

Applying and Modifying NOAA Satellite Environmental Data

Automated Image Processing at the NOAA VizLab

Rafael de Ameller, Vivek Goel, Tim Loomis

IMSG

Dec 3, 2020



Sea Surface Temperature (SST)

1. Global False Color Imagery

1. ftp://ftp.nnvl.noaa.gov/SOS/GoesSST_Daily/
2. <https://www.nnvl.noaa.gov/view/globaldata.html#SURF>
3. <https://gis.nnvl.noaa.gov/arcgis/rest/services/SURF>

2. How Is It Created?

1. Average Daily Grids If Required
2. Add Color Levels and Background Map



SST Data Discovery

1. Subscribe to Daily Data Feed

<https://www.ospo.noaa.gov/Organization/Documents/Word/ESPCDataAccessRequestForm-Nov2016.docx>

2. Request Individual Set of Days

https://www.class.noaa.gov/saa/products/search?datatype_family=GOESSST

3. sst_geo-polar-blended_5km_YYYYDDD.hdf

April 9, 2020 is sst_geo-polar-blended_5km_2020100.hdf



SST Image Processing

1. L3Harris Interactive Data Language (IDL)

<https://www.l3harrisgeospatial.com/Software-Technology/IDL>

2. Contains Useful Built-In Procedures

1. Read HDF
2. Write PNG

3. Executes from Linux Command Line

```
[vizlab@SOS]$ idl -e "@Hello_World_Run.pro"
```



Script Details

1. Hello_World.pro

```
PRO Hello_World  
  
    PRINT, "Hello World!"  
  
END
```

2. Hello_World_Run.pro

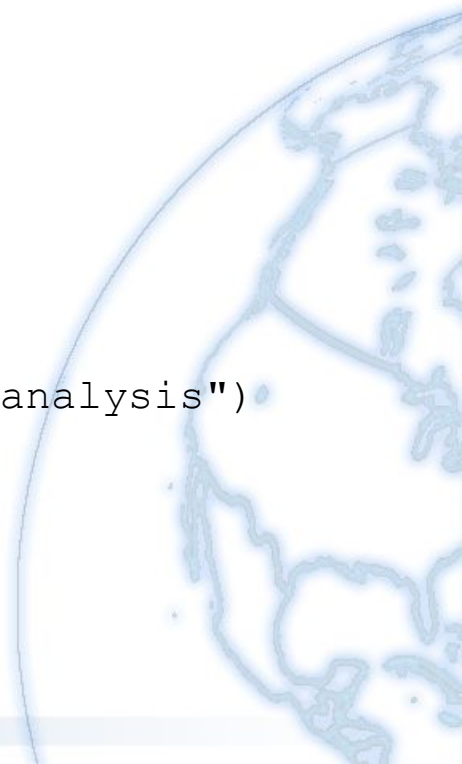
```
.COMPILE "Hello_World.pro"  
  
Hello_World
```



How to Read Data from HDF

```
FUNCTION ReadHDFVar, inpfile, variableName
  SDinterface_id = HDF_SD_START(inpfile, /READ)
  sds_index = HDF_SD_NAMETOINDEX(SDinterface_id, variableName)
  sds_id = HDF_SD_SELECT(SDinterface_id, sds_index)
  HDF_SD_GETDATA, sds_id, variableData
  HDF_SD_ENDACCESS, sds_id
  HDF_SD_END, SDinterface_id
  RETURN, variableData
END
```

```
IDL> SST_grid = ReadHDFVar("sst_geo-polar-blended_5km_2020100.hdf", "sst_analysis")
IDL> PRINT, SIZE(SST_grid)
                2           7200           3600           4           25920000
```



Loading Metadata from HDF

```
FUNCTION ReadHDFAttribute, infile, attributeName, variableName
  SDinterface_id = HDF_SD_START(infile, /READ)
  sds_index = HDF_SD_NAME_TO_INDEX(SDinterface_id, variableName)
  sds_id = HDF_SD_SELECT(SDinterface_id, sds_index)
  attributeInd = HDF_SD_ATTR_FIND(sds_id, attributeName)
  HDF_SD_ATTR_INFO, sds_id, attributeInd, NAME = attributeName, DATA = attributeData
  HDF_SD_ENDACCESS, sds_id
  HDF_SD_END, SDinterface_id
  RETURN, attributeData[0]
END
```

```
IDL> HDF_filename = "sst_geo-polar-blended_5km_2020001.hdf"
IDL> PRINT, ReadHDFAttribute(HDF_filename, "missing_value", "sst_analysis")
-999.000
```



Add Color Levels

```
FUNCTION RGB, grid, MIN = min_value, MAX = max_value, COLORBAR_FILE = colorbar_file
  READ_PNG, colorbar_file, colorbar_image
  sz = SIZE(colorbar_image)
  colorbar = REFORM(colorbar_image[*], *, sz[3] / 2)
  column_in_colorbar = (grid - min_value) / (max_value - min_value) * sz[2]
  sz = SIZE(grid)
  nature_image = BYTARR(3, sz[1], sz[2])
  nature_image[*], *, *] = colorbar[*], column_in_colorbar]
  RETURN, nature_image
END
```

```
IDL> SST_image = RGB(SST_grid, MIN = -4, MAX = 32, COLORBAR_FILE = "SST.clrbr.png")
IDL> PRINT, SIZE(SST_image)
          3          3          7200          3600          1          77760000
```



Add a Background Map

```
READ_PNG, "Snow-Cleared_4096.png", blueMarble, /ORDER  
landIndexes = WHERE(blueMarble[3, *, *] NE 0)  
SST_channel = SST_image[0, *, *]  
blueMarble_channel = blueMarble[0, *, *]  
SST_channel[landIndexes] = blueMarble_channel[landIndexes]  
SST_image[0, *, *] = SST_channel  
SST_channel = SST_image[1, *, *]  
blueMarble_channel = blueMarble[1, *, *]  
SST_channel[landIndexes] = blueMarble_channel[landIndexes]  
SST_image[1, *, *] = SST_channel  
SST_channel = SST_image[2, *, *]  
blueMarble_channel = blueMarble[2, *, *]  
SST_channel[landIndexes] = blueMarble_channel[landIndexes]  
SST_image[2, *, *] = SST_channel
```



Save Image to File

```
FUNCTION Save_Global_SST_Image, HDF_filename, SST_image
  HDF_filename_stem = FILE_BASENAME(HDF_filename, ".hdf")
  year_and_day = STRMID(HDF_filename_stem, 6, /REVERSE_OFFSET)
  Julian_day = JULDAY(1, 0, STRMID(year_and_day, 0, 4)) + STRMID(year_and_day, 4)
  CALDAT, Julian_day, month, day, year
  PNG_file = STRING(FORMAT = '("SST_", I04, I02, I02, ".png")', year, month, day)
  WRITE_PNG, PNG_file, SST_image, /ORDER
  RETURN, PNG_file
END
```

```
IDL> HDF_filename = "sst_geo-polar-blended_5km_2020100.hdf"
IDL> image_file = Save_Global_SST_Image(HDF_filename, SST_image)
IDL> PRINT, image_file
SST_20200409.png
```



Scheduled Tasks

1. Use Linux 'cron' to Automatically Run IDL When Desired

```
[vizlab@SOS]$ crontab -e
```

```
00 09 * * * idl -e "@GoesSST_Run.pro" >> GoesSST.out 2>&1
```

Executes Daily at 9:00 am

2. Enable End User Access

```
[vizlab@SOS]$ chmod 755 GoesSST_Daily.png
```

