geo.ViewingZenithAngle_Channel_Three;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.ViewingZenithAngle_Channel_Four = geo.ViewingZenithAngle_Channel_Four;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.SolarElevation = geo.SolarElevation;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.SolarAzimuth = geo.SolarAzimuth;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.EarthSunDistance = geo.EarthSunDistance;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.DetectorTemperature = geo.DetectorTemperature;
S.HDFEOS.SWATHS.Radiance.GeolocationFields.OpticalBenchTemperature = geo.OpticalBenchTemperature;

```

The following is an explanation of the structure fields.

### **FILE ATTRIBUTES:**

InstrumentName: 'MOS'. Units: String

ProcessLevel: '1B\_BALL\_yyyymmdd' where yyyymmdd is the date that the L1B level processing code was run. Type: String.

GranuleMonth: The month which was current at the first of the week in which the data was collected. Type: Integer.

GranuleDay: The day of the granule month which data was collected. Type: Integer.

GranuleYear: The year which was current at the first of the week in which the data was collected. Type: Integer.

#### **DATA FIELDS:**

Radiance = Mantissa\*10.<sup>^(Exponent)</sup>      Units:  $\frac{mW}{cm^2*sr*\mu m}$

RadianceMantissa: Mantissa of the radiance. See above for equation. Type: int16. Dimension: (spatial indices, # of images, spectral indices).

RadianceExponent: Exponent of the radiance. See above for equation. Type: int8. Dimension: (spatial indices, # of images, spectral indices).

qualityFlag: Science quality flag (0 = good quality, 1 = bad quality). Images collected within ±90 seconds of an exposure time change are considered bad (i.e., qualityFlag = 1) because the exposure time assigned to an image may be incorrect. Type: uint16. Dimension: (# of images, 1).

#### **GEOLOCATION FIELDS:**

Latitude\_Channel\_#: Latitude at the centre co-ordinate of the ground pixel, where # represents the channel number. Type: SINGLE. Dimension: (spatial indices, # of images). Units: deg.

Longitude\_Channel\_#: Longitude at the centre co-ordinate of the ground pixel, where # represents the channel number. Type: SINGLE. Dimension: (spatial indices, # of images). Units: deg.

TerrainHeight\_Channel\_#: Orthometric height of the terrain at the center co-ordinate of the ground pixel, where # represents the channel number. Type: int16. Dimension: (spatial indices, # of images). Units: m.

SolarAzimuthAngle: Solar azimuth angle at the centre co-ordinate of the ground pixel. Type: SINGLE. Dimension: (spatial indices, # of images). Units: deg.

SolarZenithAngle: Solar zenith angle at the centre co-ordinate of the ground pixel. Type: SINGLE. Dimension: (spatial indices, # of images). Units: deg.

AircraftAltitude: Altitude (i.e., orthometric height) of aircraft at time image was taken. Type: SINGLE. Dimension: (# of images, 1). Units: m.

AircraftLatitude: Latitude of aircraft at time image was taken. Type: SINGLE. Dimension: (# of images, 1). Units: deg.

AircraftLongitude: Longitude of aircraft at time image was taken. Type: SINGLE. Dimension: (# of images, 1). Units: deg.

Time: Time in International Atomic Time (TAI-93) format. Type: DOUBLE. Dimension: (# of images, 1). Units: seconds.

Time UTC: Time in UTC - 'yyyymmddTHHMMSS'. Dimension: (# of images, 1). Units: String

GPS\_ITOW: Time, from the beginning of the week, that the image was taken. . Type: DOUBLE. Dimension: (# of images, 1). Units: seconds.

ViewingAzimuthAngle\_Channel #: Viewing azimuth angle at the centre co-ordinate of the ground pixel, where # represents the channel number. Type: SINGLE. Dimension: (spatial indices, # of images). Units: Degrees.

ViewingZenithAngle\_Channel #: Viewing zenith angle at the centre co-ordinate of the ground pixel, where # represents the channel number. Type: SINGLE. Dimension: (spatial indices, # of images). Units: Degrees.

SolarElevation: Elevation angle of the Sun in the MOS reference frame. Type: SINGLE. Dimension: (# of images, 1). Units: deg.

SolarAzimuth: Azimuth angle of the Sun in the MOS reference frame. Type: SINGLE. Dimension: (# of images, 1). Units: deg.

EarthSunDistance: Distance between the Earth and the Sun. Type: Integer. Units: m.

DetectorTemperature: Temperature of the MOS detector. Type: SINGLE. Dimension: (# of images, 1). Units: °C.

OpticalBenchTemperature: Temperature of the MOS optical bench. Type: SINGLE. Dimension: (# of images, 1). Units: °C.



## MOS \*.MAT Data Structure Information

The following shows the \*.mat data structure.

```

attr.InstrumentName = 'MOS';
attr.ProcessLevel = ProcessLevel;
attr.GranuleMonth = localtime.month;
attr.GranuleDay = localtime.dayofmonth;
attr.GranuleYear = localtime.year;

data.RadianceMantissa = RadianceMantissa;
data.RadianceExponent = RadianceExponent;
data.QualityFlag = QualityFlag;

geo.Latitude_Channel_One = Latitude(:,1);
geo.Latitude_Channel_Two = Latitude(:,2);
geo.Latitude_Channel_Three = Latitude(:,3);
geo.Latitude_Channel_Four = Latitude(:,4);
geo.Longitude_Channel_One = Longitude(:,1);
geo.Longitude_Channel_Two = Longitude(:,2);
geo.Longitude_Channel_Three = Longitude(:,3);
geo.Longitude_Channel_Four = Longitude(:,4);
geo.TerrainHeight_Channel_One = TerrainHeight(:,1);
geo.TerrainHeight_Channel_Two = TerrainHeight(:,2);
geo.TerrainHeight_Channel_Three = TerrainHeight(:,3);
geo.TerrainHeight_Channel_Four = TerrainHeight(:,4);
geo.SolarAzimuthAngle = SolarAzimuthAngle;
geo.SolarZenithAngle = SolarZenithAngle;
geo.AircraftAltitude = AircraftAltitude;
geo.AircraftLatitude = AircraftLatitude;
geo.AircraftLongitude = AircraftLongitude;
geo.Time = Time;
geo.Time_UTC = Time_UTC;
geo.GPS_ITOW = GPS_ITOW;
geo.ViewingAzimuthAngle_Channel_One = ViewingAzimuthAngle(:,1);
geo.ViewingAzimuthAngle_Channel_Two = ViewingAzimuthAngle(:,2);
geo.ViewingAzimuthAngle_Channel_Three = ViewingAzimuthAngle(:,3);
geo.ViewingAzimuthAngle_Channel_Four = ViewingAzimuthAngle(:,4);
geo.ViewingZenithAngle_Channel_One = ViewingZenithAngle(:,1);
geo.ViewingZenithAngle_Channel_Two = ViewingZenithAngle(:,2);
geo.ViewingZenithAngle_Channel_Three = ViewingZenithAngle(:,3);
geo.ViewingZenithAngle_Channel_Four = ViewingZenithAngle(:,4);
geo.SolarElevation = SolarElevation;
geo.SolarAzimuth = SolarAzimuth;
geo.EarthSunDistance = EarthSunDistance;
geo.DetectorTemperature = DetectorTemperature;
geo.OpticalBenchTemperature = OpticalBenchTemperature;

```

The following is an explanation of the structure fields.

### **FILE ATTRIBUTES:**

InstrumentName: 'MOS'. Units: String

ProcessLevel: '1B\_BALL\_yyyymmdd' where yyyymmdd is the date that the L1B level processing code was run. Type: String.

GranuleMonth: The month which was current at the first of the week in which the data was collected. Type: Integer.

GranuleDay: The day of the granule month which data was collected. Type: Integer.

GranuleYear: The year which was current at the first of the week in which the data was collected. Type: Integer.

#### DATA FIELDS:

Radiance = Mantissa\*10.<sup>^(Exponent)</sup>      Units:  $\frac{mW}{cm^2*sr*\mu m}$

RadianceMantissa: Mantissa of the radiance. See above for equation. Type: int16. Dimension: (# of images, spectral indices, spatial indices).

RadianceExponent: Exponent of the radiance. See above for equation. Type: int8. Dimension: (# of images, spectral indices, spatial indices).

qualityFlag: Science quality flag (0 = good quality, 1 = bad quality). Images collected within  $\pm 90$  seconds of an exposure time change are considered bad (i.e., qualityFlag = 1) because the exposure time assigned to an image may be incorrect. Type: uint16. Dimension: (1, # of images).

#### GEOLOCATION FIELDS:

Latitude\_Channel\_#: Latitude at the centre co-ordinate of the ground pixel, where # represents the channel number. Type: SINGLE. Dimension: (# of images, spatial indices). Units: deg.

Longitude\_Channel\_#: Longitude at the centre co-ordinate of the ground pixel, where # represents the channel number. Type: SINGLE. Dimension: (# of images, spatial indices). Units: deg.

TerrainHeight\_Channel\_#: Orthometric height of the terrain at the center co-ordinate of the ground pixel, where # represents the channel number. Type: int16. Dimension: (# of images, spatial indices). Units: m.

SolarAzimuthAngle: Solar azimuth angle at the centre co-ordinate of the ground pixel. Type: SINGLE. Dimension: (# of images, spatial indices). Units: deg.

SolarZenithAngle: Solar zenith angle at the centre co-ordinate of the ground pixel. Type: SINGLE. Dimension: (# of images, spatial indices). Units: deg.

AircraftAltitude: Altitude (i.e., orthometric height) of aircraft at time image was taken. Type: SINGLE. Dimension: (1, # of images). Units: m.

AircraftLatitude: Latitude of aircraft at time image was taken. Type: SINGLE. Dimension: (1, # of images). Units: deg.

AircraftLongitude: Longitude of aircraft at time image was taken. Type: SINGLE. Dimension: (1, # of images). Units: deg.

Time: Time in International Atomic Time (TAI-93) format. Type: DOUBLE. Dimension: (1, # of images). Units: seconds.

Time.UTC: Time in UTC - 'yyyymmddTHHMMSS'. Dimension: (1, # of images). Units: String

GPS\_ITOW: Time, from the beginning of the week, that the image was taken. . Type: DOUBLE. Dimension: (1, # of images). Units: seconds.

ViewingAzimuthAngle\_Channel #: Viewing azimuth angle at the centre co-ordinate of the ground pixel, where # represents the channel number. Type: SINGLE. Dimension: (# of images, spatial indices). Units: Degrees.

ViewingZenithAngle\_Channel #: Viewing zenith angle at the centre co-ordinate of the ground pixel, where # represents the channel number. Type: SINGLE. Dimension: (# of images, spatial indices). Units: Degrees.

SolarElevation: Elevation angle of the Sun in the MOS reference frame. Type: SINGLE. Dimension: (1, # of images). Units: deg.

SolarAzimuth: Azimuth angle of the Sun in the MOS reference frame. Type: SINGLE. Dimension: (1, # of images). Units: deg.

EarthSunDistance: Distance between the Earth and the Sun. Type: Integer. Units: m.

DetectorTemperature: Temperature of the MOS detector. Type: SINGLE. Dimension: (1, # of images). Units: °C.

OpticalBenchTemperature: Temperature of the MOS optical bench. Type: SINGLE. Dimension: (1, # of images). Units: °C.