# Polar2Grid

## Version 2.0

By David Hoese

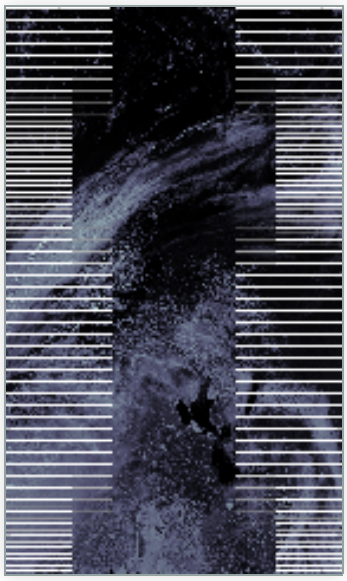
[Switch Scene]

# Outline

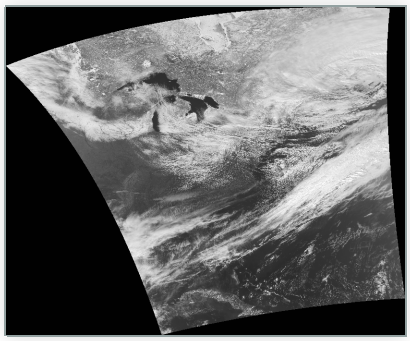
- What is Polar2Grid?
- History and Rewriting
- Features and Examples
- As a Tool and Library
- Future Plans

[Swath Scene]

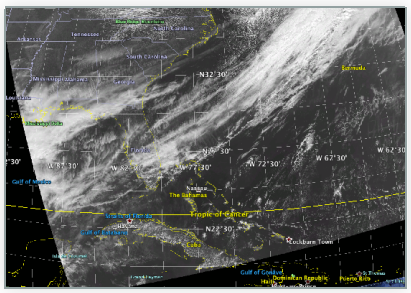
# What is Polar2Grid?



→



→



[Swath Scene]

# What is Polar2Grid? - Point A to Point B

- swath image data
- longitudes
- latitudes

→

- gridded image data
- projection definition
- width
- height
- cell width
- cell height
- upper-left X coordinate
- upper-left Y coordinate

[Switch Scene]

# What is Polar2Grid? - Installation

```
tar -xzf polar2grid-2.0.0.tar.gz
```

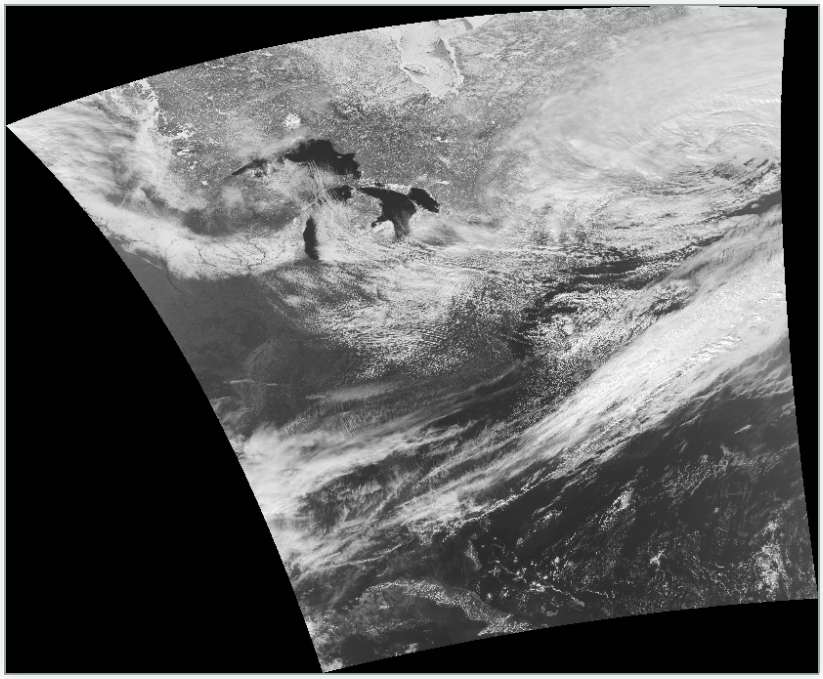
```
viirs2gtiff.sh ...
```

```
p2g_glue viirs gtiff ...
```

[Swath Scene]

# What is Polar2Grid? - 3 Main Steps

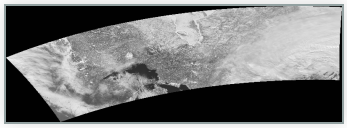
Frontend → Remapper → Backend

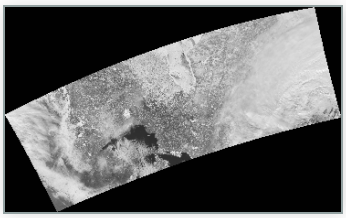


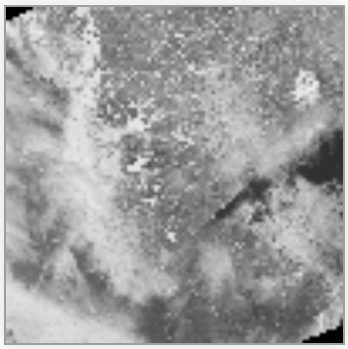
```
viirs2gtiff.sh -f /path/to/files -p i01
# 52 seconds - 7 granules
```

[Swath Scene]

# What is Polar2Grid? - Projections

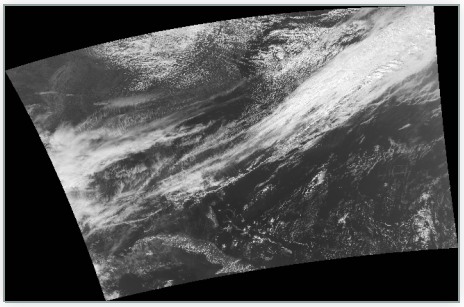
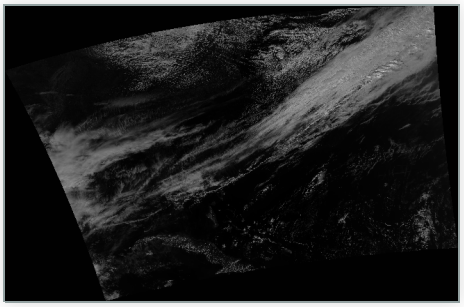
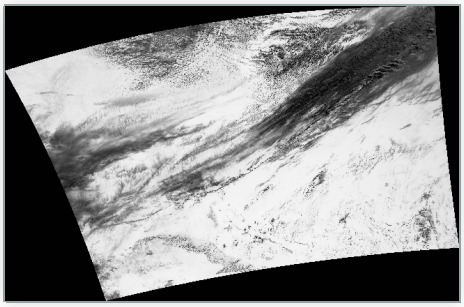






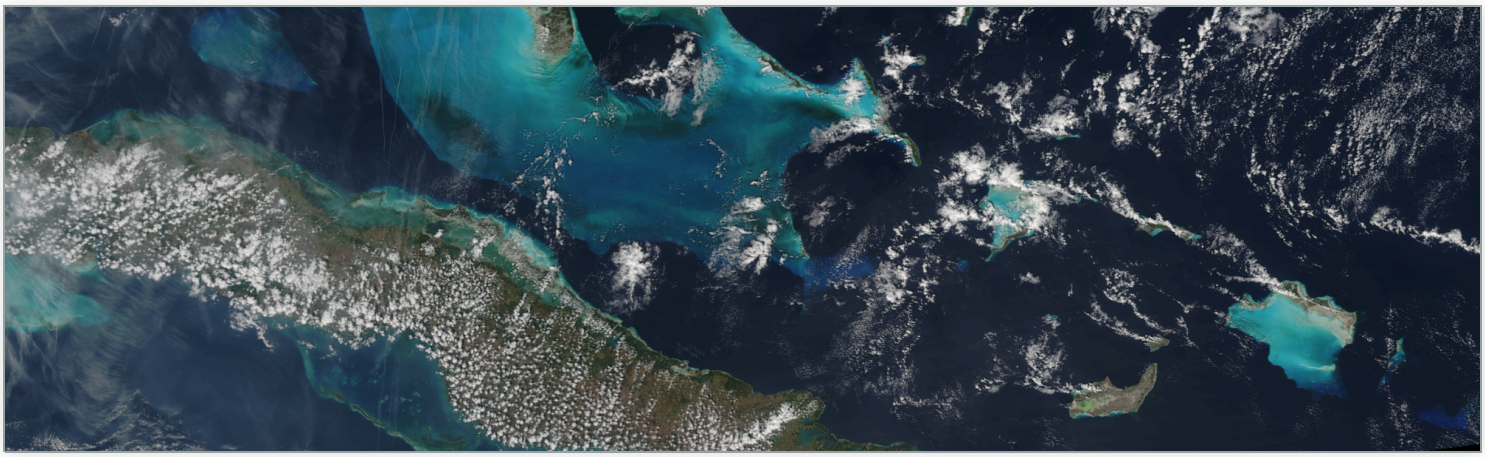
[Swath Scene]

# What is Polar2Grid? - Rescaling

[Switch Scene]

# What is Polar2Grid? - RGB



[Swath Scene]

# Polar2Grid: The Early Years

- viirs2awips → polar2awips → polar2grid
- New Features → Ugly Code
- Premature Optimization → Ugly Code
- Ugly Code → 😢

[Swath Scene]

# Rewriting Polar2Grid: Scenes and Products

Frontend → Remapping → Backend

Scenes and Products

Product Name
Binary Data
Satellite
Instrument
Begin Time
End Time
Data Kind
...

[Swath Scene]

# Rewriting Polar2Grid: Compositors

Frontend → Remapping → Backend

[Swath Scene] → Remapping

[Gridded Scene] → Backend

[Gridded Scene] → Compositor → [Gridded Scene]

[Switch Scene]

# Rewriting Polar2Grid: Configuration

```
npp, viirs, m_nav, m, 01, reflectance, sqrt, 100.0, 25.5
```

```
[rescale:m01]
product_name=m01
method=sqrt
```

```
[rescale:default_reflectance]
data_kind=reflectance
method=sqrt
```

```
[rescale:avhrr_reflectance_100]
data_kind=reflectance
instrument=avhrr
method=sqrt
max_in=100.0
```

[Swath Scene]

# Rewriting Polar2Grid: Remapping

- MODIS Swath To Grid Toolbox (MS2GT)
  - National Snow and Ice Data Center
  - ll2cr for gridding
  - fornav for interpolation
- Elliptical Weighted Averaging vs Others
- PROJ.4 vs MapX
- C vs Python vs Cython

[Swath Scene]

# Features: Remapping

- EWA vs Nearest Neighbor
- Builtin Grids
- User Grids
- Static vs Dynamic Grids

[Switch Screen]

# Features: Grid Configurations

```
211e, proj4,
+proj=lcc +a=6371200 +b=6371200 +lat_0=25 +lat_1=25 +lon_0=-95 +units=m +no_defs,
5120, 5120, 1015.9, -1015.9, -123.044deg, 59.844deg
```

```
lcc_fit, proj4,
+proj=lcc +datum=WGS84 +ellps=WGS84 +lat_0=25 +lat_1=25 +lon_0=-95 +units=m +no_defs,
None, None, 1000, -1000, None, None
```

```
p2g_grid_helper.sh grid_name -95 45 1000 -1000 5000 5000
```

[Switch Scene]

# Features: Backends

- Geotiff
  - PNG Copies
  - KMZ Files
  - Compression
- AWIPS II (replaces AWIPS I)
- NinJo
- Flat Binary
- HDF5

[Swath Scene]

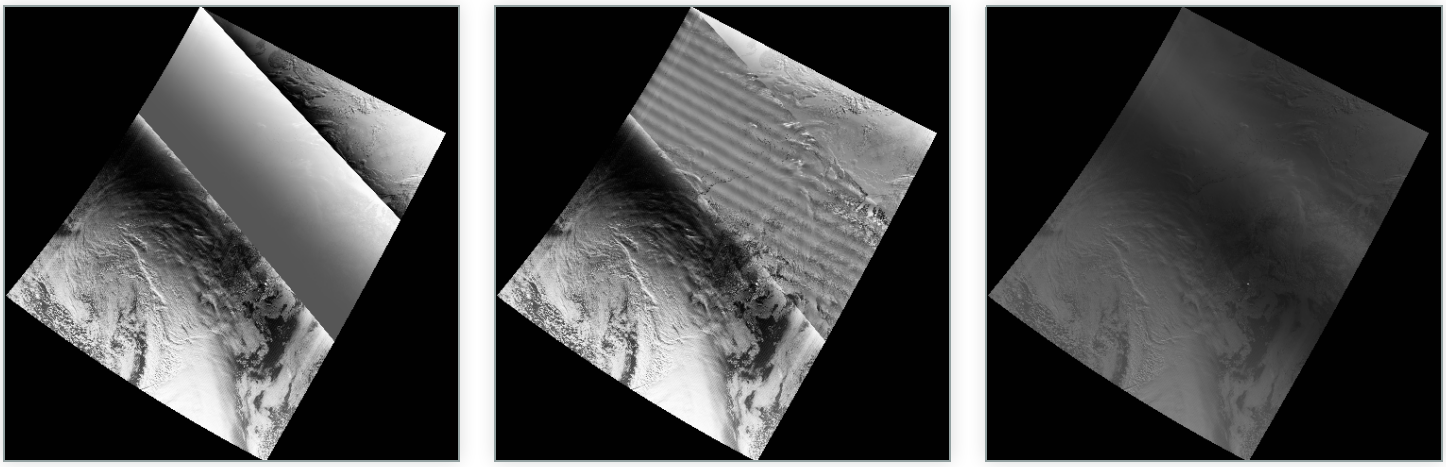
# Features: Frontends

- VIIRS SDRs
- MODIS L1B
- Corrected Reflectance (CREFL)

- Advanced Clear-Sky Processor for Oceans (ACSPO)
- Microwave Integrated Retrieval System (MIRS)
- Dual Regression Retrieval (DR-RTV): AIRS, CrIS, IASI
- AVHRR L1B

[Swath Scene]

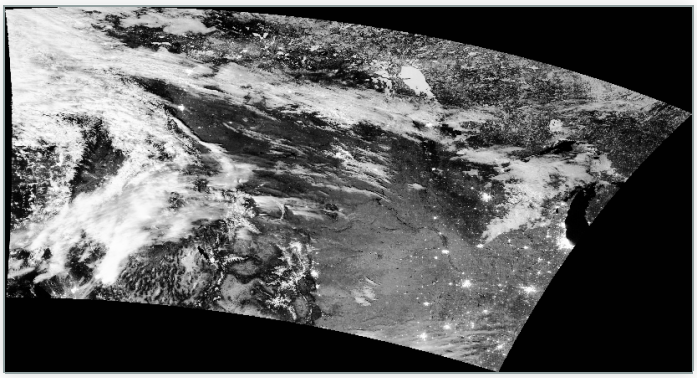
# Features: Frontends - VIIRS DNB

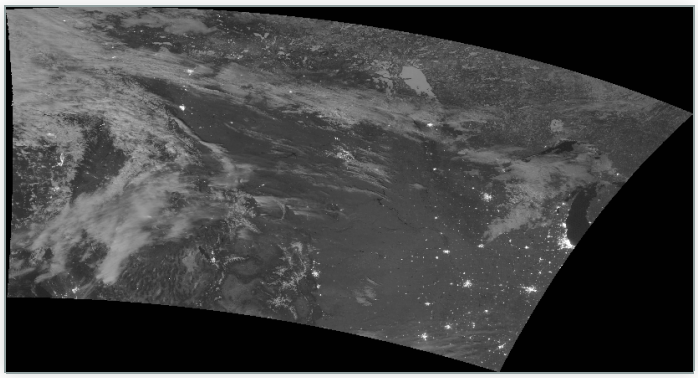


```
viirs2gtiff.sh -f /path-to-files/ -g 203 -p histogram_dnb
    adaptive_dnb dynamic_dnb
```

[Switch Scene]

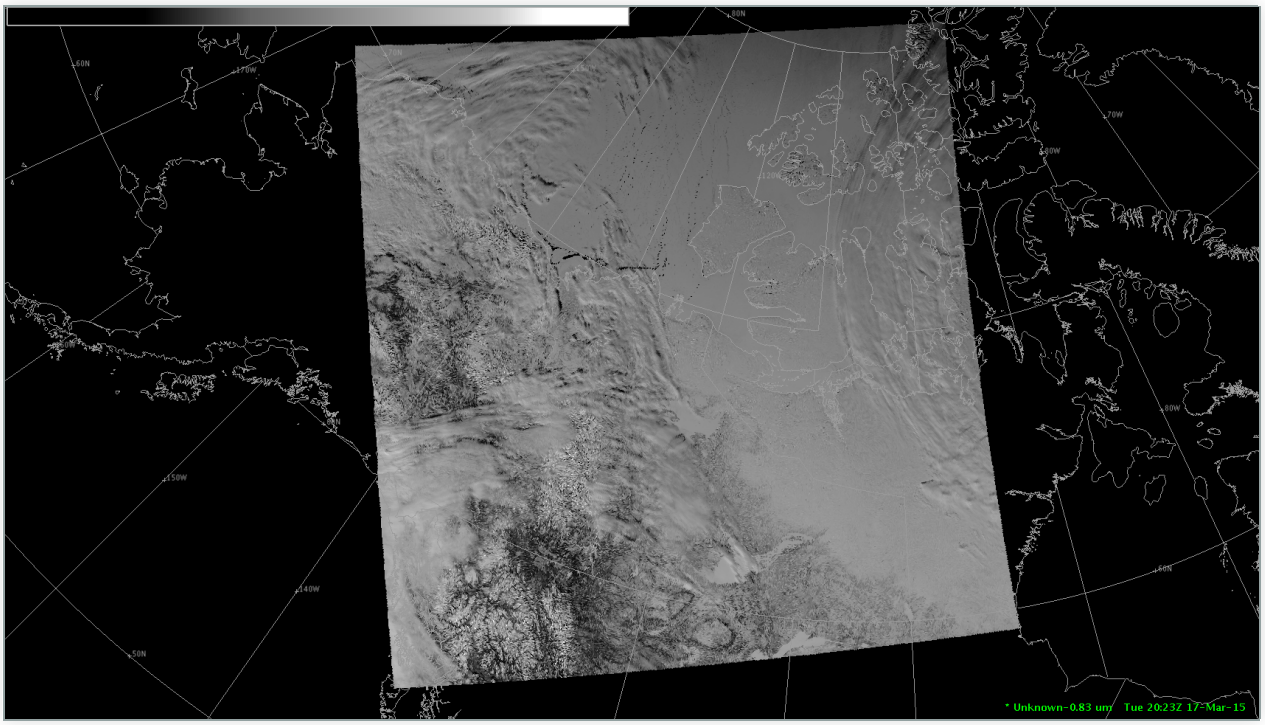
# Features: Frontends - VIIRS DNB Night





[Swath Screenshot]

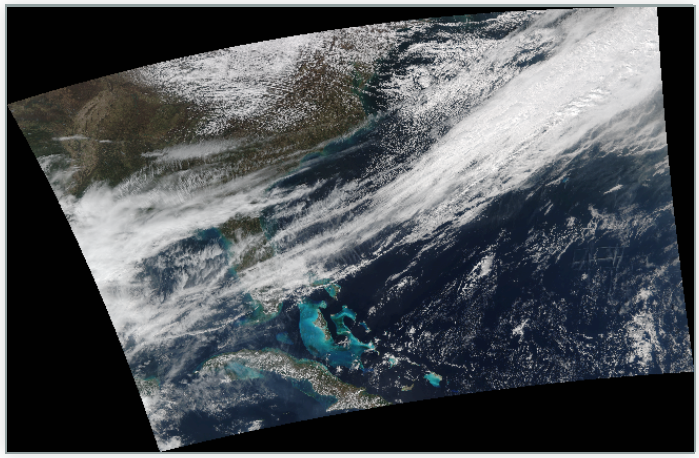
# Features: Frontends - MODIS

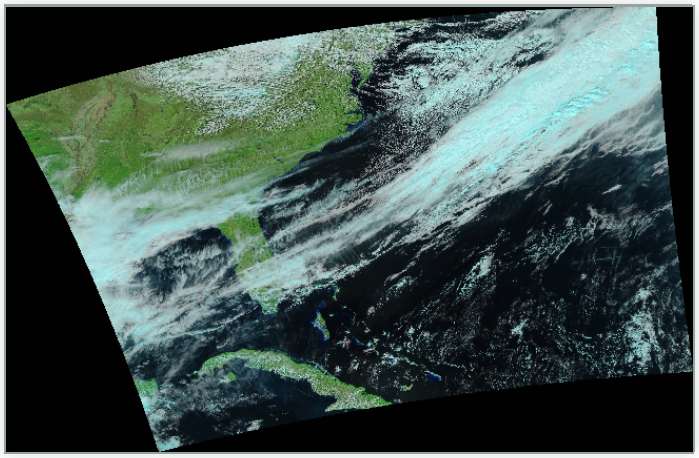



```
modis2awips.sh -f /path-to-files/ -p visible_02 -g 203
```

[Swath Scene]

# Features: Frontends - CREFL





[Switch to Screen]

# Features: Frontends - CREFL (cont.)

```
crefl2gtiff.sh -f /path-to-files/
```

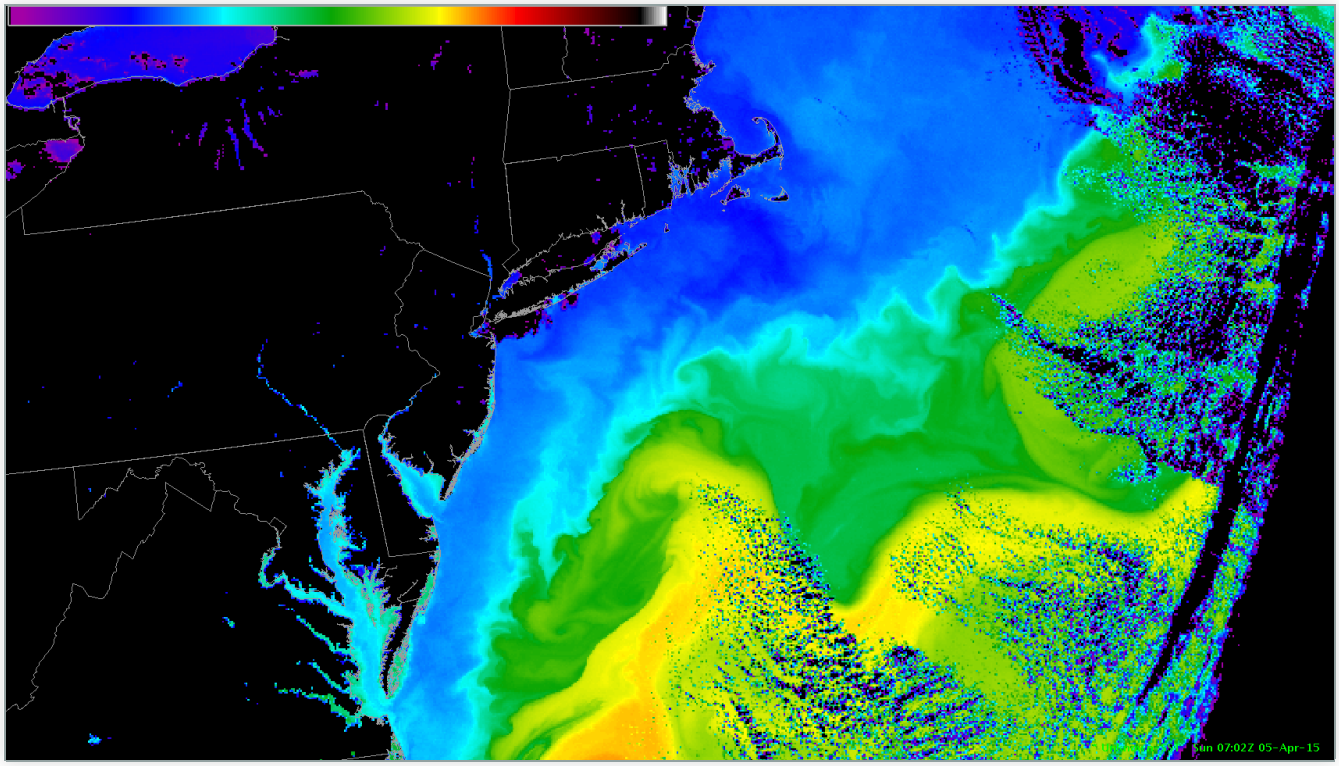
```
p2g_glue crefl gtiff true_color -f /path-to-files/
```

```
p2g_glue crefl gtiff false_color -f /path-to-files/ --false-color
```

```
p2g_glue crefl gtiff true_color false_color -f /path-to-files/
    --true-color --false-color
```

[Swath Screenshot]

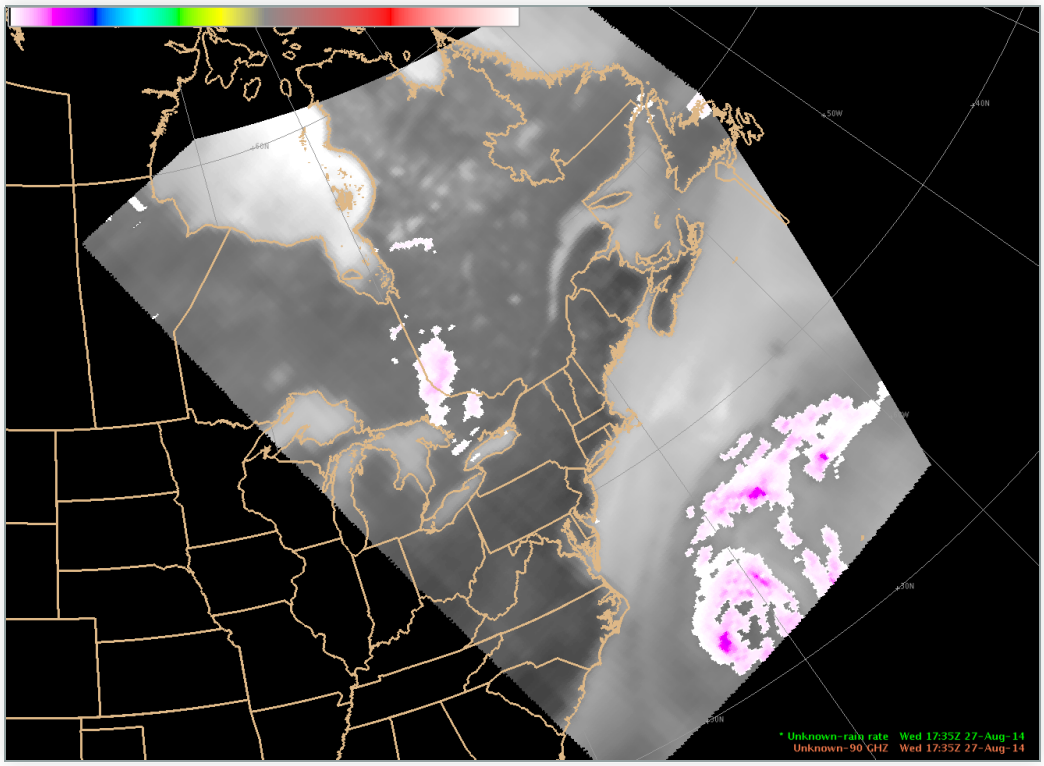
# Features: Frontends - ACSPO




```
acspo2awips.sh -g 211e -f /path-to-files/
```

[Swath Screenshot]

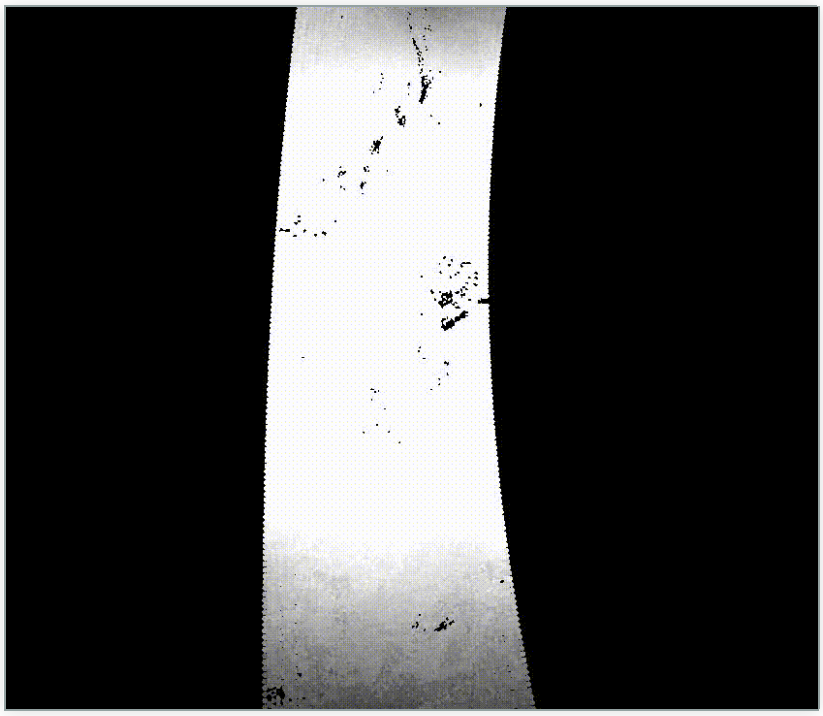
# Features: Frontends - MIRS




```
mirs2awips.sh -f /path-to-files/ -g 211e
```

[Switch Scene]

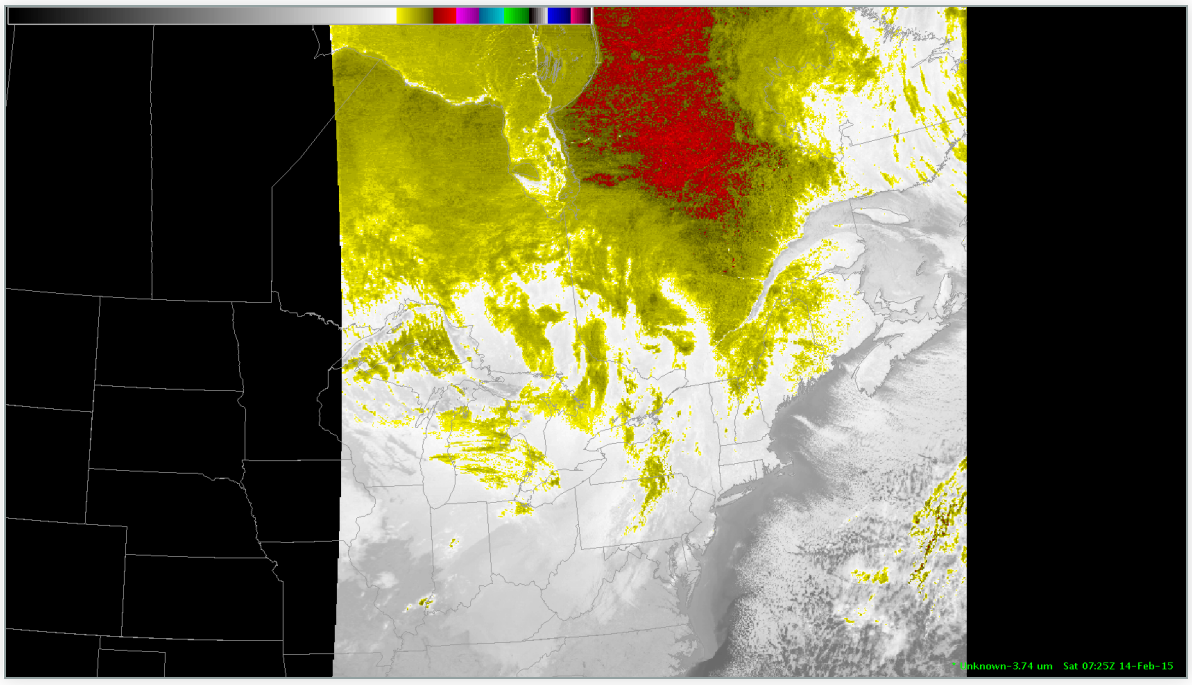
# Features: Frontends - DR-RTV



```
drrtv2gtiff.sh -f /path-to-files/ -g 203_10km
```

[Swath Scene]

# Features: Frontends - AVHRR




```
avhrr2awips.sh -f /path-to-files/ -g 211e -p band2_vis band3b_bt
```

[Swath Scene]

# Glue Scripts

```
viirs2gtiff.sh -g grid_name -f /path-to-files/
```

viirs2awips.sh
viirs2binary.sh
viirs2hdf5.sh
viirs2ninjo.sh
modis2awips.sh
modis2binary.sh
modis2gtiff.sh
modis2hdf5.sh
avhrr2awips.sh
avhrr2binary.sh
avhrr2gtiff.sh
avhrr2hdf5.sh
mirs2awips.sh
mirs2binary.sh
mirs2gtiff.sh
mirs2hdf5.sh
acspo2awips.sh
acspo2binary.sh
acspo2gtiff.sh
acspo2hdf5.sh
drrtv2awips.sh
drrtv2binary.sh
drrtv2gtiff.sh
drrtv2hdf5.sh
crefl2awips.sh
crefl2binary.sh
crefl2gtiff.sh
crefl2hdf5.sh
crefl2ninjo.sh

[Switch Scene]

# As a Tool and Library

```
p2g_glue <frontend> <backend> ...
```

```
p2g_frontend <frontend> ...
```

```
p2g_remap ...
```

```
p2g_backend <backend> ...
```

[Swath Scene]

# As a Tool and Library - Example 1


```
{
    "fancy_dnb": {
        "product_name": "fancy_dnb",
        "satellite": "npp",
        "instrument": "viirs",
        "begin_time": "2012-02-25T18:01:24.570942",
        "end_time": "2012-02-25T18:07:04.961141",
        "data_kind": "equalized_radiance",
        "swath_data": "fancy_dnb.dat",
        "data_type": "real4",
        "fill_value": NaN,
        "swath_definition": {
            "swath_name": "dnbnav",
            "latitude": "dnb_latitude.dat",
            "longitude": "dnb_longitude.dat",
            "data_type": "real4",
            "fill_value": NaN,
            "rows_per_scan": 16,
            "swath_columns": 4064,
            "swath_rows": 3072
        }
    }
}
```

[Swath Scene]

# As a Tool and Library - Example 1 (cont.)

Remap the data:

```
p2g_remap -vvv --scene fancy_dnb_swath.json -o fancy_dnb_grid.json
    --method=ewa --fornav-D=40 --fornav-d=2
```

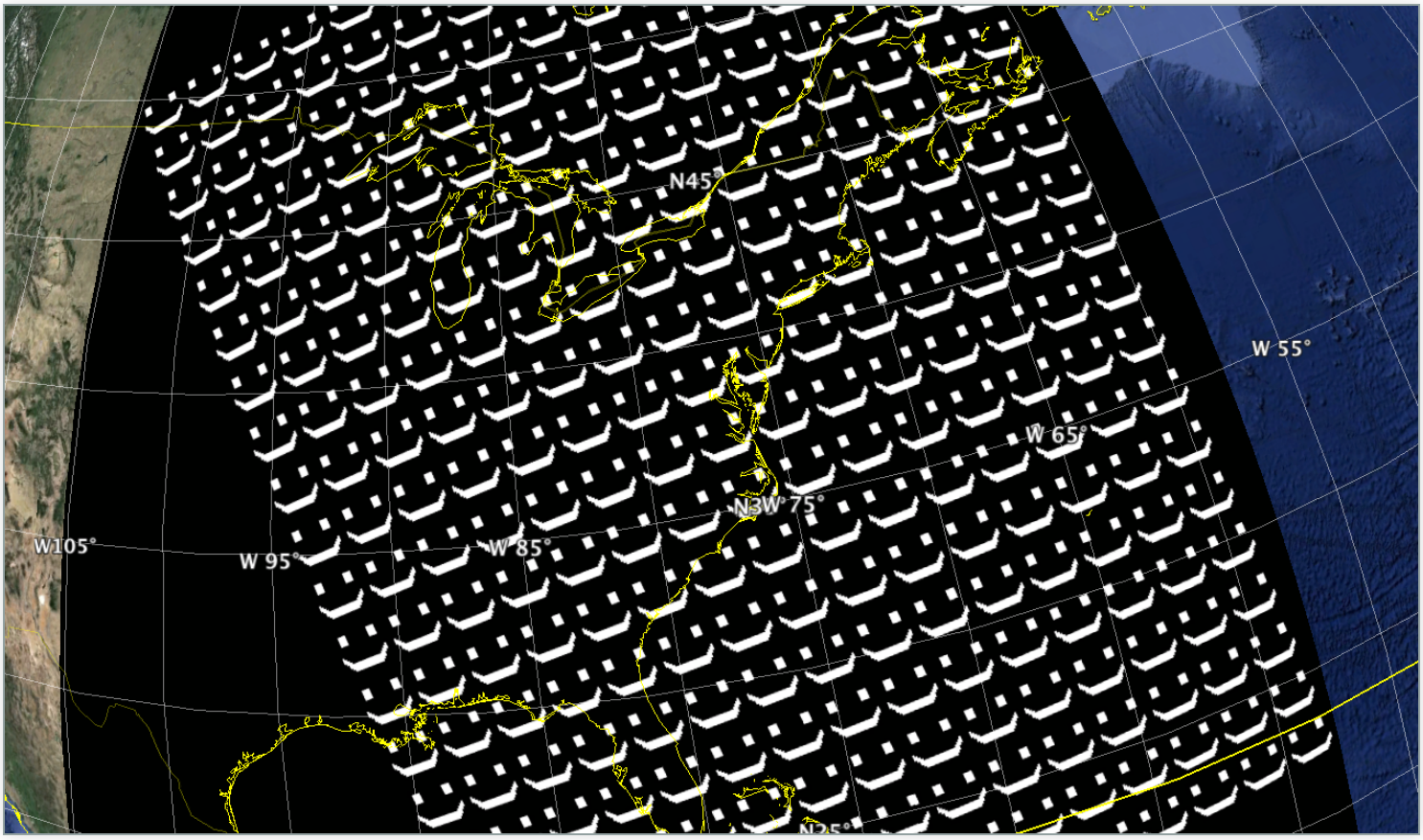
Create geotiff:

```
p2g_backend gtiff -vvv --scene fancy_dnb_grid.json
```

```
ls *.tif
    npp_viirs_fancy_dnb_20120225_180124_wgs84_fit.tif
```

[Switch Screen]

# As a Tool and Library - Example 1 (cont.)



[SwathScene]

# As a Tool and Library - Example 2

```
from polar2grid.viirs import Frontend
from polar2grid.remap import Remapper
from polar2grid.gtiff_backend import Backend

def create_fancy_dnb(hist_dnb_product):
    ...

frontend = Frontend(search_paths=["/no_backup/data/viirs/conus_day/"])
remapper = Remapper()
backend = Backend()

swath_scene = frontend.create_scene(products=["histogram_dnb"])
swath_scene["fancy_dnb"] = create_fancy_dnb(swath_scene["histogram_dnb"])
del swath_scene["histogram_dnb"]

gridded_scene = remapper.remap_scene(swath_scene,
                        "wgs84_fit", fornav_D=40, fornav_d=2)

filenames = backend.create_output_from_scene(gridded_scene)
```

[Switch Scene]

[Swath Scene]

# Polar2Grid and PyTroll

- What's the difference?
- 2 solutions, 1 problem
- 2 competing projects...that were collaborating
  - NinJoTIFFs
  - AVHRR
- Where do we go from here?
- Redesign, Collaboration, New Features

[Swath Scene]

# Questions?

Github: http://github.com/davidh-ssec/polar2grid/
Documentation: http://www.ssec.wisc.edu/software/polar2grid/

Contributors

Ray Garcia
Jordan Gerth
Liam Gumley
Katja Hungerschöfer
Scott Macfarlane
Willem Marais
Eva Schiffer
Kathy Strabala
William Straka