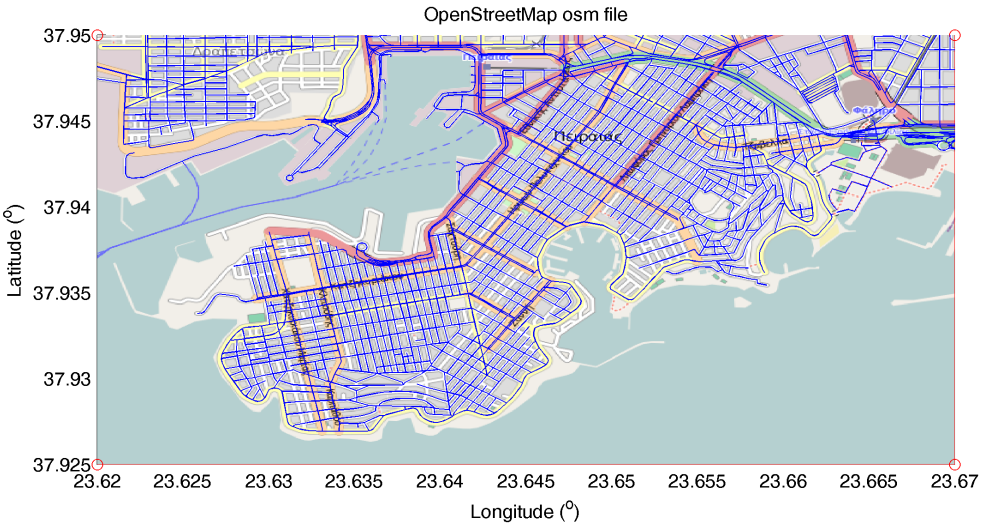
User Manual

OpenStreetMap Functions for MATLAB v.0.2



Ioannis F. Filippidis

Control Systems Lab
Department of Mechanical Engineering
National Technical University of Athens
Athens,
Greece
jfilippidis@gmail.com
Date: May 3, 2012

Cover: *Map data plot for the port of Piraeus, Greece.* Map data © OpenStreetMap contributors, CC-BY-SA.

This work has been typeset by the author using XeLaTeX & friends.

Chapter 1

Usage

1.1 Introduction

This is a small set of functions to import data from an OpenStreetMap XML Data File into MATLAB. The XML format is first loaded into MATLAB and then a parsing script extracts part of the information contained in the file. This is saved in a MATLAB structure. Then, it can be used to plot the transportation network and to extract its connectivity matrix, as well as to identify its unique nodes. An example of using these functions can be found in the file `usage_example.m`. The software architecture is shown in Fig. 1.1.

1.1.1 Dependencies

This software requires the following functions, which can be downloaded from the Mathworks central File Exchange and are copyright of their authors, under the BSD license.

1. `xml2struct` file id 28518 ©2010 by Wouter Falkena.
2. `lat_lon_proportions` file id 32462 ©2011 by Jonathan Sullivan.
3. `dijkstra` file id 24134 ©2008-2009 Stanford University, by David F. Gleich.

For the user's convenience, these functions have also been included in the distribution of this software, together with their respective licenses.

Remark: Please put these functions in your MATLAB path. You can do this by adding the `xml2struct` directory (after extracting from the archive) to your MATLAB path (`pathtool` command in MATLAB).

1.2 Parse the OpenStreetMap file

To parse an OpenStreetMap XML file (OSM XML) use the function `parse_openstreetmap`, by issuing the command `parsed_osm = parse_openstreetmap(openstreetmap_filename)`. This will parse an OSM XML file downloaded from the OpenStreetMap website. The OSM filename is provided as the string argument `openstreetmap_filename`. This function returns a MATLAB structure `parsed_osm` containing a subset of the OSM data sufficient to extract the transportation network connectivity. The structure's fields are shown schematically in Fig. 1.3.

To download an OpenStreetMap XML Data file, navigate to
http://www.openstreetmap.org/

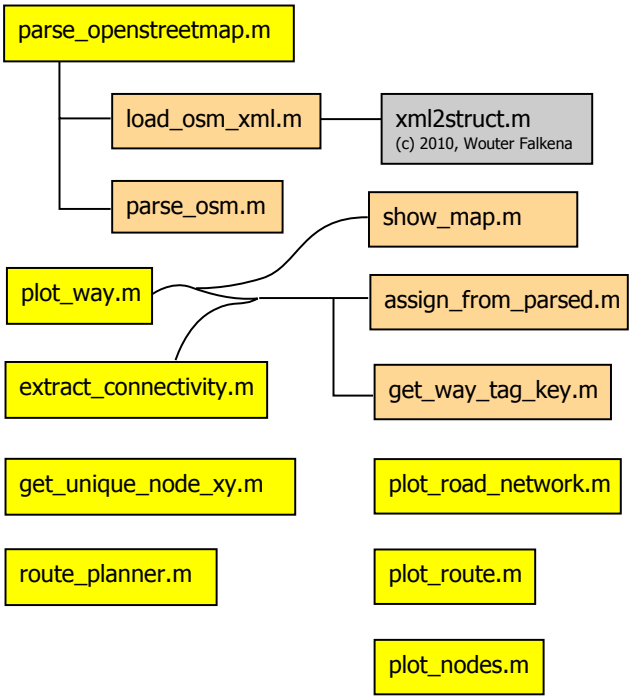


Figure 1.1: Software architecture.

(a) OpenStreetMap logo (®by Steve Coast from the OpenStreetMap wiki).

(b) Aerial photography for mapping the Praga district of Warsaw (CC BY-SA by Balrog, via OpenStreetMap wiki).

(c) GPS Mapping of Strasbourg on bicycle. (CC BY-2.0 by francois, via Wikimedia Commons).

Figure 1.2: OpenStreetMap

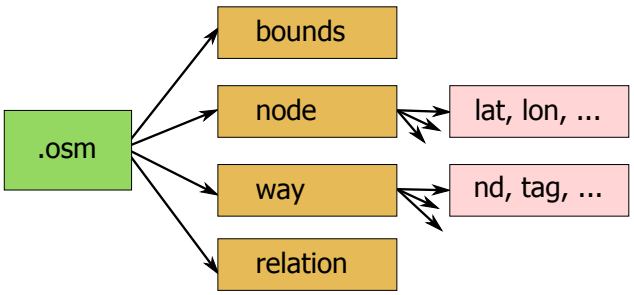
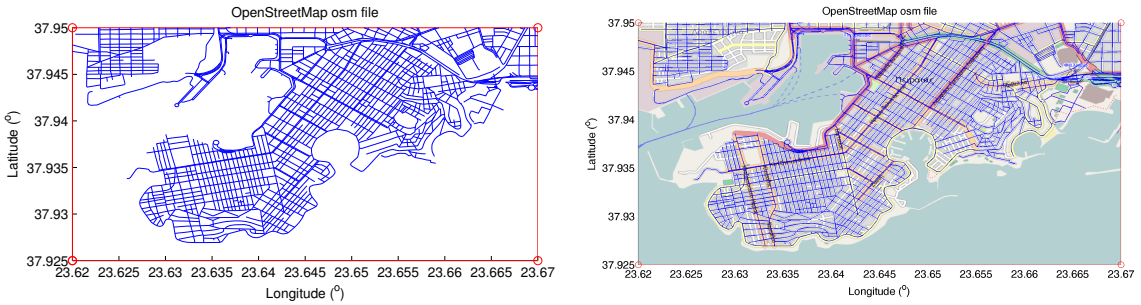


Figure 1.3: OpenStreetMap MATLAB structure fields.



(a) Without an underlying image.

(b) With an underlying image.

Figure 1.4: Examples of plotting the loaded OSM data using `plot_way`. Map data © OpenStreetMap contributors, CC-BY-SA.

and after zooming in the area of interest, export its data using the `Export` option to save it as an OpenStreetMap XML Data file, selecting this from the `Format to Export` options.

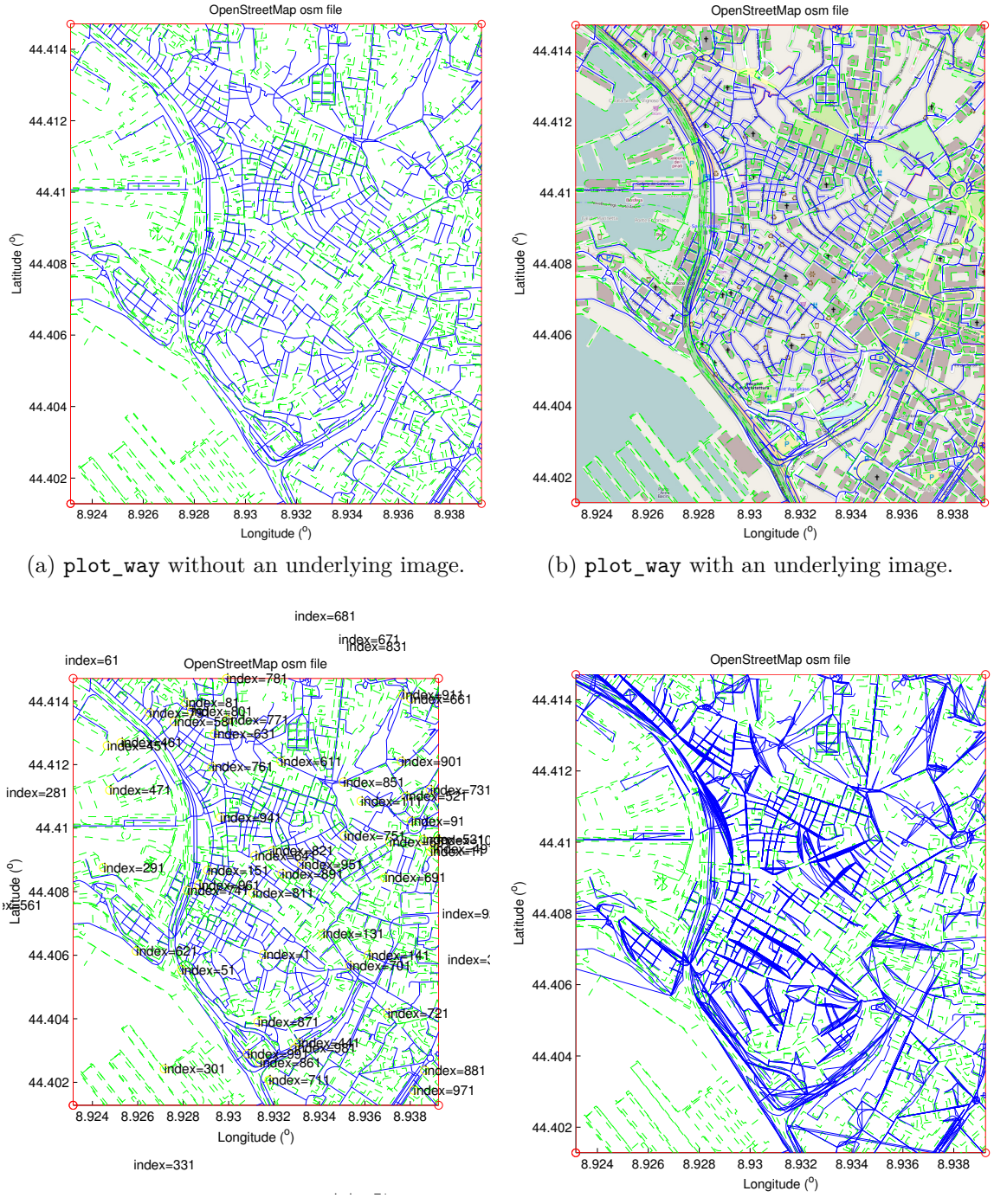
1.3 Plot the map

To plot the parsed `OSM` file, using the MATLAB structure obtained in the previous section, use the function `plot_way(ax, parsed_osm, map_img_filename)`. Argument `ax` is a handle to an axes object. `parsed_osm` is the MATLAB structure object returned by the parsing function `parse_openstreetmap`. Optionally, the last argument, `map_img_filename`, is a (raster) picture to load and plot underneath the transportation network of the map. Examples of the result with and without the underlying image are shown in Fig. 1.4a and Fig. 1.4b, respectively.

The other plotting functions are `plot_nodes`, `plot_road_network` and `plot_route`. Function `plot_nodes` can be used to plot selected (or all) nodes and optionally their IDs as text labels, as shown in Fig. 1.5c. Function `plot_road_network` plots the nodes and edges connecting them, as determined by the connectivity matrix created using `extract_connectivity` (see section 1.4), as shown in Fig. 1.5d. Function `plot_route` plots the series of nodes traversed along a route found by the `route_planner` function, as shown in Fig. 1.7a and Fig. 1.7b.

1.4 Extract road connectivity and Routing

To extract the road connectivity from the OSM data loaded into the MATLAB structure, use the function `extract_connectivity`. This function extracts the connectivity of the road



(a) `plot_way` without an underlying image.

(b) `plot_way` with an underlying image.

(c) `plot_nodes`.

(d) `plot_road_network`.

Figure 1.5: Examples of plotting the loaded OSM data. Map data © OpenStreetMap contributors, CC-BY-SA.

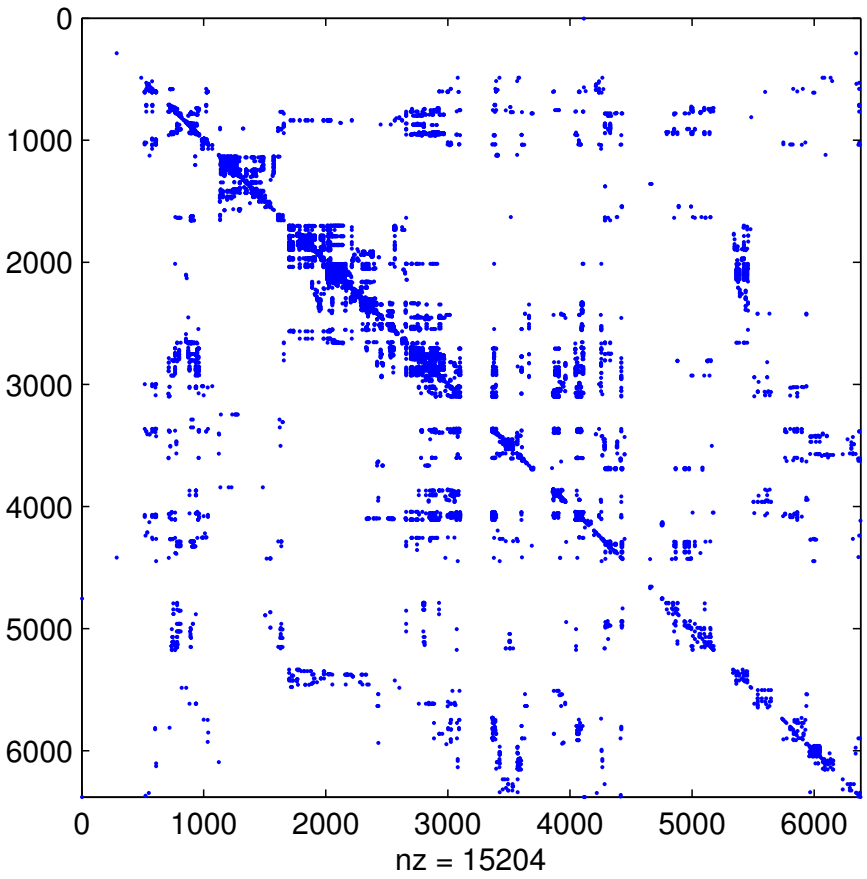
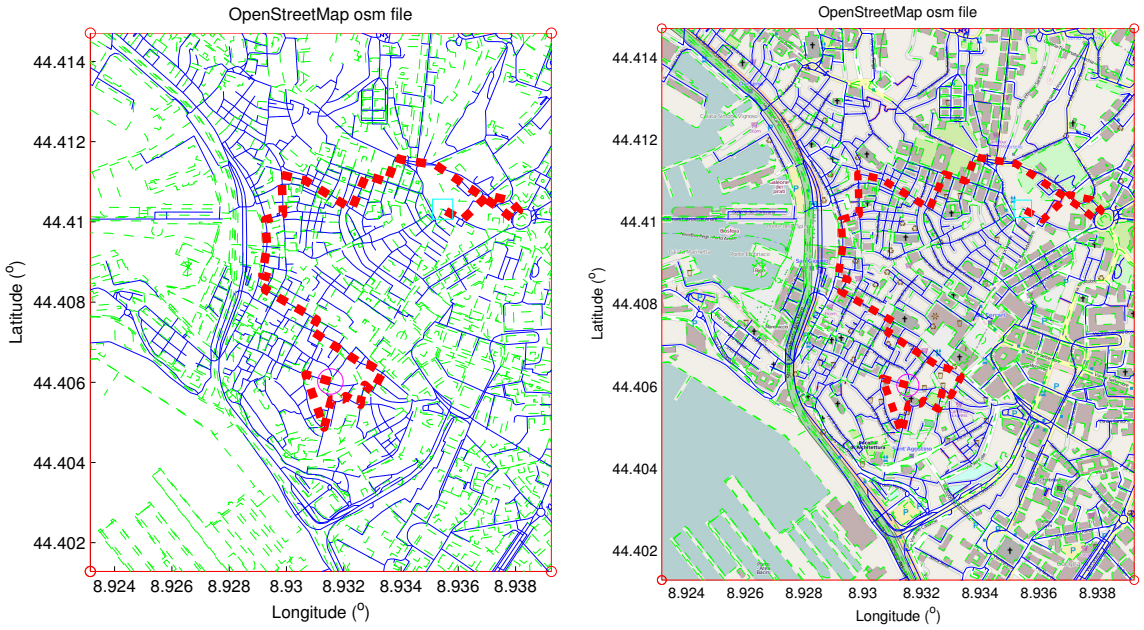


Figure 1.6: Connectivity matrix of transportation network of Piraeus (nodes are not unique), corresponding to Fig. 1.4a. Observe the sparsity of the adjacency matrix.

network of the OpenStreetMap file. This yields a set of nodes where the roads intersect. Some intersections may appear multiple times, because different roads may meet at the same intersection and because multiple directions are considered different roads. For this reason, in addition to the connectivity matrix, the unique nodes are also identified.

To call it, issue the command:
`[connectivity_matrix, intersection_nodes] = extract_connectivity(parsed_osm)`. The input `parsed_osm` is the MATLAB structure returned by function `parse_openstreetmap`. This function returns the network's adjacency matrix in `connectivity_matrix` and the unique nodes corresponding to the intersections of the network in `intersection_nodes`.

Routing can be performed based on the extracted adjacency matrix. The function performing routing is `route_planner`. It can be called by issuing
`[route, dist] = route_planner(dg, start, target)`, where `dg` is the adjacency matrix of the directed graph representing the transportation network (that is, the `connectivity_matrix` above), `start` is the source node and `target` the destination node. This function returns `route`, which is a matrix of traversed node indices, and `dist`, which is the count of nodes traversed (and not the spatial distance over the map). Plotted examples are shown in Fig. 1.7a and Fig. 1.7b.



(a) Without an underlying image.

(b) With an underlying image.

Figure 1.7: Examples of route plotting using `plot_route` and `plot_way`. Map data © OpenStreetMap contributors, CC-BY-SA.