osm2gmns

Release 0.2.0

Jiawei Lu, Xuesong (Simon) Zhou

May 26, 2021

CONTENTS

Author: Jiawei Lu, Xuesong (Simon) Zhou
Email: jiaweil9@asu.edu, xzhou74@asu.edu

OpenStreetMap (OSM) is a free, open-source, editable map website that can provide free downloads. osm2gmns, as a data conversion tool, can directly convert the OSM map data to node and link network files in the GMNS format. Users can convert and model drivable, walkable, railway, or aeroway networks with a single line of Python code.

CHAPTER

ONE

CONTENTS

1.1 Installation

You can install the latest release of osm2gmns at PyPI via pip:

```
pip install osm2gmns
```

After running the command above, the osm2gmns package along with three necessary dependency packages (Shapely, pandas and protobuf) will be installed to your computer (if they have not been installed yet).

If you install osm2gmns in a conda environment, you may get an error message: “OSError: [WinError 126] The specified module could not be found” when importing osm2gmns. To resolve this issue, you need to uninstall the Shapely package first, and reinstall it manually using the command below.

```
conda install shapely
```

1.2 GMNS

GMNS (General Modeling Network Specification), proposed by the Zephyr Foundation, which aims to advance the field through flexible and efficient support, education, guidance, encouragement, and incubation.

The two necessary files used in GMNS to describe a network: `node.csv` and `link.csv`.

- node.csv

The node file is a list of vertices that locate points on a map. Typically, they will represent intersections, but may also represent other points, such as a transition between divided and undivided highway[1]. We add several additional attributes to the node file to make it more suitable for transportation modelling. Detailed node data dictionary is listed below.

Field	Type	Required?	Comments
name	string		
node_id	int	yes	unique key
osm_node_id	string or int		corresponding point id in osm data
osm_highway	string		point type in osm data
zone_id	int		
ctrl_type	int		1: Signalized; 0: not
node_type	string		
activity_type	string		defined by adjacent links
is_boundary	bool		1: boundary; 0: not
x_coord	double	yes	WGS 84 is used in osm
y_coord	double	yes	WGS 84 is used in osm
main_node_id	int		nodes belonging to one complex intersection have the same id
poi_id	int		id of the corresponding poi

- link.csv

A link is an edge in a network, defined by the nodes it travels from and to. It may have associated geometry information[2]. Similar to node.csv, We also added several new attributes to the link file. Detailed link data dictionary is listed below.

Field	Type	Required?	Comments
name	string		
link_id	int	yes	unique key
osm_way_id	string or int		corresponding way id in osm data
from_node_id	int	yes	
to_node_id	int	yes	
dir_flag	enum		1: forward, -1: backward, 0:bidirectionial
length	float		unit: meter
lanes	int		
free_speed	float		
capacity	float		unit: veh/hr/lane
link_type_name	string		
link_type	int		
geometry	Geometry		wkt
allowed_uses	enum		auto, bike, walk
from_biway	bool		1: link created from a bidirectional way, 0: not

Other two optional files including `movement.csv` and `segement.csv` follow the exact same format as what being defined in the GMMS standard. Readers can check the GMNS website for details.

In addition to the above files defined in the GMNS standard, osm2gmns can also produce `poi.csv` files where point of interest information is stored. Detailed poi data dictionary is listed below.

Field	Type	Required?	Comments
name	string		
poi_id	int	yes	unique key
osm_way_id	string or int		corresponding way id in osm data
osm_relation_id	string or int		corresponding relation id in osm data
building	string		building tag in osm data
amenity	string		amenity tag in osm data
geometry	Geometry	yes	wkt
centroid	Geometry		wkt

[1] https://github.com/zephyr-data-specs/GMNS/blob/master/Specification/Node.md

[2] https://github.com/zephyr-data-specs/GMNS/blob/master/Specification/Link.md

1.3 Quick Start

In this section, some examples are provided to demonstrate how to use osm2gmns to generate, manipulate and output networks.

1.3.1 Download OSM Data

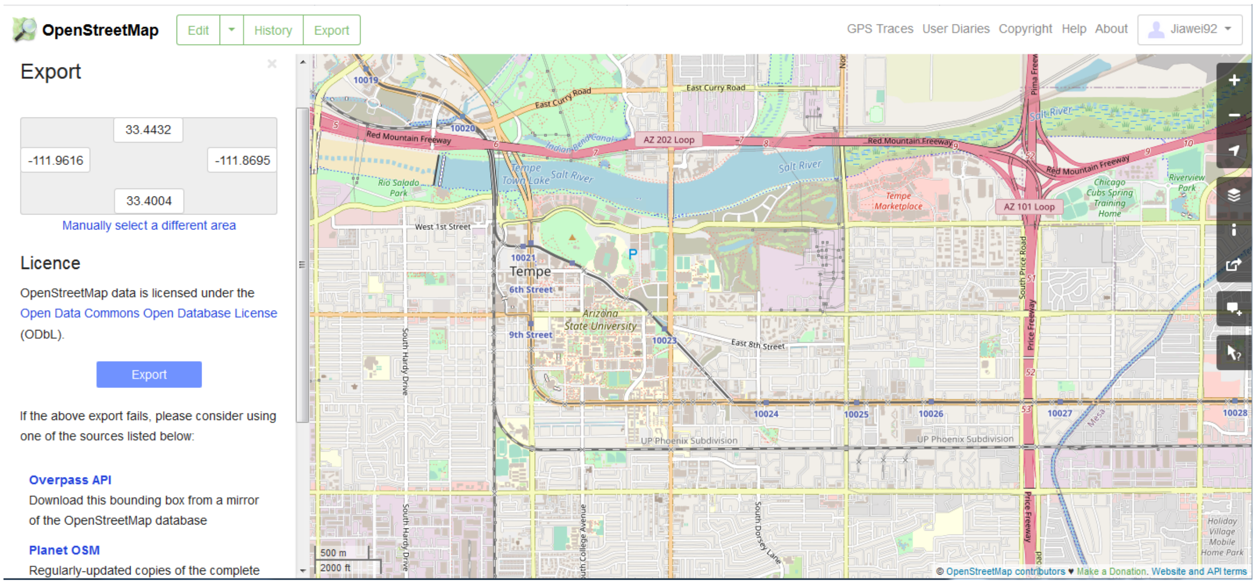
To reduce uncertainties while directly parsing network data from the osm server via APIs, osm2gmns uses downloaded osm files to extract useful network information. As a result, the first step is preparing osm files.

Thanks to the open-source nature of OpenStreetMap, there are lots of APIs and mirror sites that we can use to download osm map data. We list several popular sites here for users to choose.

1) OpenStreetMap Homepage

On OpenStreetMap homepage, click the `Export` button to enter Export mode. Before downloading, you may need to span and zoom in/out the map to make sure that your target area is properly shown on the screen. Or, you can use `Manually select a different area` to select your area more precisely. Click the `Export` button in blue to export the network you want.

Note that if the target area is too large, you may get an error message: "You requested too many nodes (limit is 50000). Either request a smaller area, or use planet.osm". In this case, you can always click `Overpass API` to download the network you need via a mirror site.



Export

33.4432

-111.9616 -111.8695

33.4004

Manually select a different area

Licence

OpenStreetMap data is licensed under the Open Data Commons Open Database License (ODbL).

Export

If the above export fails, please consider using one of the sources listed below:

Overpass API
Download this bounding box from a mirror of the OpenStreetMap database

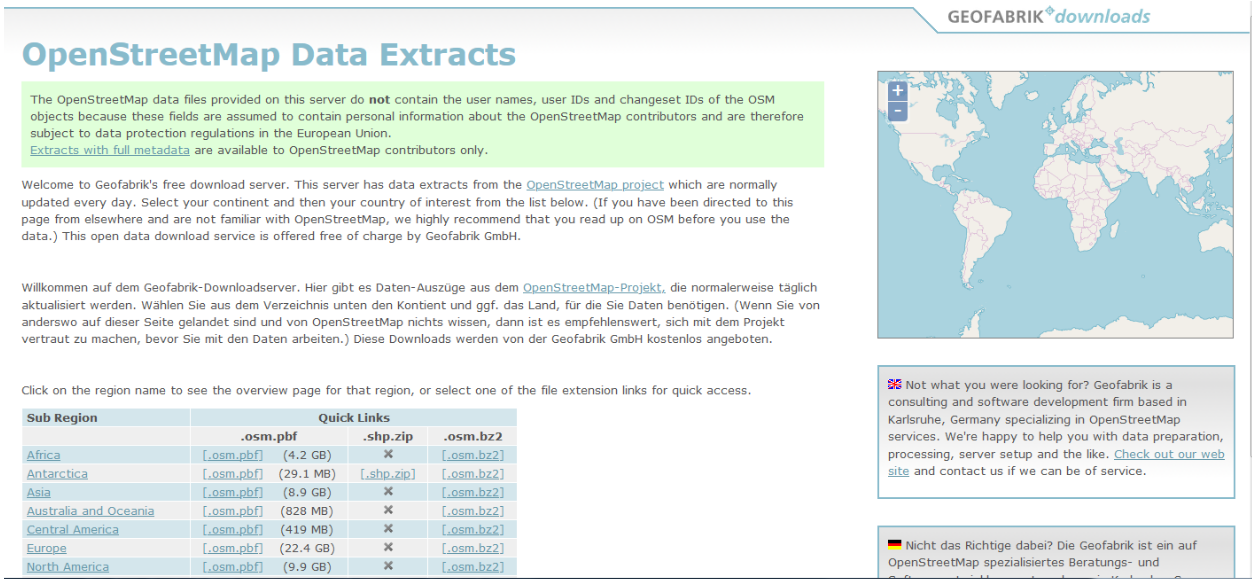
Planet OSM
Regularly-updated copies of the complete

Fig. 1: Download osm data from OpenStreetMap homepage

2) Geofabrik

Different from the way of downloading map data from OpenStreetMap homepage, Geofabrik enables you to download network data for administrative areas. On OpenStreetMap homepage, we can only download areas defined by rectangles. In Geofabrik, you can click the corresponding quick link of your interested region to download the map data you need. You can always click the name of regions to check if sub region data are available.

Generally, there are three types of file format for users to choose when downloading map data. osm2gmns supports `.pbf` and `.osm` files. In osm2gmns, networks stored in `.osm` files are parsed quickly than those stored in `.pbf` files. However, compared with `.pbf` files, `.osm` files take much more hard disk space to store networks and much more space in RAM while parsing.



OpenStreetMap Data Extracts

The OpenStreetMap data files provided on this server do **not** contain the user names, user IDs and changeset IDs of the OSM objects because these fields are assumed to contain personal information about the OpenStreetMap contributors and are therefore subject to data protection regulations in the European Union.
Extracts with full metadata are available to OpenStreetMap contributors only.

Welcome to Geofabrik's free download server. This server has data extracts from the OpenStreetMap project which are normally updated every day. Select your continent and then your country of interest from the list below. (If you have been directed to this page from elsewhere and are not familiar with OpenStreetMap, we highly recommend that you read up on OSM before you use the data.) This open data download service is offered free of charge by Geofabrik GmbH.

Willkommen auf dem Geofabrik-Downloadserver. Hier gibt es Daten-Auszüge aus dem OpenStreetMap-Projekt, die normalerweise täglich aktualisiert werden. Wählen Sie aus dem Verzeichnis unten den Kontient und ggf. das Land, für die Sie Daten benötigen. (Wenn Sie von anderswo auf dieser Seite gelandet sind und von OpenStreetMap nichts wissen, dann ist es empfehlenswert, sich mit dem Projekt vertraut zu machen, bevor Sie mit den Daten arbeiten.) Diese Downloads werden von der Geofabrik GmbH kostenlos angeboten.

Click on the region name to see the overview page for that region, or select one of the file extension links for quick access.

Sub Region	Quick Links			
	.osm.pbf		.shp.zip	.osm.bz2
Africa	[.osm.pbf]	(4.2 GB)	✗	[.osm.bz2]
Antarctica	[.osm.pbf]	(29.1 MB)	[.shp.zip]	[.osm.bz2]
Asia	[.osm.pbf]	(8.9 GB)	✗	[.osm.bz2]
Australia and Oceania	[.osm.pbf]	(828 MB)	✗	[.osm.bz2]
Central America	[.osm.pbf]	(419 MB)	✗	[.osm.bz2]
Europe	[.osm.pbf]	(22.4 GB)	✗	[.osm.bz2]
North America	[.osm.pbf]	(9.9 GB)	✗	[.osm.bz2]

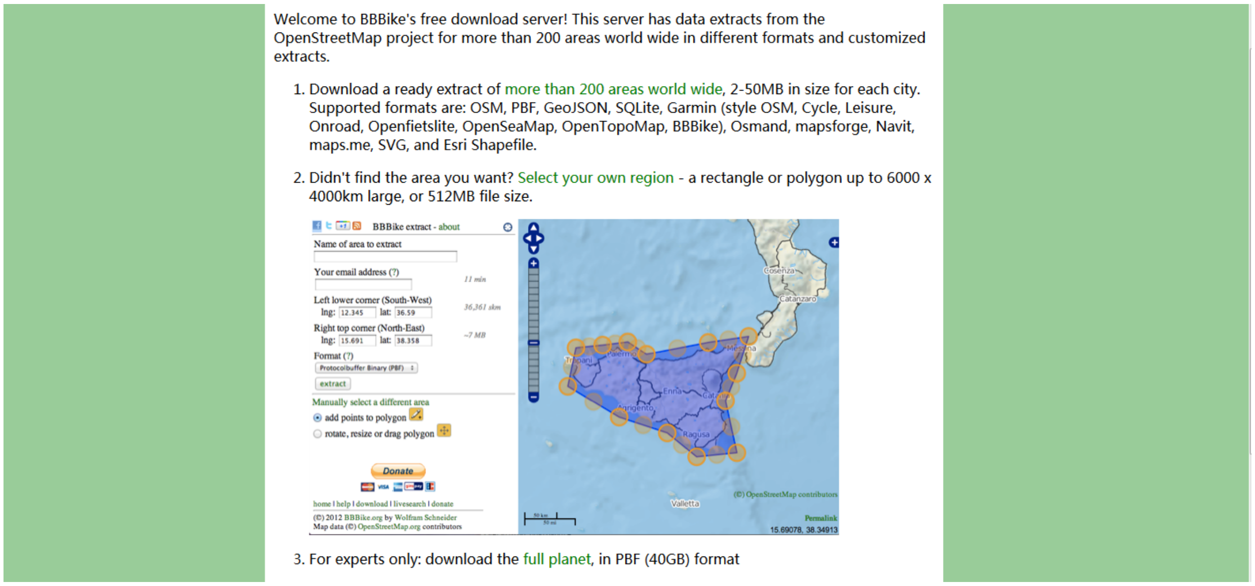
Not what you were looking for? Geofabrik is a consulting and software development firm based in Karlsruhe, Germany specializing in OpenStreetMap services. We're happy to help you with data preparation, processing, server setup and the like. Check out our web site and contact us if we can be of service.

Nicht das Richtige dabei? Die Geofabrik ist ein auf OpenStreetMap spezialisiertes Beratungs- und

Fig. 2: Download osm data from Geofabrik

3) BBBike

If your target area is neither an administrative region nor a rectangle, BBBike may be a good choice. BBBike enables you to select your region using a polygon. BBBike supports numerous file formats to output and store network data. Users can select a proper one according to their requirements.



Welcome to BBBike's free download server! This server has data extracts from the OpenStreetMap project for more than 200 areas world wide in different formats and customized extracts.

1. Download a ready extract of more than 200 areas world wide, 2-50MB in size for each city. Supported formats are: OSM, PBF, GeoJSON, SQLite, Garmin (style OSM, Cycle, Leisure, Onroad, Openfietslite, OpenSeaMap, OpenTopoMap, BBBike), Osmand, mapsforge, Navit, maps.me, SVG, and Esri Shapefile.

2. Didn't find the area you want? Select your own region - a rectangle or polygon up to 6000 x 4000km large, or 512MB file size.

BBBike extract - about

Name of area to extract

Your email address (?)

Left lower corner (South-West)
lng: 12.345 lat: 36.59

Right top corner (North-East)
lng: 15.691 lat: 38.358

Format (?)
Protocolbuffer Binary (PBF)

extract

11 min
36,361 skm
~7 MB

Manually select a different area
● add points to polygon
○ rotate, resize or drag polygon

home | help | download | livesearch | donate

(©) 2012 BBBike.org by Wolfram Schneider
Map data (©) OpenStreetMap.org contributors

3. For experts only: download the full planet, in PBF (40GB) format

Fig. 3: Download osm data from BBBike

1.3.2 Parse OSM Data

We use the region around Arizona State University, Tempe Campus in this guide to introduce some major functions in osm2gmns.

Obtain a transportation network from an osm file.

```
>>> import osm2gmns as og

>>> net = og.getNetFromOSMFile('asu.osm')
>>> # we recommend using getNetFromPBFFile() for large networks
>>> # net = og.getNetFromPBFFile('***.osm.pbf')
```

Note:

- getNetFromPBFFile() is supported in release (0.2.0) or later.

A link will be included in the network file from osm database if part of the link lies in the region that users selected. If argument strict_mode (default: True) is set as True, link segments that outside the region will be cut off when parsing osm data. If argument strict_mode is set as False, all links in the network file will be imported.

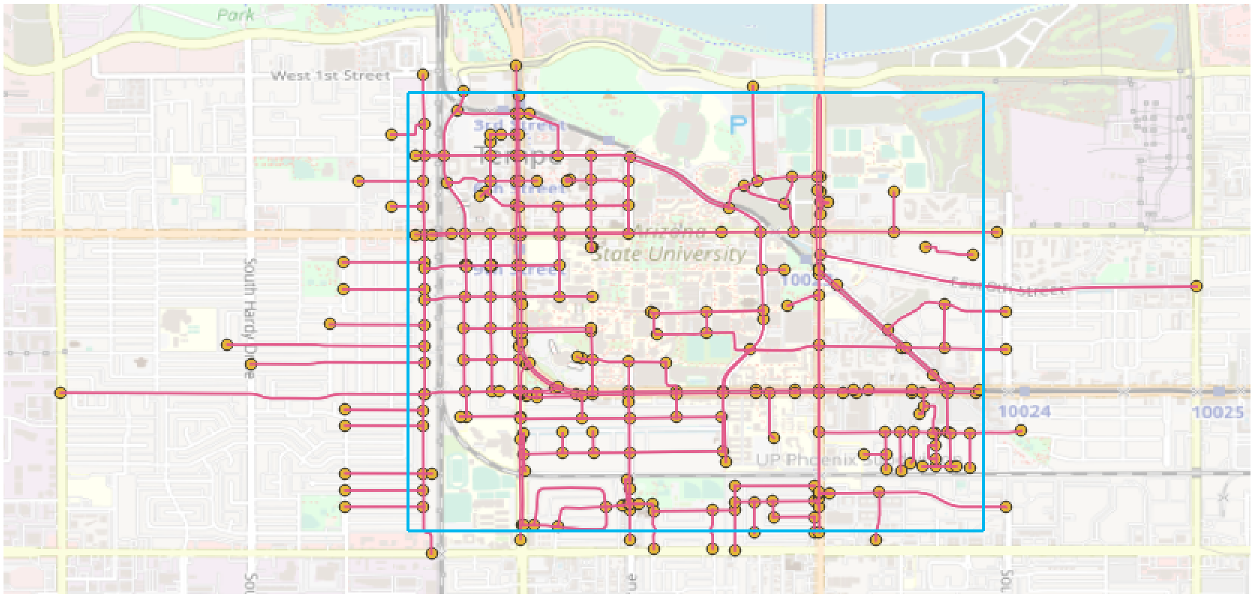


Fig. 4: Parsed network with strict_mode=False

One loaded network may contain several sub networks, with some sub networks are not accessible from others. In most cases, these sub networks include a large sub network and some isolated nodes or links. When the number of nodes of a sub network is less than argument min_nodes (default: 1), this sub network will be discarded.

Users can use argument combine (default: False) to control short link combinations. If combine is enabled, two-degree nodes (one incoming link and one outgoing link) will be removed, and two adjacent links will be combined to generate a new link.

Noticed that most links do not have “lanes” information in the map data provided by OpenStreetMap. Thus, we use a default lanes dictionary for each link type in osm2gmns. By setting default_lanes (default: False) as True, the default value will be assigned to a link if it does not come with “lanes” information. The default dictionary in osm2gmns:

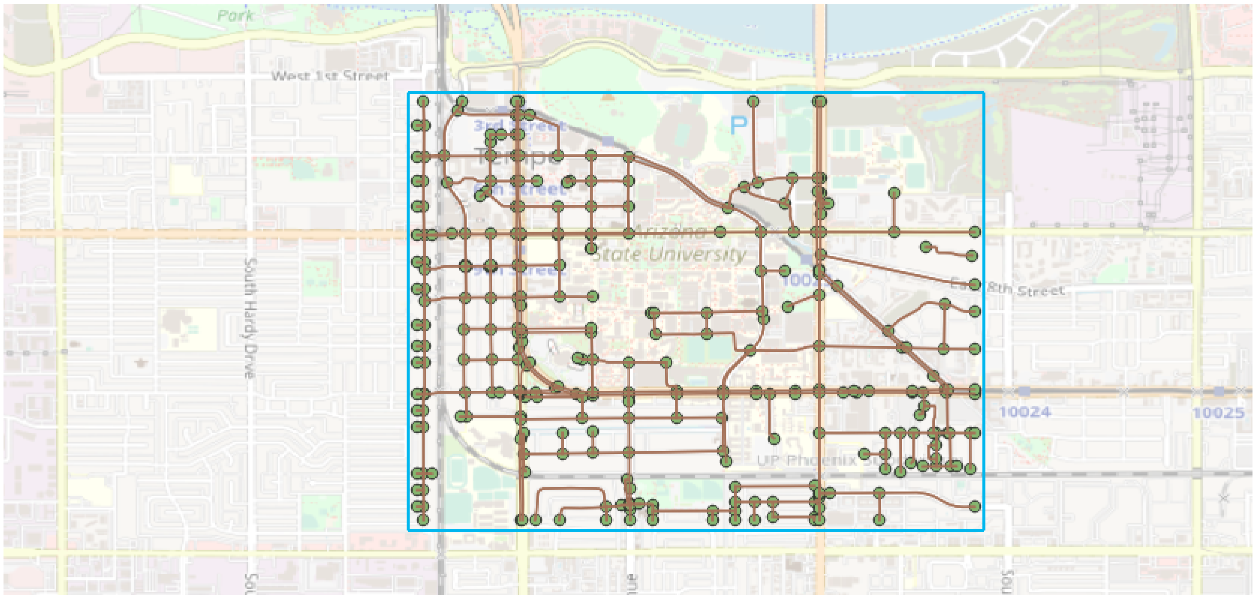


Fig. 5: Parsed network with `strict_mode=True`

```
default_lanes_dict = {'motorway': 4, 'trunk': 3, 'primary': 3, 'secondary': 2, 'tertiary
↪': 2,
                      'residential': 1, 'service': 1, 'cycleway':1, 'footway':1, 'track
↪':1,
                      'unclassified': 1, 'connector': 2}
default_speed_dict = {'motorway': 59, 'trunk': 39, 'primary': 39, 'secondary': 39,
↪'tertiary': 29,
                      'residential': 29, 'service': 29, 'cycleway':9, 'footway':4, 'track
↪':29,
                      'unclassified': 29, 'connector':59}
```

`default_lanes` also accepts a dictionary. In that case, osm2gmns will use the dictionary provided by users to update the default dictionary.

A similar fashion applies for argument `default_speed`.

1.3.3 Output Networks to CSV

Based on the `net` instance obtained from the last step, `outputNetToCSV` can be used to output the parsed network to CSV files.

```
>>> og.outputNetToCSV(net)
```

Users can use argument `output_folder` to specify the folder to store output files. Node information will be written to `node.csv`, while link information will be written to `link.csv`.

If argument `combine` is set as `True` when parsing the network, `segment.csv` will also be created to store lane changes in links. Lane changes occur when combining two adjacent links with different lanes in the combination step.

1.3.4 Consolidate Intersections

In OpenStreetMap, one large intersection is often represented by multiple nodes. This structure brings some difficulties when conducting traffic simulations (hard to model traffic signals in these intersections). osm2gmns enables users to consolidate intersections while parsing networks, i.e., generate a new node to replace existing nodes for each large intersection.

```
>>> net = og.getNetFromOSMFile('asu.osm')
>>> og.consolidateComplexIntersections(net)
>>> og.outputNetToCSV(net)
```

When executing function `getNetFromOSMFile`, osm2gmns will automatically identify complex intersections based on the argument `int_buffer` (defalut: `20.0`). Nodes that belong to one complex intersection will be assigned with the same `main_node_id`, but these nodes will not be consolidated into one node unless function `consolidateComplexIntersections` is called.

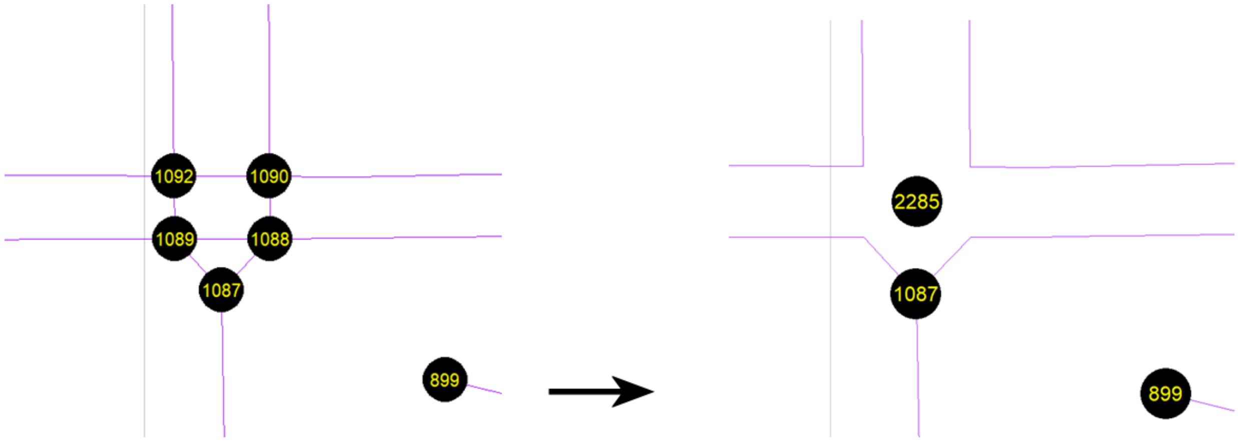


Fig. 6: Complex intersection consolidation

Users can also check and revise complex intersection identification results first, then conduct the consolidating operation to obtain more reasonable outcomes.

```
>>> net = og.getNetFromOSMFile('asu.osm')
>>> og.outputNetToCSV(net)
>>> # check the main_node_id column in node.csv
>>> net = og.getNetFromCSV()
>>> og.consolidateComplexIntersections(net)
>>> og.outputNetToCSV(net, output_folder='consolidated')
```

1.3.5 Network Types and POI

osm2gmns supports five different network types, including `auto`, `bike`, `walk`, `railway`, `aeroway`. Extract the auto and railway network from an osm file by setting `network_type` (default: `(auto,)`) as `(auto,railway)`:

```
>>> net = og.getNetFromOSMFile('asu.osm', network_type=('auto','railway','aeroway'))
```

Obtain POIs (Point of Interest) from osm map data.

```
>>> net = og.getNetFromOSMFile('asu.osm', POIs=True)
```

If `POIs` (default: `False`) is set as `True`, a file named `poi.csv` will be generated when outputting a network using function `outputNetToCSV`.

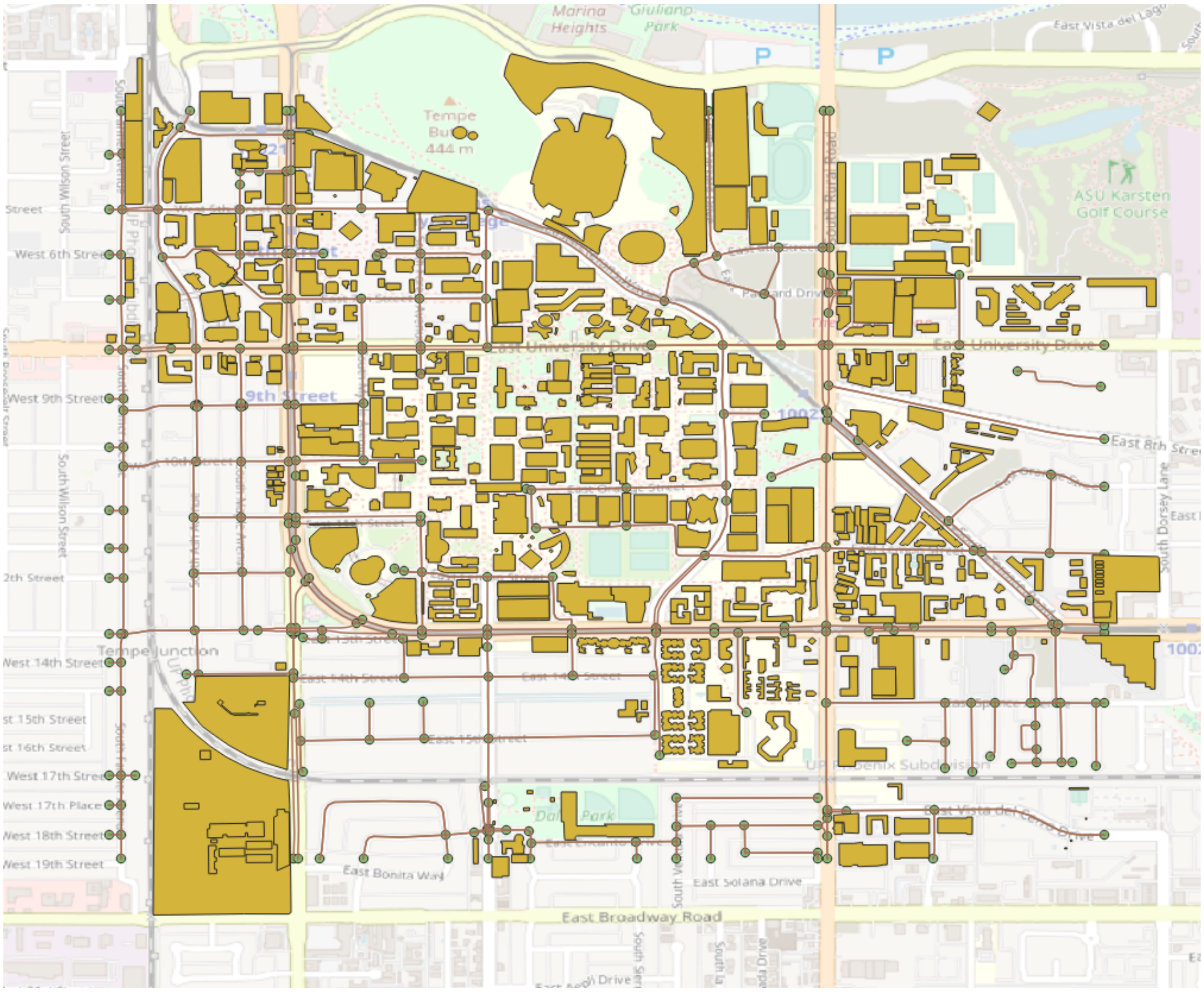


Fig. 7: Network with POIs

Connect POIs with transportation network.

```
>>> net = og.getNetFromOSMFile('asu.osm', POIs=True)
>>> og.connectPOIWithNet(net)
```

By using function `connectPOIWithNet`, a node located at the centroid of each POI will be generated to represent the POI, then connector links will be built to connect the POI node with the nearest node in the transportation network.

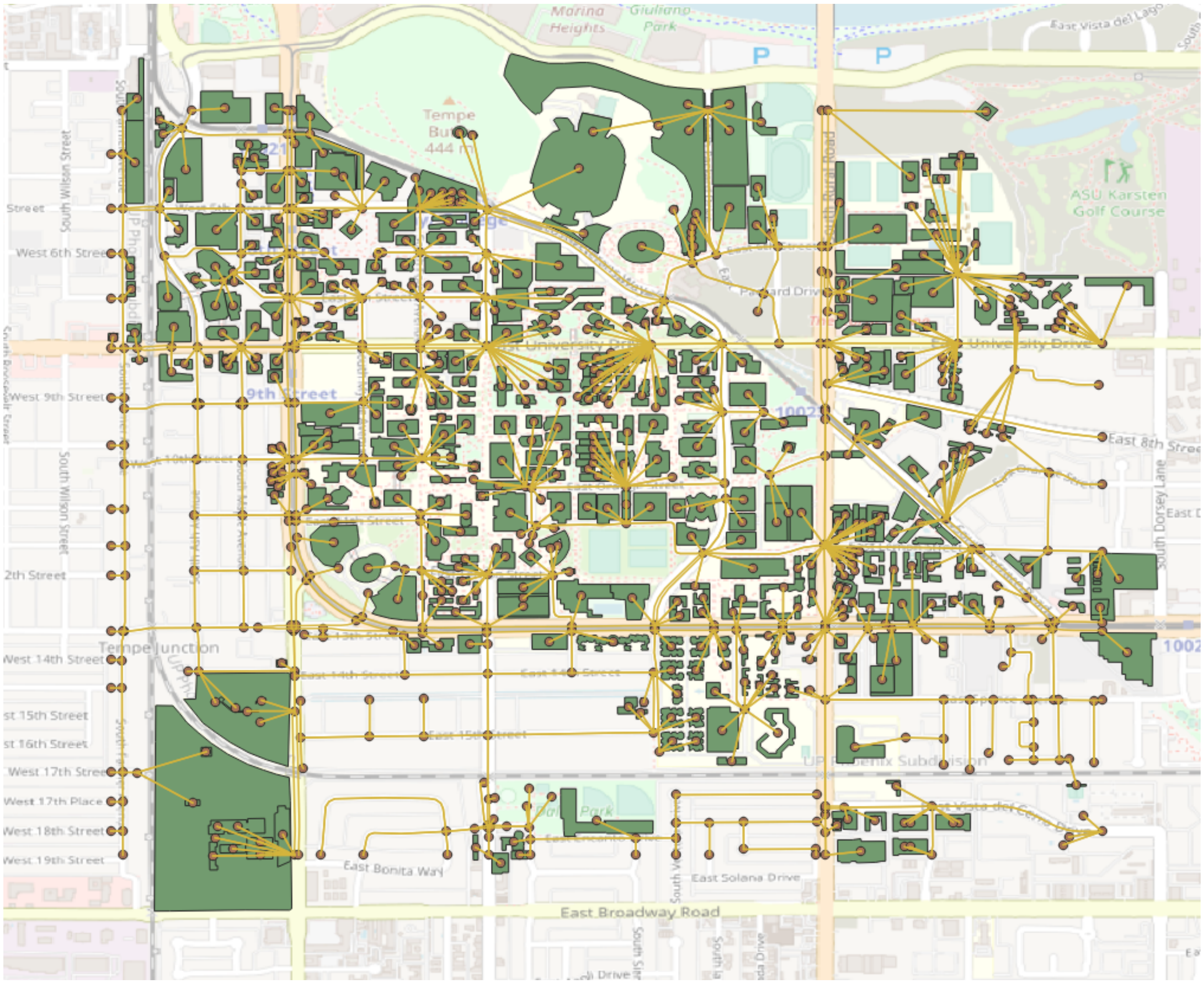


Fig. 8: Connect POIs with network

1.4 Modules

osm2gmns includes the following 12 modules:

`activity.py`	generate activity information for nodes
`classes.py`	class definitions
`combine_links.py`	combine short links
`complex_intersection.py`	identify complex intersections
`consolidate_intersections.py`	consolidate complex intersections
`network.py`	process network elements
`pois.py`	process POIs (point of interest)
`readfile.py`	load network file from hard disk
`settings.py`	default settings for osm2gmns
`util.py`	base functions used in all modules
`wayfilters.py`	filter logic for highways in osm
`writefile.py`	write network data to hard disk

1.5 Sample Networks

1.5.1 Phoenix Sky Harbor International Airport

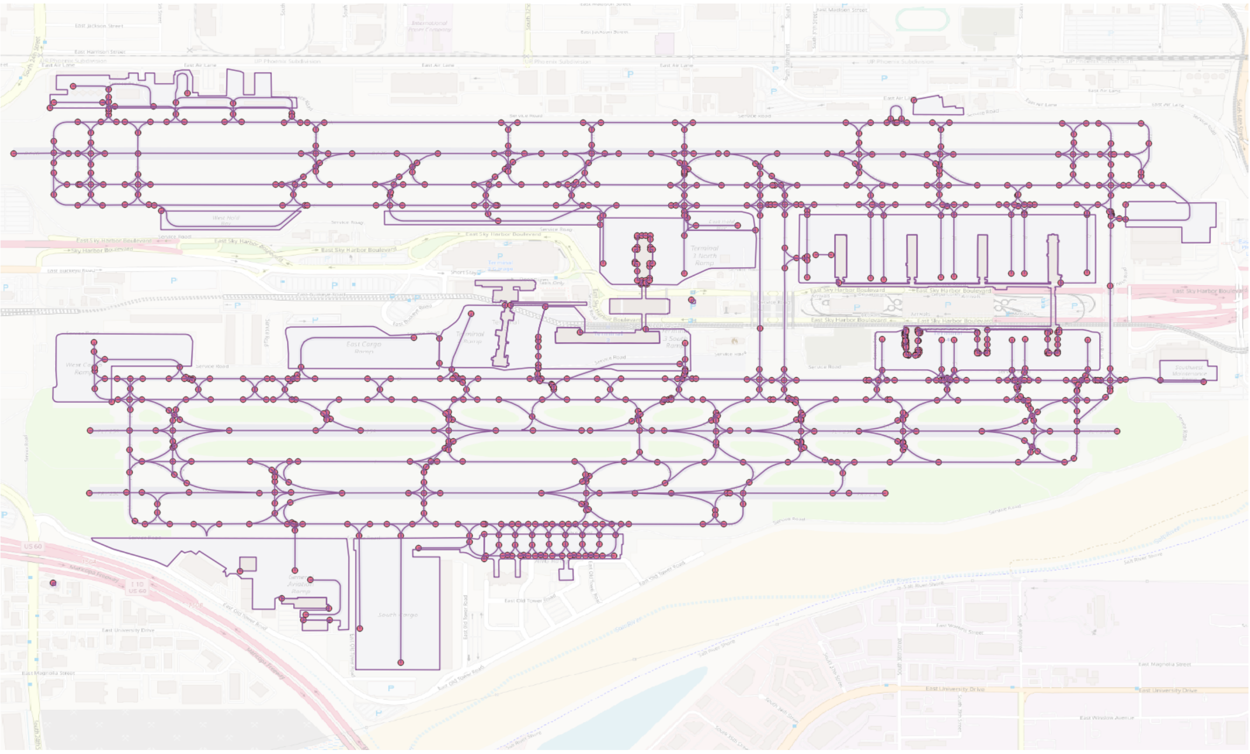


Fig. 9: Phoenix Sky Harbor International Airport

1.5.2 Arizona State University, Tempe Campus

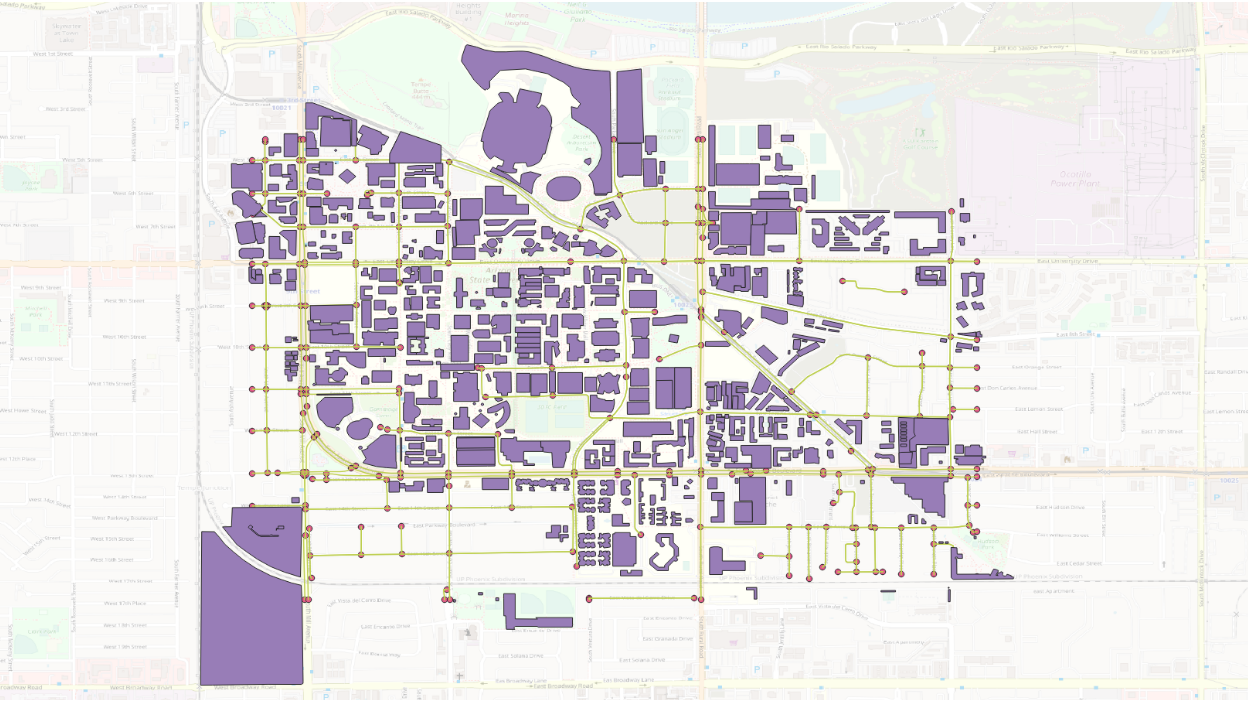


Fig. 10: Arizona State University, Tempe Campus

1.5.3 Arizona, US

1.5.4 US railway network (midwest)

1.5.5 Greater London, UK

1.5.6 Melbourne, Australia

1.5.7 Nanjing, Jiangsu, China

1.5.8 Yangzhou, Jiangsu, China

1.6 Acknowledgement

This project is partially supported by National Science Foundation – United States under Grant No. CMMI 1663657 "Collaborative Research: Real-time Management of Large Fleets of Self-Driving Vehicles Using Virtual CyberTracks"

The second author also thanks for the early support from FHWA project titled "The Effective Integration of Analysis, Modeling, and Simulation Tools-AMS DATA HUB CONCEPT OF OPERATIONS". https://www.fhwa.dot.gov/publications/research/operations/13036/004.cfm

This document is prepared with the help from Entai Wang.

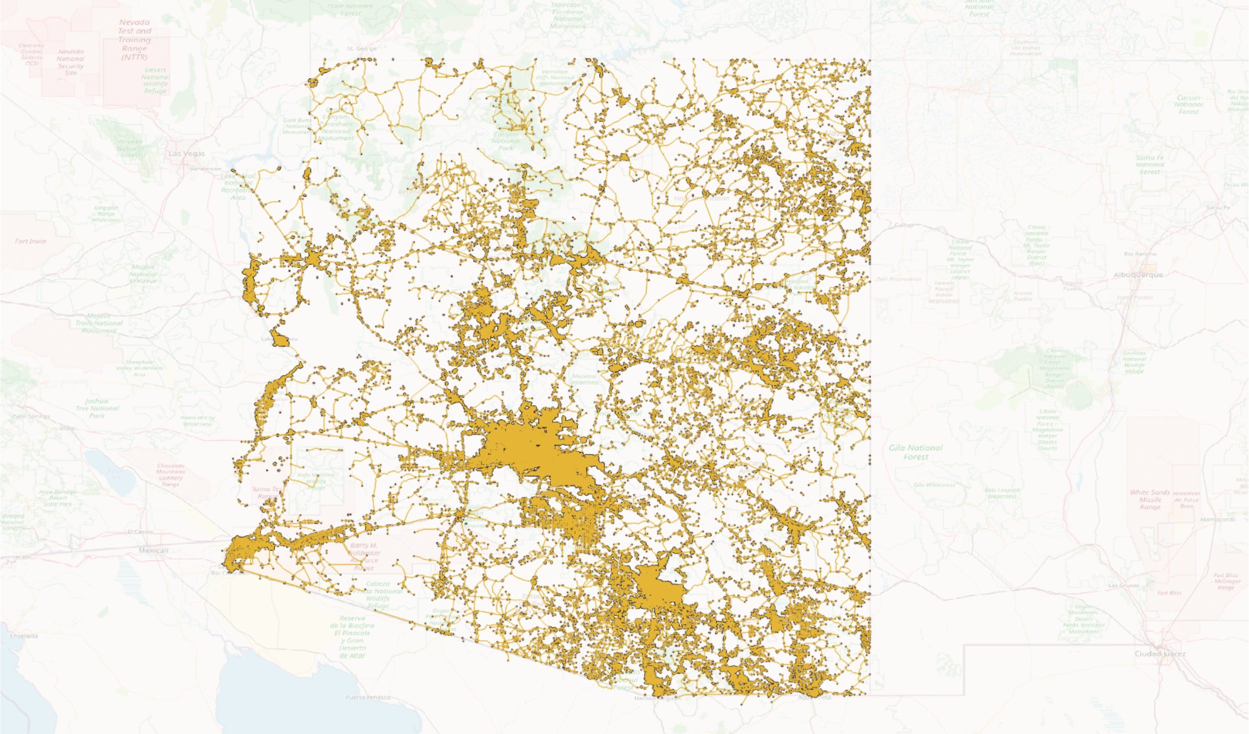


Fig. 11: Arizona, US

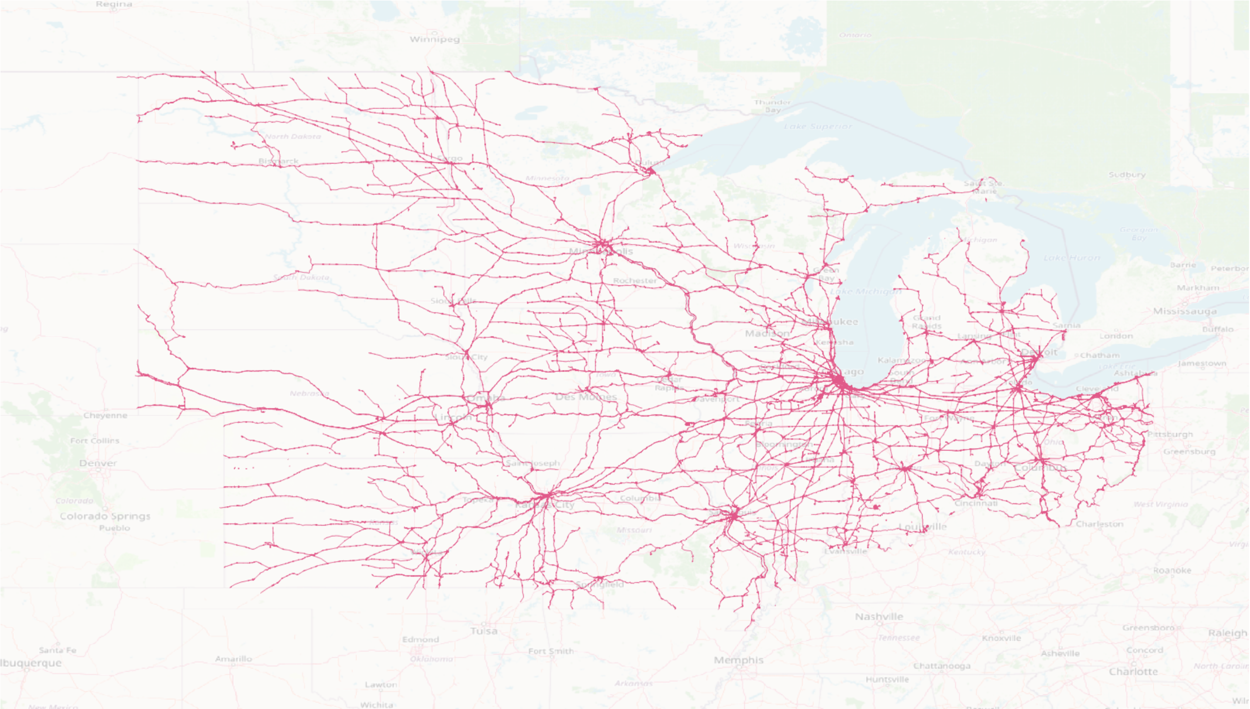


Fig. 12: US railway network (midwest)

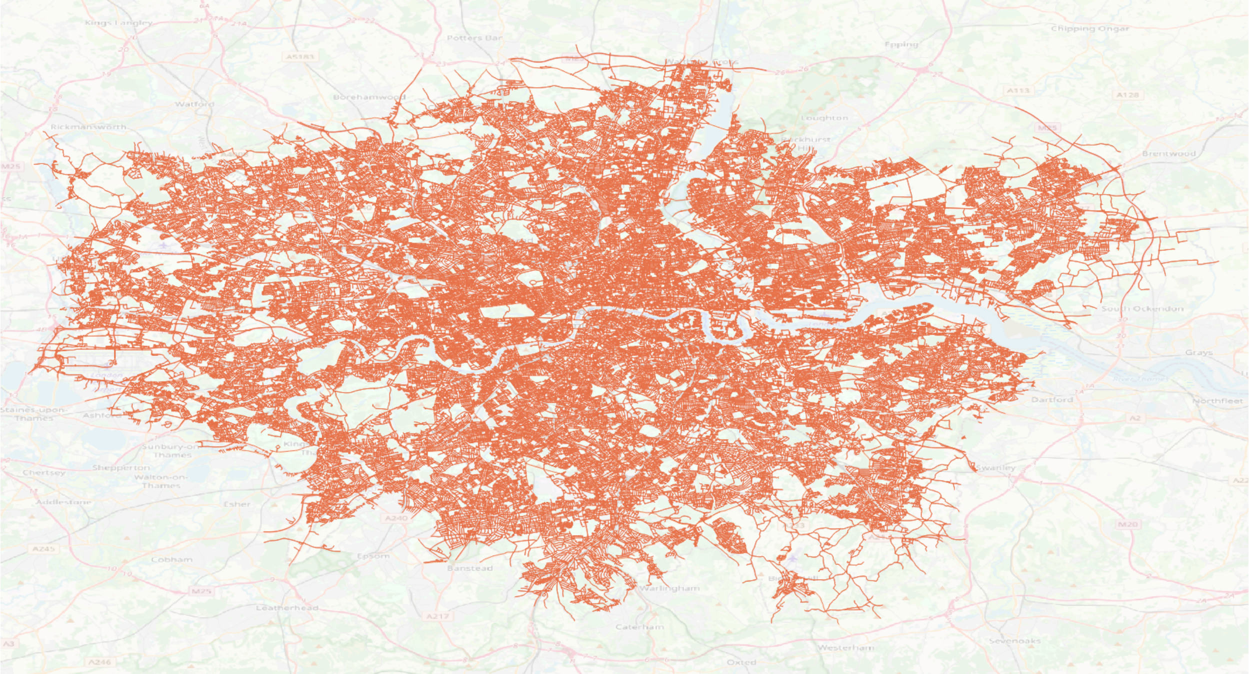


Fig. 13: Greater London, UK

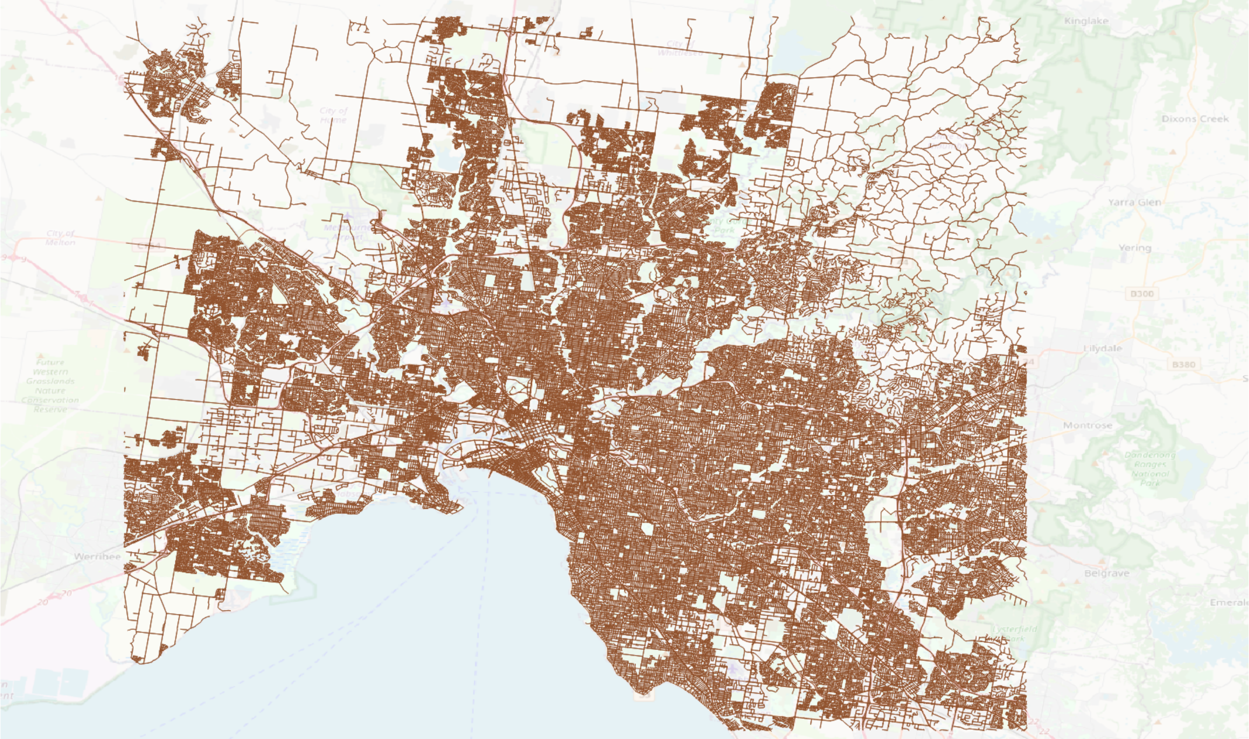


Fig. 14: Melbourne, Australia

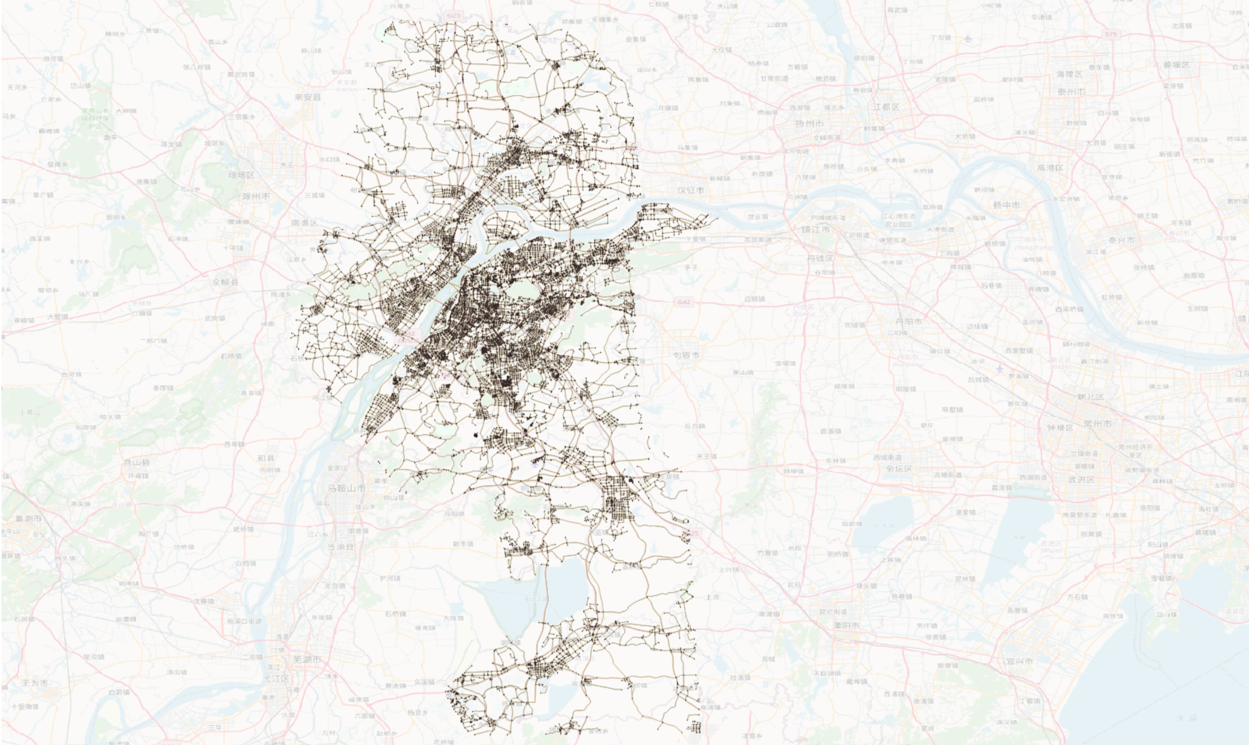


Fig. 15: Nanjing, Jiangsu, China

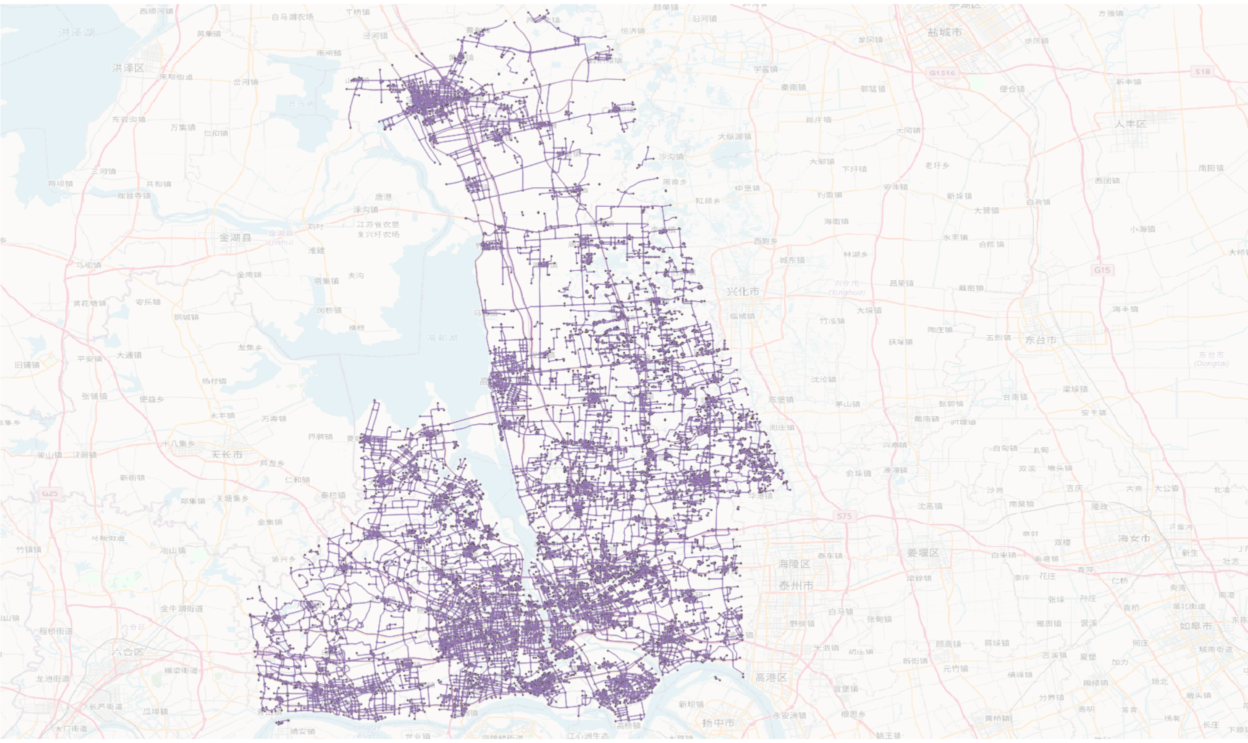


Fig. 16: Yangzhou, Jiangsu, China

For program source code and sample network files, readers can visit the project homepage at ASU Trans+AI Lab Github. Interested readers can also check the link for our online transportation modelling visualization platform, in which network data is provided by osm2gmns.