# Satellite imagery technical process

The following will provide a technical summary of the process of collecting and associating satellite images with given intersections.

## Ingest

The ingestion routine starts with a given latitude and longitude as calculated from the Open Street Map (OSM) data [1] which are calculated with the OSMNX [2] tool. The latitude and longitude are used as arguments to the SlippyMap API from OSM. The slippymap data divides the earth into a set of tiles, the size of which is dictated be the zoom level. For our purposes the zoom level is set to the value of 18, providing pixels with approximate width of 0.6 meters. Because the intersection location can land on the corner of a tile, an algorithm is used to determine how many of the neighboring tiles might need to be included to contain the intersection geometry. Once these tiles are identified, a bounding box (SlippyBox) is created and this area is requested from the MapBox [3] API. This image is cropped and centered over the intersection and saved with a unique filename into a data storage location. This ultimately will be in the cloud – S3.

Additionally the SlippyBox can be used to create an intersection’s “mask” which is a black and white image created just from the OSM data. Essentially the same data that is used render map geometries on phone applications is used to draw these intersection masks. These are mostly useful for later computer vision and feature extraction tasks.

## Data pipeline

The satellite imagery data pipeline is fully automated and populates images for all intersections provided by the OSM data. This process can be time consuming, so there exists a limit parameter to only ingest a set number of satellite images for testing.

Currently on application startup, a service container, ug-intersectionsgraph, is spun-up that takes a location search string as an input parameter. This can be a specific city, state, or region. Currently for testing, this is hardcoded to “District of Columbia”. The OSMNX library calculates all intersections from the OSM data, and an intermediate json file is created. It is from this file that the satellite imagery ingestion pipeline reads and executes the procedure outlined in the previous section. The ug-intersectionsgraph service then extracts relevant mire features from both the OSM and image data. Finally the service loads all images, features, and metadata into the central intersection database.

## Association

Due to the satellite images being created by intersection data as an input, the association of images to intersection data and subsequent mire features is trivial.

## Manual operation

If one wishes to run this process manually, see the README located in the repo at “services/intersections/README.md”.

[1] [https://www.openstreetmap.org/](https://www.openstreetmap.org/#map=13/38.8834/-77.0752)

[2] <https://osmnx.readthedocs.io/en/stable/>

[3] <https://www.mapbox.com/>