Analysis Ready Datasets

CLOUD OPTIMIZED GEOTIFF

@_VincentS_
@normanbarker

Vincent Sarago - Software Engineer @mapbox
Norman Barker - Software Engineer @mapbox

While we are introducing ourselves go ahead and install Python 3.6!

The Cloud-Optimized GeoTIFF (COG) format is a way to store satellite data to allow for fast web access and cloud analysis at scale. Adopted widely by the Earth on AWS effort, it immediately enabled new forms of access and use of imagery data.

We will share Mapbox's experience working with COGs. We'll also cover how to create them from other formats and how to use them in your application.

https://github.com/cogeotiff/cog-spec/blob/master/spec.md

Started by Even Rouault and taken forward by Chris Holmes and others as a working group

Analysis Ready Data

- Reduce the amount of data processing required by a user
- COGs are one method of reducing data transfer and hence processing when applicable

TIFF

- Libtiff released by Sam Leffler in 1988
- TIFF 6 spec released in 1992
- Extended to GeoTIFF in 1995 by Niles Ritter and Mike Ruth
  - Extended to BigTIFF in 2007
  - Maintained by Frank Warmerdam and Even Rouault
- A format defined of generic tags (Tagged Image File Format) and a registry

TIFF is the answer to the ultimate question of life, the universe, and everything

42 is the version number of TIFF v5 and v6.

TIFF constructs

- Image File Directory
- Tags and Values
- Values and Offsets (determined by type and count)
- Tiles
- Strips
- Scanlines
- Core and extensions, tiles are an extension

LibGeoTIFF and network access

- https://github.com/OSGeo/gdal/blob/master/gdal/frmts/gtiff/libgeotiff/xtiffio.h#L73

Largely unchanged from when it was written 20+ years ago.

LibGeoTIFF wrapped into GDAL.

Rasterio

Modern Python library that wraps GDAL and allows read and write geospatial raster data.

- Python 2/3 compatible
- Open Source
- Stable
- Used by GBDX, TWC, Planet and RemotePixel

Why COGs, why not JPEG2000 or another format?

- TIFF is simple
  - TIFF tiles will run over the edge of an image
- In COGs IFDs are at the beginning of the file
- No stateful server required
  - Already in Rasterio and GDAL

When not to use COGs?

- When you need sub-tile access
- When bandwidth is a concern
- Cross band analysis
- High rate lossless compression for more than 4 bands
- When you don’t need network access!!

Mapbox and COGS

- Creating open source tooling around COGS
- We are part of the working group around COGs
- We use COGs to store our intermediate data between source and tiles

WebP and COGs

- JPEG
- ZSTD
- LERC
- LZW
- DEFLATE
- JPEG2000
- ....

WORKSHOP

```
pip install rasterio rio-cogeo
```

Sample data: https://bit.ly/satsummit_cogeo

Optional deps:

`libtiff-tools` (http://brewformulas.org/Libtiff)

gdal_translate

```
gdaladdo -r average
dg_post_florence_0003113_crop.tif 2 4 8 16 32

gdal_translate dg_post_florence_0003113_crop.tif
cogeo.tif -co TILED=YES -co COMPRESS=JPEG -co
PHOTOMETRIC=YCBCR -co COPY_SRC_OVERVIEWS=YES
```

tiffinfo and tiffdump

```
tiffinfo cogeo.tif
```

```
tiffdump cogeo.tif
```

rio-cogeo

```
rio --help
```

```
rio cogeo dg_post_florence_0003113_crop.tif cogeo.tif
```

COGDumper

https://github.com/mapbox/COGDumper

```
pip install cogdumper

cogdumper file --file cogeo.tif -xyz 0 0 5
```

Debug remote access

```
CPL_CURL_VERBOSE=YES gdal_translate -outsize
256 256 -srcwin 0 0 2048 2048 /vsicurl/https://
s3-us-west-2.amazonaws.com/remotepixel-pub/cog/
dg_post_florence_0003113_cogeo.tif test.tif
2>&1 >/dev/null | grep "> GET"
```

Visual

```
pip install rio-glui

rio glui https://s3-us-west-2.amazonaws.com/remotepixel-pub/cog/dg_post_florence_0003113_cogeo.tif
```

Best Practises

- WebP for RGB(A) - Under development
- LERC for floating point
- JPEG for simple RGB (-> YCbCr)
- ZSTD compression (better than deflate)
- Validate your COGs
- Participate in the Working Group
  - http://www.cogeo.org
  - https://lists.osgeo.org/mailman/listinfo/cog

Questions