

DIY Raster Tiles with QGIS & GDAL

Using open source tools to generate orthomosaics and web map tiles from aerial imagery.

Malcolm Meyer

Ohio GIS Conference 2021

About Me

- BA in Sociology, MA in International Affairs
- 10 Years in the Geospatial Sector
- 
-  Coffee,  Gardening,  Music
- 

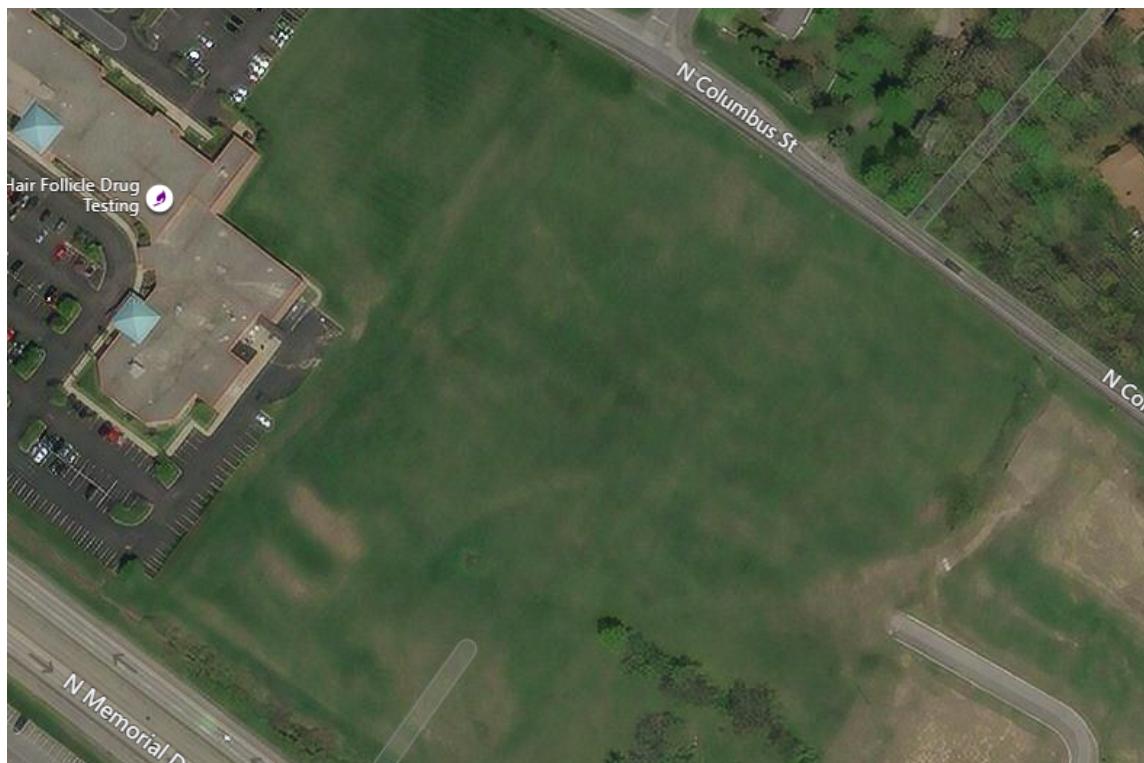


**High Resolution
Orthoimagery is one of
the foundations of a
Modern GIS.**

Why not Commercial (Satellite) Imagery?

- Commercial Imagery is Outdated
- No control over Projections/Transformations
- No control of the tile server
- Low Resolution
 - 15cm Native Imagery - Umbra
 - 15cm Maxar Upscaled Imagery

Bing Maps (~2016)



Google Maps (~2018)



Esri/Mapbox (Maxar ~2019)



Local 2020 Aerial



Why **FOSS** Image Processing

- No license needed and runs on any machine
- Full control of the mosaic to tile pipeline
- Full control over the tile server
- You get to learn more about how web maps work
- You want to plan a mission to Mars

Even More Reasons

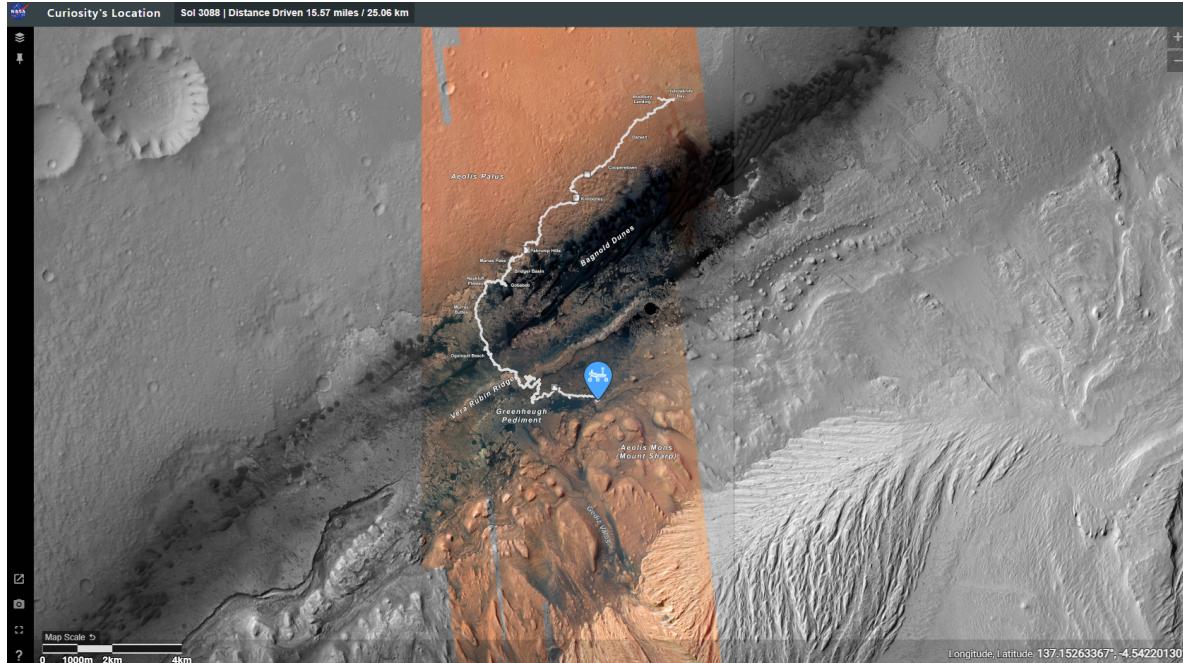
0%	338.2 MB	0 MB/s	0 Mbps	0%	GPU 0 - 3D	Very low	Very low
23.4%	161.8 MB	0 MB/s	0 Mbps	0%		Very high	Low
0%	130.7 MB	0 MB/s	0 Mbps	0%		Very low	Very low
0.7%	101.3 MB	0 MB/s	0 Mbps	0%		Very low	Very low
0%	85.6 MB	Generating (Not Responding)					
0%	66.1 MB						
0%	37.4 MB						
0.9%	37.3 MB						
0%	36.0 MB	Generating tile cache...					
0%	28.5 MB						
0.9%	26.9 MB						
0.3%	21.5 MB	0 MB/s	0 Mbps	0%		Very low	Very low
0%	19.1 MB	0 MB/s	0 Mbps	0%		Very low	Very low
0%	18.8 MB	0 MB/s	0 Mbps	0%		Very low	Very low
0%	18.3 MB	0 MB/s	0 Mbps	0%		Very low	Very low
0.5%	17.0 MB	0 MB/s	0 Mbps	0.2%	GPU 0 - 2D	Very low	Very low

Simple Deployment

**The output can be deployed
on any static web host or CDN***

*In some instances it will be advantageous to use a single mbtiles file (sqlite db) instead of a directory of tiles.

Tools are Open Source



Imagery Pipeline

- 1 Raw TIFF **.tif**
 - Available from OSIP
- 2 Mosaic **.vrt**
 - Intermediate File
- 3 Cloud Optimized GeotIFF **.tif**
 - Used in Desktop GIS
- 4 Raster Tiles **.mbtiles**
 - Used in Web Maps
- 5 Tile Server

Virtual rasters

Supported in ArcMap, ArcGIS Pro and QGIS.

[.vrt] is a file format created by the Geospatial Data Abstraction Library (GDAL). It allows a virtual dataset to be derived from other datasets that GDAL can read.

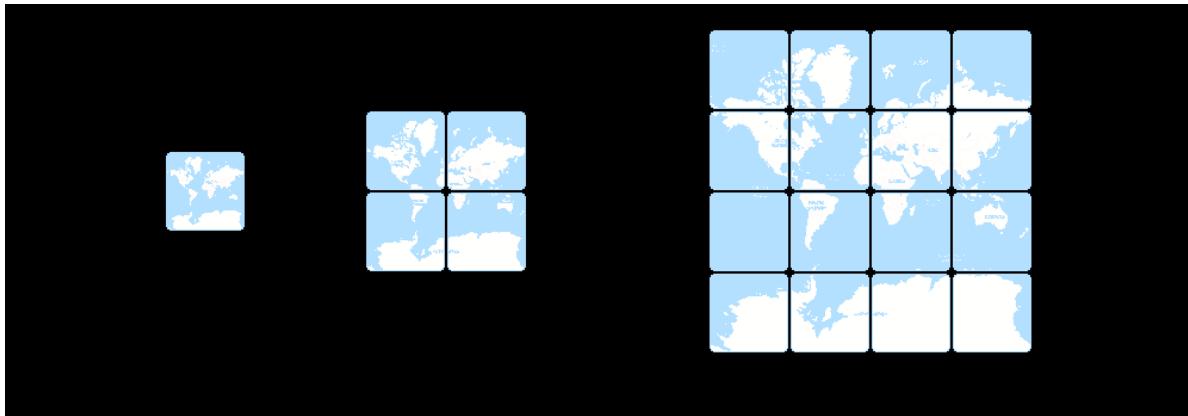
[ArcGIS Pro Supported Raster Formats](#)

Cloud Optimized GeoTIFF

A regular GeoTIFF file, aimed at being hosted on a HTTP file server, with an internal organization that enables more efficient workflows on the cloud.

Raster Tile Basics

Web Maps are usually in Web Mercator EPSG:3857



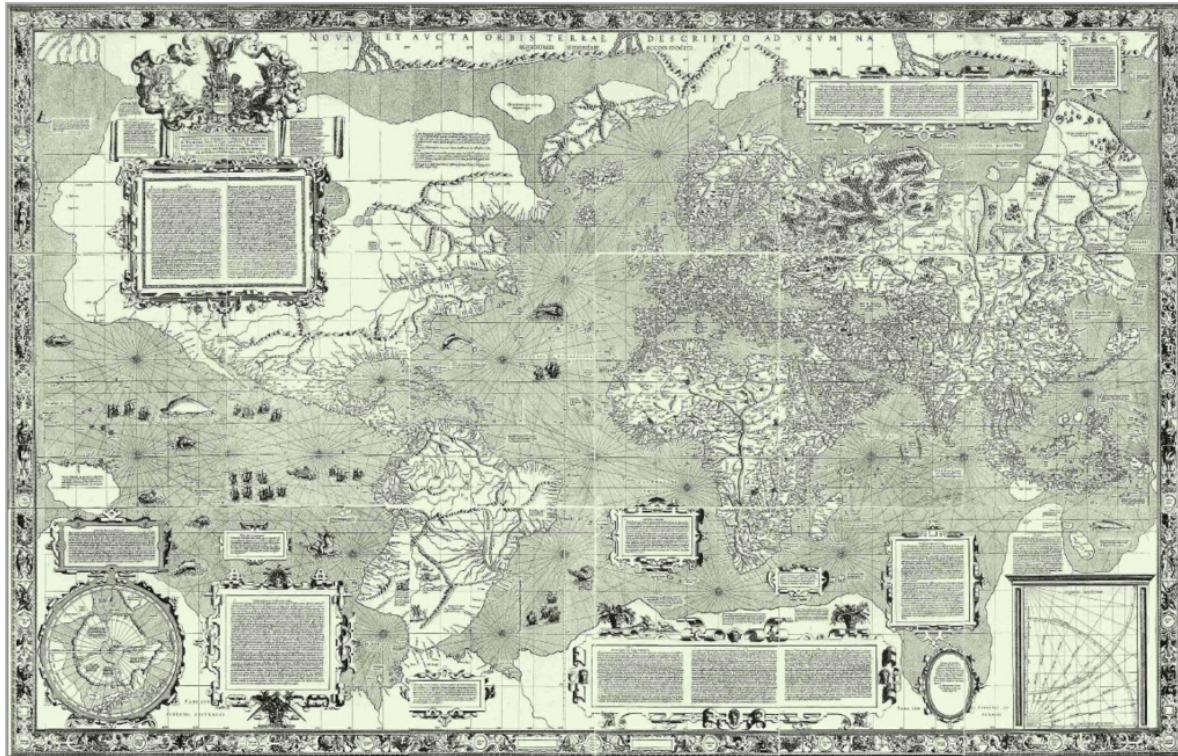
Each zoom level is a directory, each column is a subdirectory, and each tile is an image.

MBTiles

An sqlite database consisting of a **metadata** table and a **tiles** table stored in TMS format.

<u>zoom_level</u>	<u>tile_column</u>	<u>tile_row</u>	<u>tile_data</u>
zoom level	x coord	y coord	image as blob
15	8858	20330	blob

Carta do Mundo de Mercator (1569)



The 'Mercator' Projection

Firstly, to spread on a plane the surface of the sphere in such a way that the positions of places shall correspond on all sides with each other, both in so far as true direction and distance are concerned and as correct longitudes and latitudes...

On Cartography

...the limitations of ancient geography
be not unknown and that the honour
which is due to past centuries be given
to them.

~ Gerardus Mercator, 1569

Mercator Distortions



True Size

Image Processing Tools

GDAL

- gdalinfo
- gdalbuildvrt
- gdal_translate
- gdalwarp

Raster Tiling Tools

- gdal2tiles.py
- QGIS Generate xyz tiles (files or mbtiles)
- gdal_translate (mbtiles)

Hosting Options

- COG
 - Terracotta, Titiler (Python)
- XYZ Folder Tree
 - Any web host
- MBTiles
 - mbtileserver (Go)
 - wmts-server (NodeJS)
 - many more...

A Note on File Size (z21)

Source	Files	Format	Size
Raw Tiff	39,985	TIFF	1150 GB
Esri Export	4	TIFF/LERC	620 GB
gdal_translate COG	1	TIFF/JPG	82 GB
ArcGIS Cache	1,316	PNG, JPG	101 GB
Raster Tiles	11,511,665	PNG	851 GB
Raster Tiles	11,511,665	JPG	120 GB
MBTiles z20	1	PNG	200 GB

OverZoom in AGOL

OverZoom is not consistent between mapping APIs, so managing this at the server level allows us to enable overzoom for all maps.

[AGOL Cache](#)

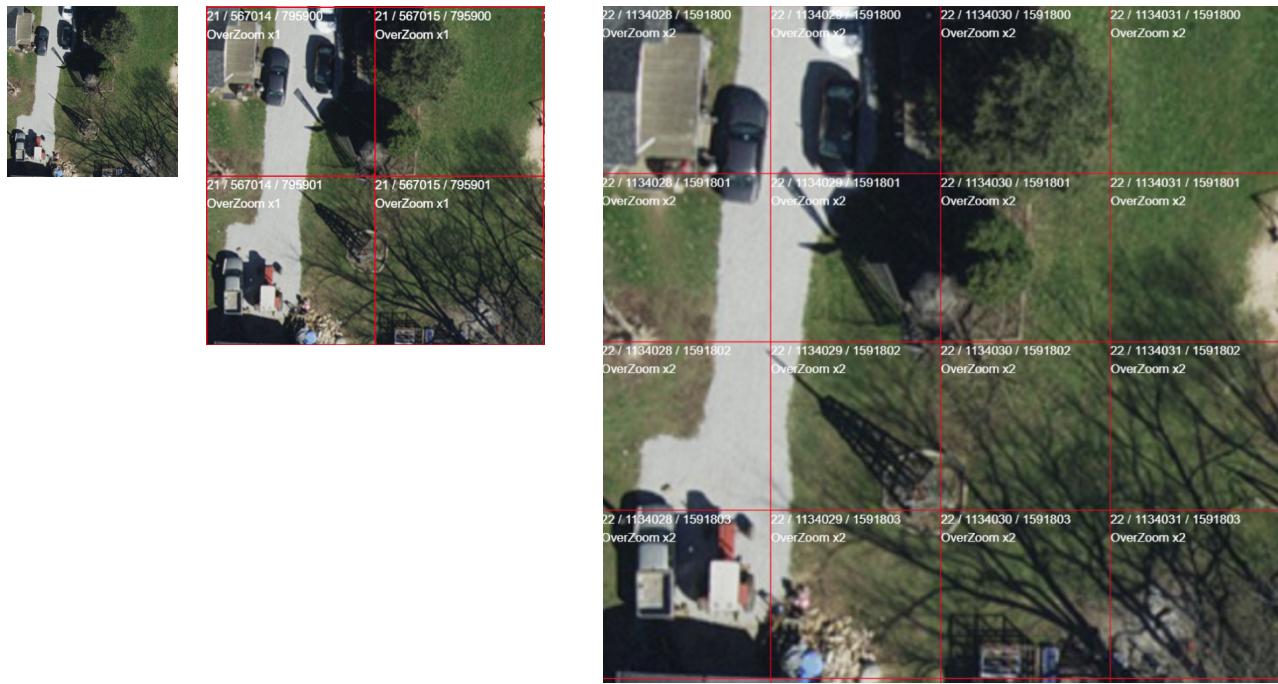
[AGOL Example](#)

Custom WMTS Tile Server

WMTS Endpoint & OverZoom

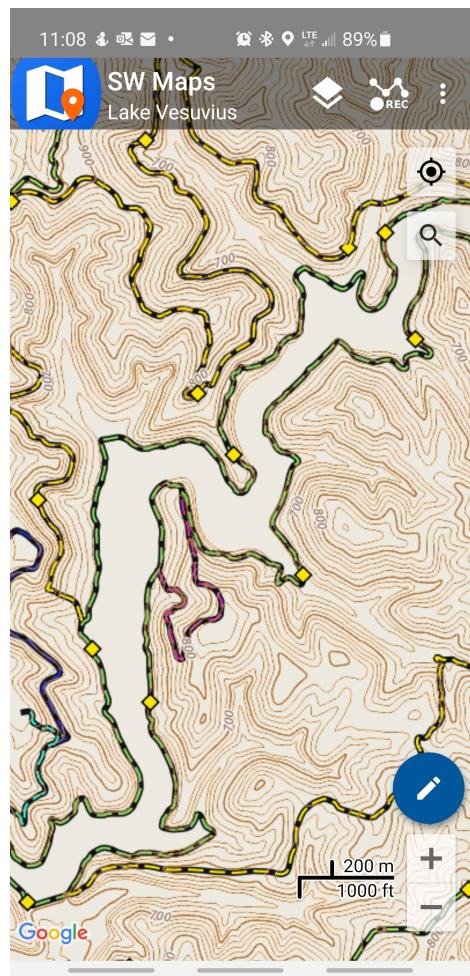
Take a tile at zoom level 20, resize it, slice it up and serve it back to the client!

OverZoom Example



Viewing MBTiles on Android

- Input/Mergin
- MapTiler
- **SWMaps**
- QField



BREAK

Let's Get Coding

For those wanting to follow along.

- Have QGIS ^3.x installed and working
- Have gdal installed and working
 - Open OSGeo4W Shell
 - Type `gdalinfo --version`
 - Return ~ `GDAL 3.3.1, released 2021/06/28`
- Obtain a copy of the sample rasters

The Process

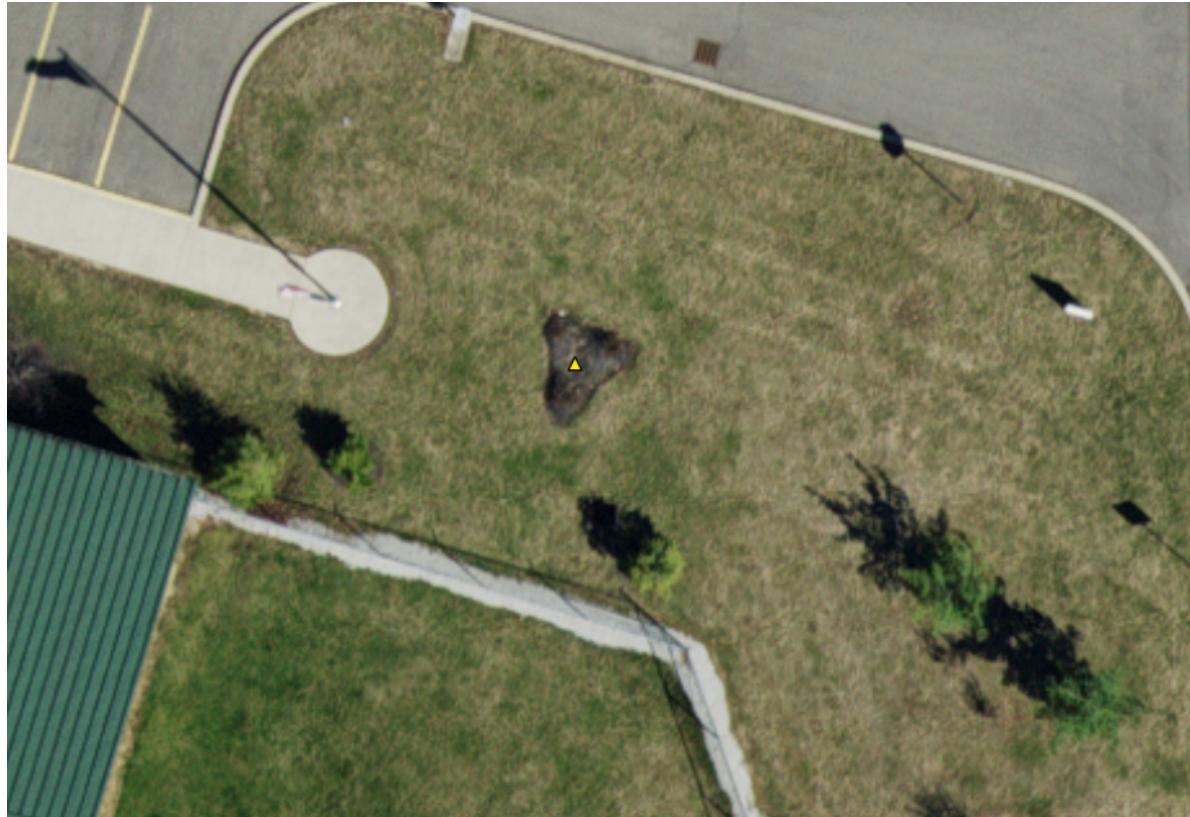
- 1 Explore raster metadata with `gdalinfo`.
- 2 Verify the accuracy of the data in QGIS with NGS ground control points.
- 3 Create a Mosaic with `gdalbuildvrt` (force NAD83 2011 SP South).
- 4 Create the COG with `gdal_translate`.
- 5 Examine the output in QGIS.
- 6 Create another VRT in WebMercator.
- 7 Generate Raster Tiles with `gdal2tiles`, `gdal_translate`, and QGIS.
- 8 Serve the raster tiles with `wmts-server`.
- 9 View the mbtiles file on mobile with SW Maps.

gdal-info

```
gdalinfo --version  
> GDAL 3.3.1, released 2021/06/28
```

```
gdalinfo OSIP\BS921630.tif  
> Lots of Metadata
```

Accuracy Validation



NGS Marker in Person



Create the Mosaic

Create the mosaic in the native projection, adding in our own alpha band.

```
gdalbuildvrt -b 1 -b 2 -b 3 -addalpha mosaic.vrt  
OSIP/*.tif
```

Transformations

We need our data in NAD 1983 (2011) EPSG:6551 and WebMercator EPSG:3857, but the imagery is in EPSG:3754!

Create the Cloud Optimized GeoTIFF

```
gdal_translate mosaic.vrt cog.tif ^
-a_srs EPSG:6551 ^
-of COG ^
-co RESAMPLING=CUBIC -co COMPRESS=JPEG -co BIGTIFF=YES -
co NUM_THREADS=ALL_CPUS
```

The `-a_srs` option assigns an output projection without performing a transformation.

Validate the COG



Cloud Optimized GeoTIFF Validator: result

Validation succeeded ! cog.tif is a valid Cloud Optimized GeoTIFF.

[Return to submit page](#)

<http://cog-validate.radiant.earth/html>

Convert to WebMercator

The default GDAL/Proj transformation from the NAD 1983 datum to WGS 1984 datum is null, and the only other option does not exactly match the default option in our commercial software.

GDAL NAD 1983 2011 to WGS 1984 3

```
+x=-0.991 +y=1.9072 +z=0.5129 +rx=-0.0257899075194932  
+ry=-0.0096500989602704 +rz=-0.0116599432323421
```

Esri ITRFOO (Default NAD83 ◇ WGS84)

```
0.9956,1.9013,0.5215,0.025915,0.009246,0.011599,-0.00062
```

geographic transformations

proj definitions

Spurious Precision

It's the classic problem of spurious precision...The problem we're trying to solve is topological accuracy, not absolute accuracy. ~ Utah Geospatial Resource Center

Proj4 String

Replacement for the default NAD83 datum transformation definition in GDAL.

```
+proj=lcc +lat_0=38 +lon_0=-82.5 ^
+lat_1=40.033333333333 +lat_2=38.733333333333 ^
+x_0=600000 +y_0=0 +ellps=GRS80 ^
+to_wgs84=-0.9956,1.9013,0.5215,0.025915, ^
0.009246,0.011599,-0.00062 ^
+units=us-ft +no_defs
```

Create a VRT in 3857

```
gdalwarp mosaic.vrt web.vrt ^
-t_srs EPSG:3857 ^
-s_srs "+proj=lcc +lat_0=38 +lon_0=-82.5
+lat_1=40.033333333333 +lat_2=38.733333333333 +x_0=600000
+y_0=0 +ellps=GRS80
+to_wgs84=-0.9956,1.9013,0.5215,0.025915,0.009246,0.011599,-0
.00062 +units=us-ft +no_defs" ^
-dstalpha ^
-et 0 ^
-r lanczos ^
-of VRT ^
-overwrite
```

-et: error threshold

Create the Raster Tiles

- gdal2tiles.py
- QGIS Generate XYZ Tiles (mbtiles)
- **gdal_translate**

gdal_translate

Create the MBTiles at zoom level 20
ZOOM_LEVEL_STRATEGY=LOWER with cubic resampling.

```
gdal_translate -of MBTILES ^
--config GDAL_NUM_THREADS ALL_CPUS ^
-co ZOOM_LEVEL_STRATEGY=LOWER ^
-co RESAMPLING=CUBIC ^
web.vrt tiles.mbtiles
```

gdaladdo

Create overview tiles until the smallest tile that covers the image is 256x256.

```
gdaladdo tiles.mbtiles -r average
# nearest,average,bilinear,cubic,cubicspline,lanczos...
# average is marginally faster
```

Hosted Tiles

gdal2tiles

```
python.exe gdal2tiles.py -xyz -v source.tif outputFolder
```

Hosting & Serving

Copying millions of small files can take a very long time, however deployment is as simple as pointing a web server to a folder.

Advantages of MBTiles

- One file vs millions of files
- Complete control of the tile server

MBTile Servers

- [mbtileserver](#)
 - Written in GO, deployed via a windows exe file
- [wmts-server](#)
 - NodeJS server with WMTS endpoints and 3x overzoom
- Many others...

References

- <https://www.researchgate.net/publication/321064>
- <https://blog.mastermaps.com/2008/03/generating->
- <http://build-failed.blogspot.com/2012/11/zoomab>
- <https://alastaira.wordpress.com/2011/07/11/mapt>
- <https://pvanb.wordpress.com/2017/03/06/raster2m>
- <https://github.com/ecometrica/gdal2mbtiles>
- <http://emapr.ceoas.oregonstate.edu/pages/educat>
- <http://www.perrygeo.com/lazy-raster-processing->
- <https://jeromegagnonvoyeur.wordpress.com/2015/08>

Thanks!

Malcolm Meyer

@getBounds on twitter and the web