SatPy

A Python Library for Weather Satellite Imagery

By David Hoese *pronounced Haze

SatPy Core Developers

- Martin Raspaud
- Adam Dybbroe
- Panu Lahtinen

Thank you

- PyTroll Users
- Kathy Strabala

About Me

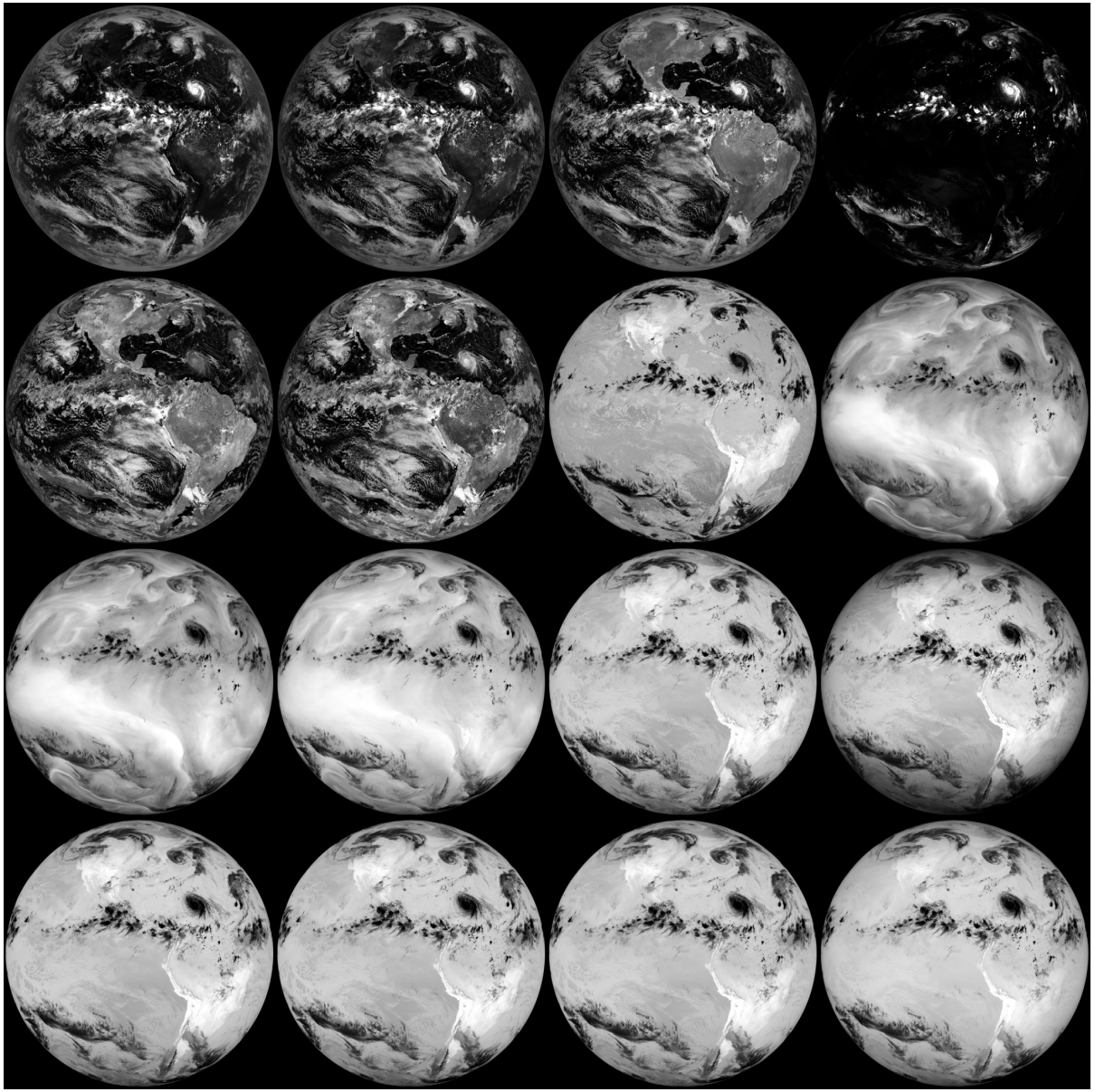
- Github: djhoese
- Twitter: djhoese
- Software Developer @ Space Science and Engineering Center (SSEC) - University of Wisconsin - Madison
- Community Satellite Processing Package (CSPP) Team
- Polar2Grid/Geo2Grid
- Satellite Information Familiarization Tool (SIFT) - Poster
- VisPy Maintainer - Plenary
- PyTroll/SatPy Maintainer

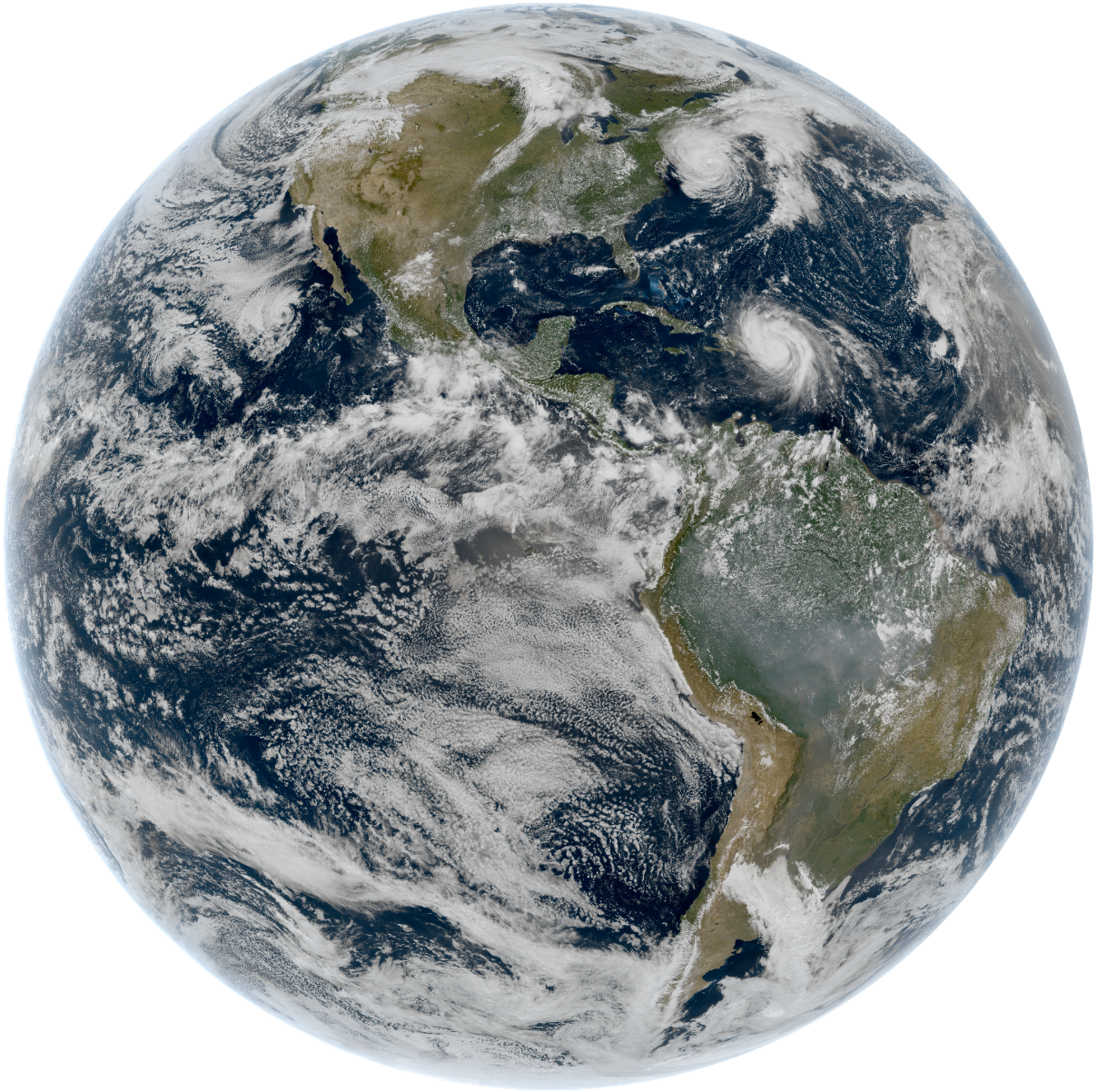
Outline

1. What do we work with?
2. How does SatPy make our work easier?
3. Cool stuff you can do with SatPy

What do we work with? Weather Satellite Data

- Forecasting and forecast models
- Research/Analysis
- Pretty pictures
- Operational processing





Our solution: SatPy

- Version 0.9 released
- Linux, OSX, Windows compatible
- Installable with `pip` and `conda` (conda-forge)

How does SatPy make our work easier?

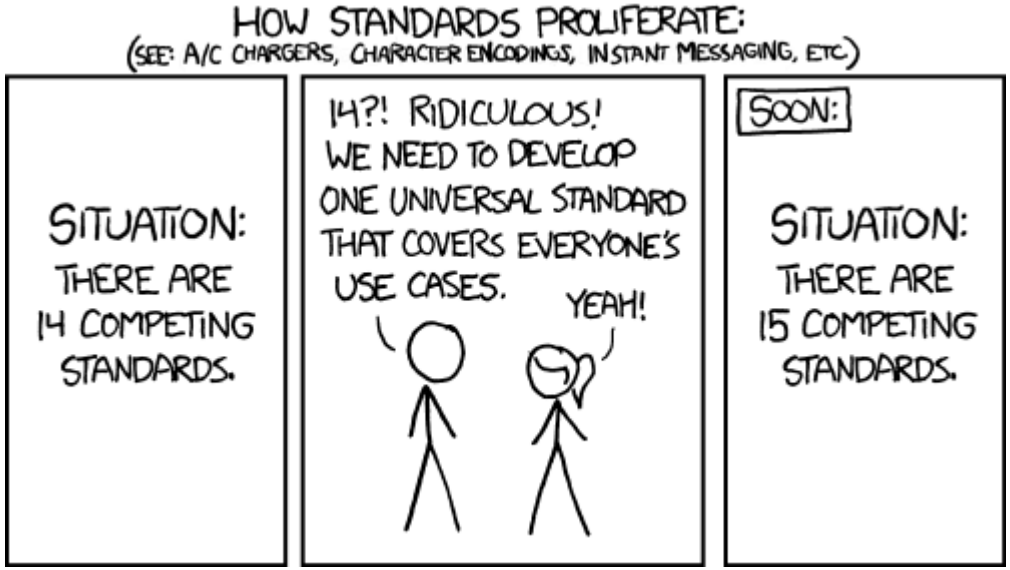
- Common interfaces
- Performance - `xarray` and `dask`
- Community

Common Interfaces

- Read
- Composite/Modify
- Resample
- Write

Read: How do I get the information I need?

- Reading satellite data is annoying
  - Formats: NetCDF, HDF5, custom binary, compressed
  - File "schemes" and organization
  - Fill values
  - Concatenation
  - Missing metadata
  - Many data variations
  - Standards



Loading data and the Scene object

- By Name
- By Wavelength
- By `DatasetID`
- Specifying variations
  - resolution
  - polarization
  - calibration - counts, radiance, reflectance, brightness temperature, etc.

In [32]:
```
from satpy import Scene, DatasetID
from glob import glob

scn = Scene(reader='abi_l1b', filenames=glob('/data/data/abi/20170920/*.nc'))
scn.load(['C01', 'C02'])
```

In [33]:
```
scn.load([0.46, 0.62])
```

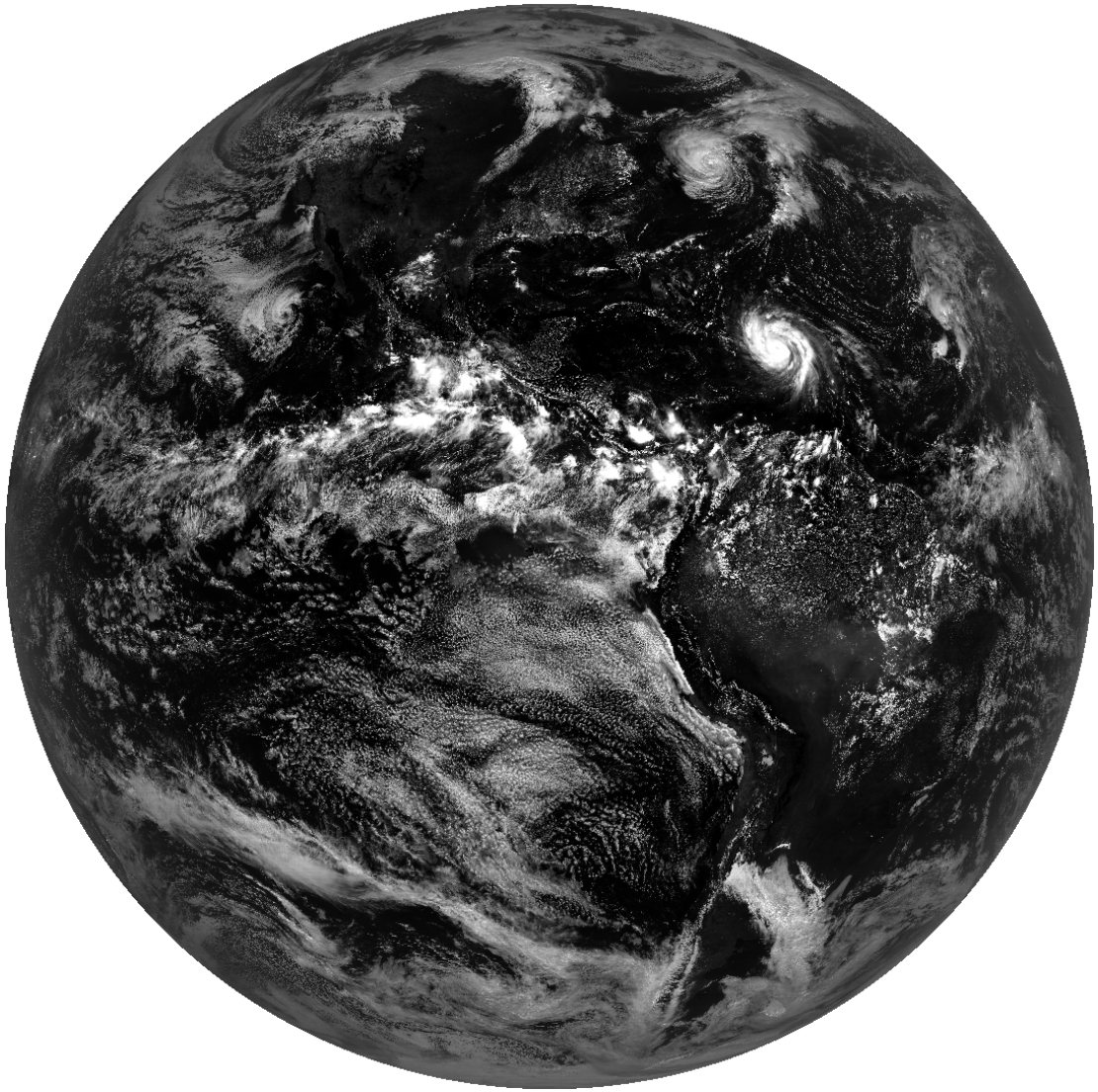
In [34]:
```
ds_id = DatasetID(name='C02', calibration='radiance')
scn.load([ds_id])
```

In [29]:
```
scn['C02']
```

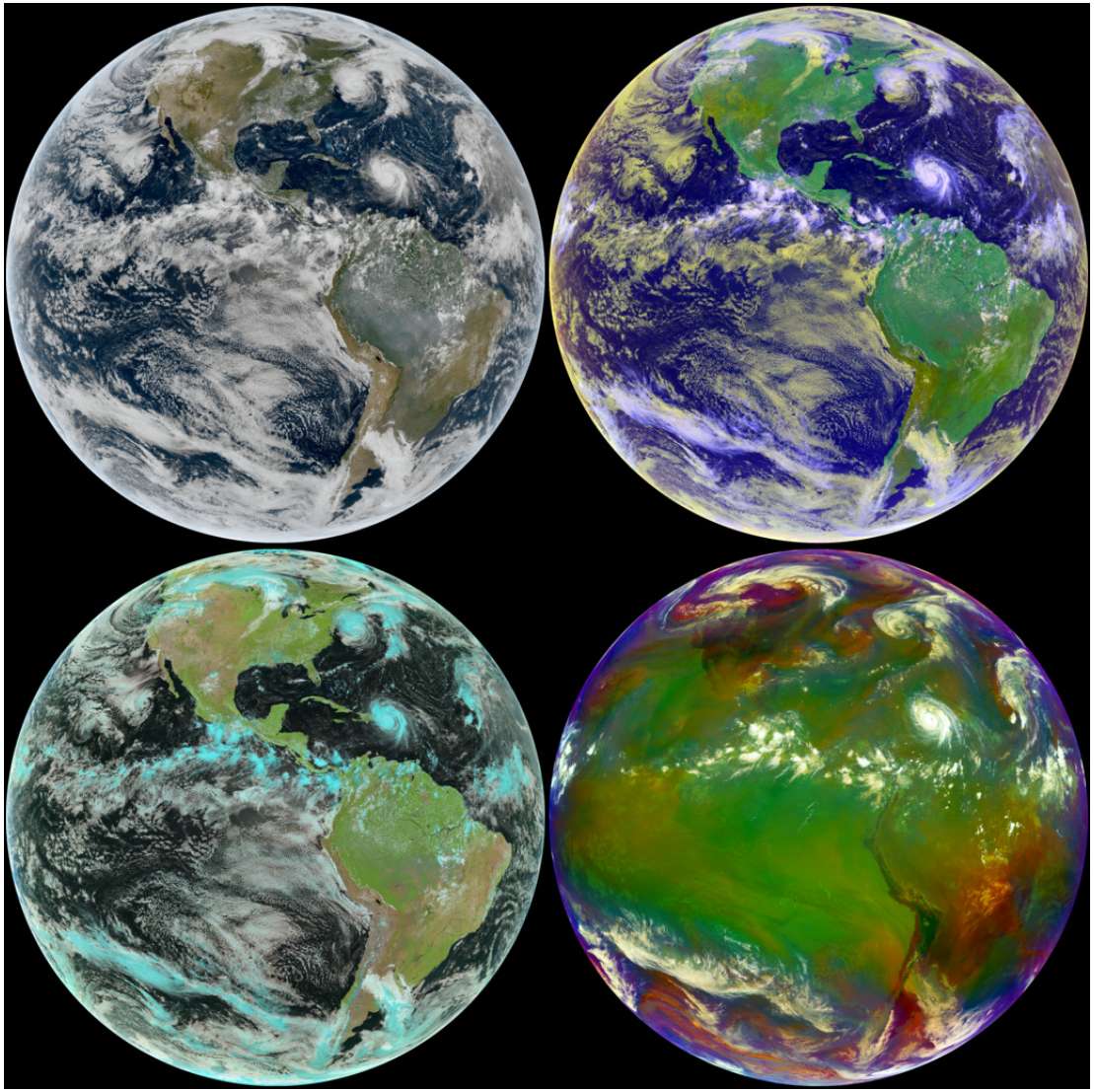
Out[29]:
```
<xarray.DataArray (y: 21696, x: 21696)>
dask.array<shape=(21696, 21696), dtype=float64, chunksize=(4096, 4096)>
Coordinates:
  * x        (x) float64 -0.1519 -0.1519 -0.1518 -0.1518 -0.1518 -0.1518 ...
    y_image  float32 0.0
    x_image  float32 0.0
  * y        (y) float64 0.1519 0.1519 0.1518 0.1518 0.1518 0.1518 0.1518 ...
Attributes:
    satellite_longitude:    -89.5
    satellite_latitude:     0.0
    satellite_altitude:     35786.0234375
    units:                  %
    modifiers:              ()
    wavelength:             (0.59, 0.64, 0.69)
    grid_mapping:           goes_imager_projection
    _Unsigned:              true
    cell_methods:           t: point area: point
    standard_name:          toa_bidirectional_reflectance
    scale_factor:           0.158592
    ancillary_variables:    []
    valid_range:            [   0 4094]
    long_name:              ABI L1b Radiances
    add_offset:             -20.2899
    resolution:             500
    calibration:            reflectance
    _FillValue:             4095
    sensor:                 abi
    sensor_band_bit_depth:  12
    name:                   C02
    platform_name:          GOES-16
    start_time:             2017-09-20 17:30:40.800000
    end_time:               2017-09-20 17:41:17.500000
    area:                   Area ID: abi_geos\nDescription: ABI L1B file are
a...
```

```
polarization:           None
level:                  None
```

In [14]:
```
scn.show('C01')
```



Composite: How do I make pretty color pictures?

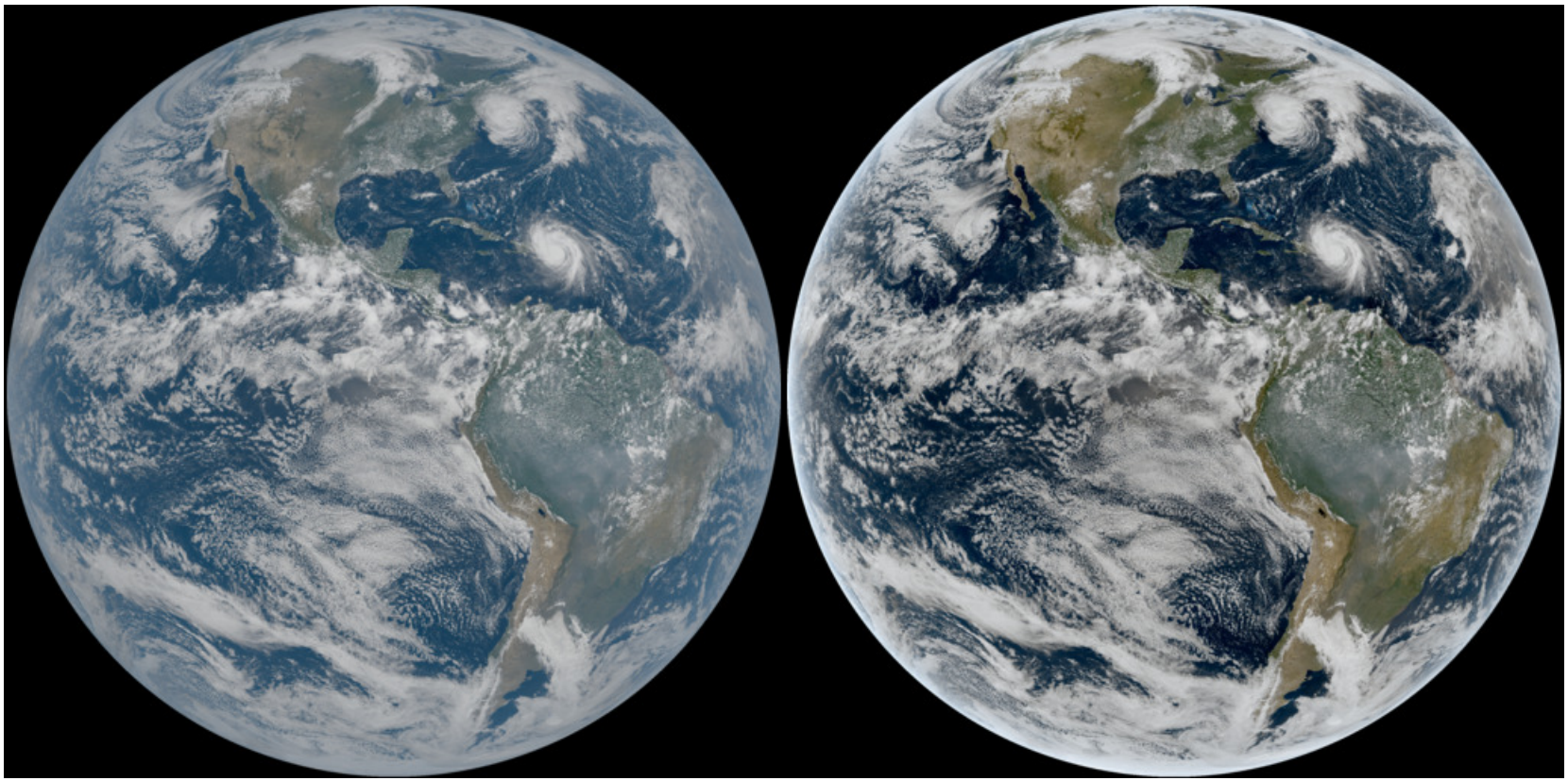


In [2]:
```
scn.load(['airmass'])
scn['airmass']
```

Out[2]:
```
<xarray.DataArray 'concatenate-d3ffb434ba4f6164ba94e209a0860c74' (bands: 3, y:
5424, x: 5424)>
dask.array<shape=(3, 5424, 5424), dtype=float64, chunksize=(1, 4096, 4096)>
Coordinates:
  * x        (x) float64 -0.1518 -0.1518 -0.1517 -0.1517 -0.1516 -0.1516 ...
    y_image  float32 0.0
    x_image  float32 0.0
  * y        (y) float64 0.1518 0.1518 0.1517 0.1517 0.1516 0.1516 0.1515 ...
  * bands    (bands) <U1 'R' 'G' 'B'
Attributes:
    long_name:              ABI L1b Radiances
    grid_mapping:           goes_imager_projection
    level:                  None
    satellite_latitude:     0.0
    sensor:                 abi
    satellite_altitude:     35786.0234375
    ancillary_variables:    []
    start_time:             2017-09-20 17:30:40.800000
    cell_methods:           t: point area: point
    polarization:           None
    resolution:             None
    area:                   Area ID: some_area_name\nDescription: On-the-fl
y...
    platform_name:          GOES-16
    _Unsigned:              true
    satellite_longitude:    -89.5
    standard_name:          airmass
    wavelength:             None
    optional_datasets:      []
    end_time:               2017-09-20 17:41:18.600000
    name:                   airmass
    prerequisites:          [6.2, 7.3, 9.7, 10.3]
    optional_prerequisites: []
```

```
calibration:            None
modifiers:              None
mode:                   RGB
```

Modifiers: How do I make my pretty picture prettier?



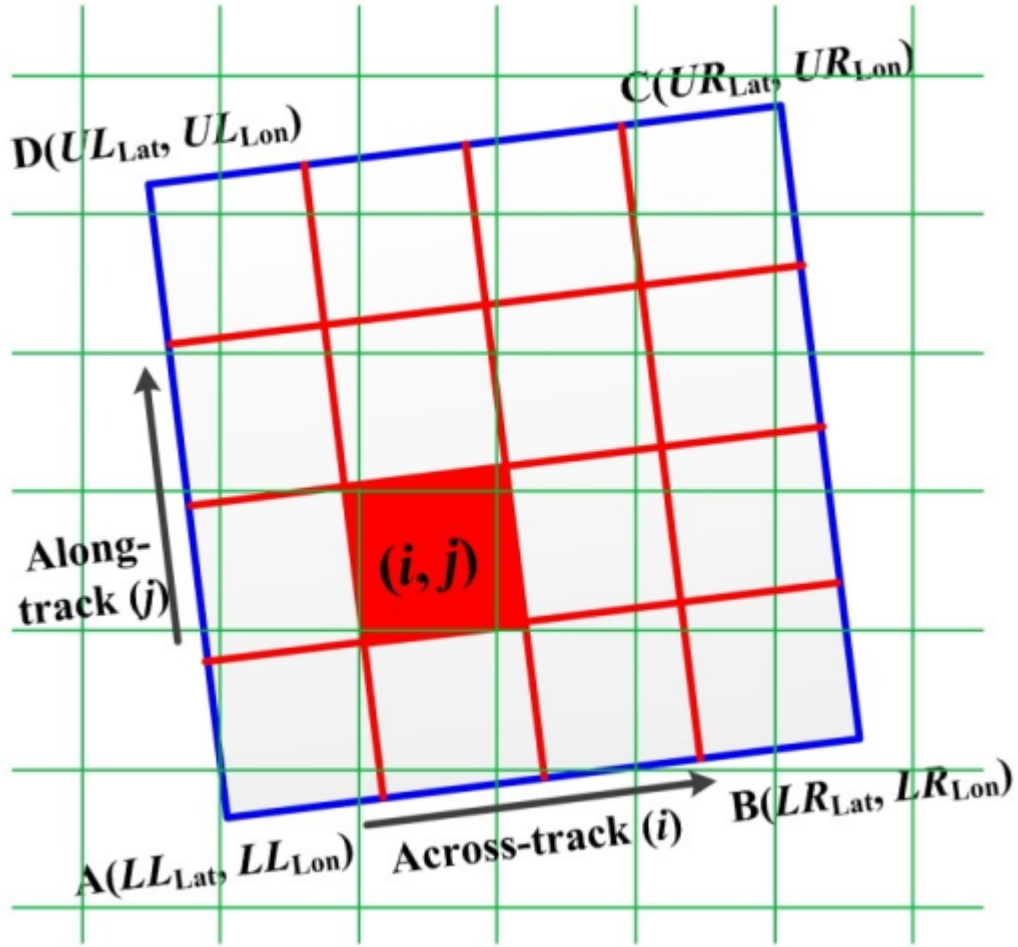
Do the impossible

In [ ]:
```
scn.load(['true_color'])
```

Operations Involved:

- 3 input bands (500m and 1000m)
- 3 complex data enhancements
- 1 pseudo-band (green)
- Upscale to 500m
- Final result: 21696 rows * 21696 cols * 8 bytes * 3 input bands = ~10.5 GB
- Dask-based SatPy loads, computes, and saves to geotiff in 6-8 minutes

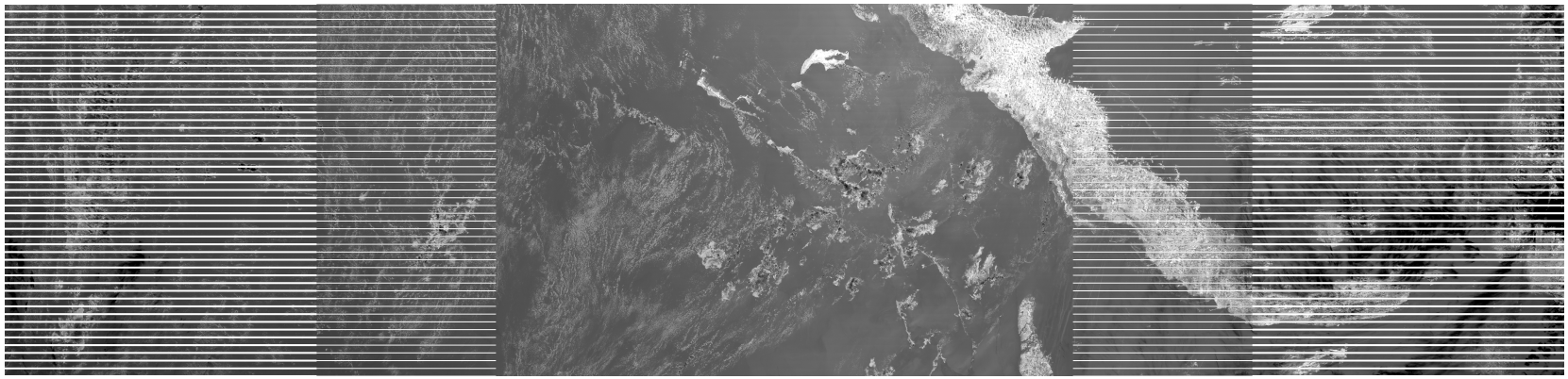
Resampling



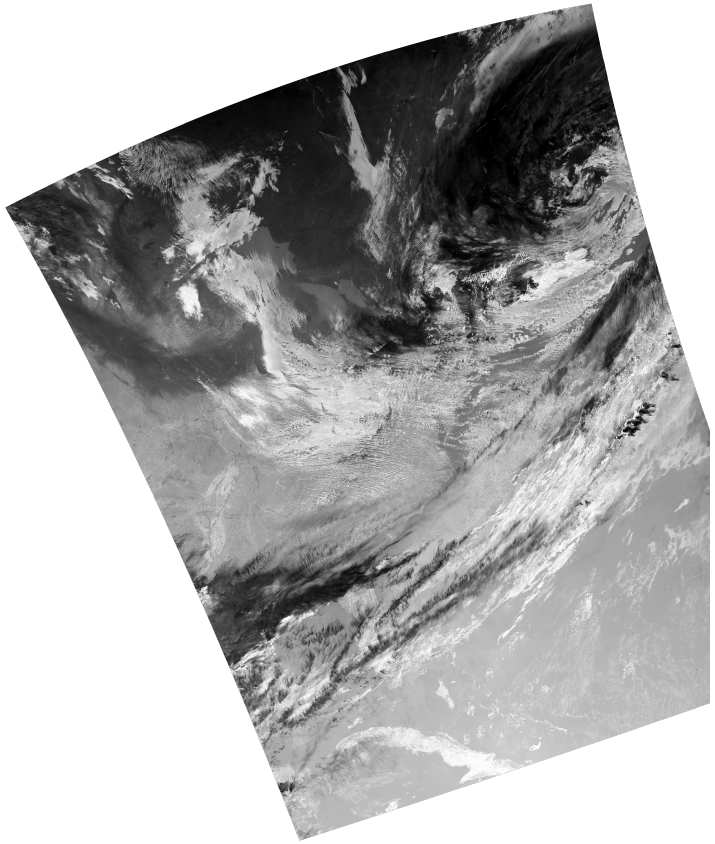
Ref (https://openi.nlm.nih.gov/detailedresult.php?img=PMC4299089_sensors-14-23822f3&req=4)

In [36]:
```
from satpy import Scene
from glob import glob
scn = Scene(reader='viirs_sdr',
            filenames=glob('/data/data/viirs/conus_day/*t1801*.h5'))
scn.load(['I04'])
```

In [ ]:
```
scn.show('I04')
```



In [ ]:
```
scn = Scene(reader='viirs_sdr',
            filenames=glob('/data/data/viirs/conus_day/*t180*.h5'))
scn.load(['I04'])
new_scn = scn.resample('211e')
new_scn.show('I04')
```



Saving to disk

- Writers
  - Geotiff (rasterio)
  - Pillow (png, jpeg, etc)
  - AWIPS SCMI
  - CF-compliant NetCDF (limited projection support)
- Use dask blocks when possible

In [ ]:
```
scn.save_datasets(writer='geotiff', base_dir='/tmp', compress='LZW')
```

The Basics

```
from satpy import Scene
scn = Scene(reader='abi_l1b', filenames=[...])
scn.load(['C01', 'C02', 'C08', 'true_color', 'air_mass', 'natural'])
new_scn = scn.resample(resampler='native')
new_scn.save_datasets(base_dir='/tmp')
```

Fun Stuff: Stacked Scenes

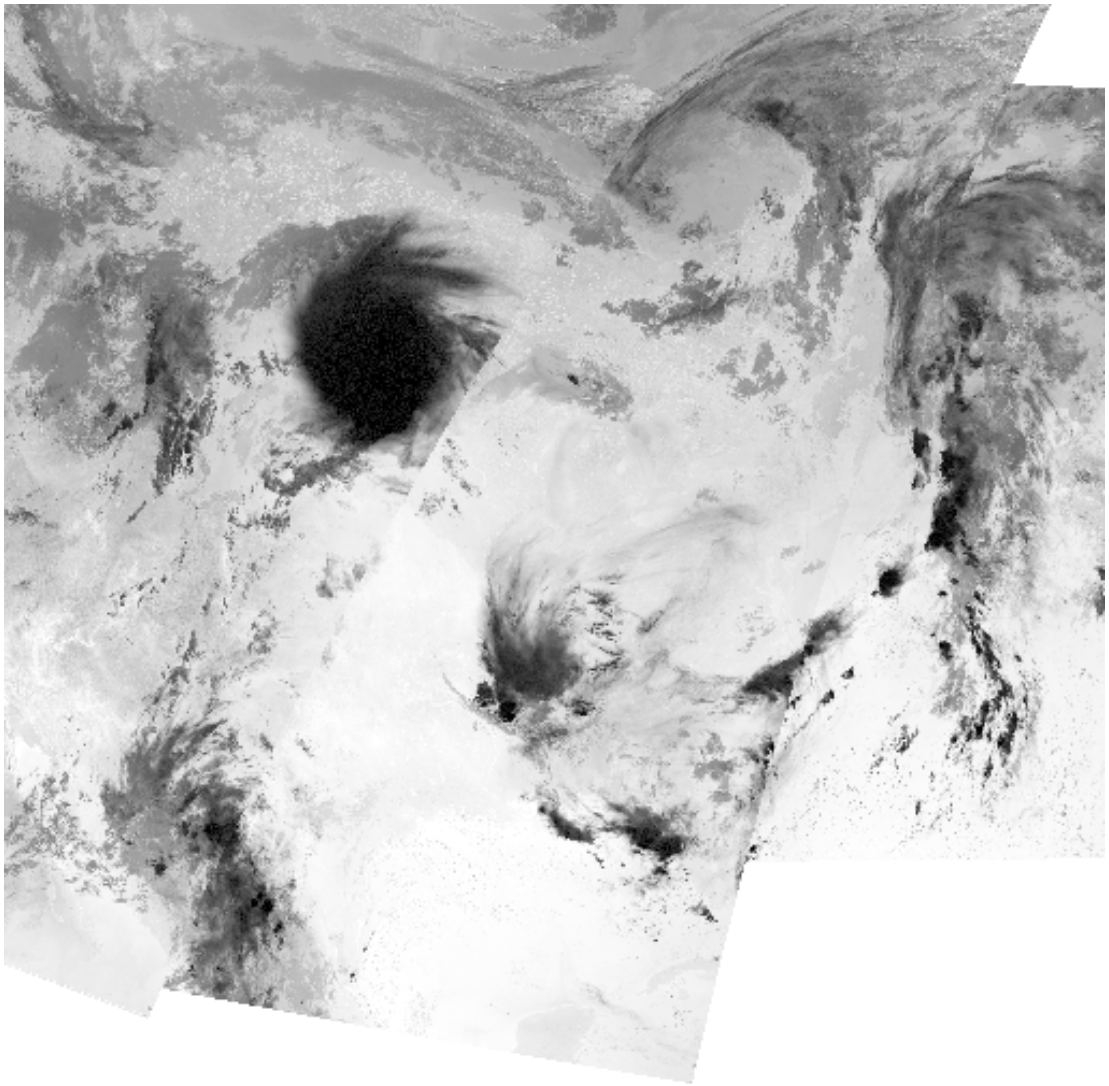
In [2]:
```
from satpy import Scene, MultiScene
from glob import glob
import os
from dask.diagnostics import ProgressBar
base_dir = '/data/data/viirs/2018_06_29/'
viirs_passes = ['2018_06_29_180_0614', '2018_06_29_180_0752', '2018_06_29_180_093
4']
viirs_scenes = [Scene(reader='viirs_sdr',
                      filenames=glob(os.path.join(base_dir, vp, '*.h5')))
                for vp in viirs_passes]

mscn = MultiScene(viirs_scenes)
mscn.load(['I04'])
new_mscn = mscn.resample('211e')
blended_scn = new_mscn.blend()
with ProgressBar():
    blended_scn.save_datasets()
```

```
[########################################] | 100% Completed |  4min 15.4s
[########################################] | 100% Completed |  0.1s
```

In [4]:
```
! gdal_translate -of PNG -outsize 10% 10% I04_20180629_061415{.tif,.png}
```

```
Input file size is 5120, 5120
0...10...20...30...40...50...60...70...80...90...100 - done.
```



Fun Stuff: Animations

```
scenes = [Scene(reader='abi_l1b', filenames=step_files)
          for step_files in grouped_files]
mscn = MultiScene(scenes)
mscn.load(['C01'])
new_mscn = mscn.resample(resampler='native')
new_mscn.save_animation('{name}_{start_time:%Y%m%d_%H%M%S}.mp4',
                        fps=5, batch_size=4,
                        output_params=["-vf", "format=yuv420p,scale=640:-1"])
```

In [5]:
```
from IPython.display import HTML
HTML("""
<video width="640" height="480" controls>
  <source src="C01_20180623_000045.mp4" type="video/mp4">
</video>
""")
```

Out[5]:

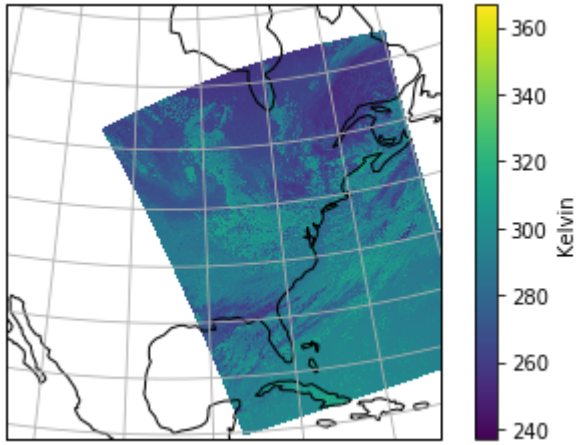
Fun Stuff: Cartopy

In [ ]:
```
from satpy import Scene
from glob import glob
import matplotlib.pyplot as plt

filenames = glob('/data/data/viirs/conus_day/*t180*.h5')
scn = Scene(reader='viirs_sdr', filenames=filenames)
scn.load(['I04'])
new_scn = scn.resample('211e')

crs = new_scn['I04'].attrs['area'].to_cartopy_crs()
ax = plt.axes(projection=crs)

ax.coastlines()
ax.gridlines()
ax.set_global()
plt.imshow(new_scn['I04'], transform=crs, extent=crs.bounds, origin='upper')
cbar = plt.colorbar()
cbar.set_label("Kelvin")
plt.show()
```


360
340
320
300
280
260
240
Kelvin

SatPy and The PyTroll Community

- Community Website: http://pytroll.github.io/ (http://pytroll.github.io/)
- GitHub: https://github.com/pytroll/satpy (https://github.com/pytroll/satpy)
- SatPy Documentation: http://satpy.readthedocs.io/en/latest/ (http://satpy.readthedocs.io/en/latest/)
- Mailing List: https://groups.google.com/d/forum/pytroll (https://groups.google.com/d/forum/pytroll)
- Workshops twice a year!
- Chat with us on slack (invite link on community website)

In [5]:
```
from IPython.display import IFrame
IFrame('https://www.youtube.com/embed/eBQi2G_fqXQ?rel=0', width=600, height=400)
```

Out[5]: