Satellite/Aerial Image Retrieval

- Venkata Akshith Reddy Kasireddy (A20455209)
- Sai Vishal Kodimela (A20453006)
- Nikhil Sarika (A20470289)
- Souporno Ghosh (A20439047)

Content

- Introduction
- Our Method
- Result
- Future Improvements
- References

Introduction

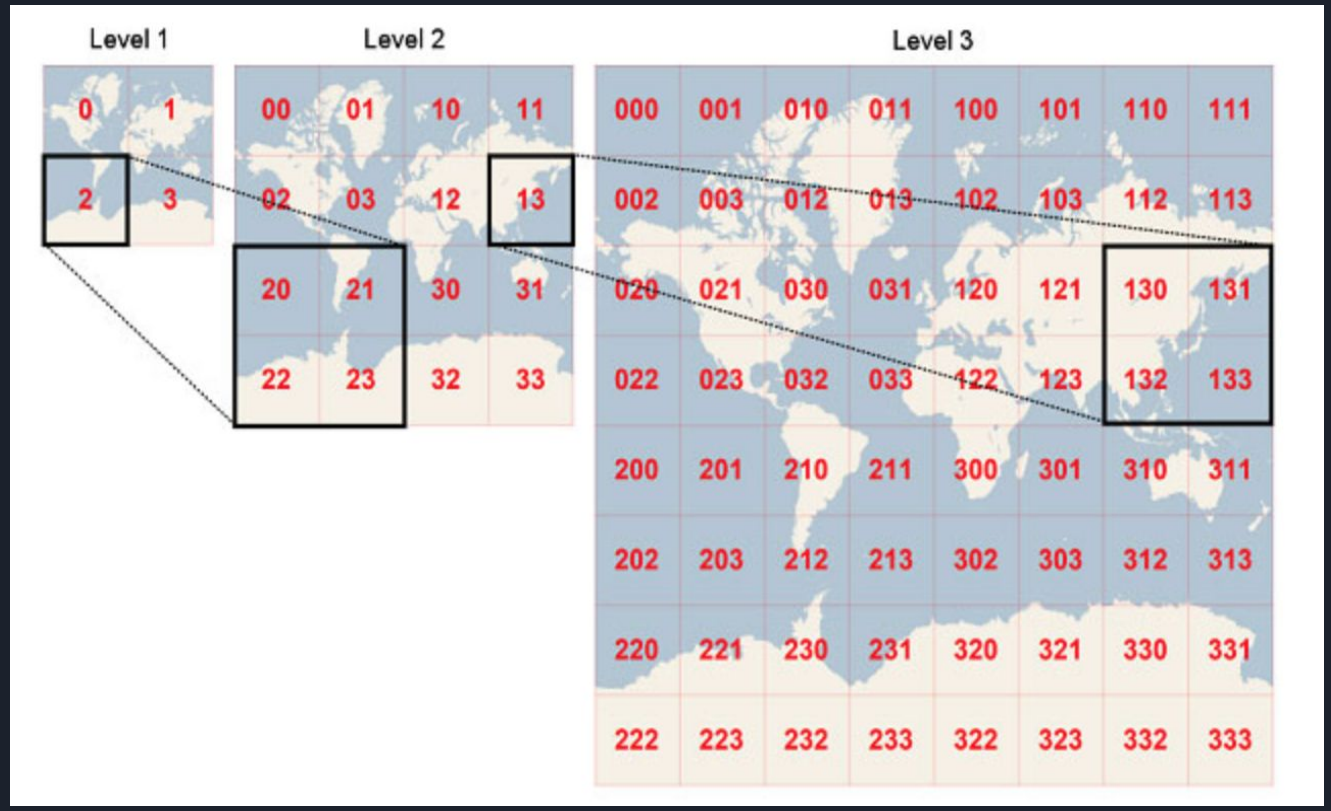
Our Goal :

- Given two points in geo coordinate, return the map image bounded by the two points with acceptably finest resolution.
- Use Bing Map Tile System to obtain the required highest resolution Aerial Image.

Input and Output :

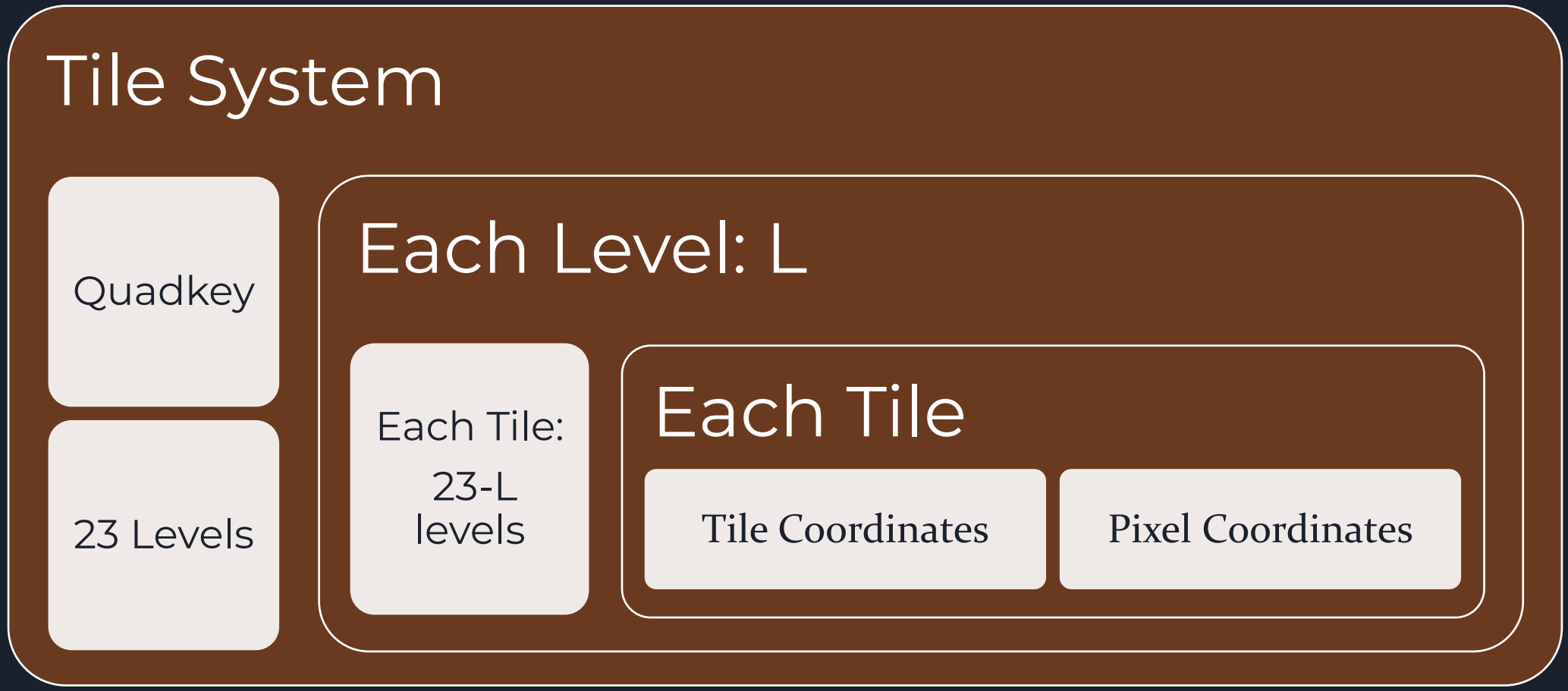
- Input: 2 bounding coordinates
  - lat1, lon1, lat2, lon2
- Output:
  - an aerial imagery within the bounding box defined above.

Bing Maps Tile System



- Project the world map to a square number of pixels. A bunch of pixels consist a tile orderly.
- In higher level, there are more pixels, bigger map size, more tiles, and finer ground resolution. There are 23 levels in total. Each tile in current level expand to 4 tiles in next finer level.

Bing Maps Tile System (Con.)



Our Method

Tools We used

- Python 3.6
- Package: Pillow
- Bing Map Tile System

Query a Aerial Image from Bing Map Tile System

- Query from URL by quadkey:
  - *http://ho.ortho.tiles.virtualearth.net/tiles/hquadkey.jpeg?g=131*
  - quadkey: can be generated by converting a pair of latitude and longitude.
- Extract the socket and obtain the queried tile image.

Generate Bounding Box of Aerial Image

Base Tile

- Find the smallest tile which bounds everything of our bounding box.
- Inside this base tile, recursively find finest tiles.

Finest Tile

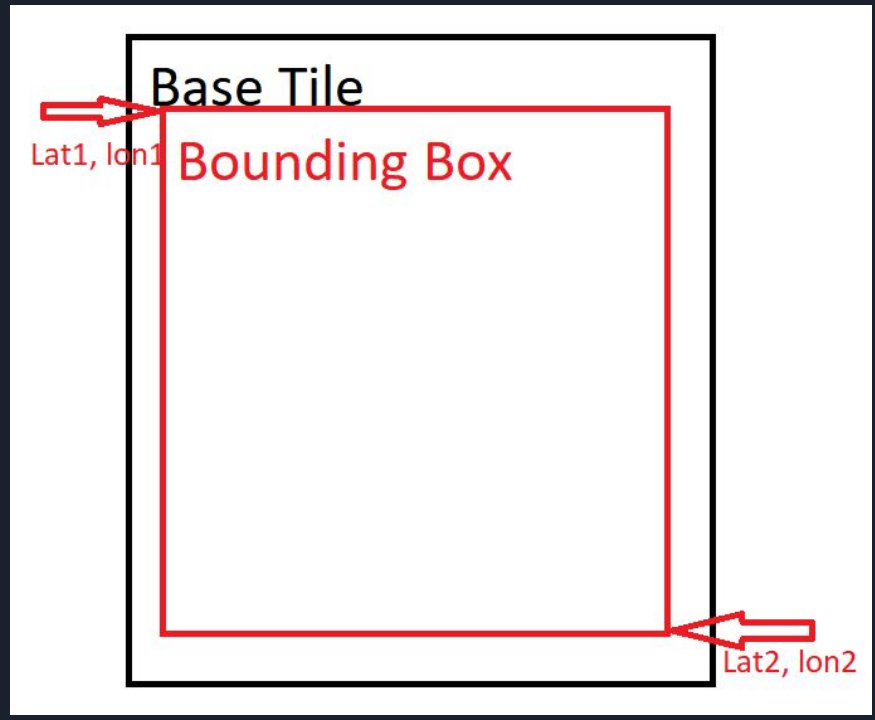
- From level 23 to lower levels, find the first non-null tile aerial images.
- Stitch these tiles together to generate a “finest tile”

Cropping

- Convert Longitudes and latitudes to pixel coordinates in the finest tile.
- Crop the finest tile to the bounding box of required aerial image.

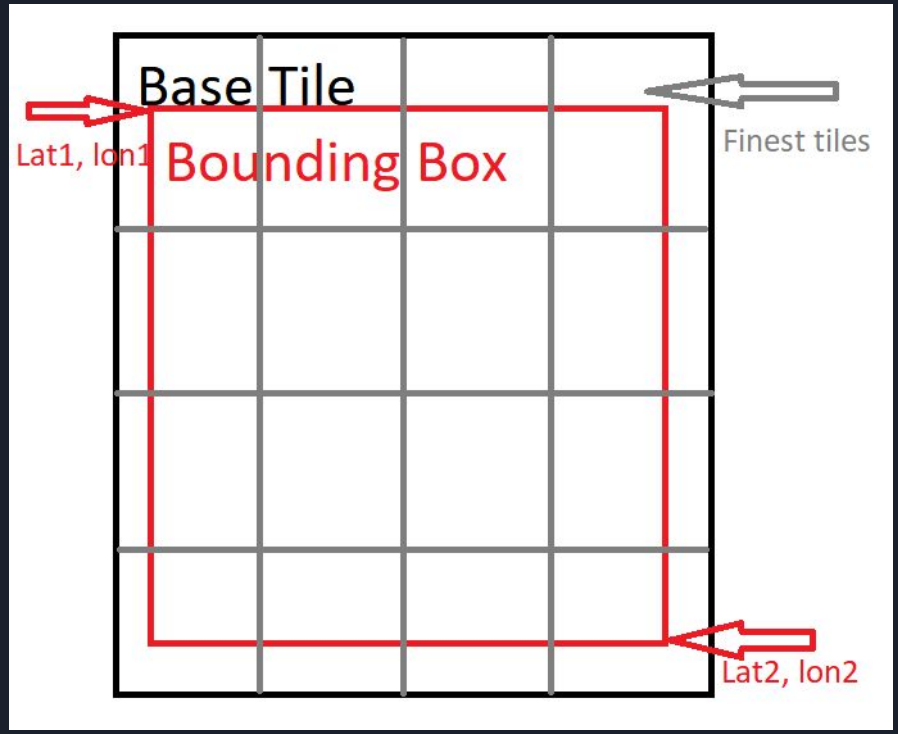
Base Tile

- Input latitudes, longitudes: lat1, lon1, lat2, lon2
- Convert inputs to tile coordinates: x1, y1, x2, y2
- Search From level 23 to level 1
- If $|x1-x2| \leq 1$ and $|y1-y2| \leq 1$
- This tile level is the base tile level
- Return the tile level and tile coordinate x1, y1



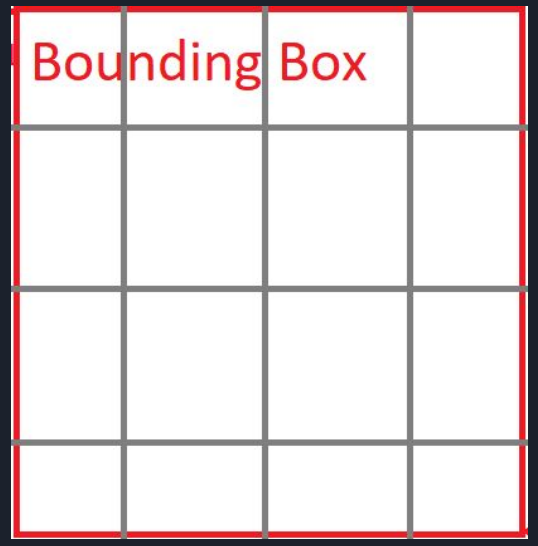
Finest Tile

- Input tile coordinates: x1, y1, x2, y2 and tile level.
- From level 23 to base tile level:
  - If we can query all tile images in this level,
  - Return all tile images inside base tile.
- Stitch these tile images together to generate finest tile image.
- Return finest tile image.



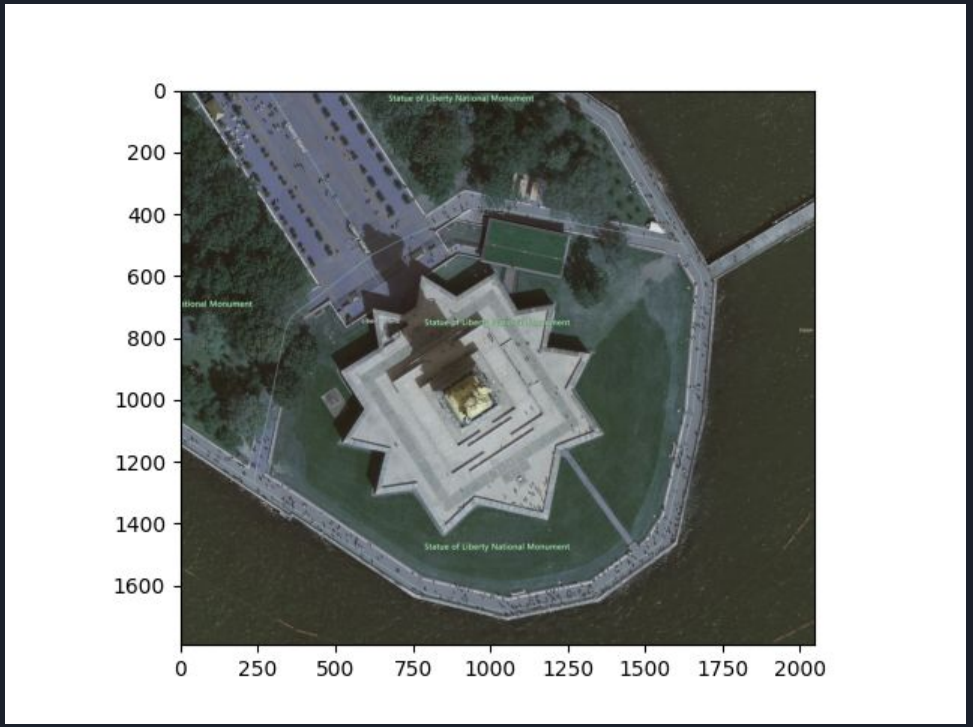
Cropping

- Input: lat1, lon1, lat2, lon2 and finest tile image.
- Convert latitudes and longitudes into pixel coordinates of finest tile.
- Cropping finest tile image by the pixel coordinates to generate required bounding box of aerial image.
- Return the bounding box image.

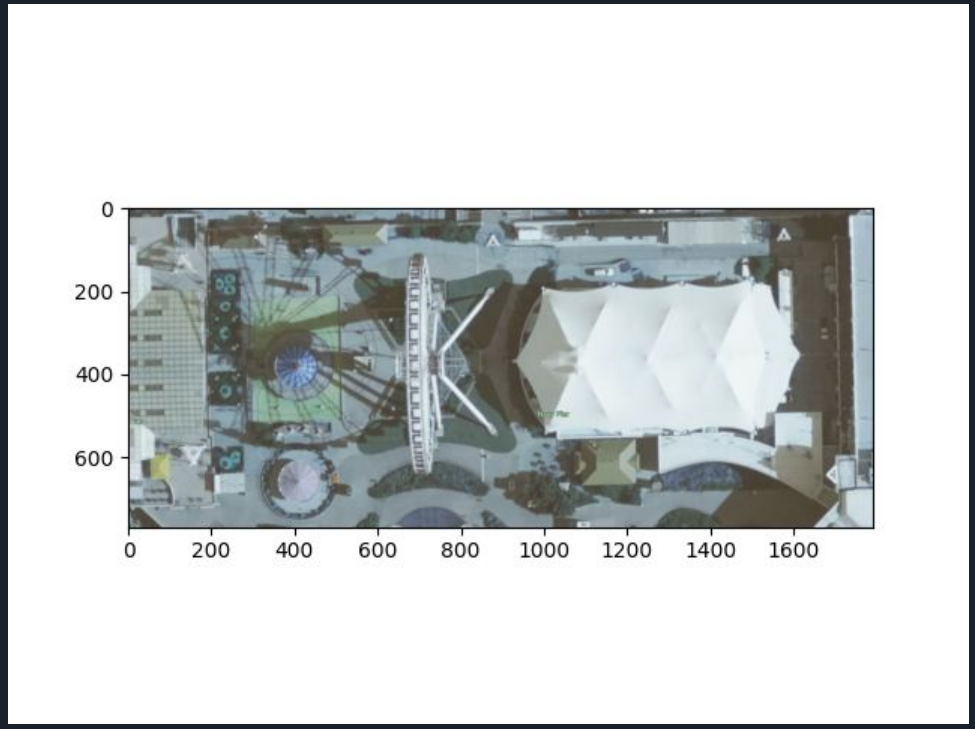


Results

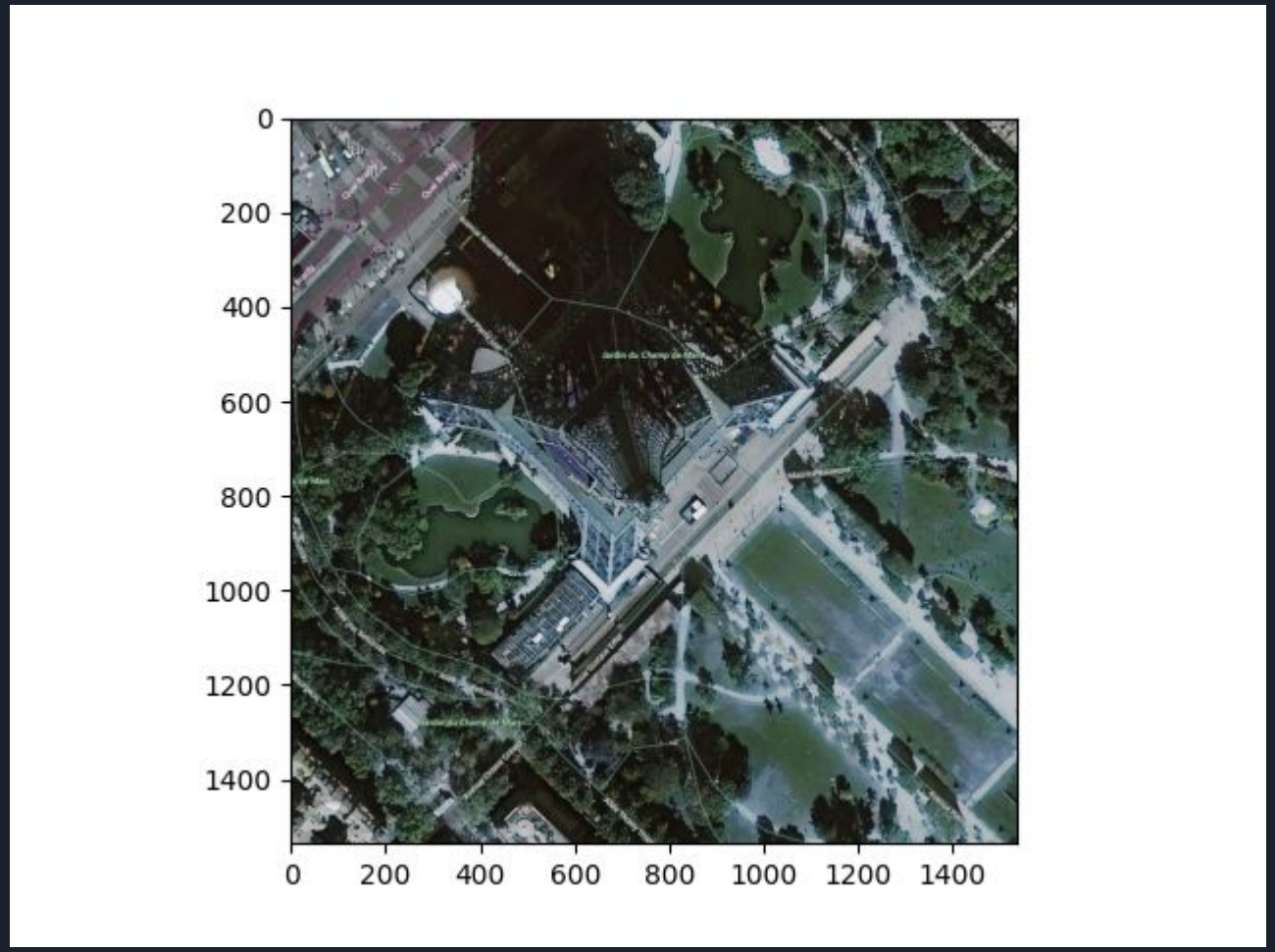
Aerial Image of Statue of Liberty



Aerial Image of Chicago Navy Pier

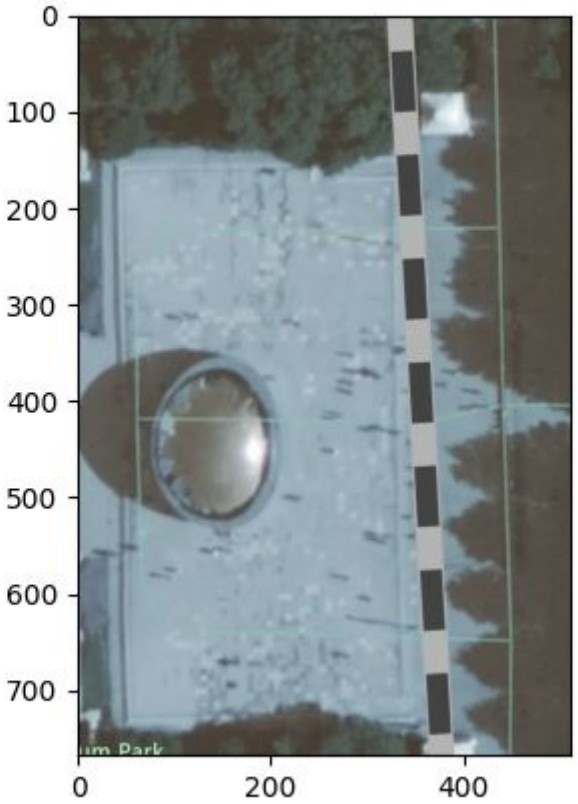


Aerial image of Eiffel Tower



Aerial image of Stuart Building

Aerial image of Chicago Bean



Future Improvements

- Python’s PIL has a memory allocation limit
  - Too many tiles == doesn’t work
  - Our limit: $2^{13}x2^{13}$ or 8192x8192 pixels
- Threaded Performance:
  - Good for High Level of detail
  - Requires lots of memory

References

- Bing Maps Tile System : “https://docs.microsoft.com/en-us/bingmaps/articles/bing-maps-tile-system”

Thanks !