NBI Data:

Bridges are points with lat/long. Points are (probably) center of the bridge

OSM:

Bridges are ways, comprised of nodes. Nodes have lat/long, ways don’t. The ways also have tag data, which we’ll push our NBI data into.

Idea:

1. Go through OSM data, getting ways tagged with “bridge: yes”.
2. For each of those ways, find the closest corresponding NBI bridge, and add that bridge data to the way.
3. (Handle this later) If there is any NBI data *not* covered, find the OSM way closest to that bridge. This will be hard…

Note: Some bridges might curve, making the optimal nodes difficult to decide. Using a bounding box around the bridge might be a good idea. A white board with writing on it

Description automatically generated with low confidence

Bridges with highway:footway or foot:yes tags are not in NBI and can be ignored.

lat/long for center of bridge across Little Papillion Creek on Vernon Avenue

OSM : 

NBI for same bridge: lat: 41.3158333, long: -96.0461111

Lat/long for center of bridge on Read/Craig Street.

OSM: 

NBI for same bridge: lat: 41.32930. long: -96.04820

NBI:

[{'id': 'C002841205P', 'lat': '41.3154', 'lon': '-96.0524'},

{'id': 'C002841215P', 'lat': '41.3208', 'lon': '-96.043'},

{'id': 'C002841207P', 'lat': '41.3209', 'lon': '-96.0449'},

{**'id': 'C002841210P', 'lat': '41.3159', 'lon': '-96.0462'},**

**Okay, so problem…This corresponding way is *not* listed as a bridge in osm. Its way is also very long (25 nodes).** [**We may need to rethink this problem. (Or not)**](#_NBI_culverts_and)

**Graphical user interface

Description automatically generated with low confidence**

{'id': 'U182514510P', 'lat': '41.3217', 'lon': '-96.0333'},

{'id': 'C002813725P', 'lat': '41.3217', 'lon': '-96.0394'},

{'id': 'C002841212P', 'lat': '41.3236', 'lon': '-96.0449'},

{'id': 'C002841220P', 'lat': '41.3293', 'lon': '-96.0482'}]

OSM with simple averaging:

[{'id': 128446034, 'lat': 41.3154, 'lon': -96.0525},

{'id': 166090694, 'lat': 41.3181, 'lon': -96.0487},

{'id': 166090695, 'lat': 41.319, 'lon': -96.0475},

{'id': 166316748, 'lat': 41.3202, 'lon': -96.0468},

{'id': 166320250, 'lat': 41.3218, 'lon': -96.0469},

{'id': 166431111, 'lat': 41.3207, 'lon': -96.0478},

{'id': 372070319, 'lat': 41.3211, 'lon': -96.0474},

{'id': 612496367, 'lat': 41.3221, 'lon': -96.0475}]

The process would need to look something like:

**For each NBI bridge:**

**Find corresponding OSM way. (To be solved below)**

**If the way is a bridge, we’re good, add the data**

**If the way is *not* a bridge,**

**Find the nodes before and after this bridge position and make it a new way. This new way should be a bridge with NBI data added.**

But how do we find corresponding ways in OSM? They don’t have definitive coordinates. We can’t really query a location with the OSM handler, either.

This sounds like a job for reverse Geocoding! Check out the OSMnx geocoding module here: <https://osmnx.readthedocs.io/en/stable/osmnx.html#module-osmnx.geocoder>

Reverse geocoding takes a lat/long coordinate and returns the closest relevant way. Once we have the way ID, we can get its tags and edit them with the relevant NBI data. This will be *much* easier than implementing our own search algorithm.

OSMNX allows for calls to Nominatim’s reverse geocoding API, but its rules are strict and we can only make, at most, one call/sec. Using this for all bridges in NBI (628,207 bridges) would take over 174 hours, assuming optimum efficiency. We’ll stick with it for now, but we should swap to a version that we can maintain ([installation here](https://nominatim.org/)).

## OSM Data:

In the future, we may want to push our NBI tags into the OSM database. This may be difficult to approve, but will require us to only merge our data once a year, since it would always be in the native OSM database.

## NBI culverts and “N” bridges:

NBI data has many bridges that are culverts, or bridges with “N” data (there is no data available for the bridge. My hunch is that the culverts are the culprits of the missing data and could likely be ignored, but I will have to research this further.

NBI Data: <https://infobridge.fhwa.dot.gov/Data/>

Field info: <http://nationalbridges.com/nbiDesc.html>

Culverts have essentially zero information listed.

However, field 62, Culverts Condition Rating, has a rating for culverts. Should these be ignored? This seems like it could still be useful. Culverts are causing issues with long ways in OSM

## Merging Process v1

Our merging process is as follows:

1. Gather NBI and OSM data for the same area.
2. Run the program with these downloaded files.
3. First, all NBI data is parsed using PandaDB
   1. For now, we are ignoring culverts in NBI data; they cause issues. We check field 62 (Culvert Condition Rating); It is ok to use if not ‘N’
4. For each bridge, we use Nominatim’s reverse geocoding to find the corresponding OSM object
   1. This should be updated to save the top *n* closest ways.
5. The Nominatim information is stored (quite poorly at the moment) alongside relevant NBI information.
6. An Osmium handler is used to read/write the OSM data.
7. For each way, if that way is a bridge, we edit the tags.
   1. Add the NBI tags in the way
   2. Leave every other data entry as is
8. This gets exported into a new osm/pbf file.
9. Profit!!! (not really)

## Merging v1 Results

It works! Using tqdm in Python, we can see that about 11 bridges are processed/sec. The whole process takes about 45 minutes. It is likely faster on a better PC.

We merge culvert-less NBI data with rich OSM data. While merging, we ignore pedestrian (footway) bridges. We do not worry about culvert data in OSM, as we dropped it from our NBI set.

While merging, certain bridges are ignored:

* In OSM, bridges with the "footway" tag are ignored.
* In NBI, culvert bridges are ignored.

For NBI in NE:

* There are 15,336 bridges.
* 4,214 of those bridges are culverts and are ignored
* There are 11,122 useful candidates

In the merged OSM file:

* There are 11,061 ways with *bridge* tags
* There are 4,483 ways with NBI tags
* 672 of the OSM bridges are "footway" and are ignored while merging.
* 4,483/10,389 potential bridge candidates are filled, culverts are not considered in this number, as OSM does not define them well.

Huge discrepancies. How can this happen?

* Obviously, there are more NBI bridges than in OSM for NE (about 4,000 more).
* We have 11,122 useful NBI candidates and 10,389 OSM bridge ways, not removing culverts from OSM data.
* It is likely the original OSM data is missing many bridges.
* It is possible Nominatim is not returning results for some bridges, but this is unlikely considering the lack of errors.

What to do:

* First, create debugging code for the OSM handler. We need to see:
  + When a footway bridge is ignored
  + When an OSM way matches a stored one
* Then, debug and explore Nominatim results
  + Check if first closest is bridge & what it is
  + If it isn’t a bridge, look at other closest results
* **The most likely scenario is rev-geo’ing NBI data that isn’t listed as a bridge in OSM**
* When merging, we should force apply NBI tags to a way if:
  + There is no "bridge" tag
  + There is no "highway=footway"
  + There is no "tunnel=culvert"
* We may want to consider the length of the way, passing those that have >2 nodes, being a road rather than a bridge.
* Try to use multiplexing to get Nominatim calls faster