Himawari Support in the CSPP-GEO Direct Broadcast Package

Geoff Cureton

Cooperative Institute for Meteorological Satellite Studies
Space Science and Engineering Center
University of Wisconsin - Madison
Madison, Wisconsin, USA

CSPP/IMAPP Users Group Meeting 2017

Outline

What is CSPP GEO-Geocat?

- The CSPP GEO-Geocat package is a collection of scripts, executables, ancillary and auxillary files used to generate level-1 and level-2 output from geostationary satellite data.
- CSPP GEO-Geocat package processes Himawari Standard Data (HSD) and HimawariCast direct broadcast data through level-1 and level-2 NetCDF files for the Advnced Himawari Imager (AHI) on Himawari-8.
- CSPP GEO-Geocat is built upon three things: Geocat, Python, and bash scripting.

What is CSPP GEO-Geocat?

- The CSPP GEO-Geocat package is a collection of scripts, executables, ancillary and auxillary files used to generate level-1 and level-2 output from geostationary satellite data.
- CSPP GEO-Geocat package processes Himawari Standard Data (HSD) and HimawariCast direct broadcast data through level-1 and level-2 NetCDF files for the Advnced Himawari Imager (AHI) on Himawari-8.
- CSPP GEO-Geocat is built upon three things: Geocat, Python, and bash scripting.

What is CSPP GEO-Geocat?

- The CSPP GEO-Geocat package is a collection of scripts, executables, ancillary and auxillary files used to generate level-1 and level-2 output from geostationary satellite data.
- CSPP GEO-Geocat package processes Himawari Standard Data (HSD) and HimawariCast direct broadcast data through level-1 and level-2 NetCDF files for the Advnced Himawari Imager (AHI) on Himawari-8.
- CSPP GEO-Geocat is built upon three things: Geocat, Python, and bash scripting.

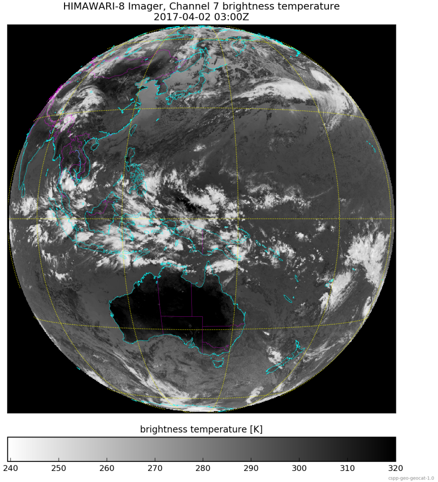
What is CSPP GEO-Geocat?

- The CSPP GEO-Geocat package is a collection of scripts, executables, ancillary and auxillary files used to generate level-1 and level-2 output from geostationary satellite data.
- CSPP GEO-Geocat package processes Himawari Standard Data (HSD) and HimawariCast direct broadcast data through level-1 and level-2 NetCDF files for the Advnced Himawari Imager (AHI) on Himawari-8.
- CSPP GEO-Geocat is built upon three things: Geocat, Python, and bash scripting.

Supported Satellites

In addition to Himawari-8, CSPP GEO-Geocat also supports. . .

- GOES-13
- GOES-15

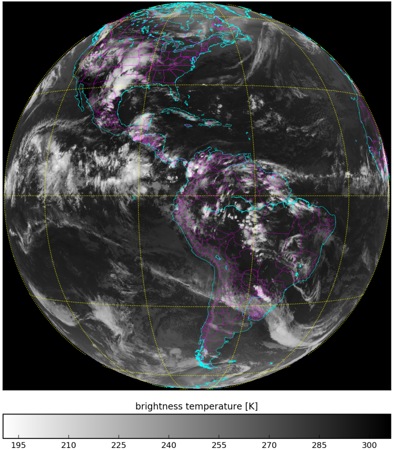


Supported Satellites

In addition to Himawari-8, CSPP GEO-Geocat also supports. . .

- GOES-13
- GOES-15

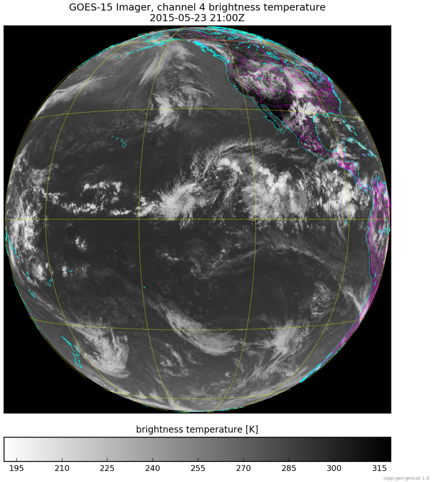
GOES-13 Imager, channel 4 brightness temperature
2015-05-23 23:45Z



Supported Satellites

In addition to Himawari-8, CSPP GEO-Geocat also supports. . .

- GOES-13
- GOES-15



The Geostationary Cloud Algorithm Testbed (Geocat)

The Guts. . .

- Compiled binary (Fortran 90)
- Runs cloud algorithms on level-1 data, outputs level-1 and level-2
- Modular algorithm handling allows algorithm developers to rapidly test and compare different cloud algorithms
- Navigation can be computed on-the-fly, or can be ingested as auxillary data

The Geostationary Cloud Algorithm Testbed (Geocat)

The Guts...

- Compiled binary (Fortran 90)
- Runs cloud algorithms on level-1 data, outputs level-1 and level-2
- Modular algorithm handling allows algorithm developers to rapidly test and compare different cloud algorithms
- Navigation can be computed on-the-fly, or can be ingested as auxillary data

The Geostationary Cloud Algorithm Testbed (Geocat)

The Guts. . .

- Compiled binary (Fortran 90)
- Runs cloud algorithms on level-1 data, outputs level-1 and level-2
- Modular algorithm handling allows algorithm developers to rapidly test and compare different cloud algorithms
- Navigation can be computed on-the-fly, or can be ingested as auxillary data

The Geostationary Cloud Algorithm Testbed (Geocat)

The Guts...

- Compiled binary (Fortran 90)
- Runs cloud algorithms on level-1 data, outputs level-1 and level-2
- Modular algorithm handling allows algorithm developers to rapidly test and compare different cloud algorithms
- Navigation can be computed on-the-fly, or can be ingested as auxillary data

The Geostationary Cloud Algorithm Testbed (Geocat)

The Guts. . .

- Compiled binary (Fortran 90)
- Runs cloud algorithms on level-1 data, outputs level-1 and level-2
- Modular algorithm handling allows algorithm developers to rapidly test and compare different cloud algorithms
- Navigation can be computed on-the-fly, or can be ingested as auxillary data

The Geostationary Cloud Algorithm Testbed (Geocat)

continued...

- Geocat requires a fair amount of ancillary data, with complicated ingest rules.
- A typical geocat invocation looks like...

```
./geocat -time_report -fast_planck -verbose -native_channels -maxsatzen 85 -aformat 1 _ \
-nscans 99999 -tmp_dir ./cspp_geo_temporal_cache_dir/HS_H08_20170402_0300_B05_FLDK_R20_S0110_seg_0 \
-l1_dir ./cspp_geo_temporal_cache_dir/HS_H08_20170402_0300_B05_FLDK_R20_S0110_seg_0 \
-dumpch 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 -nwp gfs -nwp_dir ./anc -nwp_meso rap13 \
-nwp_meso_dir ./anc -nwp_meso_forecast 2 3 -use_snow -snow_dir ./anc -use_sst -sst_dir ./anc \
-sst_source oisst_daily_avhrr_only -use_seebor -use_albedo \
-l2_dir ./cspp_geo_temporal_cache_dir/HS_H08_20170402_0300_B05_FLDK_R20_S0110_seg_0 \
-cmask eps_cmask_ahi -ctype enterprise_cldphase_10_11_13_14_15 -cldz ACHA_mode_8 -cldtau_day DCOMP_mode_3 \
-akey eps_cmask_ahi enterprise_cldphase_10_11_13_14_15 ACHA_mode_8 DCOMP_mode_3 \
night_optprop goesr_fog_bridge \
-area_dir ./hsd -file_type ahi_hsf -f HS_H08_20170402_0300_B05_FLDK \

-y 1575 1799 -x 5023 5247 1
```

- We can improve this for DB users.

The Geostationary Cloud Algorithm Testbed (Geocat)

continued. . .

- Geocat requires a fair amount of ancillary data, with complicated ingest rules.
- A typical geocat invocation looks like. . .

```
./geocat -time_report -fast_planck -verbose -native_channels -maxsatzen 85 -aformat 1 _ \
-nscans 99999 -tmp_dir ./cspp_geo_temporal_cache_dir/HS_H08_20170402_0300_B05_FLDK_R20_S0110_seg_0 \
-l1_dir ./cspp_geo_temporal_cache_dir/HS_H08_20170402_0300_B05_FLDK_R20_S0110_seg_0 \
-dumpch 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 -nwp gfs -nwp_dir ./anc -nwp_meso rap13 \
-nwp_meso_dir ./anc -nwp_meso_forecast 2 3 -use_snow -snow_dir ./anc -use_sst -sst_dir ./anc \
-sst_source oisst_daily_avhrr_only -use_seebor -use_albedo \
-l2_dir ./cspp_geo_temporal_cache_dir/HS_H08_20170402_0300_B05_FLDK_R20_S0110_seg_0 \
-cmask eps_cmask_ahi -ctype enterprise_cldphase_10_11_13_14_15 -cldz ACHA_mode_8 -cldtau_day DCOMP_mode_3 \
-akey eps_cmask_ahi enterprise_cldphase_10_11_13_14_15 ACHA_mode_8 DCOMP_mode_3 \
night_optprop goesr_fog_bridge \
-area_dir ./hsd -file_type ahi_hsf -f HS_H08_20170402_0300_B05_FLDK \

-y 1575 1799 -x 5023 5247 1
```

- We can improve this for DB users.

The Geostationary Cloud Algorithm Testbed (Geocat)

continued. . .

- Geocat requires a fair amount of ancillary data, with complicated ingest rules.
- A typical geocat invocation looks like. . .

```
 ./geocat -time_report -fast_planck -verbose -native_channels -maxsatzen 85 -aformat 1 _ \
-nscans 99999 -tmp_dir ./cspp_geo_temporal_cache_dir/HS_H08_20170402_0300_B05_FLDK_R20_S0110_seg_0 \
-l1_dir ./cspp_geo_temporal_cache_dir/HS_H08_20170402_0300_B05_FLDK_R20_S0110_seg_0 \
-dumpch 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 -nwp gfs -nwp_dir ./anc -nwp_meso rap13 \
-nwp_meso_dir ./anc -nwp_meso_forecast 2 3 -use_snow -snow_dir ./anc -use_sst -sst_dir ./anc \
-sst_source oisst_daily_avhrr_only -use_seebor -use_albedo \
-l2_dir ./cspp_geo_temporal_cache_dir/HS_H08_20170402_0300_B05_FLDK_R20_S0110_seg_0 \
-cmask eps_cmask_ahi -ctype enterprise_cldphase_10_11_13_14_15 -cldz ACHA_mode_8 -cldtau_day DCOMP_mode_3 \
-akey eps_cmask_ahi enterprise_cldphase_10_11_13_14_15 ACHA_mode_8 DCOMP_mode_3 \
night_optprop goesr_fog_bridge \
-area_dir ./hsd -file_type ahi_hsf -f HS_H08_20170402_0300_B05_FLDK \

-y 1575 1799 -x 5023 5247 1
```

- We can improve this for DB users.

The Geostationary Cloud Algorithm Testbed (Geocat)

continued...

- Geocat requires a fair amount of ancillary data, with complicated ingest rules.
- A typical geocat invocation looks like...

```
 ./geocat -time_report -fast_planck -verbose -native_channels -maxsatzen 85 -aformat 1 _ \
-nscans 99999 -tmp_dir ./cspp_geo_temporal_cache_dir/HS_H08_20170402_0300_B05_FLDK_R20_S0110_seg_0 \
-l1_dir ./cspp_geo_temporal_cache_dir/HS_H08_20170402_0300_B05_FLDK_R20_S0110_seg_0 \
-dumpch 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 -nwp gfs -nwp_dir ./anc -nwp_meso rap13 \
-nwp_meso_dir ./anc -nwp_meso_forecast 2 3 -use_snow -snow_dir ./anc -use_sst -sst_dir ./anc \
-sst_source oisst_daily_avhrr_only -use_seebor -use_albedo \
-l2_dir ./cspp_geo_temporal_cache_dir/HS_H08_20170402_0300_B05_FLDK_R20_S0110_seg_0 \
-cmask eps_cmask_ahi -ctype enterprise_cldphase_10_11_13_14_15 -cldz ACHA_mode_8 -cldtau_day DCOMP_mode_3 \
-akey eps_cmask_ahi enterprise_cldphase_10_11_13_14_15 ACHA_mode_8 DCOMP_mode_3 \
night_optprop goesr_fog_bridge \
-area_dir ./hsd -file_type ahi_hsf -f HS_H08_20170402_0300_B05_FLDK \

-y 1575 1799 -x 5023 5247 1
```

- We can improve this for DB users.

Python Scripting

The Glue...

We can improve the user interface to geocat by wrapping it in python scripting.

We use python scripting to...

- Handle command line options for controlling package behaviour.
- Inventory the input files to determine processing candidates.
- Perform any post-processing on the geocat output.
- Parsing of log files looking for error conditions, and general error handling.

Python Scripting

The Glue. . .

We can improve the user interface to geocat by wrapping it in python scripting.
We use python scripting to. . .

- Handle command line options for controlling package behaviour.
- Inventory the input files to determine processing candidates.
- Perform any post-processing on the geocat output.
- Parsing of log files looking for error conditions, and general error handling.

Python Scripting

The Glue...

We can improve the user interface to geocat by wrapping it in python scripting.

We use python scripting to...

- Handle command line options for controlling package behaviour.
- Inventory the input files to determine processing candidates.
- Perform any post-processing on the geocat output.
- Parsing of log files looking for error conditions, and general error handling.

Python Scripting

The Glue...

We can improve the user interface to geocat by wrapping it in python scripting.
We use python scripting to...

- Handle command line options for controlling package behaviour.
- Inventory the input files to determine processing candidates.
- Perform any post-processing on the geocat output.
- Parsing of log files looking for error conditions, and general error handling.

Python Scripting

The Glue. . .

We can improve the user interface to geocat by wrapping it in python scripting.
We use python scripting to. . .

- Handle command line options for controlling package behaviour.
- Inventory the input files to determine processing candidates.
- Perform any post-processing on the geocat output.
- Parsing of log files looking for error conditions, and general error handling.

Python Scripting

The Glue...

We can improve the user interface to geocat by wrapping it in python scripting.
We use python scripting to...

- Handle command line options for controlling package behaviour.
- Inventory the input files to determine processing candidates.
- Perform any post-processing on the geocat output.
- Parsing of log files looking for error conditions, and general error handling.

Bash Scripting

The Wrapper...

Not much to say here...

- The python command line interface is superficially implemented in a bash script, if that's what is preferred.
- A bit more cumbersome, but it's perhaps useful to paper over the fact that a user is running a python script, if bash is what they're used to.
- Then again, we can set up various environment varables in the bash script, in a way probably familiar to the user.

Bash Scripting

The Wrapper...

Not much to say here...

- The python command line interface is superficially implemented in a bash script, if that's what is preferred.
- A bit more cumbersome, but it's perhaps useful to paper over the fact that a user is running a python script, if bash is what they're used to.
- Then again, we can set up various environment varables in the bash script, in a way probably familiar to the user.

Bash Scripting

The Wrapper. . .

Not much to say here. . .

- The python command line interface is superficially implemented in a bash script, if that's what is preferred.
- A bit more cumbersome, but it's perhaps useful to paper over the fact that a user is running a python script, if bash is what they're used to.
- Then again, we can set up various environment varables in the bash script, in a way probably familiar to the user.

Bash Scripting

The Wrapper...

Not much to say here...

- The python command line interface is superficially implemented in a bash script, if that's what is preferred.
- A bit more cumbersome, but it's perhaps useful to paper over the fact that a user is running a python script, if bash is what they're used to.
- Then again, we can set up various environment varables in the bash script, in a way probably familiar to the user.

CSPP GEO-Geocat Usage

The bare minumum invocation. . .

- Script name
- Output directory (will create if required, otherwise current dir)
- Input files or directories

CSPP GEO-Geocat Usage

The bare minumum invocation...

```
geocat_l2.sh \
```

- Script name
- Output directory (will create if required, otherwise current dir)
- Input files or directories

CSPP GEO-Geocat Usage

The bare minumum invocation...

```
geocat_l2.sh \
          -W output_dir \
```

- Script name
- Output directory (will create if required, otherwise current dir)
- Input files or directories

CSPP GEO-Geocat Usage

The bare minumum invocation. . .

```
geocat_l2.sh \
          -W output_dir \
          inputs_dir/HS_H08_20170402_0300_B05_FLDK_R20_S0110.DAT
```

- Script name
- Output directory (will create if required, otherwise current dir)
- Input files or directories

Other Use Cases. . .

Extra command line options. . .

```
geocat_l2.sh \
            -W output_dir \
            inputs_dir/HS_H08_20170402_0300_B05_FLDK_R20_S0110.DAT \
```

- Only retrieve and process ancillary data, don't run geocat.
- List the file metadata, and exit
- Do not clean out working directory

Other Use Cases. . .

Extra command line options. . .

```
geocat_l2.sh \
            -W output_dir \
            inputs_dir/HS_H08_20170402_0300_B05_FLDK_R20_S0110.DAT \
            --ancillary-only
```

- Only retrieve and process ancillary data, don't run geocat.
- List the file metadata, and exit
- Do not clean out working directory

Other Use Cases...

Extra command line options...

```
geocat_l2.sh \
            -W output_dir \
            inputs_dir/HS_H08_20170402_0300_B05_FLDK_R20_S0110.DAT \
            --ancillary-only --interrogate
```

- Only retrieve and process ancillary data, don't run geocat.
- List the file metadata, and exit
- Do not clean out working directory

Other Use Cases...

Extra command line options...

```
geocat_l2.sh \
           -W output_dir \
           inputs_dir/HS_H08_20170402_0300_B05_FLDK_R20_S0110.DAT \
           --ancillary-only --interrogate --debug
```

- Only retrieve and process ancillary data, don't run geocat.
- List the file metadata, and exit
- Do not clean out working directory

Example HSD level-1 output

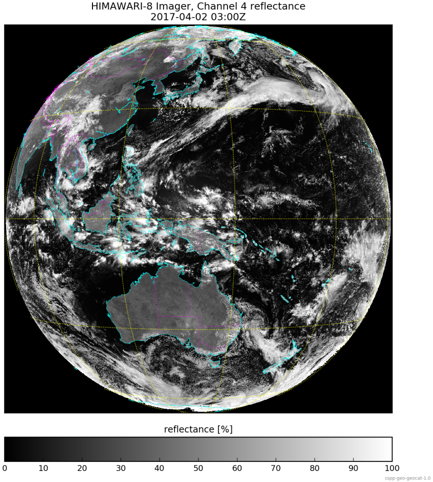
Every 10 minutes we receive from AHI. . .

- One (1) 2km resolution full disk image
- Four (4) Japan Sector images (every 2.5 minutes)
- Four (4) Mesoscale images (every 2.5 minutes)

Example HSD level-1 output

Every 10 minutes we receive from AHI. . .

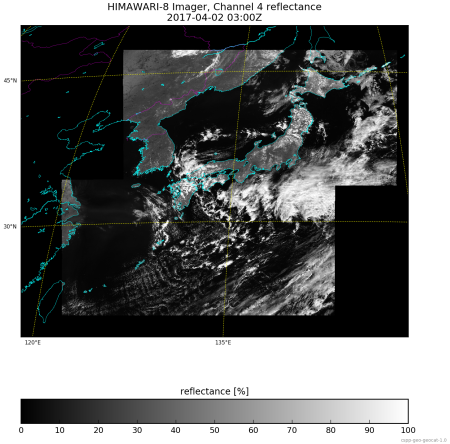
- One (1) 2km resolution full disk image
- Four (4) Japan Sector images (every 2.5 minutes)
- Four (4) Mesoscale images (every 2.5 minutes)



Example HSD level-1 output

Every 10 minutes we receive from AHI. . .

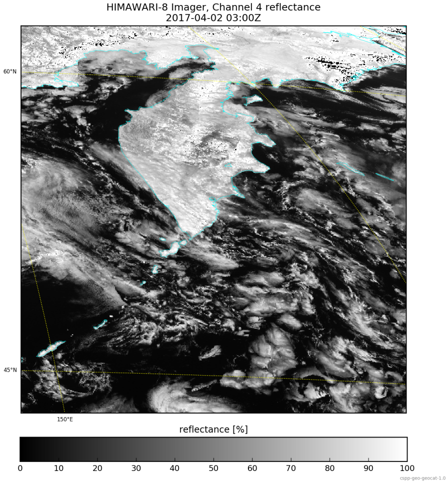
- One (1) 2km resolution full disk image
- Four (4) Japan Sector images (every 2.5 minutes)
- Four (4) Mesoscale images (every 2.5 minutes)



Example HSD level-1 output

Every 10 minutes we receive from AHI. . .

- One (1) 2km resolution full disk image
- Four (4) Japan Sector images (every 2.5 minutes)
- Four (4) Mesoscale images (every 2.5 minutes)



Example HSD level-2 output

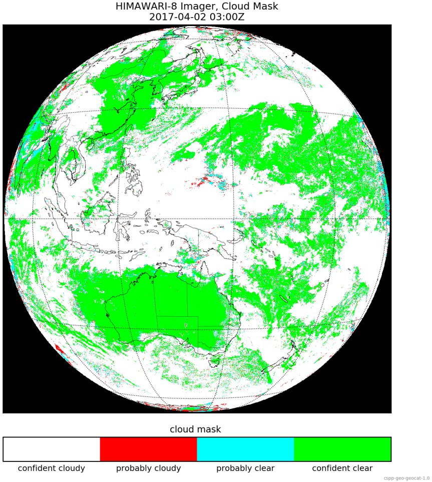
For every AHI image we generate level-2 products, examples of which are. . .

- Cloud Mask
- Cloud Type
- Cloud Top Temperature
- Fog MVFR

Example HSD level-2 output

For every AHI image we generate level-2 products, examples of which are. . .

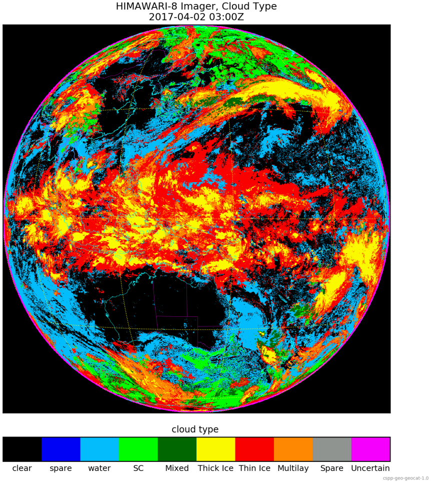
- Cloud Mask
- Cloud Type
- Cloud Top Temperature
- Fog MVFR



Example HSD level-2 output

For every AHI image we generate level-2 products, examples of which are. . .

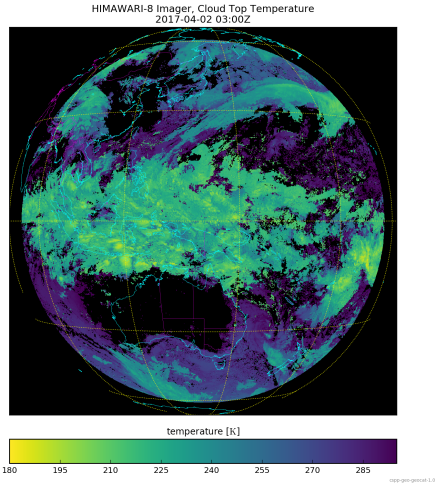
- Cloud Mask
- Cloud Type
- Cloud Top Temperature
- Fog MVFR



Example HSD level-2 output

For every AHI image we generate level-2 products, examples of which are. . .

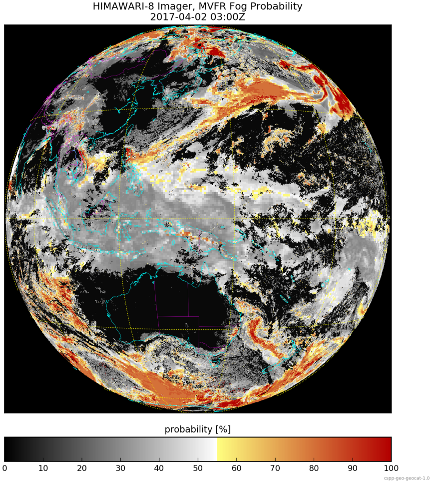
- Cloud Mask
- Cloud Type
- Cloud Top Temperature
- Fog MVFR



Example HSD level-2 output

For every AHI image we generate level-2 products, examples of which are. . .

- Cloud Mask
- Cloud Type
- Cloud Top Temperature
- Fog MVFR



Reducing Latency

To enable near-real-time processing for HSD, allowance needs to be made for the greatly increase data rate, due to...

- Increased spatial resolution
- Greater number of bands
- 10-minute duty cycle for full disk and all other regions
- To reduce latency, we use two approaches: *segmentation* and *subsetting*

Reducing Latency

To enable near-real-time processing for HSD, allowance needs to be made for the greatly increase data rate, due to...

- Increased spatial resolution
- Greater number of bands
- 10-minute duty cycle for full disk and all other regions
- To reduce latency, we use two approaches: *segmentation* and *subsetting*

Reducing Latency

To enable near-real-time processing for HSD, allowance needs to be made for the greatly increase data rate, due to...

- Increased spatial resolution
- Greater number of bands
- 10-minute duty cycle for full disk and all other regions
- To reduce latency, we use two approaches: *segmentation* and *subsetting*

Reducing Latency

To enable near-real-time processing for HSD, allowance needs to be made for the greatly increase data rate, due to...

- Increased spatial resolution
- Greater number of bands
- 10-minute duty cycle for full disk and all other regions
- To reduce latency, we use two approaches: *segmentation* and *subsetting*

Reducing Latency

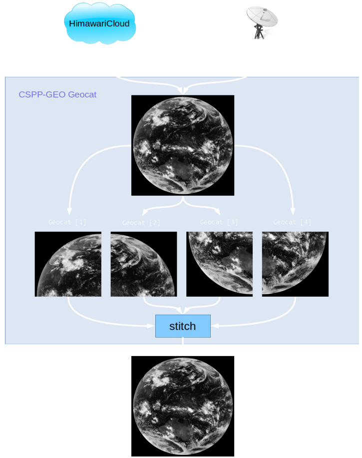
To enable near-real-time processing for HSD, allowance needs to be made for the greatly increase data rate, due to...

- Increased spatial resolution
- Greater number of bands
- 10-minute duty cycle for full disk and all other regions
- To reduce latency, we use two approaches: *segmentation* and *subsetting*

Segmentation

Each input file is (with full disk as an example)...

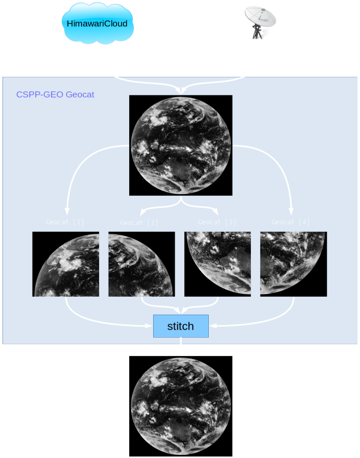
- Split into $M \times N$ segments (default is 2×2
- Submit separate geocat processes (using the python `multiprocessing` module to a processing pool to be run in parallel.
- Stitch back together the resulting $M \times N$ output files, for both level-1 and level-2.
- Processing for a complete set of regions for a 10 minute duty cycle reduced from almost 20 minutes to just under 10 minutes.



Segmentation

Each input file is (with full disk as an example)...

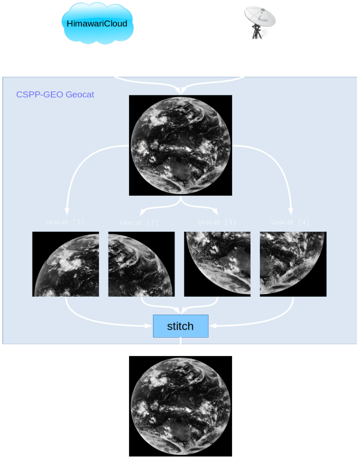
- Split into $M \times N$ segments (default is 2×2
- Submit separate geocat processes (using the python `multiprocessing` module to a processing pool to be run in parallel.
- Stitch back together the resulting $M \times N$ output files, for both level-1 and level-2.
- Processing for a complete set of regions for a 10 minute duty cycle reduced from almost 20 minutes to just under 10 minutes.



Segmentation

Each input file is (with full disk as an example)...

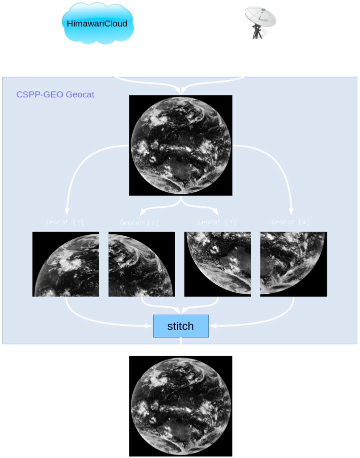
- Split into $M \times N$ segments (default is 2×2
- Submit separate geocat processes (using the python `multiprocessing` module to a processing pool to be run in parallel.
- Stitch back together the resulting $M \times N$ output files, for both level-1 and level-2.
- Processing for a complete set of regions for a 10 minute duty cycle reduced from almost 20 minutes to just under 10 minutes.



Segmentation

Each input file is (with full disk as an example)...

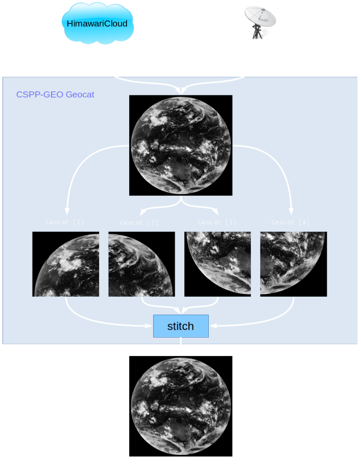
- Split into $M \times N$ segments (default is 2×2
- Submit separate geocat processes (using the python `multiprocessing` module to a processing pool to be run in parallel.
- Stitch back together the resulting $M \times N$ output files, for both level-1 and level-2.
- Processing for a complete set of regions for a 10 minute duty cycle reduced from almost 20 minutes to just under 10 minutes.



Segmentation

Each input file is (with full disk as an example)...

- Split into $M \times N$ segments (default is 2×2
- Submit separate geocat processes (using the python `multiprocessing` module to a processing pool to be run in parallel.
- Stitch back together the resulting $M \times N$ output files, for both level-1 and level-2.
- Processing for a complete set of regions for a 10 minute duty cycle reduced from almost 20 minutes to just under 10 minutes.



Segment Padding

Various level-2 algorithms have spatial dependencies, which would result in edge artifacts when the image segments are stitched back together. We mitigate this by...

- Defining each segment to include a buffer along the interior edges
- Create a mask defining the buffer region
- Stitch together the image segments, including the extra padding
- Use the mask to knock out the padding regions, which are of suspect quality

Segment Padding

Various level-2 algorithms have spatial dependencies, which would result in edge artifacts when the image segments are stitched back together. We mitigate this by...

- Defining each segment to include a buffer along the interior edges
- Create a mask defining the buffer region
- Stitch together the image segments, including the extra padding
- Use the mask to knock out the padding regions, which are of suspect quality

Segment Padding

Various level-2 algorithms have spatial dependencies, which would result in edge artifacts when the image segments are stitched back together. We mitigate this by...

- Defining each segment to include a buffer along the interior edges
- Create a mask defining the buffer region
- Stitch together the image segments, including the extra padding
- Use the mask to knock out the padding regions, which are of suspect quality

Segment Padding

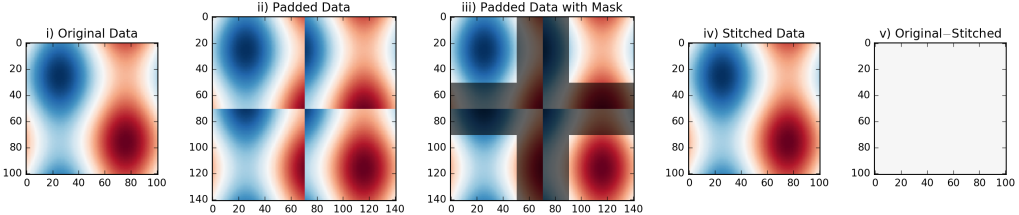
Various level-2 algorithms have spatial dependencies, which would result in edge artifacts when the image segments are stitched back together. We mitigate this by…

- Defining each segment to include a buffer along the interior edges
- Create a mask defining the buffer region
- Stitch together the image segments, including the extra padding
- Use the mask to knock out the padding regions, which are of suspect quality

Segment Padding

Various level-2 algorithms have spatial dependencies, which would result in edge artifacts when the image segments are stitched back together. We mitigate this by...

- Defining each segment to include a buffer along the interior edges
- Create a mask defining the buffer region
- Stitch together the image segments, including the extra padding
- Use the mask to knock out the padding regions, which are of suspect quality



Subsetting

Users may not be interested in processing a complete full-disk, so if you know your locations longitude and latitude, you can process just that location and surrounding area. . .

- Singapore
- Tokyo
- Oahu

Subsetting

Users may not be interested in processing a complete full-disk, so if you know your locations longitude and latitude, you can process just that location and surrounding area. . .

```
geocat_l2.sh ...
```

- Singapore
- Tokyo
- Oahu

Subsetting

Users may not be interested in processing a complete full-disk, so if you know your locations longitude and latitude, you can process just that location and surrounding area…

```
geocat_l2.sh ... --subset-lat0 $lat
```

- Singapore
- Tokyo
- Oahu

Subsetting

Users may not be interested in processing a complete full-disk, so if you know your locations longitude and latitude, you can process just that location and surrounding area. . .

```
geocat_l2.sh ... --subset-lat0 $lat
      --subset-lon0 $lon
```

- Singapore
- Tokyo
- Oahu

Subsetting

Users may not be interested in processing a complete full-disk, so if you know your locations longitude and latitude, you can process just that location and surrounding area...

```
geocat_l2.sh ... --subset-lat0 $lat
     --subset-lon0 $lon
--subset-radius 2.0
```

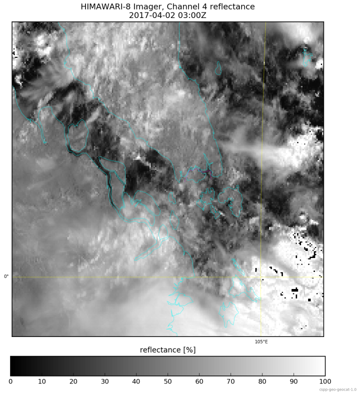
- Singapore
- Tokyo
- Oahu

Subsetting

Users may not be interested in processing a complete full-disk, so if you know your locations longitude and latitude, you can process just that location and surrounding area. . .

```
geocat_l2.sh ... --subset-lat0 $lat
      --subset-lon0 $lon
--subset-radius 2.0
```

- Singapore
- Tokyo
- Oahu

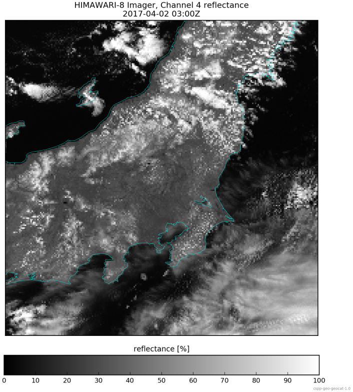


Subsetting

Users may not be interested in processing a complete full-disk, so if you know your locations longitude and latitude, you can process just that location and surrounding area. . .

```
geocat_l2.sh ... --subset-lat0 $lat
     --subset-lon0 $lon
--subset-radius 2.0
```

- Singapore
- Tokyo
- Oahu

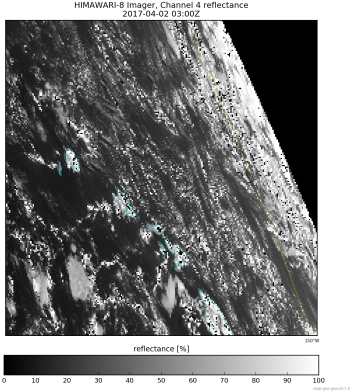


Subsetting

Users may not be interested in processing a complete full-disk, so if you know your locations longitude and latitude, you can process just that location and surrounding area...

```
geocat_l2.sh ... --subset-lat0 $lat
      --subset-lon0 $lon
--subset-radius 2.0
```

- Singapore
- Tokyo
- Oahu



CSPP GEO-Geocat v1.0

Milestones, things to do...

- Version 1.0 should be released shortly
- More level-2 algorithms
- Himawari-9 support

CSPP GEO-Geocat v1.0

Milestones, things to do...

- Version 1.0 should be released shortly
- More level-2 algorithms
- Himawari-9 support

CSPP GEO-Geocat v1.0

Milestones, things to do. . .

- Version 1.0 should be released shortly
- More level-2 algorithms
- Himawari-9 support