



UNIVERSIDAD NACIONAL DE COLOMBIA

# **A Criteria based Function for Reconstructing Low-Sampling Trajectories as a Tool for Analytics**

**Edison Camilo Ospina Álvarez**

Universidad Nacional de Colombia  
Facultad de Minas  
Escuela de la Computación y la Decisión  
Medellín, Colombia  
2014



# **A Criteria based Function for Reconstructing Low-Sampling Trajectories as a Tool for Analytics.**

**Edison Camilo Ospina Álvarez**

*A thesis submitted in partial fulfillment of the requirements for the degree of  
Msc. in Systems Engineering*

*Director:*

*PhD. Francisco Javier Moreno Arboleda*

Universidad Nacional de Colombia  
Facultad de Minas  
Escuela de la Computación y la Decisión  
Medellín, Colombia  
2014



## ACKNOWLEDGMENTS

I wish to express my earnest gratefulness to PhD. Francisco Moreno for guiding me towards achieving my M.S. degree. He was exceptional mentor to me. I thank for guide me in the initial idea for the understanding of the social networks and the physical world through the concepts of trajectories. This thesis only addresses a special issue present in this topic that I want to expand in future studies.

I also wish to express my gratitude both the PhD. Francisco Moreno and PhD. Jaime Guzman for the achievements of this thesis in the congresses of 13TH INTERNATIONAL CONFERENCE OF NUMERICAL ANALYSIS AND APPLIED MATHEMATICS in *Rhodes, Greece*, the SEVENTH INTERNATIONAL CONFERENCE ON ADVANCED GEOGRAPHIC INFORMATION SYSTEMS, APPLICATIONS, AND SERVICES in *Lisbon, Portugal*, and the INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE in *Reykjavik, Iceland*.

My parents, brothers, and friends have been a tremendous support to me all along my Msc. Studies. Without their encouragement and understanding this work would not have been possible.



## **Abstract**

Mobile applications equipped with Global Positioning Systems have generated a huge quantity of location data with sampling uncertainty that must be handled and analyzed. Those location data can be ordered in time to represent trajectories of moving objects. The data warehouse approach based on spatio-temporal data can help on this task. For this reason, we address the problem of personalized reconstruction of low-sampling trajectories based on criteria over a graph for including criteria of movement as a dimension in a trajectory data warehouse solution to carry out analytical tasks over moving objects and the environment where they move.

**KeyWords:** Personalized Routing, Graph Theory, Imputation process, Trajectory Data Warehouse, Low sampling trajectories, Criteria based Trajectory Reconstruction.

## Table of Contents

<b>Abstract</b> .....	VII
<b>List of Figures</b> .....	X
<b>List of Tables</b> .....	XIII
<b>List of Equations</b> .....	XIV
<b>List of Acronymus</b> .....	XIV
<b>INTRODUCTION</b> .....	1
<b>CHAPTER 1.STATEMENT OF THE RESEARCH PROBLEM</b> .....	3
1.1 Statement of the Research Problem .....	3
1.1.1 Research Problem.....	5
1.1.2 Research Question.....	5
1.2 Motivating example .....	6
1.3 Objectives.....	9
1.3.1 General Objective.....	9
1.3.2 Specific Objectives.....	9
1.4 Scope of the research work .....	10
<b>CHAPTER 2.STATE OF THE ART: “TRAJECTORY RECONSTRUCTION PROBLEM WITH LOW-SAMPLING DATA USING A CRITERIA BASED APPROACH FROM ROUTE PLANNING THEORY – A REVIEW FOR ANALYTICS”</b> .....	11
2.1 INTRODUCCION .....	11
2.2 ROUTING PLANNING SYSTEMS .....	12
2.3 PERSONALIZATION .....	14
2.3.1 Speed: Distance.....	17
2.3.2 Speed: Time .....	17
2.3.3 Safeness.....	17
2.3.4 Simplicity .....	17
2.3.5 Attractiveness.....	18
2.4 Personalized Route Finding Based on Trajectories.....	19
2.5 Uncertainty in Trajectories.....	21
2.6 THE DW FOR TRAJECTORIES .....	23
2.6.1 Multidimensional Modeling .....	24
2.6.2 Spatial Data Warehouse (SDW) and Spatiotemporal Data warehouse (STDW) .....	24
2.6.3 Trajectory Data Warehouse (TDW) .....	25
2.6.4 The analysis goals in a TDW .....	25
2.7 Conclusions and Future Work.....	26
<b>CHAPTER 3.LOW – SAMPLING TRAJECTORY RECONSTRUCTION USING CRITERIA BASED ROUTING OVER A GRAPH</b> .....	28
3.1 INTRODUCTION .....	28
3.2 A REPRESENTATION FOR TRAJECTORIES.....	30
3.3 ADDING TRAJECTORIES CHARACTERISTICS TO ROUTES – A FORMAL APPROACH .....	30
3.3.1 Getting the location point from routing algorithms.....	35
3.3.2 Getting the timestamps from routing algorithms .....	36
3.4 IMPLEMENTATION OF THE “TRAJ” FUNCTION .....	41
3.4.1 Algorithm 1: Reconstruction of a Trajectory .....	42
3.4.2 Function 1: “traj” function for imputation data between two observations of a trajectory 43	
3.5 CONCLUSION AND FUTURE WORK.....	55



<b>CHAPTER 4. USING CRITERIA RECONSTRUCTION OF LOW-SAMPLING TRAJECTORIES AS A TOOL FOR ANALYTICS</b> .....	56
4.1 INTRODUCTION .....	56
4.2 THE PROPOSAL OF ANALYSIS.....	57
4.2.1 A Graphical analysis .....	58
4.2.2 A Trajectory Data Warehouse analysis.....	66
4.3 CONCLUSION AND FUTURE WORK.....	76
<b>CHAPTER 5. TECHNICAL DETAILS.</b> .....	78
5.1 INTRODUCTION .....	78
5.2 Technical Requirements.....	78
5.3 Source Definition .....	79
5.3.1 Foursquare data .....	79
5.3.2 Point of Interest.....	81
5.3.3 The Graph Map .....	82
5.4 Staging Definitions .....	82
5.4.1 Tables .....	83
5.4.2 Views .....	87
5.4.3 Functions/Procedures .....	88
5.5 Data warehouse definitions.....	89
5.5.1 Types.....	89
5.5.2 Tables .....	90
5.5.3 Views .....	92
5.6 ETL Process .....	93
5.6.1 Jobs.....	93
5.6.2 Transformations .....	96
5.7 Reconstructed trajectories by criteria and day .....	103
5.7.1 Check-in by day .....	103
5.7.2 Reconstructed trajectories by criteria and days.....	109
5.8 Backup for testing .....	120
5.8.1 Database Back up.....	120
5.8.2 QGIS visualizations .....	120
<b>GENERAL CONCLUSIONS</b> .....	121
<b>REFERENCES</b> .....	122

## List of Figures

Figure 1.1. Different perspectives of "better routes" .....	7
Figure 1.2. Reconstruction of a trajectory from a set of points.....	8
Figure 2.1. Schema review of the RFP according to personalization in RPS .....	14
Figure 2.2. Problem of route finding in a road network.....	15
Figure 2.3. Different route finding criteria from point A to point B: (a) Shortest Path distance from point A to point B; (b) Fastest Path distance from point A to point B; (c) Simplest Path distance from point A to point B; (d) Scenic Path distance from point A to point B.....	19
Figure 2.4. Taxonomy of Data Warehouse (DW).....	23
Figure 3.1. Some components of a RN .....	31
Figure 3.2. The concept of road distance according different criteria vs the Euclidean distance between two observations A and B.....	32
Figure 3.3. Edges where $L_{ij}$ and $L_{ij} + 1$ fit better.....	34
Figure 3.4. Imputed observations between the observations a and b.....	35
Figure 3.5. The end vertex of an edge $ek$ is the initial vertex of the edge $ek + 1$ .....	36
Figure 3.6. Inference of time stamp of edge $ek$ .....	38
Figure 3.7. Example of time assignation to a reconstructed (sub)trajectory .....	39
Figure 3.8. Additional imputed data points for an edge $ek$ .....	41
Figure 3.9. Additional timestamps data points the start and end vertex of a same edge.....	41
Figure 3.10. Portion of the city of Medellín, Colombia.....	44
Figure 3.11. Reconstructed Trajectory between Check-in A and Check-in B using Distance criterion from the user 1.....	45
Figure 3.12. Reconstructed Trajectory between Check-in B and Check-in C using Distance criterion from the user 1.....	46
Figure 3.13. Reconstructed Trajectory using Distance Criteria from the user 1.....	47
Figure 3.14. Reconstructed Trajectory between Check-in A and Check-in B. using the Time criterion from the user 1.....	48
Figure 3.15. Reconstructed Trajectory between Check-in B and Check-in C. using the Time criterion from the user 1.....	49
Figure 3.16. Reconstructed Trajectory using the Time criterion from the user 1.....	49
Figure 3.17. Reconstructed Trajectory between Check-in A and Check-in B using the Touristic criterion from the user 1.....	50
Figure 3.18. Reconstructed Trajectory between Check-in B and Check-in C. using the Touristic criterion from the user 1.....	51
Figure 3.19. Reconstructed Trajectory using the Touristic criterion from the user 1.....	52
Figure 3.20. Original Trajectory for user 1.....	52
Figure 3.21. The similarity measure between the inferred trajectories and the original one for the user 1.....	53
Figure 3.22. The average similarity measure between the reconstructed trajectories and the original ones for a set of 80 users.....	54
Figure 4.1. A set of check-in points on August 4, 2014 (Medellín).....	59

Figure 4.2. Reconstructed trajectories using Distance criterion on August 4, 2014 (Medellín) .....	60
Figure 4.3. Reconstructed trajectories using Time criterion on August 4, 2014 (Medellín).....	61
Figure 4.4. Reconstructed trajectories using Touristic criterion on August 4, 2014 (Medellín).....	62
Figure 4.5. A color gradient of reconstructed trajectories using Distance criterion.....	63
Figure 4.6. A color gradient of reconstructed trajectories using Time criterion .....	64
Figure 4.7. A color gradient of reconstructed trajectories using Touristic criterion .....	65
Figure 4.8. A Data warehouse architecture including traj function .....	66
Figure 4.9. Dimensional Model of the Data warehouse proposal .....	68
Figure 4.10. A factTrajectory fact table example.....	69
Figure 4.11. Fuel Consumption (litres / 100km) sliced by day and criteria.....	71
Figure 4.12. CO2 emissions (grams per km) sliced by day and criteria .....	72
Figure 4.13. How many MO are traversing the “Exposiciones” Street in the city of the Medellín, Colombia on August 5, 2014 (Tuesday) between 07:00:00 am and 07:00:00 pm according to time criterion? .....	73
Figure 4.14. What are the top 5 most used segment streets on August 9, 2014 (saturday) according to touristic criterion? .....	74
Figure 4.15. What is the average distance travelled on August 8, 2014 (Friday) sliced by criteria? .....	75
Figure 5.1. Example of a Json File of the user response from the API Foursquare. ....	79
Figure 5.2. Example of a Json File of the venue response from the API Foursquare .....	80
Figure 5.3. Example of a Json File of the check-in response from the API Foursquare.....	81
Figure 5.4. Points of Interest of the city of Medellín .....	82
Figure 5.5. An example of “colombiarn_2po_4pgr_medellin” table.....	83
Figure 5.6. An example of “stg_users” table .....	84
Figure 5.7. An example of “stg_venues” table.....	84
Figure 5.8. An example of “stg_check_in_data” table.....	85
Figure 5.9. An example of “stg_pois” table.....	86
Figure 5.10. An example of “stg_pointprojection” table .....	86
Figure 5.11. An example of the factTrajectory fact table .....	90
Figure 5.12. An example of the dimroadnetwork dimension table.....	91
Figure 5.13. An example of the dimcriteria dimension table.....	91
Figure 5.14. An example of the dimTrajectory dimension table.....	92
Figure 5.15. An example of the dimMovingObject dimension table.....	92
Figure 5.16. Set of check-in points on August 4, 2014 (Medellín).....	103
Figure 5.17. Set of check-in points on August 5, 2014 (Medellín).....	104
Figure 5.18. Set of check-in points on August 6, 2014 (Medellín).....	105
Figure 5.19. Set of check-in points on August 7, 2014 (Medellín).....	106
Figure 5.20. Set of check-in points on August 8, 2014 (Medellín).....	107
Figure 5.21. Set of check-in points on August 9, 2014 (Medellín).....	108
Figure 5.22. Set of check-in points on August 10, 2014 (Medellín).....	109
Figure 5.23. Reconstructed trajectories using Distance criterion on August 4, 2014 (Medellín) ..	110
Figure 5.24. Reconstructed trajectories using Time criterion on August 4, 2014 (Medellín).....	110

Figure 5.25. Reconstructed trajectories using Touristic criterion on August 4, 2014 (Medellín)..	111
Figure 5.26. Reconstructed trajectories using Distance criterion on August 5, 2014 (Medellín) ..	111
Figure 5.27. Reconstructed trajectories using Time criterion on August 5, 2014 (Medellín).....	112
Figure 5.28. Reconstructed trajectories using Touristic criterion on August 5, 2014 (Medellín)..	112
Figure 5.29. Reconstructed trajectories using Distance criterion on August 6, 2014 (Medellín) ..	113
Figure 5.30. Reconstructed trajectories using Time criterion on August 6, 2014 (Medellín).....	113
Figure 5.31. Reconstructed trajectories using Touristic criterion on August 6, 2014 (Medellín)..	114
Figure 5.32. Reconstructed trajectories using Distance criterion on August 7, 2014 (Medellín) ..	114
Figure 5.33. Reconstructed trajectories using Time criterion on August 7, 2014 (Medellín).....	115
Figure 5.34. Reconstructed trajectories using Touristic criterion on August 7, 2014 (Medellín)..	115
Figure 5.35. Reconstructed trajectories using Distance criterion on August 8, 2014 (Medellín) ..	116
Figure 5.36. Reconstructed trajectories using Time criterion on August 8, 2014 (Medellín).....	116
Figure 5.37. Reconstructed trajectories using Touristic criterion on August 8, 2014 (Medellín)..	117
Figure 5.38. Reconstructed trajectories using Distance criterion on August 9, 2014 (Medellín) ..	117
Figure 5.39. Reconstructed trajectories using Time criterion on August 9, 2014 (Medellín).....	118
Figure 5.40. Reconstructed trajectories using Touristic criterion on August 9, 2014 (Medellín)..	118
Figure 5.41. Reconstructed trajectories using Distance criterion on August 10, 2014 (Medellín)	119
Figure 5.42. Reconstructed trajectories using Time criterion on August 10, 2014 (Medellín).....	119
Figure 5.43. Reconstructed trajectories using Touristic criterion on August 10, 2014 (Medellín)	120

## List of Tables

Table 2.1. Common terms referring to routing planning systems.....	13
Table 2.2. Quantitative and qualitative criteria of RFP.....	16
Table 3.1. Check – in data of a particular user.....	43
Table 3.2. Imputed observations using the Distance criterion between Check-in A to Check-in B.....	45
Table 3.3. Imputed observations using the Distance criterion between Check-in B to Check-in C.....	46
Table 3.4. Inferred observations using the Time criterion between Check-in A to Check-in B. ....	47
Table 3.5. Inferred observations using the Time criterion between Check-in B to Check-in C. ....	48
Table 3.6. Imputed observations using the Touristic criterion between Check-in A to Check-in B.....	50
Table 3.7. Inferred observations using the Touristic criterion between Check-in B to Check-in C.....	51
Table 4.1. Quantity of check-in´s users by day (Medellín).....	67
Table 4.2. Fact table of reconstructed trajectories of a set of objects for each criterion.....	69
Table 4.3. Some measures of interest in a TDW.....	70
Table 5.1. Staging schema views .....	87
Table 5.2. Job: jobPrincipal .....	93
Table 5.3. Job: jobLoadStage.....	93
Table 5.4. Job: jobLoadTDW.....	94
Table 5.5. Job: jobExtractData.....	94
Table 5.6. Job: jobReconstructTrajectories.....	95
Table 5.7. Job: jobLoadDimensions.....	95
Table 5.8. Job: jobLoadFactTrajectory .....	96
Table 5.9. Transformation: traExtractUserData.....	96
Table 5.10. Transformation: traExtractVenuesData .....	97
Table 5.11. Transformation: traExtractCheckinData .....	97
Table 5.12. Transformation: traExtractPOIData.....	98
Table 5.13. Transformation: traSetCostforCriteria .....	98
Table 5.14. Transformation: traSetPointProjection.....	99
Table 5.15. Transformation: traSetPointProjection.....	99
Table 5.16. Transformation: traReconstructTrajectory .....	100
Table 5.17. Transformation: traLoadDimRoadNetwork.....	100
Table 5.18. Transformation: traLoadDimCriterion.....	101
Table 5.19. Transformation: traLoadDimMovingObject.....	101
Table 5.20. Transformation: traLoadDimTrajectory.....	102
Table 5.21. Transformation: traLoadFactTrajectory .....	102

## List of Equations

Equation 3.1. Minimum distance between $L_{ij}$ and a road segment $ek$ .....	34
Equation 3.2. Computation of the (sub)trajectory between $L_{ij}$ and $L_{ij} + 1$ .....	36
Equation 3.3. The total distance between two observations $L_{ij}$ and $L_{ij} + 1$ .....	37
Equation 3.4. The timestamp of a $get\_vertex\_target(ek)$ .....	37
Equation 3.5. Line equation over the segment represented by $ek$ .....	39
Equation 3.6. The $get\_x$ function slicing of the segment represented by $ek$ .....	39
Equation 3.7. The $get\_y$ function slicing of the segment represented by $ek$ .....	40

## List of Acronymus

*Some useful acronymus used along the thesis are explained first.*

DBMS	Database Management Systems
MOD	Moving Objects Databases
MO	Moving Objects
GPS	Global Positioning System
LBS	Location Based Services
POI	Point of Interest
DW	Data Warehouse
SDW	Spatial Data Warehouse
STDW	Spatio Temporal Data Warehouse
TDW	Trajectory Data Warehouse
BI	Business Intelligence
MBR	Minimum Bounding Rectangle



## INTRODUCTION

The easy acquisition and spreading of devices with incorporated GPS have highlighted the active use of location based services. Those applications and devices are characterized by the delivering of location data which ordered in time represent trajectories. Trajectories provide information to understand moving objects and the space where they move. Research and computer technologies for processing, retrieving, and extracting knowledge from those trajectories are needed.

Although, some GPS systems can log the movement in a high sampling rate, others log the data in a low sampling rate describing the movement poorly and generating uncertainty. This is because of issues such as privacy (people do not share their location every time), energy saving, or simply, because the location based application only delivers location when a user arrives a place, e.g., the check-in in Foursquare and the geo-tagged photos in Flickr. As a result, the trajectory must be reconstructed to know how the movement was between no location-data availability.

This thesis deals with the reconstruction problem of low sampling trajectories in a network restrained environment. From the premise that each moving object has a lot of possibilities for moving in a road network (because of the roads complexity), a reconstruction operator is provided considering the possible criteria that an object follows when it moves and the underlying road network where the movement occurs. The criteria based reconstruction addressed here argues that a user deals with a path selection problem: shortest distance is not always the criterion for moving in a city. Time, simplicity of the road, and touristic criteria are also considered by the user.

The goal of reconstruction is to impute the movement between two location points to deal with the uncertainty. The reconstruction transforms “raw trajectories” in an appropriate form for the subsequent analysis. Because of the criterion of movement change, the resulting reconstructed trajectories can be different and; therefore, the analysis derived from those reconstructed trajectories can also change. For that reason, we explore the change in the analysis using approaches of data warehouse specialized in the management of spatiotemporal data to understand the reconstruction of trajectories based on criteria.

In the following paragraphs, each chapter that make up this thesis are sketched out and their purpose is summed up.



**Chapter 1. “*Statement of the Research Problem*”.** States the research problem addressed by this thesis. The research question including specific research questions are also explained using a motivating example. The general and specific objectives are enunciated. A basic motivation for the research is also depicted and the scope of the research work is stated.

**Chapter 2. “*Personalized Trajectory Reconstruction Problem with low-sampling data – A review*”.** Describes the state of art of the topics related to the development of this thesis. The related research works considered here includes topics such as: *Routing/Route Planning, Low-sampling trajectories, Data warehouse, Trajectory Data warehouse, Trajectory reconstruction.*

**Chapter 3. “*Trajectory Reconstruction using criteria based routing over a Graph*”** describes the solution related to the operator to reconstruct trajectories. Using a formal approach, a function is formulated and developed. The graph theory, route planning theory, and trajectory concepts are carefully included to accomplish the goals of the function. The delimitation of the scope of the solution proposed is also stated.

**Chapter 4. “*Using Criteria Reconstruction of Low-sampling trajectories as a tool for analytics*”** extends the solution proposed in Chapter 3, for including criteria of movement as a dimension of analysis in a trajectory data warehouse solution to enhance the analytics using dimensional modelling and graphical analysis.

**Chapter 5. “*Technical Details*”** describes the technical documentation to provide a more comprehensive understanding of the solution. It also pretends to provide the technical details to replicate the executed experiments and examples.

Both, the conclusions and the main contributions to the overall specific objectives of this thesis proposal are separated and located at the end of each chapter. Those chapter are: *Chapter 2, Chapter 3, Chapter 4, and Chapter 5.*

## **CHAPTER 1. STATEMENT OF THE RESEARCH PROBLEM**

### **1.1 STATEMENT OF THE RESEARCH PROBLEM**

The current availability of GPS equipped devices, mobile phones and other mobile computing technologies [1] that use location data as a functionality is becoming fundamental to carry out the everyday actions of people and businesses. This has opened up the possibility for the collection, representation, exploration, and analysis of moving data which demands applications for new and enhanced location-based services such as tourism, marketing, sales, location-based gaming, and transportation systems [2] e.g., Google Maps, Flickr or Foursquare. These applications and technologies keep constant the underlying concept of the movement in space. The movement has to do with the notion of change in the physical position of a spatial object, called Moving Object (MO), respect to some reference system [3], [4].

The delivering of location data ordered in time describe trajectories [5], [6], [7], which, in turn, represent the movement of a MO in space. However, because of the characteristics of some location based applications (e.g., the sharing of location data are done when a user arrives to representative places [6]; the energy saving of devices or, simply, the privacy requirements of users [8]) the movement is poorly described by low-sampled logged data. Although, sources of uncertainty are multiple (e.g., the measurement of the GPS equipped devices), the low sampling rate of trajectories is only addressed here because it involves a preprocessing stage of reconstruction of the trajectories that approximates the movements between localization points [9] later called by [10] as silent durations.

The route planning problem (i.e., the problem of finding the optimal path for a user) [11] is a related problem considered here. However, systems are limited regarding route planning [12] because they are mainly focused on a single criterion, i.e., the shortest distance. The problem of route planning considering a set of metrics different from distance and the integration of user criteria is a still an open research issue [12] and requires the adaptations of new customized metrics, and possibly combinations of them, for finding the route between two places. [11] and [12] offer a brief taxonomy to build the “best” route based on criteria like shortest distance, time, point of interest (POI), and simplicity for traversing the RN.

Addressing this same need of route planning theory, i.e., the integration of user criteria to get “better routes” [14], [15], [16], a novel and relevant task is the reconstruction of low-sampling trajectories based on particular properties of the movement, such as the criteria of the MO (i.e., user) and the geographical space where it occurs, e.g., the road network (RN) of a city (the possible whereabouts of the MO are delimited by the geometry of the RN [17], [18], [19]: the movement is constrained along the edges (streets) of the RN) to handle with the uncertainty derived from the low-sampling rate generation of location data.

The reason for trajectory imputation process, i.e., the reconstruction, is being a previous step of preprocessing before knowledge extraction and analysis of location data [20]. However, a great challenge lies in the knowledge discovery (of the environment where the movement occurs and the MO in consideration) using spatio-temporal data [21], especially about trajectories [10], even more when those trajectories are characterized by the uncertainty [6].

The resulting trajectories should be stored in appropriate repositories to accomplish this task [9]. Also, it must be done because a great number of MO providing data and the reconstruction process themselves can result in a huge data generation. Data Warehouse (DW) approaches [22] might be used to deal with these huge volumes of data and analyze it. Because many of the characteristics, such as hierarchies and aggregations, and techniques such as mining and visualization have been adapted to the spatio-temporal data into a new concept called Spatio-Temporal Data Warehouse (STDW) [23], [24], [25], the analysis of the imputation process (i.e., reconstruction) over low-sampling trajectories considering different criteria as an analysis dimension is based on this approach. Specifically, this thesis proposal, only deals with a particular case of the STDW approach called Trajectory Data Warehouse (TDW) fed by time-dependent location data describing movements of MO, i.e., trajectories [26], [27].

From the above expressed research needs and functionalities, in this thesis proposal, four main issues for managing MO data are considered:

- Trajectories derived from location data are low-sampling because of the characteristics of location based application such as: people sharing location is only done when a user arrives to a POI (make check-in), privacy issues, energy saving or communication problems. When reconstructing a trajectory, it is also necessary deal with the uncertainty of low-frequency data [6], [7].

- Dealing with multiple metrics expressing user criteria in a nontrivial way for the reconstruction of low-sampling trajectories is still an open research issue and is still expressed as a need [10], [12], [28].
- The environment in which movements take place and the characteristics of MO have significant influence on the movement; therefore, they need to be considered when the movements are studied [5], [29].
- The DW based on spatio-temporal data still lacks of analytical tasks related, e.g., to the reconstruction of low-sampling trajectories [6], [7], [10].

### 1.1.1 Research Problem

According to the issues outlined, the research problem to be addressed in this thesis is:

*The spatio-temporal data systems still lacks of analytic tasks related to the dynamic possibilities of route planning based on the reconstruction of low-sampling trajectories considering different user criteria.*

### 1.1.2 Research Question

And its corresponding research question is:

*Can an operator be developed for the reconstruction of low-sampling trajectories considering different user criteria to increase the possibilities of analytical tasks over trajectories?*

This question arise the following research questions to be developed:

- *Can data of low-sampling trajectories be reconstructed considering user criteria?*
- *Which analysis tasks could be performed using a criteria based operator over low-sampling trajectories?*

## 1.2 MOTIVATING EXAMPLE

A motivating example is developed to clarify the research problem:

See the *Figure 1.1*, two sample points generated by an app, e.g., Foursquare. An origin located in the point A (“*Parque Berrio*”) and a destiny located in point B (“*Alpujarra*”) of the city of Medellín.

The *Figure 1.1-a* presents the basic notion of road distance. The figure shows the minimum distance between point A and B. In real life, it is not possible for a car follows this path because, for example, the streets have restrictions of mobility such as the direction of movement.

The *Figure 1.1-b* is based on the relevance of time. The path shown is the best route, because, for example, the traffic flow is fastest between the point A and point B.

The *Figure 1.1.c* presents a notion of distance based on the "easiest" path, evading the most of the turns in the path. It is based on the idea that the presence of the turns implies the reductions of velocity and unnecessary maneuvers.

Another perspective is presented in the *Figure 1.1-d*. It is based on the notion of touristic perspective and POI. The idea is to travel from A to B trying to visit the more touristic places as possible in an optimal way.



Figure 1.1. Different perspectives of "better routes"

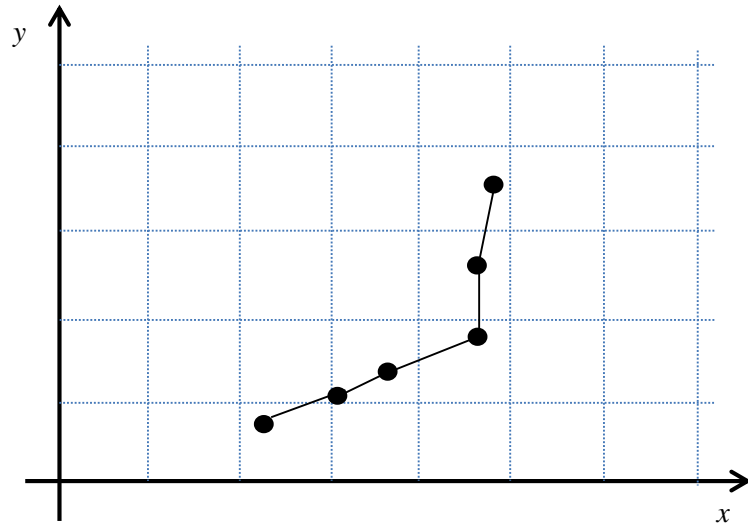
All these kinds of perspectives of "the better routes" are detailed in the state-of-art section where the possibilities, such as *time* and *touristic* criteria define the reconstruction of the trajectory.

As shown in the *Figure 1.1*, the trajectory from point A to point B has a lot of possibilities due to different movement criteria of the users. Now, suppose you must to reconstruct a similar dataset of low-sampling trajectories of a dataset of MOs. What methods do you use? Which analytical task could be performed over some MOs which follows similar trajectories in a city?

Using a determined criterion parameter as a basis, the path is reconstructed using an operator over a set of trajectories. An analysis task in a TDW could be the comparison of the different reconstructed approaches to analyze the differences and tendencies of the MO to determine the characteristics of the movement, e.g., in a city, to support decisions like how effective the mobility regarding the "best

routes" is, and implement advertising campaigns by companies of location based marketing such as billboards based on the density of trajectories.

As an example, *Figure 2.1* shows the basic reconstruction of a simple trajectory from a set of points based on linear interpolation [30]; however, simple linear interpolation as a method of reconstruction of low-sampling location data of users, does not represent people real movement because users move according to a determined goal or criteria.



*Figure 1.2. Reconstruction of a trajectory from a set of points*

## **1.3 OBJECTIVES**

### **1.3.1 General Objective**

Formulate an operator considering the dynamic possibilities of the reconstruction of low-sampling trajectories based on user criteria.

### **1.3.2 Specific Objectives**

1. Identify the different perspectives of reconstruction of low-sampling trajectories.
2. Develop a user criteria based operator for the reconstruction of a low-sampling trajectory.
3. Identify opportunities of analytical tasks using an operator over low-sampling trajectories considering the limitations of Network Constrained Environment.
4. Validate the effectiveness of the proposal using a functional prototype for testing.



## 1.4 SCOPE OF THE RESEARCH WORK

The theory of trajectories has a wide range of research issues. However, as presented in above cited works and the identification of the research problem we only deal with the imputation (reconstruction) of low-sampling trajectories in network constrained environments.

A criteria based operator is built to reconstruct low-sampling trajectories, i.e., an operator for computing the trajectory between two locations points when data are not present using an explicit parameter that describes the intention of the movement. This is a useful tool to approximate a low-sampling trajectory previously knowing certain kind of data as the type of movement followed by the MO and the underlying RN.

The imputation (reconstruction) of a low sampled trajectory based on criteria such distance, time, or speed and the limitations of space are the main contributions of this thesis. So, this research will not address the problem of how to know, in real-time, the location of a MO.

Some analysis tasks are also derived. The proposed operator is added in a TDW environment to show the effects of the reconstruction criteria over the low-sampling data. Visualizations and measures over the resulting trajectories are analyzed when the reconstruction criteria changes. So, other main contributions of this thesis proposal are aimed to enhance the TDW with other possibilities that hasn't been (or are been poorly) explored.

The functional prototype referred in the specific objectives is intended to show the proposed operator over low-sampling trajectories where the criteria variation can be simulated and the possibility of comparison options allow to determine the relative importance of the criteria. Those results can be developed over specific platforms such as available open-source DBMS and geo-data displayers. A set of the cases of studies are used to illustrate the effects of the proposed operator.

## **CHAPTER 2. STATE OF THE ART: “TRAJECTORY RECONSTRUCTION PROBLEM WITH LOW-SAMPLING DATA USING A CRITERIA BASED APPROACH FROM ROUTE PLANNING THEORY – A REVIEW FOR ANALYTICS”**

### **2.1 INTRODUCCION**

The evolving wireless communication systems and mobile computing technologies equipped with GPS systems have favored the exploitation of geo-positioning data [20] to meet a variety of requirements such as route finder applications and location based advertising management based applications. The way people live and move is daily recorded by those mobile devices [9] where the core information is the *movement of people over time* [31]. In a most accurate way, the movement is described when location data are ordered in time and it represents trajectories [5], [7]

Being able to choose the most convenient route to travel from one place to another is a desirable possibility when planning activities. For example, in a city the tourists usually ask for the best routes for visiting attractive places. Fields such as logistic, traffic control, and location based advertising also demand solutions in this regard to meet a variety of requirements, such as quality of road, cost of fuel, effectiveness of an advertising campaign, and user preferences, among others [6], [7], [16].

Current commercial solutions for finding best routes, e.g., Bing Maps are usually slow, inaccurate, or limited regarding route planning [12] because they are mainly focused on a single criterion: the shortest path routing. On the other hand, open source applications, e.g., Routino [32] or MapQuest [33] have incorporated specialized features such as road type (pedestrian, bicycle, car) or criteria based routing (simplest path, i.e., ease of description and execution of the path, or fastest path) for enhancing and improving the possibilities already provided by the commercial ones.

User criteria are not considered in these applications [11], [16], [34]. Several authors have recently been focused on the incorporation of user preferences and multi-criteria decision-making aspects in light of the route personalization [16]. Other approaches have used GPS data representing historical movements of users based on individual [34] or collective behavior [35]. The resulting routes are usually closer to the ones actually followed by users than those suggested by the route planners as optimal (the shortest, the fastest) [36], [37].

In this chapter, the request for a route to travel from one place to another in the route finding problem (RFP) is akin to the one of finding a trajectory between low-sampled points. Low-sampled points occur when the time interval between consecutive GPS points of some trajectories is higher than a threshold determined by the application analysts [6]. Therefore, the reviewed research works are analyzed in relation to the RFP, paying special attention to those taking into account *user criteria* or *low-sampling-rate data*. When low-sampling-rate data are present, the reconstruction of trajectories may be needed [20], i.e., the description of the movement of the object between the two points where no data points are available to know where the object is while travelling.

The need for reconstructing trajectories has a reason: It is a previous step for a better analysis of trajectory data in knowledge discovery environments [20], [21]. A great challenge for the knowledge discovery (both, of the environment where the movement occurs and the objects in consideration [5]) using spatiotemporal data [21] is demanding for techniques that enable the analysis of trajectories [38], especially the ones characterized by the uncertainty [6]. Conceptualization in analytics over trajectories is addressed in this state of art review but deeper oriented in the arising field of Trajectory Data warehouse (TDW) [26] as a way to deal with this analysis proposal.

The rest of this chapter is organized as follows. *Section 2.2* describes routing planning systems. *Section 2.3* describes personalization, i.e., incorporation of user criteria, in routing planning systems. *Section 2.4* addresses personalized route finding based on the concept of trajectories but focusing in the reconstruction of trajectories under low-sampling-rate data. *Section 2.5* describe de problem of uncertainty of trajectories. *Section 2.6* addresses the related works and methods regarding analytical task over trajectories. *Section 2.7* concludes the chapter establishing the relationship between the personalized route planning with the reconstruction of low-sampling trajectories proposing future works, one of them addressed in the following chapters.

## **2.2 ROUTING PLANNING SYSTEMS**

Routing (or Route) planning systems RPS are commonly recognized as decision support systems [39], [40]. These systems sometimes are referred to as geo-related decision support tools [15]. In *Table 2.1*, some variations of the term referring to RPS found in the literature review are presented. Conventional solutions provided to RFP are limited because they use an analysis based on just one dimension (criterion): the cost [41], [42], [43], [44].

<b>Author</b>	<b>Term</b>	<b>Definition</b>
[11]	Routing systems	Routing systems aim to help users on finding the optimal path to their destination regarding travel distance, travel time, among other criteria.
[15]	Route planning technique	A route planning technique is an essential geo-related decision support tool in a GIS (Geographic Information System) whose goal is the accurate route finding.
[15]	Personalized user-centric route finding	A personalized user-centric route finding application incorporates user preferences and the environmental features around a user. User preferences and environmental features are the key elements to assess a route.
[16]	Personalized route planning systems	A personalized route planning system provides a route based on minimizing a combination of user defined criteria such as travel distance, travel time, the number of traffic lights, and road types.
[41]	Route guidance systems	Route guidance systems refer to all driver decision factors considered before and during a trip to choose a route, as well as unexpected factors that may happen during the trip to adjust the route. Route guidance systems are recognized as a fundamental component of intelligent transportation systems.

*Table 2.1. Common terms referring to routing planning systems*

Many definitions include, explicitly or implicitly the notion of personalization, suggesting that user interaction is required. Recent researches have been carried out to improve these models through their personalization and the incorporation of multi-criteria decision-making including preference models [11], [16], [39], [45]. Indeed, the personalization of route finding by the incorporation of user criteria is one of the most desired features in RPS [39]. A brief schema review of the RFP in RPS is shown in *Figure 2.1*. The RPS are supported by Routing Planning Algorithms. When the personalization is included, incorporating preferences or decision strategies originates the concept of Personalized Routing Planning Systems.

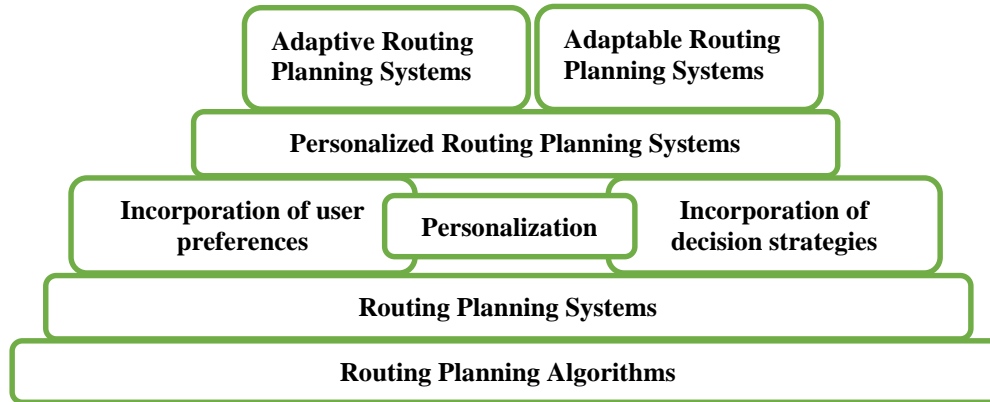


Figure 2.1. Schema review of the RFP according to personalization in RPS

Early approaches to the RFP focused on the cost of the path represented by the distance between two points. The classical algorithm for RFP based on the shortest path issue was proposed by Dijkstra [14] and it has been used widely in many applications to find the shortest path between an origin vertex and a destination vertex in a weighted graph exploring the entire graph to determine the lowest cost route. Similarly, the A\* algorithm (a modification of Dijkstra’s algorithm) finds the optimal path using an appropriate heuristic (that avoids exploring the entire graph) that defines which is the best node to be visited next based on the lowest heuristic cost [46], e.g., some Minkowski metrics [47]. The general Minkowski distance  $d_{ij}$  of order  $p$  between two points  $(x_i, y_i)$  and  $(x_j, y_j)$  in a two-dimensional space is  $d_{ij} = \{(|x_i - x_j|^p + |y_i - y_j|^p)\}^{1/p}$ . Minkowski distance is typically used with  $p$  being 1 or 2. When  $p = 1$ , it is called Manhattan distance, when  $p = 2$  is called Euclidean distance. All these early approaches are based on algorithms that use an *edge cost*, i.e., they performed a one-dimensional analysis. For this reason, these algorithms are inadequate or incomplete since users generally have different purposes and they do not share the same preferences of movement behavior, highlighting the need to *personalize* and allow the user to interact with RPS.

### 2.3 PERSONALIZATION

The personalization is a term widely used in many fields. The technology-based definition provided by the Personalization Consortium (2005) is “the use of the technology and customer information to tailor electronic commerce interactions between a business and each individual customer”.

An experiment conducted by Golledge[48] showed that the criteria used by humans to deal with path selection problems may be a complex task that covers a wide spectrum of choices. The route choice behavior based on route selection was analyzed in a real environment and in a laboratory. The routes were determined using criteria selection such as shortest distance and fewest turns. Variables such

as orientation and the possibility of retracing the route (i.e., interchange the origin and the destination) were also studied to determine the change of the user route criteria selection when traveling in one direction or the other. This set of exercises provides evidence that route selection is not a simple process that can be solved by traditional algorithms. Instead, it shows that it is a process that requires the support of decision strategies and preference models to back personalization. Indeed, their experiment showed that users not always choose the shortest route.

To illustrate the above problem, *Figure 2.2* represents a simple example of RFP in a RN. Two possible routes between an origin O and a destination D are shown. The route O-C-D is usually suggested by common RPS without considering the probability of a traffic jam or local restrictions for moving between streets. However, most users would select the route O-A-B-D even though path O-C-D has the minimum distance because more points of interest (POI) can be found along it (supermarkets, parks, or gasoline stations). This is already evidenced by Duckham and Kulik [49], showing how a *simple path* solution offers considerable advantages over shortest paths in terms of their ease of description and execution. Several researchers have stated the importance of the personalization when solving routing planning tasks [16], [34], [40].

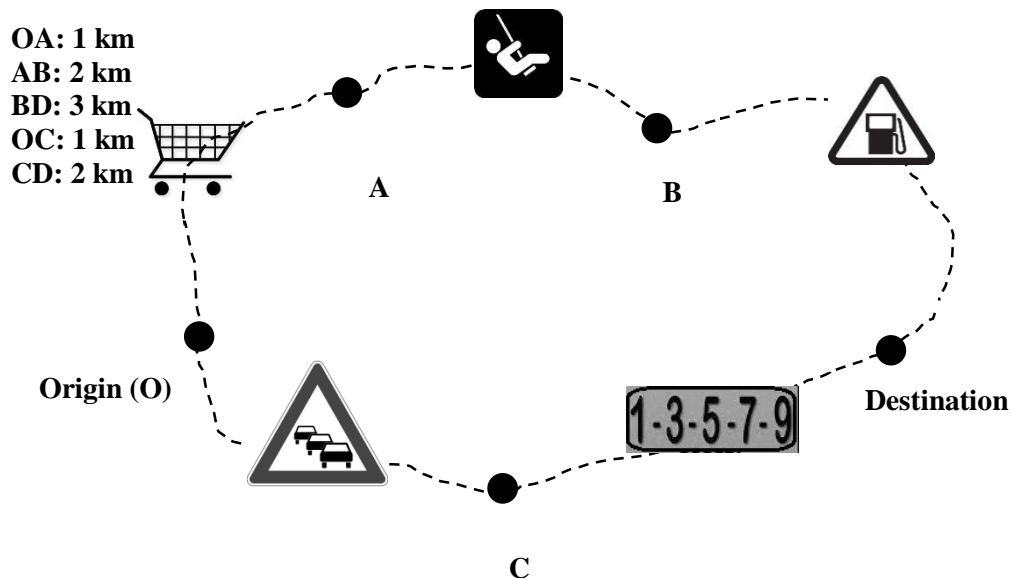


Figure 2.2. Problem of route finding in a road network<sup>1</sup>

The goal of personalization is the automatic adaption of an information service in response to the implicit or explicit needs of a specific user [40]. That is, automatic identification of preferences from

<sup>1</sup> All image [usage rights](#) are labeled for use with modification.

the user movement behavior history [36], [37] or explicit requests of the user [15], [16]. Also, Fischer [50] stated that personalization can be described by adaptable and adaptive methods, and Oppermann [51] gives the following definition: in adaptable systems the user controls the adaptation process whereas in adaptive systems the process is automatic, i.e., without user intervention. Nadi and Delavar[16] define adaptable and adaptive personalized route guidance systems in the context of RPS. Examples of adaptable [15], [16] and adaptive [52], [53], [54], [55], [56] RPS can be widely found in the literature.

In [15], static and dynamic systems; deterministic and stochastic systems; reactive and predictive systems; and centralized and decentralized systems are distinguished. In [41], descriptive and prescriptive guidance; and static and dynamic guidance are reviewed. In [42], route guidance systems are classified as infrastructure-based and infrastructure-less systems. Infrastructure-based systems are based on two components: i) hardware devices deployed in streets/roads and ii) computer systems installed in moving objects (e.g., a GPS). Infrastructure-less systems require only the second component. Personalization can also be defined in terms of user route choice criteria. Typical route algorithms are optimized regarding only one criterion [57], e.g., route length or travel time (i.e., a one-dimensional analysis). A special issue of the personalization in RPS is the characterization and incorporation of several criteria. *Table 2.2* shows some of them, classified as quantitative (they are measured from a map or any other source) and qualitative (they are no-numeric criteria that are ranked according to the impact on the user).

<b>Author</b>	<b>Criteria</b>	<b>Quantitative</b>	<b>Qualitative</b>
[11],[16]	Distance, Travel Time	x	
[11], [13], [37]	Traffic	x	
[11], [16], [49]	Costs of Turns/ Simplest Paths	x	
[58], [59]	Number of Scenic Landscapes / POIs	x	
[16]	Number of Junctions, Travel Reliability, Directness, Road Width, Number of Stop Signs	x	
[16]	Quality of Road, Type of Road		x

*Table 2.2. Quantitative and qualitative criteria of RFP*

Previous research [11], [13], [16] found that route selection criteria can be grouped into four general criteria: speed (time, distance), safeness, simplicity, and attractiveness (scenic path POIs-based):

### **2.3.1 Speed: Distance**

Distance is normally considered the most important criterion for route choosing. Even without route planning systems, the path with the shortest distance is intuitively chosen with a minimum previous knowledge of the RN structure; however, the presence of known POIs may lengthen the road trip. See *attractiveness*.

### **2.3.2 Speed: Time**

Time is a variable that depends of several factors such as length (the time is directly proportional to the length of road), average speed (higher in main avenues than in small streets), and quality of roads, weather conditions (e.g. when it rains, travel time is higher due to traffic conditions derived from it) or quality of traffic as described in [11].

### **2.3.3 Safeness**

It groups a series of criteria based on characteristics (bike lane availability, area safeness, nighting, traffic level), possibilities (lack of busy intersections, public transport, roundabouts), and features of the road (presence or lack of pavement, slope angle) [13].

### **2.3.4 Simplicity**

The simplest path is based on the idea that the turns imply reductions of velocity and unnecessary maneuvers. Thus, the path is “better” if it has less turns [11]. Moreover, the description of the path is easier when a simplest path approach is followed, as the explanation, depiction, understanding, memorizing, or execution of it [49], which is useful for users who are navigating through an unfamiliar geographic environment.



### 2.3.5 Attractiveness

Criteria such as distance, time, or turns are common route criteria for navigating a street network, but computation of the most scenic route is a special issue [60]. The scenic path notion is defined from the touristic perspective. The main idea is to travel from A to B trying to visit as much touristic places as possible and minimizing route length at the same time. The cost is related with the number of touristic attractions between the two points. A previous step for modify the cost of the edges must be done (for instance, the streets with a considerable number of POIs have the lowest cost) before a shortest path algorithm is executed if the goal is to find a route that traverses as much POIs as possible, and at the same time, the shortest route between two POIs.

*Figure 2.3* (Previously shown as a Motivating example) exhibits a section of Guarne, a small town in Colombia, with a route between two points using the shortest path algorithm. *Figure 2.3-a* shows the minimum distance between point A and B. *Figure 2.3-b* shows the route with the minimum travel time between point A and point B. *Figure 2.3-c* shows the route between the two former points using the simplest path approach. The turns in the path are less, even though the whole path may be longer. *Figure 2.3-d* shows the route between the two points using the scenic path approach: the route is draw along the street nearest to the town river where touristic attractions are (restaurants, beach games, etc.)

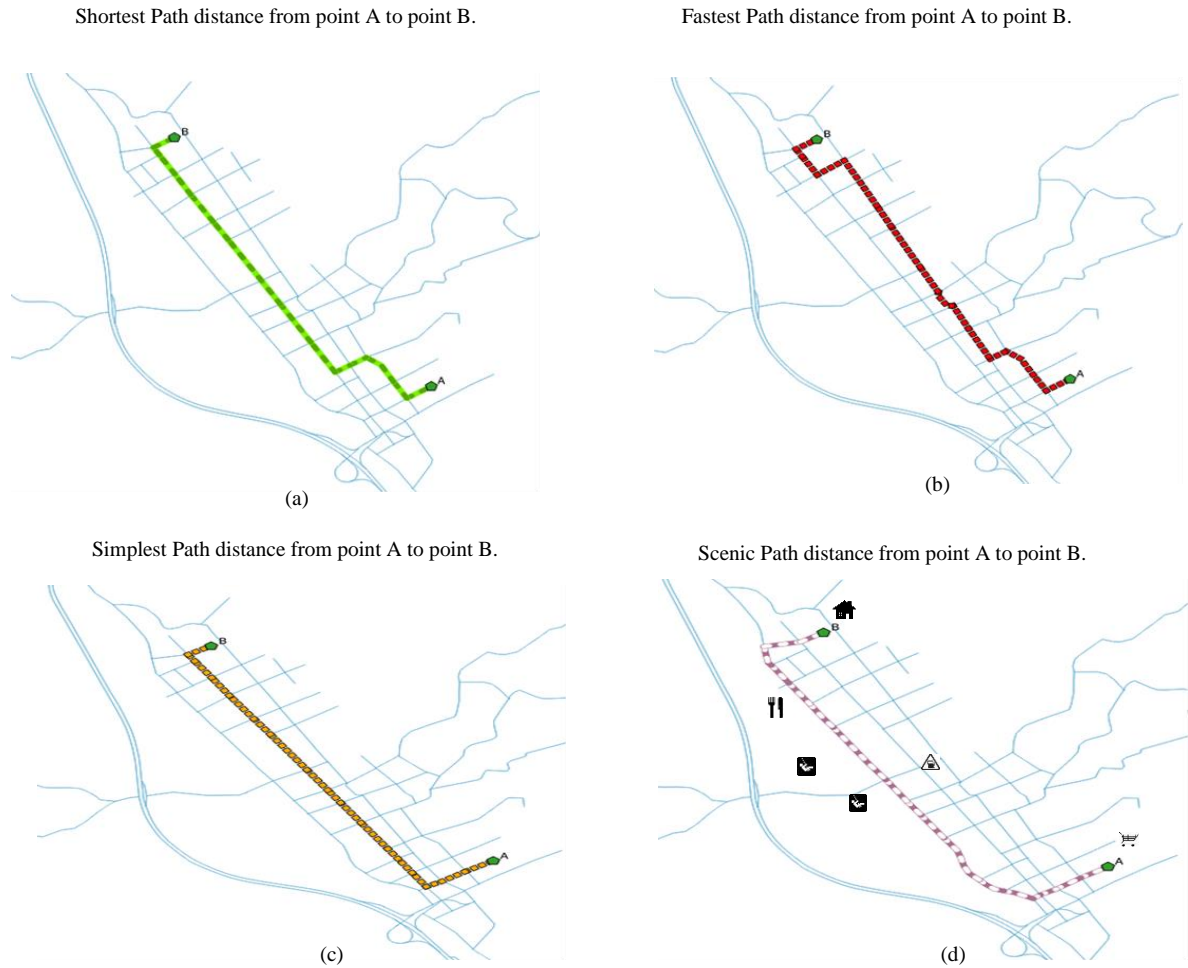


Figure 2.3. Different route finding criteria from point A to point B: (a) Shortest Path distance from point A to point B; (b) Fastest Path distance from point A to point B; (c) Simplest Path distance from point A to point B; (d) Scenic Path distance from point A to point B

## 2.4 PERSONALIZED ROUTE FINDING BASED ON TRAJECTORIES

The RFP reviewed here is related with the problem of reconstruction of trajectories, i.e., the problem of tracing a route that pass by a set of locations. Pattern-based and greedy searches approaches has been considered to solve this problem (Preference-based Greedy search, NaïVe Greedy search, Pattern+Greedy search) [61]. Pattern-based approaches allow the *offline* processing of historical trajectory data to discover mining patterns to infer routing information [6], while greedy search approaches make optimal local choices at every decision stage providing a dynamic/*online* recommendation on the best immediate location to be visited for constructing the route, instead of preprocessing historical data [61]. The most of these works deal with a general mining/prediction

problem over historical trajectories [35], [37], [61], [62]. The personalization aspect in the reviewed works is based on the trajectory history data of a particular user. Thus, these could be considered as adaptive approaches.

In [63], the problem of searching the *k-Best Connected Trajectories (k-BCT)* is addressed. A small set of locations (queried points) is given as an input to an incremental k-NN (K-Nearest Neighbor) based algorithm, which progressively retrieves trajectories nearest to each location, using best-first and depth-first k-NN algorithms. The quality of the connection between locations provided by the discovered trajectories is given by a similarity measure which determines how well a trajectory connects to the locations. A dataset of Beijing collected by the Microsoft GeoLife Project was used to analyze the efficiency of the IKNN algorithm, showing a better search performance if the best-first k-NN algorithm is chosen.

In [35], the problem of discovering the *most popular* route between two given locations using historical user trajectories is addressed. A *Coherence Expanding Algorithm* is proposed for mining users' movements together with a popularity indicator. Then, an algorithm for searching the most popular route given two locations is applied. Considering 276 truck trajectories used in Athens and applying the proposed algorithm, the most popular routes were identified. Then, these findings were compared against those obtained with the shortest path approach.

In [34], a *Pattern-aware Personalized routing framework (PPT)* is proposed using a two-step method to compute personalized routes. First, a set of frequent road segments are derived from a user's historical trajectories database to construct a familiar RN followed by a specific user. Then, while a route is computed between a specific source and a destination, a second algorithm is proposed to discover the top-k personalized routes connecting some segments that a user has previously traveled. The algorithms were tested using a real trajectory dataset from one user over a period of four months in Kaohsiung, Taiwan. The proposed algorithms derive the top-k personalized routes that approximate the real top-k personalized routes.

In [37], smart driving directions are mined from taxi drivers experience. They propose a routing algorithm to provide the fastest route from a given origin to a given destination. Thus, a time-dependent graph is built where nodes are recognized as landmarks, i.e., road segments traversed by a significant number of taxis and edges represent taxi routes between landmark roads. The method

is compared with speed-constraint and real time traffic-based methods. This demonstrates that about 16% of time can be saved with this method.

In [36], fast routes are also mined from taxi traces and are customized for a particular driver behavior. A mobile client device learns a user's driving behavior from the user's driving routes and finds the fastest route for the user. This model outperformed the previous work [37].

In [61], the construction of a preferred route using location check-in data are done based on the popularity of a certain route and the preferences ranked by a set of users. The goal is to build a trajectory where the reconstruction meets the preferred locations to be visited by a *group of persons* using Gowalla check-in data and a Pattern+Greedy method (this combination of Pattern and Greedy route search outperforms both methods when used separately). Similarly, in [62], the top-k Trajectories are extracted from interesting regions with higher scores (attractiveness) mined from historical GPS trajectories. A Framework for trajectory search is developed called Pattern-Aware Trajectory Search (PATS) which includes an off-line pattern discovery module and an online pattern-aware trajectory search module. This framework only searches for the top-k maximal trajectories with higher scores according to the number of interesting regions and does not infer new routes.

## 2.5 UNCERTAINTY IN TRAJECTORIES

Most of the former research works do not deal with trajectory uncertainty explicitly. When reconstructing a trajectory, it is also necessary consider basic characteristics of trajectories such as its *low-sampling-rate* to deal with the uncertainty of low-frequency data [6], [7]. Previous works [34], [35], [37], [63] relied on high-sampled trajectories. The effectiveness of inferred routes is poor due to its inadequate management of low-sampling trajectories where uncertainty is reflected.

The causes for *low-sampling* trajectories include: the lack of users sharing their position or taking geo-tagged photos from every place and every second. This is due to the privacy concerns publishing personal location data to potentially untrustworthy service providers may pose [64]. Research works has been carried out to preserve publishing data of a moving object to a third party for data analysis purposes because it could have serious privacy concerns [8], [65], [66]. Privacy-preserving techniques has been studied based on false location [67], space transformations [68] or spatial cloaking, i.e., the individual's location according to the number of individuals within the same quadrant [69]. However, those works are not aimed to reduce low-sampling directly. Instead, they

provide privacy – preserving techniques to *promote* location sharing information. The uncertainty of trajectories has been studied intensively [7], [70], [71], [72]. The main features of the trajectories regarding to uncertainty are highlighted in [10]:

1. **Spatial Biases:** The locations of data points in two trajectories are different, i.e., two similar trajectories can be depicted by means of different location data points.
2. **Temporal Biases:** The occurrence time of two trajectories are different, i.e., two similar trajectories visiting the same POIs could be done in two different periods.
3. **Silent Durations:** The periods when no data points are available to describe the movements of the users.

Relevant data are usually missing during silent durations. User movement criteria can fulfill partially those silent durations. For the best of our knowledge, the low-sampling-rate trajectory reconstruction problem has not considered the user preferences. We strongly believe this is a rich research area with application in several domains. For example, for location-based advertising, it might mean the possibility of advertising strategies based on data about routes followed by the users from a POI A to a POI B.

In [73], uncertainty from different sources is evidenced: i) GPS observations (accuracy of the GPS observation) and ii) the uncertainty derived between low sampled points of a trajectory. Those are also referred to the *measurement* and the *sampling* errors [70]. The first is addressed by map matching techniques and the quality of the measurement depends largely on the technique used. The second one uses notions such as space-time prisms which delimit the movement based on some background knowledge, like, for instance, a speed limit or a RN. The second one is the case of mobile social network applications enriched with geo-tagged media information where low-sampling data are common.

Several studies [61], [74], [75] infer routes from a sequence of POIs but a detailed route between two consecutive POIs is not specified. The underlying assumptions of these works are that the user movement is free. However, the infrastructure, e.g., buildings, streets direction, may be considered to obtain a reduced overall uncertainty and inaccuracy in the data.

In [7], a Route Inference framework based on Collective Knowledge (RICK) is developed. Given a set of locations and a time span, a two-step method is followed: first, a “routable graph” is built and,

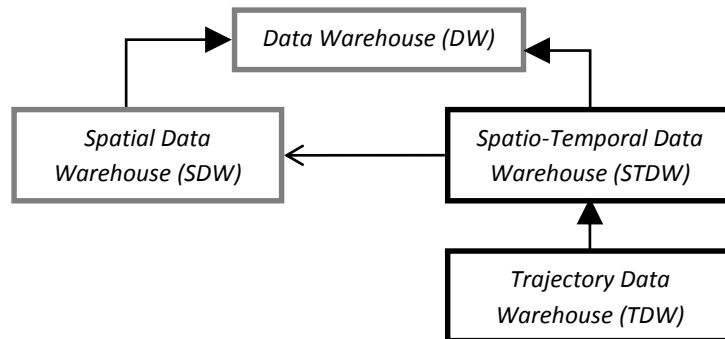
then, the top-k routes according to the route inference algorithm are constructed. Two real dataset are used: registers of Foursquare check-in application used in Manhattan city and trajectories used in Beijing. The main is to demonstrate the effectiveness and efficiency of RICK.

In [6], the problem of reducing uncertainty for a given low-sampling-rate trajectory is addressed. Historical data are used to discover popular routes as an estimation of low-sampling trajectories. A real trajectory dataset generated by taxis in Beijing in a period of three months is used to validate the effectiveness of their proposal and shows higher accuracy in comparison with the existing map-matching [76], [77].

## 2.6 THE DW FOR TRAJECTORIES

The transformation of raw trajectories into valuable information is a requirement that can be used for decision-making purposes [27]. This is the mean reason for low-sampling trajectory imputation process, i.e., the reconstruction, addressed in the current state of the art: completing the low-sampling trajectories for knowledge extraction and analysis tasks [20].

There are a variety of techniques in the field of knowledge discovery to extract valuable information from spatiotemporal data [21] adapted from the traditional ones [78], [79] (e.g., data mining with clustering, classification, and regression techniques). However, this thesis only addresses the ones based on DW concepts [22]. Therefore, the basic concepts of DW are outlined but the analysis and conceptualization are oriented to the SpatioTemporal Data Warehouse (STDW) especially, in the arising field of Trajectory Data warehouse (TDW), see *Figure 2.4*.



*Figure 2.4. Taxonomy of Data Warehouse (DW)*

Inmon [80] was the first who defined the concept of *DW* as a subject oriented, integrated, time variant and non-volatile collection of data in support of management's decision-making process. However, there are two main approaches in the design of a *DW*: the Inmon approach and the Kimball approach [22]. The Inmon approach states for the integration of a centralized place where to store the information to support analysis tasks. This is known as the *top-down* approach because having a centralized *DW*, the analytical need of the business units can be supplied using subsets of the centralized *DW* (later called by Kimball [81] as data marts).

### **2.6.1 Multidimensional Modeling**

The *DW* is modeled in a *multidimensional* way (according to Kimball structure) to facilitate a complex analysis. The multidimensional modeling starts with the factors that affect the decision-making process in the specific area of business called measures of interest, such as the number of sales in a store. This information is analyzed using diverse perspectives called *dimensions*, which in turn, are organized in *hierarchies* on which aggregations are performed. For instance, the sales can be analyzed by date and product. The product can be organized hierarchically by type and brand; and the date can be analyzed by year, semester, and month.

### **2.6.2 Spatial Data Warehouse (SDW) and Spatiotemporal Data warehouse (STDW)**

The growing popularity of spatial information generated from satellites and materialized in maps, has opened up the *SDW* as an interesting topic of research [82]. *SDW* are based on the concepts of *DW* presented above and the combination of spatiality which provide some characteristics to aggregate, analyze, and visualize spatial data based on spatial dimension with levels represented by geometries [83].

While *SDW* considers types and dimensions adding the spatial context, the considerations of the temporal aspects are also needed to analyze events like the movement of entities. [84] Offer a sight of *STDW* as a relationship between GIS systems and concepts of *DW* (facts and dimensions) highlighting the time to form spatiotemporal databases.

### 2.6.3 Trajectory Data Warehouse (TDW)

A special case of the spatiotemporal domain of STDW is related to the integration of the movement described by MO, i.e., trajectories, in a TDW. The main goal of a TDW is to transform raw data of trajectories into valuable information to support decision-making process in applications based on MO [27].

In [26], a DW is proposed to deal with the issue of huge data generation of MO by mobile applications. They focus on concepts related to trajectories to support tasks that involve data generated by modern devices like GPS and other huge amount of spatiotemporal data to support Knowledge Discovering Tasks (KDD) [85]

In [86], a TDW proposal is also provided for analyzing mobility data that takes into consideration the complete flow of tasks required for the development of a TDW and the application of trajectory-inspired mining algorithms to extract traffic patterns. The trajectory reconstruction problem is also included in a module using parameters such as temporal and spatial gap between trajectories, maximum speed, and tolerance distance.

### 2.6.4 The analysis goals in a TDW

The measures about trajectories have characteristics to be analyzed in a TDW. Pelekis and Raffaeta [27] distinguish some of them:

- 1- *Numeric Characteristics*: such as Average of the speed, direction, and duration of the trajectory.
- 2- *Spatial Characteristics*: such as the geometric shape of the trajectory.
- 3- *Temporal Characteristics*: such as the timing of the movement.

With regarding to the spatial characteristics of trajectories, Pelekis and Raffaeta [27] stated that most of the proposals [87], [88] distinguish three types of spatial dimensions about the incorporation of spatiality on members levels: non-geometric, which uses nominal spatial references (e.g. name of streets and cities); geometric-to-non-geometric, which at lower levels member has an associated geometry up to a certain level member where it becomes non-geometrics, i.e., it becomes nominal; and, fully geometric where all levels have an associated geometry. However, [89] stated that a dimension can be fully spatial even if some members' levels do not have an associated geometry.



The handling of geography also could include a simple grid, a RN or even coverage of the space with respect to the mobile cell network [90].

There is still an open research issue regarding to TDW operations for enhancing traditional ones. Pelekis and Raffaeta [27] prospect some of them, such as: trajectory clustering, extraction of a representative trajectory from a set of trajectories and operators to propagate/aggregate the uncertainty and imprecision present in the data. This thesis suggests the analysis of some measures based on a criteria based imputation process over low sampling trajectories to deal with the uncertainty and explore the possibilities of analysis over those reconstructed trajectories.

## **2.7 CONCLUSIONS AND FUTURE WORK**

The trajectory reconstruction problem is still an open research issue, especially what is related to uncertainty due to low-sampling data and the incorporation of user preferences. Simple linear interpolation [30], as a method of reconstruction of low-sampling location data, does not represent user real movement because they move according to a certain criteria such as time or the amount of touristic/scenic places. Indeed, the reconstruction of trajectories using user preferences is expressed as a need in recent research works [10], [28], [91].

To the best of our knowledge, there are no research work that involve several criteria as a way to reconstructing low-sampling trajectories. This approach can also be enhanced considering the restriction of the movement in a RN [19], [92] and methods to predict the location of moving objects in a RN [93]. Location data are low-sampling because people do not share data in a high rate due to security and privacy issues [8], energy saving, communication problems, or it is only an action done one time when a user arrives to a POI [64]. Again, considering user criteria to infer movement between consecutive points of a trajectory to deal with low-sampling issues is a task that deserves to be explored.

On the other hand, the current availability of GPS loggers gathered from mobile devices are useful in a variety of ways to make driving better [20], but effective usage of the huge amount of data generated by GPS is still a challenge [94]. Considering the different possibilities of user criteria reconstruction of trajectory and the huge amount of low-sampling data, data analysis tasks related to these possibilities of reconstruction can be conducted using e.g., TDW approaches. Therefore, analytic results over reconstructed trajectories can vary if different criteria of reconstruction are used.

For example, if a trajectory is reconstructed based on the criterion of minimize turns, the main avenues can be interesting for analysis tasks because those are the longest without less deviations but if the amount of POIs are used as a criterion of reconstruction, then the avenues nearest to tourist attractions might be the interesting ones.

**The main contributions of this chapter are:**

- The characterization of the route finding problem through the route planning systems.
- The characterization of user criteria in the route finding problem as a personalization feature.
- The establishment of the relationship between the problem of personalized route planning and the reconstruction of low-sampling trajectories.
- The characterization of the current state of the treatment of uncertainty in trajectories
- The establishment of the treatment of spatiotemporal data, especially the trajectories, in the data warehouse theory.
- This chapter develops the specific objective “*Identify the different perspectives of reconstruction of low-sampling trajectories*” from the route planning theory.

## **CHAPTER 3. LOW – SAMPLING TRAJECTORY RECONSTRUCTION USING CRITERIA BASED ROUTING OVER A GRAPH**

### **3.1 INTRODUCTION**

Due to the fast development of technologies and mobile applications, the need of analyzing the huge amount of geo-location data recorded regarding moving objects (MO) has arisen. For example, users in mobile social networks such as Foursquare and Flickr use the options of checking-in and sharing geo-tagged photos to indicate their location. However, usually it is not possible to get detailed data about the movement of a user due to privacy issues [64], energy saving, or simply because people do not share the position (make check-in) in every place where they are or do not take a geo-tagged photo every second. Each of these situations deals with movement uncertainty.

As a consequence, source (raw) trajectory data have a lot of uncertainty because data are not very accurate since there are missing data during the silent durations, i.e., the time durations when no data are available to describe the movement of an object [10]. Thus, the trajectory between two consecutive data records is uncertain. As a result, the following are some possible questions to be addressed: How does a MO moved during a silent duration? How well do the current methods describe the real trajectory of a MO? Is a MO moving according to a certain criterion?

Previous works have focused on trajectory history (a trajectory dataset of the same MO [34] or GPS historical data provided by several MOs [35]) as a way of inferring the routes or the movement patterns based on the density of the data. For trajectory reconstruction (i.e., the imputation process for silent durations) some authors [7], [72] use an uncertainty reinforcement approach (i.e., uncertain + uncertain  $\rightarrow$  certain). However, these approaches may be inadequate if the silent durations in the trajectories of a same MO are relatively large and recurrent (i.e., there are recurrent trajectory segments where no data are present).

The management of uncertainty for low-sampling data is a hard task to tackle. To facilitate this task, the trajectory reconstruction can rely on user preferences (a criterion) such as (minimize) distance or (visit) touristic places to try to fill those silent durations. As expressed in the *Chapter 2* of the state-of-art review, the request for a route to travel from one place to another in the route finding problem (RFP) is akin to the one of finding a trajectory between low-sampled points. The claim of this thesis is that the movement of an object based on user preferences would generate some clues

which may help in the trajectory reconstruction [10]. To the best of our knowledge, user preferences have not considered in the low sampling rate trajectory reconstruction problem.

The problem of trajectory reconstruction is usually addressed from describing the trajectory by a set of GPS points temporally ordered [26], [34]. However, most route planners do not consider the time dimension, i.e., they generate a sequence of geo-referenced data points *without timestamps*.

The trajectories considered here are network-constrained, i.e., it is assumed that the movements of the objects are restricted to the road networks (RN) of the cities. Thus, the trajectory reconstruction between two consecutive check-in records is limited to the geometry of the streets. This reduces the search space and the reconstruction possibilities according to a certain criterion in favor of reduction of trajectory uncertainty because the MO cannot move further than the network (streets) limits.

The route among check-in data of a trajectory is built by filling in the check-in order sequence (which represents the raw trajectory) with additional geo-referenced data points and timestamps. To help in this task, a graph, inferred from the RN is built, where the vertices save geo-related information and the edges describe the cost for reaching two vertices [29]. The routing algorithms rely on this representation to build the trajectory between two points to facilitate computational efficiency. This representation is used for the imputation process.

The rest of this chapter is organized as follows. *Section 3.2* describes the trajectories model representation followed in this proposal. *Section 3.3* discusses the reconstruction of trajectories using a formal approach. *Section 3.4* describes the proposed function and the algorithm used for reconstructing trajectories giving an application example and comparing results with the original datasets. *Section 3.5* concludes the chapter establishing the operator possibilities and proposing future works, one of them addressed in the following chapter.

### 3.2 A REPRESENTATION FOR TRAJECTORIES

Several models for trajectories have been proposed in the literature [3], [26], [34]. Most of them (except for [3]) agree in the representation of a trajectory as a set of geo-referenced points temporally ordered.

According to [26], a trajectory is a pair  $T_i = (ID_i, L_i)$  where  $ID_i$  is the unique identification of the MO and  $L_i$  is a sequence of  $M$  observations  $= \{L_i^1, L_i^2, \dots, L_i^M\}$ . Each observation  $L_i^j = (x_i^j, y_i^j, t_i^j)$  represents the presence of an object at location  $(x_i^j, y_i^j)$  where  $x_i^j, y_i^j \in \mathbb{R}$ , and at time  $t_i^j \in \mathbb{T}$ , where  $\mathbb{T}$  is a set of time points. The sequence of observations  $L_i$  is temporally ordered, i.e.,  $t_i^j < t_i^{j+1}$ . A sampling of 2D trajectories is defined as  $\mathcal{TS} = \{T_i\}$ . Note that  $L_i \in 2^L$ , where  $L$  is the set of all possible observations.

### 3.3 ADDING TRAJECTORIES CHARACTERISTICS TO ROUTES – A FORMAL APPROACH

*HELP: First, the way some index notations are used is shown:*

*Index  $i$  is used to identify a given moving object  $ID_i$  until  $n$  moving objects.*

*Index  $j$  is used to identify the sequence of observations of a given moving object  $ID_i$  until  $m$  observations.*

*Index  $k$  is used to identify the sequence of vertices obtained from certain criterion  $c$  until  $p$  vertices.*

*Index  $l$  is used to identify una determinada criterio  $C_l$  va hasta  $q$  criterios.*

Given a trajectory  $T_i$  of a MO where some points may be separated spatially or temporally in such a way that they exceed a given application threshold, our goal is to infer the sub-trajectories based on a set of reconstruction criteria from the personalized route planning theory, which in turn, is based on graph theory, i.e., we use a set of criteria widely studied in the literature [11], [13], [16] such as time and distance to reconstruct trajectories using graph modelling. Those criteria are represented by the set  $Cset$ .

Consider the network-constrained trajectories  $(\mathcal{TS}, \text{Ga})$ , where  $\mathcal{TS}$  is a set of trajectories and  $\text{Ga} \in \mathcal{G}$  ( $\mathcal{G}$  is the set of graphs) is a *directed* and *labeled* graph representing the underlying constrained RN where the set of trajectories is constrained. The graph  $\text{Ga}$  is a two-tuple  $\text{Ga} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V}$  is a set of vertices  $\{v_i\}$  and  $\mathcal{E}$  is a set of edges  $\{e_k\}$  (representing the segments of the streets). An edge  $e_k$  has a *source vertex* (the initial part of an edge), which is denoted by  $v_{k,s}$ , a *target vertex* (the end part of an edge) denoted by  $v_{k,t}$  (the edge  $e_k$  is traversed from the  $v_{k,s}$  to the  $v_{k,t}$ , but not the other way around), and an associated cost for traversing it denoted by  $c_k \in \mathbb{R}$ , i.e., an edge is a tuple  $e_k = (v_{k,s}, v_{k,t}, c_k)$ . Each vertex  $v \in \mathcal{V}$  can be described by a location  $x, y$  (longitude, latitude). Note that we consider the graph  $\text{Ga}$ , which is derived from a RN, to be fully connected and without any isolated network segments.

Consider the following functions:

*get\_vertex\_source*:  $\mathcal{E} \rightarrow \mathcal{V}$ . Function applied to an edge to get its source vertex.

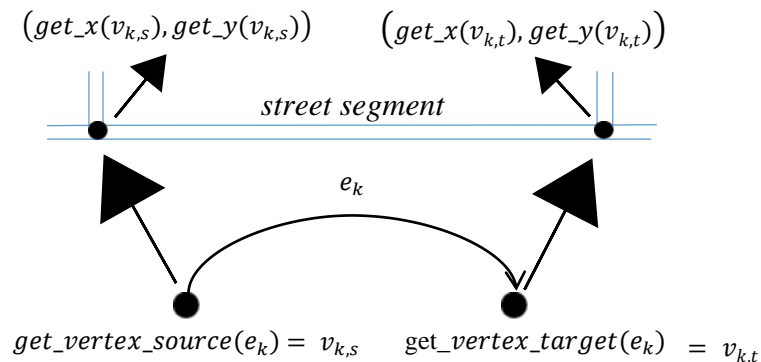
*get\_vertex\_target*:  $\mathcal{E} \rightarrow \mathcal{V}$ . Function applied to an edge to get its target vertex.

*get\_cost*:  $\mathcal{E} \rightarrow \mathbb{R}$ . Function applied to an edge to get the cost of traversing the edge.

*get\_x*:  $\mathcal{V} \rightarrow \mathbb{X}$ . Function applied to a vertex to get its longitude.

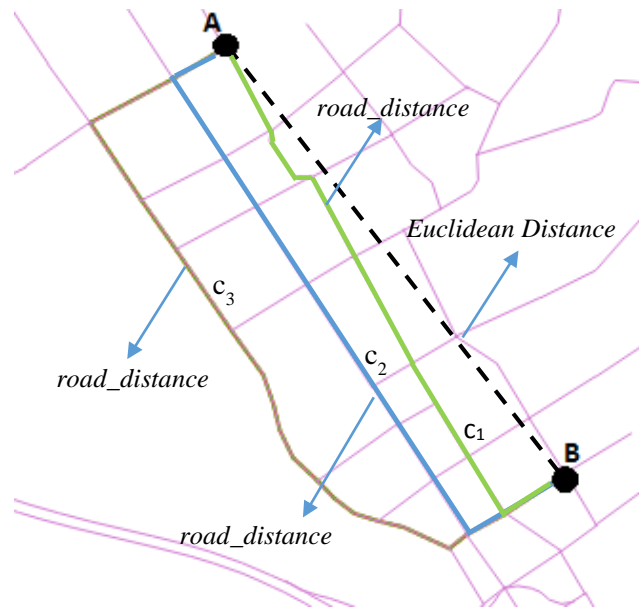
*get\_y*:  $\mathcal{V} \rightarrow \mathbb{Y}$ . Function applied to a vertex to get its latitude.

In *Figure 3.1*, some of these functions are illustrated.



*Figure 3.1. Some components of a RN*

The function  $road\_distance: L \times L \times Cset \rightarrow \mathbb{R}$  receives a pair of consecutive observations and a  $c$  criterion movements and generates the road distance between them according to the given criterion. The road distance refers to the distance of a particular path followed by a MO between two observations. In this case, it depends on the underlying RN and on the  $c$  criterion preferred by the MO (i.e., the user); therefore, the road distance may change varies when the criterion of movement change. *Figure 3.2* shows the possible roads (depicted in solid lines) between observations A and B according to some criterion. The  $c_1$  criterion used in the road drawn in green line has the lowest road distance, followed by the road distance from the road drawn in blue line using the  $c_2$  criterion. Finally, the road distance is the longest when the  $c_3$  criterion is preferred:  $road\_distance(A, B, c_1) \leq road\_distance(A, B, c_2) \leq road\_distance(A, B, c_3)$ . Note how the distance between these observations changes according to the movement criterion and the RN that were used. And also that the Euclidean distance, depicted in a dashed line, does not correspond to the road distance in any of the three cases.



*Figure 3.2. The concept of road distance according to different criteria vs the Euclidean distance between two observations A and B*

$$road\_distance(A, B, c_1) \leq road\_distance(A, B, c_2) \leq road\_distance(A, B, c_3)$$

We regard the trajectory  $T_i$  as low-sampled if  $\exists j, 1 \leq j \leq M, \left( road\_distance \left( L_i^j, L_i^{j+1}, c \right) \geq \beta \wedge t_i^{j+1} - t_i^j \geq \tau \right)$ , i.e., the road distance according to a  $c$  criterion between two consecutive observations is greater than  $\beta$  (a distance threshold) and their time difference  $(t_i^{j+1} - t_i^j)$  is greater than  $\tau$  (a user time threshold).

We consider the  $traj(L_i, c)$  function where  $L_i \in 2^L$ , is the sequence of  $M$  observations of a trajectory  $T_i$  and  $c \in Cset$  is a reconstruction criterion. The result of the  $traj$  function is a more detailed sequence of observations  $L'_i$  so that the thresholds  $\beta$  and  $\tau$  are met  $\forall j, 1 \leq j < M$ . The idea behind the trajectory reconstruction function is to fill in the trajectory with inferred observations between  $L_i^j$  and  $L_i^{j+1}$  ( $\forall j, 1 \leq j < M$ , where both thresholds  $\beta$  and  $\tau$  are not met) considering the criterion  $c \in Cset$ . Next, we explain the effect of the  $traj$  function over a pair of observations  $L_i^j$  and  $L_i^{j+1}$  (where thresholds  $\beta$  and  $\tau$  are not met) to show how the sequence of low sampling data is filled in (imputation process). Note that when a section of a trajectory is not considered low sampling, the imputation process adds this section to the whole reconstructed trajectory without imputing additional observations.

As presented by [95] for the correct (cleaned) network-constrained trajectory datasets, given any of its spatio-temporal observations  $(x_i^j, y_i^j, t_i^j)$ , its location  $(x_i^j, y_i^j)$  should be over a road edge  $\in E$  (set of edges of  $Ga$ ). Consider two sampled consecutive observations  $L_i^j$  and  $L_i^{j+1}$  where the thresholds  $\beta$  and  $\tau$  are not met. Each observation is associated with the nearest edge in a road map (represented by a graph  $Ga$ ) using the  $get\_edge$  function, i.e.,  $get\_edge(L_i^j, Ga)$  and  $get\_edge(L_i^{j+1}, Ga)$ . The signature of the  $get\_edge$  function is  $L \times G \rightarrow E$ . Here, the nearest edge in the graph  $Ga = (V, E)$  is the output of the  $get\_edge$  function. Therefore, a point  $(x_i^j, y_i^j, t_i^j)$  that is not over an edge  $\in E$  is replaced by a point  $(x_i^j, y_i^j, t_i^j)$  where  $(x_i^j, y_i^j)$  is over an edge of  $E$ , see Figure 3.3.



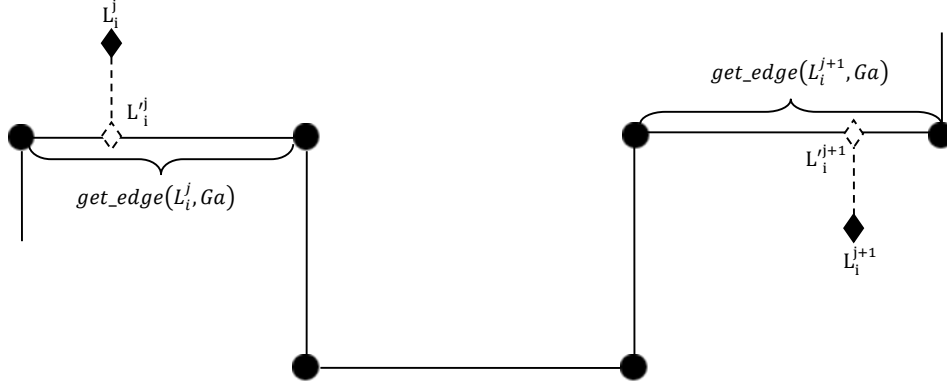


Figure 3.3. Edges where  $L_i^j$  and  $L_i^{j+1}$  fit better

That is, when we consider raw trajectories with a RN, each point is mapped over a road segment by searching for its closest road segment. For this reason, and following the approach of [95], the minimum distance between  $L_i^j$  and a road segment  $e_k$  is computed as follows.

---

Equation 3.1. Minimum distance between  $L_i^j$  and a road segment  $e_k$

---

$$d(L_i^j, e_k) = \begin{cases} d(L_i^j, e_k) & \text{if } L_i^j \in e_k \\ \min \{d(L_i^j, \text{get\_vertex\_source}(e_k)), d(L_i^j, \text{get\_vertex\_target}(e_k))\} & \text{otherwise} \end{cases}$$


---

Where  $L_i^j$  is the projection of  $L_i^j$  over  $e_k$  and  $d(L_i^j, e_k)$  is the perpendicular distance between  $L_i^j$  and  $e_k$ , and  $d(L_i^j, \text{get\_vertex\_source}(e_k))$  and  $d(L_i^j, \text{get\_vertex\_target}(e_k))$  are the Euclidean distance between  $L_i^j$  and the *source/target* vertex of  $e_k$ . Note that the  $d$  function is overloaded with the signatures  $L \times E \rightarrow \mathbb{R}$  and  $L \times V \rightarrow \mathbb{R}$ . The  $e_k$  segment, which has the minimum distance  $d(L_i^j, e_k)$  among all the RN segments is where the point  $L_i^j$  is mapped. That is,  $\text{get\_edge}(L_i^j, Ga) = e_k$ . The main reason of the outcome of the  $\text{get\_edge}$  function is being used as an input of a routing algorithm applied over the RN  $Ga$  as a tool for the imputation process.

### 3.3.1 Getting the location point from routing algorithms

Let  $a$  and  $b$  observations where  $a = \left( \text{get}_x \left( \text{get\_vertex\_target} \left( \text{get\_edge}(L_i^j, Ga) \right) \right), \text{get}_y \left( \text{get\_vertex\_target} \left( \text{get\_edge}(L_i^j, Ga) \right) \right), \text{set\_time}(L_i^j) \right)$  and  $b = \left( \text{get}_x \left( \text{get\_vertex\_source} \left( \text{get\_edge}(L_i^{j+1}, Ga) \right) \right), \text{get}_y \left( \text{get\_vertex\_source} \left( \text{get\_edge}(L_i^{j+1}, Ga) \right) \right), \text{set\_time}(L_i^{j+1}) \right)$ , where we use the  $\text{set\_time}: V \rightarrow \mathbb{T}$  function to assign a timestamp to vertices  $a$  and  $b$ . This function is explained in the *section 3.3.2*.

Then the  $\text{traj}(\{L_i^j, L_i^{j+1}\}, c)$  function returns a sequence of observations  $\{a, o_1, o_2, \dots, o_p, b\}$  describing the route between  $L_i^j$  and  $L_i^{j+1}$  according to a  $c$  criterion, see *Figure 3.4*. Note that the sequence of observations is inferred from the application of a routing algorithm over the Ga Graph between its edges  $\text{get\_edge}(L_i^j, Ga)$  and  $\text{get\_edge}(L_i^{j+1}, Ga)$ .

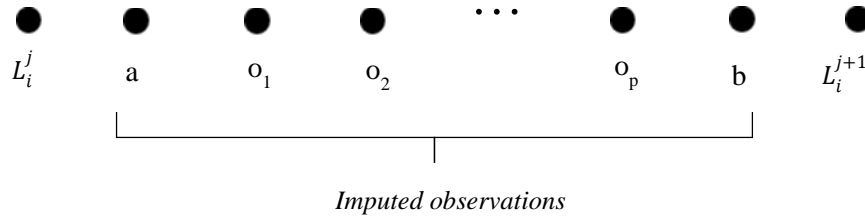


Figure 3.4. Imputed observations between the observations  $a$  and  $b$

In this way, the (sub)trajectory obtained between  $L_i^j$  and  $L_i^{j+1}$  according to a criterion  $c \in Cset$  can be described as:

---

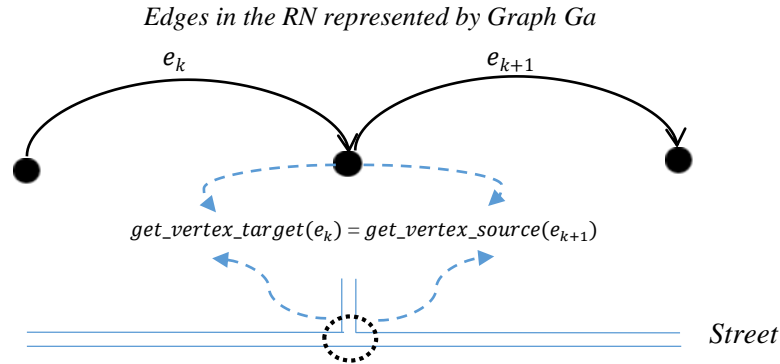
*Equation 3.2. Computation of the (sub)trajectory between  $L_i^j$  and  $L_i^{j+1}$*

---

$$\text{traj}(\{L_i^j, L_i^{j+1}\}, c) = \{L_i^j, \\ ((\text{get}_x(\text{get\_vertex\_target}(e_1)), \text{get}_y(\text{get\_vertex\_target}(e_1)), \text{set\_time}(\text{get\_vertex\_target}(e_1))), \dots, \\ ((\text{get}_x(\text{get\_vertex\_target}(e_k)), \text{get}_y(\text{get\_vertex\_target}(e_k)), \text{set\_time}(\text{get\_vertex\_target}(e_k))), \dots, \\ ((\text{get}_x(\text{get\_vertex\_target}(e_{p-1})), \text{get}_y(\text{get\_vertex\_target}(e_{p-1})), \text{set\_time}(\text{get\_vertex\_target}(e_{p-1}))), \\ L_i^{j+1}\}$$


---

The  $\text{traj}(\{L_i^j, L_i^{j+1}\}, c)$  can be overwritten using  $\text{get\_vertex\_target}(e_k) = \text{get\_vertex\_source}(e_{k+1})$ . According to the RN mapping defined by [29] the end vertex of an edge  $e_k$  is the initial vertex of the edge  $e_{k+1}$ , see *Figure 3.5*. Therefore,  $\text{get}_x(\text{get\_vertex\_target}(e_k)) = \text{get}_x(\text{get\_vertex\_source}(e_{k+1}))$  and  $\text{get}_y(\text{get\_vertex\_target}(e_k)) = \text{get}_y(\text{get\_vertex\_source}(e_{k+1}))$ . Note that  $\text{get\_edge}(L_i^j, G) = e_1$  and  $\text{get\_edge}(L_i^{j+1}, Ga) = e_p$ .



*Figure 3.5. The end vertex of an edge  $e_k$  is the initial vertex of the edge  $e_{k+1}$*

For each imputed observation used for reconstruct the trajectory between  $L_i^j$  and  $L_i^{j+1}$  a timestamp must be inferred. For this goal, we define the *set\_time* function.

### 3.3.2 Getting the timestamps from routing algorithms

To compute the timestamp of each imputed observation of the reconstructed trajectory segment, the difference  $\text{get\_time}(L_i^{j+1}) - \text{get\_time}(L_i^j)$  is to be proportionally assigned to each of them.

Then, the time-stamp of an imputed point can be computed as follows

Let:

*get\_distance*:  $E \rightarrow R$ . Function applied to an edge to get the road distance of the edge.

$D_{L_i^j}$  = The distance from the observation  $L_i^j$  to  $get\_vertex\_source(get\_edge(L_i^j, Ga))$

$D_{L_i^{j+1}}$  = The distance from  $L_i^{j+1}$  to  $get\_vertex\_source(get\_edge(L_i^{j+1}, Ga))$

Let:

---

*Equation 3.3. The total distance between two observations  $L_i^j$  and  $L_i^{j+1}$*

---


$$total\_distance(L_i^j, L_i^{j+1}) = get\_distance(get\_edge(L_i^j, Ga)) - D_{L_i^j} + \sum_{k=2}^{p-1} get\_distance(e_k) + D_{L_i^{j+1}}.$$


---

Note that the summation begins at  $k = 2$  because we suppose that  $get\_edge(L_i^j, G) = e_1$  and ends at  $p-1$  since  $get\_edge(L_i^{j+1}, Ga) = e_p$ . Both,  $e_1$  and  $e_p$  of them are part of the resulting sequence. Then, the timestamp of a  $get\_vertex\_target(e_k)$  vertex is computed as follows:

---

*Equation 3.4. The timestamp of a  $get\_vertex\_target(e_k)$ .*

---


$$\begin{aligned} & set\_time(get\_vertex\_target(e_k)) \\ &= get\_time(L_i^j) + \frac{(\sum_{k=1}^{p-1} get\_distance(e_k)) - D_{L_i^j}}{total\_distance(L_i^j, L_i^{j+1})} \\ & * (get\_time(L_i^{j+1}) - get\_time(L_i^j)) \end{aligned}$$


---

In *Figure 3.6*, the reconstructed trajectory between two observations  $L_i^j$  and  $L_i^{j+1}$  is shown using the *traj* function according to a criterion  $c$ .

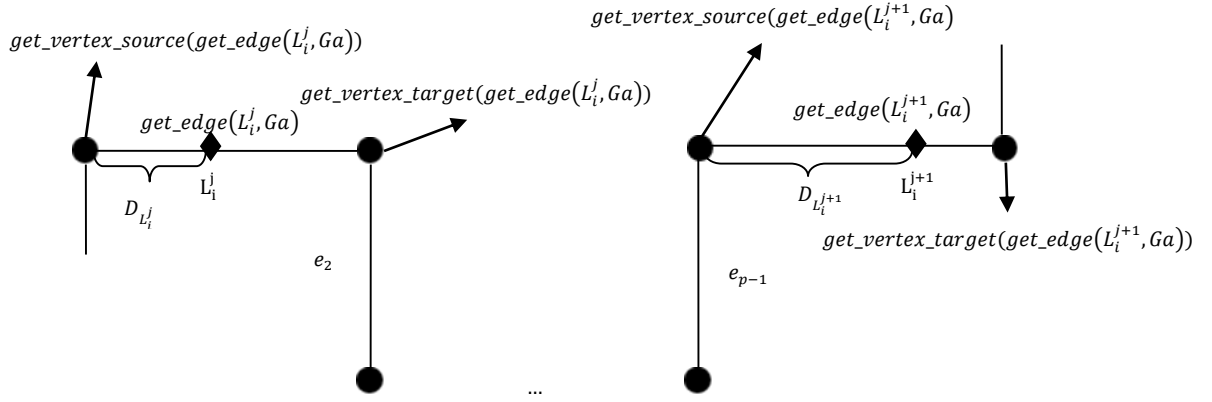


Figure 3.6. Inference of time stamp of edge  $e_k$

**Example.** Let us consider the reconstructed trajectory between the observations  $L_1^1$  and  $L_1^2$  shown in the Figure 3.7 where we get the edges  $e_1, e_2, e_3, e_4$ . Let the  $get\_time(L_1^1) = 02:00:00$  and  $get\_time(L_1^2) = 03:00:00$ , then  $get\_time(L_1^2) - get\_time(L_1^1) = 1$  hour must be proportionally divided among the edges.

Let the  $get\_distance(e_1) = 12$ ,  $get\_distance(e_2) = 10$ ,  $get\_distance(e_3) = 10$ ,  $get\_distance(e_4) = 10$ ,  $D_{L_1^1} = 2$ ,  $D_{L_1^2} = 2$ . Also, note that  $get\_edge(L_1^1, G) = e_1$  and  $total\_distance(L_1^1, L_1^2) = 40$ .

For  $k = 1$

$$set\_time(get\_vertex\_target(e_1)) = get\_time(L_1^1) + \frac{get\_distance(e_1) - D_{L_1^1}}{total\_distance(L_1^1, L_1^2)} * (get\_time(L_1^2) - get\_time(L_1^1)) = 02:00:00 + \frac{1}{4} = 0,25 = 02:15:00$$

For  $k = 2$

$$set\_time(get\_vertex\_target(e_2)) = get\_time(L_1^1) + \frac{get\_distance(e_1) + get\_distance(e_2) - D_{L_1^1}}{total\_distance(L_1^1, L_1^2)} * (get\_time(L_1^2) - get\_time(L_1^1)) = 02:00:00 + \frac{2}{4} = 02:30:00$$

For  $k = 3$

$$\begin{aligned} \text{set\_time}(\text{get\_vertex\_target}(e_3)) &= \text{get\_time}(L_1^1) + \\ &\frac{\text{get\_distance}(e_1) + \text{get\_distance}(e_2) + \text{get\_distance}(e_3) - D_{L_1^1}}{\text{total\_distance}(L_1^1, L_1^2)} * (\text{get\_time}(L_1^2) - \text{get\_time}(L_1^1)) = \\ 02:00:00 + \frac{3}{4} &= 02:45:00 \end{aligned}$$

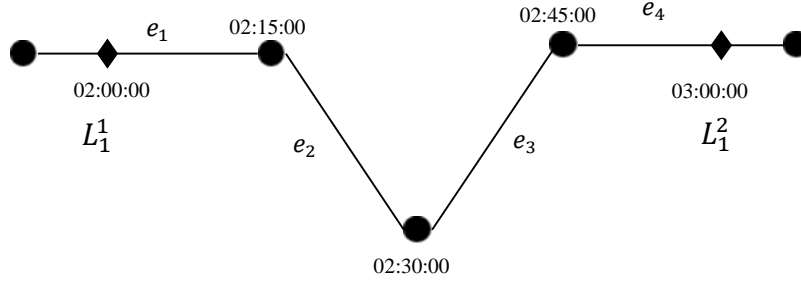


Figure 3.7. Example of time assignment to a reconstructed (sub)trajectory

Note that, after the reconstruction, it is possible that the imputed data points do not meet the  $\beta$  and  $\tau$  thresholds. In this case, the longitude of the street segments are longer than the  $\beta$  threshold because this imputation process stage only gets location points based on the edges of a graph that represents the segments of a RN where a MO moves. Additional imputed data points can be gotten using interpolation methods between the inferred points, i.e., start and end vertex of an edge. The following equations find additional data points over a segment  $e_k$  based on the line equation.

---

*Equation 3.5. Line equation over the segment represented by  $e_k$*

---

$$y = \frac{\text{get\_y}(\text{get\_vertex\_target}(e_k)) - \text{get\_y}(\text{get\_vertex\_source}(e_k))}{\text{get\_x}(\text{get\_vertex\_target}(e_k)) - \text{get\_x}(\text{get\_vertex\_source}(e_k))} * (x - \text{get\_x}(\text{get\_vertex\_target}(e_k))) + \text{get\_y}(\text{get\_vertex\_target}(e_k))$$


---

Where  $\text{get}_x$  and  $\text{get}_y$  are found slicing the segment  $e_k$  in such a way that  $A \leq \beta \wedge \text{road\_distance}(L_i^j, L_i^{j+1}, c) \leq A * \beta$ . Where  $A$  is the amplitude of the sub segments of  $e_k$ .

---

*Equation 3.6. The  $\text{get}_x$  function slicing of the segment represented by  $e_k$*

---

$$x_i = \text{get\_x}(\text{get\_vertex\_source}(e_k)) + \frac{d_i}{\text{road\_distance}(L_i^j, L_i^{j+1}, c)} * (\text{get\_x}(\text{get\_vertex\_target}(e_k)) - \text{get\_x}(\text{get\_vertex\_source}(e_k)))$$


---

---

*Equation 3.7. The get\_y function slicing of the segment represented by  $e_k$*

---

$$y_i = \text{get\_y}(\text{get\_vertex\_source}(e_k)) + \frac{d_i}{\text{road\_distance}(L_i^j, L_i^{j+1}, c)} * (\text{get\_y}(\text{get\_vertex\_target}(e_k)) - \text{get\_y}(\text{get\_vertex\_source}(e_k)))$$


---

where  $d_i = A * i$ ,  $1 \leq i \leq N-1$ .  $N$  is the number of intervals so that  $\text{road\_distance}(L_i^j, L_i^{j+1}, c) = N * A$

**Example.** In *Figure 3.8*, we show an example for finding additional data points for a segment  $e_k$  where  $\text{get\_x}(\text{get\_vertex\_source}(e_k)) = 3$ ,  $\text{get\_y}(\text{get\_vertex\_source}(e_k)) = 1$ ,  $\text{get\_x}(\text{get\_vertex\_target}(e_k)) = 6$ ,  $\text{get\_y}(\text{get\_vertex\_target}(e_k)) = 5$

Let  $\beta = 1.25$ ,  $\text{road\_distance}(L_i^j, L_i^{j+1}, c) = 5$ , then we choose  $A = 1.25$ . Then  $N = 4$ .

$$d_1 = 1.25$$

$$x_1 = 3 + \frac{1.25}{5} * (6-3) = 3 + \frac{1.25}{5} * 3 = 3.75$$

$$y_1 = 1 + \frac{1.25}{5} * (5-1) = 1 + \frac{1.25}{5} * 4 = 2$$

$$d_2 = 2.5$$

$$x_1 = 1 + \frac{2.5}{5} * (6-3) = 3 + \frac{2.5}{5} * 3 = 4.5$$

$$y_1 = 1 + \frac{2.5}{5} * (5-1) = 1 + \frac{2.5}{5} * 4 = 3$$

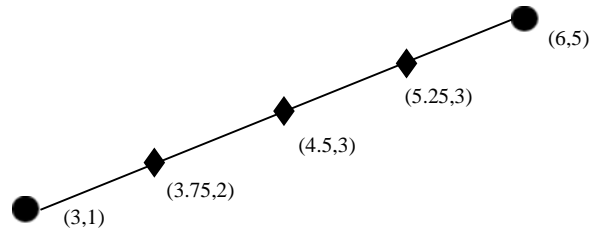
$$d_3 = 3.75$$

$$x_1 = 1 + \frac{3.75}{5} * (6-3) = 3 + \frac{3.75}{5} * 3 = 5.25$$

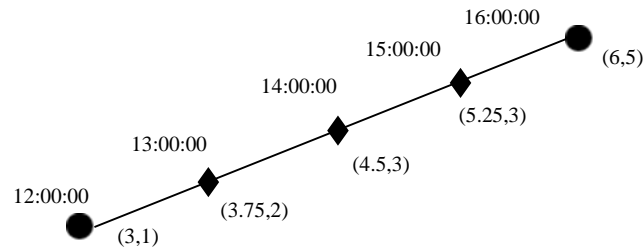
$$y_1 = 1 + \frac{3.75}{5} * (5-1) = 1 + \frac{3.75}{5} * 4 = 4$$

Thus, the set of additional data point between (3, 1) and (6, 5) is  $\{(3.75,2),(4.5,3),(5.25,4)\}$ , see *Figure 3.8*.

The timestamps for each of these points can be found by the proportional assignment of the time difference between observations. The results are shown in *Figure 3.9*, where we suppose that  $set\_time(get\_vertex\_source(e_k)) = 12:00:00$  and  $set\_time(get\_vertex\_target(e_k)) = 16:00:00$ .



*Figure 3.8. Additional imputed data points for an edge  $e_k$*



*Figure 3.9. Additional timestamps data points the start and end vertex of a same edge*

### 3.4 IMPLEMENTATION OF THE “TRAJ” FUNCTION

The application of the *traj* function, according to a criterion  $c$ , between two observations gives as a result a set of points derived from the edges of the resulting reconstructed route. It should be noted that the first stage of the imputations process (trajectory reconstruction) uses the RN for finding the segments where the trajectory traverses, i.e., a set of vertices. If the location data points found do not meet the thresholds, the edge depicted between two vertices is used to meet those thresholds as a second stage.



Given (a) users check-in records describing a set of 2D low-sampling trajectories  $\mathcal{TS} = \{T_i\}$  from a certain LBS and (b) a user criteria preference  $c$  we claim that a “good” route should (a) meet the user criteria preferences, and (b) returns a more detailed trajectory  $T'_i \in \mathcal{TS}$ . *Algorithm 1* calls the *Function 1 (traj)* for each pair of observations that make up the trajectory in a determined dataset  $\mathcal{TS}$ .

### 3.4.1 Algorithm 1: Reconstruction of a Trajectory

---

#### Algorithm 1: Reconstruction of a Trajectory

---

```

INPUT:  $\{ \mathcal{TS} \mid \forall T_i \in \mathcal{TS}, \exists L_i^j, L_i^{j+1} \mid road\_distance(L_i^j, L_i^{j+1}, c) \geq \beta \wedge t_i^j - t_i^{j+1} \geq \tau \}$ 
           $c \in Cset$ 
OUTPUT:  $\{ \mathcal{TS}' \mid \forall T_i \in \mathcal{TS}' \mid road\_distance(L_i^j, L_i^{j+1}, c) \geq \beta' \wedge t_i^j - t_i^{j+1} \geq \tau' \wedge \beta' \leq \beta \wedge \tau' \leq \tau \}$ 
 $\mathcal{TS}' \leftarrow \emptyset$ 
 $T'_i \leftarrow \emptyset$ 
For each  $T_i$  in  $\mathcal{TS}$ 
    For each  $L_i^j$  in  $T_i$ 
      if  $road\_distance(L_i^j, L_i^{j+1}, c) \geq \beta \wedge t_i^j - t_i^{j+1} \geq \tau$  then
         $Trajectory \leftarrow traj(\{L_i^j, L_i^{j+1}\}, c)$ 
        Append  $Trajectory$  to  $T'_i$ 
      else
        Append  $\{L_i^j, L_i^{j+1}\}$  to  $T'_i$ 
        Next  $L_i^j$ 
      end
    End
    Append  $T'_i$  to  $\mathcal{TS}'$ 
End
Return  $\mathcal{TS}'$ 

```

---

### 3.4.2 Function 1: “traj” function for imputation data between two observations of a trajectory

---

**Function 1:** *traj*: Function for imputation data between two observations of a trajectory

---

**INPUT:**  $\{L_i^j, L_i^{j+1} \mid \text{road\_distance}(L_i^j, L_i^{j+1}, c) \geq \beta \wedge t_i^j - t_i^{j+1} \geq \tau\}$   
 $c \in \text{Cset}$

**OUTPUT:**  $\{L_i^j, L_i^{j+1} \mid \text{road\_distance}(L_i^j, L_i^{j+1}, c) \geq \beta' \wedge t_i^j - t_i^{j+1} \geq \tau' \wedge \beta' < \beta \wedge \tau' < \tau\}$   
*// To apply a routing algorithm according to c criterion between get\_edge(L\_i^j, Ga) and get\_edge(L\_i^{j+1}, Ga)*

**For each**  $e_k$

*// Use set\_time function for setting the time to each vertex resulting from the routing algorithm*  
 $O_k \leftarrow ((\text{get\_x}(\text{get\_vertex\_target}(e_k)), \text{get\_y}(\text{get\_vertex\_target}(e_k)), \text{set\_time}(\text{get\_vertex\_target}(e_k)))$

**if**  $\text{road\_distance}(O_k, O_{k+1}, c) \geq \beta \wedge \text{get\_time}(O_{k+1}) - \text{get\_time}(O_k) \geq \tau$  **then**  
*// interpolate between O\_k and O\_{k+1}*  
 Use the **equation 3.6** and **equation 3.7**

**end**

Trajectory  $\leftarrow$   
 $\{L_i^j, ((\text{get\_x}(\text{get\_vertex\_target}(e_1)), \text{get\_y}(\text{get\_vertex\_target}(e_1)), \text{set\_time}(\text{get\_vertex\_target}(e_1))), \dots,$   
 $((\text{get\_x}(\text{get\_vertex\_target}(e_k)), \text{get\_y}(\text{get\_vertex\_target}(e_k)), \text{set\_time}(\text{get\_vertex\_target}(e_k))), \dots,$   
 $((\text{get\_x}(\text{get\_vertex\_target}(e_{p-1})), \text{get\_y}(\text{get\_vertex\_target}(e_{p-1})), \text{set\_time}(\text{get\_vertex\_target}(e_{p-1}))), L_i^{j+1}\}$

**end**

**Return** Trajectory

---

**Example.** To explain how the *traj* function works, let us consider a set of check-in data describing a trajectory of a particular user as shown in *Table 3.1* and the RN of the city of Medellín, Colombia (described by the graph  $G_a$ ) shown in *Figure 3.10*. We also get the nearest edges  $\text{get\_edge}(\text{Check-in } A, G_a)$ ,  $\text{get\_edge}(\text{Check-in } B, G_a)$  and  $\text{get\_edge}(\text{check-in } C, G_a)$ , for each check-in. Those road segments are depicted in solid lines in *Figure 3.10*:

User	Data point	POI name	(x,y,t)
15307763	Check-in A	Shop	(-75.562555,6.249437,20140809134345)
15307763	Check-in B	Restaurant	(-75.576790,6.244406;20140809145517)
15307763	Check-in C	Shop	(-75.591672,6.257514,20140809173745)

Table 3.1. Check – in data of a particular user.



Figure 3.10. Portion of the city of Medellín, Colombia.

Next, the change of the imputed data of the reconstructed trajectories is shown when the criterion changes. Let  $\beta$  less than the actual road distance between each pair of check-in and  $\tau$  less than the actual difference between time check-ins. The *Distance*, *Time*, and *Touristic* criteria are used:

#### *Imputed trajectory using the Distance criterion*

From Check-in A to Check-in B, the (sub)trajectory is computed using the *traj* function  $traj(\{Check - in A, Check - in B\}, c)$  with  $c = Distance$ . The A \* algorithm is used to find the imputed location data between  $get\_vertex\_target(get\_edge(Check - in A, Ga))$  and  $get\_vertex\_target(get\_edge(Check - in B, Ga))$  using the *get\_x* and *get\_y* functions. At the same time, the timestamps for those location data were set using the *set\_time* function and assigning proportionally the difference  $get\_time(Check - in B) - get\_time(Check - in A)$ . A partial result of the imputed observations is listed in Table 3.2. The first part of the trajectory can be seen in Figure 3.11.

<i>User</i>	$((get\_x(get\_vertex\_source(e_k)), get\_y(get\_vertex\_source(e_k)), set\_time(get\_vertex\_source(e_k)))$
<i>User 1</i>	$(-75.5625555, 6.2494373, 20140809134345)$
<i>User 1</i>	$(-75.5620212, 6.2491409, 20140809134656)$
<i>User 1</i>	$(-75.5629924, 6.2496343, 20140809134717)$
<i>User 1</i>	$(-75.5635239, 6.2488592, 20140809135054)$
<i>User 1</i>	$(-75.5620212, 6.2491409, 20140809134717)$
<i>User 1</i>	...
<i>User 1</i>	$(-75.5754726, 6.2450224, 20140809144727)$
<i>User 1</i>	$(-75.5759484, 6.2450904, 20140809144748)$
<i>User 1</i>	$(-75.5760523, 6.2451252, 20140809144748)$
<i>User 1</i>	$(-75.5767275, 6.2437119, 20140809145314)$
<i>User 1</i>	$(-75.576790, 6.244406, 20140809145517)$

Table 3.2. Imputed observations using the Distance criterion between Check-in A to Check-in B.

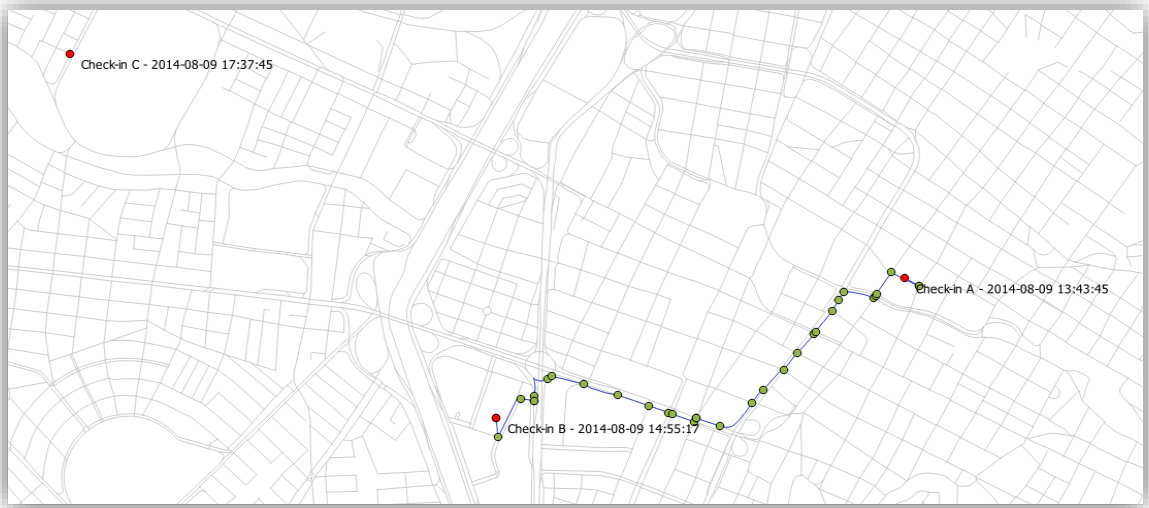


Figure 3.11. Reconstructed Trajectory between Check-in A and Check-in B using Distance criterion from the user 1.

Next, the sub(trajjectory) from Check-in B to Check-in C is computed using  $traj(\{Check - in B, Check - in C\}, c)$  with  $c = Distance$ . A partial result of the imputed observations of this trajectory section is listed in Table 3.3. The last part of the imputed trajectory can be seen in Figure 3.12.

User	$((get\_x(get\_vertex\_source(e_k)), get\_y(get\_vertex\_source(e_k)), set\_time(get\_vertex\_source(e_k)))$
User 1	(-75.576790,6.244406,20140809145517)
User 1	(-75.5767275,6.2437119,20140809150356)
User 1	(-75.5776115,6.244002,20140809150903)
User 1	(-75.5777403,6.2440447,20140809150948)
User 1	(-75.5783803,6.2442404,20140809151329)
User 1	...
User 1	(-75.5924922,6.2564856,20140809172702)
User 1	(-75.5925836,6.2566938,20140809172817)
User 1	(-75.5921396,6.2577242,20140809173431)
User 1	(-75.5916435,6.2574905,20140809173732)
User 1	(-75.591672,6.257514,20140809173745)

Table 3.3. Imputed observations using the Distance criterion between Check-in B to Check-in C.

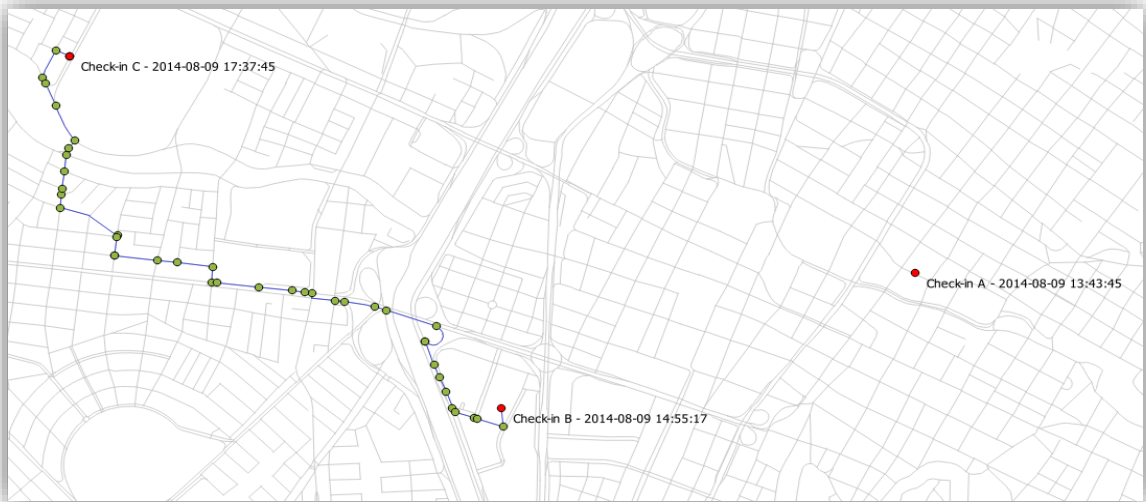


Figure 3.12. Reconstructed Trajectory between Check-in B and Check-in C using Distance criterion from the user 1.

The whole reconstructed trajectory using the Distance criterion is shown in Figure 3.13.



Figure 3.13. Reconstructed Trajectory using Distance Criteria from the user 1

#### Imputed trajectory using the Time criterion

Now, the criterion  $c = Time$  is set. The (sub)trajectory from Check-in A to Check-in B is computed using the *traj* function  $traj(\{Check - in A, Check - in B\}, c)$ . The Dijkstra's algorithm is used to find the imputed location data between  $get\_vertex\_target(get\_edge(Check - in A, Ga))$  and  $get\_vertex\_target(get\_edge(Check - in B, Ga))$  using the *get\_x* and *get\_y* functions. The difference  $get\_time(Check - in B) - get\_time(Check - in A)$  was proportionally assigned using the *set\_time* function. A partial result of the imputed observations is listed in Table 3.4. This first part of the imputed trajectory can be seen in Figure 3.14.

User	$((get\_x(get\_vertex\_source(e_k)), get\_y(get\_vertex\_source(e_k)), set\_time(get\_vertex\_source(e_k)))$
User 1	$(-75.5625555, 6.2494373, 20140809134345)$
User 1	$(-75.5620212, 6.2491409, 20140809134656)$
User 1	$(-75.5623187, 6.2483562, 20140809134924)$
User 1	$(-75.5623576, 6.2482814, 20140809134939)$
User 1	$(-75.5635619, 6.2487949, 20140809135331)$
User 1	...
User 1	$(-75.5754726, 6.2450224, 20140809144819)$
User 1	$(-75.5759484, 6.2450904, 20140809144838)$
User 1	$(-75.5760523, 6.2451252, 20140809144838)$
User 1	$(-75.5767275, 6.2437119, 20140809145314)$
User 1	$(-75.5767905, 6.2444064, 20140809145517)$

Table 3.4. Inferred observations using the Time criterion between Check-in A to Check-in B.



Figure 3.14. Reconstructed Trajectory between Check-in A and Check-in B. using the Time criterion from the user 1.

Next, from Check-in B to Check-in C the (sub)trajectory is computed using  $\text{traj}(\{\text{Check} - \text{in } B, \text{Check} - \text{in } C\}, c)$  with  $c = \text{Time}$ . A partial result of the imputed observations of this trajectory section is listed in Table 3.5. The last part of the imputed trajectory can be seen in Figure 3.15.

User	$((\text{get\_x}(\text{get\_vertex\_source}(e_k)), \text{get\_y}(\text{get\_vertex\_source}(e_k)), \text{set\_time}(\text{get\_vertex\_source}(e_k))))$
User 1	$(-75.5767905, 6.2444064, 20140809145517)$
User 1	$(-75.5767275, 6.2437119, 20140809150317)$
User 1	$(-75.5776115, 6.244002, 20140809150801)$
User 1	$(-75.5777403, 6.2440447, 20140809150843)$
User 1	$(-75.5783803, 6.2442404, 20140809151207)$
User 1	...
User 1	$(-75.5924922, 6.2564856, 20140809172750)$
User 1	$(-75.5925836, 6.2566938, 20140809172900)$
User 1	$(-75.5921396, 6.2577242, 20140809173445)$
User 1	$(-75.5916435, 6.2574905, 20140809173733)$
User 1	$(-75.5916722, 6.2575147, 20140809173745)$

Table 3.5. Inferred observations using the Time criterion between Check-in B to Check-in C.



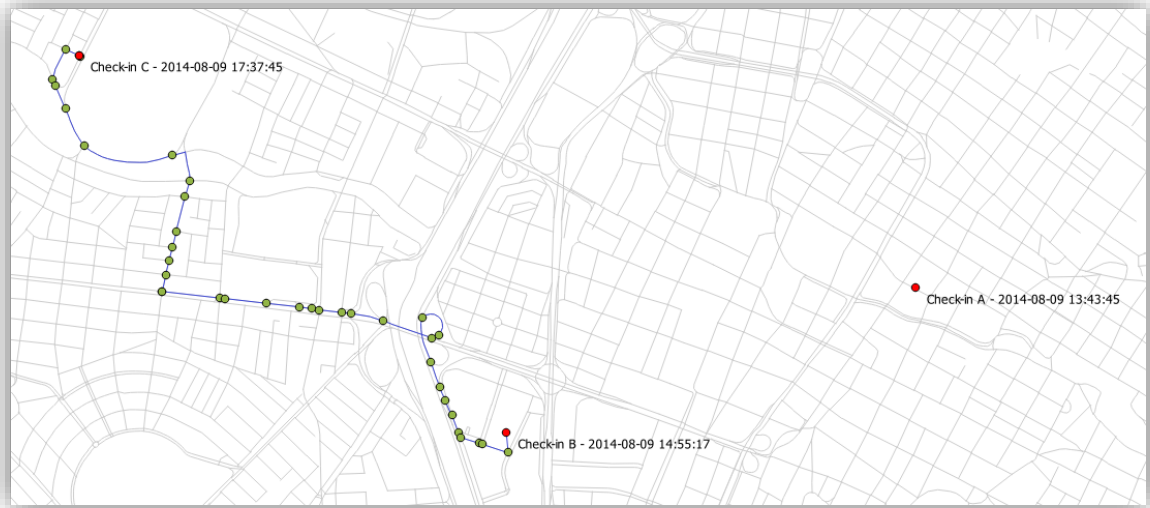


Figure 3.15. Reconstructed Trajectory between Check-in B and Check-in C. using the Time criterion from the user 1.

The whole reconstructed trajectory using the *Time* criterion is shown in Figure 3.16.

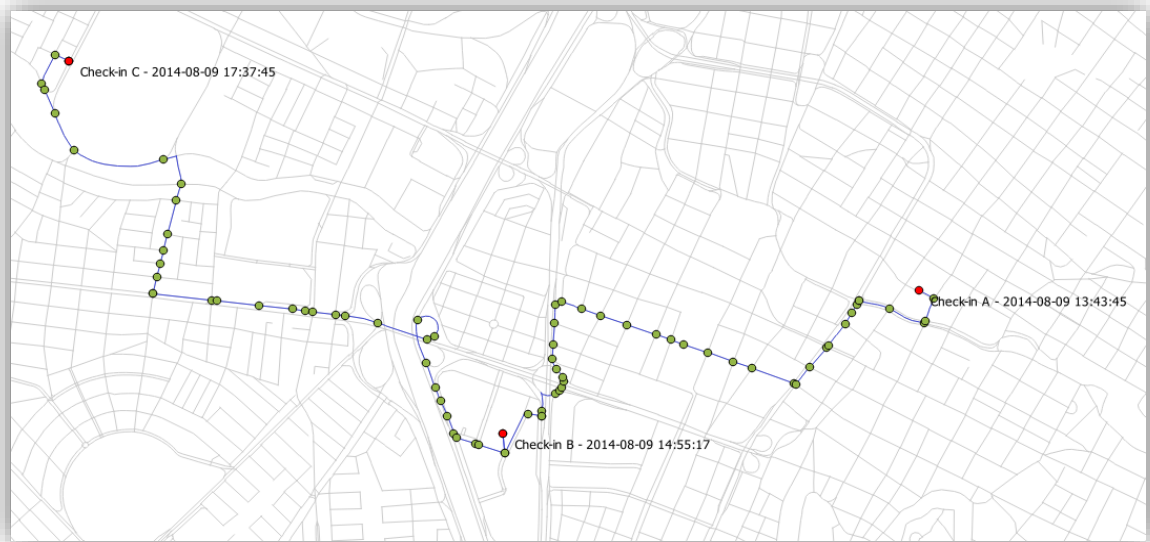


Figure 3.16. Reconstructed Trajectory using the Time criterion from the user 1.



*Imputed trajectory using the Touristic criterion*

Again, the criterion to  $c = \textit{Touristic}$  is set. The (sub)trajectory from Check-in A to Check-in B is computed using  $\textit{traj}(\{\textit{Check-in A}, \textit{Check-in B}\}, c)$ . A partial result of the imputed observations is listed in *Table 3.6*. This first part of the imputed trajectory can be seen in *Figure 3.17*.

<i>User</i>	$((\textit{get\_x}(\textit{get\_vertex\_source}(e_k)), \textit{get\_y}(\textit{get\_vertex\_source}(e_k)), \textit{set\_time}(\textit{get\_vertex\_source}(e_k))))$
<i>User 1</i>	$(-75.5625555, 6.2494373, 20140809134345)$
<i>User 1</i>	$(-75.5620212, 6.2491409, 20140809134717)$
<i>User 1</i>	$(-75.5629924, 6.2496343, 20140809134717)$
<i>User 1</i>	$(-75.5635239, 6.2488592, 20140809135050)$
<i>User 1</i>	$(-75.5620212, 6.2491409, 20140809135050)$
<i>User 1</i>	...
<i>User 1</i>	$(-75.5760523, 6.2451252, 20140809144731)$
<i>User 1</i>	$(-75.5759484, 6.2450904, 20140809144753)$
<i>User 1</i>	$(-75.5760523, 6.2451252, 20140809144753)$
<i>User 1</i>	$(-75.5767275, 6.2437119, 20140809145300)$
<i>User 1</i>	$(-75.5767905, 6.2444064, 20140809145517)$

*Table 3.6. Imputed observations using the Touristic criterion between Check-in A to Check-in B.*



*Figure 3.17. Reconstructed Trajectory between Check-in A and Check-in B using the Touristic criterion from the user 1.*

Next, the (sub)trajectory from Check-in B to Check-in C is computed using  $\text{traj}(\{\text{Check-in B}, \text{Check-in C}\}, c)$  with  $c = \text{Touristic}$ . A partial result of the imputed observations of this trajectory section is listed in Table 3.7. The last part of the imputed trajectory can be seen in Figure 3.18.

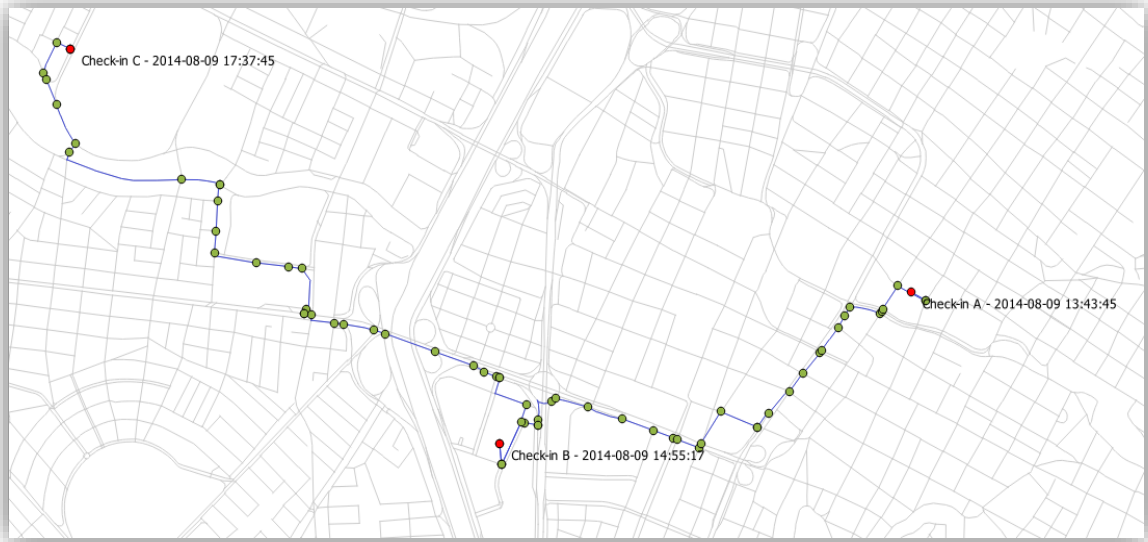
User	$((\text{get\_x}(\text{get\_vertex\_source}(e_k)), \text{get\_y}(\text{get\_vertex\_source}(e_k)), \text{set\_time}(\text{get\_vertex\_source}(e_k))))$
User 1	(-75.5767905,6.2444064,20140809145517)
User 1	(-75.5767275,6.2437119,20140809150344)
User 1	(-75.5760523,6.2451252,20140809150344)
User 1	(-75.5767275,6.2437119,20140809151211)
User 1	(-75.5758606,6.2456828,20140809151211)
User 1	...
User 1	(-75.5924922,6.2564856,20140809172717)
User 1	(-75.5925836,6.2566938,20140809172830)
User 1	(-75.5921396,6.2577242,20140809173435)
User 1	(-75.5916435,6.2574905,20140809173732)
User 1	(-75.5916722,6.2575147,20140809173745)

Table 3.7. Inferred observations using the Touristic criterion between Check-in B to Check-in C.



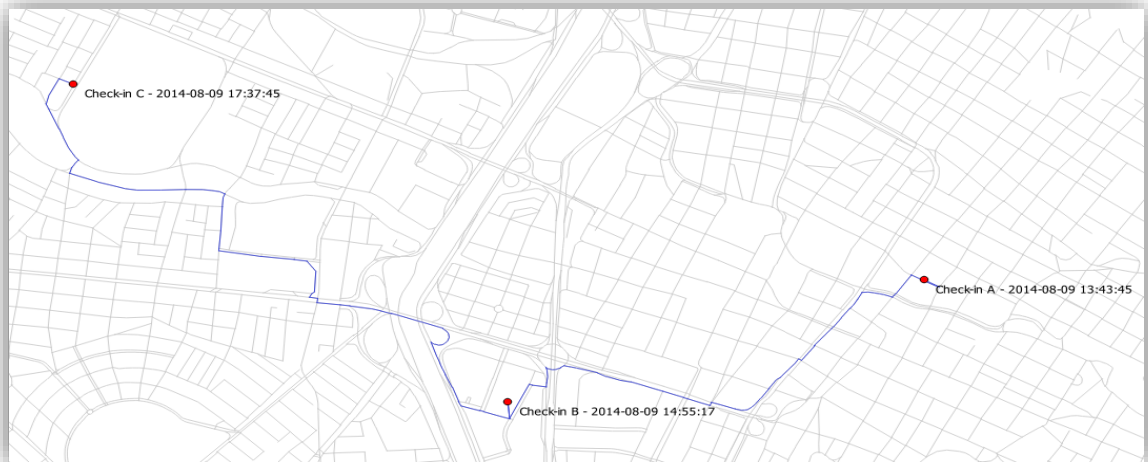
Figure 3.18. Reconstructed Trajectory between Check-in B and Check-in C. using the Touristic criterion from the user 1.

The whole reconstructed trajectory using the Touristic criterion is shown in Figure 3.19.



*Figure 3.19. Reconstructed Trajectory using the Touristic criterion from the user 1.*

Note that the reconstruction changes when a RN and a set of criteria are considered. Other (sub)trajectories described by others imputed observations can be found if other criteria are used. Now, the original trajectory registered by this user in the city of Medellín is presented, see *Figure 3.20*. It differs slightly in some segments streets from the imputed ones.



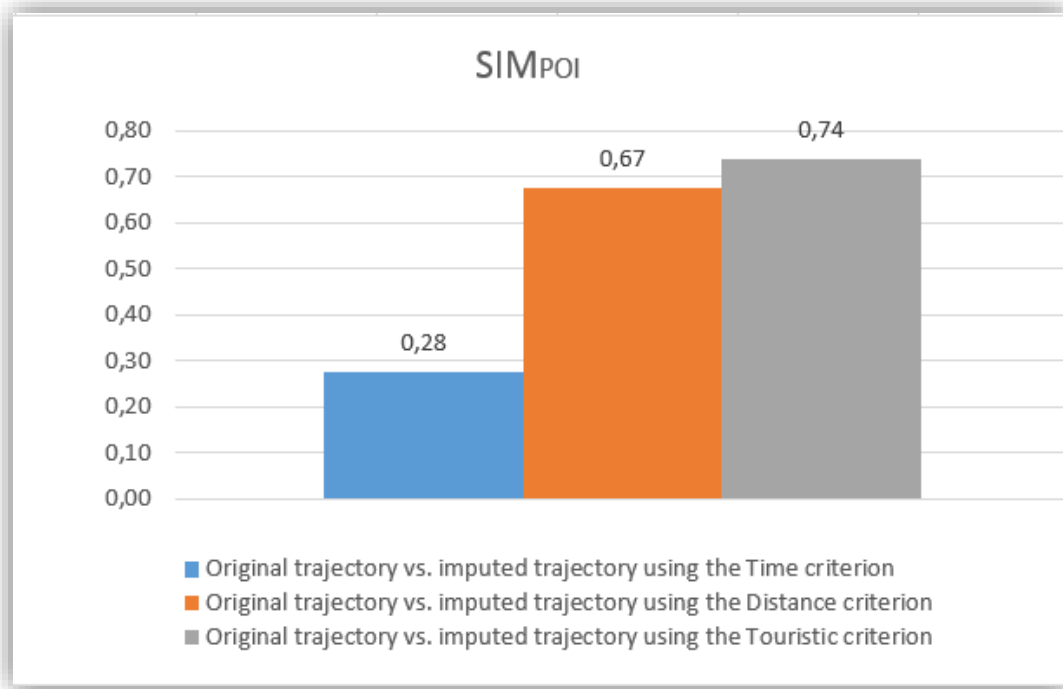
*Figure 3.20. Original Trajectory for user 1.*

*Measuring and comparing the resulting reconstructed trajectories using different criteria with the original one*

There are many approaches for measuring the similarity between trajectories in the literature review [96], [97], [10]. A similar approach proposed by [96] is followed:

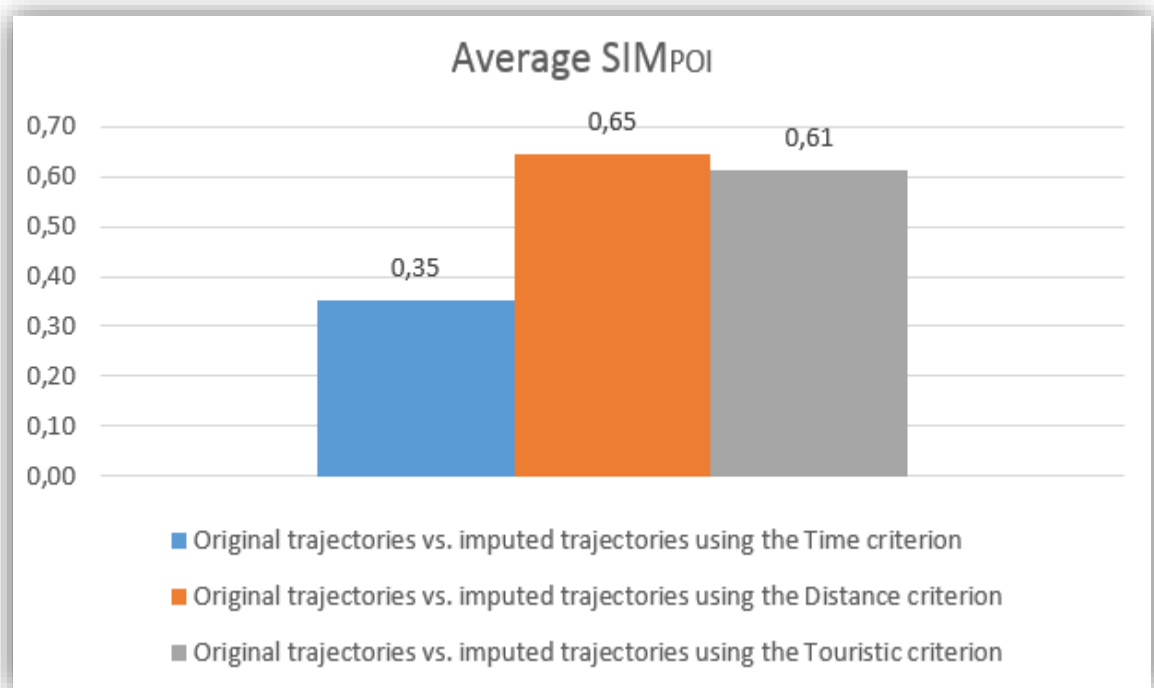
Two trajectories  $T_1$  and  $T_2$  are spatio-temporally similar, iff a) Trajectories  $T_1$  and  $T_2$  have the same temporal granularity, and the trajectories are spatially similar, i.e.,  $SIM_{POI}(T_1, T_2, \theta) < \theta$ , where  $SIM_{POI}(T_1, T_2, \theta) = \frac{POI_{T_1} \cap POI_{T_2}}{POI_{T_1} \cup POI_{T_2}}$  is the a spatial similarity measure,  $\theta$  is a threshold to consider a trajectory spatially similar with other and that the POI regards to important roads or places.

The reconstructed trajectories have the same temporal granularity according to [96] because they have similar time stamp assignment according to the method proposed here, in which the timestamps are assigned proportionally. We consider the POIs as the road segments that a trajectory traverses. The  $SIM_{POI}$  is computed for the reconstructed trajectories, and then compared with the original one, see *Figure 3.21*. Note that the user 1 tends to move using the Touristic criterion instead of the criteria generally provided by common route planners (the shortest distance).



*Figure 3.21. The similarity measure between the inferred trajectories and the original one for the user 1.*

Next, the computation of the  $SIM_{POI}$  measure for 80 highly sample rate trajectories in the city of Medellín, Colombia is carried out. The check-in data were simulated (time and location data were deleted) for those trajectories to get low sampled trajectories and the (sub)trajectories were computed based on some criteria using the *traj* function between the simulated check-ins, see *Figure 3.22*. Note how the average  $SIM_{POI}$  is higher when the Distance criterion is used followed by the Touristic criterion, i.e., the best imputation process for this 80 trajectories can be achieved when some of these criteria are used. However, remember that the purpose of the trajectory reconstruction proposed here is to discover the new possibilities of reconstruction as an imputation process to infer the original trajectories. The trajectory reconstruction procedure takes place in order to transform low sampled location data into trajectories with a better sampling so that we can acquire some useful knowledge. In this case, based on reconstruction criteria.



*Figure 3.22. The average similarity measure between the reconstructed trajectories and the original ones for a set of 80 users.*

### 3.5 CONCLUSION AND FUTURE WORK

Valuable information can be extracted from trajectories. It can be useful for location-based services applications including trip planning, personalized navigation routing services, mobile commerce, and location-based recommendation services. In this chapter, low-sampling trajectories were reconstructed using the personalization features of the routing theory based on a criterion decision over a graph. Using the *traj* function with different criteria can be used as an input for different mining algorithms over trajectories as a way to deal with analytics using uncertain trajectories. Here, it is claimed that analytics over reconstructed trajectories can change depending on the criterion used for the trajectory reconstruction. Also, this criteria based reconstruction can be used to perform analytical tasks and offer the possibility of answers questions based on user criteria, such as:

- How the *presence measure* (the number of distinct trajectories that lie in a spatial region) [26] varies according to the reconstruction criterion selected?
- How do regions of interest [98] vary according to a chosen reconstruction criterion during a determined time?
- What are the main bottlenecks in the city in a determined time according to a certain reconstruction criterion of movement?
- What would be the fuel consumption of movement if the people moved according to a certain criterion in a determined period?

**The main contributions of this chapter are:**

- The development of a method of reconstructing low-sampling trajectories according to user criteria.
- The modeling of the incorporation of user criteria for the reconstruction of low-sampling trajectories.
- The reconstruction process can be used in an imputation process [99] over low-sampling trajectory data.
- This chapter develops the specific objective “*Develop a user criteria based operator for the reconstruction of a low-sampling trajectory*” using the *traj* function.

## CHAPTER 4. USING CRITERIA RECONSTRUCTION OF LOW-SAMPLING TRAJECTORIES AS A TOOL FOR ANALYTICS

### 4.1 INTRODUCTION

Today, a lot of applications with incorporated Geo Positional Systems (GPS) deliver huge quantities of spatio-temporal data. *Trajectories* followed by MOs can be generated from this data. However, these trajectories may have silent durations, i.e., time durations when no data are available for describing the route of a MO [10]. As a result, the movement during silent durations must be described and the low sampling data trajectory need to be filled in using specialized techniques of data imputation to study and discover new knowledge based on movement.

A novel and relevant task when MOs are analyzed is the characterization of trajectories based on some criteria and the geographical space where they occur. In [11], the authors offer a brief taxonomy to build the “best” trajectory based on criteria like shortest distance, time, point of interest (POI) and simplicity of the road network (RN). Multiple options regarding to the user decision strategies must be also integrated [16]. The problem of route reconstruction using a set of metrics different from distance is still an open research issue [12] and requires the adaptation of new customized metrics, and possibly combinations of them, for reconstructing the trajectories.

In *Chapter 3*, we proposed a function called “*traj*” for reconstructing low-sampling trajectories based on user criteria. An imputation process is carried out for handling uncertainty for trajectories followed by a set of MO in a RN. The function is defined using an explicit criterion parameter that describes the intention of the movement with metrics. The inclusion of the movement criteria in the analysis of trajectories is an important contribution. It will be even greater when uncertainty trajectory data is reconstructed and analyzed as a whole for studying and discovering knowledge. In this chapter, we propose several analytics possibilities using several tools of analysis such as: graphics and data warehouse (DW).

As expressed by [5], the movement expressed by trajectories themselves are not always the main focus of analysis. The trajectories can be analyzed with the aim of gain knowledge about MO or about the environment where trajectories take place, e.g., the RN. For this reason, some measures are explored and some questions are sketched out to show their variation and results according to a given criterion using tools such as DW techniques. Basic properties of the trajectories such as travelled distance, travel time including fuel consumption (if the trajectories under consideration are

done by vehicles) can be object of analysis. Our interest is to show other opportunities of analytical tasks using a criteria based operator over reconstructed low-sampling trajectories. Also, a simple visual analysis of the reconstructed trajectories is done to offer a simple analytic perspective of the reconstruction and how the criterion of movement can change the analysis. To the best of our knowledge, this work is the first attempt to use the different reconstruction of trajectories criteria to identify the opportunities of analytical tasks over reconstructed low-sampling trajectories as a whole.

Although in *Chapter 3* uncertainty is handled using the criteria based method for reconstructing trajectories, analytical tasks are not applied to these reconstructed trajectories. As expressed in previous chapters, the ultimate objective of reconstructing trajectories is to perform better analysis tasks over trajectories. DW approaches might be used to deal with these tasks. Elements such as hierarchies and aggregations, and techniques such as mining and visualization have been adapted to the spatiotemporal data to support such analysis into a new concept called Spatio-Temporal Data Warehouse (STDW) [24]. One step further from modeling a STDW is related to the integration of the movement described by a MO, i.e., trajectories, in a trajectory data warehouse (TDW) [26], [27], [87].

Because of the DW based on spatiotemporal data still lacks of analytical tasks [10], [27] we extend the approach proposed in *Chapter 3* for analytic tasks to determine how analysis changes when the movement criterion is incorporated in the reconstruction of low-sampling trajectories.

The rest of this chapter is organized as follows. *Section 4.2* describes the analytical proposal including visualization in *Section 4.2.1* and analysis tasks in a DW architecture in *Section 4.2.2*. Some analytical question are addressed to show analytical possibilities. *Section 4.3* concludes the chapter describing the results of the proposed analysis task and proposing future works.

## **4.2 THE PROPOSAL OF ANALYSIS**

The idea behind the trajectory reconstruction proposed in *Chapter 3*, is to be applied to a set of trajectories to impute missing data in a preprocessing stage. This approach is extended here for analytic tasks to determine how analysis of MOs change when a movement criterion is incorporated for reconstructing low-sampling trajectories. Analysis tool such as: *graphical* and *TDW* approaches are used to accomplish this task. The first is referred to a simple visual analysis of the reconstructed trajectories. In the second, we use the *traj* function in a stage of a TDW solution to support analytics



over a set of reconstructed trajectories. The *traj* function is used for preprocessing the location data trajectory for each criterion of interest and then each criterion is mapped as a member in a dimension.

The reasons for using a TDW approach are:

- In a TDW environment, the criteria can be represented in a dimension of analysis.
- A huge data generation from GPS based application.
- The analyst must slice and dice the trajectory data in every possible way.
- Companies based on location marketing or mobility can use trajectory data information to support more fact-based decision making.

Next, the approach proposed in *Chapter 3* is summarized:

- First, cost is applied to each segment of the RN to represent the criteria needed. The survey made in *Chapter 2* highlights three main routing criteria: time, distance, and attractiveness (scenic path POIs-based).
- The RN where the observations occurs are mapped into a graph representation.
- Each observation of each low-sampling trajectory is mapped into a road segment by searching for its closest road segment.
- The *traj* function is applied between the mapped observations for each trajectory. Here, a routing algorithm such as Dijkstra is called passing as a parameter the cost of each edge and each pair of observations for each trajectory of the data set.
- A set of edges is retrieved describing the route in the RN between the observations of each trajectory. We get the longitude and latitude of each vertex of each edge and set the time for each vertex proportionally according to the total distance following the criteria applied.
- Additional imputed data points can be gotten using interpolation methods between the inferred points, i.e., the start and the end vertex of an edge if the previous steps do not meet the thresholds of time and distance required.

#### **4.2.1 A Graphical analysis**

A basic visual analysis of the reconstructed trajectories for each criterion offers a simple analytic perspective for the reconstruction proposed here. Check-in data in the city of Medellín, Colombia on August 14, 2014 is used for proposing visual analysis. See *Figure 4.1*, the name and time-stamp

of each check-in is shown. The details of how this source data were obtained are explained in *Section 3.5.2*. Additional data check-in points by days of the collected dataset are drawn in *Chapter 5*.

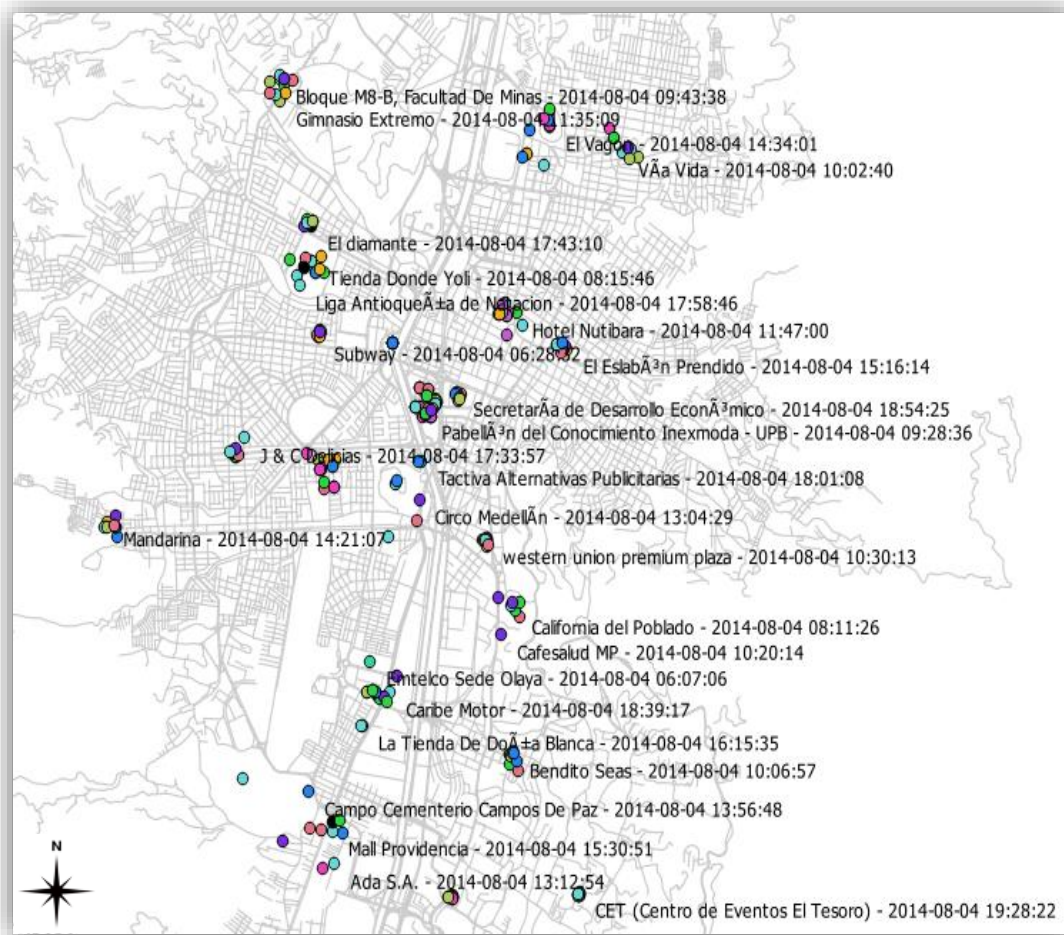
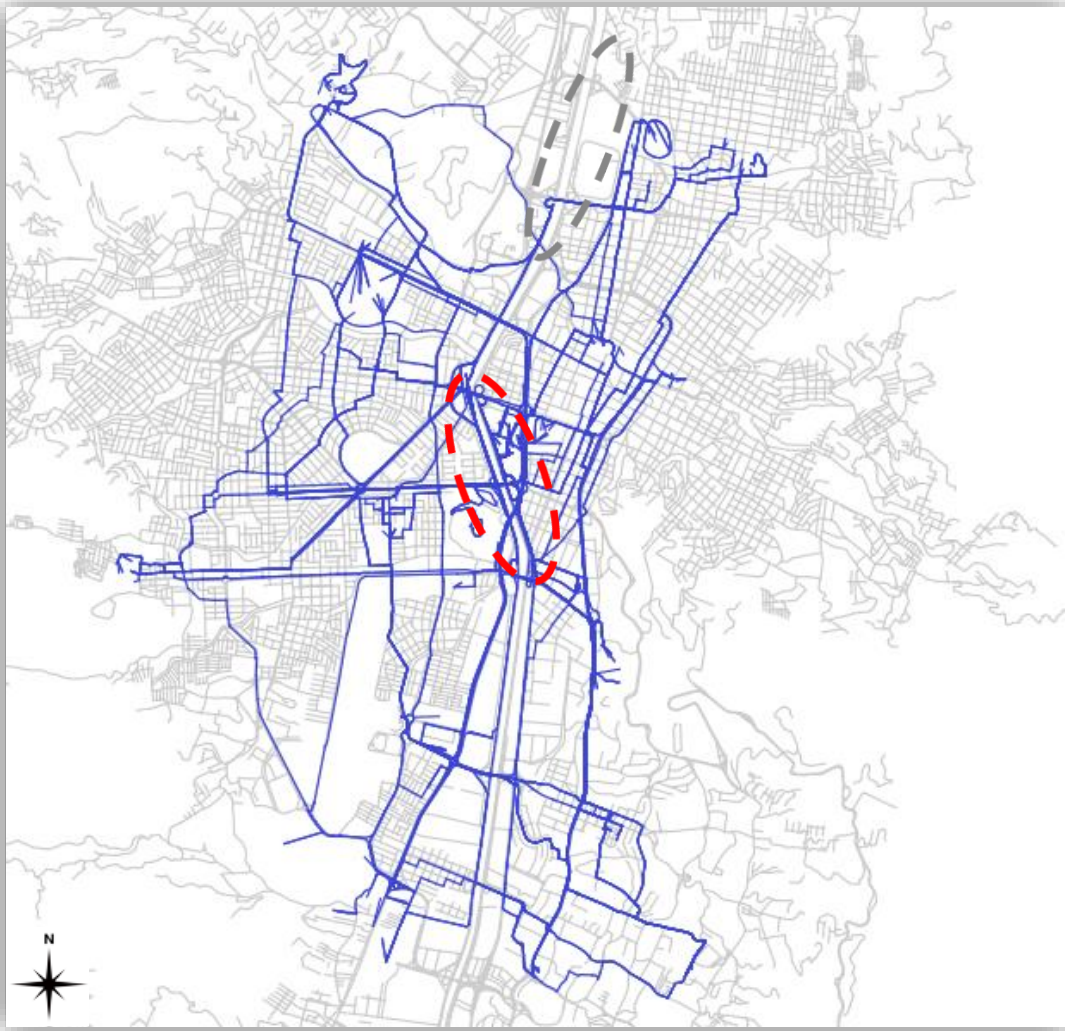


Figure 4.1. A set of check-in points on August 4, 2014 (Medellín)

The *traj* function proposed in the *Chapter 3*, is applied to the set of low-sampling data on August 4, 2014 using criteria such as *distance*, *time*, and *touristic*. The resulting reconstructed trajectories are shown in the *Figure 4.2*, *Figure 4.3*, and *Figure 4.4* when *distance*, *time*, and *touristic* criteria are applied, respectively. Additional reconstructed trajectories by criteria and days of the collected dataset are drawn in the *Chapter 5* for more analysis tasks.

Note that some routes are not used when the criterion changes, see some examples highlighted with the gray dashed ellipses in the *Figure 4.2*, *Figure 4.3*, and *Figure 4.4*. Some of those road segments can be the representative ones for each criterion such as in the *Figure 4.3*, where that segment seems to be the fastest choice when *time* criterion is considered. Also, note that some segments of streets remains used whatever the criterion of movement is selected. See examples highlighted with the red

dashed ellipses in *Figure 4.2*, *Figure 4.3*, and *Figure 4.4*. Common segments present in all the criteria can be target as possible bottlenecks if they do not change when criterion of movement change.



*Figure 4.2. Reconstructed trajectories using Distance criterion on August 4, 2014 (Medellín)*

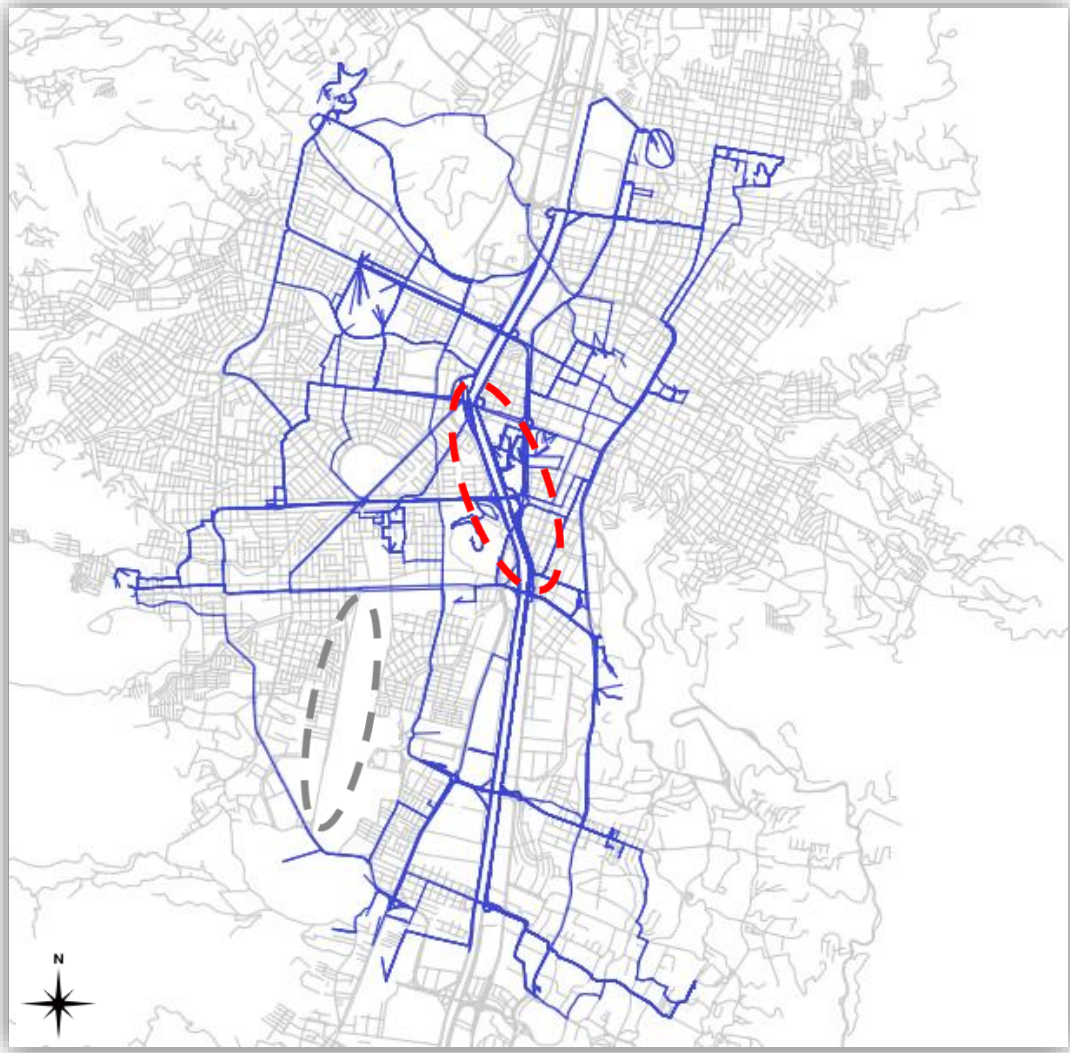


Figure 4.3. Reconstructed trajectories using Time criterion on August 4, 2014 (Medellín)



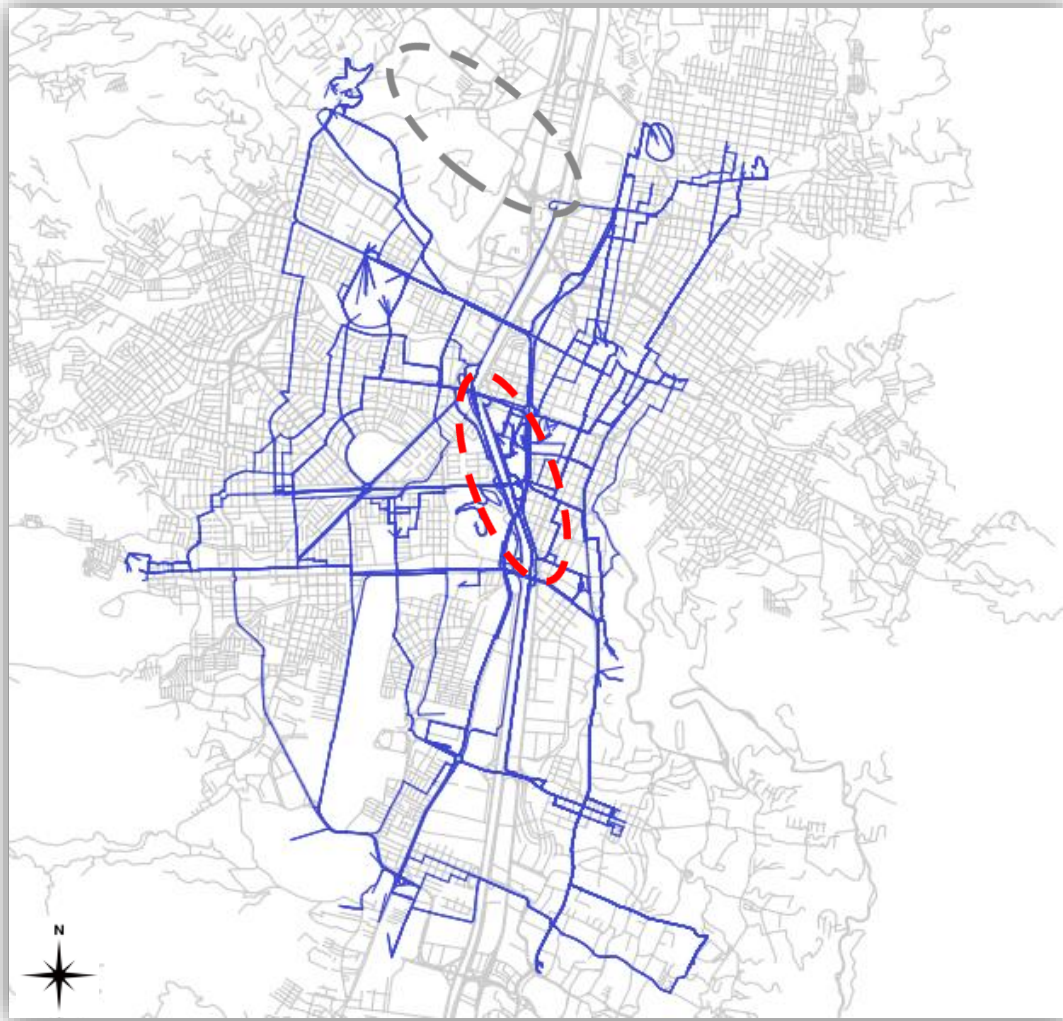


Figure 4.4. Reconstructed trajectories using Touristic criterion on August 4, 2014 (Medellín)

A visual analysis with a color gradient shows the segments with the most trajectories traversing them, see Figure 4.5, Figure 4.6, and Figure 4.7. Those simple gradient visualization can help to identify the streets where a possible bottleneck can be formed if all the users follow the same movement criteria. Again, segments with a higher color gradient in each criterion can help to identify, possible bottlenecks.

In Figure 4.5, the reconstruction of the trajectories based on the *distance* criterion is shown. Note that the segments of the Medellín RN with the most visible color, shows a higher traffic for those streets/avenues.

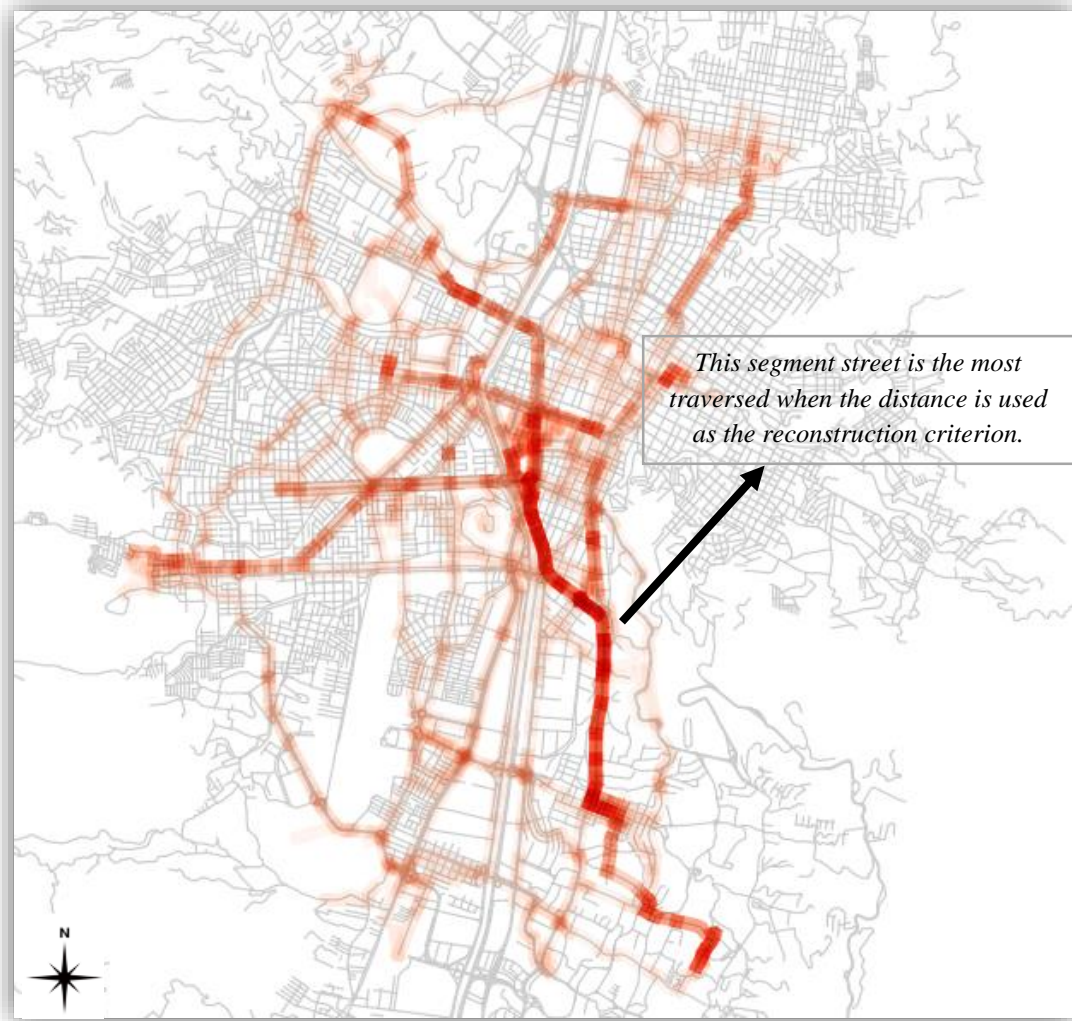


Figure 4.5. A color gradient of reconstructed trajectories using Distance criterion

In Figure 4.6, the trajectories are built using the *time* criterion. Note that a higher number of trajectories are passing through a long segment traversing the city from north to south. This is a highway with three lanes in Medellín, Colombia city (Regional Avenue, as presented in the detailed image from OpenStreetMap); therefore, it is considered to have a fastest traffic flow.

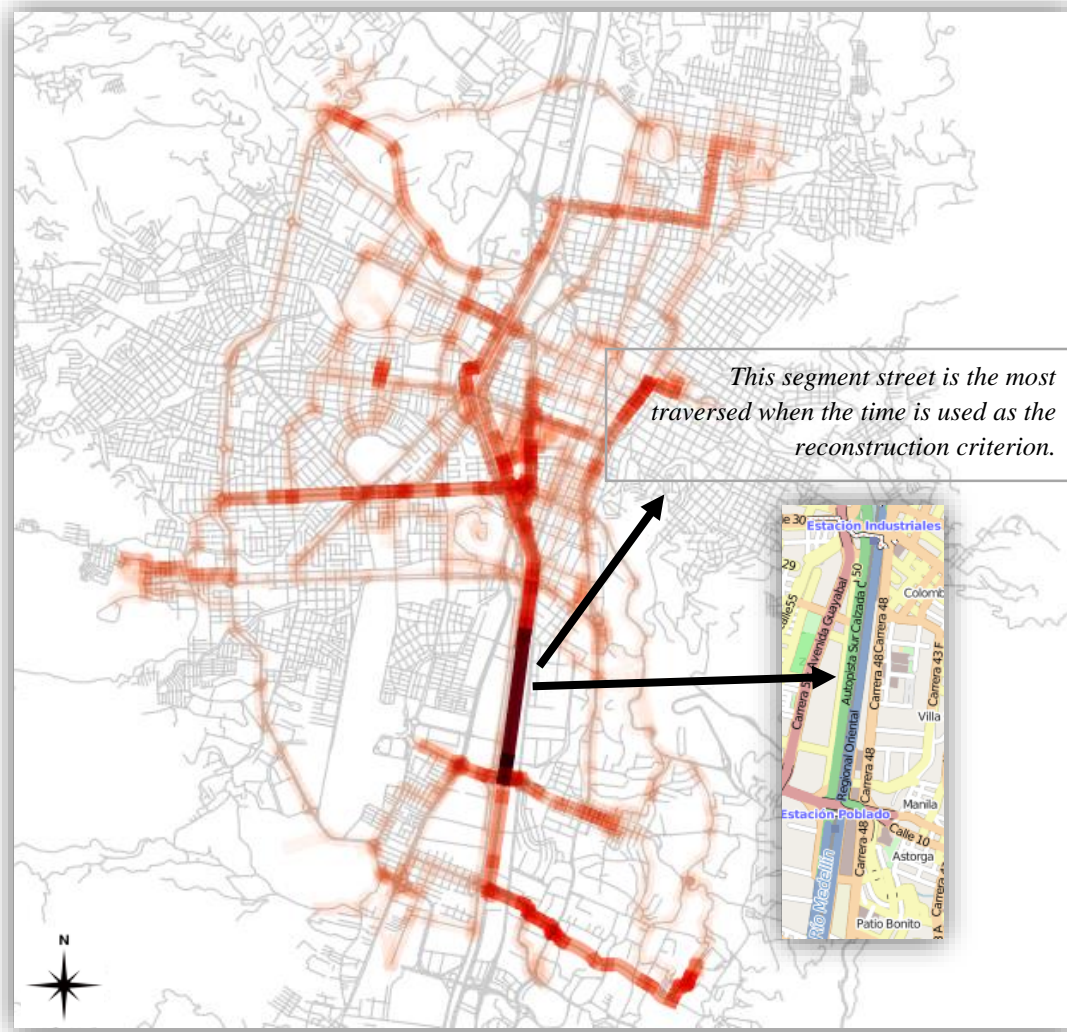
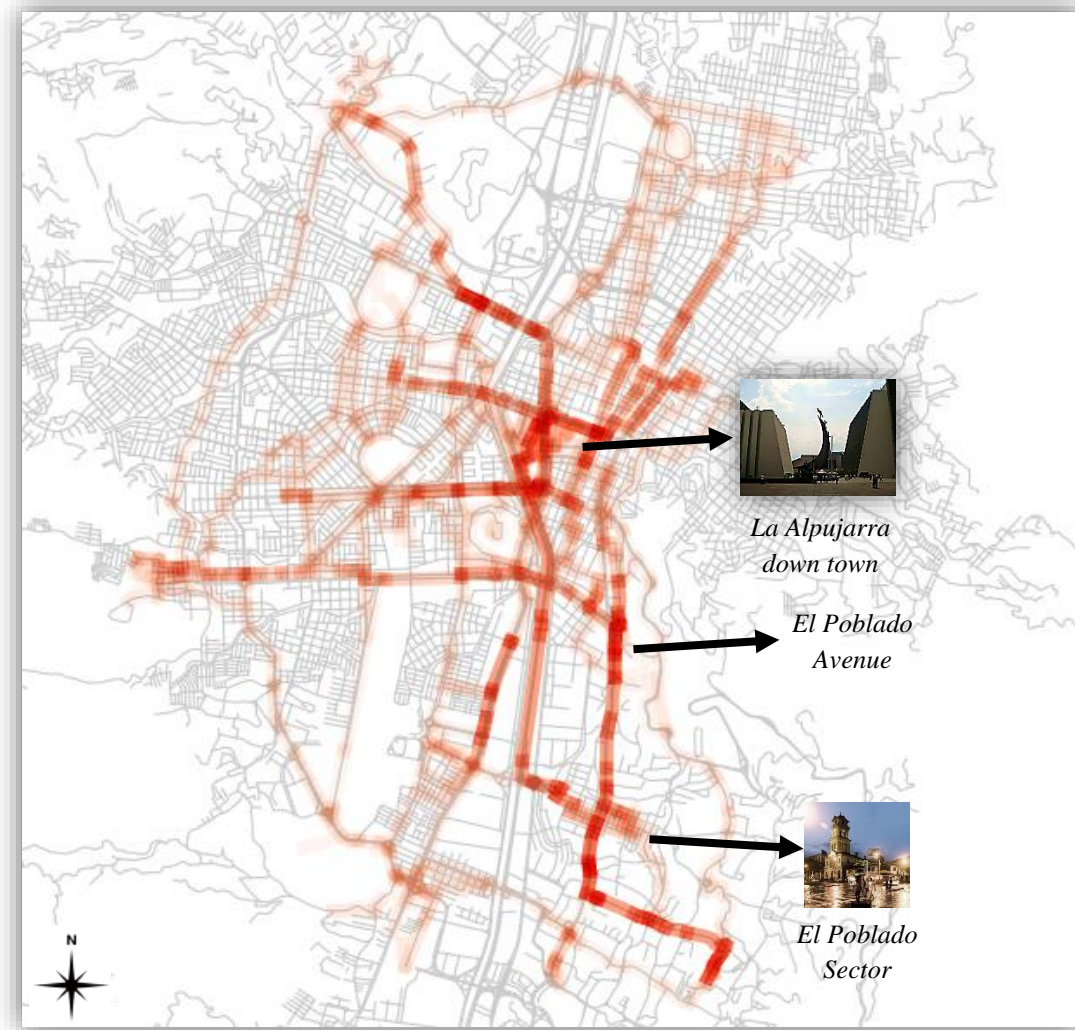


Figure 4.6. A color gradient of reconstructed trajectories using Time criterion

In Figure 4.6, the trajectories are built using the *touristic* criterion. Note that a higher number of trajectories are passing through the downtown ("La Alpujarra" administrative center and "San Juan" street) and "El Poblado" sector (including the avenue leading to this sector), where the most restaurants and clubs are located (see the images attached to the map). In a marketing campaign, the most visible segments can be targeted for visual directed advertising or for enhancing mobile applications such as Foursquare helping merchants to boost his/her business nearest to those segments.



Demographic information (e.g. description, gender, date of birth, profession) and device-related techno-graphic information (e.g. GPS or Cell type) can be also included to slice the data by user profile.



*Figure 4.7. A color gradient of reconstructed trajectories using Touristic criterion*



## 4.2.2 A Trajectory Data Warehouse analysis

Another possible tool for showing the analysis variation of the criteria is a TDW architecture, where the criteria can be considered dimensionally. Here, the *traj* function is used in a data preprocessing step in the stage of data transformation. Every low-sampled trajectory is imputed and marked for each criteria and then stored in the fact table, i.e., each trajectory is reconstructed and stored as many times as the number of criteria are incorporated to the analysis. In *Figure 4.8* a basic a TDW architecture is shown including low-sampling trajectory reconstruction. In the following we expand and explain each stage of the TDW proposed architecture.

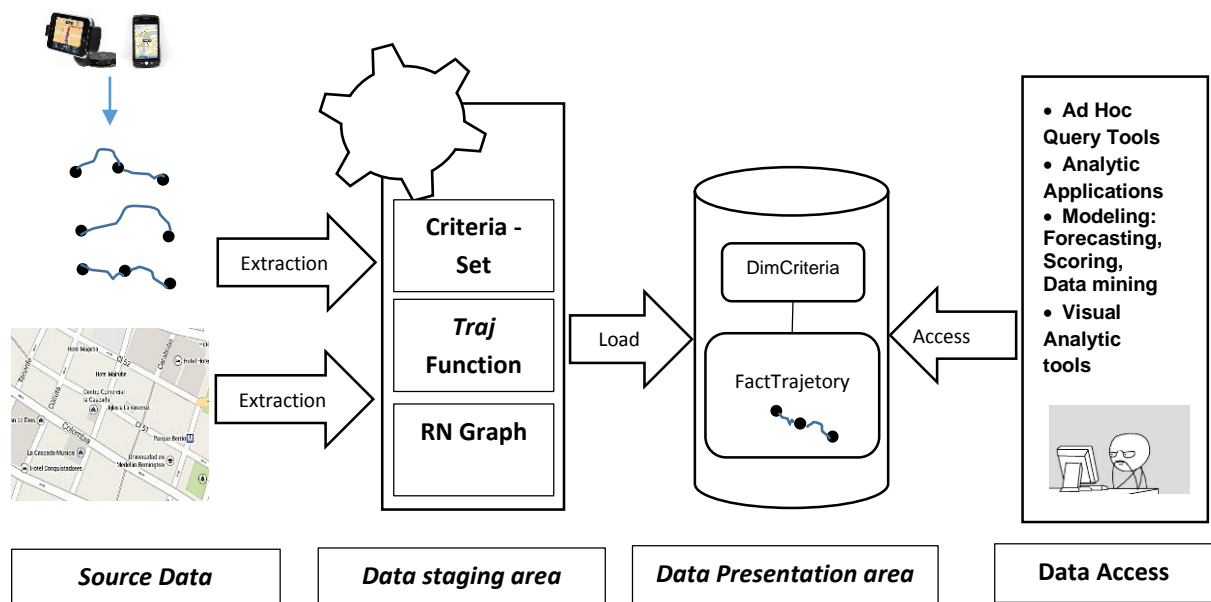


Figure 4.8. A Data warehouse architecture including traj function

### 4.2.2.1 Source Data

The source location data may come from diverse location based data such as: GPS Logs, Check-in data and geotagged photos. For the purpose of this thesis, a set of Foursquare check-ins of 80 random active users during a week in the city of Medellín, Colombia were collected using the public API from Foursquare [100], see *Chapter 5* for technical details. A basic distribution of the data is shown in *Table 4.1* (other data details are included in *Chapter 5*). The check-in data are used to show with examples the change of the analysis according to movement criteria. In *Figure 4.1*, those check-in points on August 14, 2014 were shown. Note that the check-in data have a lot of uncertainty due to the characteristics and purposes of these mobile applications.

Date	Min Time	Max Time	Check-in Quantity	User Quantity
2014-08-04	06:07:06	19:53:26	257	79
2014-08-05	06:00:30	19:55:45	232	75
2014-08-06	06:04:35	21:57:38	222	76
2014-08-07	06:00:25	22:56:26	188	73
2014-08-08	06:01:14	22:46:38	224	77
2014-08-09	00:00:31	22:53:25	235	77
2014-08-10	00:00:00	19:34:43	242	78

Table 4.1. Quantity of check-in's users by day (Medellín)

Also, we need to load the graph  $G_a$  that represents the RN where the trajectories take places according to the parameters of the *traj* function. The RN is then mapped into a spatial dimension of the TDW proposal. In order to get the RN graph  $G_a$  of the city of Medellin, we use *osm2po's* [101] converter that uses *OpenStreetMap's* [102] XML-Data and makes it routable. It generates SQL files for *PostGIS* [103] databases, compatible with *pgRouting* [104]. The TDW was implemented using *Postgress 9.2 DBMS* [105].

#### 4.2.2.2 Data Staging area

The storing and transformation of data between the sources of information and a (DW) is done in the staging area. The stage tables that stores the data were loaded, cleaned, and standardized from the described sources developing an ETL process (check *Chapter 5* for technical details). Functions for computing the imputed data were also developed. Those are:

- Road Network Graph: The XML files generated by *osm2po's* converter are loaded in the stage area and the  $G_a$  is now represented by a stage table containing data of road connection, directions, and cost of the streets segments.
- The whole functions described in *Chapter 3* are implemented in *Postgress 9.2 DBMS* using functions and view objects [105].
- Each observation is mapped into the nearest edge.
- Cost is applied to each segment of the RN according to the set criteria.
- In a *Postgis* database, the *traj* function is implemented. Each trajectory is computed for each criterion and stored using the *traj* function.

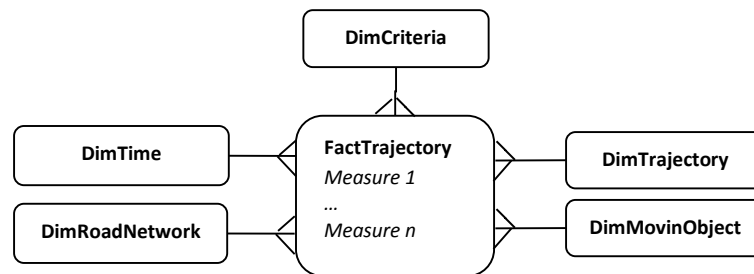
The trajectory reconstruction procedure takes place to impute low sampled location data originated, e.g., from GPS recordings into trajectories with a better sampling. The ETL process that feeds the

TDW is implemented using Pentaho Data Integration 5.0.1 [106]. The technical details of the ETL developed are explained in *Chapter 5*.

#### 4.2.2.3 Data Presentation area

As it usually happens in data management world, the challenge after storing the data is to make the right analysis that could extract useful knowledge [91]. Considering that a trajectory is a spatial object whose location changes in time [27], a TDW should include a spatial and a temporal dimension describing geography and time, respectively [87]. As such, different features need to be described: numeric, spatial, and temporal [27], [87]. Another dimension regarding to *conventional* data about MO (including demographic data, such as gender, age, occupation etc.) could be considered as well.

From *Figure 4.8* we zoom in the TDW model in *Figure 4.9*, the model is composed by the Dimension of MO (*dimMovingObject*), that stores the objects that describes the trajectory; the dimension of trajectories (*dimTrajectory*), that stores the ID for each raw trajectory to differentiate them from its reconstructed ones; the dimension of criteria (*dimCriteria*) that stores the description for each criterion of reconstruction; the dimension of time (*dimTime*); the geometric dimension of the underlying RN (*dimRoadNetwork*); and the fact table of reconstructed trajectories (*factTrajectory*), that stores a set of measures of interest for each segment/edge that make up the reconstructed trajectory.



*Figure 4.9. Dimensional Model of the Data warehouse proposal*

In *Table 4.2*, we zoom in the *factTrajectory* entity. It shows an example of a fact table with the reconstructed trajectories according to a set of criteria after the *traj* function is applied in a preprocessing stage. Each low sampled trajectory  $T_j$  of the object  $ID_i$  is reconstructed for each  $c$  criterion considered, computed and stored. Similarly, each measure of interest is computed for each segment of the trajectory delimited by the interval of the two consecutive observations. For the

purpose of this thesis, we assume explicitly that a trajectory portion can be mapped into a RN segment.

MO ID	Trajectory ID	Observation	RN ID	time	Criterion ID	Measure1	...	Measure n
				...				
ID <sub>i</sub>	T <sub>j</sub>	[L <sub>1</sub> <sup>1</sup> , L <sub>1</sub> <sup>2</sup> ]	RN <sub>x</sub>	[t <sub>1</sub> <sup>1</sup> , t <sub>1</sub> <sup>2</sup> ]	C <sub>1</sub>			
ID <sub>i</sub>	T <sub>j</sub>	...	...	...	C <sub>1</sub>			
ID <sub>i</sub>	T <sub>j</sub>	[L <sub>1</sub> <sup>j</sup> , L <sub>1</sub> <sup>j+1</sup> ]	RN <sub>x</sub>	[t <sub>1</sub> <sup>j</sup> , t <sub>1</sub> <sup>j+1</sup> ]	C <sub>1</sub>			
ID <sub>i</sub>	T <sub>j</sub>	...	...	...	C <sub>1</sub>			
ID <sub>i</sub>	T <sub>j</sub>	[L <sub>1</sub> <sup>M-1</sup> , L <sub>1</sub> <sup>M</sup> ]	RN <sub>x</sub>	[t <sub>1</sub> <sup>M-1</sup> , t <sub>1</sub> <sup>M</sup> ]	C <sub>1</sub>			
			...		...			
ID <sub>i</sub>	T <sub>j</sub>	[L <sub>1</sub> <sup>1</sup> , L <sub>1</sub> <sup>2</sup> ]	RN <sub>x</sub>	[t <sub>1</sub> <sup>1</sup> , t <sub>1</sub> <sup>2</sup> ]	C <sub>p</sub>			
ID <sub>i</sub>	T <sub>j</sub>	...	...	...	C <sub>p</sub>			
ID <sub>i</sub>	T <sub>j</sub>	[L <sub>1</sub> <sup>j</sup> , L <sub>1</sub> <sup>j+1</sup> ]	RN <sub>x</sub>	[t <sub>1</sub> <sup>j</sup> , t <sub>1</sub> <sup>j+1</sup> ]	C <sub>p</sub>			
ID <sub>i</sub>	T <sub>j</sub>	...	...	...	C <sub>p</sub>			
ID <sub>i</sub>	T <sub>j</sub>	[L <sub>1</sub> <sup>N-1</sup> , L <sub>1</sub> <sup>N</sup> ]	RN <sub>x</sub>	[t <sub>1</sub> <sup>M-1</sup> , t <sub>1</sub> <sup>M</sup> ]	C <sub>p</sub>			
				...				

Table 4.2. Fact table of reconstructed trajectories of a set of objects for each criterion.

In Figure 4.10, a sample of this fact table is shown with some measures of interest. The query sentence is also shown next:

```

SELECT movingobjectid, trajectoryid, criterionid, roadnetworkid, observationinitial,
observationfinal, distance, fuelconsumption,
FROM facttrajectory
    
```

movingobject integer	trajectory integer	criterion integer	roadnetwork integer	observationinitial observation	observationfinal observation	distance double precision	fuelconsumption double precision
13307560	201408052	344038		(-75.5612667,6.1951028,20140805135854)	(-75.5611122,6.1951028,20140805135854)	0.021239435	0.0021239435
18518211	201408053	355672		(-75.5808303,6.2178794,20140805102203)	(-75.5810046,6.2178794,20140805102203)	0.0773824	0.00773824
24822919	201408052	352904		(-75.5791741,6.2494495,20140805102050)	(-75.5784825,6.2494495,20140805102050)	0.16217731	0.016217731
28763109	201408053	341805		(-75.5744685,6.2433998,20140805181452)	(-75.57341956,6.2433998,20140805181452)	0.26975274	0.026975274
32847494	201408051	354468		(-75.5724901,6.2421884,20140805122738)	(-75.5719087,6.2421884,20140805122738)	0.068614714	0.0068614714

Figure 4.10. A factTrajectory fact table example

Note that this is a simple dimensional model. The idea behind this proposal is the aggregation of each measure along each criteria and evidencing the change of the analysis if a specific criterion is considered such as the total distance when the criteria of POIs is used or the total fuel consumption if the time is considered. The measures are properties of interest about each one of the segments of the trajectories. As it is shown in *Figure 4.8*, the level of granularity, i.e., the detail of the units of data in the DW, is given by the segment between inferred observations and the time intervals determined by those observations. Note also that the aggregations of the measures have a semi-additive behavior (the measures only make sense if they are added up when this dimension is included) [82] with the criteria dimension. In *Table 4.3*, some measures of interest about trajectories are shown.

Measure	Description
Quantity of trajectories	Count all distinct trajectory ids that pass through a street segment
Quantity of users	Count all the MO IDs that pass through a street
Total Distance Traveled	Adds up the computed distance for each segment of the reconstruction trajectory. The total distance of a set the trajectories is the sum of the distance of each one.
Total Travel Duration	Adds up the computed time for each segment of the reconstruction trajectory. The total distance of a set the trajectories is the sum of the distance of each one.
Fuel consumption	Adds up the fuel consumption according to the distance traveled
CO2 emissions	Adds up the co2 emission according to the distance traveled

*Table 4.3. Some measures of interest in a TDW*

If the comparison of fuel consumption using a reconstruction criterion against another is needed, a measure of fuel consumption for each segment traveled can be defined. Of course, vehicle's fuel consumption changes according to vehicle types and other variables such as road, traffic, and weather conditions, driving style, vehicle speed, load, and condition. However, manufacturers provide average fuel consumption data. In most countries, this ratio is given in litres / 100km as the most commonly used measure of fuel consumption [107] In a most accurate way, many vehicles are fitted with a trip computer that provides an average fuel consumption function. However, for the goal of this thesis, the MOs are supposed as vehicles of the same type, i.e., they have the same fuel consumption. The fuel consumption is estimated based on the distance travelled. This method offers

a reasonably accurate means of determining actual fuel usage for a particular trip. Suppose that the set of MO analyzed here use 10 Litres / 100km. In *Figure 4.11*, fuel consumption (litres / 100km) sliced are by criteria and day between August 4, 2014 and August 10, 2014 are shown.

```

SELECT      criteria.CriterionDesc,                               TimeIni.IdDate,
            SUM(fact.FuelConsumption)
FROM        factTrajectory fact
INNER JOIN  dimCriteria criteria
ON         fact.criterionid = criteria.criterionid
INNER JOIN  dimTime TimeIni
ON        fact.(observationlinitial).t = TimeIni.IdTime
INNER JOIN  dimTime TimeFin
ON        fact.(observationlfinal).t = TimeFin.IdTime
WHERE      TimeIni.IdTime >= 20140804 AND TimeFin.IdTime <= 20140810
GROUP BY   criteria.CriterionDesc, TimeIni.iddate

```



*Figure 4.11. Fuel Consumption (litres / 100km) sliced by day and criteria*

Also, measures such as CO<sub>2</sub> emissions can be used for analysis in function of distance traveled [107]. Suppose that the set of MO analyzed here emit 300 grams of CO<sub>2</sub> per km. In *Figure 4.12*, CO<sub>2</sub> emissions (grams per km) are sliced by criteria and day between August 4, 2014 and August 10, 2014 are shown.

```

SELECT      dim.DescCriteria, TimeIni.IdDate, SUM(fact.co2emision)
FROM        factTrajectory fact
INNER JOIN  dimCriteria criteria
ON          fact.idCriteria = criteria.idCriteria
INNER JOIN  dimTime TimeIni
ON          fact.(observationlinitial).t = TimeIni.IdTime
INNER JOIN  dimTime TimeFin
ON          fact.(observationlfinal).t = TimeFin.IdTime
WHERE       TimeIni.IdTime >= 20140804
AND        TimeFin.IdTime <= 20140810
GROUP BY   criteria.DescCriteria, TimeIni.iddate

```

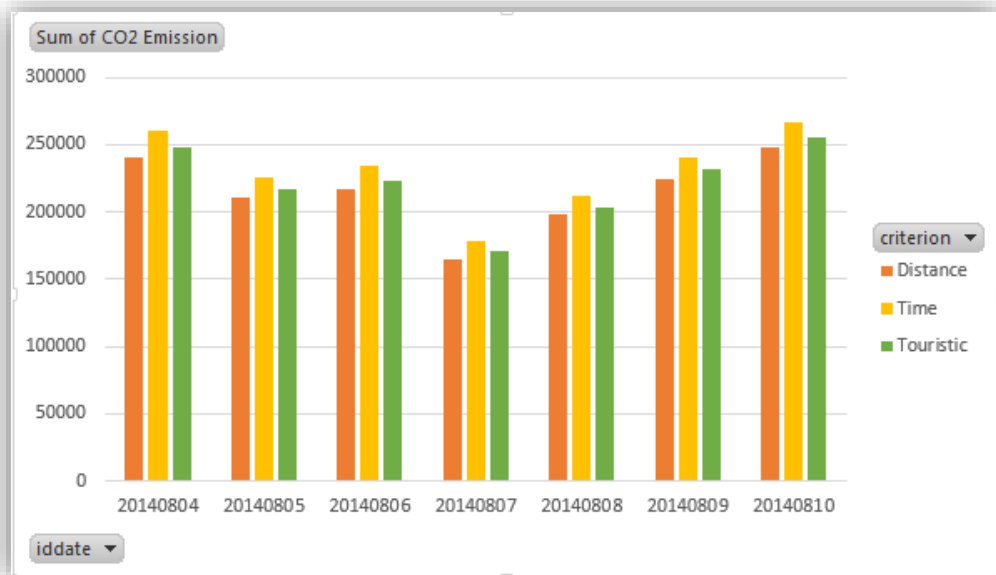


Figure 4.12. CO2 emissions (grams per km) sliced by day and criteria

As we have done with the fuel consumption and CO2 emissions, we performed a series of queries and show some results to explore the analytical possibilities in function of the criterion variation.

How many MO are traversing the “Exposiciones” roundabout in the city of the Medellín, Colombia on August 5, 2014 (*Tuesday*) between 07:00:00 am and 07:00:00 pm according to *time* criterion? The correspondent query and the resulting query answer are shown in *Figure 4.13*.

```

SELECT          COUNT(DISTINCT movingobjectid)
FROM            factTrajectory fact
INNER JOIN      dimroadnetwork rn
ON              fact.roadnetworkid = rn.roadnetworkid
INNER JOIN      dimcriteria criteria
ON              fact.criterionid = criteria.criterionid
WHERE           (fact.observationlinitial).t >= 20140805070000
AND             (fact.observationlfinal).t <= 20140805190000
AND             criteria.criterionid = 2 -- Time Criterion
AND             rn.roadnetworkdesc = 'Glorieta Exposiciones'

Answer: 21

```

*Figure 4.13. How many MO are traversing the “Exposiciones” Street in the city of the Medellín, Colombia on August 5, 2014 (Tuesday) between 07:00:00 am and 07:00:00 pm according to time criterion?*

What are the top 5 most traversed streets between 07:00:00 am and 09:00:00 pm on August 9, 2014 (Saturday) according to *touristic* criterion? The correspondent query and the resulting query answer are shown in *Figure 4.14*.



```

SELECT      rn.roadnetworkdesc
FROM        factTrajectory fact
INNER JOIN  dimroadnetwork rn
ON          fact.roadnetworkid = rn.roadnetworkid
INNER JOIN  dimcriteria criteria
ON          fact.criterionid = criteria.criterionid
WHERE       (fact.observationlinitial).t >= 20140809070000
AND         (fact.observationlfinal).t <= 20140809210000
AND         criteria.criterionid = 3 – Touristic criterion
GROUP BY   roadnetworkdesc
ORDER BY   COUNT(DISTINCT trajectoryid) DESC
LIMIT 5;

```

Order	Road Network Street
1	Avenida del Ferrocarril
2	Carrera 65
3	Calle 44
4	Carrera 43
5	Carrera 70

*Figure 4.14. What are the top 5 most used segment streets on August 9, 2014 (saturday) according to touristic criterion?*

What is the average distance travelled on August 8, 2014 (*Friday*) grouped by criteria? The correspondent query and the resulting query answer are shown in *Figure 4.15*.

<b>SELECT</b>	criteria.CriterionDesc, <b>AVG</b> (fact.distance)
<b>FROM</b>	factTrajectory fact
<b>INNER JOIN</b>	dimCriteria criteria
<b>ON</b>	fact.criterionid = criteria.criterionid
<b>INNER JOIN</b>	dimTime TimeIni
<b>ON</b>	fact.(observationlinitial).t = TimeIni.IdTime
<b>INNER JOIN</b>	dimTime TimeFin
<b>ON</b>	fact.(observationlfinal).t = TimeFin.IdTime
<b>WHERE</b>	TimeIni.IdTime >= 20140808
<b>AND</b>	TimeFin.IdTime <= 20140808
<b>GROUP BY</b>	criteria.CriterionDesc

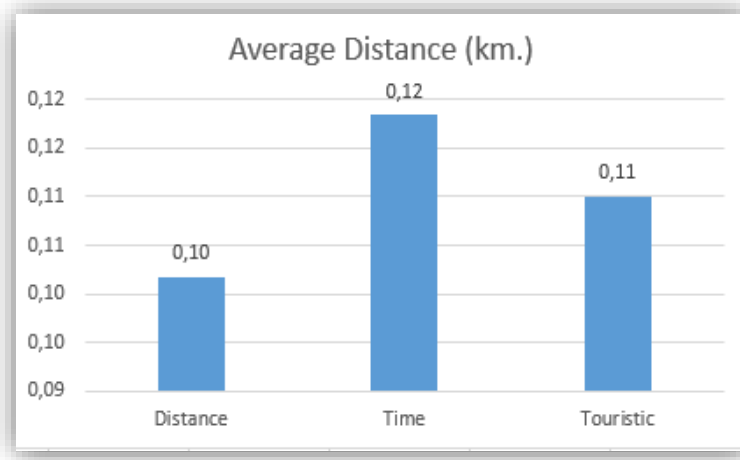


Figure 4.15. What is the average distance travelled on August 8, 2014 (Friday) sliced by criteria?

#### 4.2.2.4 Data Access

In order to present the data, we used a tool called *Quantum Gis* [108] Free and Open Source Geographic Information System application that provides data viewing, editing, and analysis capabilities. Layers of a PostGis database [103], [105] were added and drawn in a desktop platform. Figures shown in this chapter were generated with this tool.

### 4.3 CONCLUSION AND FUTURE WORK

In this chapter, we lay the groundwork for enhance the analysis of trajectories where low-sampling is present. We extend the approach proposed in *Chapter 3* for analytic tasks to find how the analysis change when the movement criterion is incorporated to reconstruct low-sampling trajectories. A complete flow of task required during a TDW developing were described.

The results shown here evidence the variation of the analysis of the imputation process when criteria of movement is considered. A simple graphical analysis can find the segments in the RN with the most concurrence of MO during a period. This approach can be useful for support decision-making in companies with location based advertising for make advertising campaigns or in tourism companies for determining the routes with the most touristic POIs visited. The analysis supported by TDW including criteria as a dimension for measuring trajectory characteristics such distance Travel distance or fuel consumption can also be helpful for companies such as logistic for expenses saving or traffic control division for determining the segments with the most MO flow.

The analysis proposed here can be enhanced when trajectories are not considered low-sampling (the ones from mobile applications such as Foursquare or Flickr), i.e., a process of integration between imputed data derived from *traj* function and most detailed information gotten from devices with higher configured sampling such as GPS loggers.

Although, a DW approach has been followed here, in the last years a new paradigm has been adopted to deal with huge amount of data: Big Data. The big data is about finding new value within and outside conventional data sources as a complementary extension to current TDW architectures to support new data types [109]. This proposal can be enhanced by Big Data techniques and data mining tasks can also be carried out over the fact table *factTrajectory* showing the variation of the mining analysis over all trajectories when the reconstruction criteria is changed.

**The main contributions of this chapter are:**

- The mapping of the *traj* function in a data warehouse architecture.
- The incorporation of user criteria as a dimension in a dimensional modelling.
- The development of a trajectory data warehouse to show the different variations of the criteria in the analysis of low-sampling trajectories.

- The different visualizations proposals of the reconstructed trajectories according to the criteria.
- This chapter develops the specific objective “*Identify opportunities of analytical tasks using an operator over low-sampling trajectories considering the limitations of Network Constrained Environment*” using a data warehouse approach.

## CHAPTER 5. TECHNICAL DETAILS.

### 5.1 INTRODUCTION

This chapter details each one of the components of the *traj* function and the *Trajectory Data Warehouse* proposal presented in the *Chapter 3* and *Chapter 4*. This technical documentation is intended to offer a more comprehensive understanding of the solution and it serves as a reference for future implementation of the system. It also pretends to provide the technical details to replicate the previously executed experiments.

With this proposal, the aiming is to create a DW based on the low-sampling trajectories reconstructed according the proposal of the *Chapter 3* and *Chapter 4*. Each detail of the TDW are explained here as well as the implementation of the *traj* function.

### 5.2 TECHNICAL REQUIREMENTS

In this section, all software are listed.

**Apigee.** The leading infrastructure for creating & operating APIs and apps [110].

**Foursquare API.** Foursquare for developers. Access to world-class places database of Foursquare. Understanding the intersection of social data and the physical world [100].

**Openstreetmap** is a map of the world free to use under an open license [102].

**Osm2po-4.8.8.** Routing On OpenStreetMap, is both, a converter and a routing engine, converter parses OpenStreetMap's XML-Data and makes it routable [101].

**Pentaho Data Integration 5.0.1:** Delivers Extraction, Transformation, and Loading (ETL) capabilities, using a groundbreaking, metadata-driven approach [106].

**PgRouting.** Extends the PostGIS / PostgreSQL geospatial database to provide geospatial routing functionality. The “cost” parameter can be dynamically calculated through SQL and its value can come from multiple fields or tables [104].

**Postgress 9.2.** An object-relational database management system (ORDBMS) [105].

**Qgis Desktop 2.0.1.** A Free and Open Source Geographic Information System. Create, edit, visualise, analyse, and publish geospatial information on Windows, Mac, Linux, BSD [108].

### 5.3 SOURCE DEFINITION

In this section, all needed sources are defined.

#### 5.3.1 Foursquare data

As it have been said before, the source data can be extracted from multiple location-based devices and applications. For this technical proposal, Json files are generated using Foursquare API [100] and then read using Pentaho Data Integration [106].

The Foursquare API has been accessed using Apigee, An API management and predictive analytics platform that helps to create and operate APIs and apps [110]. The technical details of the components of the Json file can be found in [100]. Some interesting foursquare API responses related to the thesis proposal are listed:

##### 5.3.1.1 User

Get details of the users of Foursquare (<https://developer.foursquare.com/docs/users/users>). Figure 5.1 shows an instance of this file gotten with this response. Information of the *venues* (find this file in `\sources\UsersList1.js`) registered in Foursquare in the city of Medellín, Colombia were collected.

```

1046 {
1047   "id": "28763109",
1048   "firstName": "Lorena",
1049   "lastName": "Velasco",
1050   "gender": "female",
1051   "relationship": "friend",
1052   "photo": {
1053     "prefix": "https://i.s3.amazonaws.com/user/",
1054     "suffix": "/ANE12RW4U2YLGWZ5.jpg"
1055   },
1056   "tips": {
1057     "count": 1
1058   },
1059   "lists": {
1060     "groups": [
1061     ]
1062   },
1063   "homeCity": "Medellin, Antioquia",
1064   "bio": "",
1065   "contact": {
1066     "email": "[redacted]@gmail.com",
1067     "twitter": "[redacted]",
1068     "facebook": "[redacted]"
1069   }
1070 },
1071 },
1072 },
1073 },
1074 },
1075 }

```

Figure 5.1. Example of a Json File of the user response from the API Foursquare.

### 5.3.1.2 Venues

Get details of the venues of Foursquare (The points where the people make check-in). (<https://developer.foursquare.com/docs/responses/venue>). Figure 5.2 shows an instance of the file gotten with this response. Information of 80 active random users (find this file in: `\sources\VenuesListi.js` where  $0 < i < 21$ ) living in the Medellin, Colombia city were collected.

```

14  "venues": [
15    {
16      "id": "4bd24659462cb7139e17dc07",
17      "name": "Jardin Botánico Joaquín Antonio Uribe"
18    },
19    {
20      "contact": {
21        "phone": "+5744445500",
22        "formattedPhone": "+57 4 4445500",
23        "twitter": "jbotanicomed"
24      },
25      "location": {
26        "address": "Cll. 73 # 51D - 14",
27        "lat": 6.270594259210969,
28        "lng": -75.56383609771729,
29        "distance": 0,
30        "cc": "CO",
31        "city": "Medellin",
32        "state": "Antioquia",
33        "country": "Colombia",
34        "formattedAddress": [
35          "Cll. 73 # 51D - 14",
36          "Medellin, Antioquia",
37          "Colombia"
38        ]
39      },
40      "categories": [
41        {
42          "id": "5bf58dd8d48988d15a941735",
43          "name": "Garden",
44          "pluralName": "Gardens",
45          "shortName": "Garden",
46          "icon": {
47            "prefix": "https://ssl.4sqi.net/img/categories_v2/parks_outdoors/garden_",
            "suffix": ".png"
          }
        }
      ]
    }
  ]

```

Figure 5.2. Example of a Json File of the venue response from the API Foursquare

### 5.3.1.3 Check-in

Get details of a check-in (<https://developer.foursquare.com/docs/checkins/checkins>). Figure 5.3 shows an instance of the file gotten with this response. Information of a List of check-in of the users described above were gathered during a week. A file by day was generated (find this file in `\sources\DSFoursquare201408XX.js`)

```

14  "response": {
15    "recent": [
16      {
17        "id": "53fbc881d276a4eff386ad",
18        "timestamp": 1409010376,
19        "type": "checkin",
20        "timeZoneOffset": -180,
21        "user": {
22          "id": "66384609",
23          "firstName": "acris11",
24          "gender": "male",
25          "relationship": "friend",
26          "photo": {
27            "prefix": "https://i.sq1.net/img/user/",
28            "suffix": "/66384609-LBF1XJGIRHJMQG.jpg"
29          }
30        },
31        "venue": {
32          "id": "5314b2d2e4b0d805d3740d6a",
33          "name": "San Nicol s",
34          "contact": {},
35          "location": {
36            "lat": -34.6042381883081,
37            "long": -58.38327534895184,
38            "cc": "AR",
39            "city": "Baires",
40            "state": "Buenos Aires C.F.",
41            "country": "Argentina",
42            "formattedAddress": [
43              "Baires",
44              "Buenos Aires F.D.",
45              "Argentina"
46            ]
47          }
48        }
49      }
50    ]
51  }

```

Figure 5.3. Example of a Json File of the check-in response from the API Foursquare.

### 5.3.2 Point of Interest.

A list of touristic points of Medell n, Colombia city were defined. Those were extracted from OpenStreetMap where people can tag those places as *touristic*. Find this file in `\sources\map_pois_nodes.xml`. A process of standardization and filtering where also done (find the file used in `\DBObjects\SQLsentences\CleanPOIS.sql`). See an example of this file in Figure 5.4. The location for each one was also included. The idea behind this definition is to assign a lower cost to segments of the streets near to those touristic points.





Figure 5.4. Points of Interest of the city of Medellín

### 5.3.3 The Graph Map

The Graph Map was gotten using osm2po-4.8.8 [101]. osm2po's converter parses OpenStreetMap's XML-Data and makes it routable. The OpenStreetMap of the Country of Colombia was downloaded from <http://download.geofabrik.de/south-america/colombia.html>. Find this file in: `\sources\colombia-latest.osm.pbf`. Both executable and resulting .sql file from osm2po are available in `\Software\osm2po-4.8.8`

The specifically data for Medellín, Colombia city were gotten performing geometry operation in Postgress 9.2. This .sql sentence can be found in `\DBObjects\SQLsentences\GetMedellinRN.sql`

## 5.4 STAGING DEFINITIONS

The storing data between the sources of information and a DW is done in the staging area. Next, the objects used for load the sources and the ETL process built to extract, transform (mapping and reconstruction) and load the TDW are defined.

## 5.4.1 Tables

### 5.4.1.1 colombiarn\_2po\_4pgr

The *colombiarn\_2po\_4pgr* table stores the RN of Colombia. It is the result of the process explained in section 5.3.3. The *colombiarn\_2po\_4pgr* table definition can be found in `\DBObjects\staging schema\tables\colombiarn_2po_4pgr.sql`.

### 5.4.1.2 colombiarn\_2po\_4pgr\_medellin

The *colombiarn\_2po\_4pgr\_medellin* table stores the RN definition of Medellin, Colombia. This table is the outcome of the next SQL sentence.

```
CREATE TABLE colombiarn_2po_4pgr_medellin AS
SELECT *
FROM colombiarn_2po_4pgr
WHERE ST_INTERSECTS( ST_MAKEENVELOPE(-75.6488, 6.1887, -75.5317,
6.3238,4326) , geom_way ) = TRUE
```

The longitude and latitude values are the delimiter coordinates of the Medellín city. The table definition can be found in `\DBObjects\staging schema\tables\ colombiarn_2po_4pgr_medellin.sql`. An example of the *colombiarn\_2po\_4pgr\_medellin* table is shown in Figure 5.5.

id	osm_id	osm_name	osm_meta	osm_source_id	osm_target_id	claz	flags	source	target	km	kmh	cost	reverse_cost	x1	y1	x2
integer	bigint	character varying	character varying	bigint	bigint	integer	integer	integer	integer	double precision	integer	double precision	double precision	double precision	double precision	double precision
6339	340145	301756 Calle 46, Fincinchá	430538043	332505767	21	1	250454	247026	0.35593149	60	0.0059321905	1000000	-75.595924	6.2340406	-75.51	
6340	353098	987423 Avenida Nutibara	325986078	342599353	21	1	249040	247502	0.066917464	60	0.001155291	1000000	-75.598294	6.2483559	-75.51	
6341	343473	350567 Carrera 43 A	393993406	393992180	21	1	249462	249719	0.22147106	60	0.0036931843	1000000	-75.5715867	6.2034679	-75.51	
6342	356098	108957 Carrera 30 A	416720205	416726215	21	1	252112	250072	0.24545416	60	0.004090903	1000000	-75.5597932	6.2036717	-75.51	
6343	363308	203167 Via al mar	2131466351	2131466310	13	1	263849	254127	0.08731134	90	0.000970126	1000000	-75.6190515	6.2767148	-75.61	
6344	363469	205667 Camino de Jalisco	2075052995	2111373630	42	3	264015	264016	0.79387035	30	0.025795678	0.025795678	-75.5907572	6.3043501	-75.51	
6345	341094	313122 Salida a Avenida Gua	344805030	830075410	21	1	247827	247828	0.19456777	60	0.003242796	1000000	-75.5857861	6.203573	-75.51	
6346	355743	998766 Calle 30A	359750123	321924386	31	3	248099	247957	0.14104657	40	0.0035261642	0.0035261642	-75.578238	6.2326701	-75.51	
6347	340871	309002 Carrera 43 A	1077838526	394237753	21	1	256147	249483	0.25610662	60	0.0042684437	1000000	-75.5743279	6.1976158	-75.51	
6348	359497	156522	351608274	1687606031	32	3	258917	258150	0.27180266	50	0.0054360535	0.0054360535	-75.5403489	6.2318417	-75.51	
6349	342994	343497 Carrera 43 A	393932976	1966182967	21	1	261950	263196	0.22690853	60	0.003781809	1000000	-75.5690764	6.2227149	-75.51	
6350	364334	231389 Glorieta Av. 80 con	2397786988	344809881	21	1	264847	264852	0.03174872	60	0.0005291453	1000000	-75.6018589	6.231013	-75.61	
6351	361453	173108	1839409027	583016255	31	3	262015	253753	0.020390324	40	0.00050975813	0.00050975813	-75.5341376	6.2356941	-75.51	
6352	357895	130444	1436721496	2378655326	32	3	264489	264501	0.17971042	50	0.0035942083	0.0035942083	-75.5427134	6.2267503	-75.51	
6353	354799	997568 Circular 4	377077900	415481964	31	3	249069	256682	0.15113273	40	0.0037783184	0.0037783184	-75.5923691	6.2455649	-75.51	
6354	353912	987742 Calle 38	368147192	368147200	31	3	248727	269409	0.10780882	40	0.0026952205	0.0026952205	-75.6175214	6.2520798	-75.61	
6355	340913	309547 Carrera 52	429612991	344191636	21	1	247634	247635	0.32229352	60	0.0053715585	1000000	-75.5649199	6.2720323	-75.51	
6356	341029	310101 Carrera 41	344809970	344809978	21	1	249819	247742	0.17740722	60	0.002958787	1000000	-75.6019271	6.2293189	-75.61	
6357	341868	324917 Carrera 47	365307126	365307127	31	3	256301	244653	0.20764957	40	0.0051912395	1000000	-75.5647292	6.2511872	-75.51	
6358	341076	310670 Diagonal 79	344807718	344807719	21	1	247732	263242	0.18129848	60	0.0030216414	1000000	-75.5994478	6.2341223	-75.51	
6359	343805	354775 Calle 8	393991040	416048978	31	3	249722	250002	0.09185337	40	0.0022963344	0.0022963344	-75.5710041	6.2089242	-75.51	
6360	348779	451036 Calle 8	416037579	416038448	31	3	249897	249889	0.05431561	40	0.0013578903	0.0013578903	-75.5690687	6.2075334	-75.51	
6361	361723	173395 Carrera 48A	1371904425	1370561722	32	3	262286	262287	0.09798307	50	0.0019596615	0.0019596615	-75.557668	6.2687777	-75.51	
6362	355010	986887 Calle 53	841944982	841944964	31	3	255398	255398	0.17855449	40	0.004463862	0.004463862	-75.591782	6.2649382	-75.51	

Figure 5.5. An example of “colombiarn\_2po\_4pgr\_medellin” table

### 5.4.1.3 stg\_users

The *stg\_users* table stores the content of the file of the response of the venues method of foursquare API, see *Section 5.3.2* and *Section 5.3.3*. The *stg\_users* table definition can be found in `\DBObjects\staging schema\tables\stg_users.sql`. An example of the *stg\_users* table is shown in *Figure 5.6*.

	iduser character varying(100)	firstnamesuser character varying(100)	genderuser character varying(100)	filename character varying(100)
1	80365074	Christa	female	UsersList1.js
2	11564028	Ronal	male	UsersList1.js
3	13290580	Leonardo	male	UsersList1.js
4	47810113	Sonny	male	UsersList1.js
5	44645162	Romario	male	UsersList1.js
6	59790192	Zhanna	female	UsersList1.js
7	57061671	Andres	male	UsersList1.js
8	81989400	Carolina	female	UsersList1.js
9	77286522	Chris	male	UsersList1.js
10	3417494	Jesus	male	UsersList1.js
11	7807394	Noely	female	UsersList1.js
12	22819819	Jorge HernÃn	male	UsersList1.js
13	4587343	Camilo	male	UsersList1.js
14	18041968	Juan Carlos	male	UsersList1.js
15	64380661	Jhonatan	male	UsersList1.js
16	8591004	Helena	female	UsersList1.js
17	6842900	Hector	male	UsersList1.js
18	37228698	Luis-O	male	UsersList1.js
19	5547014	MarÃa de los Ãnge	female	UsersList1.js
20	21561255	Andres	male	UsersList1.js
21	8548578	Juan Pablo	male	UsersList1.js
22	10991615	Luis Gabriel	male	UsersList1.js
23	26905168	Katta	female	UsersList1.js
24	13325839	Marco	male	UsersList1.js

Figure 5.6. An example of “*stg\_users*” table

### 5.4.1.4 stg\_venues

The *stg\_venues* table stores the content of the file of the response of the venues method of foursquare API, see *section 5.3.1.2*. The *stg\_venues* table definition can be found in `\DBObjects\staging schema\tables\stg_venues.sql`. An example of the *stg\_venues* table is shown in *Figure 5.7*.

	idvenue character varying(200)	namevenue character varying(200)	latvenue character varying(200)	lngvenue character varying(200)	categories character varying(1000)	namefile character varying(100)
26	5086d2abe4b08b9153	Kaldi kaffe	6.26871888419598	-75.56613718939887	{["id": "4bf58dd8d489	VenuesList1.js
27	51800cbe4b00ad005d	Acuario Parque Explora	6.270053496913541	-75.545499494947159	{["id": "4f0eaa171993	VenuesList1.js
28	4fea5198e4b0348f7f1	Dogger Parque de los Deseos	6.268204684065241	-75.56614855950959	{["id": "4bf58dd8d489	VenuesList1.js
29	512e13de4b03634254	Clinica LeÃn XIII Bloque 2	6.266632	-75.564169	{["id": "4bf58dd8d489	VenuesList1.js
30	5216512611d2ded2c9c	Patio de las Azaleas	6.271966	-75.563775	{["id": "4bf58dd8d489	VenuesList1.js
31	5020983fe4b05f2aebc	Viva Auditorium	6.261122405130832	-75.58960891751553	{["id": "4bf58dd8d489	VenuesList2.js
32	4dfcf64be4b0d331690	Centro Comercial El Diamante	6.260954659465193	-75.58955207599375	{["id": "4bf58dd8d489	VenuesList2.js
33	5238cfd4f93cdf599e1c	Baruc Shop	6.268852013242268	-75.58968722820282	{["id": "4bf58dd8d489	VenuesList2.js
34	51a4eeb9490e546428	Procel	6.261119	-75.589557	{["id": "4f04afcc02f6	VenuesList2.js
35	51700004e4b03aad0f2	Comprodigios S.A.S.	6.2611799240112305	-75.58960723876953	{["id": "4bf58dd8d489	VenuesList2.js
36	4e9f10b90cd61c79dc5	Variedades Luz Dary	6.261222616511686	-75.58975428342819	{["id": "4bf58dd8d489	VenuesList2.js
37	5101647be4b004f288e	Store Games	6.261003	-75.589743	{["id": "4f04afcc02f6	VenuesList2.js
38	50d0fe60e4b0d30ee15	Zimbabwe	6.261129	-75.589833	{["id": "4bf58dd8d489	VenuesList2.js
39	4f91f7e6e4b0935b82d	Los Colores - Centro Comercio	6.2609780251570495	-75.58971691634648	{}	VenuesList2.js
40	4b98ce91f964a520832	El diamante	6.260784561990819	-75.58931902552304	{["id": "4bf58dd8d489	VenuesList2.js
41	5287992311d2937bd04	Sun & Time	6.260942664426005	-75.58968186378479	{["id": "4bf58dd8d489	VenuesList2.js
42	5079868e4b0468863a	Lovyna Ink	6.2612576484468018	-75.5899658203125	{["id": "4bf58dd8d489	VenuesList2.js
43	510a7490e89cd85f67	Fruta Fresca el Diamante	6.260931999581656	-75.58978915214539	{["id": "4bf58dd8d489	VenuesList2.js
44	502590f9e4b0a90de1d	RelojerÃa Tino	6.261258	-75.589853	{["id": "4bf58dd8d489	VenuesList2.js
45	512572b0e4b0ca39e9e	Change The Look	6.260861	-75.589531	{["id": "4bf58dd8d489	VenuesList2.js
46	4bd1f19e0caff9521d4f	Souffish. C.C El Diamante	6.261379922855892	-75.58924889685059	{["id": "4bf58dd8d489	VenuesList2.js
47	50258cde4b04e431b5	Electro Horizonte	6.261442	-75.58976	{["id": "4e41bee8e3b7b	VenuesList2.js
48	4dc4dcf1922ef0b1125	Viva	6.261102637064758	-75.58988571166992	{["id": "4bf58dd8d489	VenuesList2.js
49	4e2b19247db7deda6c	Barrio El Estadio	6.260208674462219	-75.58760808274357	{["id": "4f2a25ac4b90	VenuesList2.js

Figure 5.7. An example of “*stg\_venues*” table

### 5.4.1.5 stg\_check\_in\_data

The *stg\_check\_in\_data* table stores the content of the file of the response of the check-ins method of foursquare API, see *Section 5.3.1.3*. The *stg\_check\_in\_data* table definition can be found in `\DBObject\staging schema\tables\stg_check_in_data.sql`. An example of the *stg\_check\_in\_data* table is shown in *Figure 5.8*.

objectid	trajectoryid	integer	poi_name	x	y	t	point geometry
9	3417494	20140805	watusai	-75.59740168237609	6.238995672851286	2014-08-05 12:53:28	0101000020E61000005E
10	3417494	20140805	Teatro Metropolitan	-75.57734769350883	6.242355729271313	2014-08-05 14:58:44	0101000020E6100000A7
11	3417494	20140805	Homeplastic	-75.579049	6.231261	2014-08-05 10:00:41	0101000020E610000004
12	3417494	20140805	Plaza Botero	-75.56829929951807	6.252122763815575	2014-08-05 13:17:02	0101000020E61000000C
13	3417494	20140805	Liga Antioqueña de Gimnasia	-75.58878326416016	6.25616029244626	2014-08-05 16:49:53	0101000020E61000000C
14	4494609	20140805	Jc Delicias C.C. Santafe	-75.57451103889267	6.19594023049268	2014-08-05 14:16:32	0101000020E61000000A
15	4494609	20140805	Parque De Los Pies Descalzos	-75.57679653167725	6.214806506085907	2014-08-05 12:37:32	0101000020E61000000C
16	4494609	20140805	Change The Look	-75.589531	6.260861	2014-08-05 10:39:00	0101000020E61000000A
17	4654709	20140805	Gimnasio Uros	-75.58183140381538	6.257561580985405	2014-08-05 06:44:02	0101000020E61000000C
18	4654709	20140805	Laguna Jardín Botánico	-75.56365848438433	6.271508876651686	2014-08-05 08:15:29	0101000020E61000000C
19	5018348	20140805	Premios Medellín La Más Educada	-75.5773687	6.2428115	2014-08-05 14:39:42	0101000020E61000000F
20	5018348	20140805	Parque De Malibá	-75.58842660934471	6.237365244921854	2014-08-05 13:49:14	0101000020E61000000D
21	5018348	20140805	Kokoriko	-75.56032694841524	6.196198104819887	2014-08-05 06:22:45	0101000020E61000000B
22	5367615	20140805	Fase II	-75.57087276479228	6.230358150747934	2014-08-05 17:23:34	0101000020E61000007C
23	5367615	20140805	COOSALUD EPS-S	-75.58835839896834	6.250688134559359	2014-08-05 18:20:55	0101000020E610000057
24	5367615	20140805	II Foro Base Internacional 2013	-75.576759	6.243269	2014-08-05 13:26:40	0101000020E61000001E
25	5367615	20140805	Los Asados	-75.57679059931037	6.2444064091954585	2014-08-05 12:35:49	0101000020E61000006F
26	5547014	20140805	watusai	-75.59740168237609	6.238995672851286	2014-08-05 19:22:55	0101000020E61000005E
27	5547014	20140805	Mundo Verde El Tesoro	-75.560248	6.196482	2014-08-05 17:47:21	0101000020E61000009C
28	5547014	20140805	Papelería del bloque m9	-75.5924301147461	6.273852348327637	2014-08-05 06:03:46	0101000020E61000000C
29	5914689	20140805	Exito Cafetero	-75.58680660437278	6.238240576621373	2014-08-05 16:42:02	0101000020E6100000EE

Figure 5.8. An example of “*stg\_check\_in\_data*” table

### 5.4.1.6 stg\_pois

The *stg\_pois* table stores the content of the most representative POIs of the city of Medellín. See *Section 5.3.2*. The *stg\_pois* table definition can be found in `\DBObject\staging schema\tables\stg_pois.sql`. An example of the *stg\_pois* table is shown in *Figure 5.9*.

	nodoid text	nodolat text	nodolon text	nodotagk text	nodotagv text
6	339473513	6.2117445	-75.5746838	name	Exito Poblado
7	339473860	6.2115205	-75.5794060	name	Politécnico Colombia
8	339474570	6.2034549	-75.5766139	amenity	hospital
9	339905710	6.2359141	-75.5698291	name	Exito San Diego
0	339907169	6.2539681	-75.5638982	amenity	place of worship
1	339907385	6.2523129	-75.5690309	name	Museo de Antioquia
2	339907744	6.2463405	-75.5730993	amenity	library
3	339908706	6.2497789	-75.5856821	name	MAKRO San Juan
4	339908843	6.2514341	-75.5850984	name	Jumbo La 65
5	339909492	6.2496253	-75.5838882	name	Homecenter de La 65
6	339909792	6.2566898	-75.5900680	leisure	stadium
7	339909933	6.2552053	-75.5894672	leisure	sports centre
8	339916589	6.2321537	-75.6036490	name	EXITO Los Molinos
9	342167168	6.2390909	-75.6034694	amenity	place of worship
0	342167558	6.2449354	-75.6087222	amenity	place of worship
1	343456460	6.2363521	-75.5729029	amenity	place of worship
2	343527819	6.2292869	-75.5707480	name	Jumbo
3	343601473	6.2509391	-75.6046495	amenity	place of worship
4	344799743	6.2443382	-75.5735530	alt name	Medellín Capital
5	344924583	6.2011113	-75.5783886	amenity	library
6	358107931	6.2516598	-75.5681261	name	Palacio de la Cultur
7	358111823	6.2498851	-75.5674395	amenity	place of worship

Figure 5.9. An example of “stg\_pois” table

### 5.4.1.7 stg\_pointprojection

The *stg\_pointprojection* table stores the result of the *vw\_stg\_pointprojection* view. It is intended to be used as an input with all projected check-in for the reconstruction process. This table stores information of *get\_edge*, *get\_vertex\_source*, *get\_vertex\_target*, *get\_x* and *get\_y* functions. The *stg\_pointprojection* table definition can be found in `DBObjects\staging schema\tables\stg_pointprojection.sql`. An example of the *stg\_pointprojection* table is shown in Figure 5.10.

objectid	trajectoryid	integer	poi name	x	y	t	timestamp	point geometry	row_n	nearestedgeid	pointsource	pointtarget	proxy	double precision	source	target	pointprojection	dist
1	1330756	20140805	Corral Gourmet Teso	-75.560	6.19612	2014-08-05	14:03:33	01010000:88	344045	01010000:20	01010000:20	-75.560440262	6.19621737114	250152	250151	01010000:20E61.35.		
2	8548578	20140805	Rotiseria La nueva	-75.555	6.26936	2014-08-05	19:42:11	01010000:50	348087	01010000:20	01010000:20	-75.555017266	6.26929327661	252845	265661	01010000:20E61.16.		
3	7807394	20140805	Arturo Contador	-75.589	6.26118	2014-08-05	13:35:20	01010000:41	342211	01010000:20	01010000:20	-75.589210309	6.26044622816	248738	248739	01010000:20E61.40.		
4	4915818	20140805	Parmessano santafe	-75.574	6.19592	2014-08-05	11:19:56	01010000:182	342110	01010000:20	01010000:20	-75.57448769	6.19589640858	248643	248634	01010000:20E61.42.		
5	4464516	20140805	Plaza De Toros La M	-75.580	6.24947	2014-08-05	07:22:41	01010000:166	341391	01010000:20	01010000:20	-75.579632645	6.24945970193	248053	248054	01010000:20E61.140		
6	4915818	20140805	South Cafe	-75.570	6.23001	2014-08-05	10:31:11	01010000:180	340856	01010000:20	01010000:20	-75.570724110	6.23004297239	247594	247587	01010000:20E61.157		
7	5818679	20140805	McDonald's Postres	-75.593	6.27465	2014-08-05	10:43:01	01010000:190	350305	01010000:20	01010000:20	-75.594246006	6.27524025914	254460	254463	01010000:20E61.63.		
8	7029595	20140805	Carnes & Vinos	-75.567	6.20966	2014-08-05	07:51:34	01010000:35	343253	01010000:20	01010000:20	-75.567756647	6.20953634245	249980	253352	01010000:20E61.23.		
9	1329058	20140805	Teatro Metropolitan	-75.577	6.24235	2014-08-05	06:51:09	01010000:82	341815	01010000:20	01010000:20	-75.577299421	6.24221616252	248437	248441	01010000:20E61.52.		
10	8620229	20140805	Santa bebeta	-75.597	6.23895	2014-08-05	08:37:48	01010000:94	342604	01010000:20	01010000:20	-75.597899421	6.23893040457	249119	249120	01010000:20E61.139		
11	4915818	20140805	la cava del brangus	-75.597	6.23896	2014-08-05	08:16:40	01010000:179	342604	01010000:20	01010000:20	-75.597297262	6.23894653351	249119	249120	01010000:20E61.72.		
12	8591004	20140805	Plaza De Toros La M	-75.580	6.24947	2014-08-05	16:14:56	01010000:52	341391	01010000:20	01010000:20	-75.579632645	6.24945970193	248053	248054	01010000:20E61.140		
13	1099121	20140805	Los Petros de Buler	-75.596	6.23813	2014-08-05	18:57:16	01010000:69	350758	01010000:20	01010000:20	-75.59629547	6.23812421170	254909	254908	01010000:20E61.21.		
14	5367615	20140805	Hamburguesas El Gar	-75.560	6.19648	2014-08-05	10:36:04	01010000:25	344045	01010000:20	01010000:20	-75.560476688	6.19655314007	250152	250151	01010000:20E61.76.		
15	1035106	20140805	Magifoto	-75.567	6.20973	2014-08-05	19:46:07	01010000:65	343253	01010000:20	01010000:20	-75.567683769	6.20950709914	249980	253352	01010000:20E61.32.		
16	1099121	20140805	Santa bebeta	-75.597	6.23895	2014-08-05	10:54:19	01010000:67	342604	01010000:20	01010000:20	-75.597899421	6.23893040457	249119	249120	01010000:20E61.139		
17	5979019	20140805	Coliseo Iván de Be	-75.588	6.25628	2014-08-05	16:09:19	01010000:194	355059	01010000:20	01010000:20	-75.587368930	6.25596059015	250456	250453	01010000:20E61.229		
18	7581892	20140805	Afuera de Vertigo	-75.597	6.23909	2014-08-05	10:34:43	01010000:215	342604	01010000:20	01010000:20	-75.597919657	6.2389298256	249119	249120	01010000:20E61.141		
19	3417494	20140805	Los Asados	-75.565	6.26788	2014-08-05	07:46:39	01010000:14	344431	01010000:20	01010000:20	-75.565722046	6.2678490615	247635	250402	01010000:20E61.151		
20	1776633	20140805	Cafetto	-75.610	6.23278	2014-08-05	06:11:22	01010000:101	364209	01010000:20	01010000:20	-75.610706899	6.2328171120	256976	264750	01010000:20E61.109		
21	4915818	20140805	Museo de la Ciudad	-75.579	6.23609	2014-08-05	12:38:21	01010000:183	337767	01010000:20	01010000:20	-75.580095084	6.23578564677	255474	245272	01010000:20E61.234		
22	7728652	20140805	CEI (Centro de Even	-75.560	6.19656	2014-08-05	12:10:46	01010000:221	344045	01010000:20	01010000:20	-75.560454686	6.19659830339	250152	250151	01010000:20E61.82.		
23	6587343	20140805	Rur1ev	-75.560	6.19633	2014-08-05	13:54:40	01010000:31	344045	01010000:20	01010000:20	-75.560524303	6.19645540066	250152	250151	01010000:20E61.64.		

Figure 5.10. An example of “stg\_pointprojection” table

### 5.4.1.8 stg\_notfoundroutes

The *stg\_notfoundroutes* table is an auxiliary table used for storing the not found routes.

## 5.4.2 Views

Some staging tables were mapped in to a view respectively. The *Table 5.1* show the respective table, the view assigned and the source file to be executed.

Table	view	source
stg_users	vw_stg_users	<i>\DBObjects\staging schema\views\vw_stg_users.sql</i>
stg_venues	vw_stg_venues	<i>\DBObjects\staging schema\views\vw_stg_venues.sql</i>
stg_check_in_data	vw_stg_checkindata	<i>\DBObjects\staging schema\views\vw_stg_checkindata.sql</i>
stg_pois	vw_stg_pois	<i>\DBObjects\staging schema\views\vw_stg_pois.sql</i>
colombiarn_2po_4pgr_medellin	vw_roadnetwork	<i>\DBObjects\staging schema\views\vw_roadnetwork.sql</i>

*Table 5.1. Staging schema views*

### 5.4.2.1 vw\_stg\_setTouristicCost

The *vw\_stg\_setTouristicCost* view finds every nearest edge to the POIs defined in table *stg\_pois*.

### 5.4.2.2 vw\_stg\_pointprojection

The *vw\_stg\_pointprojection* view implements implicitly the *get\_edge*, *get\_vertex\_source*, *get\_vertex\_target*, *get\_x* and *get\_y* functions joining the set of check in data stored in the *stg\_check\_in\_data* table the the RN network data of the *vw\_roadnetwork* view. The view definition can be found in *\DBObjects\STAGING schema\views\vw\_stg\_pointprojection.sql*

For a set of check-in data the *get\_edge* function finds its nearest edge of the RN as presented in *section 3.3* of the *Chapter 3* but setting a rectangle around the check-in point with a longitude of 0.0025 units from the check-in point. It is done to reduce the searching time of the possible nearest edges.

### **5.4.2.3 vw\_settime**

The *vw\_settime* view implements implicitly the proportional assignation of the time of the reconstructed trajectories proposed by *set\_time* function. The view definition can be found in `\DBObjects\STAGING schema\views\vw_settime.sql`

## **5.4.3 Functions/Procedures**

### **5.4.3.1 set\_pointprojection**

The *set\_pointprojection* function inserts the outcome of *vw\_stg\_pointprojection* view. The definition of the *set\_pointprojection* function can be found in `\DBObjects\STAGING schema\functions\set_pointprojection.sql`

### **5.4.3.2 set\_costforcriteria**

The *set\_costforcriteria* function uses the *vw\_stg\_setTouristicCost* view to set a lower cost for the edges nearest to each point defined in the *stg\_pois* table. The definition of the *set\_costforcriteria* function can be found in `\DBObjects\STAGING schema\functions\set_costforcriteria.sql`

### **5.4.3.3 traj**

The *traj* function is implemented implicitly and carries out the reconstruction task proposed by this thesis. The definition of the *traj* function can be found in `\DBObjects\STAGING schema\functions\traj.sql`

### **5.4.3.4 set\_time**

The *set\_time* function implements the function *set\_time* proposed by this thesis. The definition of the *set\_time* function can be found in `\DBObjects\STAGING schema\functions\set_time.sql`

### **5.4.3.5 load\_dimRoadNetwork**

The *load\_dimRoadNetwork* function loads the *dimroadnetwork* dimension. The definition of the *load\_dimRoadNetwork* function can be found in `\DBObjects\STAGING schema\functions\load_dimroadnetwork.sql`

#### **5.4.3.6 load\_dimcriteria**

The *load\_dimcriteria* function loads the *dimcriteria* dimension. The definition of the *load\_dimcriteria* function can be found in `\DBObjects\STAGING schema\functions\load_dimcriteria.sql`

#### **5.4.3.7 load\_dimmovingobject**

The *load\_dimmovingobject* function loads the *dimmovingobject* dimension. The definition of the *load\_dimmovingobject* function can be found in `\DBObjects\STAGING schema\functions\load_dimmovingobject.sql`

#### **5.4.3.8 load\_dimtrajectory**

The *load\_dimtrajectory* function loads the *dimtrajectory* dimension. The definition of the *load\_dimtrajectory* function can be found in `\DBObjects\STAGING schema\functions\load_dimtrajectory.sql`

#### **5.4.3.9 load\_facttrajectory**

The *load\_facttrajectory* function loads the *facttrajectory* fact table. The definition of the *load\_facttrajectory* function can be found in `\DBObjects\STAGING schema\functions\load_facttrajectory.sql`

### **5.5 DATA WAREHOUSE DEFINITIONS**

Next Database tables that make up the dimensional model of the TDW are listed and defined. Also, auxiliary views are also shown.

#### **5.5.1 Types**

##### **5.5.1.1 Observation**

The *observation* type describes the observation. It is composed by longitude x, latitude y, and a timestamp t. The definition of the *observation* type can be found in `\DBObjects\TDW schema\types\Obervation.sql`



## 5.5.2 Tables

### 5.5.2.1 Facttrajectory

The *factTrajectory* table is the fact table of the dimensional model of the TDW. It stores the facts of the reconstructed trajectory, i.e., the outcome of the transformations and computations in the staging area. Measures are reported by trajectory section between imputed observations. See *Section 4.2.2*. The *factTrajectory* table definition can be found in `\DBObjects\TDW schema\tables\factTrajectory.sql`. An example of the *factTrajectory* table is shown in *Figure 5.11*.

	movingobjectid integer	trajectoryid integer	row_ni integer	row_nf integer	criterionid integer	observationinitial observation	observationfinal observation	distance double precision	traveltime double precision	fuelconsumption double precision	co2emision double precision	pointinitial geometry	pointfinal geometry	edge geometry
1	327199	20140804	1	2	1	(-75.6098195, 6.	(-75.610434172, 6.	0.10325823	452.385448	0.010325823	30.977469	010100002	010100000	010200000
2	327199	20140804	1	2	1	(-75.6097124, 6.	(-75.6098195, 6.	0.10325823	652.844306	0.010325823	30.977469	010100002	010100000	010200000
3	327199	20140804	1	2	1	(-75.6088988, 6.	(-75.6097124, 6.	0.0901086	569.706515	0.00901086	27.03258	010100002	010100000	010200000
4	327199	20140804	1	2	1	(-75.6084269, 6.	(-75.6088988, 6.	0.05220371	330.054998	0.005220371	15.661113	010100002	010100000	010200000
5	327199	20140804	1	2	1	(-75.6080281, 6.	(-75.6084269, 6.	0.044099055	278.813775	0.0044099055	13.2297165	010100002	010100000	010200000
6	327199	20140804	1	2	1	(-75.6076184, 6.	(-75.6080281, 6.	0.045304433	286.43471	0.0045304433	13.5913299	010100002	010100000	010200000
7	327199	20140804	1	2	1	(-75.6075652, 6.	(-75.6076184, 6.	0.07815804	494.149777	0.007815804	23.447412	010100002	010100000	010200000
8	327199	20140804	1	2	1	(-75.6067459, 6.	(-75.6075652, 6.	0.054238804	342.921764	0.0054238804	16.2716412	010100002	010100000	010200000
9	327199	20140804	1	2	1	(-75.6067459, 6.	(-75.6075652, 6.	0.090707935	573.495777	0.0090707935	27.2123805	010100002	010100000	010200000
10	327199	20140804	1	2	1	(-75.6041245, 6.	(-75.6067459, 6.	0.28874943	1825.601903	0.028874943	86.624829	010100002	010100000	010200000
11	327199	20140804	1	2	1	(-75.6023292, 6.	(-75.6041245, 6.	0.19941914	1260.816208	0.019941914	59.825742	010100002	010100000	010200000
12	327199	20140804	1	2	1	(-75.6022169, 6.	(-75.6023292, 6.	0.015484639	97.900753	0.0015484639	4.6453917	010100002	010100000	010200000
13	327199	20140804	1	2	1	(-75.6021429, 6.	(-75.6022169, 6.	0.008405394	53.142627	0.0008405394	2.5216182	010100002	010100000	010200000
14	327199	20140804	1	2	1	(-75.601995, 6.2	(-75.6021429, 6.	0.016812392	106.295395	0.0016812392	5.0437176	010100002	010100000	010200000
15	327199	20140804	1	2	1	(-75.6019193, 6.	(-75.601995, 6.2	0.010510839	66.454184	0.0010510839	3.1532517	010100002	010100000	010200000
16	327199	20140804	1	2	1	(-75.6006465, 6.	(-75.6019193, 6.	0.14205785	898.152704	0.014205785	42.617355	010100002	010100000	010200000
17	327199	20140804	1	2	1	(-75.5987791, 6.	(-75.6006465, 6.	0.2066758	1306.69603	0.02066758	62.00274	010100002	010100000	010200000
18	327199	20140804	1	2	1	(-75.5961991, 6.2	(-75.5987791, 6.	0.28548768	1804.979674	0.028548768	85.646304	010100002	010100000	010200000
19	327199	20140804	1	2	1	(-75.595378, 6.2	(-75.5961991, 6.2	0.09079215	574.028221	0.009079215	27.237645	010100002	010100000	010200000
20	327199	20140804	1	2	1	(-75.5936058, 6.	(-75.595378, 6.2	0.19603693	1239.432378	0.019603693	58.811079	010100002	010100000	010200000

Figure 5.11. An example of the factTrajectory fact table

### 5.5.2.2 Dimroadnetwork

The *Dimroadnetwork* table stores the information about the RN here (Medellín, Colombia). Each trajectory segment can be mapped in to a RN Segment. The *Dimroadnetwork* table definition can be found in `\DBObjects\TDW schema\tables\dimroadnetwork.sql`. An example of the *Dimroadnetwork* dimension table is shown in *Figure 5.12*.

roadnetworkid	roadnetworkdesc	roadnetworkgeom
5713	339980 Carrera 92	0102000020E61000000200
5714	339981 Carrera 89	0102000020E61000000200
5715	339982 Calle 34 F	0102000020E61000000200
5716	339983 Carrera 91	0102000020E61000000200
5717	339984 Calle 34 EE	0102000020E61000000200
5718	339985 Calle 34 E	0102000020E61000000300
5719	339986 Carrera 89	0102000020E61000000400
5720	340140 Calle 4B, Pichincha	0102000020E61000000200

Figure 5.12. An example of the *dimroadnetwork* dimension table

### 5.5.2.3 Dimcriteria

The *DimCriteria* table is the criteria dimension. This is an essential table in the TDW analysis. It stores the criteria of reconstruction and it distinguish each distinct reconstructed trajectory according to criteria. The *DimCriteria* table definition can be found in `\DBObjects\TDW schema\tables\dimCriteria.sql`. An example of the *dimCriteria* dimension table is shown in Figure 5.13.

criterionid	criteriondesc
1	1 Distance
2	2 Time
3	3 Touristic
4	4 Turns

Figure 5.13. An example of the *dimcriteria* dimension table

### 5.5.2.4 Dimtrajectory

The *Dimtrajectory* table stores the information about the whole trajectories registered here. For the simplicity of the problem addressed here, we set the trajectoryid identifier according each day, i.e., each day, a user makes a different trajectory. The *Dimtrajectory* table definition can be found in `\DBObjects\TDW schema\tables\dimtrajectory.sql`. An example of the *Dimtrajectory* dimension table is shown in Figure 5.14.

The screenshot shows a SQL Editor window with a query: `SELECT * FROM dimTrajectory`. The output pane displays the following data:

trajectoryid	trajectorydesc
integer	character varying(100)
1	Trajectory of Tuesday, 5th August of 2014
2	Trajectory of Monday, 4th August of 2014
3	Trajectory of Sunday, 10th August of 2014
4	Trajectory of Saturday, 9th August of 2014
5	Trajectory of Thursday, 7th August of 2014
6	Trajectory of Wednesday, 6th August of 2014
7	Trajectory of Friday, 8th August of 2014

Figure 5.14. An example of the *dimTrajectory* dimension table

### 5.5.2.5 DimMovingObject

The *DimMovingObject* table stores the information about the whole MOs (users) considered here. The *DimMovingObject* table definition can be found in `\DBObjects\TDW schema\tables\DimMovingObject.sql`. An example of the *DimMovingObject* dimension table is shown in Figure 5.15.

The screenshot shows a SQL Editor window with a query: `SELECT * FROM dimmovingobject`. The output pane displays the following data:

movingobjectid	movingobjectdesc
integer	character varying(100)
1	Diego
2	Johan
3	Bruno
4	Davas
5	Loisna
6	Alexandro
7	Leonardo
8	Gio
9	Henl
10	Victor
11	Camilo

Figure 5.15. An example of the *dimMovingObject* dimension table

### 5.5.2.6 Dimtime

The *Dimtime* table stores the information about the kind of timestamps considered here. The level of granularity is seconds. Each timestamp has the YYYYMMDDHHMMSS format. The *Dimtime* table definition can be found in `\DBObjects\TDW schema\tables\dimtime.sql`

## 5.5.3 Views

### 5.5.3.1 vw\_reconstructedtrajectory\_congestion

The *vw\_reconstructedtrajectory\_congestion* view lets to rate the road segments with the most trajectories traversing them. The *vw\_reconstructedtrajectory\_congestion* view definition can be found in `\DBObjects\TDW\schema\tables\vw_reconstructedtrajectory_congestion.sql`

## 5.6 ETL PROCESS

For implementing the ETL process we use Data Integration of the Pentaho suite [106]. Next the orchestration of the load of a low-sampling trajectory dataset is documented:

### 5.6.1 Jobs

#### 5.6.1.1 Job: jobPrincipal

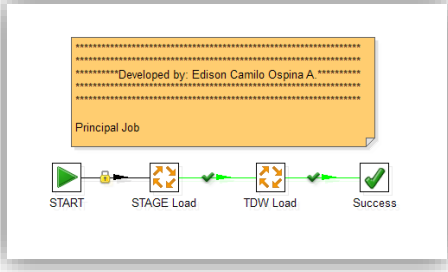
<b>File</b>	\ETL Solution\jobs\jobPrincipal.kbj
<b>Name</b>	jobPrincipal
<b>Description</b>	Principal Job than orchestrates the Stage loading and TDW loading together.
	

Table 5.2. Job: jobPrincipal

#### 5.6.1.2 Job: jobLoadStage

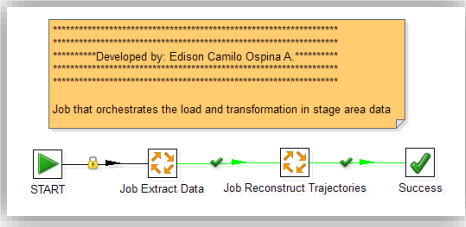
<b>File</b>	\ETL Solution\jobs\jobLoadStage.kbj
<b>Name</b>	jobLoadStage
<b>Description</b>	Job that orchestrates the load and transformation in the stage area data
	

Table 5.3. Job: jobLoadStage

### 5.6.1.3 Job: jobLoadTDW

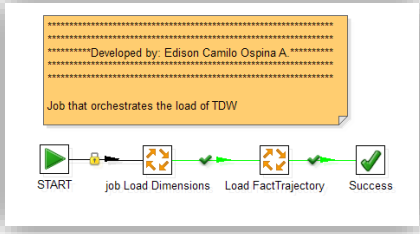
<b>File</b>	\\ETL Solution\jobs\jobLoadTDW.kbj
<b>Name</b>	jobLoadTDW
<b>Description</b>	Job that orchestrates the load of TDW
	

Table 5.4. Job: jobLoadTDW

### 5.6.1.4 Job: jobExtractData

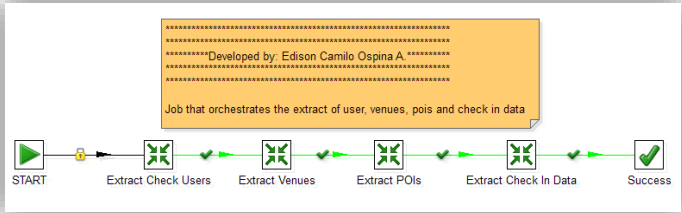
<b>File</b>	\\ETL Solution\jobs\ jobExtractData.kbj
<b>Name</b>	jobExtractData
<b>Description</b>	Job that orchestrates the extract of user, venues, pois and check in data
	

Table 5.5. Job: jobExtractData

**5.6.1.5 Job: jobReconstructTrajectories**

<b>File</b>	\\ETL Solution\jobs\jobReconstructTrajectories.kbj
<b>Name</b>	jobReconstructTrajectories
<b>Description</b>	Job that orchestrates the reconstruction of trajectories

Table 5.6. Job: jobReconstructTrajectories

**5.6.1.6 Job: jobLoadDimensions**

<b>File</b>	\\ETL Solution\jobs\ jobLoadDimensions.kbj
<b>Name</b>	jobLoadDimensions
<b>Description</b>	Job that orchestrates the load of Trajectory Datawarehouse Dimensions

Table 5.7. Job: jobLoadDimensions

### 5.6.1.7 Job: jobLoadFactTrajectory

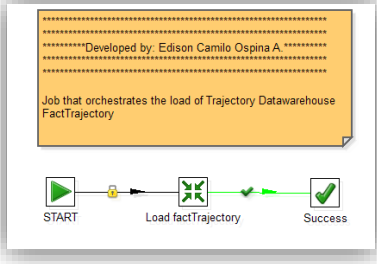
<b>File</b>	\\ETL Solution\jobs\jobLoadFactTrajectory.kbj
<b>Name</b>	jobLoadFactTrajectory
<b>Description</b>	Job that orchestrates the load of TDW FactTrajectory
	

Table 5.8. Job: jobLoadFactTrajectory

## 5.6.2 Transformations

### 5.6.2.1 Transformation: traExtractUserData

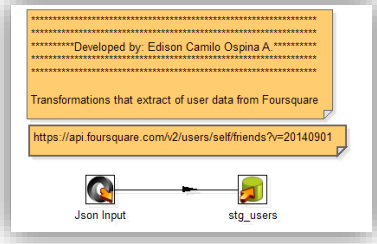
<b>File</b>	\\ETL Solution\transformations\traExtractUserData.ktr
<b>Name</b>	traExtractUserData
<b>Description</b>	Transformation that load the user data from foursquare
	

Table 5.9. Transformation: traExtractUserData

### 5.6.2.2 Transformation: traExtractVenuesData

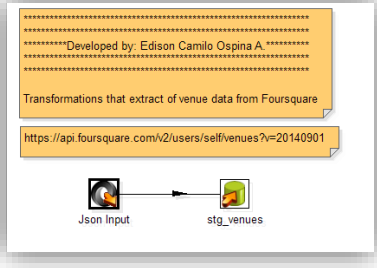
<b>File</b>	\ETL Solution\transformations\traExtractVenuesData.ktr
<b>Name</b>	traExtractVenuesData
<b>Description</b>	Transformation that load the venues data from foursquare
	

Table 5.10. Transformation: traExtractVenuesData

### 5.6.2.3 Transformation: traExtractCheckinData

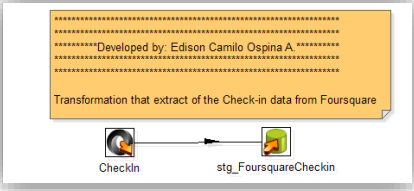
<b>File</b>	\ETL Solution\transformations\traExtractCheckinData.ktr
<b>Name</b>	traExtractCheckinData
<b>Description</b>	Transformation that load the check-in data from Foursquare
	

Table 5.11. Transformation: traExtractCheckinData



### 5.6.2.4 Transformation: traExtractPOIData

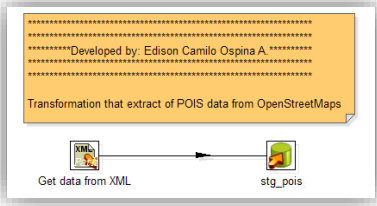
<b>File</b>	\\ETL Solution\transformations\traExtractPOIData.ktr
<b>Name</b>	traExtractPOIData
<b>Description</b>	Transformation that load the POI data from OpenstreetMap
	

Table 5.12. Transformation: traExtractPOIData

### 5.6.2.5 Transformation: traSetCostforCriteria

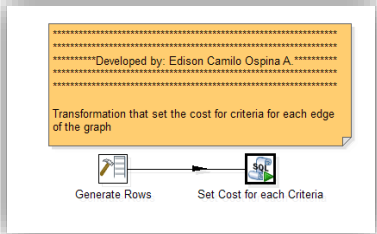
<b>File</b>	\\ETL Solution\transformations\ traSetCostforCriteria.ktr
<b>Name</b>	traSetCostforCriteria
<b>Description</b>	Transformation set the Cost for Criteria in the RN
	

Table 5.13. Transformation: traSetCostforCriteria

### 5.6.2.6 Transformation: traSetPointProjection

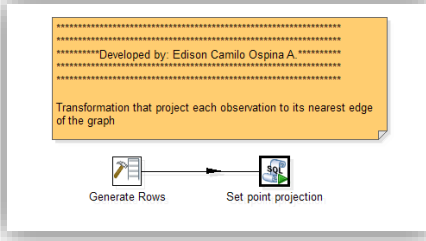
<b>File</b>	\ETL Solution\transformations\traSetPointProjection.ktr
<b>Name</b>	traSetPointProjection
<b>Description</b>	Transformation that finds the nearest edge
	

Table 5.14. Transformation: traSetPointProjection

### 5.6.2.7 Transformation: traSetPointProjection

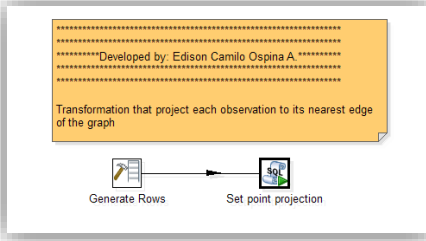
<b>File</b>	\ETL Solution\transformations\traSetPointProjection.ktr
<b>Name</b>	traSetPointProjection
<b>Description</b>	Transformation that finds the nearest edge
	

Table 5.15. Transformation: traSetPointProjection

### 5.6.2.8 Transformation: traReconstructTrajectory

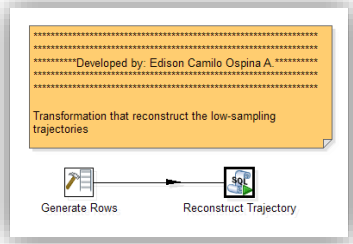
<b>File</b>	\\ETL Solution\transformations\traReconstructTrajectory.ktr
<b>Name</b>	traReconstructTrajectory
<b>Description</b>	Transformation that implements the reconstruction of trajectories. It calls the traj function.
	

Table 5.16. Transformation: traReconstructTrajectory

### 5.6.2.9 Transformation: traLoadDimRoadNetwork

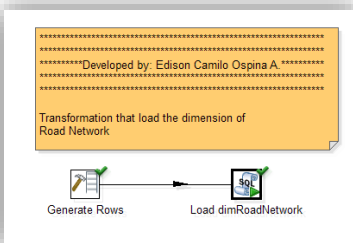
<b>File</b>	\\ETL Solution\transformations\traLoadDimRoadNetwork.ktr
<b>Name</b>	traLoadDimRoadNetwork
<b>Description</b>	Transformation that load the dimension of Road Network
	

Table 5.17. Transformation: traLoadDimRoadNetwork

### 5.6.2.10 Transformation: traLoadDimCriterion

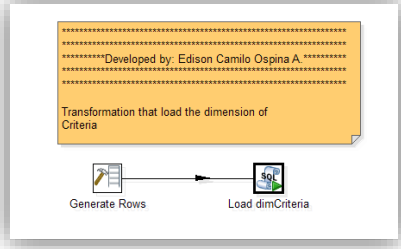
<b>File</b>	\\ETL Solution\transformations\ traLoadDimCriterion.ktr
<b>Name</b>	traLoadDimCriterion
<b>Description</b>	Transformation that load the dimension of Criteria
	

Table 5.18. Transformation: traLoadDimCriterion

### 5.6.2.11 Transformation: traLoadDimMovingObject

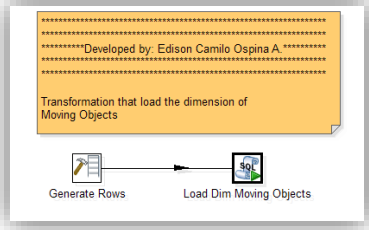
<b>File</b>	\\ETL Solution\transformations\traLoadDimMovingObject.ktr
<b>Name</b>	traLoadDimMovingObject
<b>Description</b>	Transformation that load the dimension of Moving Objects
	

Table 5.19. Transformation: traLoadDimMovingObject

### 5.6.2.12 Transformation: traLoadDimTrajectory

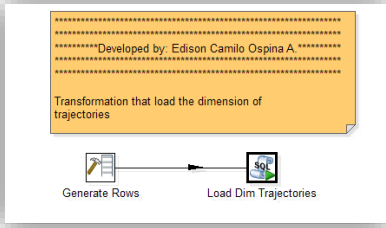
<b>File</b>	\\ETL Solution\transformations\traLoadDimTrajectory.ktr
<b>Name</b>	traLoadDimTrajectory
<b>Description</b>	Transformation that load the dimension of Trajectories
	

Table 5.20. Transformation: traLoadDimTrajectory

### 5.6.2.13 Transformation: traLoadFactTrajectory

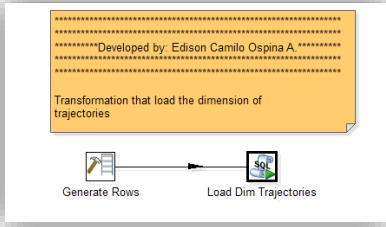
<b>File</b>	\\ETL Solution\transformations\traLoadFactTrajectory.ktr
<b>Name</b>	traLoadFactTrajectory
<b>Description</b>	Transformation that load the reconstructed the low-sampling trajectories to the facttrajectory fact table
	

Table 5.21. Transformation: traLoadFactTrajectory

## 5.7 RECONSTRUCTED TRAJECTORIES BY CRITERIA AND DAY

In the following section, some images about the check-in collected by day and the reconstructed trajectories by criteria along those days are shown. The dataset were collected from August 4, 2014 to August 10, 2014. Both, the check-in data and the reconstructed trajectories were visualized and analyzed using layers in **Qgis Desktop 2.0.1** [108]. For further description of how analyses were made, see the proposal of the *Chapter 4*.

### 5.7.1 Check-in by day

Next, images about the collected check-ins by days in the city of Medellín, Colombia are shown.

#### 5.7.1.1 Check-in data on August 4, 2014 (Medellín)

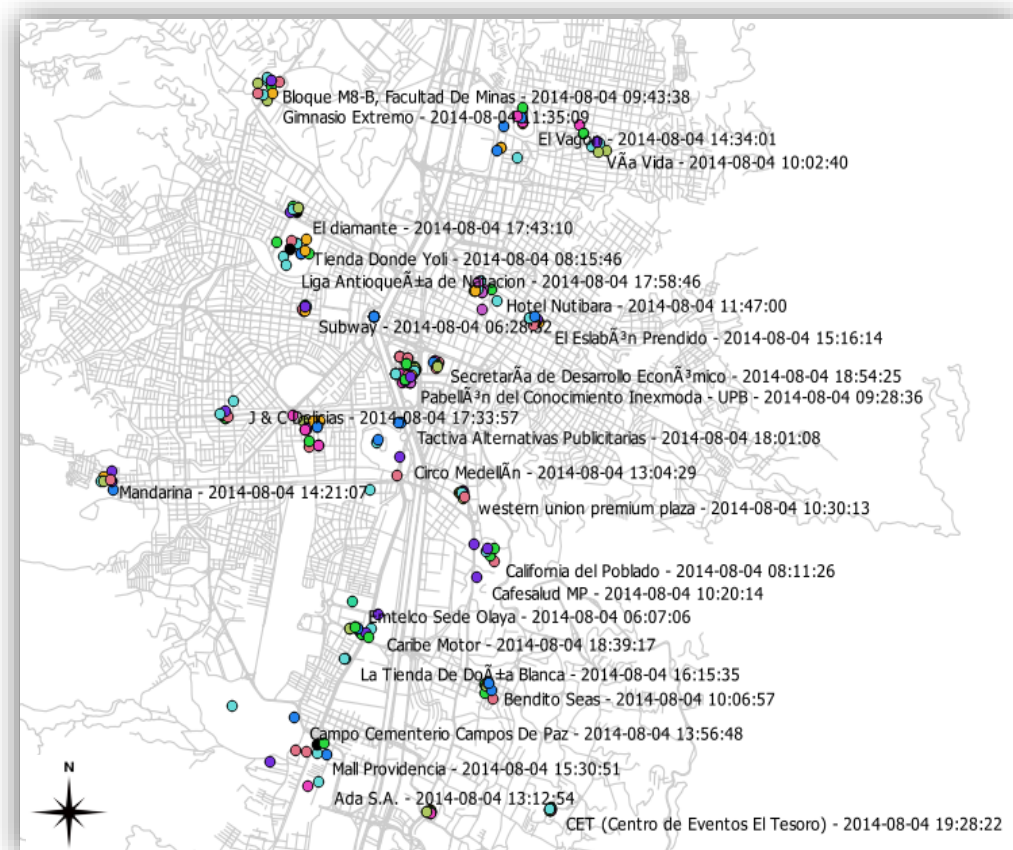


Figure 5.16. Set of check-in points on August 4, 2014 (Medellín)

### 5.7.1.2 Check-in data on August 5, 2014 (Medellín)

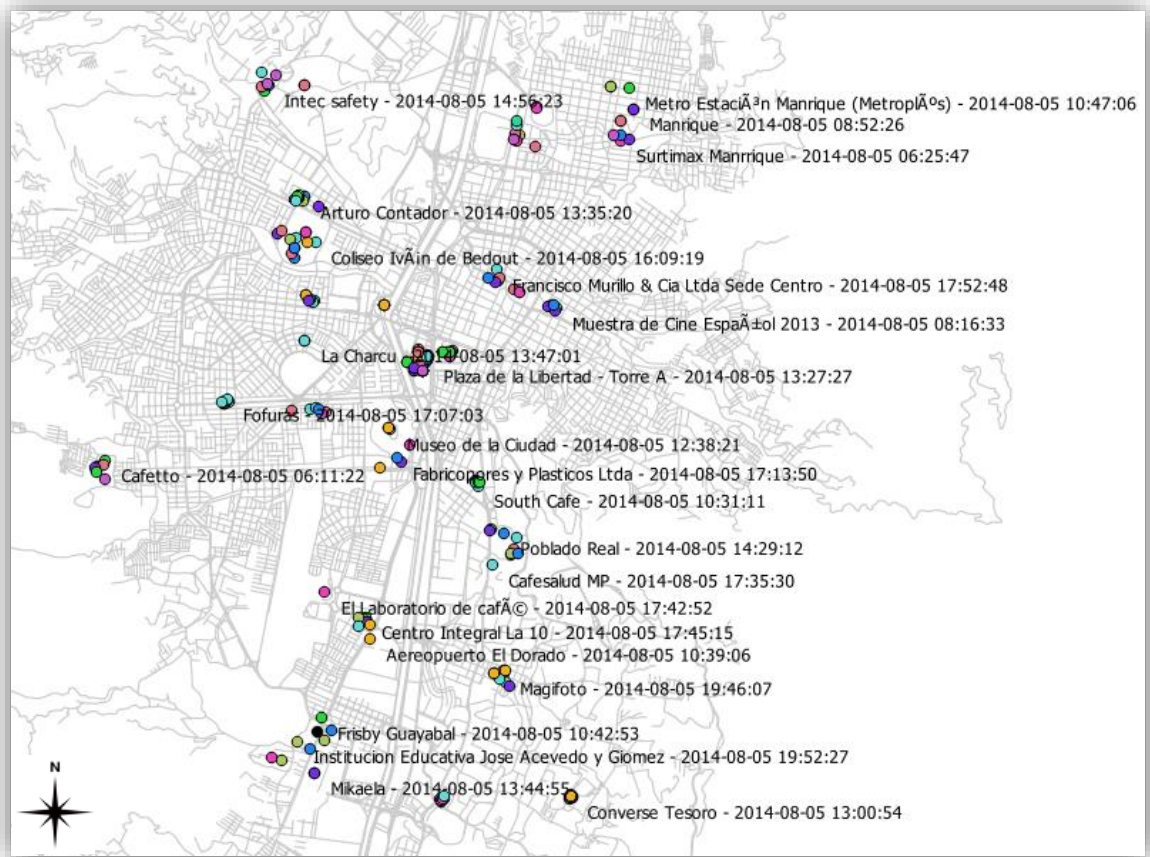


Figure 5.17. Set of check-in points on August 5, 2014 (Medellín)

### 5.7.1.3 Check-in data on August 6, 2014 (Medellín)

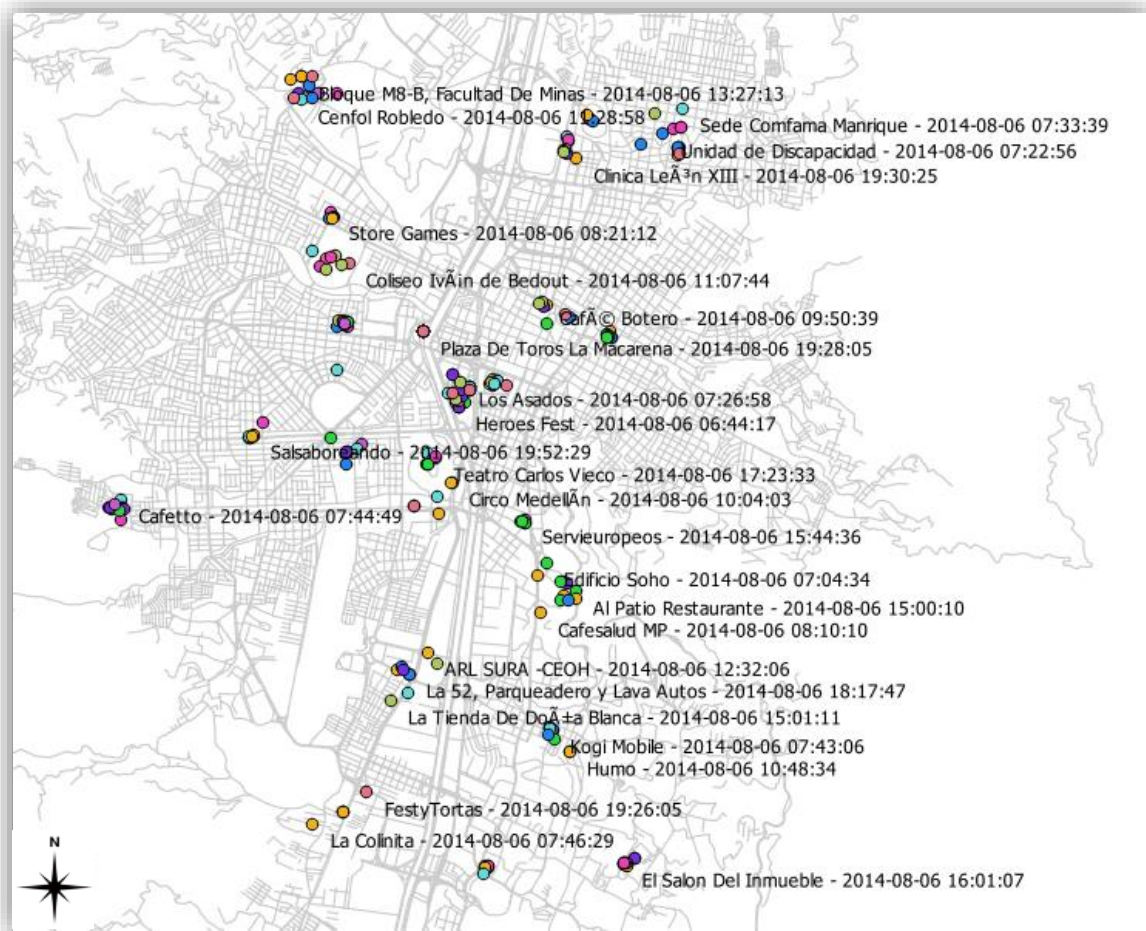


Figure 5.18. Set of check-in points on August 6, 2014 (Medellín)



### 5.7.1.4 Check-in data on August 7, 2014 (Medellín)

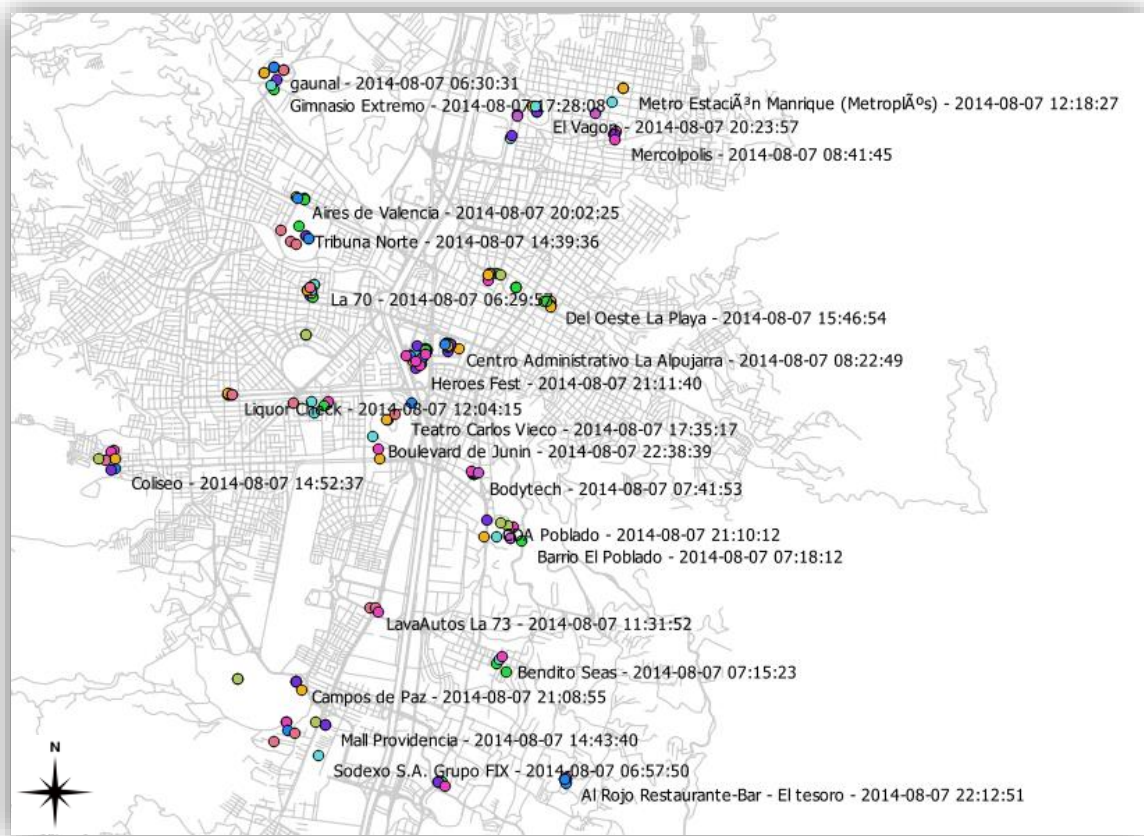


Figure 5.19. Set of check-in points on August 7, 2014 (Medellín)

### 5.7.1.5 Check-in data on August 8, 2014 (Medellín)

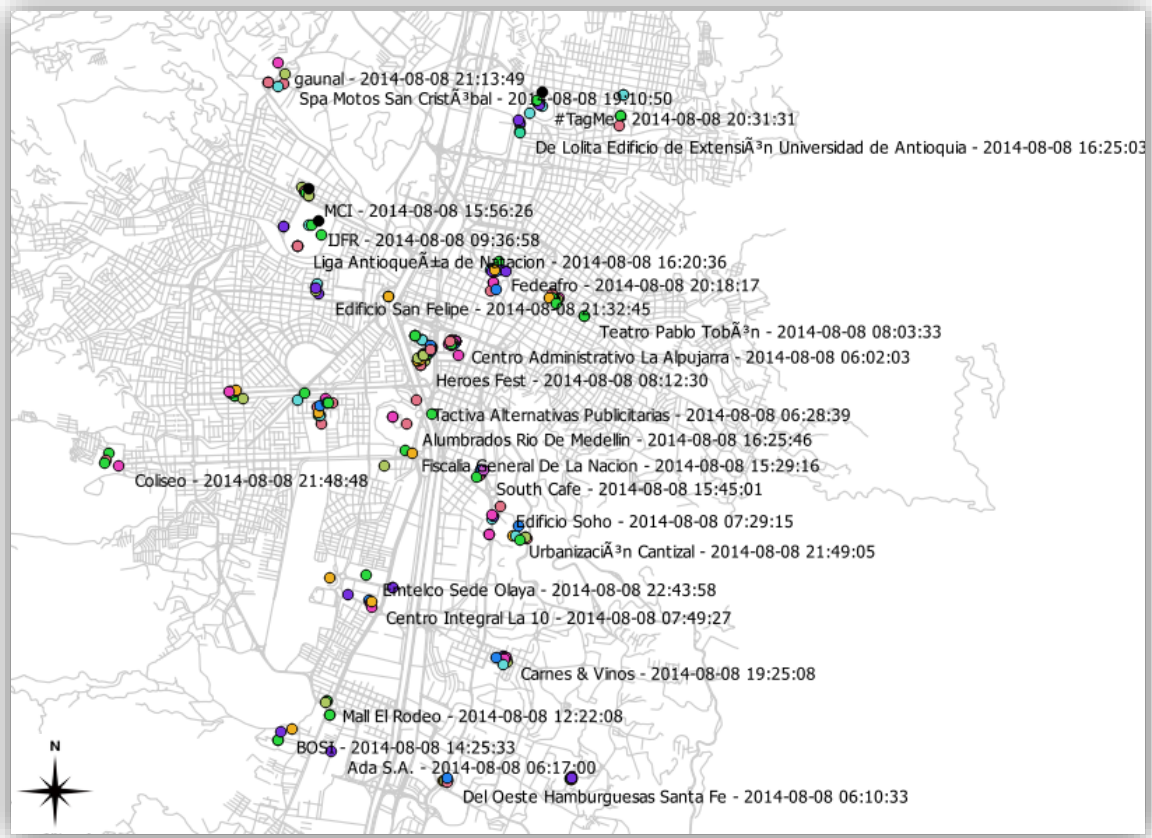


Figure 5.20. Set of check-in points on August 8, 2014 (Medellín)

### 5.7.1.6 Check-in data on August 9, 2014 (Medellín)

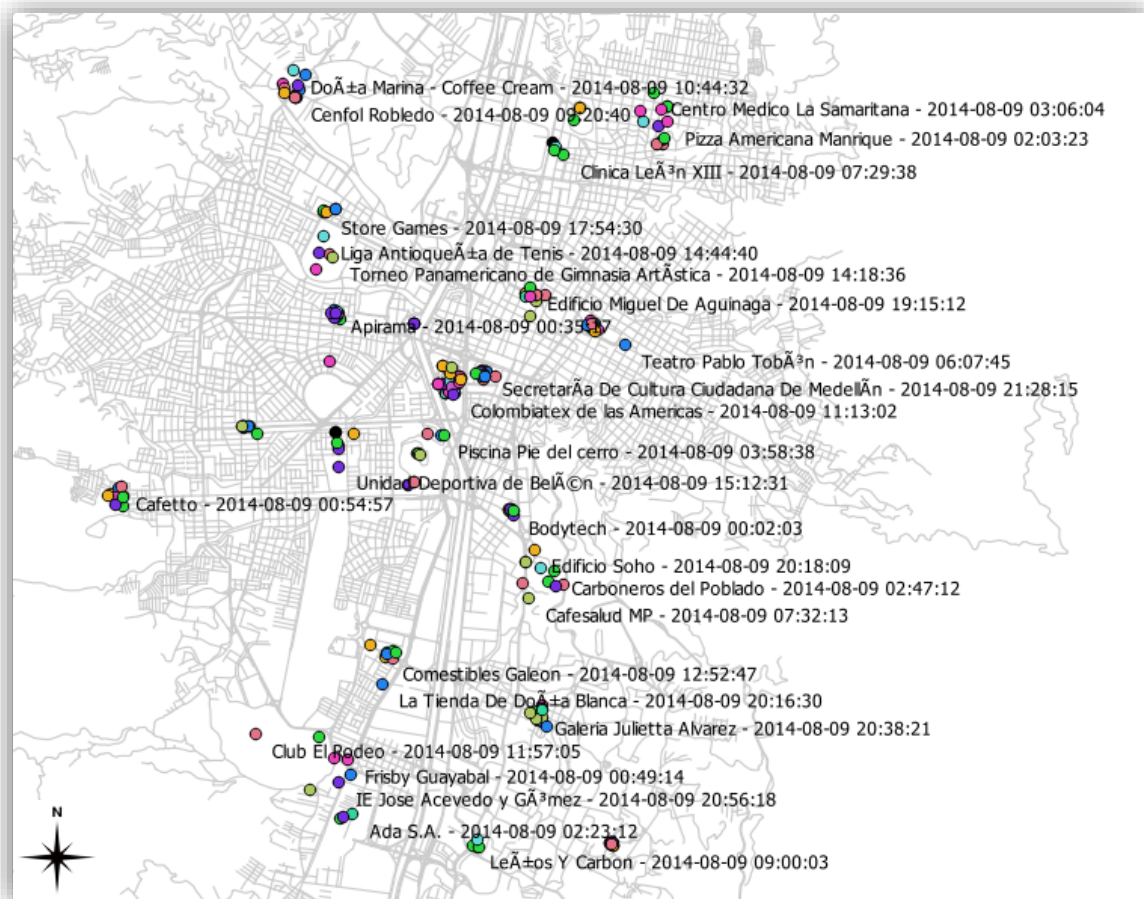


Figure 5.21. Set of check-in points on August 9, 2014 (Medellín)

### 5.7.1.7 Check-in data on August 10, 2014 (Medellín)

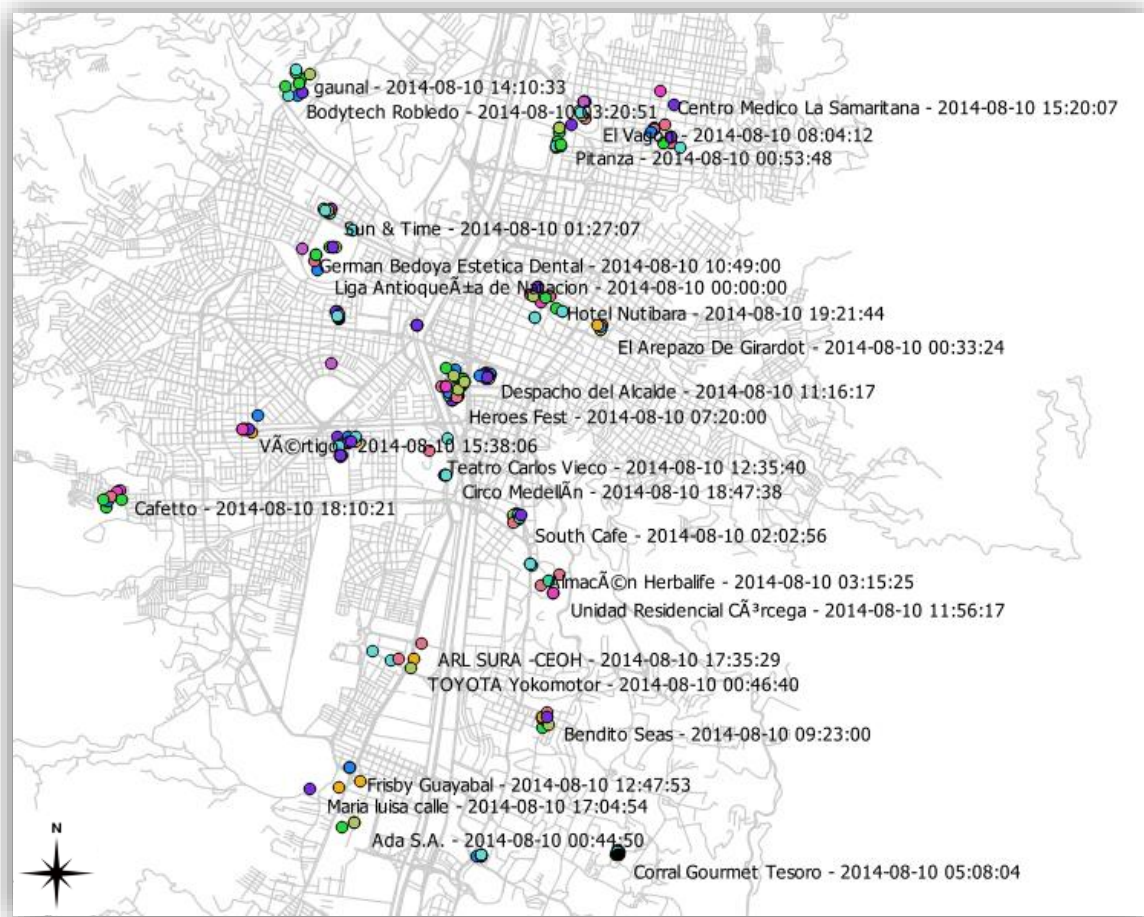


Figure 5.22. Set of check-in points on August 10, 2014 (Medellín)

### 5.7.2 Reconstructed trajectories by criteria and days

Next, images about the reconstructed trajectories of the dataset of check-ins by days and criteria in the city of Medellín, Colombia are shown.



### 5.7.2.1 The reconstructed trajectories on August 4, 2014

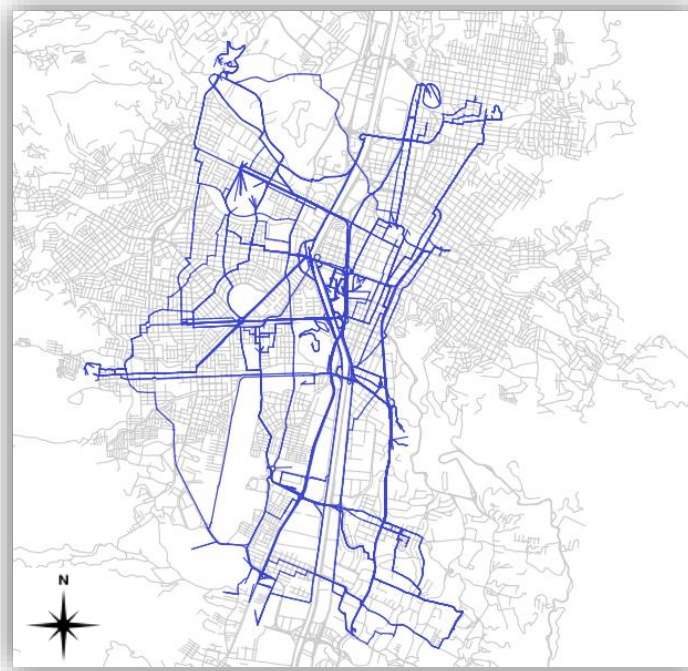


Figure 5.23. Reconstructed trajectories using Distance criterion on August 4, 2014 (Medellín)

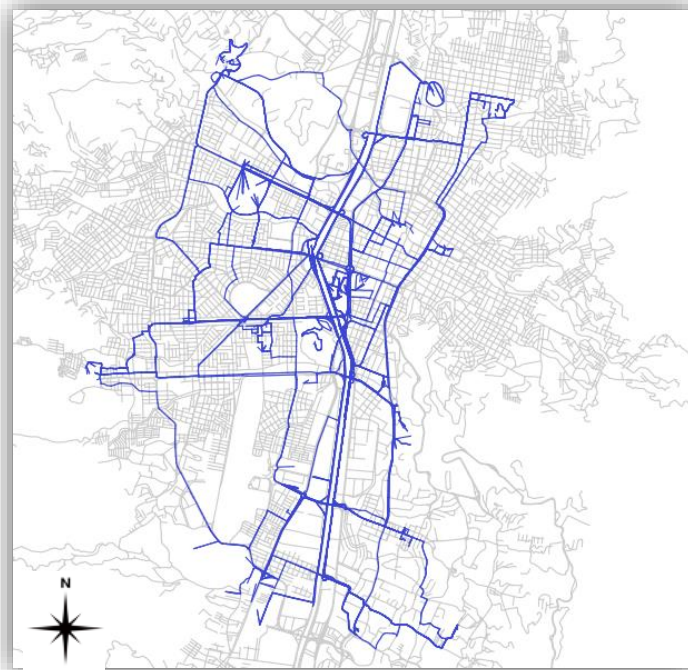


Figure 5.24. Reconstructed trajectories using Time criterion on August 4, 2014 (Medellín)

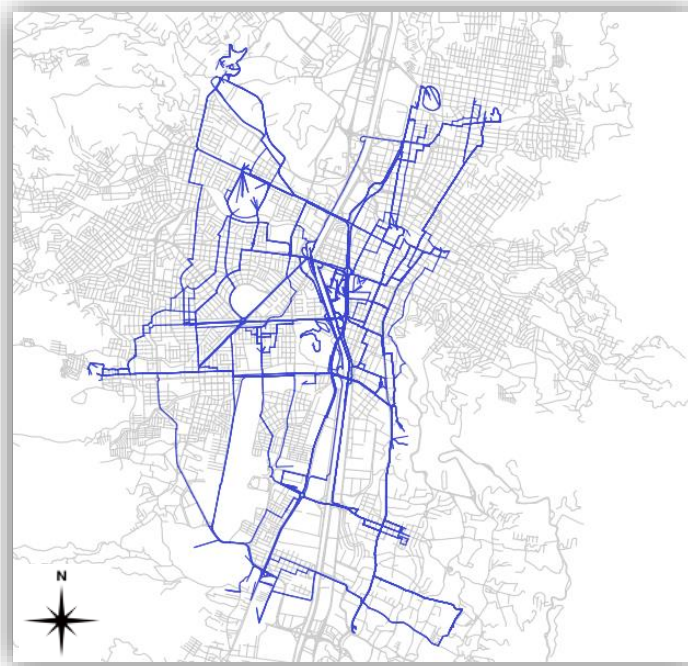


Figure 5.25. Reconstructed trajectories using Touristic criterion on August 4, 2014 (Medellín)

#### 5.7.2.2 The reconstructed trajectories on August 5, 2014

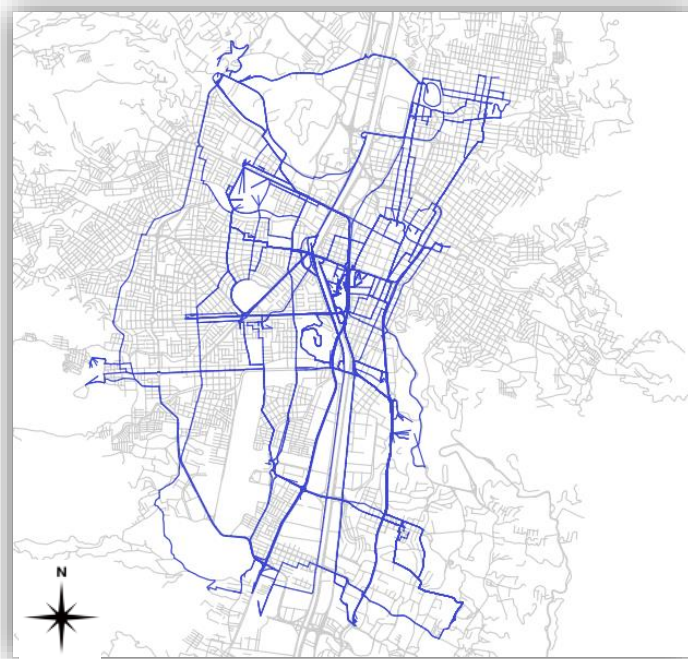


Figure 5.26. Reconstructed trajectories using Distance criterion on August 5, 2014 (Medellín)

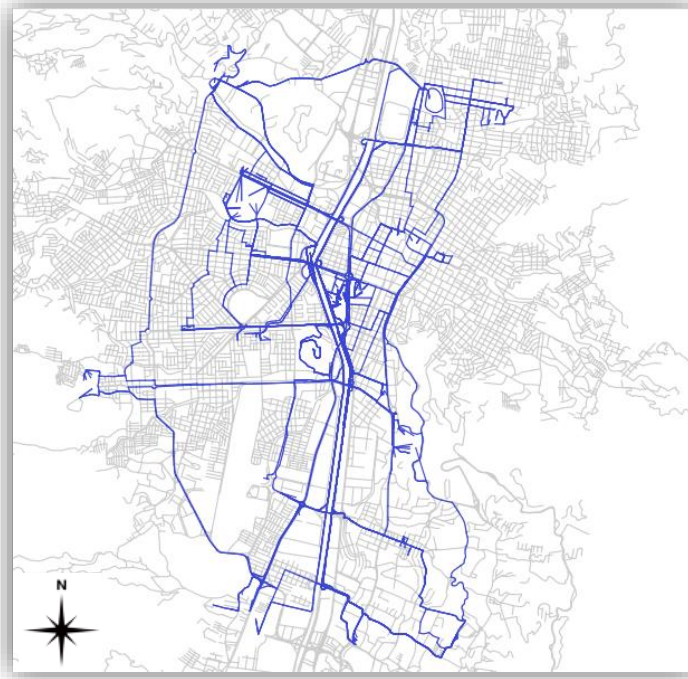


Figure 5.27. Reconstructed trajectories using Time criterion on August 5, 2014 (Medellín)

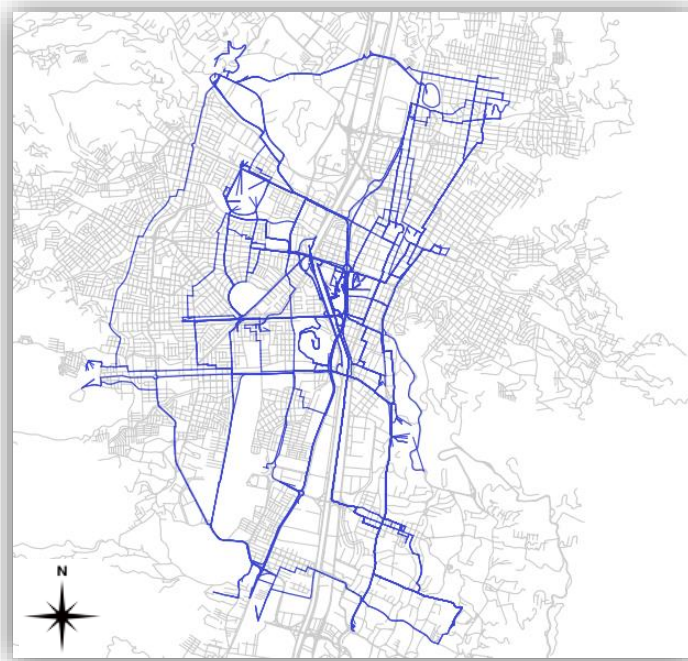


Figure 5.28. Reconstructed trajectories using Touristic criterion on August 5, 2014 (Medellín).



### 5.7.2.3 The reconstructed trajectories on August 6, 2014

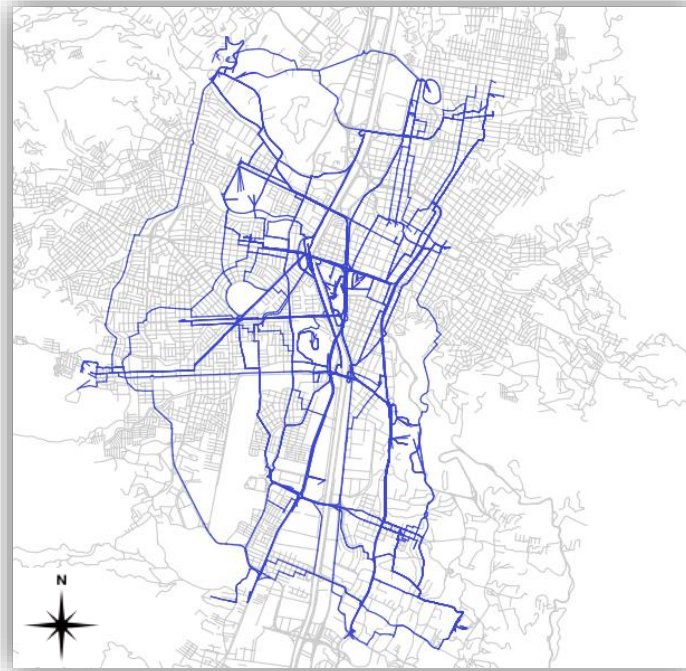
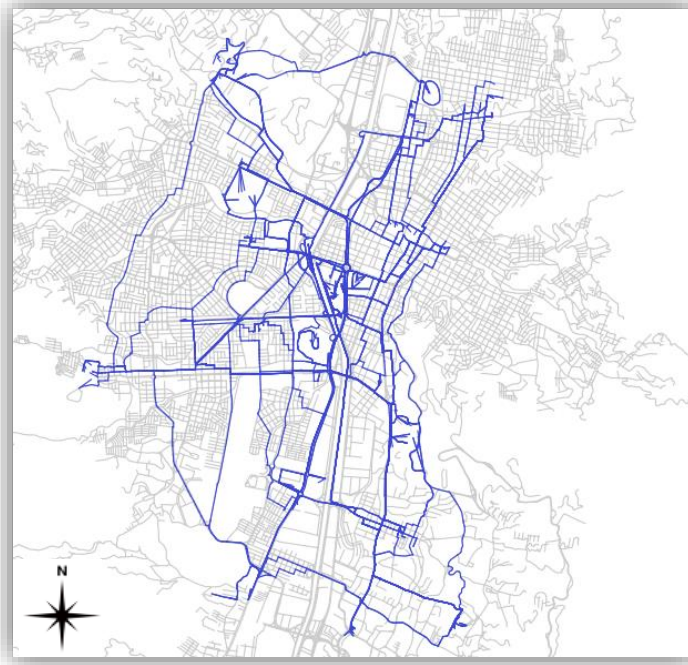


Figure 5.29. Reconstructed trajectories using Distance criterion on August 6, 2014 (Medellín)



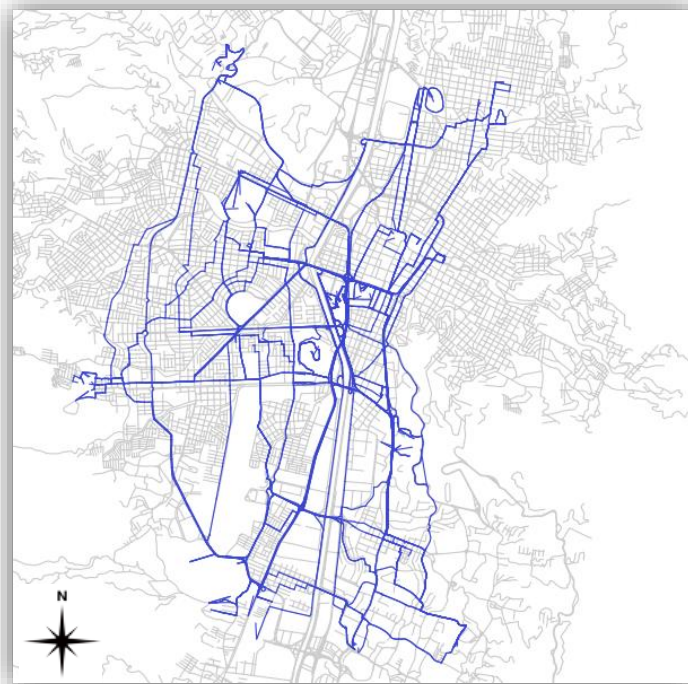
Figure 5.30. Reconstructed trajectories using Time criterion on August 6, 2014 (Medellín)





*Figure 5.31. Reconstructed trajectories using Touristic criterion on August 6, 2014 (Medellín)*

#### **5.7.2.4 The reconstructed trajectories on August 7, 2014.**



*Figure 5.32. Reconstructed trajectories using Distance criterion on August 7, 2014 (Medellín)*

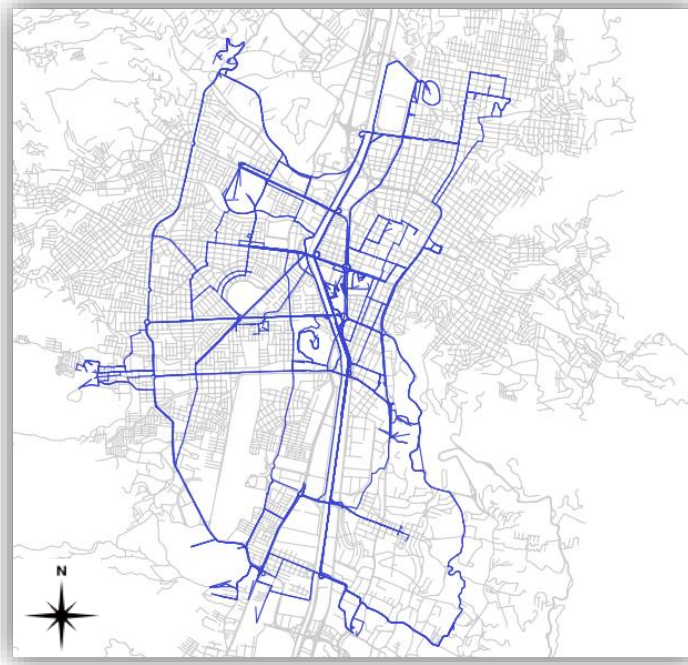


Figure 5.33. Reconstructed trajectories using Time criterion on August 7, 2014 (Medellín)

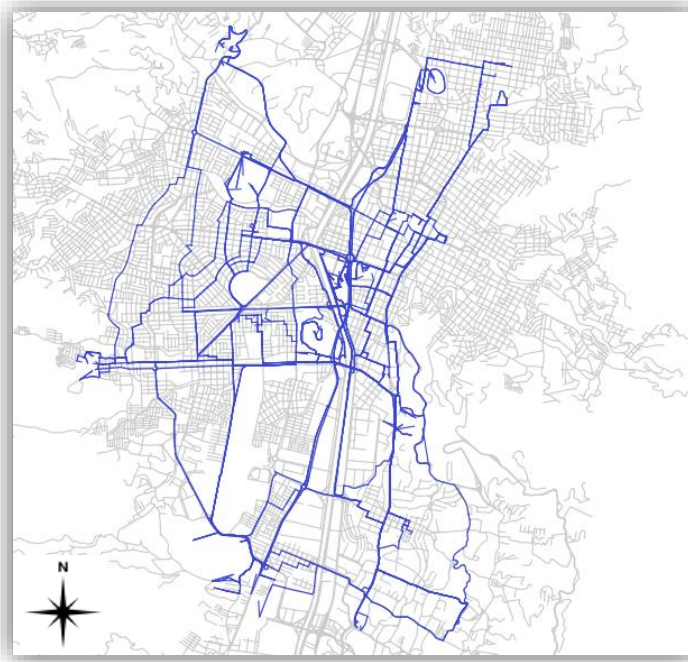
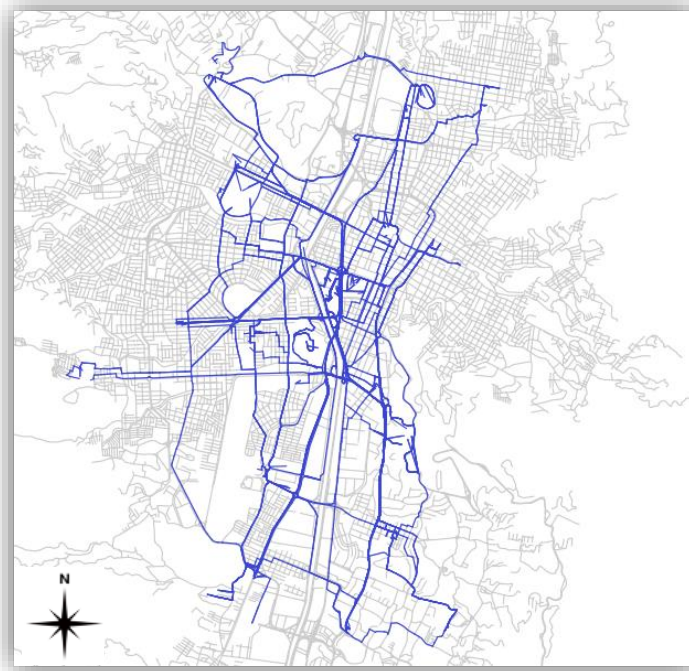
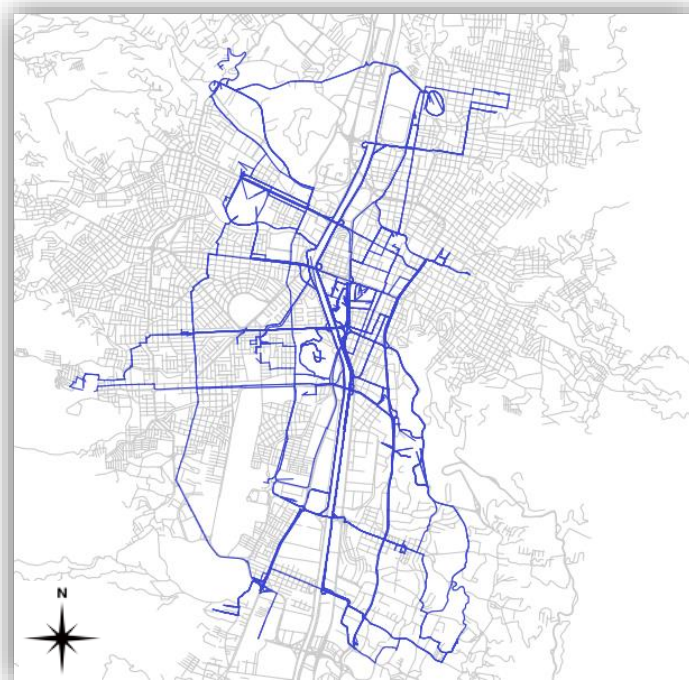


Figure 5.34. Reconstructed trajectories using Touristic criterion on August 7, 2014 (Medellín)

**5.7.2.5 The reconstructed trajectories on August 8, 2014.**



*Figure 5.35. Reconstructed trajectories using Distance criterion on August 8, 2014 (Medellín)*



*Figure 5.36. Reconstructed trajectories using Time criterion on August 8, 2014 (Medellín)*



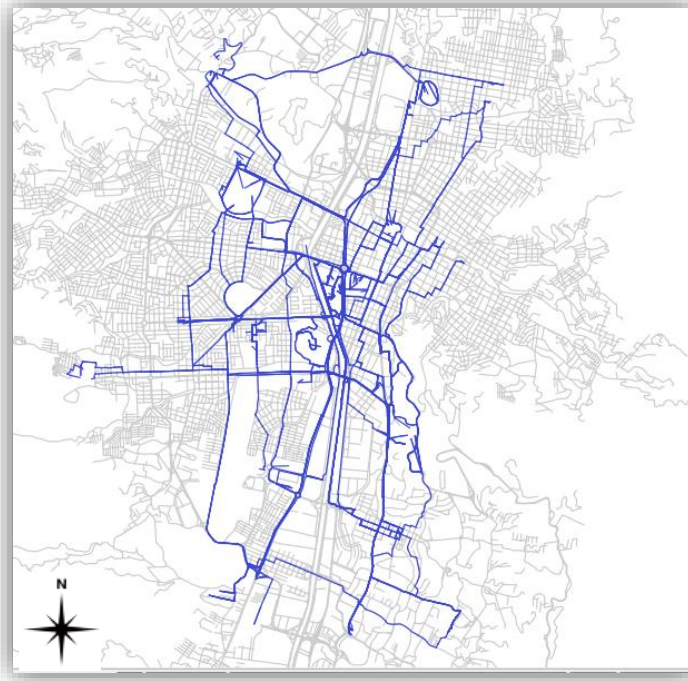


Figure 5.37. Reconstructed trajectories using Touristic criterion on August 8, 2014 (Medellín)

**5.7.2.6 The reconstructed trajectories on August 9, 2014.**

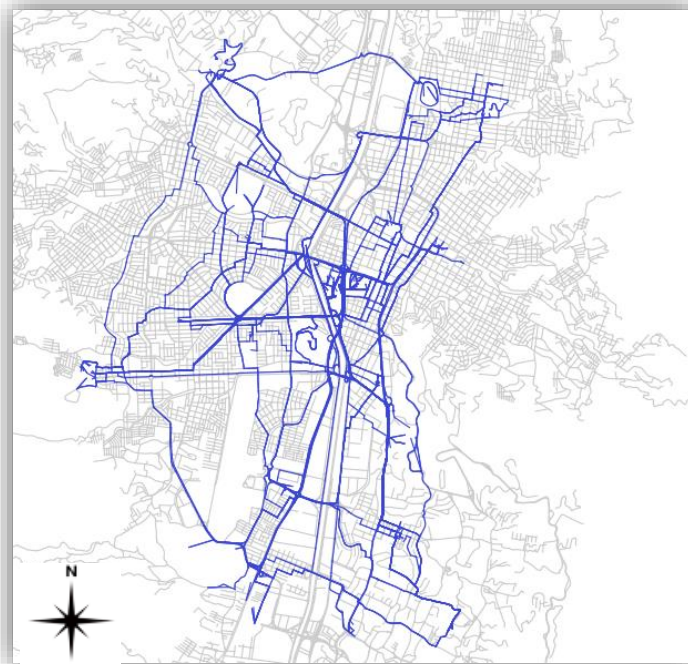


Figure 5.38. Reconstructed trajectories using Distance criterion on August 9, 2014 (Medellín)

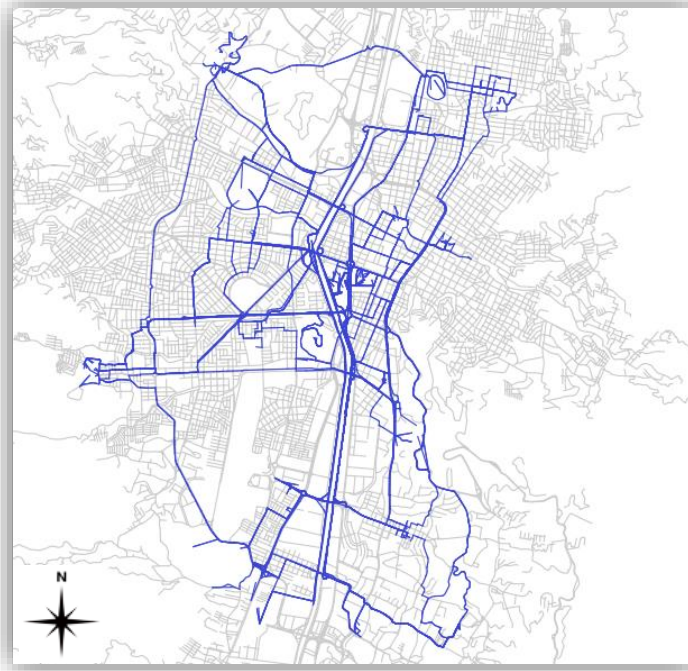


Figure 5.39. Reconstructed trajectories using Time criterion on August 9, 2014 (Medellín)

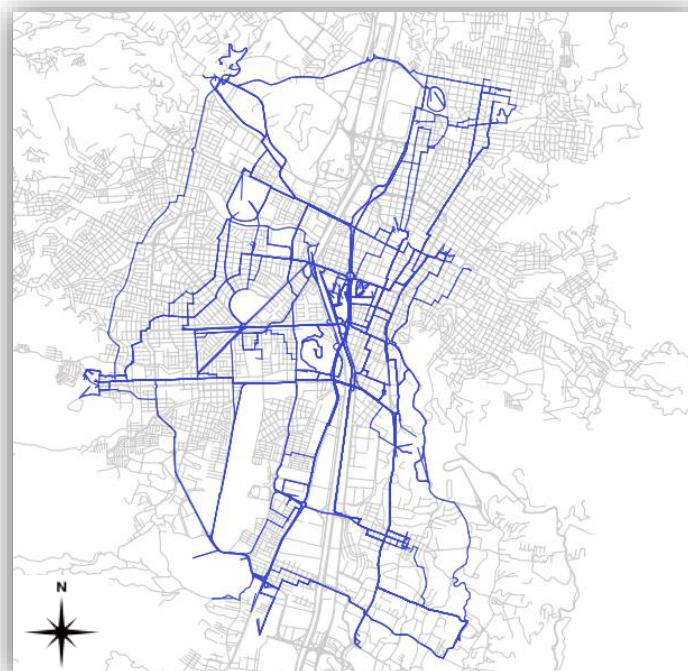


Figure 5.40. Reconstructed trajectories using Touristic criterion on August 9, 2014 (Medellín)

### 5.7.2.7 The reconstructed trajectories on August 10, 2014.

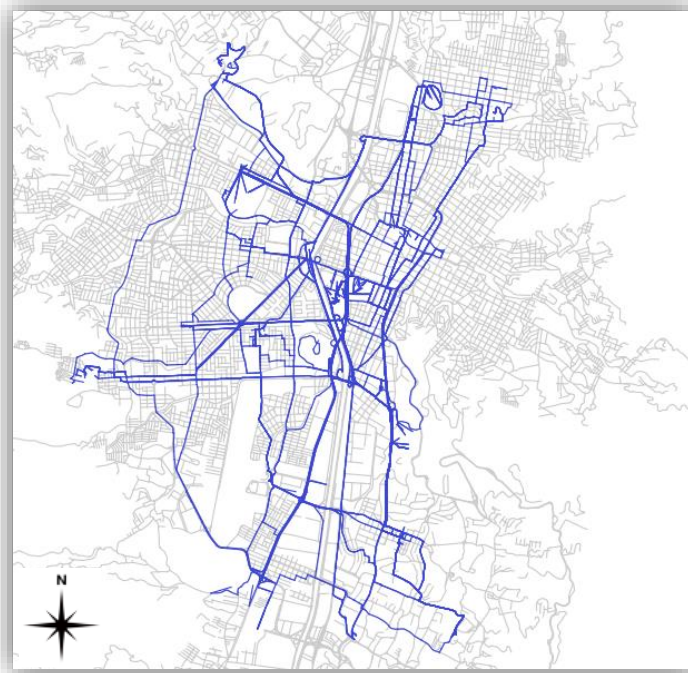


Figure 5.41. Reconstructed trajectories using Distance criterion on August 10, 2014 (Medellín)



Figure 5.42. Reconstructed trajectories using Time criterion on August 10, 2014 (Medellín)



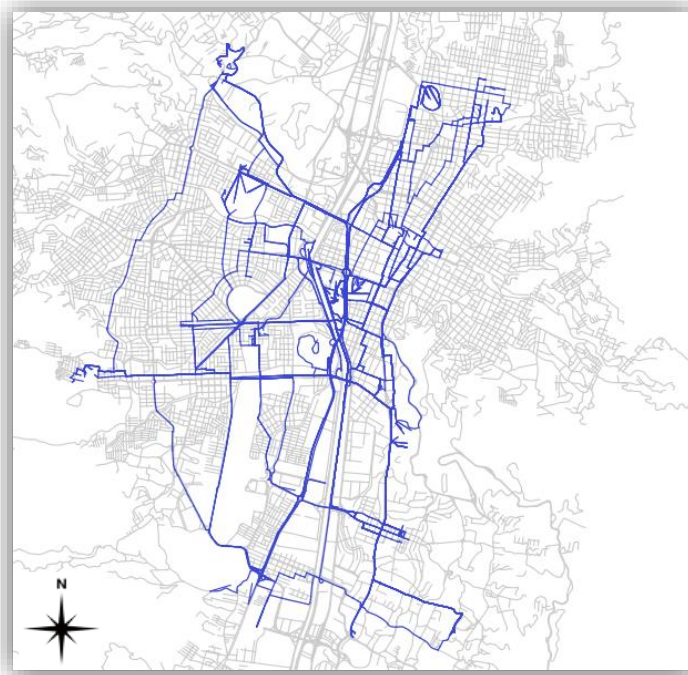


Figure 5.43. Reconstructed trajectories using Touristic criterion on August 10, 2014 (Medellín)

## 5.8 BACKUP FOR TESTING

### 5.8.1 Database Back up

In order to reproduce the tests carried out in this thesis, the next backup must be restored in **Postgress 9.2** [105]: `\DBObjects\DBBackup\BackupTar.backup`. Remember create a database with PostGIS [103] capabilities (It is a requirement for the restoring).

### 5.8.2 QGIS visualizations

In order to reproduce the visualizations analysis carried out in this thesis, load the file `\QGis\MedellinLastChapterProofs` in **QGIS Desktop 2.0.1** [108] after the restoring of the backup as explained in the *Section 5.8.1*.

#### **The main contributions of this chapter are:**

- This chapter develops the specific objective “*Validate the effectiveness of the proposal using a functional prototype for testing*”.

## GENERAL CONCLUSIONS

Next, the overall conclusions are summed up.

The trajectory reconstruction problem is still an open research issue, especially what is related to uncertainty due to low-sampling data and the incorporation of user preferences. Simple linear interpolation [30], as a method of reconstruction of low-sampling location data, does not represent user real movement because they move according to a certain criteria such as time or the amount of touristic/scenic places. To the best of our knowledge, there are no research work that involve several criteria as a way to reconstructing low-sampling trajectories. In this thesis, low-sampling trajectories were reconstructed using the personalization features of the routing theory based on a criterion decision over a graph. Although, the real trajectories are not guessed, a useful imputation process can be developed for specific analysis.

Considering the different possibilities of user criteria reconstruction of trajectory and the huge amount of low-sampling data, data analysis tasks related to these possibilities of reconstruction were conducted using e.g., TDW approaches. Therefore, analytic results over reconstructed trajectories vary if different criteria of reconstruction are used. Using the *traj* function with different criteria can be used as an input for different mining algorithms over trajectories as a way to deal with analytics using uncertain trajectories. Here, it is claimed that analytics over reconstructed trajectories can change depending on the criterion used for the trajectory reconstruction.



## REFERENCES

- [1] P. Ross. "Top 11 technologies of the decade," IEEE Spectrum 48, pp. 27-63, 2011.
- [2] J. Raper and G. Gartner, "Applications of location-based services: a selected review," Journal of Location Based Services 1, no. 2, pp. 89-111, 2007.
- [3] S. Spaccapietra, C. Parent, M. L. Damiani, J. A. De Macedo, F. Porto, and C. Vangenot, "A conceptual view on trajectories," Data & knowledge engineering 65, no. 1, pp. 126-146, 2008.
- [4] Z. Yan and J. Macedo, "Trajectory ontologies and queries," Transactions in GIS 12, no. s1, pp. 75-91, 2008.
- [5] N. Andrienko and G. Andrienko, "Basic concepts of movement data," In Mobility, Data Mining and Privacy, pp. 15-38. Springer Berlin Heidelberg, 2008.
- [6] K. Zheng, Y. Zheng, X. Xie, and X. Zhou, "Reducing uncertainty of low-sampling-rate trajectories," In Data Engineering (ICDE), 2012 IEEE 28th International Conference on, pp. 1144-1155. IEEE, 2012.
- [7] L. Wei, Y. Zheng, and W. Peng, "Constructing popular routes from uncertain trajectories," In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 195-203. ACM, 2012.
- [8] O. Abul, F. Bonchi, and M. Nanni, "Never walk alone: Uncertainty for anonymity in moving objects databases," In Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on, pp. 376-385. IEEE, 2008.
- [9] F. Giannotti and D. Pedreschi, "Mobility, data mining and privacy: A vision of convergence," In Mobility, data mining and privacy, pp. 1-11. Springer Berlin Heidelberg, 2008.
- [10] C. Hung, L. Wei, and W. Peng, "Clustering Clues of Trajectories for Discovering Frequent Movement Behaviors," The VLDB Journal, pp. 1-24, 2011.
- [11] E. Silva and C. de Baptista, "Personalized path finding in road networks," In Networked Computing and Advanced Information Management, 2008. NCM'08. Fourth International Conference on, vol. 2, pp. 586-591. IEEE, 2008.
- [12] D. Schultes, "Route planning in road networks," In Ausgezeichnete Informatikdissertationen, pp. 271-280, 2008.
- [13] H. Hochmair, "Towards a classification of route selection criteria for route planning tools," In Developments in Spatial Data Handling, pp. 481-492. Springer Berlin Heidelberg, 2005.
- [14] E. Dijkstra, "A note on two problems in connexion with graphs," Numerische mathematik 1, no. 1, pp. 269-271, 1959.

- [15] A. S. Niaraki and K. Kim, "Ontology based personalized route planning system using a multi-criteria decision making approach," *Expert Systems with Applications* 36, no. 2, pp. 2250-2259, 2009.
- [16] S. Nadi and M. Delavar, "Multi-criteria, personalized route planning using quantifier-guided ordered weighted averaging operators," *International Journal of Applied Earth Observation and Geoinformation*, No. 3, pp. 322-335, 2011.
- [17] G. Trajcevski, "Uncertainty in spatial trajectories," In *Computing with Spatial Trajectories*, pp. 63-107. Springer New York, 2011.
- [18] V. De Almeida and R. Güting, "Supporting uncertainty in moving objects in network databases," In *Proceedings of the 13th annual ACM international workshop on Geographic information systems*, pp. 31-40. ACM, 2005.
- [19] H. Gowrisankar and S. Nittel, "Reducing uncertainty in location prediction of moving objects in road networks," In *Proceedings of the 2nd International Conference on Geographic Information Science*, 2002.
- [20] J. Lee, W. C., & Krumm, "Trajectory Preprocessing," In *Computing with spatial trajectories*, pp. 3-33. Springer New York, 2011.
- [21] S. Rinzivillo, F. Turini, and V. Bogorny, "Knowledge discovery from geographical data," In *Mobility, Data Mining and Privacy*, pp. 243-265. Springer Berlin Heidelberg, 2008.
- [22] S. Luján-Mora and J. Trujillo, "A data warehouse engineering process," In *Advances in Information Systems*, pp. 14-23. Springer Berlin Heidelberg, 2005.
- [23] W. Choi, D. Kwon, and S. Lee, "Spatio-temporal data warehouses using an adaptive cell-based approach," *Data & Knowledge Engineering* 59, no. 1, pp. 189-207, 2006
- [24] L. Gómez and B. Kuijpers, "A survey of spatio-temporal data warehousing," *International Journal of Data Warehousing and Mining (IJDWM)* 5, no. 3, pp. 28-55, 2009.
- [25] L. Savary, T. Wan, and K. Zeitouni, "Spatio-temporal data warehouse design for human activity pattern analysis," *International Journal of Data Warehousing and Mining (IJDWM)* 5, no. 3, pp. 28-55, 2009.
- [26] S. Orlando, R. Orsini, and A. Raffaetà, "Trajectory data warehouses: design and implementation issues," In *Proceedings of International Conference on Data Warehousing and Knowledge Discovery*, pp. 66-77, 2007.
- [27] N. Pelekis and A. Raffaeta, "Towards trajectory data warehouses," In *Mobility, Data Mining and Privacy*, pp. 189-211. Springer Berlin Heidelberg, 2008.

- [28] M. Baratchi, "Finding frequently visited paths: dealing with the uncertainty of spatio-temporal mobility data," In *Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on*, pp. 479-484. IEEE, 2013.
- [29] L. Speičvcys, C. Jensen, and A. Kligys, "Computational data modeling for network-constrained moving objects," In *Proceedings of the 11th ACM international symposium on Advances in geographic information systems*, pp. 118-125. ACM, 2003.
- [30] D. Pfoser, C. Jensen, and Y. Theodoridis, "Novel approaches to the indexing of moving object trajectories," In *Proceedings of VLDB*, pp. 395-406. 2000.
- [31] C. Renso, S. Puntoni, and E. Frenzos, "Wireless network data sources: tracking and synthesizing trajectories," In *Mobility, Data Mining and Privacy*, pp. 73-100. Springer Berlin Heidelberg, 2008.
- [32] Routino: Router for OpenStreetMap Data. [Online] Available from: <http://www.routino.org/> 2014.10.20
- [33] Mapquest. [Online] Available from: <http://www.mapquest.com/> 2014.10.20
- [34] K. Chang, L. Wei, M. Yeh, and W. Peng, "Discovering personalized routes from trajectories," In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, pp. 33-40. ACM, 2011.
- [35] Z. Chen, H. Shen, and X. Zhou, "Discovering popular routes from trajectories," In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, pp. 900-911. IEEE, 2011.
- [36] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 316-324. ACM, 2011.
- [37] J. Yuan, Y. Zheng, C. Zhang, and W. Xie, "T-drive: driving directions based on taxi trajectories," In *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*, pp. 99-108. ACM, 2010.
- [38] H. Jeung, M. Yiu, and C. Jensen, "Trajectory pattern mining," In *Computing with Spatial Trajectories*, pp. 143-177. Springer New York, 2011.
- [39] K. Park and M. Bell, "Learning user preferences of route choice behaviour for adaptive route guidance," *IET Intelligent Transport Systems* 1, no. 2, pp. 159-166, 2007.
- [40] L. McGinty and B. Smyth, "Shared experiences in personalized route planning," In *FLAIRS Conference*, pp. 111-115, 2002.
- [41] W. Herbert and F. Mili, "Route guidance: state of the art vs. state of the practice," In *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 1167-1174. IEEE, 2008.

- [42] M. Khanjary and S. Hashemi, "Route guidance systems: review and classification," In Proceedings of the 6th Euro American Conference on Telematics and Information Systems, pp. 269-275. ACM, 2012.
- [43] E. Schmitt and H. Jula, "Vehicle route guidance systems: Classification and comparison," In Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE, pp. 242-247. IEEE, 2006.
- [44] J. Adler and V. Blue, "Toward the design of intelligent traveler information systems," Transportation Research Part C: Emerging Technologies 6, no. 3, pp. 157-172, 1998.
- [45] D. Delling, A. Goldberg, T. Pajor, and R. Werneck, "Customizable route planning," In Experimental Algorithms, pp. 376-387. Springer Berlin Heidelberg, 2011.
- [46] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," Systems Science and Cybernetics, IEEE Transactions on 4, no. 2, pp. 100-107, 1968.
- [47] J. Van de Geer, "SOME ASPECTS OF MINKOWSKI DISTANCES," 1995.
- [48] R. Golledge, "Path selection and route preference in human navigation: A progress report," Springer Berlin Heidelberg, pp. 207-222, 1995.
- [49] M. Duckham and L. Kulik, "'Simplest' Paths: Automated Route Selection for Navigation," In Spatial Information Theory. Foundations of Geographic Information Science, pp. 169-185. Springer Berlin Heidelberg, 2003.
- [50] G. Fischer, "Shared knowledge in cooperative problem-solving systems - Integrating adaptive and adaptable components," In Schneider-Hufschmidt, M., Kuehme, T., & Malinowski, U., (Eds) Adaptive User Interfaces, Principles and Practice. Elsevier Science Publishers, Amsterdam, pp. 49 - 68, 1993.
- [51] R. Oppermann, Adaptive user support: ergonomic design of manually and automatically adaptable software. CRC Press, 1994.
- [52] G. Pang and K. Takabashi, "Adaptive route selection for dynamic route guidance system based on fuzzy-neural approaches," Vehicular Technology, IEEE Transactions on 48, no. 6, pp. 2028-2041, 1999.
- [53] S. Rogers, C. Fiechter, and P. Langley, "An adaptive interactive agent for route advice," In Proceedings of the third annual conference on Autonomous Agents, pp. 198-205. ACM, 1999.
- [54] A. Zipf and M. Jöst, "Implementing adaptive mobile GI services based on ontologies: Examples from pedestrian navigation support," Computers, Environment and Urban Systems 30, no. 6, pp. 784-798, 2006.

- [55] I. Lin and S. Chou, "Developing adaptive driving route guidance systems based on fuzzy neural network," IET International Conference on Intelligent Environments, Seattle, Wa, Usa, pp 1-8, 2008.
- [56] J. Letchner, J. Krumm, and E. Horvitz, "Trip router with individualized preferences (trip): Incorporating personalization into route planning," In Proceedings of the National Conference on Artificial Intelligence, vol. 21, no. 2, p. 1795. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.
- [57] T. Völkel and G. Weber, "RouteCheckr: personalized multicriteria routing for mobility impaired pedestrians," In Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility, pp. 185-192. ACM, 2008.
- [58] M. Kenteris and D. Gavalas, "Near-optimal personalized daily itineraries for a mobile tourist guide," In Computers and Communications (ISCC), 2010 IEEE Symposium on, pp. 862-864. IEEE, 2010.
- [59] D. Gavalas, "Personalized routes for mobile tourism," In Wireless and Mobile Computing, Networking and Communications (WiMob), 2011 IEEE 7th International Conference on, pp. 295-300. IEEE, 2011.
- [60] H. Hochmair and G. Navratil, "Computation of scenic routes in street networks," In Proceedings of the Geoinformatics Forum Salzburg, 2008.
- [61] H. Hsieh and C. Li, "Constructing trip routes with user preference from location check-in data," In Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication, pp. 195-198. ACM, 2013.
- [62] L. Wei and W. Peng, "Pats: A framework of pattern-aware trajectory search," In Mobile Data Management (MDM), 2010 Eleventh International Conference on, pp. 372-377. IEEE, 2010.
- [63] Z. Chen, H. Shen, X. Zhou, Y. Zheng, and X. Xie, "Searching trajectories by locations: an efficiency study," In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, pp. 255-266. ACM, 2010.
- [64] C. Chow and M. Mokbel, "Privacy of spatial trajectories," In Computing with spatial trajectories, pp. 109-141. Springer New York, 2011.
- [65] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," In Proceedings of the 1st international conference on Mobile systems, applications and services, pp. 31-42. ACM, 2003.

- [66] B. Gedik and L. Liu, "Location privacy in mobile systems: A personalized anonymization model," In *Distributed Computing Systems*, 2005. ICDCS 2005. Proceedings. 25th IEEE International Conference on, pp. 620-629. IEEE, 2005.
- [67] J. Hong and J. Landay, "An architecture for privacy-sensitive ubiquitous computing," In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pp. 177-189. ACM, 2004.
- [68] A. Khoshgozaran and C. Shahabi, "Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy," In *Advances in Spatial and Temporal Databases*, pp. 239-257. Springer Berlin Heidelberg, 2007.
- [69] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," In *Pervasive computing*, pp. 152-170. Springer Berlin Heidelberg, 2005.
- [70] D. Pfoser and C. Jensen, "Capturing the uncertainty of moving-object representations," In *Advances in Spatial Databases*, pp. 111-131. Springer Berlin Heidelberg, 1999.
- [71] G. Trajcevski and O. Wolfson, "Managing uncertainty in moving objects databases," *ACM Transactions on Database Systems (TODS)* 29, no. 3, pp. 463-507, 2004.
- [72] H. Liu, L. Wei, and Y. Zheng, "Route discovery from mining uncertain trajectories," In *Data Mining Workshops (ICDMW)*, 2011 IEEE 11th International Conference on, pp. 1239-1242. IEEE, 2011.
- [73] B. Kuijpers and B. Moelans, "Analyzing trajectories using uncertainty and background information," In *Advances in Spatial and Temporal Databases*, pp. 135-152. Springer Berlin Heidelberg, 2009.
- [74] T. Kurashima, T. Iwata, G. Irie, and K. Fujimura, "Travel route recommendation using geotags in photo sharing sites," In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 579-588. ACM, 2010.
- [75] X. Long, L. Jin, and J. Joshi, "Exploring trajectory-driven local geographic topics in foursquare," In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pp. 927-934. ACM, 2012.
- [76] J. Greenfeld, "Matching GPS observations to locations on a digital map," In *Transportation Research Board 81st Annual Meeting*. 2002.
- [77] Y. Lou, C. Zhang, Y. Zheng, and X. Xie, "Map-matching for low-sampling-rate GPS trajectories," In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 352-361. ACM, 2009.

- [78] T. Cheng and J. Haworth, "Spatiotemporal Data Mining," In Handbook of Regional Science, pp. 1173-1193. Springer Berlin Heidelberg, 2014.
- [79] M. Ester, H. Kriegel, and J. Sander, "Algorithms and applications for spatial data mining," Geographic Data Mining and Knowledge Discovery 5, no. 6, 2001.
- [80] W. H., Inmon, Building the data warehouse. John wiley & sons, 2005.
- [81] R. Kimball and M. Ross, The data warehouse toolkit: the complete guide to dimensional modeling. 2011.
- [82] J. Han, N. Stefanovic, and K. Koperski, "Selective materialization: An efficient method for spatial data cube construction," In Research and Development in Knowledge Discovery and Data Mining, pp. 144-158. Springer Berlin Heidelberg, 1998.
- [83] E. Malinowski and E. Zimányi, Advanced data warehouse design: From conventional to spatial and temporal applications. Springer, 2008.
- [84] A. Vaisman and E. Zimányi, "What is spatio-temporal data warehousing?," What is spatio-temporal data warehousing?. Springer Berlin Heidelberg, 2009.
- [85] W. Frawley, G. Piatetsky-Shapiro, and C. Matheus, "Knowledge discovery in databases: An overview," AI magazine 13, no. 3, pp. 57, 1992.
- [86] G. Marketos and Y. Theodoridis, "Mobility Data Warehousing and Mining,," In VLDB PhD Workshop. 2009.
- [87] Y. Bédard, T. Merrett, and J. Han, "Fundamentals of spatial data warehousing for geographic knowledge discovery," Geographic data mining and knowledge discovery 2, pp. 53-73, 2001.
- [88] N. Stefanovic, J. Han, and K. Koperski, "Object-based selective materialization for efficient implementation of spatial data cubes," Knowledge and Data Engineering, IEEE Transactions on 12, no. 6, pp. 938-958, 2000.
- [89] E. Malinowski and E. Zimányi, "Representing spatiality in a conceptual multidimensional model," In Proceedings of the 12th annual ACM international workshop on Geographic information systems, pp. 12-22. ACM, 2004.
- [90] G. Marketos, E. Frentzos, and I. Ntoutsis, "A framework for trajectory data warehousing," In Proceedings of ACM SIGMOD Workshop on Data Engineering for Wireless and Mobile Access. 2008.
- [91] M. Lv, L. Chen, and G. Chen, "Mining user similarity based on routine activities," Information Sciences, pp. 17-32, 2013.
- [92] Z. Ding and R. Güting, "Uncertainty management for network constrained moving objects," In Database and Expert Systems Applications, pp. 411-421. Springer Berlin Heidelberg, 2004.

- [93] C. Schlenoff, R. Madhavan, and S. Balakirsky, "An approach to predicting the location of moving objects during on-road navigation," In Proceedings of IJCAI-03-Workshop on Issues in Designing Physical Agents for Dynamic Real-Time Environments: World modeling, planning, learning, ancommunicating, pp. 71–79, 2003.
- [94] M. Wachowicz and A. Ligtenberg, "Characterising the next generation of mobile applications through a privacy-aware geographic knowledge discovery process," In Mobility, Data Mining and Privacy, pp. 39-72. Springer Berlin Heidelberg, 2008.
- [95] Z. Yan, "Semantic trajectories: computing and understanding mobility data," PhD diss., 2011.
- [96] H. Zhao, Q. Han, H. Pan, and G. Yin, "Spatio-temporal similarity measure for trajectories on road networks," In Internet Computing for Science and Engineering (ICICSE), 2009 Fourth International Conference on, pp. 189-193. IEEE, 2009.
- [97] E. Tiakas and A. Papadopoulos, "Searching for similar trajectories in spatial networks," Journal of Systems and Software 82, no. 5, pp. 772-788, 2009.
- [98] H. Cao, N. Mamoulis, and D. Cheung, "Mining frequent spatio-temporal sequential patterns," In Data Mining, Fifth IEEE International Conference on, pp. 8-pp. IEEE, 2005.
- [99] A. Farhangfar, "A novel framework for imputation of missing values in databases," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 37, no. 5, pp. 692-709, 2007.
- [100] Foursquare API. Foursquare for developers. [Online] Available from: <https://developer.foursquare.com/> 2014.11.21.
- [101] Osm2po: Routing On [OpenStreetMap](#). [Online] Available from: <http://osm2po.de/> 2014.11.21.
- [102] Openstreetmap. [Online] Available from: <http://www.openstreetmap.org/> 2014.11.21.
- [103] Postgis: Spatial and Geographic objects for PostgreSQL. [Online] Available from: <http://postgis.net/> 2014.11.21.
- [104] PgRouting: osm2p. Routing On OpenStreetMap. [Online] Available from: <http://osm2po.de/> 2014.11.21.
- [105] Postgress 9.2. An object-relational database management system. [Online] Available from: <http://www.postgresql.com/> 2014.11.21.
- [106] Pentaho Data Integration 5.0.1. [Online] Available from: <http://www.pentaho.com/> 2014.11.21.



- [107] Fuel Economy: the official U.S. government source for fuel economy information. [Online] Available from: <http://www.fueleconomy.gov/> 2014.11.21.
- [108] Quantum Gis: A Free and Open Source Geographic Information System. [Online] Available from: <http://www.qgis.org/> 2014.11.21.
- [109] A. McAfee and E. Brynjolfsson, "Big data: the management revolution," Harvard Bus Rev 90, no. 10 pp. 61-67, 2012.
- [110] Apigee. [Online] Available from: <http://apigee.com/about/> 2014.11.21.