OSM2Rail

An open-source education tool for constructing modeling datasets of railway transportation

Authors:

Jiawei Lu: Arizona State University

Dr. Qian Fu: Birmingham Centre for Railway Research and Education, University of Birmingham

Zanyang Cui: Beijing JiaoTong University

Dr. Junhua Chen: Beijing JiaoTong University

Email :

jiaweil9@asu.edu, qian.fu@outlook.com,

zanyangcui@outlook.com, cjh@bjtu.edu.cn

1. Introduction

The python tool of OSM2Rail is an integrated and enhanced version of two python packages, namely OSM2GMNS (https://pypi.org/project/osm2gmns/) and Pydriosm (https://pypi.org/project/pydriosm/). The former aims to convert OpenStreetMap (https://www.openstreetmap.org/) data to generic node and link files in GMNS format, while the latter aims to enable a batch process of downloading, reading and PostgreSQL-based I/O of OpenStreetMap data.

OSM2GMNS is currently developed and maintained by Jiawei Lu and Dr. Xuesong Zhou at Arizona State University. PyDriosm published by Dr. Qian Fu at Birmingham Centre for Railway Research and Education, University of Birmingham is an open-source tool for researchers or practitioners to easily download and read OSM map data in popular file formats such as protocolbuffer binary format (PBF) and shapefile, which are available for free download from Geofabrik and BBBike. This package also provides a convenient way for PostgreSQL-based I/O and storage of

parsed OSM data.

Integrating the data conversion and online data downloading capabilities from the above 2 packages, OSM2Rail allows users to rapidly obtain OSM data for a given set of areas and convert them to node, link, and poi files for further system modeling and optimization. Users are recommended to download OSM map data in .osm or .osm.pbf format and convert them to commonly used csv files.

2. Data Format

To fully utilize open data sources from OSM, OSM2Rail extracts most of fields related to railway infrastructure. The output csv files are using an extended version of the base GMNS formation by adding many rail-related attributes. In this preliminary version, we will call the extended format as GMNS-rail, and we expect other community members to contribute in standardizing such data specifications for broader applications of rail operations and management.

node.csv

name	GMNS
node_id	GMNS
x_coord	GMNS
y_coord	GMNS
geometry	GMNS
osm_node_id	GMNS
railway	Extended
level_crossing	Extended
access	Extended
description	Extended

link.csv

name	GMNS
link_id	GMNS
osm_way_id	GMNS

from_node_id	GMNS
to_node_id	GMNS
link_type_name	GMNS
length	GMNS
geometry	GMNS
railway	GMNS
electrified	Extended
frequency	Extended
highspeed	Extended
max_speed	Extended
maxspeed_designed	Extended
passenger_lines	Extended
railway_ctcs	Extended
railway_traffic_mode	Extended
start_date	Extended
usage	Extended
voltage	Extended
gauge	Extended
service	Extended

poi.csv

name	GMNS
poi_id	GMNS
osm_way_id	GMNS
railway	Extended
geometry	GMNS

3. Using OSM2Rail

There are 5 essential steps, namely installation, downloading, conversion, visualization and output.

Step 1: Installation

```
pip install osm2rail
```

Step 2: Downloading OSM data online

(1) import the package

```
import osm2rail as orl
```

Option 1: downloading from Overpass

parameter list:

subarea_names	str or list of str, the names of target area.
boxs	tuple or list of tuple, the coordinates of target area. (minlat, maxlat, minlon, maxlon)
download_dir	str, directory for saving the downloaded files.
interval_sec	float, interval (in sec) between downloading two subregions.
random_header	bool, whether to go for a random request agent.

The downloader Overpass only allows download .osm files.

(1) download '.osm' files **by given area names**

```
subarea=' london waterloo'
#subarea=[' London waterloo','Washington Union Station']
download_dir='osmfile'
orl.download_osm_data_from_overpass(subarea_names=subarea,download_dir=dow
nload_dir,random_header=False)
```

(2) download '.osm' files **by given area coordinates**

```
box=(38.90984,38.91721,-77.00492,-76.98808)
#    box=[(38.90984,38.91721,-77.00492,-76.98808),(38.80068,38.80805,-77.09378,-
77.07693)]
download_dir='osmfile'
```

```
orl.download_osm_data_from_overpass(boxs=box,download_dir=download_dir,random_header=True)
```

Option 2: downloading from Geofabrik

parameter list:

subregion_names	str or list of str, the names of target area.
osm_file_format	str, file format of the OSM data available on the free download server, defaults to .osm.pbf.
download_dir	str, directory for saving the downloaded files, defaults to None.
interval_sec	float, interval (in sec) between downloading two subregions, defaults to 10 s.
verbose	bool, whether to print relevant information in console, defaults to False.
update	bool, whether to update the data if it already exists, defaults to False.
confirmation_required	bool, whether asking for confirmation to proceed, defaults to True.
ret_download_path	whether to return the path(s) to the downloaded file(s), defaults to False.
random_header	bool, whether to go for a random request agent, defaults to False.(suggested when downloading more than one files)

The downloader Geofabrik allows a wide range of file formats for downloading, including ‘.osm.pbf ’, ‘.shp.zip ’, and ‘.osm.bz2 ’.

(1) download ‘.osm.pbf ’ file by given region names

```
subregion='delaware'
# subregion=['delaware','london']
download_dir='osmfile'
osm_file_format='.pbf'
```

```
orl.download_osm_data_from_geofabrik(subregion_names=subregion,osm_file_format=osm_file_format,download_dir=download_dir,

random_header=False,verbose=True,confirmation_required=False)
```

(2) download ‘.shp.zip ’ file by given region names

```
subregion='delaware'
# subregion=['delaware','london']
download_dir='osmfile'
osm_file_format='.shp'
orl.download_osm_data_from_geofabrik(subregion_names=subregion,osm_file_format=osm_file_format,download_dir=download_dir,

random_header=True,verbose=True,confirmation_required=False)
```

(3) download ‘.osm.zip ’ file by given region names

```
subregion='delaware'
# subregion=['delaware','london']
download_dir='osmfile'
osm_file_format='.osm.zip'
orl.download_osm_data_from_geofabrik(subregion_names=subregion,osm_file_format=osm_file_format,download_dir=download_dir,

random_header=True,verbose=True,confirmation_required=False)
```

Option 3: downloading from BBBike

The parameters of BBBike downloader are the same as Geofabrik API.

The downloader BBBike allows a list of possible formats in addition to ‘.osm.pbf ’, ‘.shp.zip ’, and ‘.osm.bz2 ’.

```
subregion='delaware'
download_dir='osmfiles'
osm_file_format='.pbf'
orl.download_osm_data_from_bbbike(subregion_names=subregion,osm_file_format=osm_file_format,download_dir=download_dir,
                                                confirmation_required=False,verbose=True)
```

Step 3: Converting downloaded OSM data to GMNS-rail network files

parameter list:

osm_filename	str, directory where the .osm data file is located/saved.
Bbox	tuple, defaults to the boundary of .osm data file.
strict_mode	bool, whether delete objects beyond the boundary, defaults to False.
POIs	bool, whether parse POIs of rail (such as platform, depot, etc), defaults to False.

(1) convert OSM map data in .osm format to GMNS-rail network files

```
net=orl.get_network_from_OSMFile('waterloo_london.osm',strict_mode=True,POIs=True)
```

(2) convert OSM map data in .osm.pbf format to GMNS-rail network files

```
net=orl.get_network_from_PBFFile('delaware-latest.osm.pbf',strict_mode=True,POIs=True)
```

Step 4: Visualizing rail network data set

parameter list:

savefig	dict, specify a filename and dpi value in the dict field to save the figure when showing network, defaults to None.

```
orl.showNetwork(net)
# orl.showNetwork(net,savefig={'filename':'waterloo.png','dpi':300})
```

Step 5: Outputting Networks to CSV

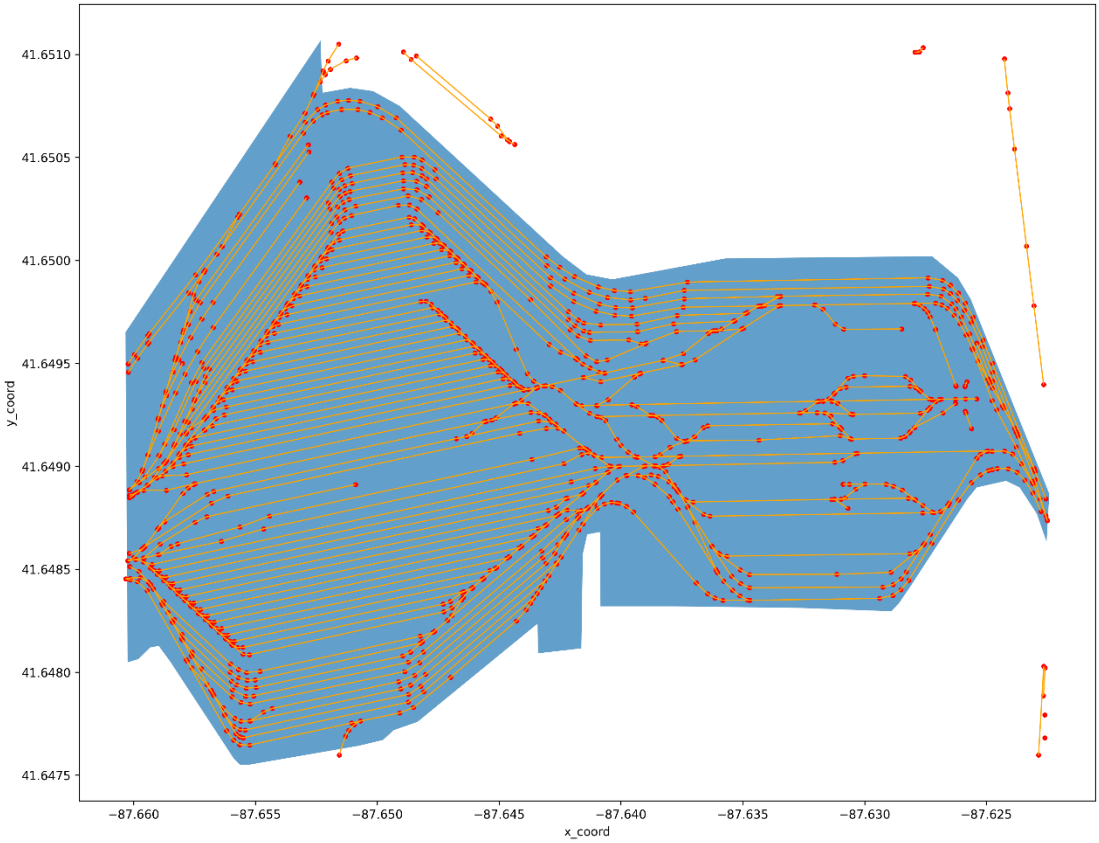
parameter list:

output_folder	str, directory for saving the network files, defaults to ‘./csvfile’.
enconding	str, the encoding of output files, defaults to None.

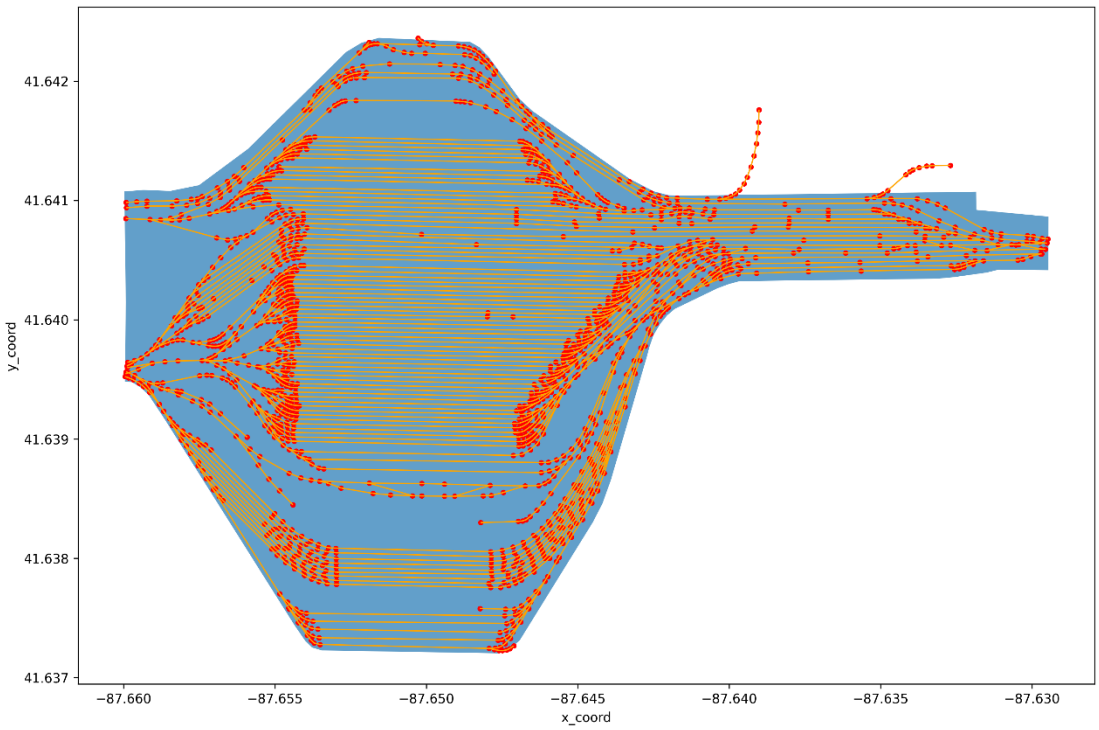
```
orl.saveNetwork(net)
```

4. Sample Networks

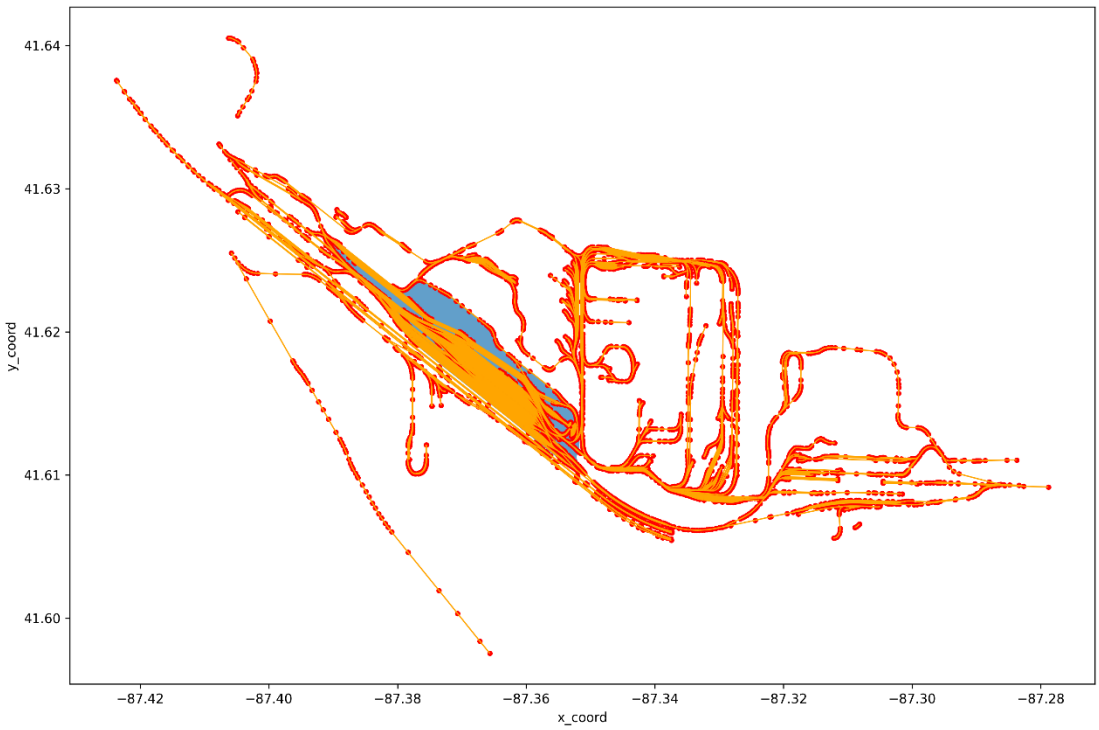
4.1 Example of CSX Barr Yard, USA



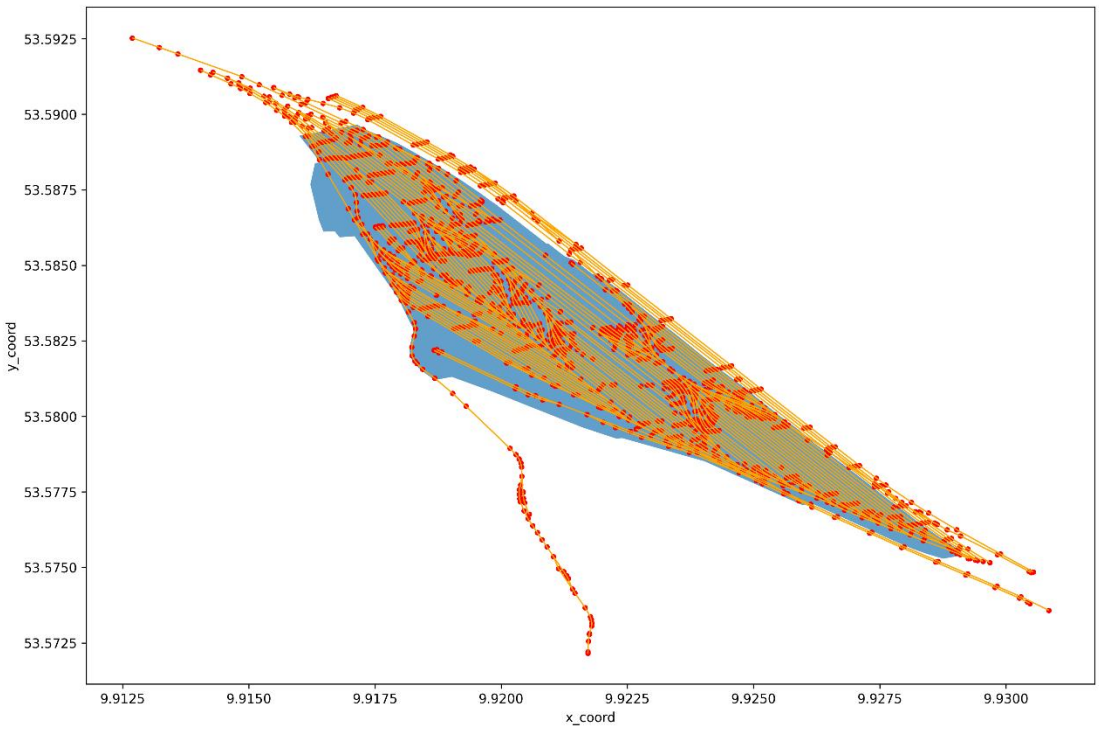
4.2 Example of IHB Blue Island Yard, USA



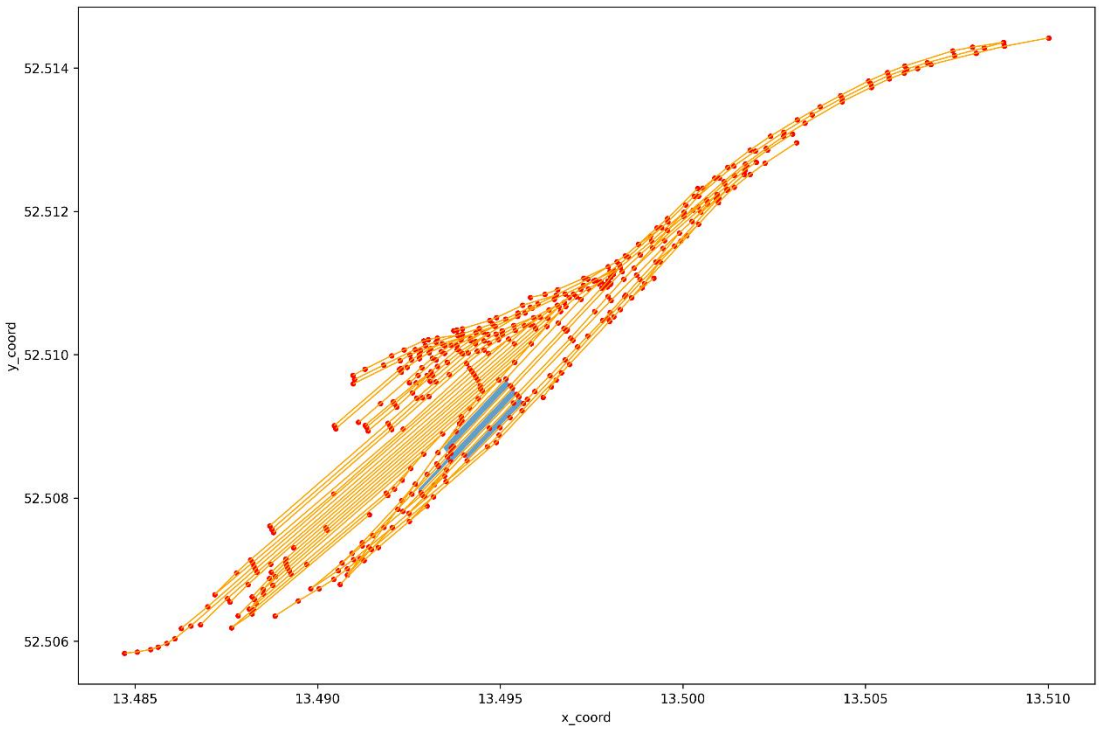
4.3 Example of U.S. Steel Gary Works, USA



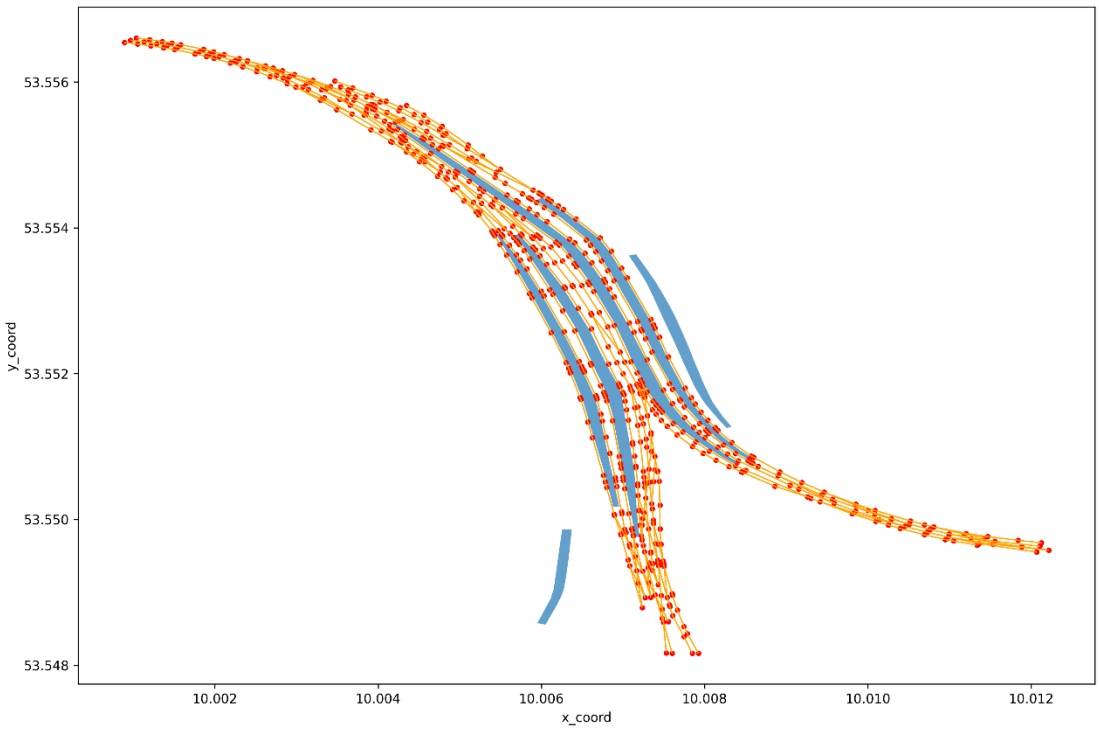
4.4 Example of Bahnbetriebswerk Hamburg Langenfelde, Germany



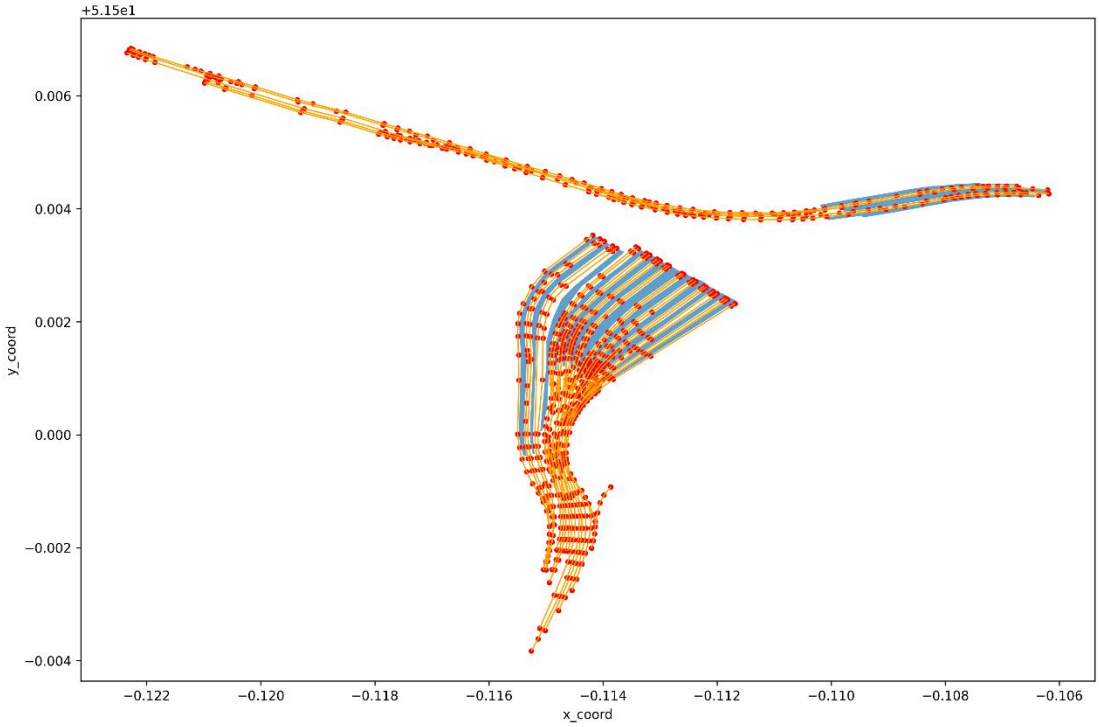
4.5 Example of Berlin-Lichtenberg, Germany



4.6 Example of Hauptbahnhof Nord, Germany

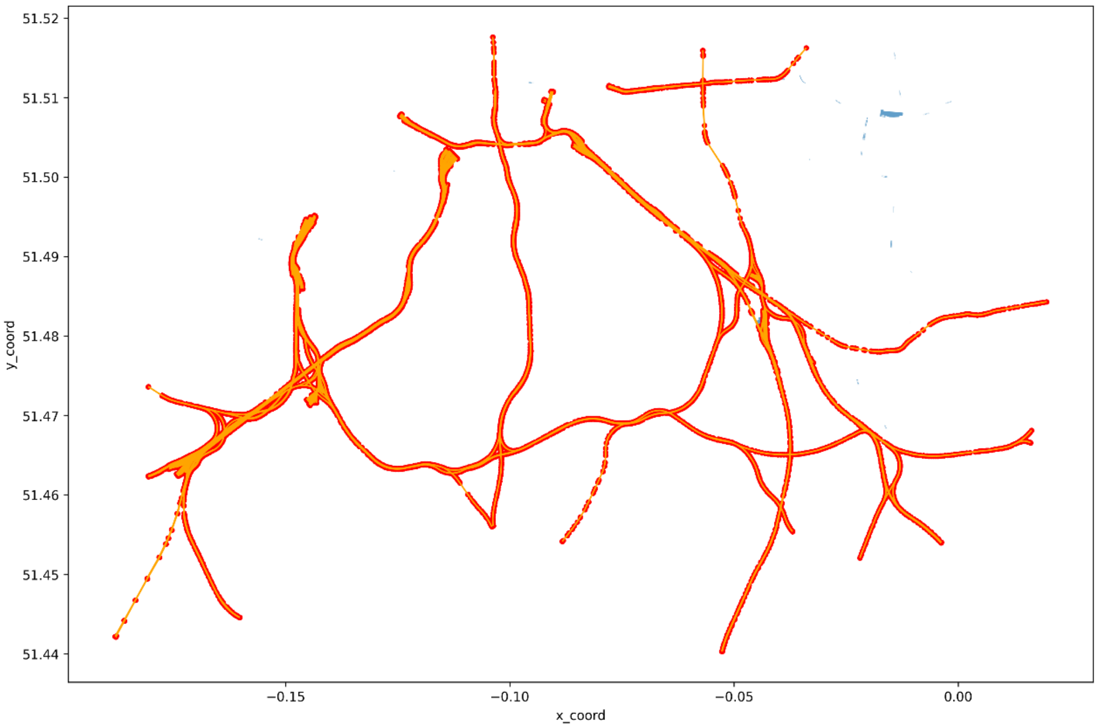


4.7 Example of London Waterloo, UK



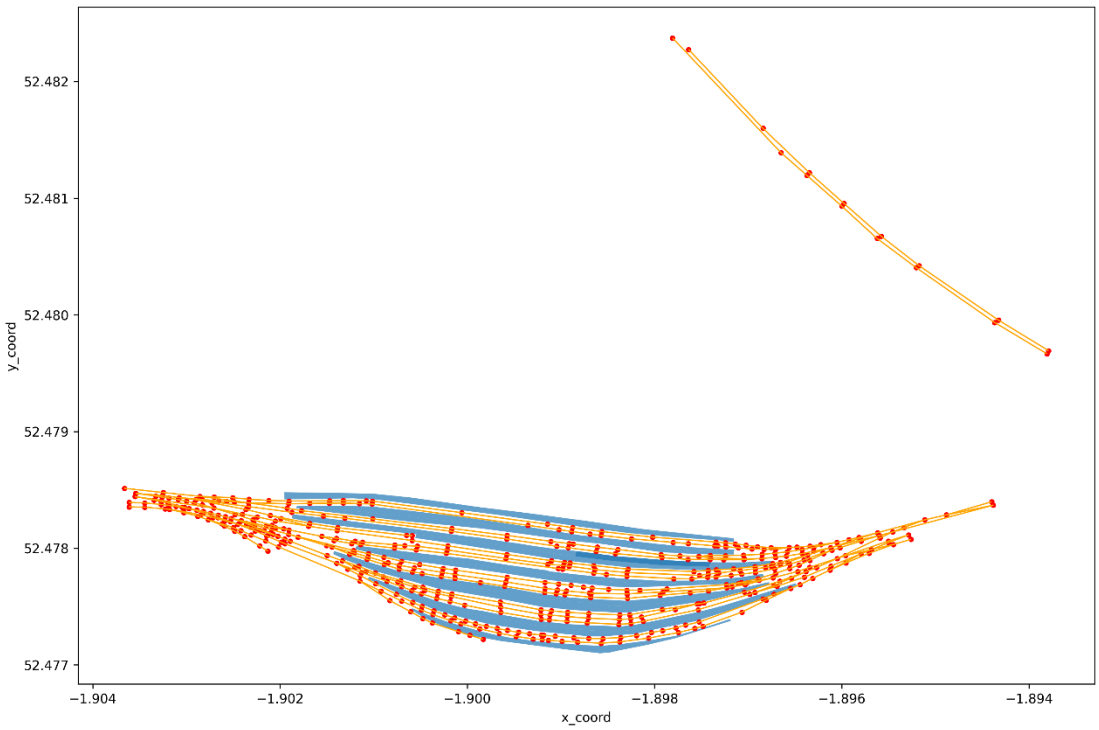
(London waterloo station is a terminal)

4.8 Example of Near London Waterloo, UK

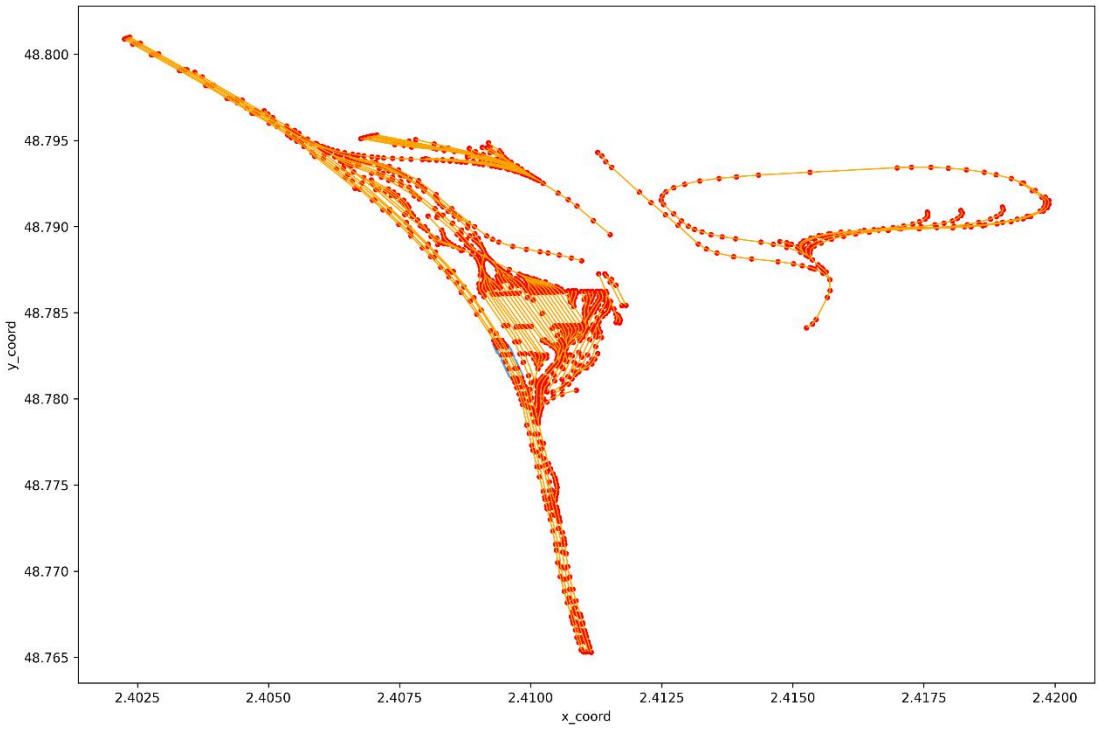


shown in QGIS

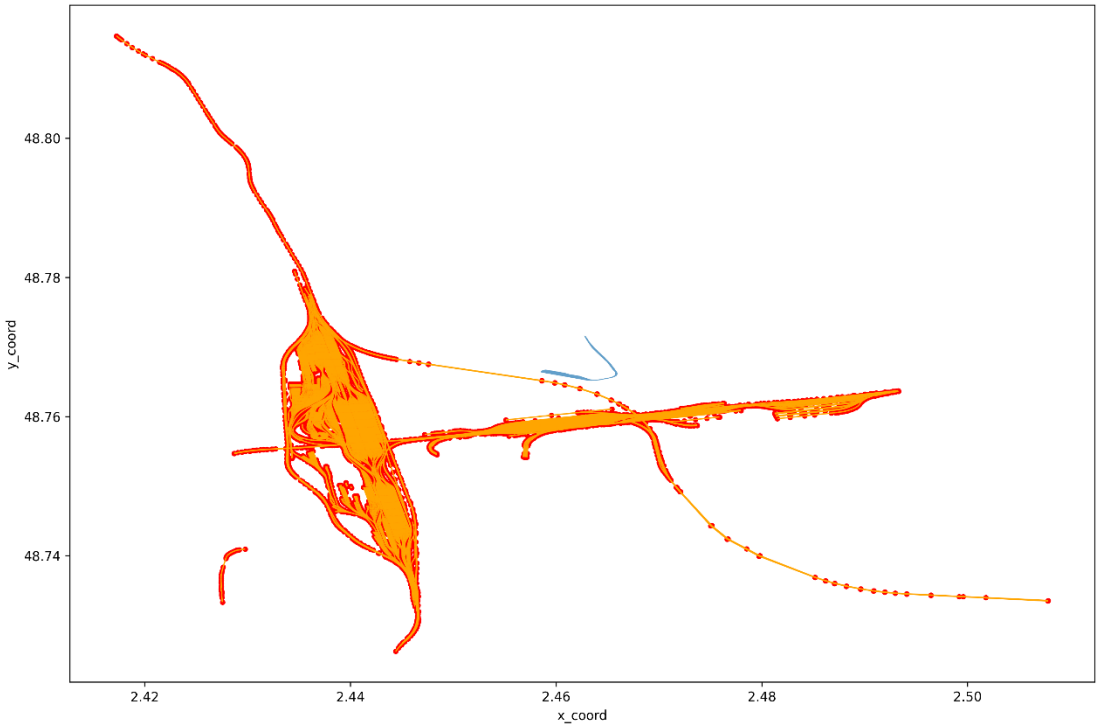
4.9 Example of Birmingham New Street, UK



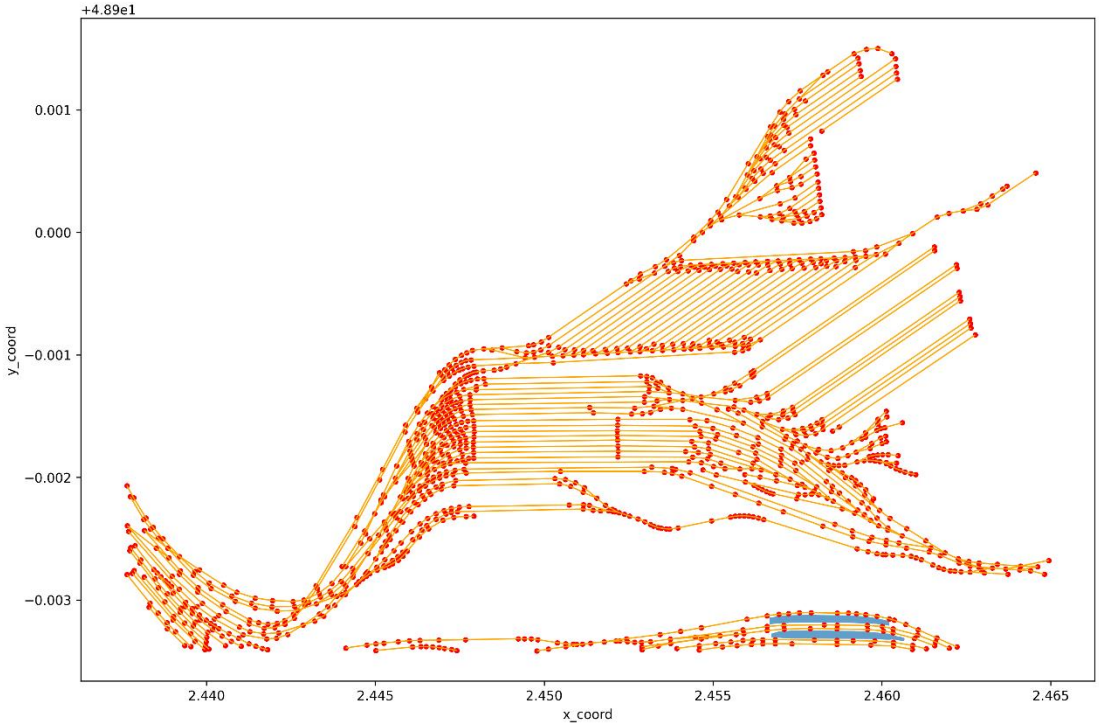
4.10 Example of Near Ancienne centrale thermique de Vitrysur-Seine, France



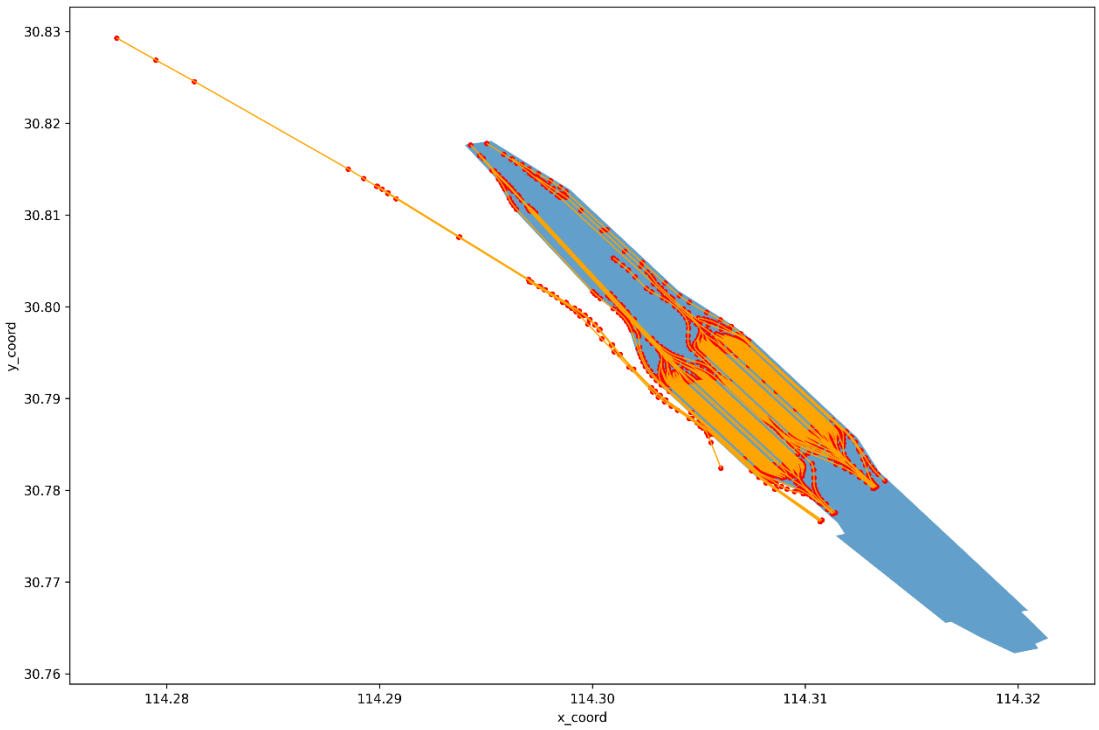
4.11 Example of Gare de Triage de Villeneuve Saint Georges, France



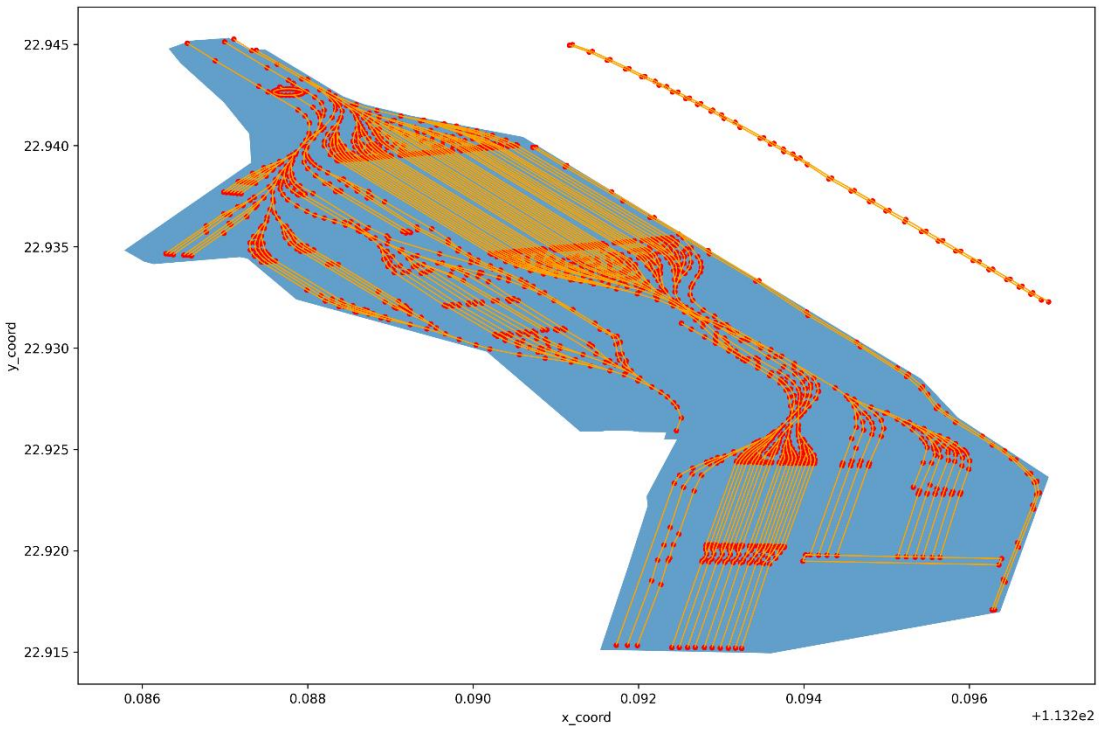
4.12 Example of Gare de triage de Noisy le Sec, France



4.13 Example of Wuhan North Railway Station, China



4.14 Example of Guangzhou EMU Maintenance Facility, China



4.15 Example of Shanghai-Hongqiao Railway Station, China