UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Tècnica Superior d'Enginyeria Industrial de Barcelona

ETSEIB

TECHNISCHE UNIVERSITÄT WIEN

Analysis of typical traffic networks

MASTER THESIS

Master's degree in Industrial Engineering

September 2017

Author: Maria del Mar Llompart Jaume

Advisor: Stefan Jakubek

Supervisor: Elvira Thonhofer

Abstract

The origin of the project comes from the need of classifying any city traffic network, in order to regulate it in smart ways later. Knowing this previous city classification is a fast way to have a first impression of the city, with some parameters that clearly describe its structure and its connections with other cities. These steps can save time to the researchers or, in terms of business, they can save money.

The main objective is to do an analysis of typical traffic networks. What this means is to classify any city by its traffic network topology. The development of the project will be done by using the MATLAB programming language and observing the results.

The first step, in order to carry out this project and achieve the objectives, is to do a deep literature research of the main topics. This literature investigation has been divided in five parts, considered as the main fields of study: map data (using reliable information is crucial to do a good study of different cities and compare them), graph theory (the basis of mathematics in this project), social network analysis (a strategy for looking into social networks and structures, by using mathematical formulations extracted from graph theory), traffic network topologies (many studies propose measures to characterize networks) and intersections (the classification of streets unions).

Then, an experimental process is proposed, by using MATLAB and parameters chosen from other studies observations and the project limits (time of realization and difficulty to program). The main parameters here used, plotted in graphs, are: distance between two intersections which are connected by a street, the angle between the streets that arrive to an intersection, the degree of a node (number of streets that arrive and leave an intersection) and the significant orientation of the streets of a node (respect to 0˚). Then, other parameters are also displayed, as number of nodes, number of nodes, alpha index, beta index, etc.

The process holds all the MATLAB functions and is divided in four parts. At first, the part which converts the OpenStreetMap (source of map data) file to MATLAB useful language. Then, to fix the matrices and to fix the map (two different steps) obtained in the first step is necessary. Finally, the functions to obtain results and characteristics are programmed.

Once explained the functions, the results are summarized in graphs and tables, and then are analysed. This analysis gives a global idea of the maps, by comparing 7 maps at the same time. The grids are totally recognised by their unmistakable characteristics, but the other types of maps are classified by the values obtained. In conclusion, this project studies a set of parameters but it is not a closed project, it can be extended and improved by whoever wants to.

Acknowledgements

In the first instance, I want to thank the TU Wien (*Technische Universität Wien,* my hosting university) and ETSEIB (*Escola Tècnica Superior d'Enginyeria Industrial de Barcelona,* my home university) for giving me the opportunity of living this experience in Vienna with the Erasmus+ program.

Then, I want to thank Stefan Jakubek, the advisor of this thesis, and the ANDATA organization, the venture for giving me the possibility of working in its projects and believed in my engineering skills. They let me to contribute in an important investigation about traffic networks and lent me a laptop from the university in order to carry out the research and work. In this organization, I have always found good people who have been willing to give me a hand in my challenges and the project complications. I specially appreciate the help of Toni Palau, Ana Herrera and, above all, Elvira Thonhofer, my supervisor, for all her support and help whenever I needed it. I really hope this thesis helps you in your PhD research.

Finally, I would like to mention all the friends from all over the world I made during this experience in Vienna. Without them and without their support, it would not be the same.

Contents

Abbreviations and symbols list

a	Scaling vector
ax	Axes object handle
a_{ij} or $a_{v,t}$	Elements of the matrix A
A	Incidence matrix or size of the area (km^2)
A_i	Average Shimbel index
b	Scope of the weight of the centrality
BC_i	Betweenness of the node i
BP_i	Bonacich power of node i
$cut(A, B)$	Cutpoint of component A and B
C	Average circuity in the metropolitan area
$Cycle_i$	Number of times random walk cycled back to node i
$C_{Closeness}(N_i)$	Closeness
$\hat{C}$	Cyclicity
C_i	Clustering coefficient of node i
dist	Link length accumulation
d	Distance in the haversine formula
$\mathrm{d(v)}$	Degree of the vertex v
$d^+(v)$	Out-degree of the vertex v
$d^-(v)$	In-degree of v
$\bar{d}$	Average index
d_{ij}	Elements of the matrix D or geodesic distance between i and j
D	Adjacency matrix or diameter
D^T	Transposed matrix of D
D_e	Sum of the Euclidean distance (km) between all OD pairs in the subsample
D_n	Sum of the network distance (km) between all OD pairs in the subsample
e	Number of links in the graph
E	Set of edges
ETSEIB	Escola Tècnica Superior d'Enginyeria Industrial de Barcelona
F_i	Fragmentation of the node i
gca	Current axes or chart
G	Graph or number of sub-graphs in the graph
GIS	Geographic Information Systems
G_u	Underlying undirected graph of the digraph G
H	Strongly connected component of the digraph G
$H(G, P)$	Entropy

id	Identification
I	Identity matrix (1s down the diagonal)
k_i	Number of neighbours for node i
L	Total roadway kilometers in the area
L_f	Number of freeway kilometers in the area
L_t	Length (in km) of street segments belonging to a branch network in the area
m	Number of edges
m, m_1, m_2	Inclinations
m_i	Mass of the particle i
M	Total mass of the system
M_i	Number of pair of neighbours of node i that are connected
$M(v)$	Set of neighbors of v
n or N	Number of vertices
nd	Sets of nodes that form each way in MATLAB
n_i	Number of arcs incident on node i
node 1	Beginning vertex of the link
node 2	Ending vertex of the link
OD or O-D	Origen – Destination
OSM	OpenStreetMap
O_t	Number of destinations that can be reached in a given time threshold
p and p'	Petal in the Vehicle Routing Problem
P	Probability distribution on the node set of $\mathrm{V(G)}$
r	Radius of the Earth
r_i	Coordinates of the particle i
r_{ij}	Component of the matrix R
R	Adjacency matrix
SNA	Social Network Analysis
St. Dev.	Standard Deviation
S_n	Average network speed in $\mathrm{km/h}$
S_v	2-step centrality value
t	Time threshold (in minutes)
tag	Sets of vectors with strings about the key and value of the way
TSP	Travelling salesman problem
TU	Technische Universität
$\mathrm{u, v, w}$	Vertices of a graph
v	Number of nodes in the graph
V	Set of vertices
w_{ij}	Weight of the connections between i and j

WMS	Web Map Service
xy	Vector with coordinates longitude and latitude of each node
x, y	Coordinates of a point that fits in the line
x_1, y_1	Coordinates of one point of the line
x_v	Eigenvector centrality
α	Alpha index
β	Beta index
γ	Gamma index
$\delta(\mathrm{G})$	Minimum degree of the vertices in a graph G
$\Delta\, distance(\%)$	Percentage variation of distance
$\Delta(\mathrm{G})$	Maximum degree of vertices in G
λ	Constant
λ_1, λ_2	Longitude of point 1 and longitude of point 2, in radians
μ	Cyclomatic number
ρ_e	Completeness
ρ_{lm}	Street density
ρ_{pm}	Urban area population density $(\mathrm{person} \cdot \mathrm{km}^{-2})$
σ_{jk}	Total number of shortest paths from node j to k
$\sigma_{jk(i)}$	Number of shortest paths from node j to k that pass through node i
φ_1, φ_2	Latitude of point 1 and latitude of point 2, in radians
ϕ_{tree}	Treeness
1	Matrix of all ones
$\%F$	Percentage of freeways
$\lvert R \rvert$	Number of random walks

List of Figures

List of Tables

1. Preface

This project is a Master Thesis of an exchange student (from ETSEIB – *Escola Tècnica Superior d'Enginyeria Industrial de Barcelona*) at the TU Wien (*Technische Universität Wien*), carried out at ANDATA.

ANDATA is an independent technology venture specialized in the development and application of methods from the fields of *Data Mining* and *Artificial Intelligence* in combination with an extensive use of *numerical simulation* in technical development.

1.1. Origin of the project

The origin of the project comes from the necessity of classifying any city traffic network, in order to regulate it in smart ways later. From here, it is possible to compare two cities by their topology or compare two intersections by their characteristics.

Moreover, knowing this previous city classification is a fast way to have a first impression of the city, with some parameters that clearly describe its structure and its connections with other cities. These steps can save time to the researchers or, in terms of business, they can save money.

1.2. Motivation

ANDATA researchers are looking for having a traffic classification in order to facilitate others projects and studies. The analysis of typical traffic networks will be carried out in parallel with similar projects and it will be really helpful in future projects.

The field of study is selected, in agreement with the advisor, by the author for the interest in *Traffic networks*, due to previous works in *Transportations* (subject done during the Master in Industrial Engineering).

The motivation comes from the want of increase the knowledge in *Transportations* and having the opportunity of working with traffic networks specialists in this venture.

In addition, this project is a chance to learn about traffic of all types of cities, with real database obtained from safe and official sources, also a chance to work with specialists in this field and, by last, to improve in computing language this project requires to work with.

1.3. Previous requirements

The field of study *Traffic networks* is a field which is growing by the development of new technologies and transportation increase, so this Master Thesis requires having some previous knowledge on the topic. Other fields that involve this project are widespread and the information about them is abundant and, sometimes difficult to understand if the reader is not used to the topics. For these other topics this project has relation with, it is desirable that the author is willing to learn about them, such as graph theory, social network analysis, etc.

Some engineering mathematics can become difficult to resolve or to program. These boundaries could hinder or slow down the development of the programming. Having good engineering and mathematics skills is the basis to be able to set the problem and try to solve it. The problem is expected to be solved in programming language, so having familiarity with programming is necessary, such as MATLAB or Python/Numpy.

Finally, in order to achieve the project goals, it is necessary, from the author, to have commitment and joy with new technologies.

2. Introduction

Humanity is surrounded by different ways of moving between two points (origin – destination). We normally have many types of transportation to choose (by bicycle, car, bus, train, plane and ship, among others), depending on lots of parameters, such as available time, system availability, traffic, distance, money… Sometimes, for one transportation system, we also have the chance to choose among different paths in the same network.

Transportation systems are the basis of human connectivity. These systems can be roadways, railways, sea links, airspace or intermodal combinations, which every of them define a network topology. All these systems are being increasingly developed and studied with the technological advances, such as new technologies in transportation systems (e.g. autonomous vehicles), developed technologies in networks (e.g. sensors and cameras to control traffic) and powerful programming for complex mathematical problems.

The study of traffic networks is becoming increasingly necessary in order to regulate traffic in an intelligent way and ensure quality of service. Then, appropriate traffic models must be developed beforehand. The quality of the models is ensured by means of simulation and subsequent validation using real data.

Figure 2.1. Traffic network intersection [Source: https://www.adgeco.com/uae-launched-advanced-traffic-systems-gadgets-avert-road-collisions/]

In order for a traffic model to be universally applicable, it has to be able to deal with very different types of road networks. For example, the road network of a typical American big city with a regular square grid (an intersection of this network is visible in the Figure 2.1) is in stark contrast to a European old town. It is therefore necessary to detect and quantify these differences mathematically.

2.1. Project objectives

The main objective, as the project tittle indicates, is to do an analysis of typical traffic networks. What this tittle means is to classify any city by its traffic network topology. The development of the project will be done by using programming languages and observing the results.

In order to achieve the main goal, it is necessary to carry out previous tasks, like to do a wide literature research about some fields that involve the project: graph theory, social networks analysis, Map Data, etc.

After the deep literature review, a study of suitable features to describe street networks has to be done. Many studies investigate the way to classify a network, with lots of traffic parameters. These traffic parameters change depending on the field of study and the author.

Furthermore, once studied the features, MATLAB functions will be developed to compute them, as well as a development of generic standards configurations. MATLAB is the main working tool to carry out the aim of this thesis.

After the proposed study and once obtained the results, by comparing networks, one must be able to classify the network.

3. Literature research

In order to carry out this project and achieve the objectives, a deep literature research of the main topics has to be done. This research has been divided in five parts, considered as the five main fields of study.

First of all, to do a research of Map Data is considered advisable, as data analysis is the basis of this project. For that reason, using reliable information is crucial to do a good study of different cities and compare them. A review of the history of Map Data, a study of the problematics in this field (lack of information) and the web map service are going to be studied.

Related to graph theory, there are lots of studies, papers, investigations and theories. Graph theory is widespread, so in this section, a selection of useful information is trying to be done and summarized.

Finally, to do a review of social networks analysis and a traffic network analysis is necessary to discuss different points of view depending on the author and to extract the parameters that are considered useful for this project. A chapter about intersections is also added to take a look at a classification.

3.1. Map Data

3.1.1. Introduction

"On a planet of finite resources faced with mounting population pressures, geographic information systems already have become indispensable for resource management, policy assessments and strategic decisions."

George E. Brown in 1992

During the past decade, a revolution has drastically altered the world of cartography [20]. The changes started trying to automate the standard map products with various computer technologies, a new industry known as geographic information systems (GIS).

Now the effort is on varied uses of geographic information in digital forms. Thus, maps are frequently viewed as one of a wide variety of potential products.

GIS treats data as different layers. Every type of data can be customized to meet specific criteria and then the various layers can combine to form a single map. One of the key features of GIS is that any type of information with a geographic component can be mapped. In this way, thematic maps can be constructed from layers of data that represent traditional cartographic information and from data sets that the user supplies from other sources.

A Web map service (WMS) is a standard protocol that describes how to serve any georeferenced map images over the Internet, which is usually generated by a map server that uses data from s geographic information system database [22].

3.1.2. Lack of data focused on traffic

The main trouble at the time of searching reliable data about traffic is the lack of this. Many resources are private for governments or researchers, so for this thesis public data from free sources is going to be used.

There are many web map services where to find satellite images, street maps, panoramic street views, traffic in real-time and route planning for travelling by foot, car, bicycle or public transportation. Some of the most important web map services and their more relevant features for the Master Thesis are summarized in the following table [21]:

Feature	Google Maps	Bing Maps	MapQuest	OpenStreetMap	Here	Apple Maps
Full Extra Functionality	Australia, Canada, China, France, Germany, Israel, Italy, Netherlands, Spain, UK, United States.	Andorra, Australia, Austria, Bahrain, Belgium, Canada, Croatia, Czech Republic, Denmark, Finland, France, Germany, Gibraltar, Guernsey, Hong Kong SAR, Hungary, Iceland, Ireland, Isle of Man, Italy, Japan, Jersey, Jordan, Kuwait, Liechtenstein, Luxembourg, Malaysia, Monaco, Netherlands, New Zealand, Norway, Oman, Portugal, Puerto Rico, Qatar, Romania, San Marino, Saudi Arabia, Singapore, South Africa, Spain, Sweden, Switzerland, Taiwan, United Arab Emirates, United Kingdom, United States, Vatican City	United States	All	More than 180 navigable countries	Austria, Belgium, Canada, Denmark, France, Germany, Ireland, Italy, Netherlands, Norway, Spain, Sweden, Switzerland, UK, USA
Degrees of motion	Vertical, Horizontal, Depth, Rotation(beta), 360 Panoramic (Street View), 3D Mode (Google Earth JavaScript)	Vertical, Horizontal, Depth, 360 Panoramic (Streetside), 3D Mode(Tilt, Pan, Rotate)	Vertical, Horizontal, Depth	Vertical, Horizontal, Depth	Vertical, Horizontal, Depth (zoom), Tilt (3D), Rotate 360 degrees	Vertical, Horizontal, Depth, Rotate 360 degrees, 3D
Map Zoom	22 (more levels available through parameter)	19-22 (Depending on which map control is used)	17	19	18	Unknown (vector-based)

Feature	Google Maps	Bing Maps	MapQuest	OpenStreetMap	Here	Apple Maps
Map Types	6: Map with traffic data (separate transit and bicycle view), Satellite with Traffic Data (3D LiDar for certain places), Hybrid	9: Road, Satellite, Hybrid, Bird's Eye, Traffic, 3D, London Street Map, Ordnance Survey Map, Venue Maps	3: Road, Satellite, Traffic	5: Standard Map, Transport Map, Cycle Map, Humanitarian	7: Map View, Satellite, Terrain, 3D, Traffic, Public Transportation, Heat Map, Map Creator, Explore Places, Community	3: Standard, Hybrid, Satellite. All include a traffic data layer
3D Mode	Yes (with plugin) Limited to certain areas	Yes (Windows 8/10)	No	Yes, third-party	Yes limited to certain areas	Yes limited to certain areas
Age of Map Imagery	Updated Daily	Updated Monthly		Updated Live		1–2 years
Map Data Providers	MAPIT, Tele Atlas, DigitalGlobe, MDA Federal, user contributions	NAVTEQ, Intermap, Pictometry International, NASA, Blom, Ordnance Survey, SK	TomTom, OpenStreetMap, and others	User contributions, open data and data donations	Navteq	TomTom, and others
Directions	Yes	Yes	Yes	Yes	Yes – by car, foot, public transport	Yes
Live Traffic Information	Yes	Yes (35 Countries)	Yes	Yes, partial in a third-party	Yes	Yes
Historic Traffic	Yes	No	No	No	Yes	No

Table 3.1. Comparison of web map services

3.1.3. Map Data used: *OpenStreetMap*

Among the available sources to choose to use, one that accomplishes the features in order to achieve the goals of the project is the web map service OpenStreetMap (© OpenStreetMap contributors). The data is available under the Open Database License and the cartography is licensed as CC BY-SA (see all these information in https://www.openstreetmap.org/copyright/en). For a printed copy of this project, the reader can consult the following websites: openstreetmap.org, opendatacommons.org and creativecommons.org.

OpenStreetMap (OSM) has map information of all parts of the world, it is updated in live (so it is always actualized) and has life traffic information, among others. It is important to underline that downloading maps from openstreetmap.org [24] is very easy and fast (*Export* option), but only works for small maps. For bigger maps, to look for alternatives is necessary. The same OSM website page gives some alternatives: *Overpass API*, *Planet OSM*, *OpenStreetMap Data Extract* and *metro extracts*, among others.

OpenStreetMap is a collaborative project to create a free editable map of the world. The creation (2004) and growth of OSM has been motivated by limitations on use or availability of map information across the world and the advent of inexpensive portable satellite navigation devices. It is considered a prominent example of volunteered geographic information [23].

OpenStreetMap uses a topological data structure, with four core elements:

- Nodes: points with a geographic position, stored as coordinates (pairs of latitude and longitude). Outside of their usage in ways, they are used to represent map features without size, such as points of interest.
- Ways: ordered lists of nodes, representing a polyline, or possibly a polygon if they form a closed loop. They are used for representing linear features such as streets and rivers.
- Relations: ordered lists of nodes, ways and relations (together called “members”). They are used for representing the relationship of existing nodes and ways.
- Tags: key-value pairs (both arbitrary strings). They are used to store metadata about the map objects, such as their type, their name and their physical properties).

OpenStreetMap is going to be used in the experimental part of this project, complemented with MATLAB functions that use OSM input data.

3.2. Graph theory

3.2.1. Introduction

Graph theory started with the Königsberg Bridge Problem. This problem was set out by the citizens, who used to spend Sunday free time walking though the city. They created a game for themselves: to cross the seven bridges of the city of Königsberg over the river Preger just once (bridge representation in the Figure 3.1) and, moreover, that the trip ended in the same place it began [3].

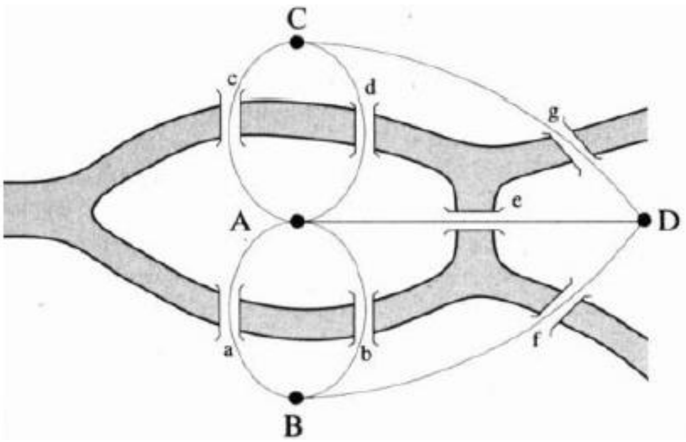


Figure 3.1. Representation of the Königsberg bridges [Source: in reference [5], 2nd slide]

None of the citizens was able to find a solution, but they could not prove it was impossible. The problem was proposed to the mathematician Leonard Euler, who said:

"This question is so banal, but seemed to me worthy of attention in that [neither] geometry, nor algebra, nor even the art of counting was sufficient to solve it."

Euler proposed a solution in 1736 and it represented the beginning of graph theory. The complex and interesting solution can be found in the article *Leonard Euler's Solution to the Konigsberg Bridge Problem* [4].

William Rowan Hamilton was also a pioneer in the field of graph theory [5]. In 1859, he developed a toy based on finding a path through all the cities in a graph exactly once, as seen in the *Figure 3.2*. The toy never was a big success, but he left the "Hamiltonian" concept, what means to cross by all vertices just once.

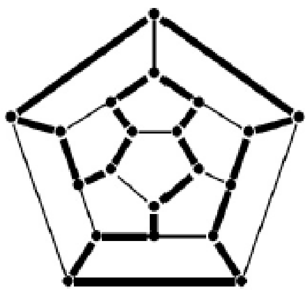


Figure 3.2. Hamilton toy representation [Source: in reference [5], 3rd slide]

Nowadays, graph theory is widespread and is used in lots of fields, such as sociology, biology and physics. Graph size can become quite big so that computing programs are required. Commonly used applications in our daily lives use graph theory and solve complex algorithms, for example to find the shortest path home in a GPS or the fastest public transportation.

Basic concepts with some examples are explained in the following parts. These parts follow the lecture notes and images of graph theory that Keijo Ruohonen did in 2013 [1], with additional information from lecture notes of *Transportations*, a subject of the Master of Industrial Engineering in Barcelona, references [10] and [11]. Moreover, these parts are complemented with some missing definitions found in the book [7].

3.2.2. Definition and fundamental concepts

Conceptually, a *graph* is formed by two kinds of elements: *vertices* (also called *nodes* or *points*) and *edges* (also called *lines*) connecting the vertices. This type of graph may be explained as undirected and simple, as the one we see in the Figure 3.3.

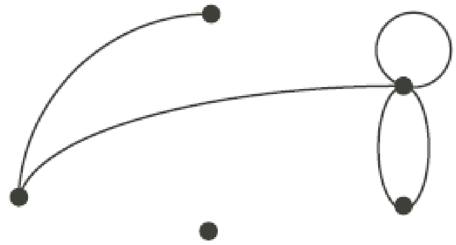


Figure 3.3. Example of an undirected graph

Formally, a graph is a pair of sets (*V, E*), where *V* is the *set* of *vertices* and *E* is the set of *edges*, where each edge is formed by pairs of vertices. *E* is a *multiset*, in other words, its elements can occur more than once so that every element has a *multiplicity*. Often, the vertices are labelled with letters (for example: $a, b, c, \dots\ or\ v1, v2, \dots$) or numbers $1, 2, \dots$ Similarly, the edges are labelled with letters (for example: $a, b, c, \dots\ or\ e1, e2, \dots$) or numbers $1, 2, \dots$ for simplicity.

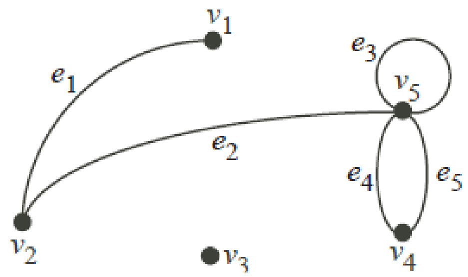


Figure 3.4. Same example with labels

For the example in the Figure 3.4, the notation is the following:

$$V = \{v_1, \dots, v_5\} \tag{3.1}$$

$$E = \{e_1, \dots, e_5\} = \{(v_1, v_2), (v_2, v_5), (v_5, v_5), (v_5, v_4), (v_5, v_4)\} \tag{3.2}$$

In this example, the orientation of the edges is not considered, in other words, the edge (v_1, v_2) is the same as the edge (v_2, v_1). In general, the two edges (u, v) and (v, u) are the same.

A list of terminologies can be done from the concepts explained until now:

1. The two vertices u and v are *end vertices* of the edge (u, v).
2. Edges that have the same end vertices are *parallel*.
3. An edge of the form (v, v) is a *loop*.
4. A graph is *simple* if it has no parallel edges or loops.
5. A graph with no edges is *empty*.
6. A graph with no vertices is a *null graph*.
7. A graph with only one vertex is *trivial*.
8. Edges are *adjacent* if they share a common end vertex.
9. Two vertices u and v are adjacent if they are connected by an edge: (u, v) is an edge.
10. The number of edges with v as an end vertex is the *degree* of the vertex v and it is written as $d(v)$. By convention, a loop counted twice and parallel edges contribute separately.
11. A *pendant vertex* is a vertex whose degree is 1.
12. An edge that has a pendant vertex as an end vertex is a *pendant edge*.
13. An *isolated vertex* is a vertex whose degree is 0.

Examples for the graph of the *Figure 3.4*:

- $v_4 \; and \; v_5 \; are \; end \; vertices \; of \; e_5.$
- $e_4 \; and \; e_5 \; are \; parallel.$
- $e_3 \; is \; a \; loop.$
- $The \; graph \; is \; not \; simple.$
- $e_1 \; and \; e_2 \; are \; adjacent.$
- $v_1 \; and \; v_2 \; are \; adjacent.$
- $The \; degree \; of \; v_1 \; is \; 1 \; so \; it \; is \; a \; pendant \; vertex.$
- $e_1 \; is \; a \; pendant \; edge.$
- $The \; degree \; of \; v_5 \; is \; 5.$
- $The \; degree \; of \; v_4 \; is \; 2.$
- $The \; degree \; of \; v_3 \; is \; 0, so \; it \; is \; an \; isolated \; vertex.$

Continuing with graph terminology, the *minimum degree* of the vertices in a graph G is denoted $\delta(G)$. If the minimum degree is 0, it means that there is an isolated vertex in G. The *maximum degree* of vertices in G is $\Delta(G)$.

Example for the graph of the *Figure 3.4*: $\delta(G) = 0;\ \Delta(G) = 5$

<u>Theorem</u>: The graph $G = (V, E)$, where $V = \{v_1, \dots, v_n\}$ and $E = \{e_1, \dots, e_m\}$, satisfies

$$\sum_{i=1}^{n} d(v_i) = 2m. \tag{3.3}$$

Consequently, every graph has an even number of vertices of odd degree.

Example for the graph of the *Figure 3.4*:

$$The\ sum\ of\ the\ degrees\ is\ 1 + 2 + 0 + 2 + 5 = 10; the\ same\ as\ 2 \cdot 5 = 10.$$

A simple graph that has every possible edge between all the vertices is a *complete graph*. A complete graph with n vertices is denoted as K_n. The first four complete graphs are the following:

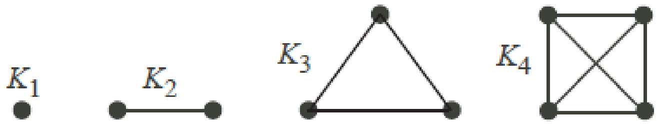


Figure 3.5. First four complete graphs

The graph $G_1 = (V_1, E_1)$ is a *subgraph* of $G_2 = (V_2, E_2)$ if:

1. $V_1 \subseteq V_2$ and
2. Every edge of G_1 is also an edge of G_2.

3.2.3. Walks, Trails, Paths, Circuits, Connectivity, Components

A *walk* in the graph $\mathrm{G} = (\mathrm{V}, \mathrm{E})$ is a finite sequence of the form

$$v_{i0}, e_{j1}, v_{i1}, e_{j2}, \dots, e_{jk}, v_{ik}, \tag{3.4}$$

which consists of alternating vertices and edges of G. The walk begins at a vertex. Vertices $v_{i_{t-1}}$ and v_{i_t} are end vertices of $e_{j_t} (t = 1, \dots, k)$. v_{i_0} is the *initial vertex* and v_{i_k} is the *terminal vertex*. k is the *length* of the walk. A zero length walk is only a single vertex v_{i_0}. It is permitted to visit a vertex or go through an edge more than once. A walk is *open* if $v_{i_0} \neq v_{i_k}$. Otherwise it is *closed*.

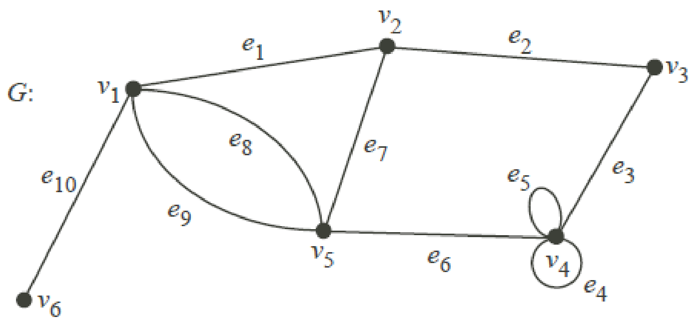


Figure 3.6. Example of an undirected graph

Example for the graph of the *Figure 3.6*:

$The\ walk\ v_2, e_7, v_5, e_8, v_1, e_8, v_5, e_6, v_4, e_5, v_4, e_5, v_4\ is\ open.$

$On\ the\ other\ hand, the\ walk\ v_4, e_5, v_4, e_3, v_3, e_2, v_2, e_7, v_5, e_6, v_4\ is\ closed.$

A walk is a *trail* if any edge is traversed at most once.

Example for the graph of the *Figure 3.6*:

$The\ walk\ in\ the\ graph\ v_1, e_8, v_5, e_9, v_1, e_1, v_2, e_7, v_5, e_6, v_4, e_5, v_4, e_4, v_4\ is\ a\ trail.$

A trail is a *path* if any vertex is visited at most once except possibly the initial and terminal vertices when they are the same. A closed path is a *circuit.*

Example for the graph of the *Figure 3.6*:

$The\ walk\ v_2, e_7, v_5, e_6, v_4, e_3, v_3\ is\ a\ path.$

$And\ the\ walk\ v_2, e_7, v_5, e_6, v_4, e_3, v_3, e_2, v_2\ is\ a\ circuit.$

The walk beginning at u and ending at v is called an $u - v$ walk. u and v are *connected* if there is a $u - v$ walk in the graph (then there is also a $u - v$ path!). If u and v are connected and v and w are connected, then u and w are also connected. A graph is *connected* if all the vertices are connected to each other.

The subgraph G_1 (not a null graph) of the graph G is a *component* of G if

1. G_1 is connected and
2. Either G_1 is trivial (one single isolated vertex of G) or G_1 is not trivial and $G_1$is the subgraph induced by those edges of G that have one end vertex in G_1.

3.2.3.1. Trees and forest

A *forest* is a circuitless graph. A *tree* is a connected forest. A *subforest* is a subgraph of a forest. A connected subgraph of a tree is a *subtree*. A subforest, in general, of a graph is its subgraph, which is also a forest.

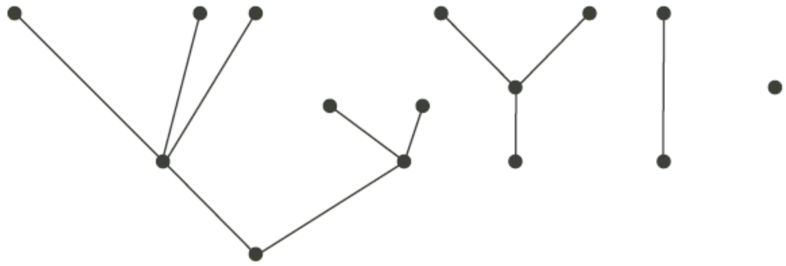


Figure 3.7. Four trees, which together form a forest

3.2.4. Directed graphs

Conceptually, a *directed graph* or *digraph* is formed by vertices connected by *directed edges*, *arcs* or *arrows*.

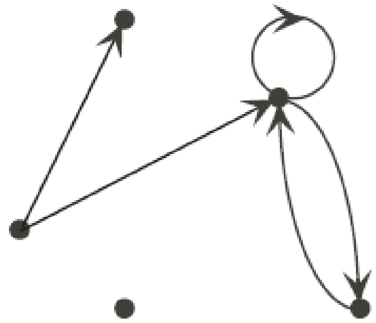


Figure 3.8. Example of a directed graph

Formally, a digraph is a pair (V, E), where V is the vertex set and E is the set of vertex pairs as in "usual" graphs. The difference is that now the elements of E are ordered pairs: the arcs (u, v) and (v, u) are different, they have the opposite direction. Now we have to take care about the definitions exposed before:

1. Vertex u is the *initial vertex* and vertex v is the *terminal vertex* of the arc (u, v). The arc is *incident out* of u and *incident into* v.
2. The *out-degree* of the vertex v is the number of arcs out of it, denoted as $d^+(v)$, and the *in-degree* of v is the number of arcs going into it, denoted as $d^-(v)$.
3. In the directed walk (trail, path or circuit), the equation is the same as (3.4), where v_{i_l} is the initial vertex and $v_{i_{l-1}}$ is the terminal vertex of the arc e_{j_l}.
4. When we want to use a graph (V, E) as a usual undirected graph, it is the *underlying undirected graph* of the digraph $G = (V, E)$, denoted as G_u.

5. Digraph G is *connected* if G_u is connected. The *components* of G are the directed subgraphs of G that correspond to the components of G_u. The vertices of G are connected if they are connected in G_u.
6. Vertices u and v are *strongly connected* if there is a directed $u - v$ path and also a directed $v - u$ path in G.
7. Digraph G is *strongly connected* if every pair of vertices is strongly connected. By convention, the trivial graph is strongly connected.
8. A *strongly connected component* H of the digraph G is a directed subgraph of G (not null) such that H is strongly connected, but if we add any vertices or arcs to it, then it is not strongly connected anymore.

3.2.4.1. Other definitions

In order to complete this introduction to graph theory, adding some other definitions has been considered opportune. These definitions have been extracted from the resource [7].

A *mixed graph* G is a graph with some directed edges and some undirected edges. It is written as $G = (V, E, A)$, where V is a set of vertices, E is a set of edges and A is a set of arcs.

A *loop* is a directed or undirected edge which starts and ends on the same vertex. Depending on the application, a loop may be permitted or not. In this context, an edge with two different ends is called a *link*. The term “multigraph” is used to mean that multiple edges are allowed.

A *simple graph* is an undirected graph without loops and with no more than one edge between two vertices. In this type of graphs, the edges form a set and every edge is a pair of *distinct* vertices. A simple graph with n vertices has a degree smaller than n.

A *weighted graph* is a graph with a number (weight) assigned to every edge. This weight might represent costs, lengths, time, capacities, etc. The weight of the graph is the sum of all the weights given.

A graph where each vertex has the same number of neighbours is a *regular graph*, what means that every vertex has the same degree.

A *finite graph* is a graph $G = (V, E)$, where V and E are finite sets. An *infinite graph* has an infinite set of vertices or edges or both.

A *bridge* (also called *cut-edge* or *cut arc*) is an edge whose deletion increases the number of connected components.

3.2.4.2. Directed trees

An *arborescence* is a directed graph in which, for a vertex u called the root and any other vertex v, there is only one directed path from u to v. An arborescence is thus the directed-graph form of a rooted tree (understood here as an undirected graph).

A digraph is *quasi-strongly connected* if one of the following states holds for each pair of vertices u and v (reference [1]):

i. $u = v$ or
ii. there is a directed $u - v$ path in the directed graph or
iii. there is a directed $v - u$ path in the directed graph or
iv. there is a vertex w so that is a directed $w - u$ path and a directed $w - v$ path.

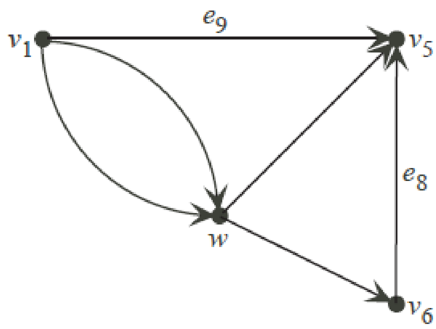


Figure 3.9. Example of a directed graph

Example for the graph of the Figure 3.9:

$The\ digraph\ is\ quasi - strongly\ connected\ and\ has\ one\ root, v_1.$

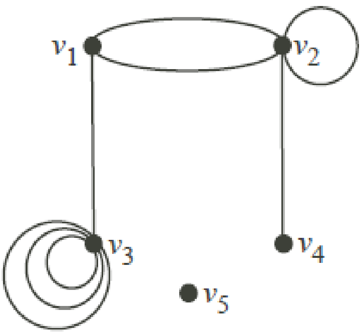
<u>Theorem</u>: A digraph has at least one root if and only if it is quasi-strongly connected.

If there is a root in the digraph, it follows from the definition that the digraph is quasi-strongly connected.

3.2.5. Matrices of Graphs

A graph $G = (V, E)$ is mathematically represented by the *adjacency matrix*, an $n \times n$ matrix $D = (d_{ij})$, where n is the number of vertices in G, V is the set of vertices and d_{ij} is the number of edges between v_i and v_j. The fact $d_{ij} = 0$ means that there is not the edge (v_i, v_j). If the graph is not directed, the matrix D is symmetric, so that $D^T = D$.

Example:

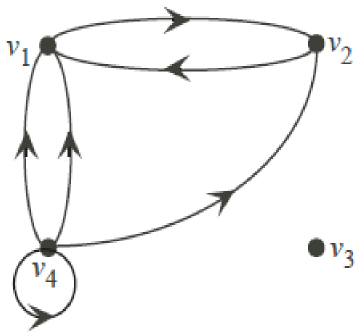


$$D = \begin{pmatrix} 0 & 2 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 1 & 0 & 3 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 3.10. Example of a graph

In the adjacency matrix of a directed graph G, each element d_{ij} is the number of arcs that come out of vertex v_i and go into vertex v_j.

Example:



$$D = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 1 \end{pmatrix}$$

Figure 3.11. Example of a directed graph

In this project, a simpler adjacency matrix will be used. This is an $n \times n$ matrix $D = (d_{ij})$, where n is the number of vertices in G, and

$$d_{ij} = \begin{cases} 1 \; there\ is\ a\ link\ from\ vertex\ i\ to\ vertex\ j \\ 0 \; otherwise. \end{cases}$$

In the case of the example of the Figure 3.11, the adjacency matrix would become to:

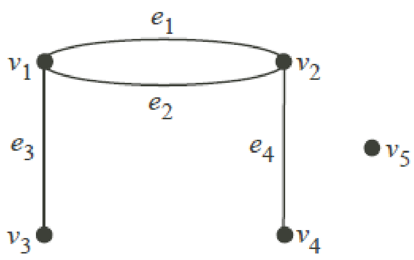
$$D = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

In other words, if two vertices are connected for more than one path, it is considered as if it was just one. But if the path can be traversed in the two directions, it is represented by two different elements in the matrix.

A graph can also be represented by the *incidence matrix* A, of $n \times m$ dimension, where m is the number of edges in G. It changes if the graph is undirected or directed. For an undirected graph:

$$a_{ij} = \begin{cases} 1 \ if \ v_i is \ an \ end \ vertex \ of \ e_j \\ 0 \ otherwise. \end{cases}$$

Example:



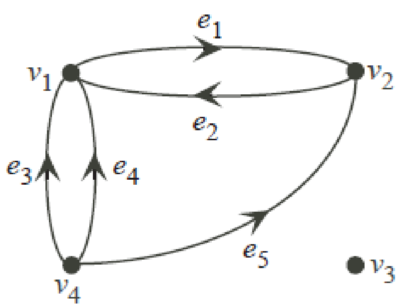
$$A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 3.12. Example of a graph

In case the graph is directed, the element a_{ij} can be:

$$a_{ij} = \begin{cases} 1 \ if \ v_i is \ the \ initial \ vertex \ of \ e_j \\ -1 \ if \ v_i is \ the \ terminal \ vertex \ of \ e_j \\ 0 \ otherwise \end{cases}$$

Example:



$$A = \begin{pmatrix} 1 & -1 & -1 & -1 & 0 \\ -1 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Figure 3.13. Example of a directed graph

3.2.6. Types of graphs

To sum up the differences between undirected graphs and directed graphs (or digraphs), a comparative table has been considered interesting and easier to consult [10].

<table>
<tr><th>UNDIRECTED GRAPH</th><th>DIRECTED GRAPH / DIGRAPH</th></tr>
<tr><td colspan="2">Vertices</td></tr>
<tr><td>Edge</td><td>Directed edge / arc</td></tr>
<tr><td rowspan="2">Degree</td><td>Out-degree</td></tr>
<tr><td>In-degree</td></tr>
<tr><td>Walk</td><td>Directed walk</td></tr>
<tr><td colspan="2">Eulerian concept</td></tr>
<tr><td colspan="2">Hamiltonian concept</td></tr>
<tr><td>Connected</td><td>Strongly connected</td></tr>
<tr><td>Tree</td><td>Arborescence</td></tr>
</table>

Table 3.2. Undirected/Directed graph comparison

Related to transportation, a graph classification could be [10]:

- $Road\ graph \begin{cases} node\ \ \rightarrow intersection \\ arc\ \ \rightarrow street\ section \\ Directed\ graph\ by\ definition \end{cases}$

- $Superficial\ public\ transportation\ graph\ \begin{cases} vertex\ \ \rightarrow stop \\ arc\ \ \rightarrow section\ among\ stops \end{cases}$

- $Rail\ transport\ graph \begin{cases} vertex\ \ \rightarrow station \\ arc\ \ \rightarrow \begin{cases} section\ among\ stations \\ interchange \end{cases} \end{cases}$

- $On\ foot\ graph \begin{cases} vertex\ \ \rightarrow bifurcation\ point\ of\ possible\ routes \\ edges\ (no\ arcs) \end{cases}$

3.2.7. Algorithms

Lots of algorithms are the basis of software that solve some transportation problems. Here, a summary of four of them as example is considered interesting of study.

- Shortest path problem → Dijkstra's algorithm

Problem: to find a path between any pair of nodes (called initial and final) of a directed or undirected graph. A cost magnitude or concept (distance, time, cost, generalized cost…) must be associated to each edge or arc.

Solution: Dijkstra's algorithm is an iterative algorithm that acts like an "oil stain". The algorithm creates a tree of shortest paths from the starting vertex (the source) to all other vertex in the graph, by building a set of nodes that have minimum distance (sometimes someone talks about distance meaning cost magnitude) from the source [8].

- Transportation problem (factory – warehouse) → Vögel algorithm

Problem: to determine the transportation policy that minimizes the total cost of transportation.

Solution: Vögel's algorithm is a technique for finding a good initial feasible solution. It works as the following steps [9]:

1) Balance the given transportation problem if either (total supply>total demand) or (total supply<total demand)
2) Determine the penalty cost for each row and column by subtracting the lowest cell cost in the row or column from the next lowest cell cost in the same row or column.
3) Select the row or column with the highest penalty cost (breaking ties arbitrarily or choosing the lowest-cost cell).
4) Allocate as much as possible to the feasible cell with the lowest transportation cost in the row or column with the highest penalty cost.
5) Repeat steps 2, 3 and 4 until all requirements have been meet.
6) Compute total transportation cost for the feasible allocations.

- Travelling salesman problem (TSP) → Maximum savings algorithm

Problem: Given a set of cities and known the distance of going from any city to any other, it is about finding a cycle that goes through all cities just once, so that the distance is minimum [10].

<u>Solution</u>: this algorithm is based in the *centrality* concept (for one vertex, sum of the distances of the edges of the vertex). The vertex with the minimum centrality is considered the centre C of the graph.

From now on, the *saving* concept of the edge (i, j) is defined as the distance reduction that is obtained going from the vertex i to the vertex j going through the edge in question instead of going through the centre.

In consequence, finding the cycle of minimum distance is equal to find the cycle of maximum saving. Therefore, the saving of each edge is calculated and that one with the maximum value is taken and then the others in descending order, if possible (in other words: not doing partial cycles nor pitchforks).

- Vehicle Routing Problem → Heuristic algorithm of Clarke & Wright

<u>Problem</u>: given a vehicle or a set of vehicles with a capacity smaller than the demand quantity, the problem to solve is to find the different partial cycles, called *petals*, which the vehicles have to complete, so that the sum of the petals distances is minimum [11].

<u>Solution</u>: the Clarke & Wright algorithm follows these steps:

1) All the savings are calculated for each edge, regarding the warehouse.
2) The savings are ordered in descending order.
3) For each pair, the following situations are considered:
   a) If i or j don't belong to any petal, the corresponding petal is created.
   b) If i belongs to the petal p and is adjacent to the warehouse and j doesn't belong to any petal and the demand is admissible, the client j is added to the petal, in adjacent position to the warehouse.
   c) If i belongs to the petal p, j belongs to the petal p' and the sum of the two demands is smaller than the transportation capacity and both are adjacent to the warehouse, both petals join together in one.
   d) If i and j already belong to the same petal, the pair is ignored.
   e) If i and j belong to different petals but one of them o both are not adjacent to the warehouse, the pair is ignored.
4) If all clients are already assigned to a petal, the possibility of fusing together is analysed.

3.3. Social networks analysis

3.3.1. Introduction

Social network analysis, also called structural analysis, is a strategy for looking into social networks and structures, by using mathematical formulations extracted from graph theory. Data is an input in social network analysis in the form of vertices and edges [13].

Real world situations or physical environments can be represented with different types of network structures, depending on relations and transitory structures. To confirm this state, a social network topology changes into a complex and a scale free network. Road networks and their spatial relations generate specific network structures. This fact is visible with the considerably different characteristic when comparing road networks with others [12].

Complex networks present characteristic topological features which specify their connectivity and influence the processes executed on the social network. The analysis, division and synthesis of complex networks depend on the use of measurements that are able to express the most relevant topological features. The main existing measurements will be presented and analysed [6].

Complex network research has been a focus of attention only recently. This fact is explained by the discovery that real networks have characteristics which are no explained by uniformly random connectivity. Instead, networks that come from real data might involve other structural features, like community structure, power law degree distributions and hubs [6].

The network representations have been commonly used in diverse problem areas. An Internet network and a web site link network are among common examples. In Biology and chemistry, networks are used to explain, for example, how proteins and atoms interact with each other. In sociology, researchers study relationships between actors with representations based on social networks. All these representations served as a big tool to help researchers to analyse various technological or social phenomena [12].

Current analysis tools are expensive, consume a lot of time and require rigorous data in order to have reliable results. In consequence, a quick and inexpensive study through networks is the best way to preliminarily analyse traffic networks. This means to realise a literature review into papers, books, articles, etc. written by researchers and specialists on this field in order to contrast different points of view and analyse which parameters are considered significant.

3.3.2. Research

According to Park and Yilmaz (2010) in their investigation "A social network analysis approach to analyze road networks" [12], there are two main concepts in analysis of road network. The first one is the centrality of nodes in a road network, which is used to detect important nodes and find nodal characteristics in networks. They compute three types of centralities: degree, closeness and betweenness.

The second concept they introduce to road network analysis is the entropy of distributions computed from the network. Entropy explains the uncertainty from a probability distribution and is commonly used in information coding theory. In this context, graph entropy is used to measure information encoded in the distributions generated from the road networks.

When describing the methodology they carried out, these four parameters are described mathematically:

- Degree: one of the more widely used measures and counts the number of direct connections a node has to other nodes in the network. The large number of degree means how many ways are linked at a junction point. It may imply that higher degree of nodes could have crowded traffic at those points than lower degree nodes.

$$d(i) = \sum_{j} m_{ij} \tag{3.5}$$

  Where $d(i)$ is the degree centrality of node i and $m_{ij} = 1$ if there is a link between the i and j vertices and $m_{ij} = 0$ otherwise.

- Closeness: in social network, closeness indicates how a node is close to the other node. It is computed as the shortest geodetic path between two nodes. Since closeness finds the shortest path in the whole network structure, it considers the global connectivity of network structure.

$$C_{Closeness}(N_i) = \frac{1}{\sum_{j=1}^{n} d_{ij}} \tag{3.6}$$

  Where d_{ij} is a geodetic distance between N_i and N_j.

- Betweenness: number of times a node is crossed by shortest paths in the graph. This parameter explains how a node can control the other nodes which have no direct connectivity between them. In a road network application, betweenness tells how the intersection points are important to reach the destination from the start points.

$$BC_i = \frac{\sigma_{jk(i)}}{\sigma_{jk}} \tag{3.7}$$

Where:

σ_{jk} Total number of shortest paths from node j to k

$\sigma_{jk(i)}$ Number of shortest paths from node j to k that pass through node i

- Entropy: is a quantitative measurement used to explain the probability distributions. If the probability has uniform distribution, it is called the uncertainty of the distribution is uniform (the states of the system are highly disordered). On the contrary, when the probability distribution has not uniform distribution, some of states could be predictable. Let P be the probability distribution on the node set of $V(G)$ and $p_i \in [0,1]$. Thus, the entropy of the graph G is:

$$H(G,P) = \sum_{i=1}^{N} p_i \log_2(p_i) \tag{3.8}$$

The results after analyzing four cases (unweighted and undirected network structures) showed that generally the downtown area (in the example they use, the downtown area has nearly grid type of network) tends to have more nodal degree than the residential area (a more circular/ring area). When degree entropy is computed, it seems that entropy of degree distribution does not discriminate different types of road network topologies. Then, after studying the closeness and the betweenness with both entropies, the study samples explain that the downtown area has more alternative shortest routes than the residential area.

In conclusion, in a downtown area which has grid-like network topology has higher entropy than the residential area having radiant network topology.

The paper “Social Network Analysis Approach for Improved Transportation Planning” (reference [13]) uses SNA (Social Network Analysis) to analyze transportation networks and corroborate the effectiveness of SNA as a complementary tool for improved transportation planning.

The authors use SNA centrality measures to reach the goals of their investigations. They defend that if money were invested to improve an intersection with the highest centrality, not only would that intersection improve, but travel time throughout the network would decrease. For this reason, a central intersection should be given more focus in order to maintain consistent and nonextended travel time. The centrality measures they use are the following:

- Degree centrality: explained previously in (3.5).
- 2-step reach centrality value: sums the number of nodes within 2 steps or links of a particular node, i.e. it is basically the sum of the degree centralities of the adjacent nodes.

$$S_v = \sum_{i \in M(v)} d(i) \tag{3.9}$$

Where $d(i)$ is the degree of node i; S_v is the 2-step centrality value; and $M(v)$ is the set of neighbors of v.

- Bonacich power: evaluates the road's centrality as a function of how many connections it has, and how many connections the roads in the neighbourhood have. The more connections the neighbourhood roads have, the more central the road is.

$$BP_i = a\,(I - b * R)^{-1}\,R * 1 \tag{3.10}$$

Where:

BP_i	Bonacich power of node i
a	Scaling vector (set to normalize the score)
b	Scope of the weight of the centrality
R	Adjacency matrix (can be valued)
I	Identity matrix (1s down the diagonal)
1	Matrix of all ones

- Eigenvector centrality: determines nodal centrality based on the closeness centrality of adjacent nodes. It is a function of how many intersections lie between any two selected intersections. It is another type of weighted centrality measure in which the centrality of adjacent nodes contributes to the overall centrality of the studied node. For network $G = (V, E)$ and an adjacent matrix A consisting of elements $a_{v,t}$, the eigenvector centrality score of v is obtained as:

$$x_v = \frac{1}{\lambda} \sum_{t \in M(v)} x_t = \frac{1}{\lambda} \sum_{t \in G} a_{v,t}\, x_t \tag{3.11}$$

Where $M(v)$ is the set of neighbors of v; $a_{v,t} = 1$ if vertex v is linked to vertex t and $a_{v,t} = 0$ otherwise; and λ is a constant.

- Betweenness centrality: explained before in (3.7).

Although simple centrality measures look at total traffic volume only at individual intersections, the measures used in the research they did weight intersections based on their location and connection strength.

After the studies realized, they saw that Eigenvector and Bonacich power explain the important role of nodes in a social network. In a road network, the degree of a node can create connectivity in and popularity of an intersection with respect to spatially neighboring intersections. A node with a high Bonacich value has more opportunities and alternatives than other nodes to reach anywhere in the network. Eigenventor values reflect the closeness of nodes in terms of global or overall network structures.

Betweenness indicates the controlling power that a subject node has over other, unconnected nodes. Thus, high betweenness means that the subject node connects the other nodes.

Two-step reach is considered a good measure of accessibility if networks have relatively harmonic traffic counts. However, it is not recommended in nonuniform traffic networks because it does not take nodal weight (traffic count in this case) into consideration.

It is recommended that all of the centrality measures be used in parallel, because the exact interpretation of each one is yet to be completely developed.

Two other researchers who used some of the same measurements such as the above ones were Matthias Kowald & Kay W. Axhausen in their investigations [16]. They described *density* as the proportion of all possible connections between actors of a given graph and all actual connections. Therefore, density values and network sizes should be considered together. Measurements of degree and betweenness centralization are complementary to the density concept. Both are global measurements for a personal network topology.

The last research about social network analysis taken into account in the Master Thesis in question is the one carried out by Parthasarathi [17], whose aim was to develop quantitative measures that capture various aspects of the network structure, using data from fifty metropolitan areas across the US. The argument presented is that while the metropolitan transportation network need not be the only indicator of travel in a region, an understanding of relationship between network architecture and travel is essential for the design of sustainable and efficient cities. The following measures of street network structure were estimated for each of the fifty metropolitan areas:

- Hierarchy: Entropy and Percentage of freeways
- Topology: Treeness, Completeness and Average circuity
- Scale: Street density

- Entropy: explained previously in (3.8). Entropy captures the differentiation that exists among road networks. Roadways networks typically have a very high proportion of local streets compared to other functional categories. The differentiation in network structure, as measured by the entropy variable, identifies a variation from this typical condition. Therefore, a higher entropy measure indicates the presence of higher functional classification links in the network such as arterials compared to local streets.
- Percentage of freeways: is also designed to capture the hierarchy in real-world street networks. It focuses specifically on the freeways in the area. The percentage of freeways is estimated as:

$$\%F = \frac{L_f}{L} * 100 \tag{3.12}$$

Where L_f is the number of freeway kilometers in the area and L is the total roadway kilometers in the area.

- Treeness: based on the two basic structures of a planar transportation network: circuit and tree. As has been explained before, a circuit is a closed path (with more than two links) that begins and ends at the same node. A tree is defined as a set of connected lines that do not form a complete circuit. The treeness for each street network was estimated as:

$$\phi_{tree} = \frac{L_t}{L} \tag{3.13}$$

Where L_t is the length (in km) of street segments belonging to a branch network in the area and L is the total roadway kilometers in the area.

The treeness measures capture the differences in topology and connection patterns that exist among real-world street network.

- Completeness: captures the level of completeness in the network using a link-node approach. This concept is explained in advance in the Figure 3.5 part. The number of links in a real world network is typically less than the maximum number of links and the completeness index used captures this difference. Considering a road network with E links (edges) and V nodes (vertices), the completeness of the street network is defined as:

$$\rho_e = \frac{E}{V^2 - V} \quad (3.14)$$

Nodes in network	Maximum number of 1-way links	Maximum number of 2-way links	Completeness
2	1	2	100%
3	3	6	100%
4	6	12	100%
5	10	20	100%
6	15	30	100%

Table 3.3. Maximum number of links in sample networks

- Average Circuity: the ratio of the shortest path network distance to the Euclidean or straight-line distance between an origin and destination (OD) pair. It is designed to capture the inefficiency in the network from the point of view of a traveler. The average circuity for a subsample of OD pairs in each area wan estimated as:

$$C = \frac{D_n}{D_e} \quad (3.15)$$

Where:

C	Average circuity in the metropolitan area
D_n	Sum of the network distance (km) between all OD pairs in the subsample
D_e	Sum of the Euclidean distance (km) between all OD pairs in the subsample

- Street density (ρ_{lm}): for each metropolitan, it is estimated as:

$$\rho_{lm} = \frac{L}{A} \quad (3.16)$$

Where L is roadway kilometers in the area and A is the size of the area (km^2).

The estimated value has a unit of $1/km$. This measure differs from the completeness measure in that it provides a measure of the size of the actual network in comparison to the size of the urban area.

- Accessibility: refers to the ease of reaching destinations or activities. The cumulative opportunity is one of the many methods to estimate accessibility in the region; it estimates the number of destinations (O_t) that can be reached in a given time threshold. A similar approach is used here to measure the average number of people that can be reached in t minutes by automobile at uniform average metropolitan density. The accessibility measure is estimated as:

$$O_t = \pi * \left[\frac{S_n \cdot t}{C}\right]^2 * \rho_{pm} \tag{3.17}$$

Where:

ρ_{pm}	Urban area population density ($person \cdot km^{-2}$)
t	Time threshold (in minutes)
S_n	Average network speed in km/h
C	Average circuity, as estimated above

Parthasarathi's research aimed to develop quantitative measures that capture various aspects of network structure. The influence of these measures on system performance was then tested using two linear regression models. The results from both models confirm that the quantitative measures of network structure affect the system performance, after controlling for independent variables that are non-network based.

The first model shows the influence of network treeness while the second model shows the influence of accessibility, street density, completeness and the percentage of freeways in the urban area.

The author confirms that it is absolutely essential that we consider network architecture in the design of new transportation facilities. We are at a stage worldwide where the transportation systems in the developed world are in mature stage while the developing countries are just getting started with designing new infrastructure. This provides us a valuable opportunity where we can apply the lessons learned from the mature transportation systems to help design efficient and sustainable facilities.

3.4. Traffic network topologies

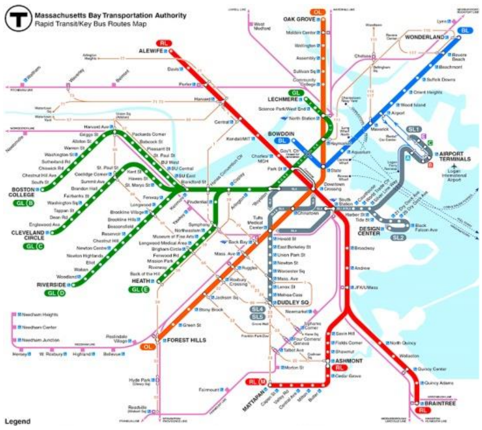
3.4.1. Introduction

Many studies propose measures to characterize networks for different types of applications: physics, geography, the Internet, and biological and social systems. Some examples in these fields include Kansky (1963), Hagget and Chorley (1967) and Garrison and Marble (1974). Kansky used graph theory to develop measures to quantify the special structure of transportation networks. Kansky considered three main indices as nodal importance and network complexity in transportation networks: Alpha, Beta, and Gamma indices, all measures of connectivity. These and other measures are going to be defined later. Their studies, however, were limited by computational resources.

More recently, relations between network shape and transportation system arrangement have been studied, due to advances in computers, in a considered number of works, which include, for example, road and air networks (e.g., Gastner and Newman, 2006; Reggiani et al., 2011) and subway networks (e.g., Derrible and Kennedy, 2010).

Moreover, random, scale-free and small-world network structures were found to be particularly significant. In random graphs, nodes are randomly linked with an equal probability of placing a link between any pair of nodes. As defined in Barabási and Albert (1999), a scale-free network has a nodal degree distribution that follows a power law. Thus, some nodes have a degree that greatly exceeds the average. Small-world networks, on the other hand, are densely connected in local regions, creating highly connected subgraphs with few crucial connections between distant neighbours. Wu et al. (2004) showed that scale-free type characteristics exist in urban transit networks in Beijing (example of the bus landscape in the Figure 3.14), while Latora and Marchiori (2002) suggested that the Boston subway system has a small-world network structure (seen in the Fgure 3.15).

Figure 3.14. Beijing Bus Landscape [Source: https://www.beijingcitylab.com/projects-1/3-bus-landscapes /]



Fgure 3.15. Boston Subway [Source: http://www.boston-discovery-guide.com/boston-subway.html]

Watts and Strogatz (1998) studied the performance of neural and power grid networks in terms of shortest average path length and clustering. They found that some neural and power grid networks have the shape of small-world networks. Zhao and Gao (2007) studied the performance of small-world, scale-free and random networks in terms of total travel time and traffic volume in the context of a traffic network.

Other works have studied connections between system topology and performance. In the context of transit networks, Li and Kim (2014), for example, proposed a connectivity-based survivability measure to study the Beijing subway system (visible in the *Figure 3.16*). Similarly, Rodríguez-Núñez and García-Palomares (2014) presented a vulnerability measure and applied it to study the Madrid Metro, whose system is shown in the Figure 3.17.

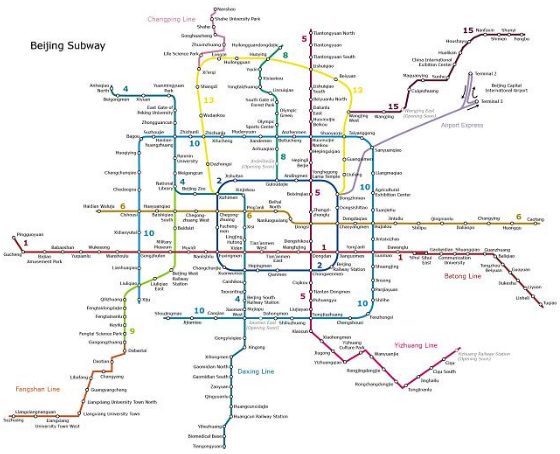


Figure 3.16. Beijing Subway System [Source: http://www.beijingchina.net.cn/transportation/subway.html]

Figure 3.17. Madrid Subway System [Source: http://www.planometromadrid.org/]

In work by Derrible and Kennedy (2010), the robustness of 33 metro systems around the world was investigated. In their work, robustness is defined in terms of cyclicity. Cyclicity is a connectivity measure that like average degree is used to characterize a network topology herein. Exploiting noted relationships between these real system layouts and scale-free and small-world network structures, they provided strategies for improving performance of both small and large systems. They provide a comprehensive review of related works, as well. O'Kelly (forthcoming) discusses the role of hubs in network vulnerability and resilience of various network structures.

Finally, Reggiani et al. (forthcoming) propose the use of connectivity as a unifying framework for considering resilience and vulnerability in relation to transport networks. They test this concept through a synthesis of related literature. Numerous additional articles consider the performance of specific transportation networks under various resilience-related measures, but they do not investigate the general role of network topology.

3.4.2. Research

In their investigations, Zhang et al. used 17 networks topologies of the 25 topologies discovered in a search [2]. These basic structures supply the fundamental elements to build larger comparable networks. In Figure 3.18, the chosen networks are represented, as well as the extrapolation to larger network sizes (with a bigger number of nodes and links).

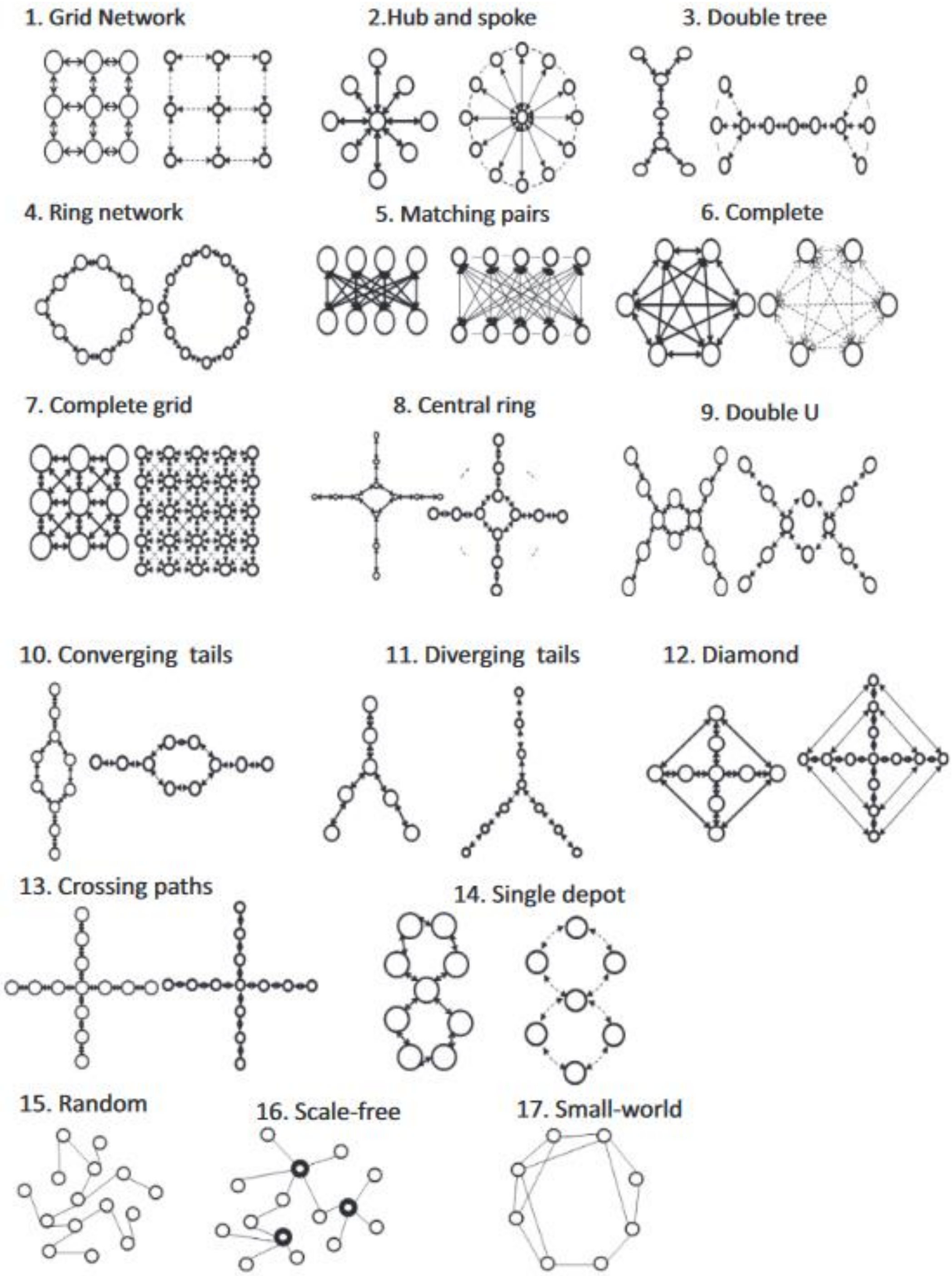


Figure 3.18. Network topology and extrapolation (Zhang et al. [2], 2015, p. 37)

According to them, a network topology can be characterized in terms of connectivity and accessibility measures. Connectivity measures are used to evaluate redundancies and connectedness, while accessibility measures are used to compare the relative position of nodes in the network. In their works, they studied relationships between network topology and vulnerability or similar measures, by investigating the role of network topology in system resilience. Here, the resilience is considered the innate ability of the system to absorb externally induced changes, and also the cost-effective and efficient, adaptive actions that can be taken to preserve or restore performance post-event.

Before describing the measures, to define some parameters is necessary:

e	Number of links in the graph
v	Number of nodes in the graph
G	Number of sub-graphs in the graph
n_i	Number of arcs incident on node i
d_{ij}	Distance of the shortest path between O – D pairs (i, j)
$Cycle_i$	Number of times random walk cycled back to node i
$\|R\|$	Number of random walks
σ_{jk}	Total number of shortest paths from node j to k
$\sigma_{jk(i)}$	Number of shortest paths from node j to k that pass through node i

As connectivity measures, Zhang et al. used the following six:

- Cyclomatic number: number of fundamental circuits in the network

$$\mu = e - v + G, \quad where\ \mu \geq 0 \tag{3.18}$$

- Alpha index: ratio of number of cycles to possible maximum number of cycles.

$$\alpha = \frac{\mu}{2v - 5} \qquad where\ 0 \leq \alpha \leq 1 \tag{3.19}$$

- Beta index: ratio between number of links and number of nodes, equivalent to average degree

$$\beta = \frac{e}{v} \qquad where\ \beta \geq 0 \tag{3.20}$$

- Gamma index: ratio of number of links to maximum possible number of links

$$\gamma = \frac{e}{3(v - 2)} \qquad where\ 0 \leq \gamma \leq 1 \tag{3.21}$$

- Average index: average number of arcs incident on the nodes

$$\bar{d} = \frac{\sum_i n_i}{v} \qquad where\ \bar{d} \geq 0 \tag{3.22}$$

- Cyclicity: number of times random walk led to a cycle back to a previously visited node/number of random walks

$$\hat{C} = \frac{\sum_{j=1}^{n} Cycle_i}{|R|} \qquad where\ 0 \leq \hat{C} \leq 1 \tag{3.23}$$

And as accessibility measures, Zhang et al. worked with these three ones:

- Diameter: the maximum distance among all shortest distances between all O-D pairs in the network

$$D = \max(d_{ij}) \tag{3.24}$$

- Average Shimbel index: average of the sum of the lengths of all shortest paths connecting all pairs of nodes in the network

$$A_i = \frac{\sum_{j=1}^{n} d_{ij}}{v-1} \tag{3.25}$$

- Betweenness centrality: explained before in (3.7).

Analysing the results from large networks (with 100 nodes) gave several important discoveries. In general, resilience measures increase with average degree and greater cyclicity, but decrease with network diameter. Thus, the complete network has the highest values of resilience, while the ring network has the lowest.

The general ordering of the network topologies from most resilient to least resilient was found to be: complete, matching pairs, complete grid, diamond, grid, single depot, central ring, hub-and-spoke, double-U, converging tails, random, scale-free, small-world, crossing path, double tree, diverging tails and ring network. This ordering indicates a strong connection between resilience and average degree.

The studied network topologies might also be categorized by type of connections: group 1 (highly connected) – grid, matching pair, complete grid and diamond networks; group 2 (centrally connected) – hub-and-spoke, double tree, ring, diverging tails and crossing paths networks; group 3 (circuit-like connected) – central ring, double U and converging tails; group 4 (randomly connected) – random, scale-free and small-world networks.

Another interesting work, related to the parameters used on it, is the one carried out by Freiria et al. in 2015 [14]. This work is due to road networks interruptions caused by natural disasters are, every time, more frequent and their consequences, more varied. The main goal of their work was to identify the most important roads in a network, defined as those whose interruption would cause the most significant consequences (in terms of connectivity loss and average distance increase among the nodes).

They propose a new model to evaluate the most important roads in the network by applying the biclustering technique, identifying patterns of attributes (road performance measures) and patterns of roads (connectivity patterns).

In order to reach the main goal of their work, the indicators were selected on the basis of analysis of centrality, cohesion and density and also based on the literature review. The indicators selected are explained as follow, excluding those ones Zhang et al. used in their investigations, of which only extra information focused on Freiria et al.'s work is going to be given.

- Alpha index: synonym of expansiveness and refers to the effect of each road's out-degree on the probability that it will have links to other roads. Equation exposed before in (3.19).
- Betweenness: important for analysing the hierarchical structure of the network and ranking the links of the network. Equation exposed before in (3.7).
- Bonacich power: explained before in (3.10).
- Cluster index: measure of the likelihood that any two associates of a road will actually be associates.

$$C_i = \frac{M\_i}{k_i(k_i - 1)/2} \tag{3.26}$$

Where:

C_i	Clustering coefficient of node i
M_i	Number of pair of neighbours of node i that are connected
k_i	Number of neighbours for node i

- Cutpoint: node that if removed, would leave the network with disconnected components, i.e. the structure would become divided into-unconnected parts.

$$cut(A,B) = \sum_{i \in A, j \in B} w_{ij} \tag{3.27}$$

Where node i belongs to the component A and j belongs to the component B and w_{ij} is the weight of the connections between i and j.

- Degree: explained before in (3.5).
- Fragmentation: proportion of pairs of vertices that cannot reach each other. Given a matrix R in which $r_{ij} = 1$ if i can reach j and $r_{ij} = 0$ otherwise, the fragmentation can be defined as:

$$F_i = \frac{2\sum_i \sum_{j<1} r_{ij}}{n(n-1)} \tag{3.28}$$

F_i is the fragmentation of the node i. Fragmentation centrality of a node i is the difference in the total score with the node i included in the network n and the score with the node removed from the network n.

The results of their investigations showed that the higher the level of connectivity of a link, the greater its interruption impact on the network. It is worth noting that while a link's high level of connectivity might be considered a strength in a normal situation, it can be viewed as vulnerability in a road interruption scenario.

The thesis presented by B. Janis in 2014 [15] is about shape grammars, which are promising tools for designing urban areas. They can be applied in urban design methods, are applicable to different planning sites, and are suitable for solving interdisciplinary planning tasks. Three methodologies are proposed in his thesis to define and evaluate grammars and their effects in network design.

In each era, network patterns were designed for specific requirements, by using the available technologies of each era. The network patterns of one era replaced these of the prior era, and many patterns were passed on to following generations.

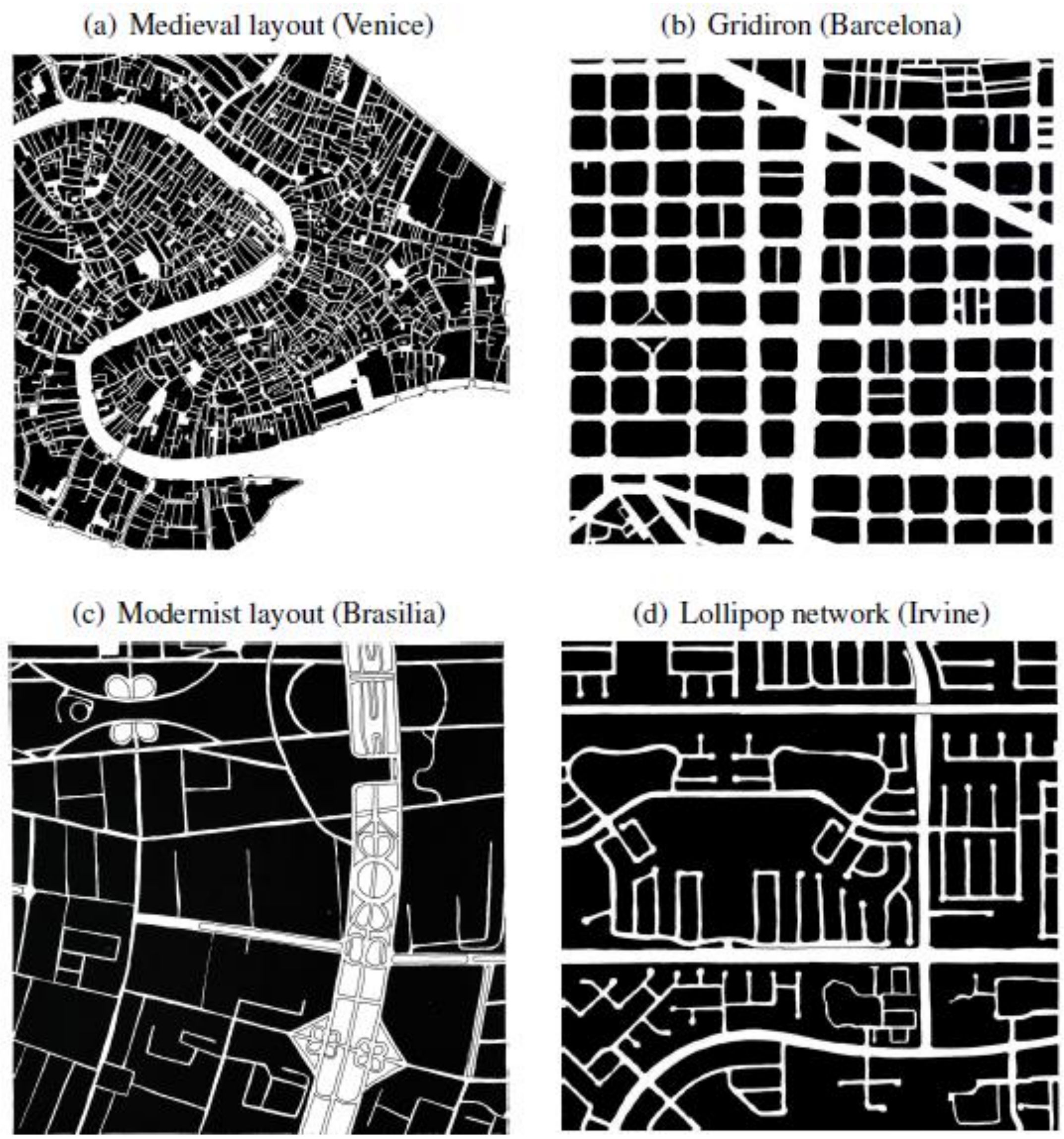


Figure 3.19. Example network patterns from Jacob (1993) [Source: in reference [15], p.2]

The literature review that Janis did seems to be interesting for the Master Thesis in question.

Medieval structures as Venice (*Figure 3.19* (a)) contrast with baroque layouts and gridirons as Barcelona (*Figure 3.19* (b)), and these differ again from garden cities, modernist layouts as Brasilia (*Figure 3.19*(c)), as well as lollipop networks as Irvine (*Figure 3.19*(d)).

Medieval networks grew through a largely self-organized, historical process. They therefore contrast to more recent patterns, which have been realized over a short period of time with a specifically designed pattern in a rather top-down approach, for example gridirons.

In Europe, the medieval networks were built mainly for pedestrians and foot carriages. Later, industrialization had an important impact on transportation. Public transportation (trains and tramways for urbanized areas) emerged as a major transport mode at the turn of the century (1900).Regarding the urban design, gridirons were suitable for such modes and were designed and applied. Later, automobiles enabled individual transportation. Modernist and lollipop networks were designed, leading the new car age.

Current and future technological developments allow now more optimized travel in ever more congested and complex networks.

In the future, vehicle assistance might enable a more generic design, which could be detached from the existing network patterns. Advances in technology will allow to enhance transportation systems in the future, which will influence spatial development again.

It is evident that technology and urban design interdependent. Advances in technology enable new transport modes, which require new infrastructure, both for transportation and spatial developments. New modes and new infrastructure enable further cost reduction, improve productivity and increase economic well-being. It is expected that urban network design and redesign will prevail in urban planning due to population changes and continuous technological progress.

Changes on the travel demand side are difficult to predict in long-term forecast models. Therefore, planners aim at reliable and robust transport systems which allow to absorb variation in travel demand and infrastructure supply. Reliability and robustness of the transport systems has become a priority as there are changing or increasing travel demand, longer trip distance and higher flows and therefore also capacity problems and delays due to the capacity limits on the infrastructure side.

Summing up, it can be stated that our transport systems face two major tasks: The systems should be as efficient and productive as possible in the current state, and at the same time reliable and robust to short and long term changes.

3.5. Intersections

An intersection is the area where two or more streets join or cross at-grade [18]. The intersection includes the areas needed for all modes of travel: pedestrian, bicycle, motor vehicle, and transit. Intersections are a key feature of street design in four respects:

- Focus of activity – The land near intersections might contain a set of travel destinations.
- Conflicting movements – Intersections typically concentrate several movements: pedestrian crossings, and motor vehicle and bicycle turning and crossing movements.
- Traffic control – At intersections, movements of users is assigned by traffic control devices such as yield signs, stop signs, and traffic signals. Traffic control helps to organize traffic and decrease the potential for conflict.
- Capacity – It is defined as the number of users that can be accommodated within a given time period. In several cases, capacity of the intersecting roadways is limited by traffic control at intersections.

The *major street* is typically the intersection street with greater traffic volume, larger cross-section, and higher functional class. The *minor street* is the intersecting street likely to have less traffic volume, smaller cross-section and lower functional classification than the major street.

Two geometric features are common to all intersections. The *angle of intersection* is formed by the intersection streets' centerlines. Where the angle of intersection departs significantly (more than approximately 20 degrees) from right angles, the intersection is referred to as a *skewed intersection*.

Intersection legs are those segments of roadway connecting to the intersection. The leg used by traffic approaching the intersection is the *approach leg*, and that used by traffic leaving is the *departure leg*.

Intersections can be categorized into four major types, as illustrated in Figure 3.20:

- Simple intersections: maintain the street's typical cross-section and number of lanes throughout the intersection, on both the major and minor streets.
- Flared intersections: expand the cross-section of the street (main, cross or both).
- Channelized intersections: use pavements markings or raised islands to designate the intended vehicle paths.
- Roundabouts: has one-way traffic flow circulating around a central island.

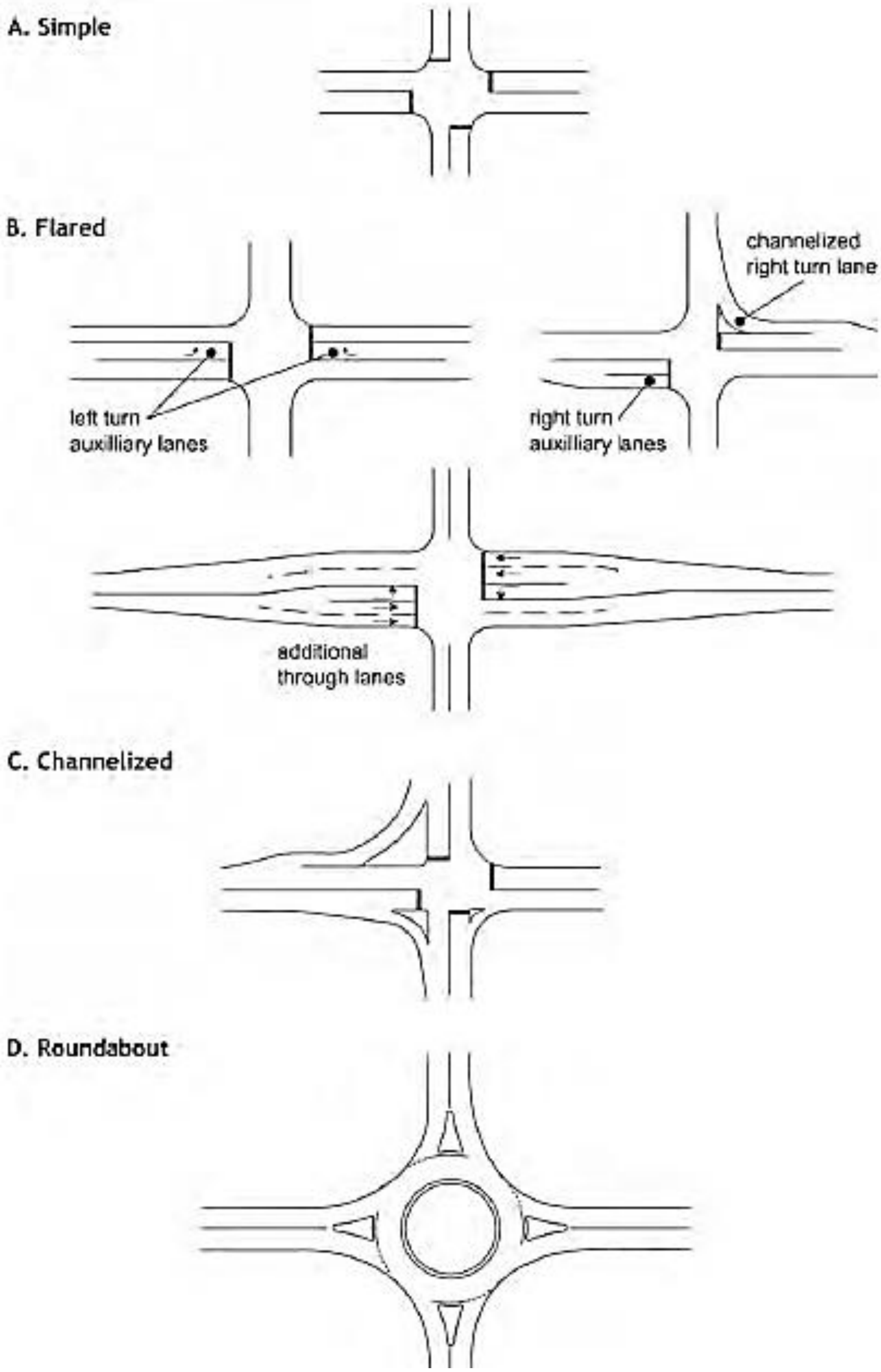


Figure 3.20. Intersection types [Source: in reference [18], p. 6-11]

Most intersections have three or four legs, but multi-leg intersections (five and six-legs intersections) are not unusual. Ideally, streets in three-leg and four-leg intersections cross at right angles or nearly so. Typical intersection configurations are shown in the next figure (Figure 3.21).

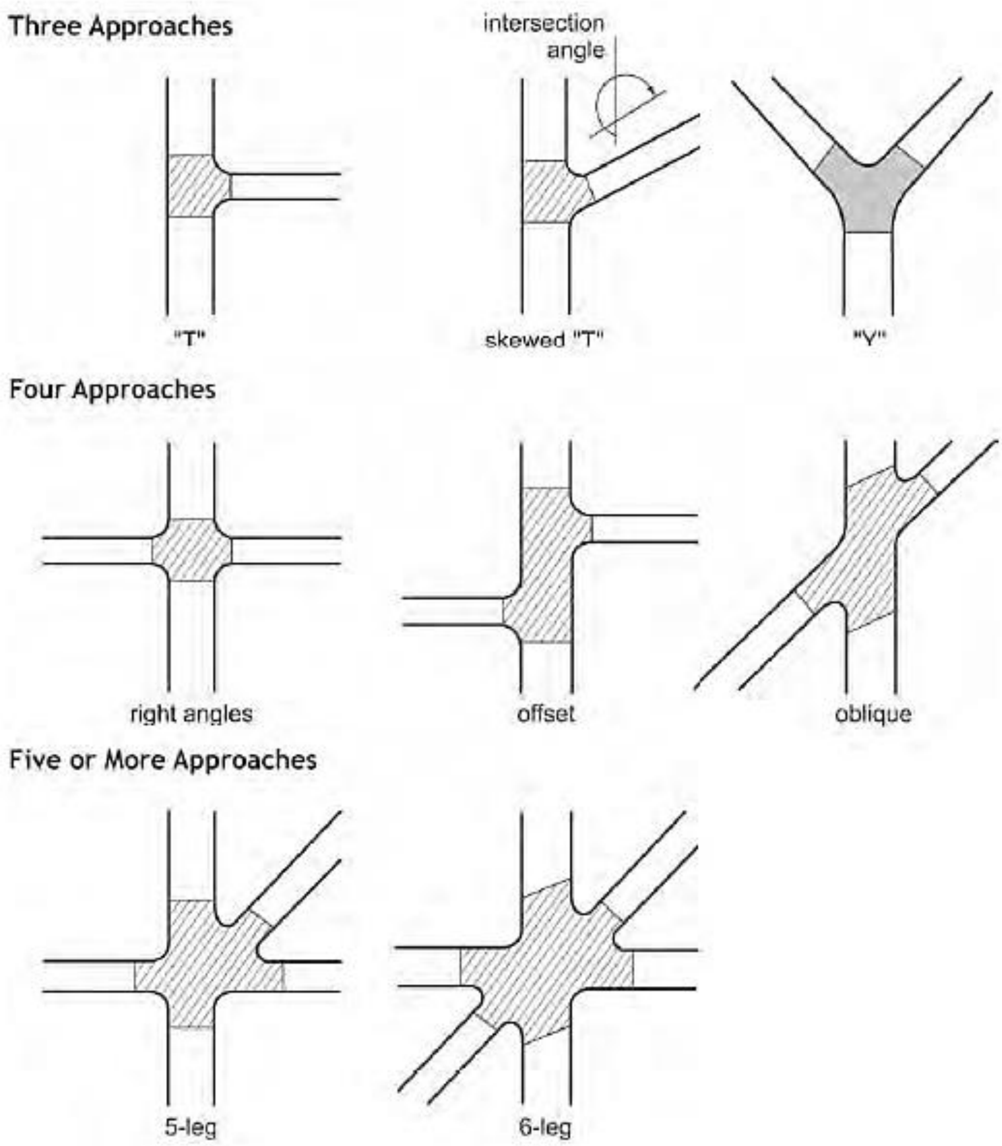


Figure 3.21. Intersecting street configuration [Source: in reference [18], p. 6-14]

In another theoretic study about intersections [19], talking about intersection spacing, it is mentioned that, when introducing a new intersection, the designer must ensure that there is sufficient distance between the new and the adjacent intersections so that they form distinct intersections. For example, if two intersections are close together, they must be considered as one for signal phasing purposes. Therefore, distance among intersections is an important parameter that might define networks in places with determined intersections.

4. Experimental part

Once the research has been done, the experimental part carried out by the author is going to be exposed. First of all, to define the parameters that are used to characterize a network is necessary. Then, the process to achieve the results (all MATLAB challenges and the solutions proposed) is explained. And finally, the results obtained are going to be analyzed and explained in order to verify that the objective of the project is achieved.

4.1. Useful parameters

The parameters used to analyse the maps are the key of the project. Choosing correctly the values that best fit to this project is not an easy work. The better chosen the parameters are, the more accurate results are going to be. Unfortunately, these are limited by the difficulty of the programming. Despite programming difficulties and among all the possible features to choose, the following four parameters are the selected:

1) Distance between two intersections connected by a link

The distance among the intersections (example in the Figure 4.1) might be a good characteristic to start the classification. The used graph (histogram: graph of the representation of frequency distributions, in which rectangles are used within coordinates) can give important information about the map in question. As we will see later, if the histogram has only one bar, this could indicate a square grid. If the histogram has two defined bars, it could be a rectangular grid. And finally, an irregular histogram might indicate other types of networks: random, scale-free, diamond, etc.

Figure 4.1. Distance between two intersections of Barcelona

2) Angle between streets that arrive to an intersection

To obtain all the angles of a map gives an idea of the network. Again, if the angles histogram has only one bar in 90˚ or two (one in 90˚ and another in 180˚) might indicate the network is a grid (square or rectangular) and, together with the distances obtained in the first parameter, we could classify it. Then, if the distribution is irregular, we should look into other parameters to decide.

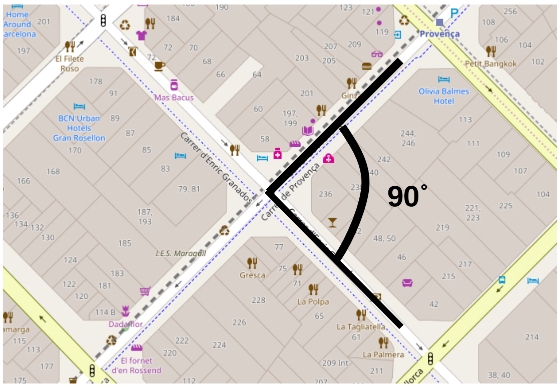


Figure 4.2. Angle between two streets in a intersection of Barcelona

3) Degree centrality

This is a parameter used in other network analysis seen in the literature research part. The distribution of number of streets that arrive and leave an intersection gives an idea of the network connectivity. Analysing the degree histogram, we can expect the kind of network we are studying.

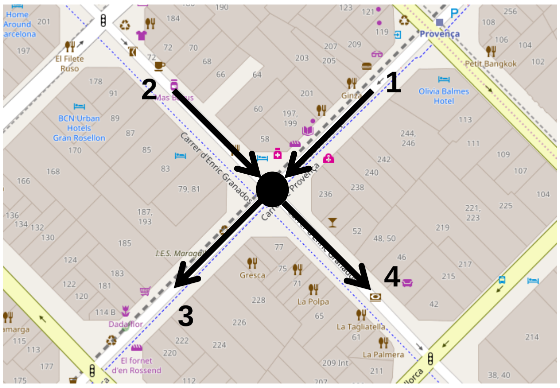


Figure 4.3. Degree of an intersection of Barcelona

4) Significant streets orientation

And finally, the streets orientation can be decisive to classify a grid (there are just two significant streets orientation) or in case there are more than two orientations, combined with the other three parameters, might be decisive to make a decision. This parameter is always included in a range of 0˚ to 180˚.

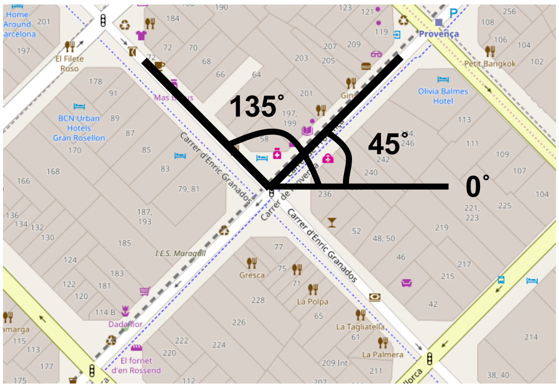


Figure 4.4. Orientations of the grid

Apart of these four parameters, to obtain other simple parameters is also possible, such as:

- number of nodes (intersections) in the map,
- number of links (streets),
- number of streets of one direction and two directions,
- cyclomatic number (seen in the equation *(3.18)* and if we consider $G = 1$, as we cannot obtain the number of sub-graphs in the map),
- the alpha index (seen in the equation *(3.19)* and also considering that $G = 1$, as it depends on the cyclomatic number),
- the beta index (seen in the equation *(3.20)(3.20)*),
- and the gamma index (seen in the equation *(3.21)(3.21)*).

4.2. Process

The experimental part of this thesis has been developed with MATLAB, which combines an improved desktop environment for iterative analysis and design processes with a programming language that expresses the mathematics of matrices and arrays directly. The toolboxes of MATLAB are professional developed, rigorous and fully documented [25].

The *process* holds all the MATLAB functions that include from the exported map (from OpenStreetMap) to get the results (graphics, parameters, etc.). This process has been divided in four parts. First of all, the part where the OSM map is converted to MATLAB useful language and is plotted to see the map. Right after, to fix the matrices and map obtained in the first step is necessary and depends on each map, i.e. not all the maps need to be *cleaned* (to be modified in order to have maps with all the streets plotted just once) the same way. Finally, the functions to obtain results and characteristics are programmed.

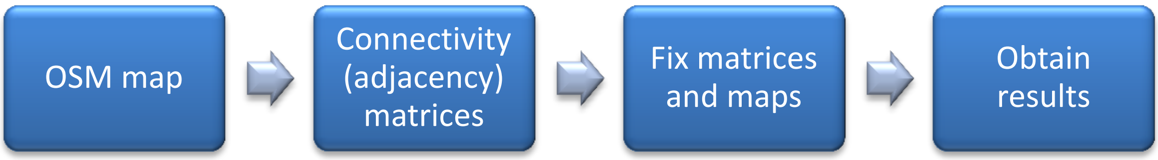


Figure 4.5. Process representation

4.2.1. Convert an OSM map to a connectivity matrix

The first part of the project is characterized by taking an OSM file from OpenStreetMap and converting it to a usable format for MATLAB (matrices). Then, a function to read the matrices and plot them is also necessary in order to see if the matrices are correct.

Due to the difficulty of this step, the three functions that are part of it are extracted from https://es.mathworks.com/matlabcentral/fileexchange/35819-openstreetmap-functions created by an expert in MATLAB programming [26]. From this source, only three functions have been used, although they call other functions, so the author recommends downloading the entire package. The three functions used are the following:

- First function: *parse_openstreetmap*

This function parses an OpenStreetMap XML file (OSM XML) downloaded from the *Export* option from openstreetmap.org.

It has as a unique input: the string of OpenStreetMap XML Data file name (the path where the map is saved, called *openstreetmap_filename*). Moreover, it calls two other functions (*load_osm_xml* and *parse_osm*) and has another one as dependency (*xml2struct*, renamed to *xml2struct_fex28518*). The outputs are two: a MATLAB data structure of XML of parsed OpenStreetMap file (called *parsed_osm*) and a MATLAB data structure of XML OpenStreetMap file (called *osm_xml*). Both are very similar.

The parsed_osm variable (see Figure 4.6) is a structure formed by four fields:

- Bounds: map limits, expressed as longitude and latitude,
- Node: formed by a vector with the nodes identification “id” and another vector “xy” with the coordinates longitude and latitude of each node,
- Way: created by a vector with the ways identification “id”, a structure “nd” with sets of nodes that form each way and the structure “tag” with sets of vectors with strings about the key and value of the way, although may not be available for all the ways, and
- Attributes: structure with five fields about the map (attribution, copyright, generator, license and version).

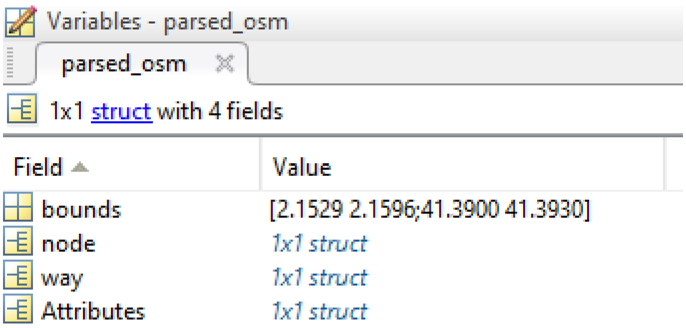


Figure 4.6. Parsed_osm variable from MATLAB

➢ Second function: *extract_connectivity*

This function extracts the connectivity of the road 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.

The unique input is *parsed_osm*, as returned by the first function (*parse_openstreetmap*). This function calls others: *assign_from_parsed* and *get_way_tag_key*. The outputs are two parameters: the connectivity matrix (called *connectivity_matrix*, the adjacency matrix of the directed graph of the transportation network, where $adj(i,j) = 1\ if\ a\ road\ leads\ from\ node\ i\ to\ j\ and\ 0\ otherwise$ as it is explained before in 3.2.5. Matrices of Graphs) and the parameter with the unique nodes of the intersections (called intersection_node_indices).

➢ Third function: *plot_road_network*

The third function taken from the package plots the nodes and the edges by connecting them. It needs three parameters: the axes object handle (called *ax* and normally used “gca”, which means current axes or chart), the *connectivity_matrix* (obtained in Second function: *extract_connectivity*) and the parsed_osm structure (obtained in First function: *parse_openstreetmap*). It returns the plot of the map.

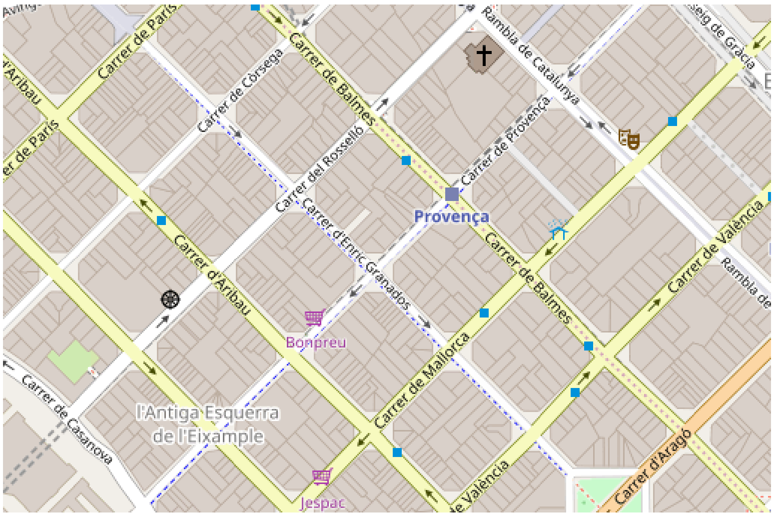


Figure 4.7. A little area of l’Eixample (Barcelona) from OpenStreetMap

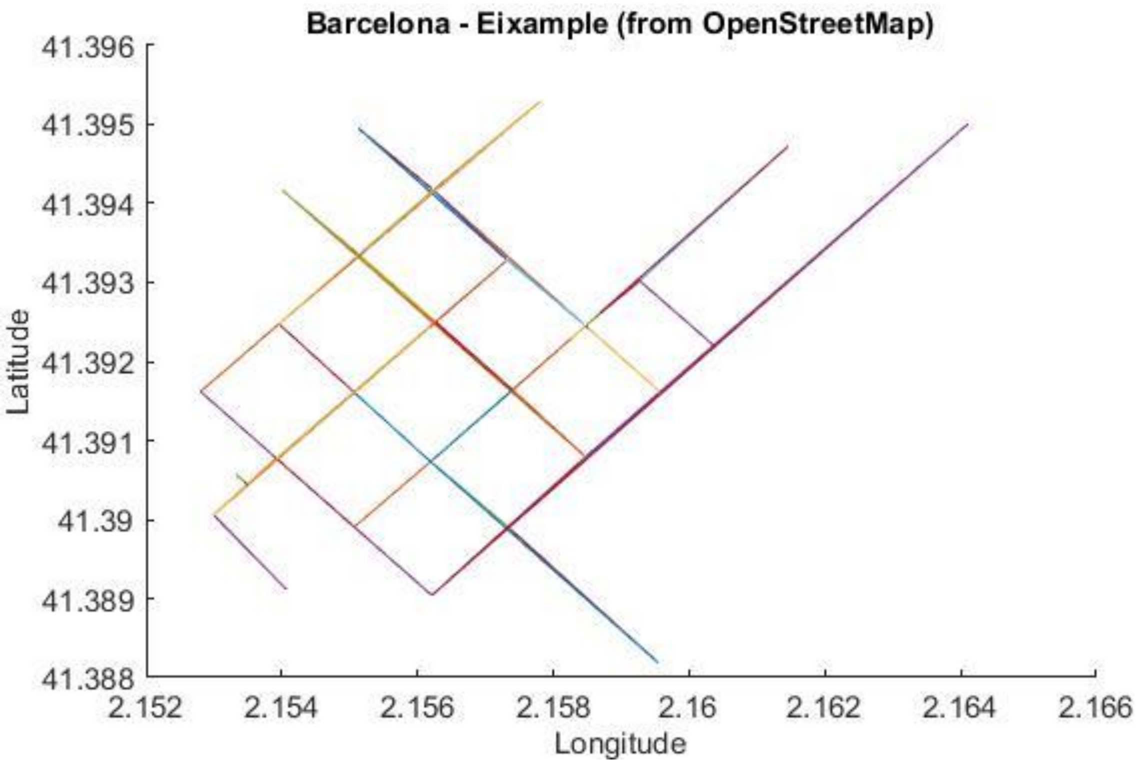


Figure 4.8. Map graph of l’Eixample after using the three functions

This useful function is used during the whole experimental part. It gives a global vision of how the map with the actual connectivity_matrix and parsed_osm is, and can be plotted with or without the nodes numeration (starting at 1 and until the number of nodes).

Let’s see how these functions work in a little example of l’Eixample (an area of Barcelona). In the Figure 4.7 we can see the part of map that has been exported as an OSM file and in the Figure 4.8 we can see how it is represented with the MATLAB functions. A perfect grid where only the roads are printed (with the right direction) is expected. As we can see in the Figure 4.9 (zooming in a part of the graph), the grid is not perfect.

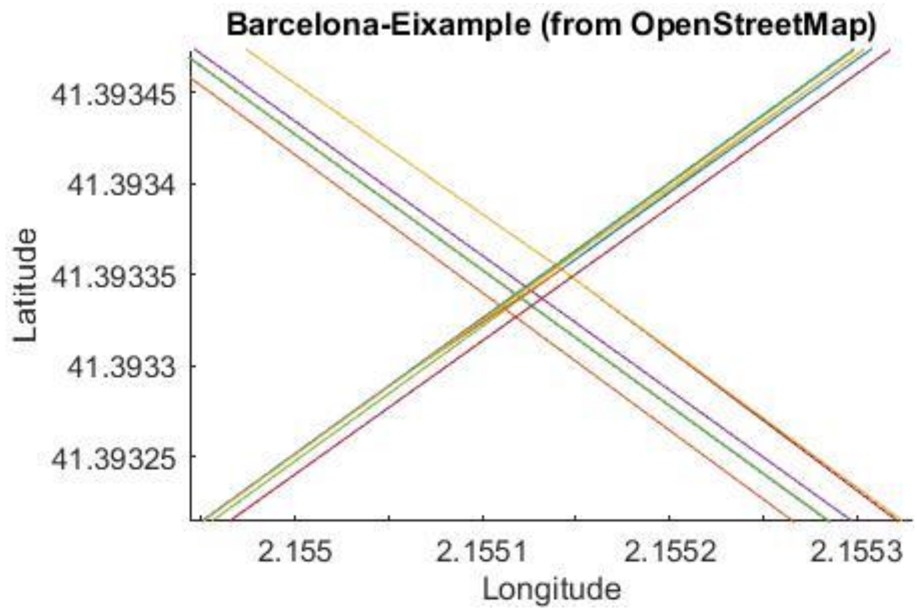


Figure 4.9. Intersection with multiple lines

Some links are connected multiple times because there is more than one node when it has to be just one, so more than one way exists when we only want one. This fact might happen because the map from OpenStreetMap exports the bicycle lane, the sidewalks, the bus lanes, the vehicles lanes, etc. or because the function is designed to join every pair of nodes that are connected (even there is one intersection in the middle that joins both).

Moreover, some links have a node in the middle that divides the link in two and should not be there, as we can see in the Figure 4.10. This fact might falsify the results about the intersections, so it has to be resolved.

These are the problems that are tried to be fixed with the functions explained in 4.2.3. Fix map.

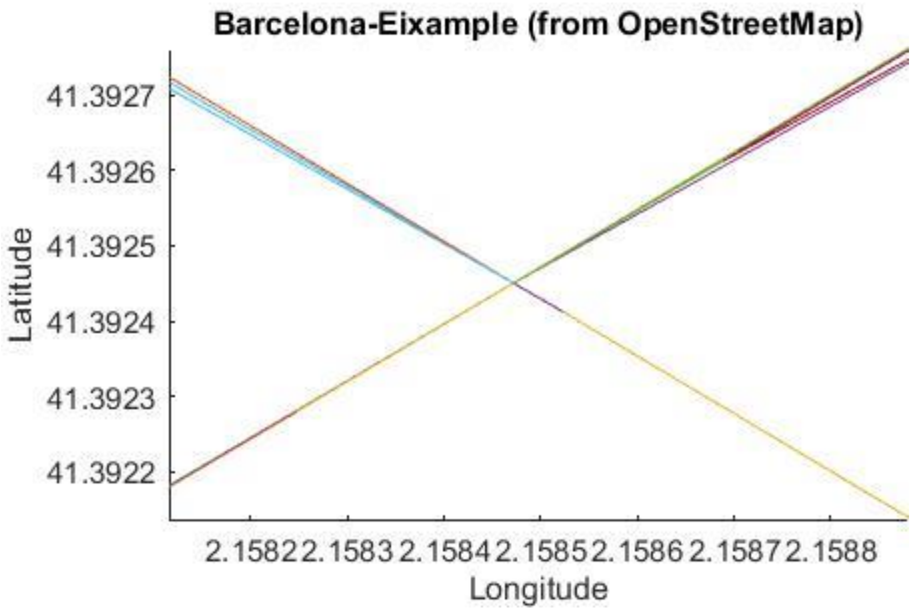


Figure 4.10. Intersection with links divided into two

4.2.2. Fix matrices

So far, the three functions explained (and the called ones) have been taken from the package [26], as mentioned before. From now on, most of the functions are programmed by the author of this thesis.

The matrices extracted from the previous functions are bigger than needed, i. e. they have all the nodes of the map, in spite of not being used in any link. For this reason, a function to reduce the useful matrices is programmed. Moreover, latitude and longitude are not a useful mode of coordinates for calculating distances and angles, so it is going to be changed.

- The *reduce_connectivity_matrix* function

This function is created to eliminate those nodes that are not used in the process. It means to delete the rows and/or the columns of the matrices to make them smaller. In order to do that, the intersection_node_indices parameter is applied: all those nodes that are not contained in this vector are deleted.

The inputs are the matrices to reduce: the connectivity matrix and the parsed_osm (the “node” field, where there are the node identification and the node coordinates). As only two parameters of this size can be inputs, the intersection_node_indices vector is calculated again. The outputs are the same parameters as the input, but modified (the nodes are renumbered). Since the links are not changed, the map plot is the same as in the Figure 4.8.

A representation of the MATLAB code function is summarized in the following box:

```
For each node i:
    If node i is in intersection_node_indices vector:
        Delete row "i" of connectivity matrix
        Delete column "i" of connectivity matrix
        Delete column "i" of parsed_osm.node.id
        Delete column "i" of parsed_osm.node.xy(all rows, i)
    End
End
```

Taking the same example of l'Eixample (Barcelona), the original connectivity matrix has a 1409×1409 dimension and, after applying this function, it becomes a 38×38 matrix (logically, the same as the vector intersection_node_indices length).

- The *convert_to_meters* function

Another problem to fix is that working with meters is preferred (as distance unit) to work with variation of latitude and longitude coordinates. To find a solution is needed.

Latitude and longitude are the units that represent the coordinates at geographic coordinate system [28]. Every single point on the surface of the Earth can be specified by the latitude and longitude coordinates.

The latitude shows the angle between the straight line in the certain point and the equatorial plane. It is specified by degrees, starting from 0˚ and ending up with 90˚ to both sides of the equator, making latitude Northern and Southern. The longitude is another angular coordinate defining the position of a point on a surface of the Earth. It is defined as an angle pointing west or east from the Greenwich Meridian, which is taken as the Prime Meridian. It can be defines maximum as 180˚ east from the Prime Meridian and 180˚ west from the Prime Meridian. We can see a drawing representation in the following figure.

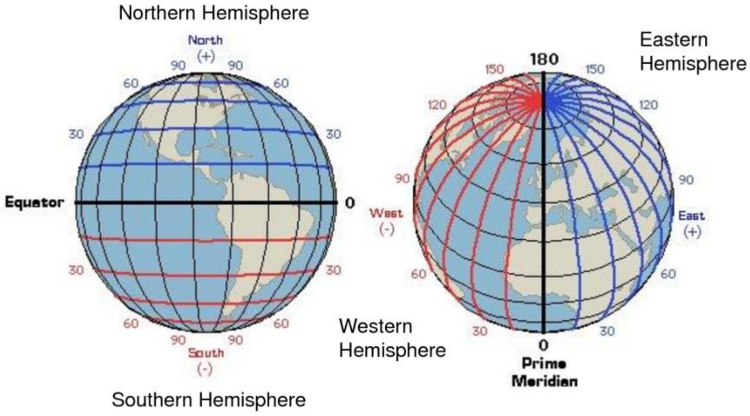


Figure 4.11. Latitude (left) and longitude (right) representation [Source: https://blog.eogn.com/2014/09/16/convert-an-address-to-latitude-and-longitude/]

The solution comes with the *haversine formula*, which determines the great-circle distance between two points on a sphere given their longitudes and latitudes [27]. For any two points on a sphere and solving the haversine of the central angle between them, the distance between them is:

$$d = 2r\ arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos\varphi_1 \cos\varphi_2 \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \qquad (4.1)$$

Where:

r	Radius of the Earth
φ_1, φ_2	Latitude of point 1 and latitude of point 2, in radians
λ_1, λ_2	Longitude of point 1 and longitude of point 2, in radians

The aim of this function is to point out all the nodes to a reference point. This means to name every node with a distance variation “x” and a distance variation “y” in meters respect to the reference point. This reference point in question, with new coordinates (0,0) is chosen as the center of mass (giving mass m “1” to all the points) to minimize distances and its definition is:

$$Center\ of\ mass = \frac{\sum m_i r_i}{M} \qquad (4.2)$$

Where:

m_i	Mass of the particle i
r_i	Coordinates of the particle i
M	Total mass of the system

But when we consider the same mass for each point, we are becoming the center of mass equation into the average equation. So, the reference point in that one with the average latitude and the average longitude of the map.

The problem of the haversine formula is that it gives the distance between any two points, but it does not give the distance variation “x” and the distance variation “y” (an angle is needed). Therefore, an alternative that uses geometric variables based on the haversine formula has been applied. The MATLAB code utilized is in the following link “http://pordlabs.ucsd.edu/matlab/coord.htm” although some code lines have been modified. The new map plot is as the one seen in the following figure.

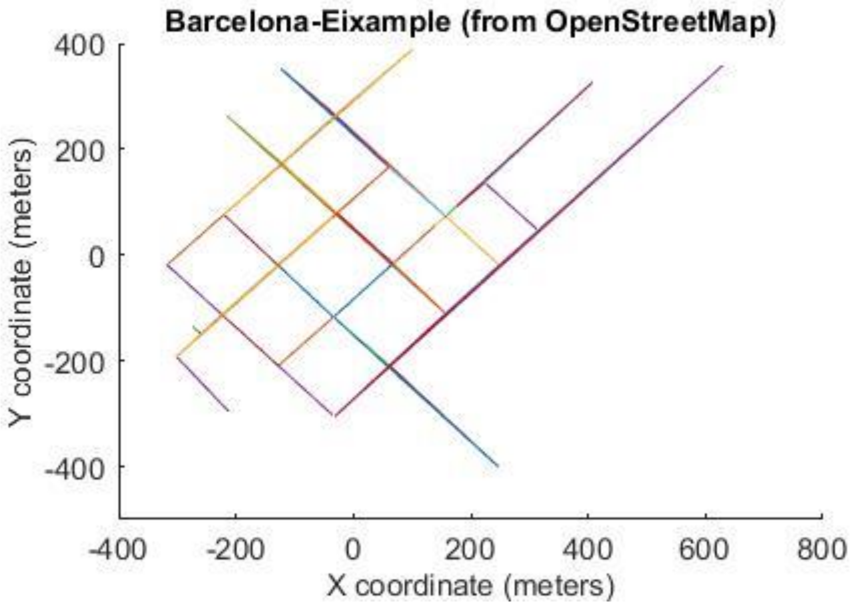


Figure 4.12. Map graph of l'Eixample after using "convert to meters" function

Let's prove that it is a reliable function. If we take a link close to the reference point (supposed to have the more accurate coordinates), which vertices are A (-128.67, -18.84) and B (-33.95, -117.26), we obtain these parameters:

$$analytical\ distance\ (from\ MATLAB\ coordinates) = 136{,}60\ m$$

$$theoretical\ distance\ (approx.\ from\ Google\ Maps) = 134{,}57\ m$$

With these two parameters, the relative variation is calculated. We suppose from the beginning that a good relative variation result is a value smaller than the 5%.

$$\Delta\ distance(\%) = \frac{136{,}60 - 134{,}57}{136{,}60} * 100 = 1{,}49\ \%$$

To have a 1,49 % of variation is a good result. Let's calculate the same variation for a further link respect to the reference point (worse results are expected), with vertices A (248.59, -18.93) and B (156.95, 73.79).

$$analytical\ distance\ (from\ MATLAB\ coordinates) = 130{,}36\ m$$

$$theoretical\ distance\ (approx.\ from\ Google\ Maps) = 132{,}94\ m$$

Then, again the relative variation is calculated:

$$\Delta\ distance(\%) = \frac{132{,}94 - 130{,}36}{132{,}94} * 100 = 1{,}94\ \%$$

In spite of this result is worse than the first one, it is less than a 5 %, so we consider the *convert_to_meters* function as a good approximation to convert the variation of latitude and longitude to variation of meters.

4.2.3. Fix map

In order to obtain good results in the parameters explained in the 4.1 section, to fix the map mistakes (as the two mistakes shown in the Figure 4.9 and Figure 4.10) is needed. It is logical to think not all the maps will have the same troubles; consequently, to adapt the functions to correct “on the go” is necessary. This part is not methodical and the person who uses the fix-map functions has to check every time that the function is working as it is wanted to, by plotting the map and/or by checking the matrices.

These functions are focused on deleting those nodes and ways which should not be there. Now, the problem is to know which nodes and ways are the ones that have to be eliminated.

- The *eliminate_nodes* function

The first try to fix the map is this function, *eliminate_nodes*, which eliminates the nodes that are close to a caught node. It replaces the connections that the removed node has with new connections with the node that stays.

In the following figure, a graphic representation of a possible situation is represented:

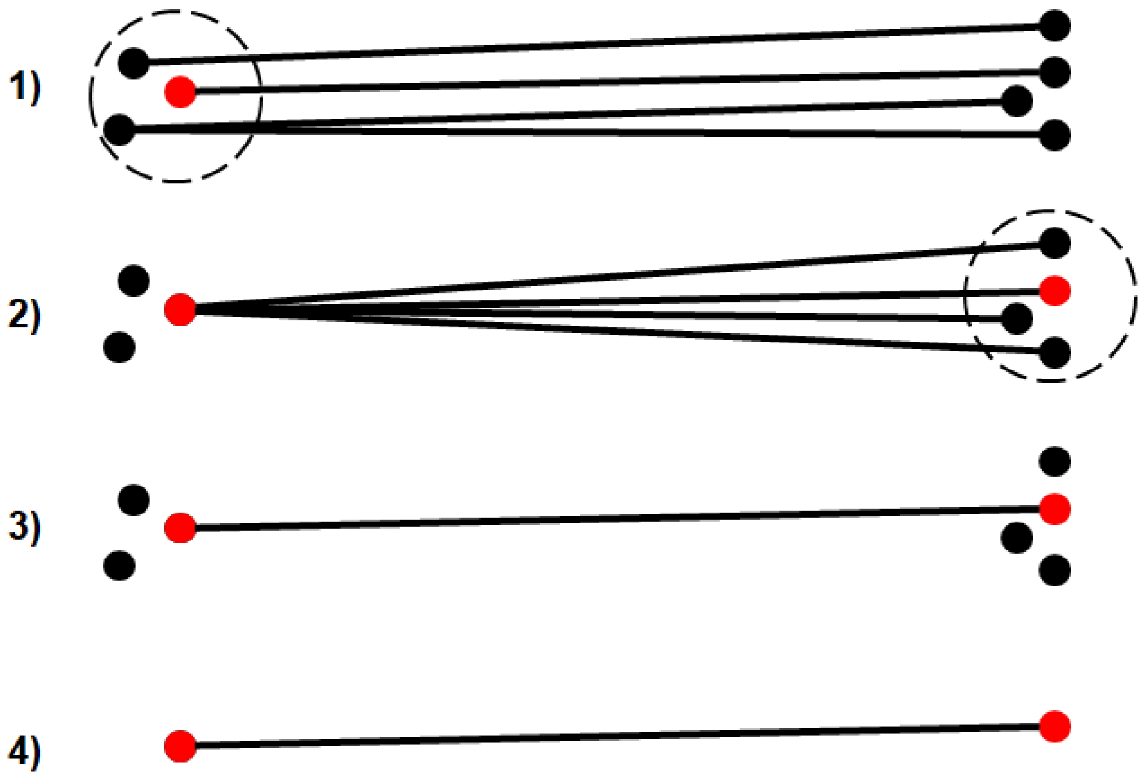


Figure 4.13. Steps representation of the “eliminate nodes” function

Given the connectivity matrix and the *parsed_osm* parameter, the first step is to detect a node (in order it comes in the connectivity matrix; the red node on the left in the example) and see which nodes (if exist) are close to it in an accurate distance (threshold) in order to consider them all as a unique node. The threshold depends on the map and finding the one that fits better is a challenge for the function user. This is not an automatic step.

The next step is to redirect the links of those nodes that are going to be deleted to the detected node. The third step is basically repeating the first step for all the nodes of the map. At the end (fourth step), the function “reduce connectivity matrix” is used to delete those nodes that have no links. The problem of this function comes when the superposed links do not have the same beginning and ending vertices.

A representation of the MATLAB code function is summarized in the following box:

```
For each node i:
    For each node j:
        If node i and node j are different nodes:
            Calculate distance between them
            If distance < threshold:
                Eliminate links of the node j
                Create links of node j into links of node i
            End
        End
    End
End
```

- The *eliminate_ways* function package (from 1 to 9)

The whole “eliminate ways” package is dedicated to delete all those ways (links) that are multiple and are unnecessary. To understand what every function does and to know if it is applicable to the map we have in every situation is necessary in order to optimize this process and have a perfect map as soon as possible.

- The *eliminate_ways1* function

This function does basically the same as the eliminate_nodes function, but catching a way instead of a node. Let's see it represented in the same example as seen in the Figure 4.13.

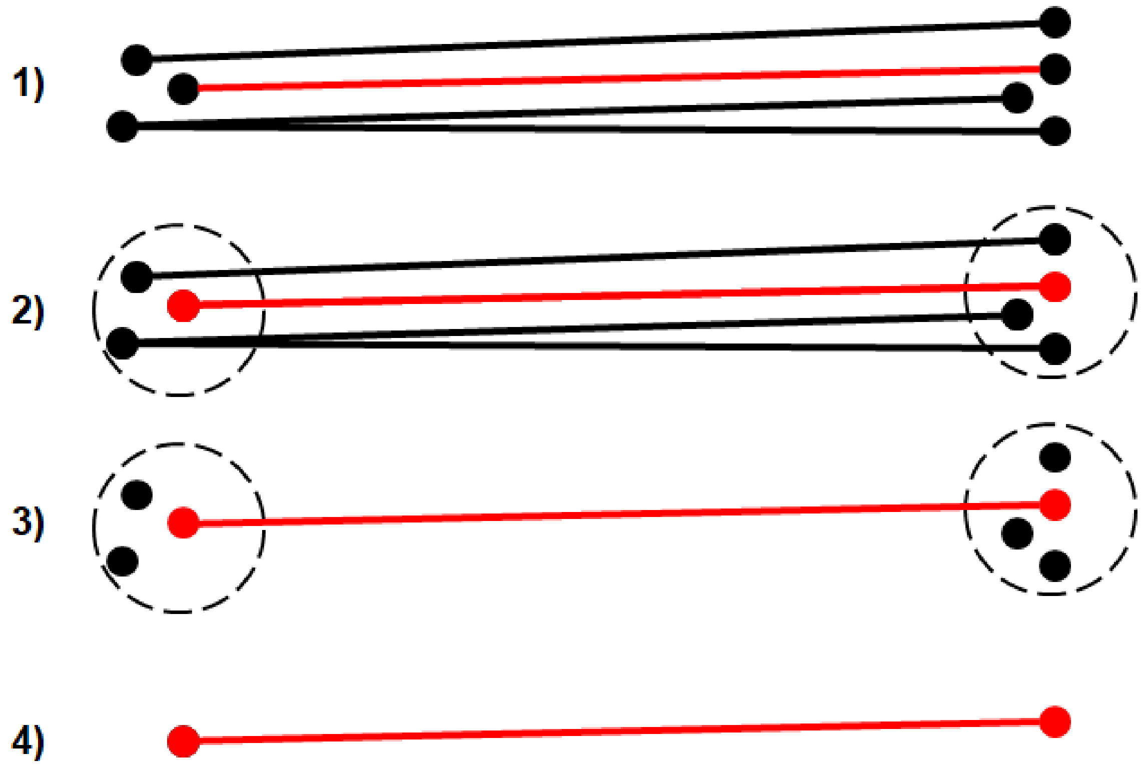


Figure 4.14. Steps representation of the "eliminate ways 1" function

Given the connectivity matrix and the *parsed_osm* parameter, the first step is to catch a link and see which links have beginning and ending vertices close enough (again the threshold parameter) to the vertices of the caught link. Then, the links that are not useful are eliminated by cancelling them in the connectivity matrix (to make "0" the pertinent element of the matrix that represents the link). Finally, here the "reduce connectivity matrix" function is also applied to accelerate the MATLAB processing time and to work with simpler matrices.

As commented before, the problem of this function is that only works in situations when the links start and end in similar points.

The MATLAB code used here is very similar to the "eliminate nodes" function, but here both vertices limits are compared at the same time.

- The *eliminate_ways2* function

This function is based on the inclinations of the links. When the inclination of the links is very similar and the links start at the same vertex means that the links are superposed and are unnecessary. In the following figure, a possible situation is visible and then how the function works is explained.

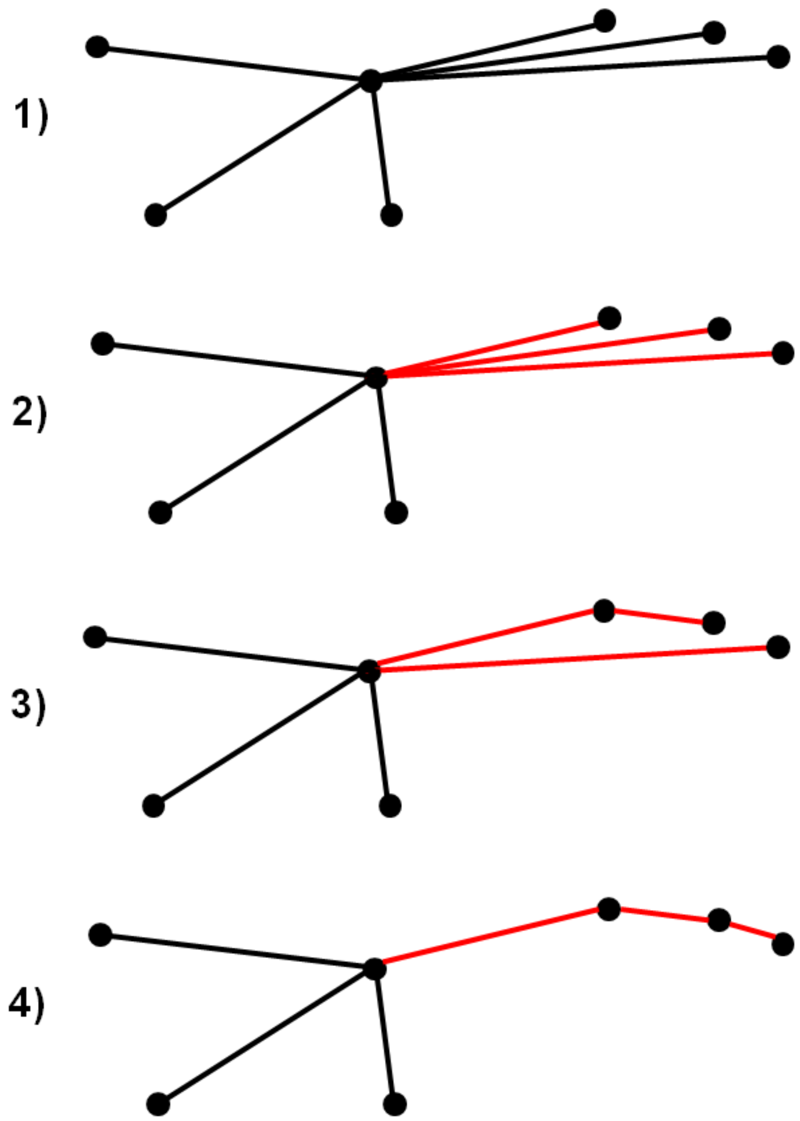


Figure 4.15. Steps representation of the "eliminate ways 2" function

The first step is the initial situation. When a node has more than one connection (always the connection can be in or out), the function calculates the inclination of all of them and saves the values in a vector. If the link is in and out (has both directions), is saved only once, for the good performance of the function; for that, it calls the *eliminate_repeated_elements* function.

After going through the first step, the next one is to detect those inclinations that are similar. The second step is based on the threshold value again. The vector before used is now reorganized with only those links with an inclination difference less than the threshold parameter imposed by the user, according to the map requirements. Moreover, the vector saves the links ordered by length (in ascending order).

The third and the fourth steps in the Figure 4.15 are the key of this function. As the ending vertices do not have to be close, they might have other respective links that need to be saved. In this way, the nodes are not lost and all have their correspondent links. This part uses the vector ordered by length and does what is represented in the third and fourth steps in the Figure 4.15. In the drawing, the inclinations are exaggerated, but in the reality the lines should be seen as straight lines. Anyway, to notice or not the lines *deformation* will depend on the threshold value used.

A representation of the MATLAB code function is summarized in the following box:

```
For each node i:
        Find links (in and out) & eliminate repeated elements
        For each link w:
                Calculate length and inclination
                New column vector C: limit vertices, inclination & length of link w
                Matrix Ways: previous Ways, adding vector C
        End
        For every two inclinations in matrix Ways:
                If (inclination1 – inclination2)<threshold & coord. restrictions
                        Matrix D: previous D, adding column of matrix Ways
                        referred to the second inclination
                End
        End
        Sort the matrix D by length, in ascending order
        Delete the correspondent links and create the new ones
End
```

The code might look complicated, but it is the unique way found to carry out this function after trying simpler options and learning about the mistakes.

- The *eliminate_ways3* function

This function is created to solve the problem when a link is crossed by a node and the link is not divided when it should be. This problem comes from the map extraction or as result of the application of other functions.

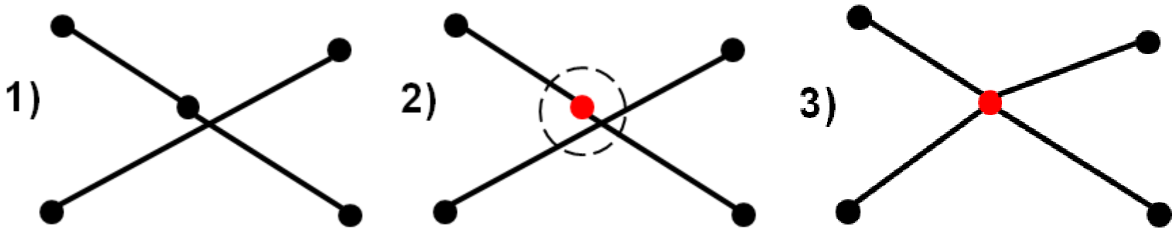


Figure 4.16. Steps representation of the "eliminate ways 3" function

After several attempts, the solution that best fits is based on the threshold concept and the equation on a line.

The first two steps shown in the *Figure 4.16* are detecting the situation and proving that is the indicated situation. First of all, detecting the situation means to take every combination of one node and one link and see if the distance between them is close enough (threshold parameter, which depends on the author). Considering the link as the explicit form of the equation of a line with respective coordinate's limits:

$$y - y_1 = m(x - x_1) \tag{4.3}$$

Where:

m	Inclination of the link
x_1, y_1	Coordinates of one point of the line
x, y	Coordinates of a point that fits in the line (red point, in this case)

Then, the function uses the next condition to know if the red point in step 2 is the adequate:

$$y - y_1 - m(x - x_1) < Threshold \tag{4.4}$$

If this condition is true, to prove that the coordinates x, y of the red node are comprised between (not equal) the x, y coordinates of the beginning and ending nodes of the link, respectively, is necessary. That means the same as proving the red vertex is contained inside the rectangle or the square that is formed by the limits of the link. Let's see this in the following image:

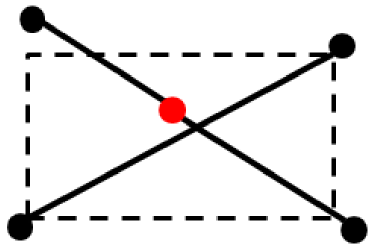


Figure 4.17. Required condition for the “eliminate ways 3” function

Once these conditions are true, the third step is implemented: to eliminate the original link and create two new links, through the node. As mentioned before, the threshold value will determine the accurate execution of this function.

A representation of the MATLAB code function is summarized in the following box:

For each node i:

    For each link z with limit nodes ≠ node i:

        Calculate inclination m of link z

        If $y_z - y_i - m(x_z - x_i) <$ threshold & coordinate restrictions:

            Delete link z

            Create two new links from limit nodes of link z to node i with the appropriate direction

        End

    End

End

- The *eliminate_ways4* function

Basically, this function does the opposite to the previous function. A link is sometimes divided into two when it is not expected (as seen in Figure 4.10 and graphically visible in the next figure).

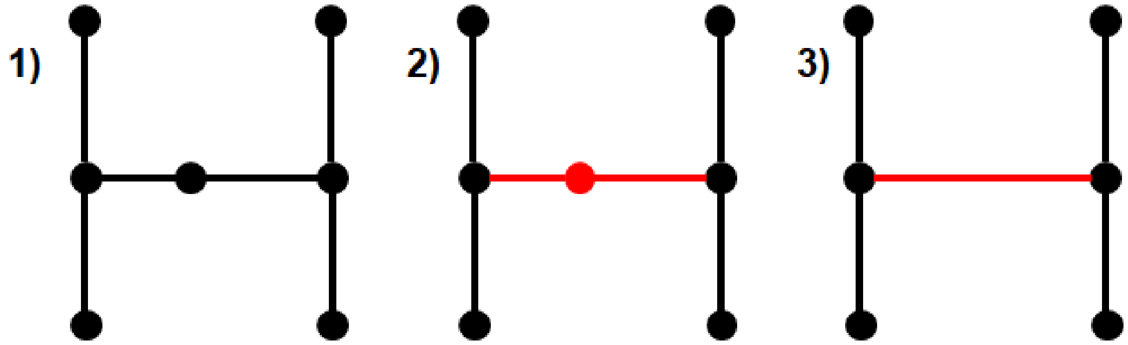


Figure 4.18. Steps representation of the "eliminate ways 4" function

As always, the first thing to do is to detect the situation. In this case, the situation is a node with two links (they can be four links if we think of both directions of the links, but it becomes two when the *eliminate_repeated_elements* functions is applied). These two links must have similar inclination (in this function, it is checked by the standard deviation of the set of inclinations, so that it is less than a threshold value) but they must be in different quadrants too (checked by the x-coordinates).

Once the situation is detected, the connectivity matrix is changed by becoming the short links into one (the directions must be respected). Then, the *reduce_connectivity_matrix* function is applied.

A representation of the MATLAB code function is summarized in the following box:

```
For each node i:
        Find links (in and out) & eliminate repeated elements
        For each link w of node i:
                Calculate inclination
                New vector C: previous C, adding inclination of link w
        End
        If standard deviation of C < Threshold & X-coordinates restrictions
                Change connectivity matrix, respecting the link directions
        End
End
```

- The *eliminate_ways5* function

Once the main problems have been solved, other little detected problems can be figured out. In several maps, it is common to see a set of links that cross each other, as we can see in the Figure 4.19, which it is a zoomed part of the center of Vienna. This fact might be because of the presence of zones such as parks, churches, lakes, rivers, etc.

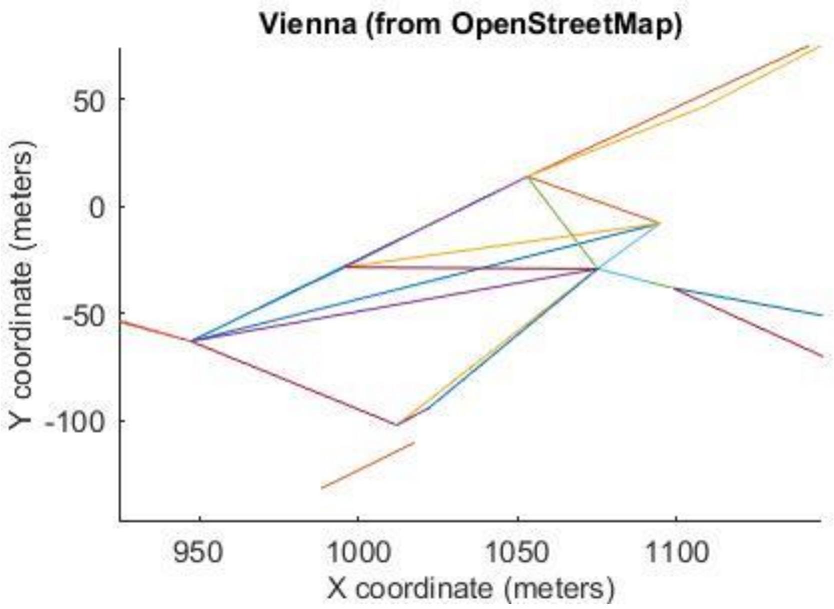


Figure 4.19. Set of links that cross each other

This function works by eliminating the set of links that cross each other. Consequently, if we think there is a link that doesn't have to be deleted, we should not run this function. It is based on finding if two lines intersect in a point inside the coordinate restrictions between these two lines.

A representation of the MATLAB code function is summarized in the following box:

```
For each link w:
    Calculate limit vertices coordinates & inclination of the link w
    For each link z ≠ link w
        Calculate limit vertices coordinates & inclination of the link z
        Calculate intersection point (x, y)
        If x ∈ X coordinates restrictions & y ∈ Y coordinates restrictions
            Delete both links
        End
    End
End
```

- The *eliminate_ways6* and *eliminate_ways7* functions

Several times, the links are extremely long, especially those in the boundaries of the map. Therefore, these two functions have been created. On the one hand, the sixth function eliminates the links whose length is greater than X meters (value to choose by the user). On the other hand, the seventh function eliminates the links whose length is X times greater than the average length of all links.

The *eliminate_ways6* function has no mystery, it is very simple. It compares, for each link, its length with the X length given. If this condition is true, the link is deleted. We must ensure we delete those links we want to.

However, the *eliminate_ways7* function is a little bit more complicated. It has to ensure that the average length does not count those links that have both directions as two links, so the first step is to occult one direction in this calculation. Then, the average length is calculated after having all links length in a vector. Finally, we can compare the “X times” average length to each link length (previously calculated in the average calculation).

- The *eliminate_ways8* function

This function is created to delete those links that have no other connections on the vertices, neither in the beginning, nor at the end (subgraph of only one link), as we can see on the bottom of the Figure 4.19. This may happen when possible links have not been taken into account in the extraction (first functions problems). It is sometimes useful in order to no alter the characteristics results.

- The *eliminate_ways9* function

And finally, this last function to eliminate ways (links) works for deleting all the links that have a coordinate greater or smaller than one given. Sometimes we want to delete a whole part of the map, so we can apply this function. The programming code is not very complicated. It compares every vertex with the given x or y coordinates.

After applying the proper functions in the example seen before of l’Eixample, an area of Barcelona, the map result is shown in the Figure 4.20. The result is a square grid with only a line between two intersections. Having a look at the graph, we can’t know if a street has one direction or both. We should examine the connectivity matrix and see if the (i, j) and (j, i) elements are both 1. After all, this map is now in the optimal conditions for being extracted the characteristics we need for the project.

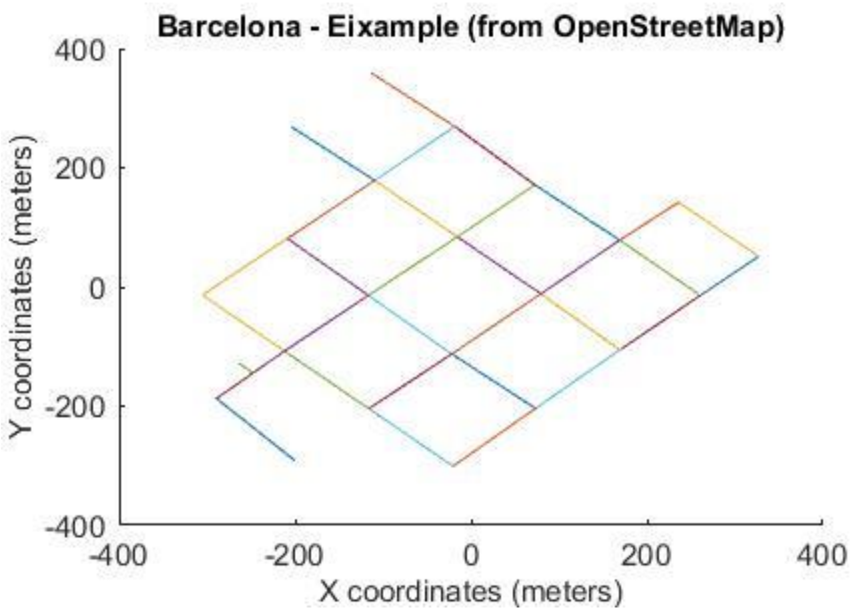


Figure 4.20. Map graph of l'Eixample after using the appropriate functions

4.2.4. Functions to get parameters

When the matrices and the map are ready to be examined, it's time to apply the functions designed to get parameters, which have been explained before, in 4.1.Useful parameters.

- The *one_two_ways* function

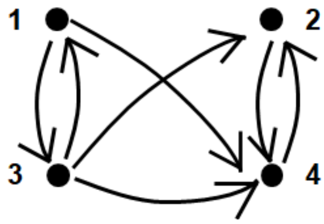
First of all, this function works in order to know how many streets (links) are just "one way" and how many are bidirectional. As mentioned before, these are parameters that can't be seen in the map graph. Given the connectivity matrix, it calculates the parameters without necessity of programming with *for*, *if*, *while*, etc. structures. The general equations are the following for a given connectivity matrix D of $n \times n$ is:

$$E = abs(D - D^T) \tag{4.5}$$

$$Number\ of\ one\ way\ streets = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} E}{2} \tag{4.6}$$

$$Number\ of\ two\ way\ streets = \frac{\left(\sum_{i=1}^{n}\sum_{j=1}^{n} D\right) - Number\ of\ one\ way\ streets}{2} \tag{4.7}$$

Let's see how these equations work with an example:



$$D = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Figure 4.21.Example of a directed graph

For an example like this, with four nodes and a connectivity matrix D of 4×4 dimensions ($n = 4$), where we can see there are 3 one-way links and 2 two-way links, we calculate them analytically:

$$E = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

$$Number\ of\ one\ way\ streets = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} E}{2} = \frac{6}{2} = 3$$

$$Number\ of\ two\ way\ streets = way2 = \frac{(\sum_{i=1}^{n}\sum_{j=1}^{n} D) - 3}{2} = \frac{7-3}{2} = 2$$

These parameters can be used in statistics and for having an overlook of the map.

- The *distance* function

The *distance* function calculates one of the most important parameters: the link lengths plotted in a histogram, among other characteristics. Remember what a histogram is: graph of the representation of frequency distributions, in which rectangles are used within coordinates.

The first step is not to consider a link as two streets for not falsifying the results. So, as it is seen before, the connectivity matrix is changed (in another variable) to make it all one-way and to get the link lengths. Then, the distance between intersections is saved in a vector and the average and the standard deviation are calculated. The histogram is plotted with the more appropriate bar width and limits. For the example of Barcelona (Eixample), the histogram is as seen in the Figure 4.22.

After this, to calculate the average distance among parallel streets (useful for grids) and to find the significant orientations has been considered interesting. For that, the matrix *slopes* is created and it is designed as a set of columns with the following structure:

$$C = \begin{pmatrix} node\ 1 \\ node\ 2 \\ slope \\ dist \\ counter \end{pmatrix} \tag{4.8}$$

Where *node 1* is the beginning vertex of the link, *node 2* is the ending vertex of the link, *slope* is the inclination of the link, *dist* is the link length accumulation and *counter* is a counter of links. It works the following way:

- If the *slopes* matrix is empty, the column (4.8) of the first link is added, with counter equal to 1.
- If the *slopes* matrix is not empty (it has at least one column), the *slope* of the current link is compared to the slopes of the links that are already in the matrix (all third row) by calculating the difference between the current slope and the slope i in the matrix. Now there are two options:
    - If the difference is less than a threshold parameter so that the links have similar inclination, what means that the two links are parallel or are in the same line, the *dist* parameter is now the previous value plus the link length of the current link and the counter is increased in one.
    - If the difference is greater than a threshold parameter so that the links have different inclination, the *slopes* matrix is increased in one column with the vector C (4.8) of the current link.

A representation of this MATLAB code function part is summarized in the following box:

```
slopes = empty matrix
For each link w:
    Calculate inclination and length and C vector
    If slopes is empty
        Slopes = vector C, with counter = 1
    Else if number of columns of slopes > 0
        For each column i in slopes
            If (inclination i – current inclination) < Threshold
                slopes(4,i) = slopes(4,i)+dist
                slopes(5,i) = slopes(5,i)+1
            End
        End
    End
End
```

At the end of this function part, four parameters can be extracted:

1) Number of orientations in the map. This parameter is the number of columns in the *slopes* matrix, and how many streets form each orientation is the counter of each column. In order to see the most important orientations or the significant orientations and to avoid possible threshold mistakes, the orientations with less than a 5% of the total are eliminated. This is the number of significant orientations parameter and is also plotted in a histogram.

Then, a set of parameters can be calculated if the number of significant orientations is two (when a grid is expected).

2) When the number of significant orientations is two, with the *slopes* matrix, to know the average of the distance among parallel streets is possible. The accumulate distance (fourth row) divided by the accumulate number of streets (fifth row) of one orientation is the average distance among parallel streets of the <u>other</u> orientation.
3) Although the angle between the two orientations is expected to be 90˚, it is also calculated (angle between two inclinations m_1 and m_2) as:

$$\alpha = \tan^{-1}\left|\frac{m_2 - m_1}{1 + m_1 m_2}\right| \tag{4.9}$$

4) And finally, the orientation of the grid (minimum angle of the orientations respect to the 0˚ or X axe) is also computed.

In addition, the percentage of one-way streets with the total, using the one_two_ways function, is figured out.

To sum up, the example used during the process explanation (Barcelona) is shown. The information displayed in this function and the graph it plots are the following (in the Results part are going to be commented):

"Found 2 significant directions in this zone.
And the minimum angle among streets is 88.077°.
The grid is oriented + 44.5028° East.
The distance among parallel streets are 126.3079 meters and 118.7696 meters.
The average distance among intersections is 122.653 meters.
The percentage of one-way streets with the total is 100%.
The standard deviation of the distance among streets is 27.7332 meters."

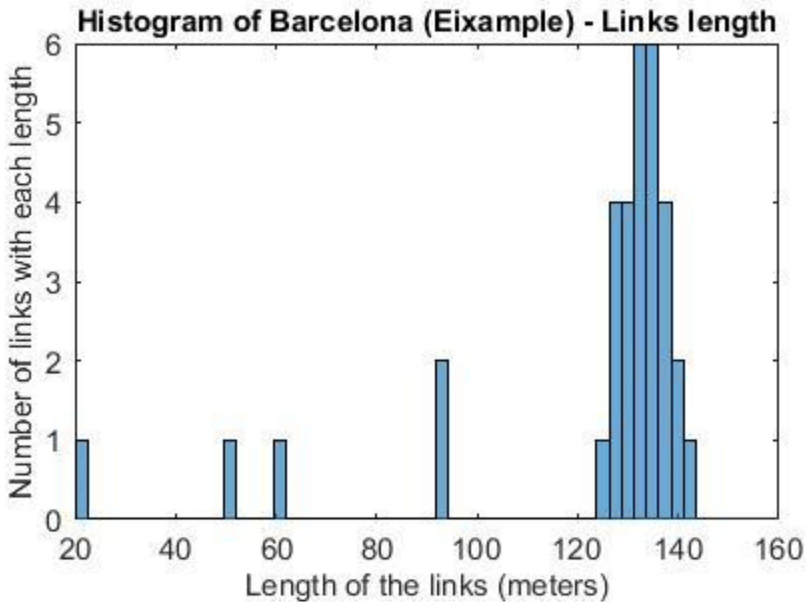


Figure 4.22. Histogram of distances between intersections of Barcelona

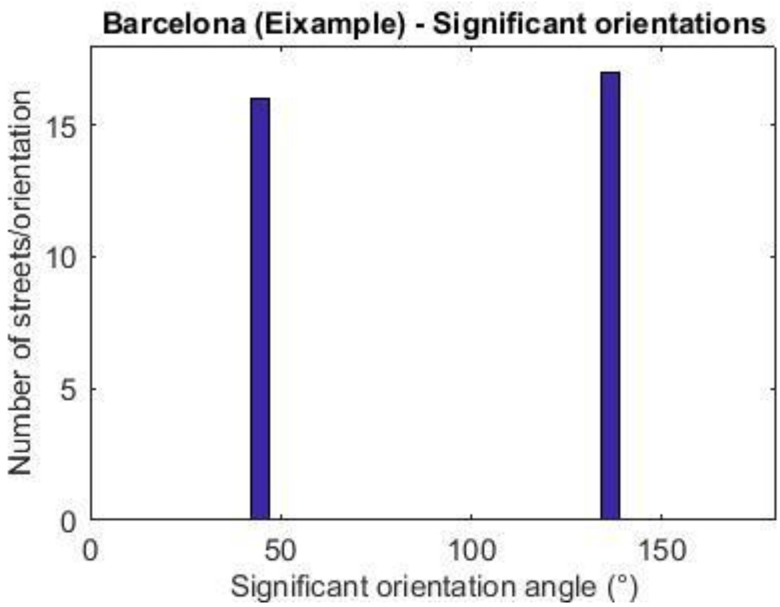


Figure 4.23. Histogram of significant orientations of Barcelona

- The *angles* function

The main objective of this function is to plot in a histogram the angles formed by two streets when arrive to an intersection. Taking advantage of this, other parameters are also calculated: average and standard deviation of the list of angles and a histogram of the number of streets that arrive to an intersection (degree centrality) and average.

The strategic process to carry out this function is the following: for every intersection, find the links; for each of these links, calculate the angle respect to 0˚; then, sort the angles (0-360˚)

since they can be disordered; and finally, calculate the angle difference of two consecutive links. A representation of this MATLAB code function part is summarized in the following box:

```
For each node (intersection) i:
        Find links in and out, and eliminate repeated elements
        For each link
                Calculate angle respect to 0° at the height of the node i
                Save this angle into the angles vector
        End
        Sort the angles (0-360°)
        For each pair of consecutive angles
                Calculate the difference (1) and save it in angles_abs vector
        End
End
```

(1) This step is done as follows. If the pair of angles does not contain the last angle in the vector, the difference is calculated as the second angle minus the first angle. However, when the difference is about the last angle of the vector and the first one, it is calculated this way:

$$\beta = first\ angle + 360^{\circ} - last\ angle \tag{4.10}$$

In the meantime, the degree centrality is being extracted for each node and saved in a vector in order to plot then the histogram. At the end, what the function displays in the *Command Window* and the plots are the next, about the example:

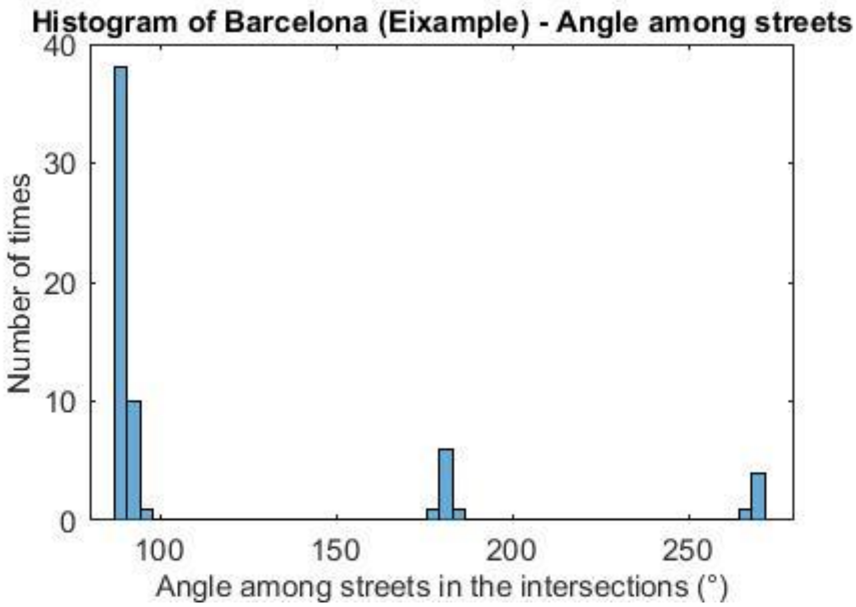


Figure 4.24. Histogram of angles among streets of Barcelona

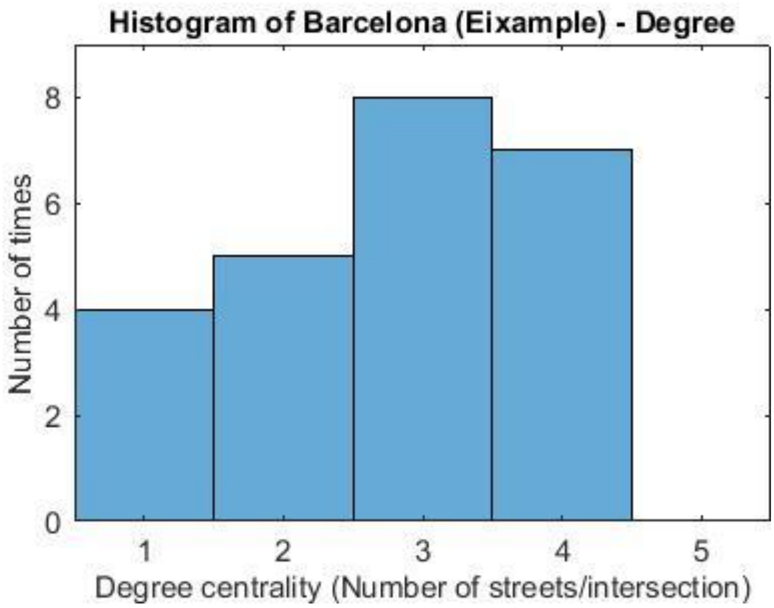


Figure 4.25. Histogram of degree centrality of Barcelona

"The average angle among streets is 116.129°.
The standard deviation of the angle among streets is 54.7305°.
An average degree centrality is 2.75 streets/intersection."

- The *characteristics* function

This last function is designed to get the characteristics explained at the end of the Useful parameters section. The extracted parameters here are complements of the functions explained before, i.e. they are not necessary to classify a network but give more information. As already mentioned, the number of subgraph G is always considered one. What the function returns when it is run, is the following:

"Number of nodes: 24
Number of links: 33
Cyclomatic number: 10
Alpha index: 0.23256
Beta index: 1.375
Gamma index: 0.5"

4.3. Results

The aim of this part is to expose and analyze the information (matrices, graphs and written function information) obtained when the suitable *process* functions are applied in every studied map.

At first, some relevant results are exposed. Since the quantity of graphs and images is considerable, the whole part of results is attached in Appendix.

Then, basically, an analysis of the information will be done in order to, in a future, classify a network with only this information, i.e. try to extrapolate the knowledge of the examples here used to apply it when a random traffic network is given.

Classifying a map by taking a look at it is relatively easy in some cases (if the map is, for example, clearly a grid). However, sometimes, it is a difficult task and several graphs and parameters are needed in order to guess the class of network.

The limited time of the project execution has allowed studying four cities of the world. First of all, the functions have been tested with a simple map as it is l'Eixample, Barcelona; then, with other cities such as Vienna, and finally, with other more complicated maps (mix of grids or mix of grid with old town areas). To sum up, the areas which have been analyzed are the following:

- Barcelona: Eixample area, Barri Gòtic area and a mix of Eixample and Barri Gòtic
- Vienna: inner city
- Brooklyn: one grid and two grids with different orientations
- Hoorn (Holland): outskirts of the city

4.3.1. Presentation of results

Here are exposed the more relevant results. To get started, the graphs are presented. Some of them are correlative and need to go together to understand how the network is. Afterwards, the characteristics extracted from the MATLAB Command Window are summarized in tables.

Although they are presented separately (graphs and characteristics), it is important to remember they might depend on each other. As commented before, the whole studies (graphs and map zones) and how the results are extracted "step by step" are in the Appendix.

The graphs are presented in trios: for each studied map, lengths (left) and angles (middle) and degree (right).

4.3.1.1. Graphs

A.1.

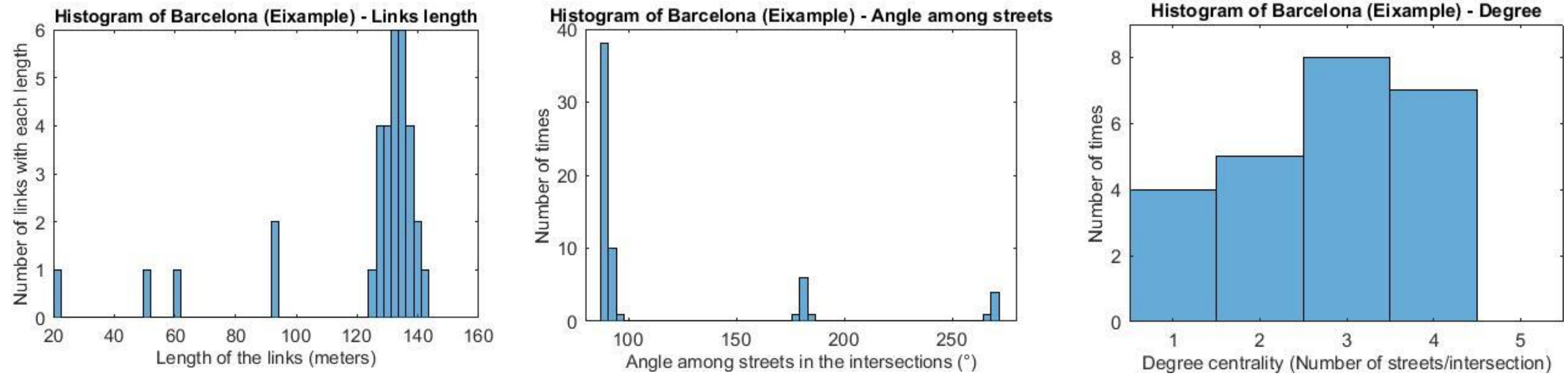


Figure 4.26. Eixample – Results

A.2.

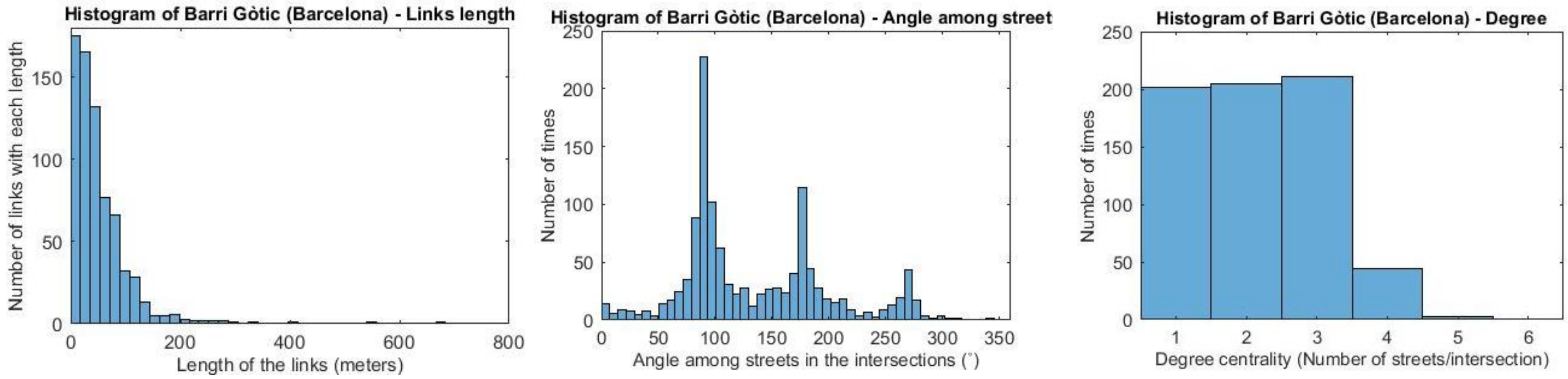


Figure 4.27. Barri Gòtic – Results

A.3.

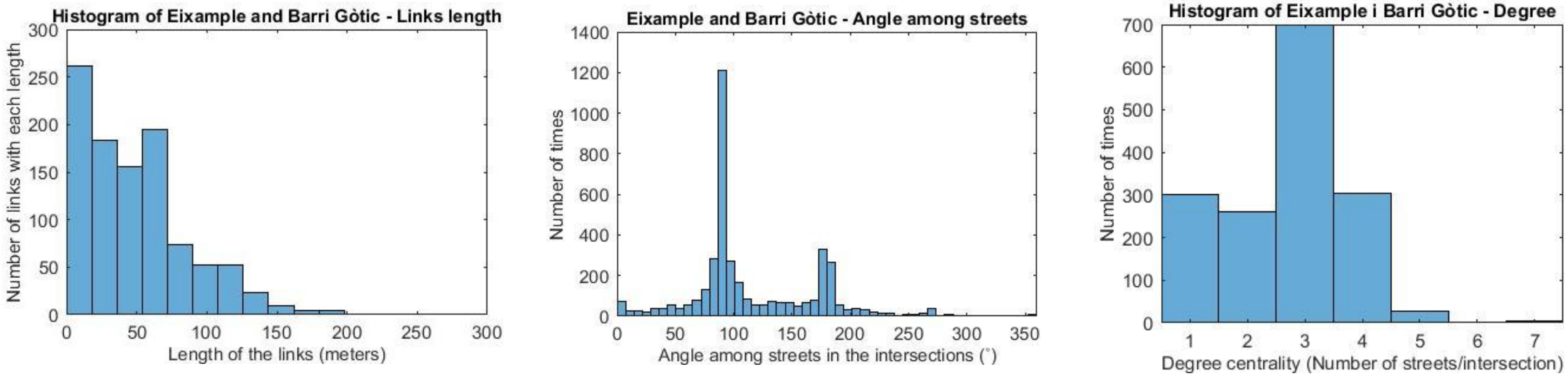


Figure 4.28. Barcelona mix – Results

A.4.

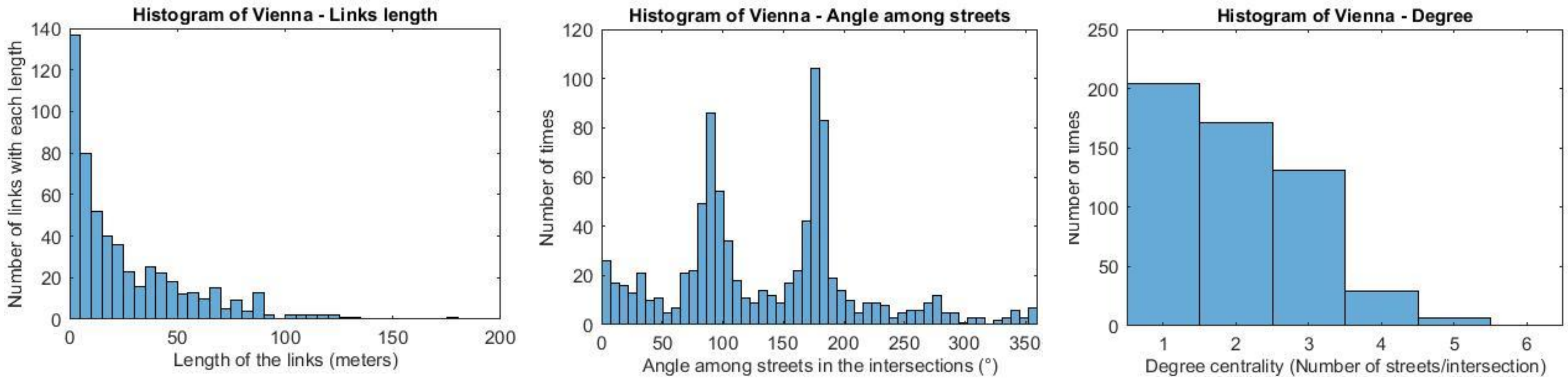


Figure 4.29. Vienna – Results

A.5.

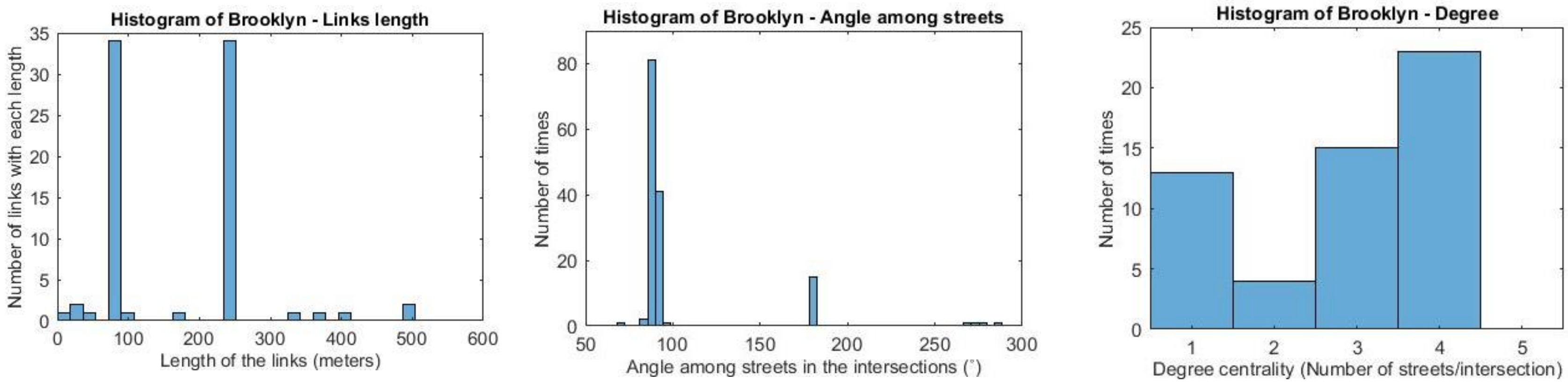


Figure 4.30. Brooklyn (1 grid) – Results

A.6.

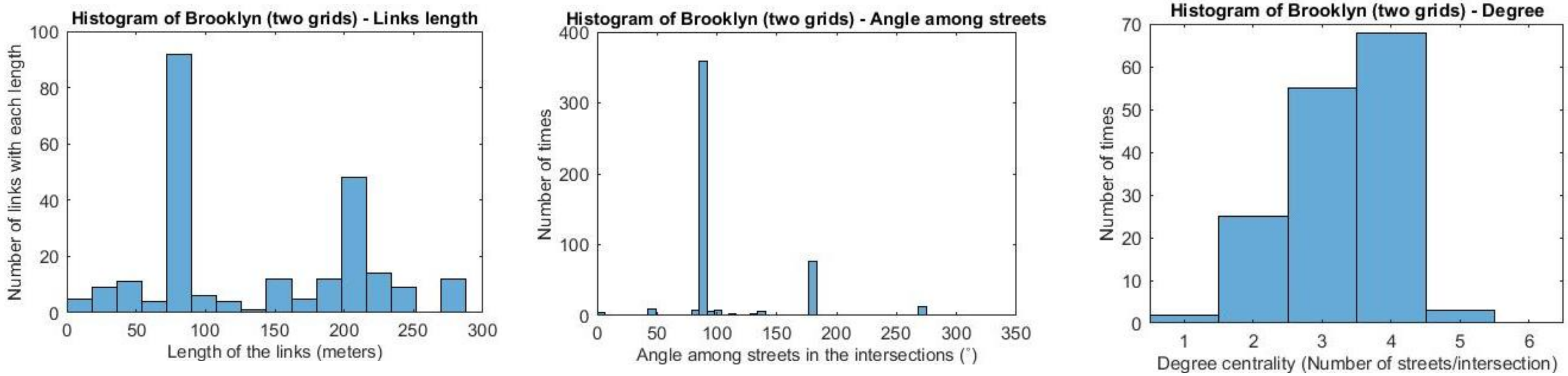


Figure 4.31. Brooklyn (2 grids) – Results

A.7.

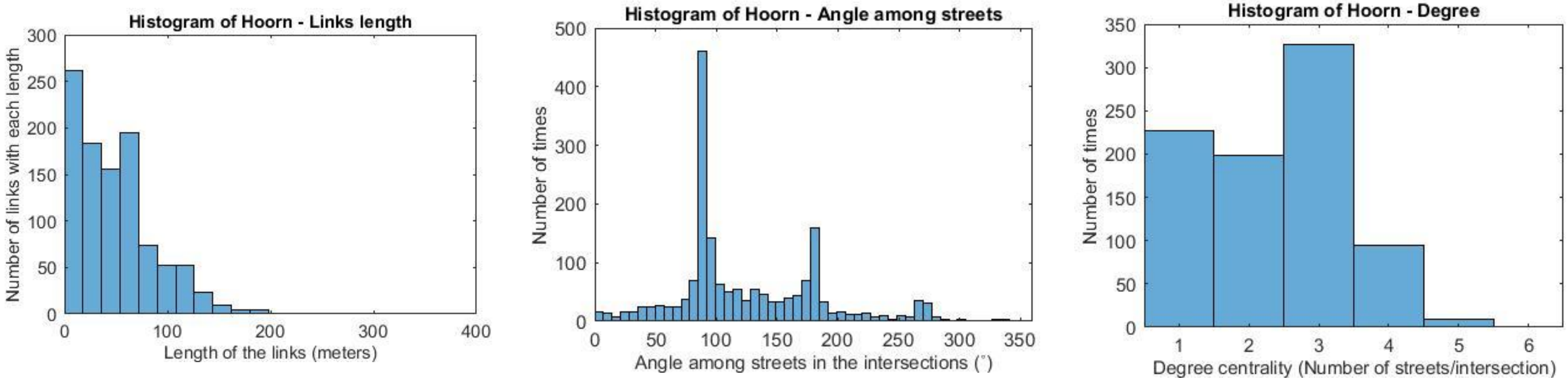


Figure 4.32. Hoorn – Results

4.3.1.2. Tables

Green cells: minimum values; red cells: maximum values.

Parameters\City	Eixample (Barcelona)	Barri Gòtic (Barcelona)	Eixample and Barri Gòtic (Barcelona)	Brooklyn (one grid)	Brooklyn (two grids)	Vienna	Hoorn
Number of significant orientations	2	7	6	2	4	7	6
Minimum/Maximum angle among streets	88,08° / 91,92°	-	-	89,97° / 90,03°	-	-	-
Grid orientation	+ 44,50° East	-	-	+ 8,59° East	-	-	-
Average angle among streets	116,13°	134,71°	116,36°	104,28°	108,29°	137,34°	125,12°
St. dev. of angle among streets	54,73°	65,44°	55,13°	40,16°	45,42°	75,63°	60,63°
Distance among parallel streets in grid	1) 126,30 m 2) 118,77 m	-	-	1) 247,36 m 2) 114,15 m	-	-	-
Average distance among intersections	122,65 m	53,73 m	49,60 m	174,33 m	143,37 m	26,27 m	49,60 m
St. dev. of distance among intersections	27,73 m	63,38 m	39,46 m	107,07 m	84,21 m	27,68 m	39,46 m

Table 4.1. Information extracted from MATLAB after running the distance and the angles functions

Parameters\City	Eixample (Barcelona)	Barri Gòtic (Barcelona)	Eixample and Barri Gòtic (Barcelona)	Brooklyn (one grid)	Brooklyn (two grids)	Vienna	Hoorn
% one-way streets	100 %	81,00 %	93,73 %	98,73 %	99,20 %	99,82 %	93,73 %
Average degree centrality (streets/intersection)	2,75	2,16	2,70	2,87	3,29	2,01	2,38

Table 4.2. Information extracted from MATLAB

Parameters\City	Eixample (Barcelona)	Barri Gòtic (Barcelona)	Eixample and Barri Gòtic (Barcelona)	Brooklyn (one grid)	Brooklyn (two grids)	Vienna	Hoorn
Approximately map area	0,25 km^2	1,8 km^2	7,02 km^2	0,64 km^2	1,92 km^2	0,7 km^2	2,28 km^2
Number of nodes	24	666	1725	55	153	542	858
Number of links	33	858	2264	80	254	546	1085
Cyclomatic number	10	193	540	26	102	5	228
Alpha index	0,23	0,15	0,16	0,25	0,34	0,0046	0,13
Beta index	1,38	1,29	1,31	1,45	1,60	1,01	1,26
Gamma index	0,50	0,43	0,44	0,50	0,56	0,34	0,42

Table 4.3. Information extracted from MATLAB after running the characteristics function

4.3.2. Analysis of results

In spite of the studied maps do not cover all types of maps we have seen in Figure 3.18 (17 networks topologies Zhang et al. used in a search), they can be reference object for future results and analysis.

First of all, when we observe the “links length” graphs (on the left in each study), three behaviours are observed. However, this may also depend on the bars width.

The first behaviour is when the graph seems a normal distribution (such as in *Figure 4.26. Eixample – Results*, on the left). This might mean, if the graph has a very pronounced peak in the distribution, the whole map is formed by links with similar length and, due to the accumulated variation in the functions, it appears as a normal distribution. For a not very pronounced peak in the normal distribution, it could simply mean that the map is dominated by links with a mean length and has other links (in less quantity) with greater or smaller length than the mean length, as seen in Figure 4.33.

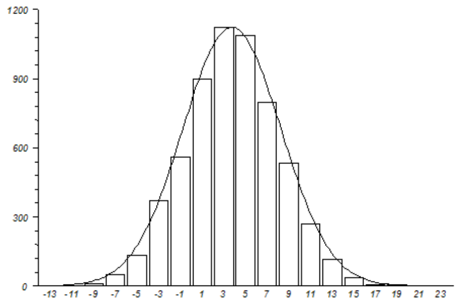


Figure 4.33. Normal distribution [Source:http://www.statsdirect.com/ help/distributions/normal.htm]

The second behaviour found is the map that has defined peaks like if they were impulses (as it is shown in *Figure 4.30. Brooklyn (1 grid) – Results*, on the left). That means the map is dominated by the links with the *peak* length.

And the third one is when the graph has a great number of small links and few ones for longer links (the graph is decreasing, as it is seen on the left in *Figure 4.27. Barri Gòtic – Results* and *Figure 4.29. Vienna – Results*, both on the left). This is normally related to non-regular maps: random, scale-free and small-world networks.

Continuing with next type of graph, angles among streets that arrive to an intersection (graph in the middle in the presentation of results), we can see in all of them that there are peaks in 90˚ and in 180˚. The right angle predominates in almost all the cases. The 180˚ angle can appear in the map limits and, when the map has few nodes, can predominate in the *angle* graph. Nevertheless, two kinds of graphs can be distinguished.

On the one hand, there is the type of map with the mentioned peaks (90˚ and 180˚) but almost without other bars (without other angles), as shown in the *Figure 4.30. Brooklyn (1 grid) – Results*, in the middle. These graphs are matched with those graphs with a *link length* with a normal distribution or with defined peaks, both explained before. Everything indicates these maps must be grids. That one that has only one peak in the length graph must be a square grid. If it has two peaks in the length graph must be a rectangular grid. However, this information should be confirmed with other parameters.

On the other hand, there is the possibility of the type of map with the 90˚ and 180˚ peaks but also with other bars (with other angles), as it can be seen in the graph in the middle in *Figure 4.27. Barri Gòtic – Results* and *Figure 4.29. Vienna – Results*. This indicates, firstly, it is clearly not a grid. Then it is observable that this happens when the lengths graphs are decreasing, the third behaviour commented before about links length graphs.

And to finish with graphs, the third graph can give extra information to corroborate the previous hypothesis. It can be seen, on the one hand, that the graphs that match with the supposed grids have peaks in the degree 3 and 4. This is logical if we think in 4 streets per intersection in the middle of the grids and 3 streets per intersection in the limits of the map (for big maps, the degree 3 should not be very significant in grids). On the other hand, the graphs that have high bars in 1, 2 or 3 degree match with those maps with irregular angle graph and with decreasing length graphs. This corroborates that they could be random, scale-free and small-world networks.

Moreover, the parameters extracted in the tables can give extra information. For example, looking at the table 4.1, the grids have a lot of clear information that corroborate they are grids: two orientations (the 4 orientations should be looked carefully in order to know if they are 2 combined grids); an angle between them of approximately 90˚; and the distance among parallel streets, in the case of Barcelona, the distances are not very similar (relative variation of nearly 6%) although a square grid should be considered if we think these values come from an average of all the links with the other orientation, whereas in the case of Brooklyn, the distances are significantly different and it is clearly a rectangular grid. However, the average angle among streets is quite higher than 90˚ (and the standard deviation of this value is also high; this fact is due to the boundaries of the map, where most of the angles are 180˚ and they are significant because the map is small.

Having a look at the table 4.1 and table 4.2, we observe that the old town areas have the minimum average distance among intersections (such as Vienna and Barri Gòtic) while modern cities have this parameter higher. The same fact happens with the average degree centrality: higher in grid cities and smaller in old towns. Instead, with the average angle among streets, is backwards: higher in old towns and smaller in grid cities. The percentage of one-way streets with the total is not a significant parameter to classify networks, but we can maintain it for curiosity.

The table 4.3 contains information about the connectivity measures. The first three parameters (map area, number of links and number of nodes) do not have a significant role in classifying networks. The map area values have been manually calculated from the MATLAB map with meter coordinates, they are not displayed in the *characteristics* function.

Let's remember the meaning of the other four parameters. The cyclomatic number is the number of fundamental circuits in the network. The alpha index is the ratio of number of cycles to possible maximum number of cycles. The beta index is the ratio between number of links and number of nodes, equivalent to average degree. The gamma index is the ratio of number of links to maximum possible number of links.

The alpha, beta and gamma indices (connectivity measures) coincide in the order of the cities, i.e. from maximum to minimum the order is: Brooklyn (two grids), Brooklyn (one grid), Eixample, Eixample and Barri Gòtic, Barri Gòtic, Hoorn and Vienna.

As Zhang et al. discussed in their investigations, the studied 17 network topologies might also be categorized by type of connections: group 1 (highly connected) – grid, matching pair, complete grid and diamond networks; group 2 (centrally connected) – hub-and-spoke, double tree, ring, diverging tails and crossing paths networks; group 3 (circuit-like connected) – central ring, double U and converging tails; group 4 (randomly connected) – random, scale-free and small-world networks.

Brooklyn (two grids), Brooklyn (one grid) and Eixample, as they are grids or combination of grids, have the higher levels of connectivity. The mixed map of Eixample and Barri Gòtic (see A.3. Barcelona – Eixample and Barri Gòtic), as it is grid in a 60% of the area and old town in the remaining 40%, it must be in the group 2 or 3. In mixed maps, the connectivity degree depends on how all the parts influence on the measures; moreover, the graphs of this mixed map is a vertical sum of the individual graphs (graphs in Figure 4.26 + graphs in Figure 4.27 ≈ graphs in Figure 4.28) clearly dominated by the Barri Gòtic as it has more links. Finally, the maps with smaller alpha, beta and gamma values (Barri Gòtic, Hoorn and Vienna) are included in group 4. Hoorn has a curious map and it is not an old town, but has some features like them (small links and all types of angles) but a set of intersections with 3 streets.

To sum up, grids have determined links length (normal distribution or peaks on the links length graph), angles among streets of 90˚ (and sometimes also of 180˚ in the boundaries), and should have a majority of 4 streets/intersection (maybe 3 and 4 if the map is small). Furthermore, basing these theories in the results here obtained, they have higher average distance among streets, higher average degree and smaller average angle among streets than the other maps, and are highly connected. Instead, old towns as Vienna or Barri Gòtic of Barcelona have little links length (decreasing graph), all type of angles among streets and abundance intersections with degrees of 1, 2 or 3. Additionally, they have smaller average distance among streets, smaller average degree and higher average angle among streets, and are randomly connected.

Conclusions

So far, a deep research about map data, graph theory, social networks, traffic network topologies and intersections has been done, and then, a method to classify traffic networks has been applied: the tool OpenStreetMap to download maps, a set of functions to adapt the maps to the requirements and an analysis of results.

This thesis is not a closed project. Due to all the fields it reaches, it is open to be spread out. First of all, the reader should consider that the literature research is large and most probably it will be actualized with new discoveries and theories over the years.

Secondly, the used maps are the base of the project, so they must come from a reliable source. The user is free to choose the source he wants while it is reliable and it is in OSM XML format. Anyway, it is advisable not to use very large maps in order to the MATLAB functions work in optimal conditions. In addition, in order to have more results to compare, more cities, village or any area of the world all welcome to this project.

Then, the MATLAB functions have been tried to develop the best way possible and are totally open to improvements. The author, before of carrying out this project, had basic knowledge in MATLAB, so programming most of the functions has been a challenge. Consequently, all improvements are welcome to ameliorate the code for decreasing the process time that means, in other terms, to reduce project expenses. Moreover, some maps might need functions that have not been developed in this thesis, so, if required, the author encourage anyone to update, change or create new functions.

And finally, although a great number of network parameters are analysed in this project, they can be extended in order to obtain more accurate results and be able to do a better classification of the traffic networks. The actual results must be used as a reference in future studies and show a way to observe the graphs and the parameters. At times, it is more interesting and more productive to analyse a set of maps than do it only for one map.

In a global vision, the project objective (to classify cities in traffic network topologies) has been satisfactorily reached through achieving the steps that lead to it, commented in this section. Moreover, to carry out this project has been a great experience for the author in order to extend the knowledge in all the fields the project includes, improve the programming skills and improve in data analysis.

Bibliography

Bibliographic references

[1] RUOHONEN, KEIJO (2013) *Graph theory*

[2] X. ZHANG, E. MILLER-HOOKS AND K. DENNY (2015) *Assessing the role of network topology in transportation network resilience. Journal of Transport Geography*, 46, p. 35-45

[3] WEISSTEIN, ERIC. *Königsberg Bridge Problem.* From MathWorld--A Wolfram Web Resource. [http://mathworld.wolfram.com/KoenigsbergBridgeProblem.html, 13.08.2017]

[4] PAOLETTI, TEO. *Leonard Euler's Solution to the Konigsberg Bridge Problem* [https://www.maa.org/press/periodicals/convergence/leonard-eulers-solution-to-the-konigsberg-bridge-problem, 13.08.2017]

[5] VAN DOOREN, PAUL (2009) *Graph Theory and Applications.* Course notes of UCL (Université catholique de Louvain)

[6] L. DA F. COSTA, F.A. RODRIGUES, G. TRAVIESO AND P.R. VILLAS BOAS (2008) *Characterization of Complex Networks: A Survey of measurements*

[7] Social Network Analysis: Theory and Applications

[8] BRILLIANT ORG. *Dijkstra's Shortest Path Algorithm* [https://brilliant.org/wiki/dijkstras-short-path-finder/, 15.08.2017]

[9] ROSETTA CODE ORG. *Vogel's approximation problem* [https://rosettacode.org/wiki/Vogel%27s_approximation_method, 15.08.2017]

[10] UNIVERSITAT POLITÈCNICA DE CATALUNYA. MECHANICAL ENGINEERING DEPARTMENT (2016) Graphs

[11] UNIVERSITAT POLITÈCNICA DE CATALUNYA. MECHANICAL ENGINEERING DEPARTMENT (2016) Goods transportation

[12] KYOUNGJIN PARK AND ALPER YILMAZ (2010) *A social network analysis approach to analyse road networks*

[13] ISLAM EL-ADAWAY ET AL. (2016) *Social Network Analysis Approach for Improved*

Transportation Planning

[14] S. FREIRIA, B. RIBEIRO AND A. O. TAVARES (2015) *Understanding road network dynamics: Link-based topological patterns. Journal of Transport Geography*, 46, pages 55-66

[15] B. JANIS VITINS (2014) *Shape grammars for urban network design.* A thesis submitted to attain the degree of Doctor of Sciences of ETH Zurich.

[16] MATTHIAS KOWALD AND KAY W. AXHAUSEN. *Transport Planning and social network analysis – An introduction*

[17] PARTHASARATHI, P. (2014) *Network structure and metropolitan mobility. The Journal of Transport and Land Use*, vol. 7, no. 2, p. 153-170

[18] MASS HIGHWAY. *Chapter 6 – Intersections.* 2006 Edition

[19] BUREAU OF DESIGN AND ENVIRONMENTAL MANUAL (2010). *Chapter Thirty-six Intersections.* Illinois

[20] LIBRARY OF CONGRESS – GEOGRAPHY AND MAPS. *Digital Data and Geographic Information Systems* [http://www.loc.gov/rr/geogmap/guide/gmilldma.html, 26.08.2017]

[21] *Comparison of web map services* [https://en.wikipedia.org/wiki/Comparison_of_web_map_services, 26.08.2017]

[22] *Web Map Service (WMS)* [https://www.techopedia.com/definition/876/web-map-service-wms, 26.08.2017]

[23] *OpenStreetMap* [https://en.wikipedia.org/wiki/OpenStreetMap, 26.08.2017]

[24] OpenStreetMap website [https://www.openstreetmap.org, 26.08.2017]

[25] MATLAB from MathsWorks [https://es.mathworks.com/products/matlab.html?s_tid=hp_products_matlab, 29.08.2017]

[26] IOANNIS FILIPPIDIS (2016) *OpenStreetMaps Functions* [https://es.mathworks.com/matlabcentral/fileexchange/35819-openstreetmap-functions, 29.08.2017]

[27] *Haversine formula*, from Wikipedia [https://en.wikipedia.org/wiki/Haversine_formula, 31.08.2017]

[28] *Get Latitude and Longitude* [http://www.latlong.net/, 31.08.2017]

Complementary bibliography

L. H. IMMERS AND S. LOGGHE (2002) *Traffic Flow Theory*. Lecture notes of *Katholieke Universiteit Leuven*, Department of civil engineering, Section traffic and infrastructure.

A. Appendix

This appendix contains the graphs and results that come from the MATLAB functions in the order they are being displayed. For digital copies: the source links of the first figure in each part redirect to the city in OpenStreetMap.

A.1. Barcelona – Eixample

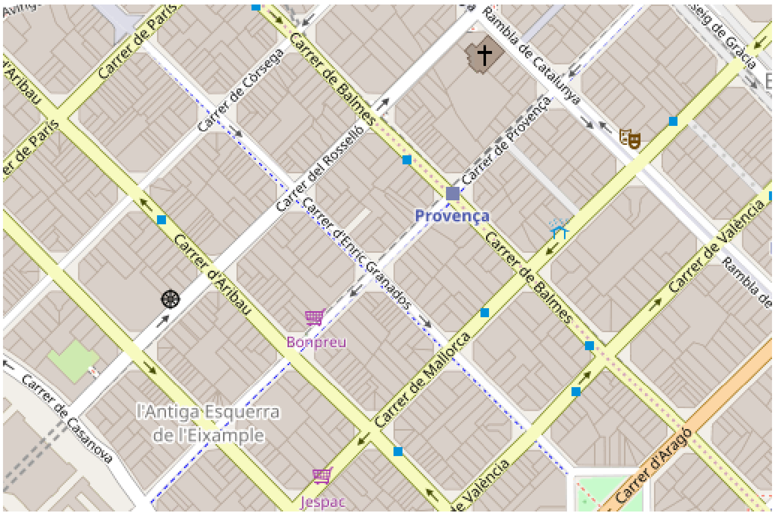


Figure A.1. Capture of l'Eixample (Barcelona) [Source: openstreetmap.org]

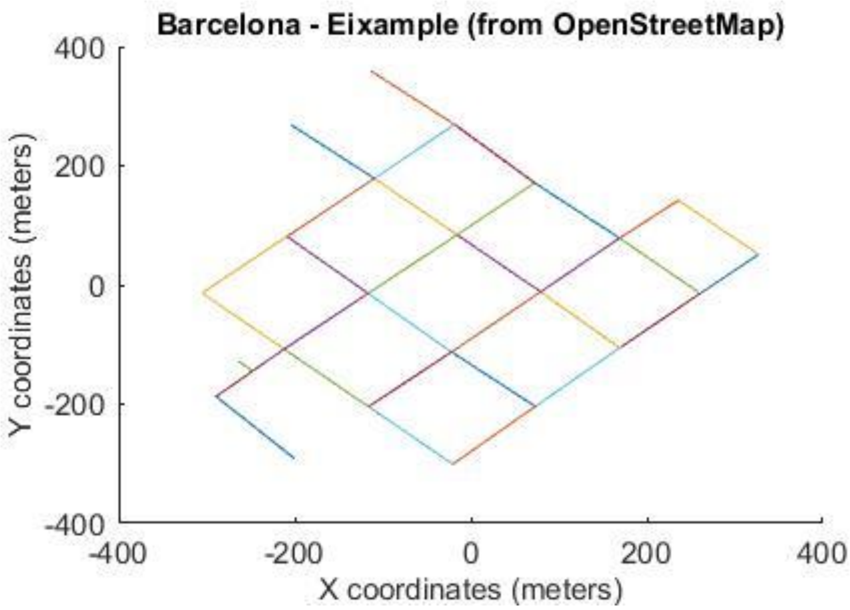


Figure A.2. L'Eixample (Barcelona) map after applying the MATLAB functions

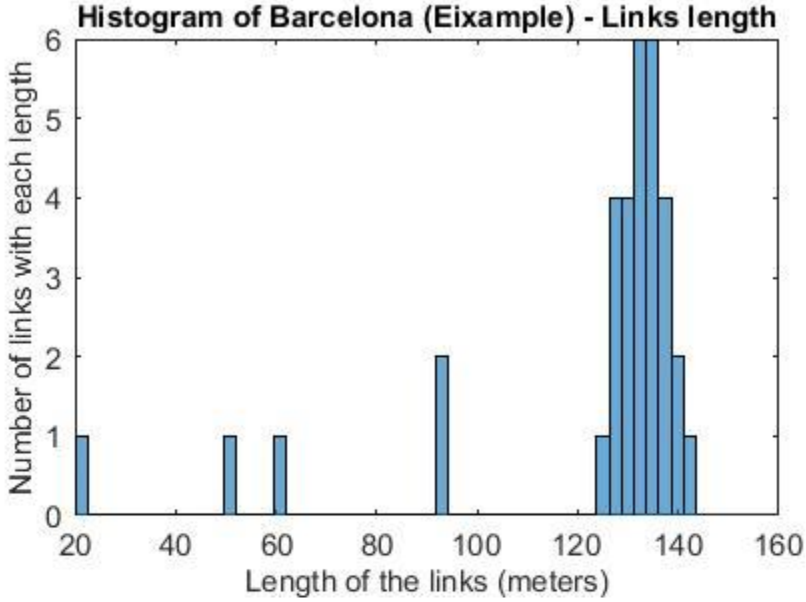


Figure A.3. Links length graph of l'Eixample (Barcelona)

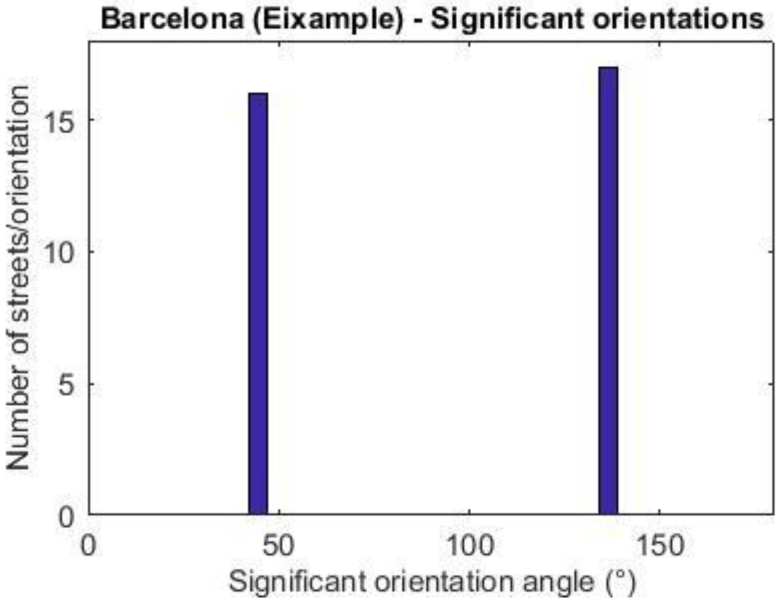


Figure A.4. Significant orientations graph of l'Eixample (Barcelona)

"Found 2 significant orientations in this zone.
And the minimum angle among streets is 88.077°.
The grid is oriented + 44.5028° East.
The distance among parallel streets are 126.3079 meters and 118.7696 meters.
The average distance among intersections is 122.653 meters.
The percentage of one-way streets with the total is 100%.
The standard deviation of the distance among streets is 27.7332 meters."

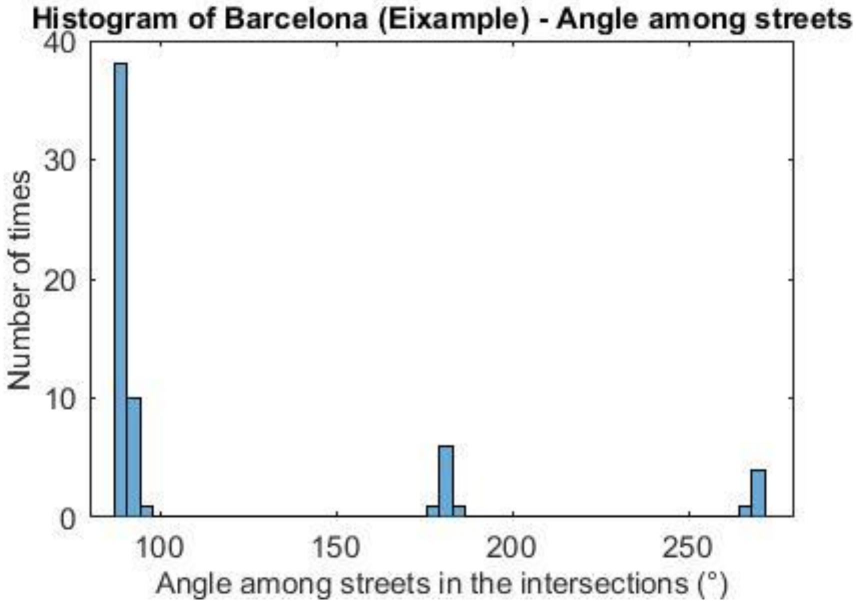


Figure A.5. Angle among streets graph of l'Eixample (Barcelona)

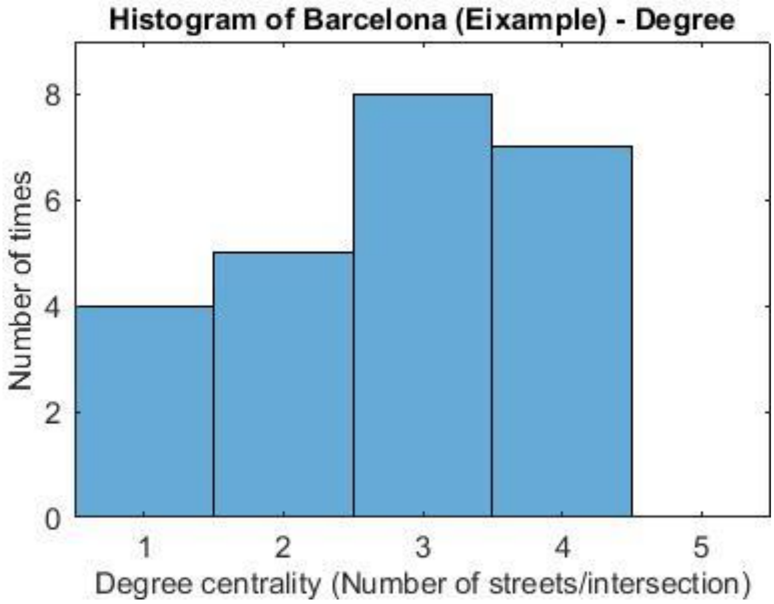


Figure A.6. Degree centrality graph of l'Eixample (Barcelona)

"The average angle among streets is 116.129°.
The standard deviation of the angle among streets is 54.7305°.
An average degree centrality is 2.75 streets/intersection."
"Number of nodes: 24
Number of links: 33
Cyclomatic number: 10
Alpha index: 0.23256
Beta index: 1.375
Gamma index: 0.5"

A.2. Barcelona – Barri Gòtic

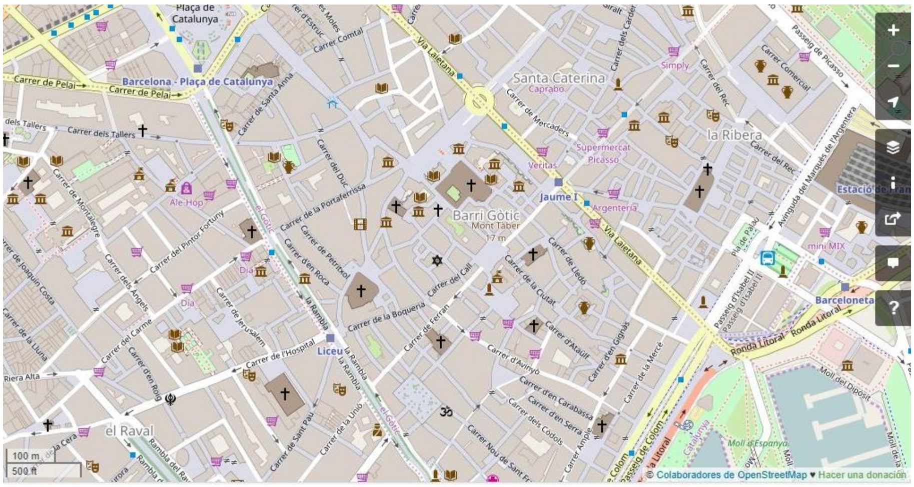


Figure A.7. Capture of Barri Gòtic (Barcelona) [Source: openstreetmap.org]

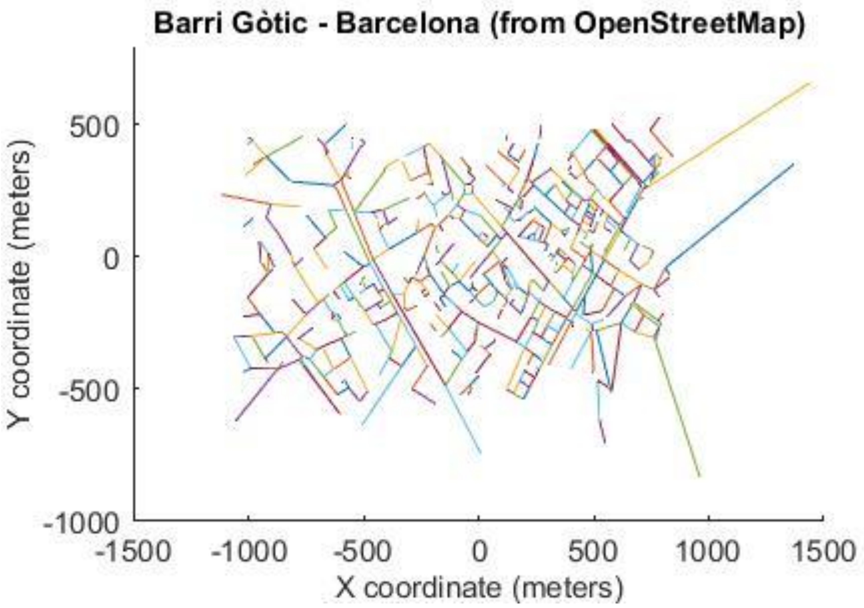


Figure A.8. Barri Gòtic (Barcelona) map after applying the MATLAB functions

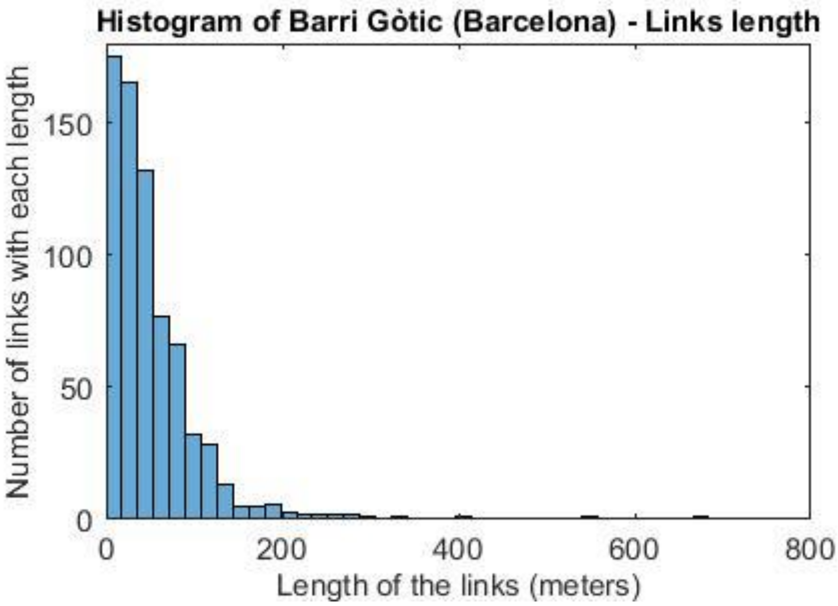


Figure A.9. Links length graph of Barri Gòtic (Barcelona)

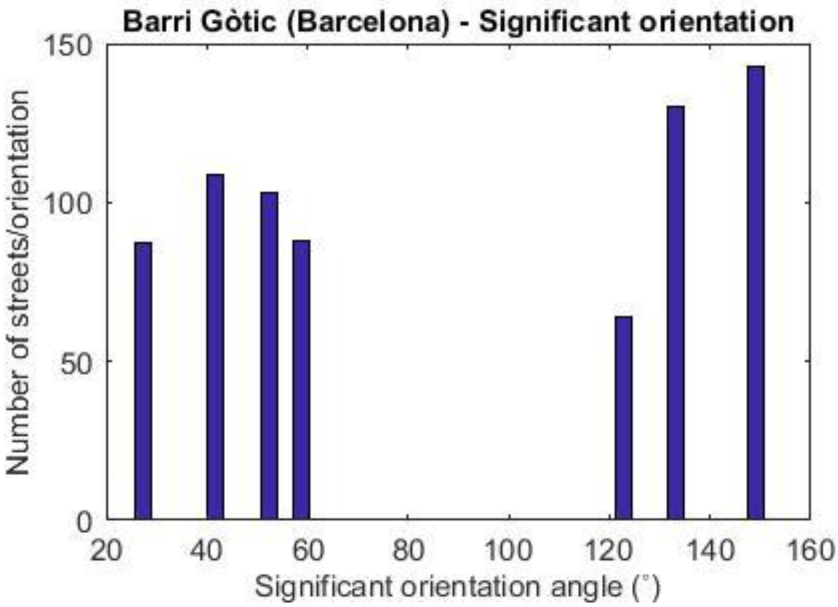


Figure A.10. Significant orientations graph of Barri Gòtic(Barcelona)

"Found 7 significant orientations in this zone.
The average distance among intersections is 53.7348 meters.
The percentage of one-way streets with the total is 80.9986%.
The standard deviation of the distance among intersections is 63.376 meters."

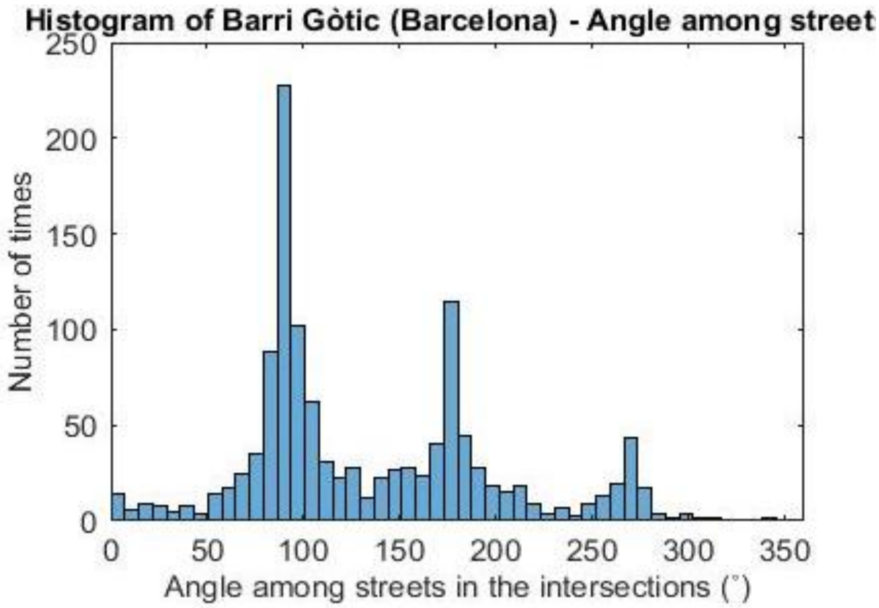


Figure A.11. Angle among streets graph of Barri Gòtic(Barcelona)

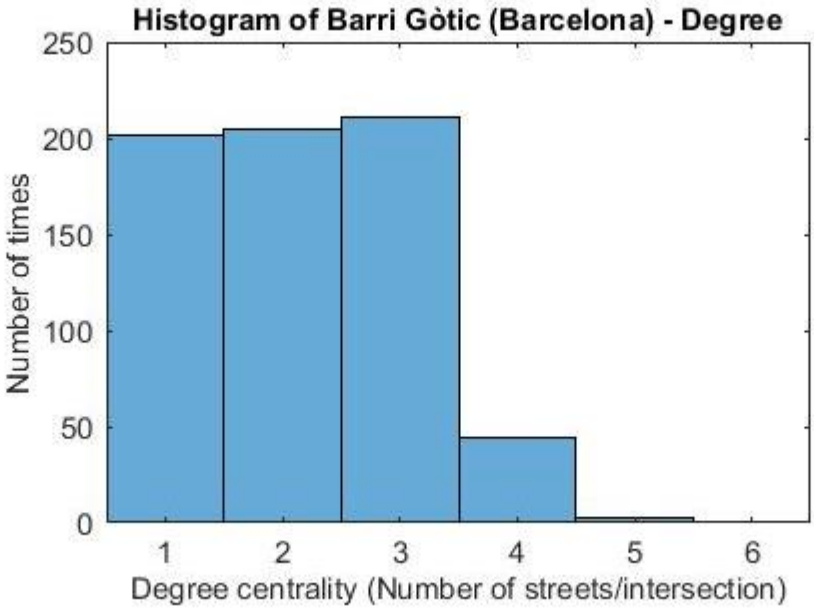


Figure A.12. Degree centrality graph of Barri Gòtic (Barcelona)

"The average angle among streets is 134.7097°.
The standard deviation of the angle among streets is 65.4378°.
The average degree centrality is 2.1652 streets/intersection."
"Number of nodes: 666
Number of links: 858
Cyclomatic number: 193
Alpha index: 0.14544
Beta index: 1.2883
Gamma index: 0.43072"

A.3. Barcelona – Eixample and Barri Gòtic

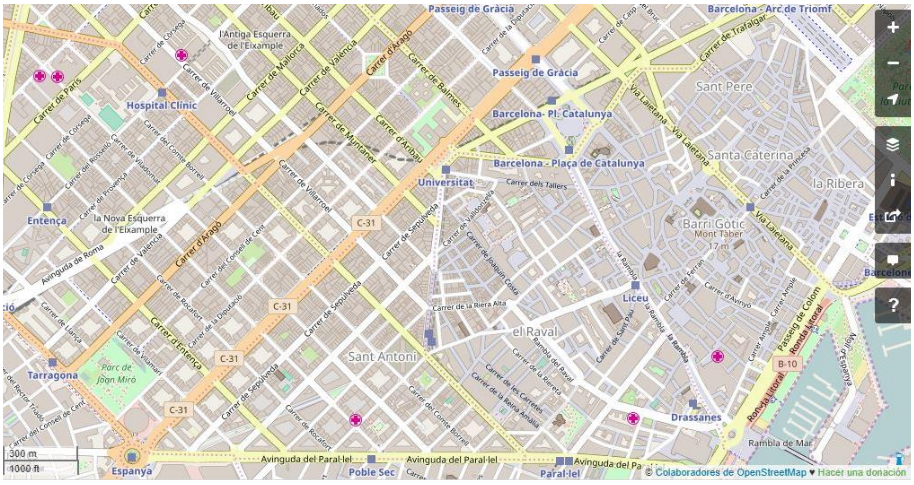


Figure A.13. Capture of l'Eixample and Barri Gòtic (Barcelona) [Source: openstreetmap.org]

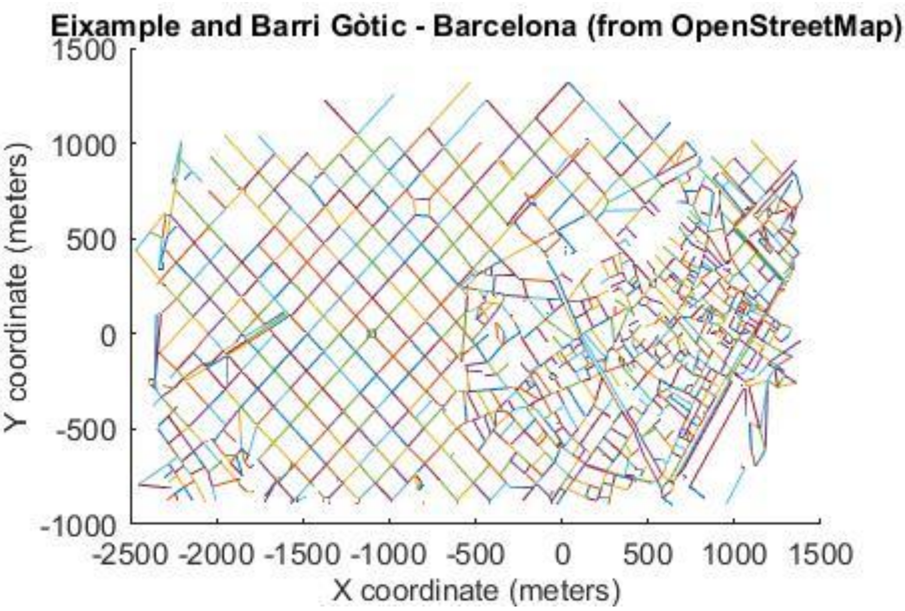


Figure A.14. L'Eixample and Barri Gòtic (Barcelona) map after applying the MATLAB functions

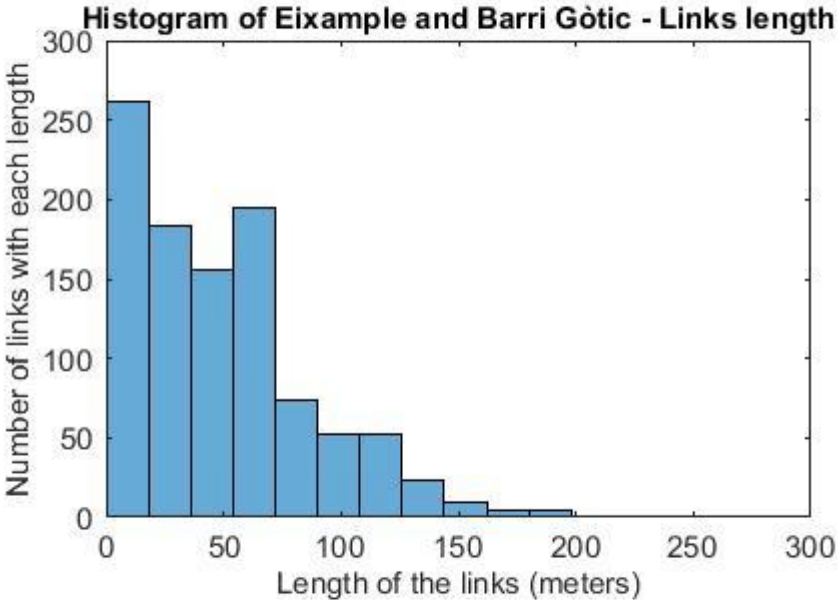


Figure A.15. Links length graph of l'Eixample and Barri Gòtic (Barcelona)

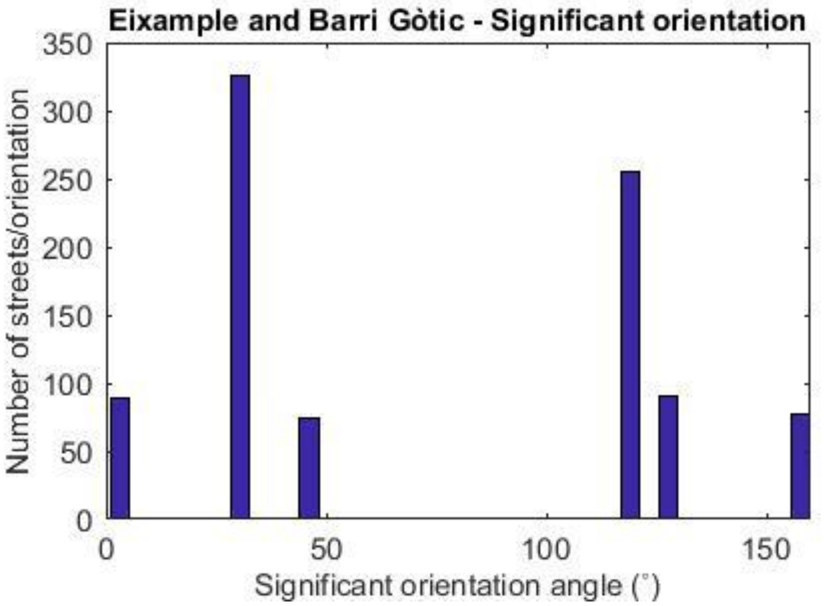


Figure A.16. Significant orientations graph of l'Eixample and Barri Gòtic (Barcelona)

"Found 6 significant orientations in this zone.
The average distance among intersections is 49.6006 meters.
The percentage of one-way streets with the total is 93.7316%.
The standard deviation of the distance among intersections is 39.4557 meters."

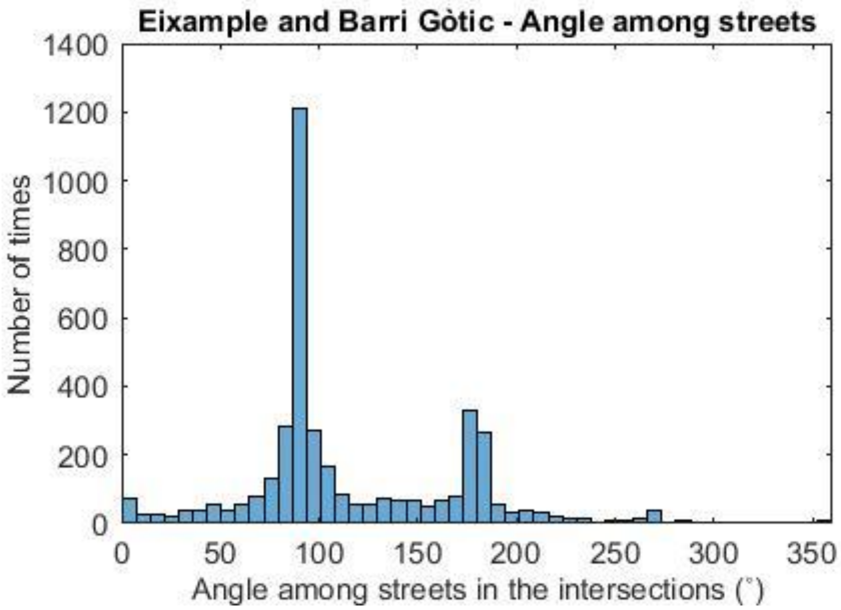


Figure A.17. Angle among streets graph of l'Eixample and Barri Gòtic (Barcelona)

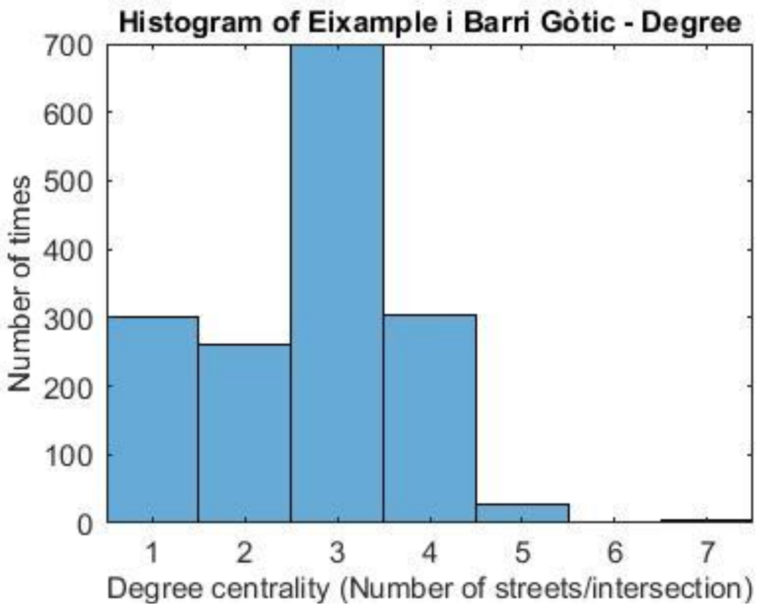


Figure A.18. Degree centrality graph of l'Eixample and Barri Gòtic (Barcelona)

"The average angle among streets is 116.36°.
The standard deviation of the angle among streets is 55.1261°.
The average degree centrality is 2.7002 streets/intersection."
"Number of nodes: 1725
Number of links: 2264
Cyclomatic number: 540
Alpha index: 0.15675
Beta index: 1.3125
Gamma index: 0.438"

A.4. Vienna

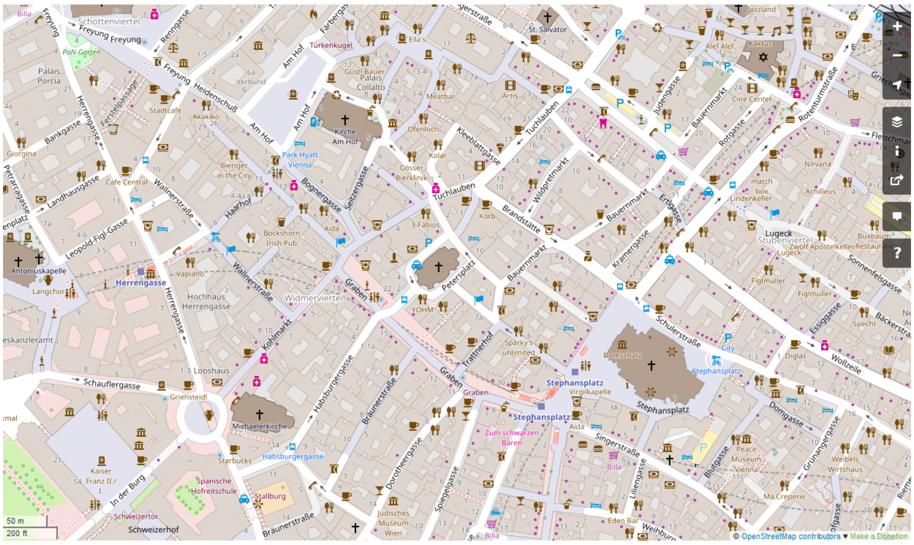


Figure A.19. Capture of Vienna [Source: openstreetmap.org]

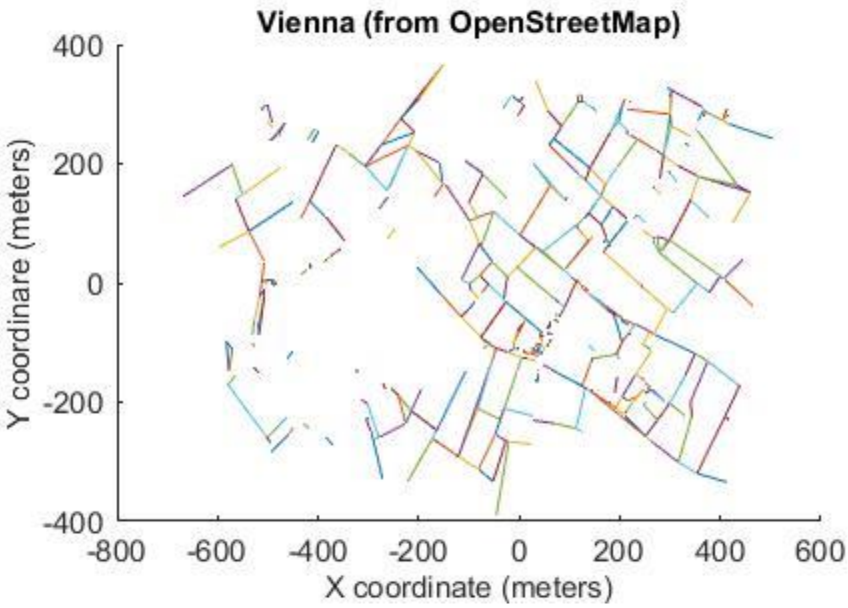


Figure A.20. Vienna map after applying the MATLAB functions

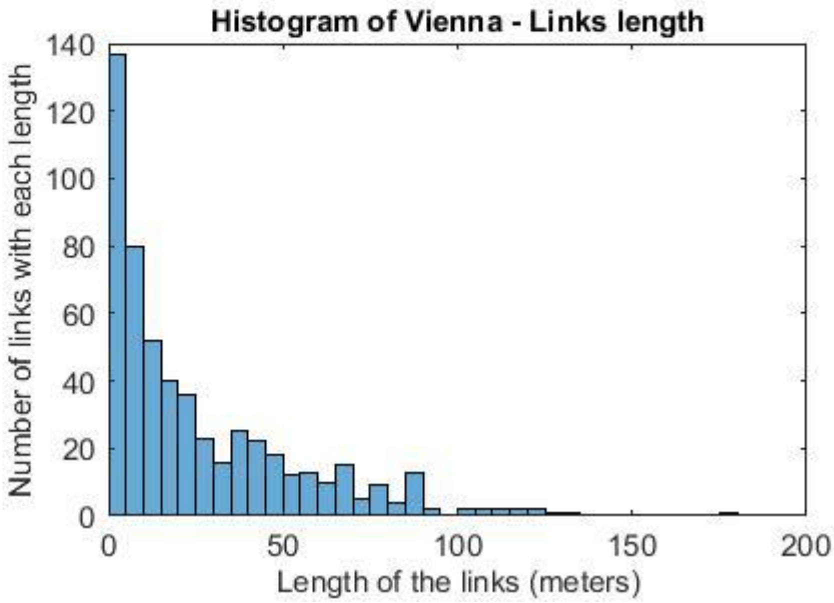


Figure A.21. Links length graph of Vienna

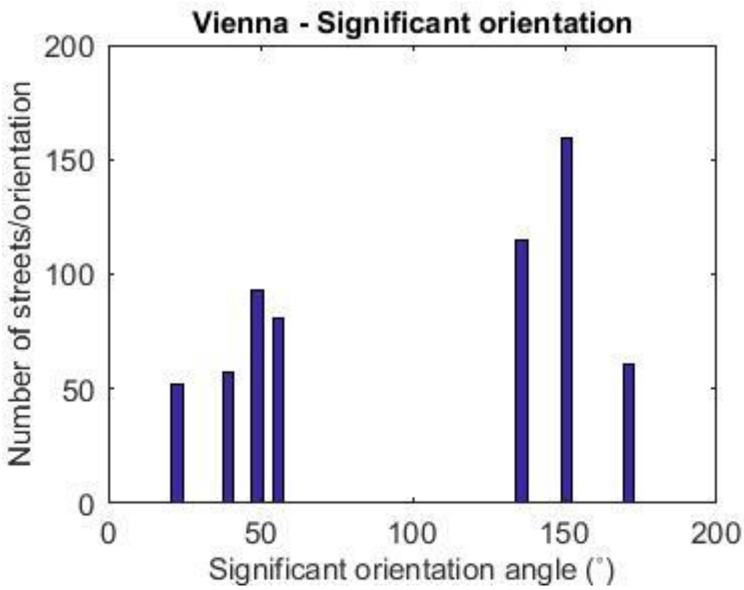


Figure A.22. Significant orientations graph of Vienna

"Found 7 significant orientations in this zone.
The average distance among intersections is 26.2711 meters.
The percentage of one-way streets with the total is 99.8165%.
The standard deviation of the distance among intersections is 27.6774 meters."

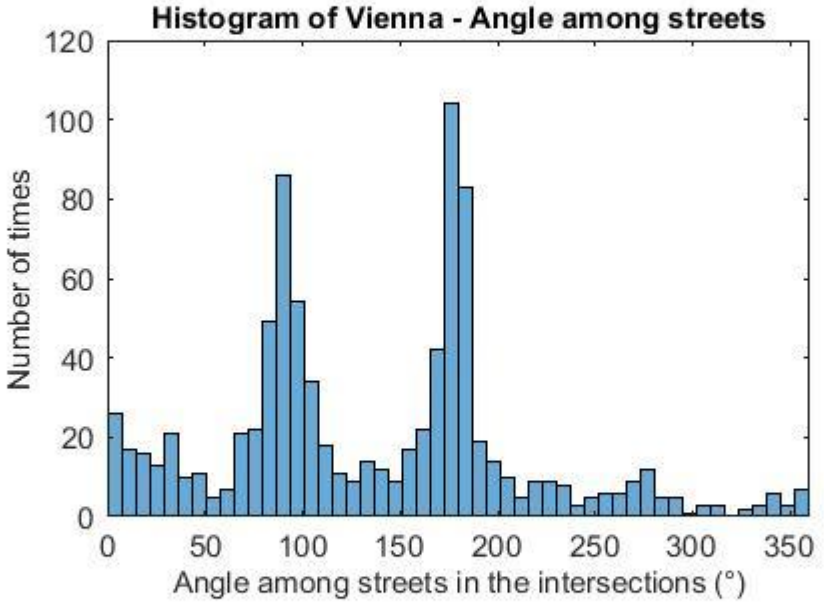


Figure A.23. Angle among streets graph of Vienna

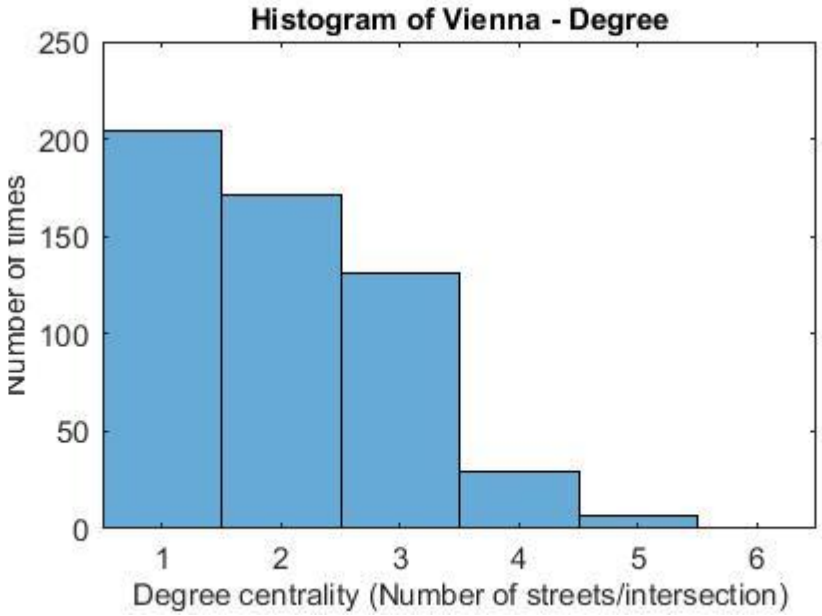


Figure A.24. Degree centrality graph of Vienna

“The average angle among streets is 137.3363°.
The standard deviation of the angle among streets is 75.6338°.
The average degree centrality is 2.0111 streets/intersection.”
“Number of nodes: 542
Number of links: 546
Cyclomatic number: 5
Alpha index: 0.0046339
Beta index: 1.0074
Gamma index: 0.33704”

A.5. Brooklyn (one grid)

Figure A.25. Capture of Brooklyn (one grid) [Source: openstreetmap.org]

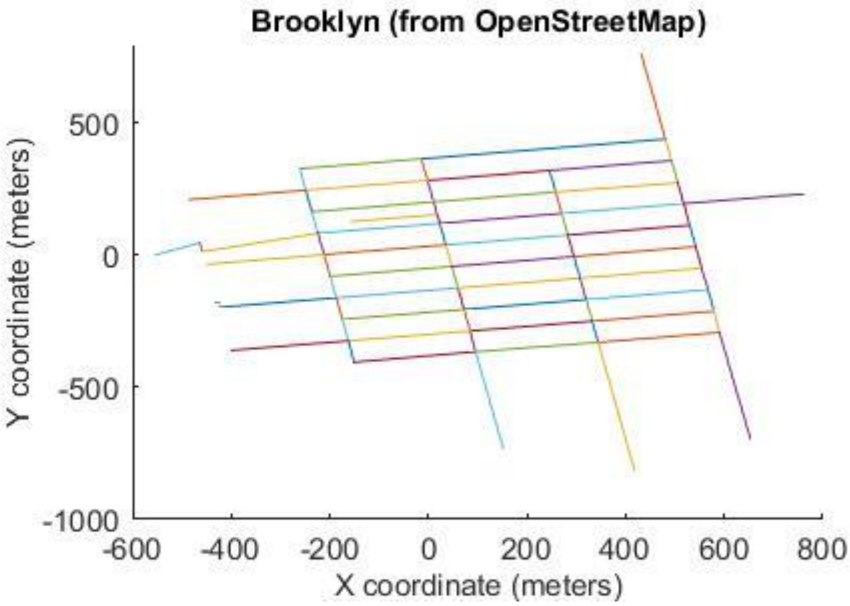


Figure A.26. Brooklyn (one grid) map after applying the MATLAB functions

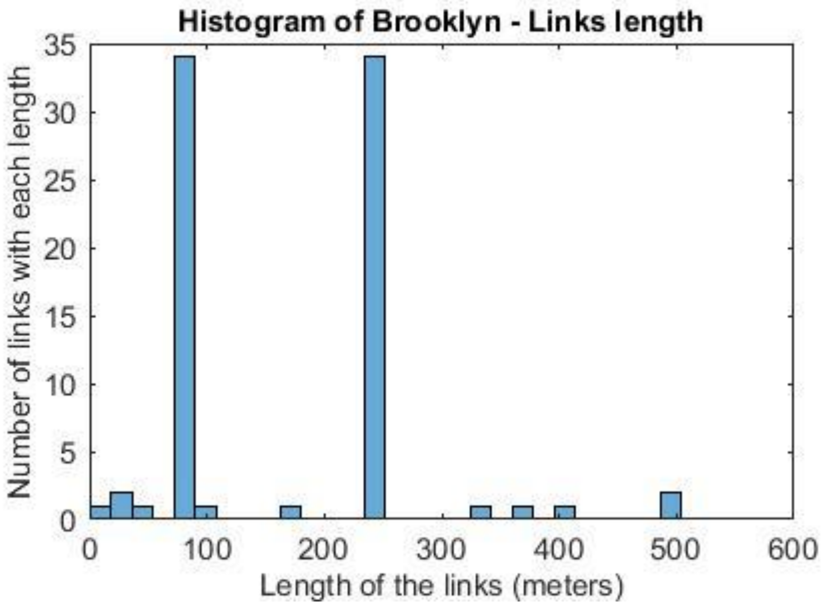


Figure A.27. Links length graph of Brooklyn (one grid)

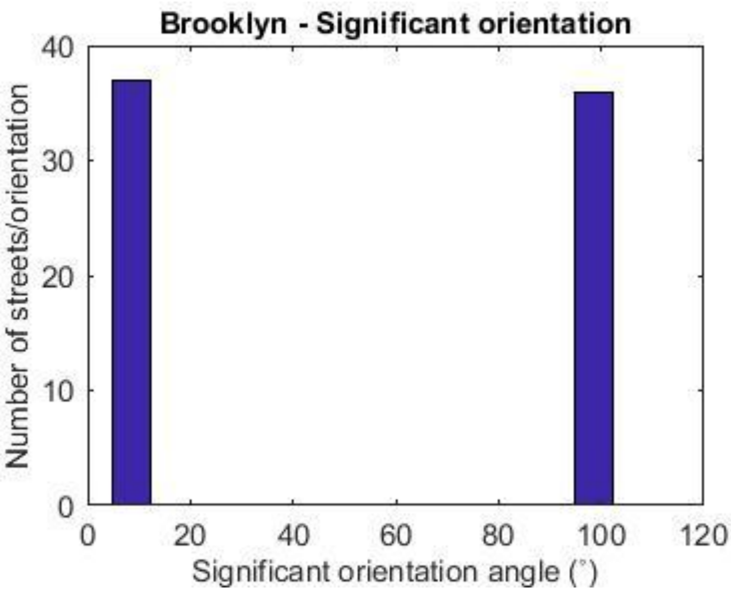


Figure A.28. Significant orientations graph of Brooklyn (one grid)

"Found 2 significant orientations in this zone.
And the minimum angle among streets is 89.969°.
The grid is oriented + 8.5943° East.
The distance among parallel streets are 247.3558 meters and 114.1517 meters.
The average distance among intersections is 174.3271 meters.
The percentage of one-way streets with the total is 98.7342%.
The standard deviation of the distance among intersections is 107.0727 meters."

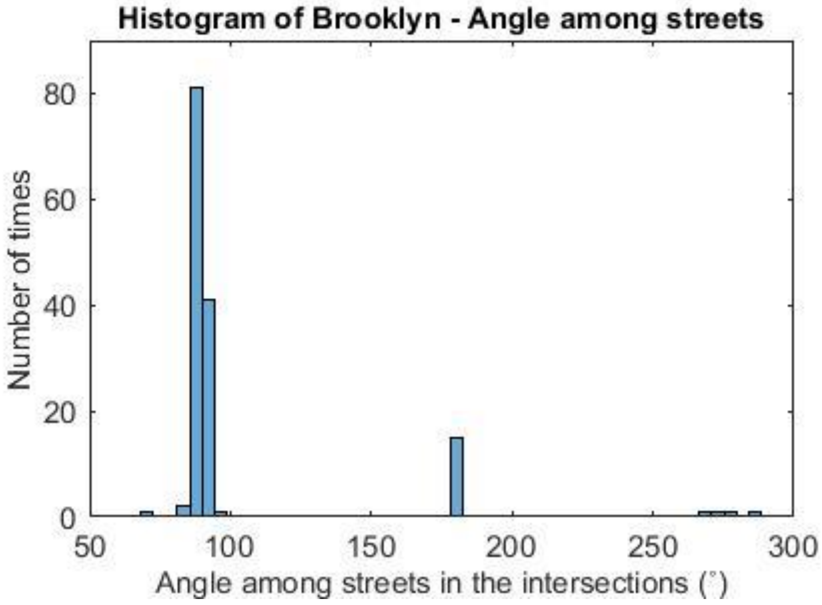


Figure A.29. Angle among streets graph of Brooklyn (one grid)

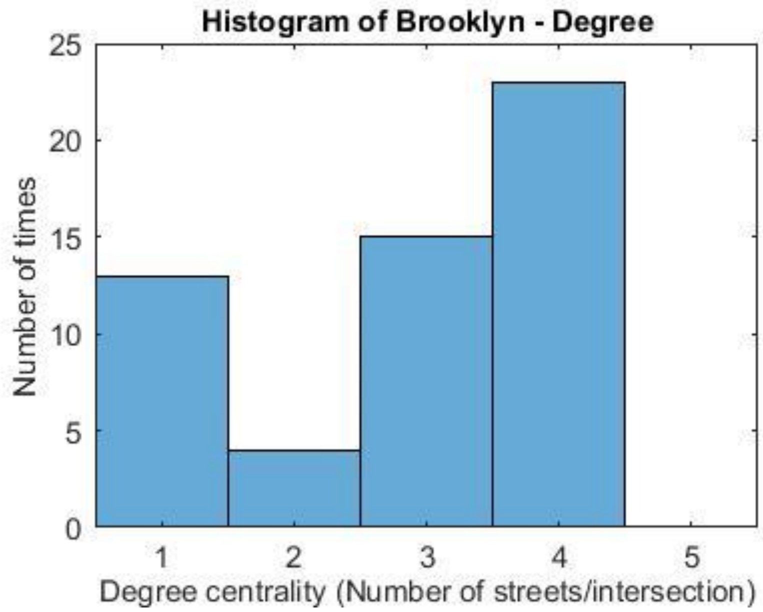


Figure A.30. Degree centrality graph of Brooklyn (one grid)

“The average angle among streets is 104.2759°.
The standard deviation of the angle among streets is 40.1604°.
The average degree centrality is 2.8727 streets/intersection.”
“Number of nodes: 55
Number of links: 80
Cyclomatic number: 26
Alpha index: 0.24762
Beta index: 1.4545
Gamma index: 0.50314”

A.6. Brooklyn (two grids)

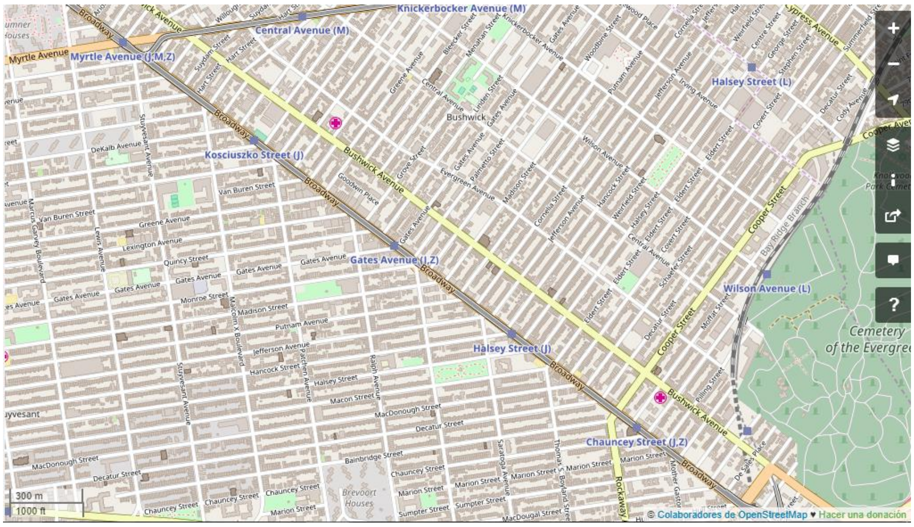


Figure A.31. Capture of Brooklyn (two grids) [Source: openstreetmap.org]

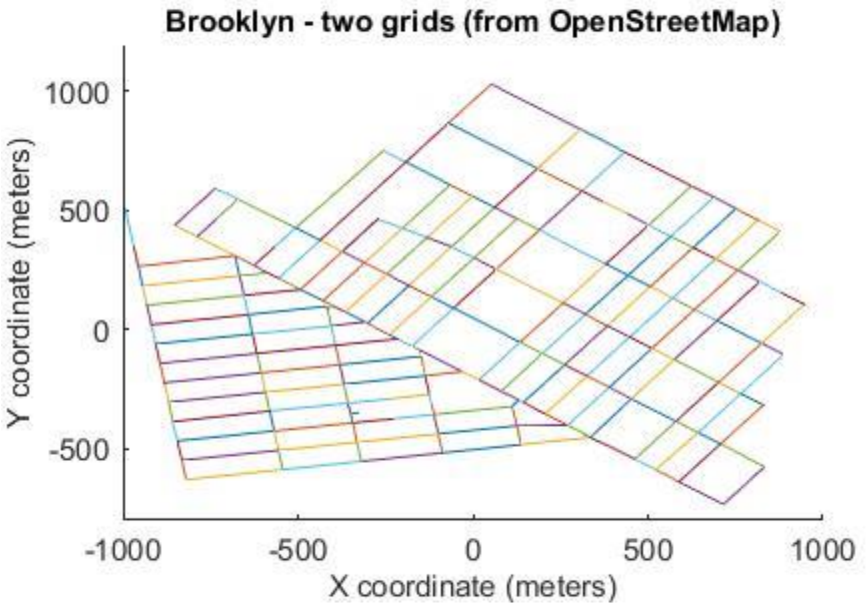


Figure A.32. Brooklyn (two grids) map after applying the MATLAB functions

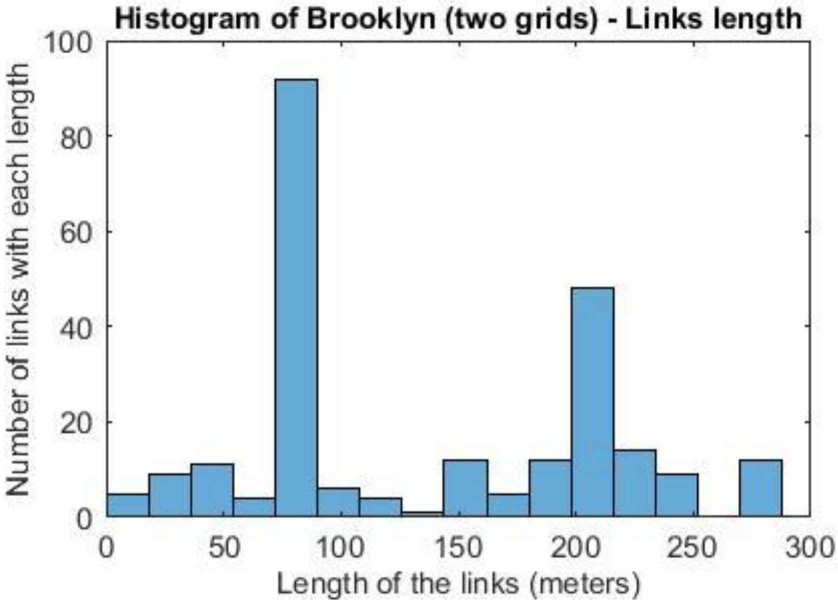


Figure A.33. Links length graph of Brooklyn (two grids)

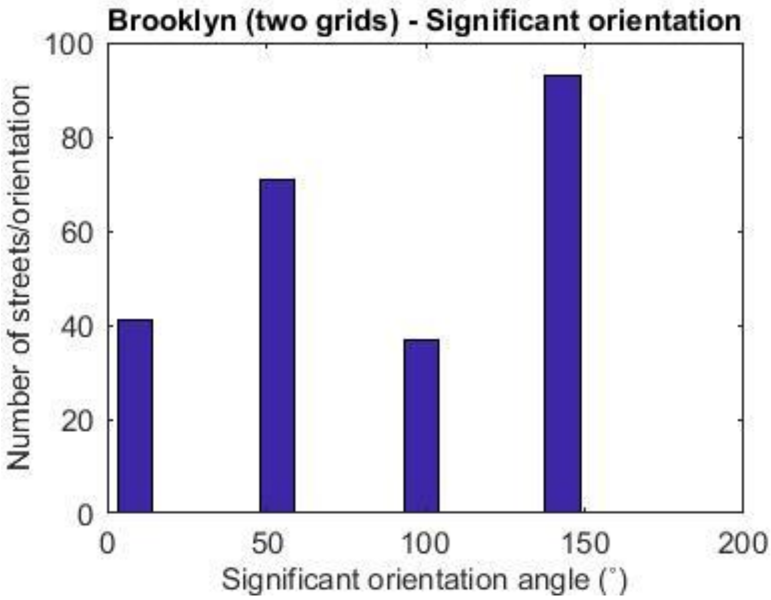


Figure A.34. Significant orientations graph of Brooklyn (two grids)

“Found 4 significant orientations in this zone.
The average distance among intersections is 143.3685 meters.
The percentage of one-way streets with the total is 99.2063%.
The standard deviation of the distance among intersections is 84.2149 meters.”

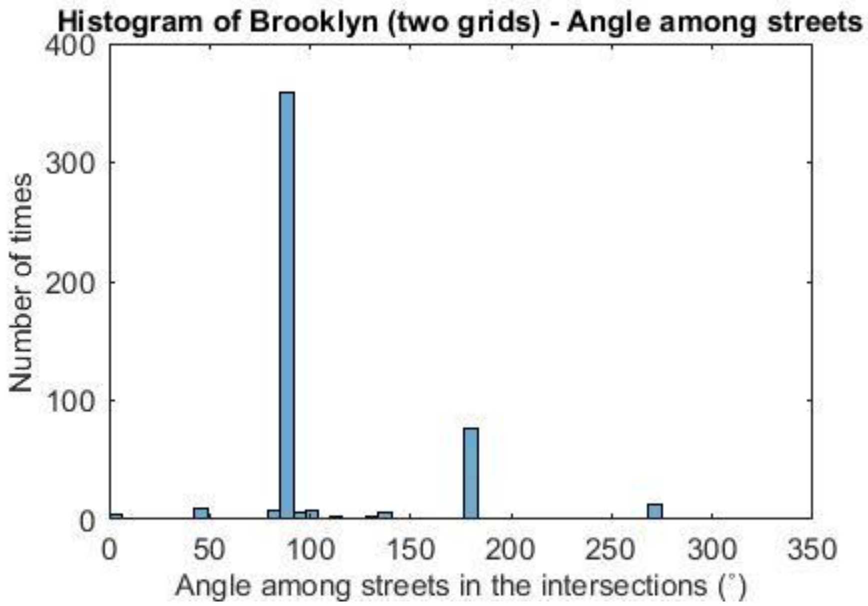


Figure A.35. Angle among streets graph of Brooklyn (two grids)

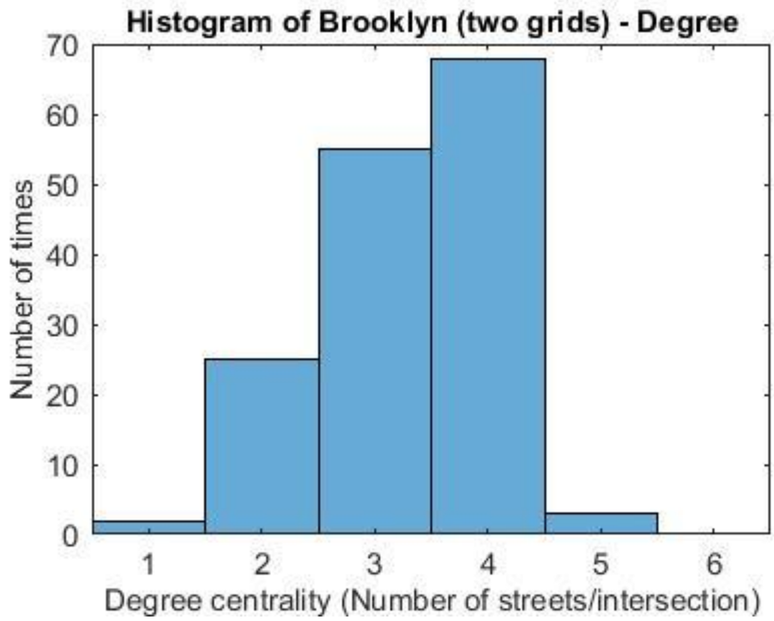


Figure A.36. Degree centrality graph of Brooklyn (two grids)

"The average angle among streets is 108.2869°.
The standard deviation of the angle among streets is 45.4171°.
The average degree centrality is 3.2941 streets/intersection."
"Number of nodes: 153
Number of links: 254
Cyclomatic number: 102
Alpha index: 0.33887
Beta index: 1.6601
Gamma index: 0.56071"

A.7. Holland – Hoorn

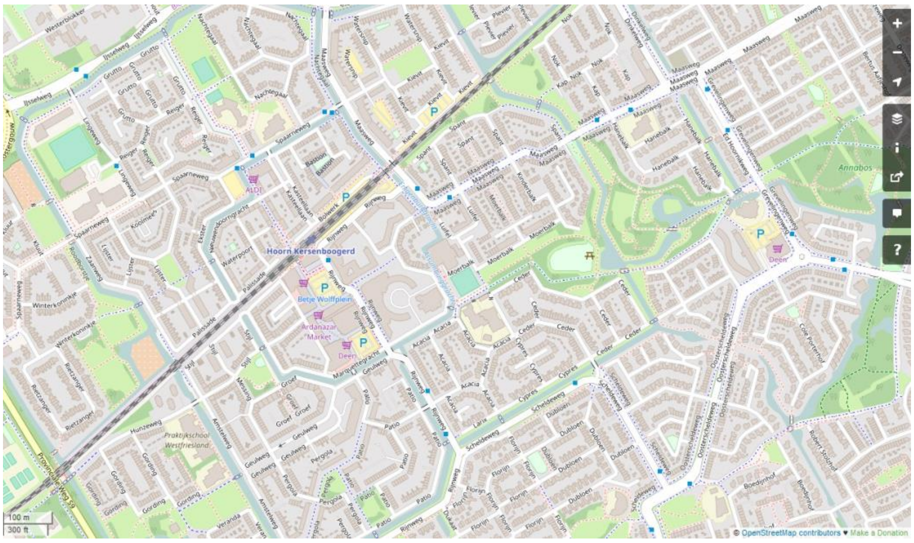


Figure A.37. Capture of Hoorn [Source: openstreetmap.org]

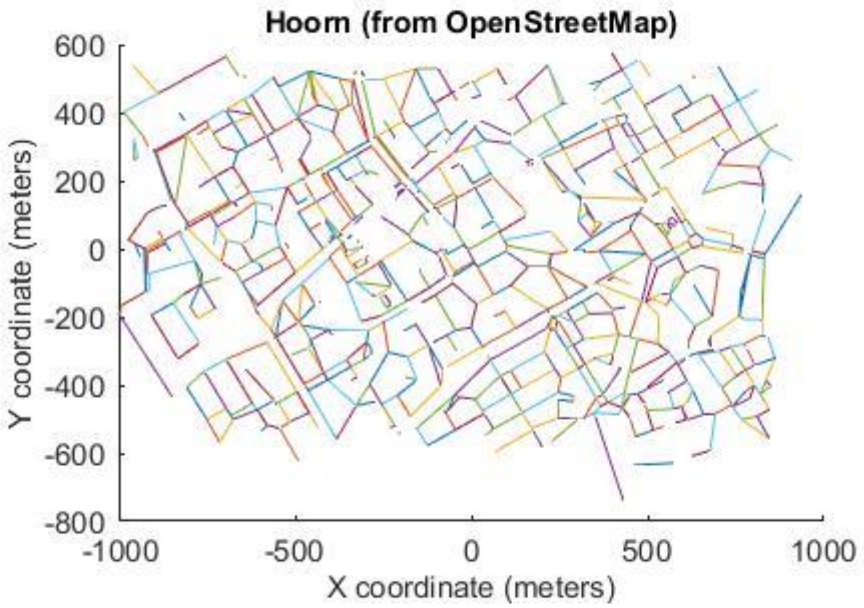


Figure A.38. Hoorn map after applying the MATLAB functions

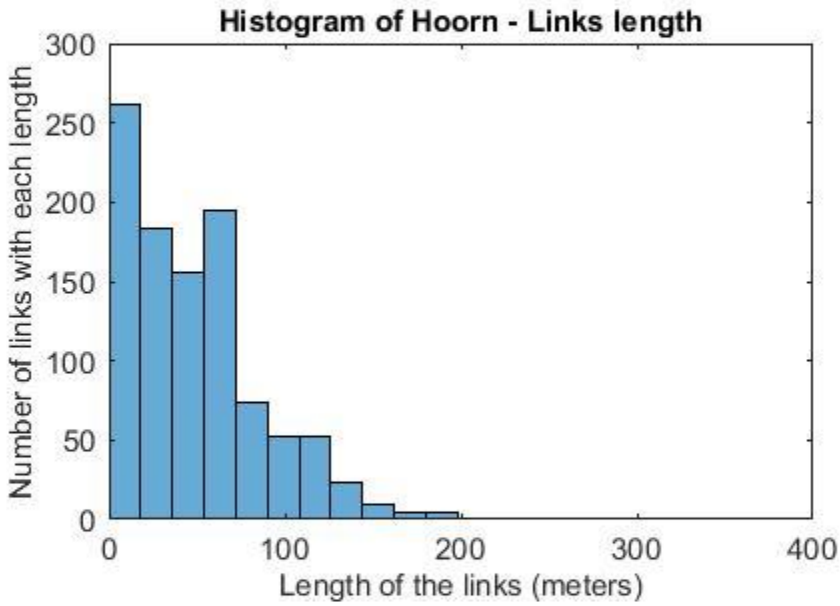


Figure A.39. Links length graph of Hoorn

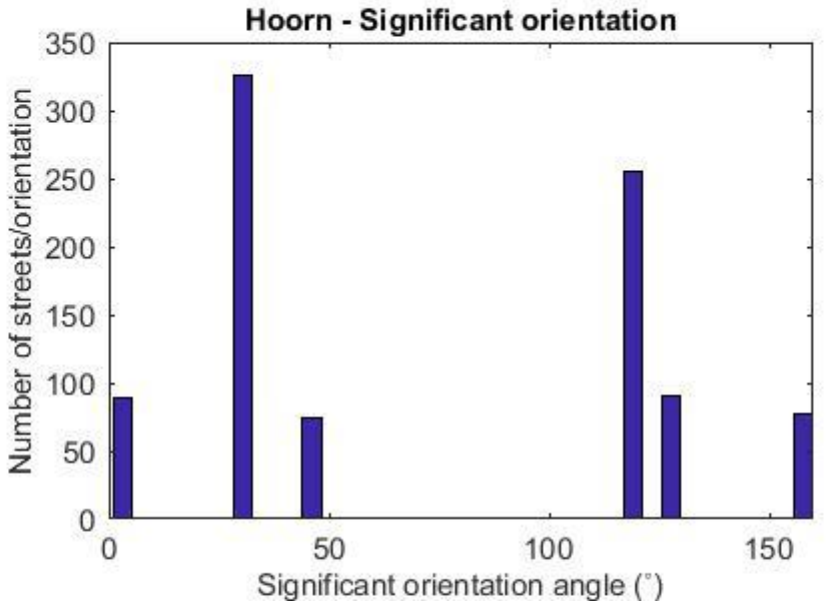


Figure A.40. Significant orientations graph of Hoorn

"Found 6 significant orientations in this zone.
The average distance among intersections is 49.6006 meters.
The percentage of one-way streets with the total is 93.7316%.
The standard deviation of the distance among intersections is 39.4557 meters."

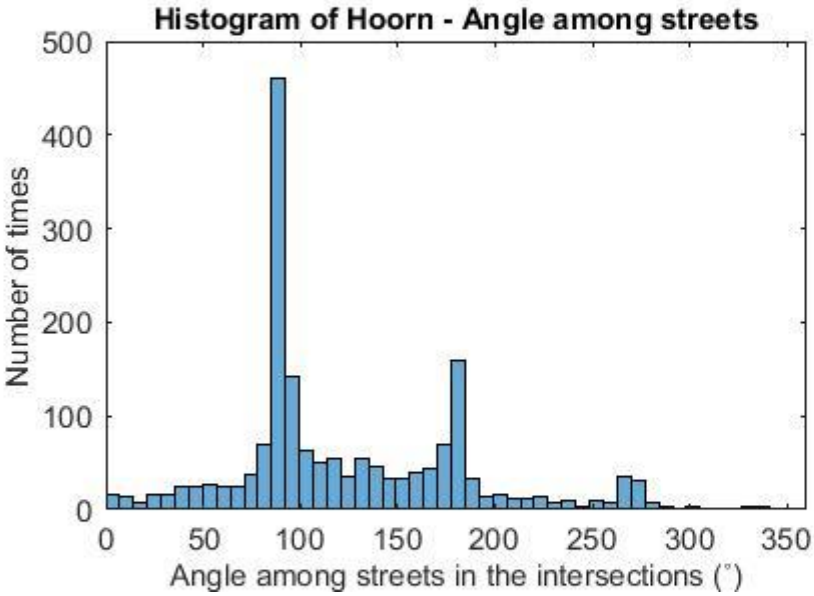


Figure A.41. Angle among streets graph of Hoorn

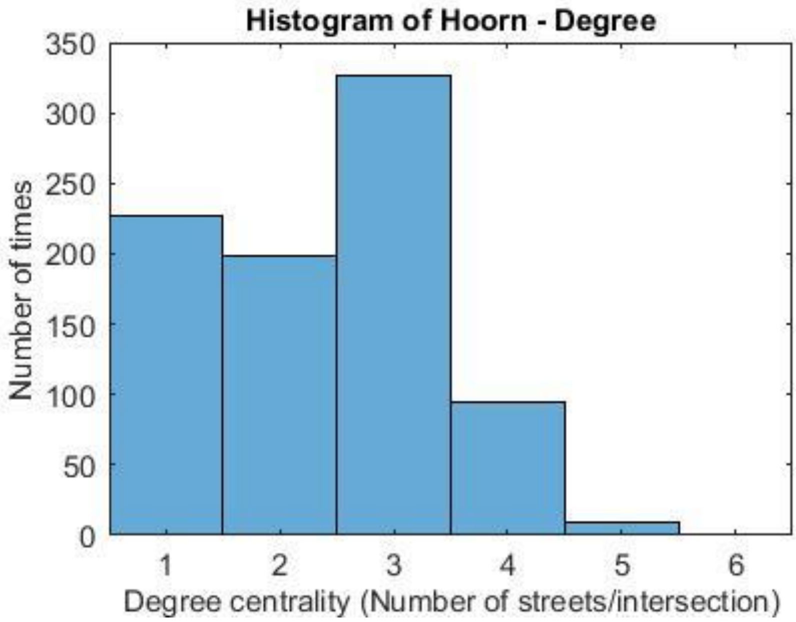


Figure A.42. Degree centrality graph of Hoorn

"The average angle among streets is 125.157°.
The standard deviation of the angle among streets is 60.6345°.
The average degree centrality is 2.38 streets/intersection."
"Number of nodes: 858
Number of links: 1085
Cyclomatic number: 228
Alpha index: 0.13326
Beta index: 1.2646
Gamma index: 0.42251"