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

Edison Camilo Ospina Álvarez
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.
A Criteria based Function for Reconstructing Low-Sampling Trajectories as a Tool for Analytics

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

Abstract ........................................................................................................................................ VII
List of Figures .................................................................................................................................. X
List of Tables ................................................................................................................................... XIII
List of Equations ............................................................................................................................... XIV
List of Acronymus ............................................................................................................................ XIV

INTRODUCTION ............................................................................................................................. 1

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

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” ....................................................... 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

CHAPTER 3. LOW – SAMPLING TRAJECTORY RECONSTRUCTION USING CRITERIA BASED ROUTING OVER A GRAPH ................................................................................... 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
CHAPTER 4. USING CRITERIA RECONSTRUCTION OF LOW-SAMPLING TRAJECTORIES AS A TOOL FOR ANALYTICS ........................................... 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

CHAPTER 5. TECHNICAL DETAILS ................................................................. 78
  5.1 INTRODUCON ....................................................................................... 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

GENERAL CONCLUSIONS ........................................................................... 121
REFERENCES ............................................................................................ 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 \( e_k \) is the initial vertex of the edge \( e_k + 1 \) ................. 36
Figure 3.6. Inference of time stamp of edge \( e_k \) ...................................................................... 38
Figure 3.7. Example of time assignation to a reconstructed (sub)trajectory ............................... 39
Figure 3.8. Additional imputed data points for an edge \( e_k \) ..................................................... 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
XI. A Criteria based Function for Reconstructing Low-Sampling Trajectories as a Tool for Analytics

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

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) 114
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) 115
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 $e_k$ ........................................ 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 $get\_vertex\_target(e_k)$ ................................................................. 37
Equation 3.5. Line equation over the segment represented by $e_k$ ...................................................... 39
Equation 3.6. The $get\_x$ function slicing of the segment represented by $e_k$ ................................. 39
Equation 3.7. The $get\_y$ function slicing of the segment represented by $e_k$ ................................. 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
A Criteria based Function for Reconstructing Low-Sampling Trajectories as a Tool for Analytics
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, Flicker 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.
A Criteria based Function for Reconstructing Low-Sampling Trajectories as a Tool for Analytics

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

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 ](image)
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 describes the 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].
<table>
<thead>
<tr>
<th>Author</th>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>[11]</td>
<td>Routing systems</td>
<td>Routing systems aim to help users on finding the optimal path to their destination regarding travel distance, travel time, among other criteria.</td>
</tr>
<tr>
<td>[15]</td>
<td>Route planning technique</td>
<td>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.</td>
</tr>
<tr>
<td>[15]</td>
<td>Personalized user-centric route finding</td>
<td>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.</td>
</tr>
<tr>
<td>[16]</td>
<td>Personalized route planning systems</td>
<td>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.</td>
</tr>
<tr>
<td>[41]</td>
<td>Route guidance systems</td>
<td>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.</td>
</tr>
</tbody>
</table>

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 $d_{ij} = \sqrt[p]{(|x_i - x_j|^p + |y_i - y_j|^p)}$. 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](image)

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

---

1 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).

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

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.)
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 prepossessing 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 \textit{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 \textit{most popular} route between two given locations using historical user trajectories is addressed. A \textit{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 \textit{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
A Criteria based Function for Reconstructing Low-Sampling Trajectories as a Tool for Analytics

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 aproaches. 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 → 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.
A Criteria Based Function for Reconstructing Low-Sampling Trajectories as a Tool for Analytics

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\(_{i1}\), L\(_{i2}\), ..., L\(_{iM}\)\}. Each observation \( L_{ij} = (x_{ij}, y_{ij}, t_{ij}) \) represents the presence of an object at location \((x_{ij}, y_{ij})\) where \( x_{ij}, y_{ij} \in \mathbb{R} \), and at time \( t_{ij} \in \mathbb{T} \), where \( \mathbb{T} \) is a set of time points. The sequence of observations L\(_i\) is temporally ordered, i.e., \( t_{ij} < t_{ij+1} \). A sampling of 2D trajectories is defined as \( \mathcal{T}_S = \{T_i\} \). 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 \( \mathbf{C}_{\text{set}} \).
Consider the network-constrained trajectories ($\mathcal{T}$, $G_a$), where $\mathcal{T}$ is a set of trajectories and $G_a \in G$ ($G$ 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 $G_a$ is a two-tuple $G_a = (V, E)$, where $V$ is a set of vertices $\{v_i\}$ and $E$ 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 $v \in V$ can be described by a location $x, y$ (longitude, latitude). Note that we consider the graph $G_a$, which is derived from a RN, to be fully connected and without any isolated network segments.

Consider the following functions:

* $\text{get\_vertex\_source}: E \rightarrow V$. Function applied to an edge to get its source vertex.
* $\text{get\_vertex\_target}: E \rightarrow V$. Function applied to an edge to get its target vertex.
* $\text{get\_cost}: E \rightarrow \mathbb{R}$. Function applied to an edge to get the cost of traversing the edge.
* $\text{get\_x}: V \rightarrow X$. Function applied to a vertex to get its longitude.
* $\text{get\_y}: V \rightarrow Y$. 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](image)
The function \( \text{road\_distance}: \text{L} \times \text{L} \times \text{Cset} \rightarrow \mathbb{R} \) receives a pair of consecutive observations and a 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 criterion preferred by the MO (i.e., the user); therefore, the road distance may change 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: \( \text{road\_distance}(A, B, c_1) \leq \text{road\_distance}(A, B, c_2) \leq \text{road\_distance}(A, B, c_3) \). 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.

\[ \text{Figure 3.2. The concept of road distance according to different criteria vs the Euclidean distance between two observations A and B} \]

\[ \text{road\_distance}(A, B, c_1) \leq \text{road\_distance}(A, B, c_2) \leq \text{road\_distance}(A, B, c_3) \]
We regard the trajectory $T_i$ as low-sampled if \( \exists j, 1 \leq j \leq M, \left( \text{road distance} \left( L_i^j, L_i^{j+1}, c \right) \geq \beta \wedge t_i^{j+1} - t_i^j \geq \tau \right) \), 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 $\left( t_i^{j+1} - t_i^j \right)$ is greater than $\tau$ (a user time threshold).

We consider the $\text{traj}(L_i, c)$ function where $L_i \in 2^L$, is the sequence of M observations of a trajectory $T_i$ and $c \in C\text{set}$ is a reconstruction criterion. The result of the $\text{traj}$ 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 $c \in C\text{set}$. Next, we explain the effect of the $\text{traj}$ 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 $\left( x_i^j, y_i^j, t_i^j \right)$, its location $\left( x_i^j, y_i^j \right)$ 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 $\text{get\_edge}$ function, i.e., $\text{get\_edge}(L_i^j, Ga)$ and $\text{get\_edge}(L_i^{j+1}, Ga)$. The signature of the $\text{get\_edge}$ function is $L \times G \rightarrow E$. Here, the nearest edge in the graph $Ga = (V, E)$ is the output of the $\text{get\_edge}$ function. Therefore, a point $\left( x_i^j, y_i^j, t_i^j \right)$ that is not over an edge $\in E$ is replaced by a point $\left( x_i^j, y_i^j, t_i^j \right)$ where $\left( x_i^j, y_i^j \right)$ is over an edge of $E$, see Figure 3.3.
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.

\[
d(L_{i}^j, e_k) = \begin{cases} 
    d(L_{i}^j, e_k) & \text{if } L_{i}^j \in e_k \\
    \min \{ d(L_{i}^j, \text{get}_\text{vertex}_\text{source}(e_k)), d(L_{i}^j, \text{get}_\text{vertex}_\text{target}(e_k)) \} & \text{otherwise}
\end{cases}
\]

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 $d(L_{i}^j, \text{get}_\text{vertex}_\text{source}(e_k))$ and $d(L_{i}^j, \text{get}_\text{vertex}_\text{target}(e_k))$ 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, $\text{get}_\text{edge}(L_i^j, Ga) = e_k$. The main reason of the outcome of the $\text{get}_\text{edge}$ 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 \( a = (\text{get}_x(\text{get}_\text{vertex}\_\text{target}(\text{get}_\text{edge}(L_i^j, G_a))), \text{get}_y(\text{get}_\text{vertex}\_\text{target}(\text{get}_\text{edge}(L_i^j, G_a)))), \text{set}_\text{time}(L_i^j) \) and \( b = (\text{get}_x(\text{get}_\text{vertex}\_\text{source}(\text{get}_\text{edge}(L_i^{j+1}, G_a))), \text{get}_y(\text{get}_\text{vertex}\_\text{source}(\text{get}_\text{edge}(L_i^{j+1}, G_a)))), \text{set}_\text{time}(L_i^{j+1}) \). where we use the \( \text{set}_\text{time}: V \rightarrow \mathbb{T} \) 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, ..., 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 \( G_a \) Graph between its edges \( \text{get}_\text{edge}(L_i^j, G_a) \) and \( \text{get}_\text{edge}(L_i^{j+1}, G_a) \).

![Figure 3.4. Imputed observations between the observations a and b](image)

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:
Let:
then the time difference to compute the timestamp of each imputed observation of the reconstructed trajectory segment, the must be inferred. For this goal, we define the For each imputed observation used for reconstruct the trajectory between trajectory (sub)trajectory between $L_i^j$ and $L_i^{j+1}$

\[
\text{traj}([L_i^j, L_i^{j+1}], c) = (L_i^j, (get_x(get_vertex_target(e_1)), get_y(get_vertex_target(e_1)), set_time(get_vertex_target(e_1))), \ldots, (get_x(get_vertex_target(e_p)), get_y(get_vertex_target(e_p)), set_time(get_vertex_target(e_p))), L_i^{j+1})
\]

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 $\text{get_edge}(L_i^{j+1}, G) = e_p$.

**Edges in the RN represented by Graph Ga**

![Graph Ga](image)

*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 $\text{set_time}$ 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 → R. Function applied to an edge to get the road distance of the edge.

\( D_{L_i} = \) The distance from the observation \( L_i \) to \( \text{get}_\text{vertex} \text{source} \left( \text{get}_\text{edge} \left( L_i, G_a \right) \right) \)

\( D_{L_i+1} = \) The distance from \( L_i^{+1} \) to \( \text{get}_\text{vertex} \text{source} \left( \text{get}_\text{edge} \left( L_i^{+1}, G_a \right) \right) \)

Let:

\[
\text{total\_distance}(L_i, L_i^{+1}) = \text{get}_\text{distance} \left( \text{get}_\text{edge} \left( L_i, G_a \right) \right) + D_{L_i} + \sum_{k=2}^{p-1} \text{get}_\text{distance}(e_k) + D_{L_i^{+1}}.
\]

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

\[
\text{set\_time} \left( \text{get}_\text{vertex} \text{target} (e_k) \right) = \text{get\_time} \left( L_i \right) + \left( \frac{\sum_{k=1}^{p-1} \text{get\_distance} (e_k) - D_{L_i}}{\text{total\_distance}(L_i, L_i^{+1})} \right) \times \left( \text{get\_time} \left( L_i^{+1} \right) - \text{get\_time} \left( L_i \right) \right)
\]

In Figure 3.6, the reconstructed trajectory between two observations \( L_i \) and \( L_i^{+1} \) is shown using the \( \text{traj} \) function according to a criterion \( c \).
Example. Let us consider the reconstructed trajectory between the observations $L^1_1$ and $L^2_1$ 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 $get\_time(L^2_1) = 03:00:00$, then $get\_time(L^2_1) - get\_time(L^1_1) = 1$ hour must be proportionally divided among the edges.

Let the $get\_distance(e_1) = 12$, $get\_distance(e_2) = 10$, $get\_distance(e_3) = 10$, $get\_distance(e_4) = 10$, $D_{L^1_1} = 2$, $D_{L^2_1} = 2$. Also, note that $get\_edge(L^1_1, G) = e_1$ and $total\_distance(L^1_1, L^2_1) = 40$.

For $k = 1$

$$set\_time(get\_vertex\_target(e_1)) = get\_time(L^1_1) + \frac{get\_distance(e_1) - D_{L^1_1}}{total\_distance(L^1_1, L^2_1)} \times (get\_time(L^2_1) - get\_time(L^1_1)) = 02:00:00 + \frac{1}{4} = 02:15:00$$

For $k = 2$

$$set\_time(get\_vertex\_target(e_2)) = get\_time(L^1_1) + \frac{get\_distance(e_1) + get\_distance(e_2) - D_{L^1_1}}{total\_distance(L^1_1, L^2_1)} \times (get\_time(L^2_1) - get\_time(L^1_1)) = 02:00:00 + \frac{2}{4} = 02:30:00$$
For $k = 3$

\[
\text{set_time (get\_vertex\_target(e_3))} = \text{get\_time}(L_1) + \\
\frac{\text{get\_distance}(e_1) + \text{get\_distance}(e_2) + \text{get\_distance}(e_3) - D_{L_1}}{\text{total\_distance}(L_1, L_2)} \times (\text{get\_time}(L_2) - \text{get\_time}(L_1)) = \\
02:00:00 + \frac{3}{4} = 02:45:00
\]

![Diagram](image)

**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 $e_k$**

\[
y = \frac{\text{get\_y}\text{(get\_vertex\_target}(e_k)) - \text{get\_y}\text{(get\_vertex\_source}(e_k))}{\text{get\_x}\text{(get\_vertex\_target}(e_k)) - \text{get\_x}\text{(get\_vertex\_source}(e_k))} \times (x - \text{get\_x} (\text{get\_vertex\_target}(e_k))) + \\
\text{get\_y}(\text{get\_vertex\_target}(e_k))
\]

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

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

\[
x_i = \frac{\text{get\_x}(\text{get\_vertex\_source}(e_k)) + \frac{d_i}{\text{road\_distance}(L_{i,i+1}, c)} \times (\text{get\_x}(\text{get\_vertex\_target}(e_k)) - \\
\text{get\_x}(\text{get\_vertex\_source}(e_k)))}{\text{get\_x}(\text{get\_vertex\_source}(e_k)) + \frac{d_i}{\text{road\_distance}(L_{i,i+1}, c)} \times (\text{get\_x}(\text{get\_vertex\_target}(e_k)) - \\
\text{get\_x}(\text{get\_vertex\_source}(e_k)))}
\]
A Criteria based Function for Reconstructing Low-Sampling Trajectories as a Tool for Analytics

Equation 3.7. The get_y function slicing of the segment represented by $e_k$

\[
y_i = \text{get}_y(\text{get}_\text{vertex}_\text{source}(e_k)) + \frac{d_i}{\text{road_distance}(L_i^{jx}, L_i^{jx+1}, c)} \times (\text{get}_y(\text{get}_\text{vertex}_\text{target}(e_k)) - \text{get}_y(\text{get}_\text{vertex}_\text{source}(e_k)))
\]

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

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

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

\[d_1 = 1.25\]

\[
x_i = 3 + \frac{1.25}{5} \times (6-3) = 3 + \frac{1.25}{5} \times 3 = 3.75
\]

\[
y_i = 1 + \frac{1.25}{5} \times (5-1) = 1 + \frac{1.25}{5} \times 4 = 2
\]

\[d_2 = 2.5\]

\[
x_i = 1 + \frac{2.5}{5} \times (6-3) = 3 + \frac{2.5}{5} \times 3 = 4.5
\]

\[
y_i = 1 + \frac{2.5}{5} \times (5-1) = 1 + \frac{2.5}{5} \times 4 = 3
\]

\[d_3 = 3.75\]

\[
x_i = 1 + \frac{3.75}{5} \times (6-3) = 3 + \frac{3.75}{5} \times 3 = 5.25
\]

\[
y_i = 1 + \frac{3.75}{5} \times (5-1) = 1 + \frac{3.75}{5} \times 4 = 4\]
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 \( \mathcal{T} \mathcal{S} = \{ T_i \} \) 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 \( T'_i \in \mathcal{T} \mathcal{S} \). Algorithm 1 calls the Function 1 (\( \text{traj} \)) for each pair of observations that make up the trajectory in a determined dataset \( \mathcal{T} \mathcal{S} \).

### 3.4.1 Algorithm 1: Reconstruction of a Trajectory

**Algorithm 1: Reconstruction of a Trajectory**

**INPUT:** \( \{ \mathcal{T} \mathcal{S} \mid \forall T_i \in \mathcal{T} \mathcal{S}, \exists L_i, L_i^{j+1} \mid \text{road.distance}(L_i, L_i^{j+1}, c) \geq \beta \land t_i^j - t_i^{j+1} \geq \tau \} \)

**c \in Cset**

**OUTPUT:** \( \{ \mathcal{T} \mathcal{S}' \mid \forall T_i \in \mathcal{T} \mathcal{S}', \text{road.distance}(L_i, L_i^{j+1}, c) \geq \beta' \land t_i^j - t_i^{j+1} \geq \tau' \land \beta' \leq \beta \land \tau' \leq \tau \} \)

\( \mathcal{T} \mathcal{S}' \leftarrow \emptyset \)

\( T_i' \leftarrow \emptyset \)

**For each** \( T_i \) **in** \( \mathcal{T} \mathcal{S} \)

**For each** \( L_i^j \) **in** \( T_i \)

**if** \( \text{road.distance}(L_i^j, L_i^{j+1}, c) \geq \beta \land t_i^j - t_i^{j+1} \geq \tau \) **then**

Trajectory \( \leftarrow \text{traj}(\{ L_i^j, L_i^{j+1} \}, c) \)

**Append** Trajectory **to** \( T_i' \)

**else**

**Append** \( \{ L_i^j, L_i^{j+1} \} \) **to** \( T_i' \)

**Next** \( L_i^j \)

**End**

**Append** \( T_i' \) **to** \( \mathcal{T} \mathcal{S}' \)

**End**

**Return** \( \mathcal{T} \mathcal{S}' \)
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:** \{1, 1[+1] | road_distance(1,1[+1], c) ≥ \(β \land t_1^i - t_1^{i+1} ≥ \tau\) \}

\(c \in \text{Cset}\)

**OUTPUT:** \{1, 1[+1] | road_distance(1,1[+1], c) ≥ \(β' \land t_1^i - t_1^{i+1} ≥ \tau' \land β' < β \land \tau' < \tau\) \}

// To apply a routing algorithm according to c criterion between get_edge(1, Ga) and get_edge(1[+1], Ga)

For each \(e_k\)

// Use set_time function for setting the time to each vertex resulting from the routing algorithm
\(O_k \leftarrow \{(\text{get}_x(\text{get}_\text{vertex}_\text{target}(e_k)), \text{get}_y(\text{get}_\text{vertex}_\text{target}(e_k)), \text{set}_\text{time}(\text{get}_\text{vertex}_\text{target}(e_k)))\}\)

If \(\text{road_distance}(O_k, O_{k+1}, c) ≥ \beta \land \text{get}_\text{time}(O_{k+1}) - \text{get}_\text{time}(O_k) ≥ \tau\) then

// interpolate between \(O_k\) and \(O_{k+1}\)

Use the equation 3.6 and equation 3.7

end

Trajectory \(\leftarrow\)

\{\{(\text{get}_x(\text{get}_\text{vertex}_\text{target}(e_1)), \text{get}_y(\text{get}_\text{vertex}_\text{target}(e_1)), \text{set}_\text{time}(\text{get}_\text{vertex}_\text{target}(e_1)))\}, \ldots, \{(\text{get}_x(\text{get}_\text{vertex}_\text{target}(e_p)), \text{get}_y(\text{get}_\text{vertex}_\text{target}(e_p)), \text{set}_\text{time}(\text{get}_\text{vertex}_\text{target}(e_p)))\}\}\)

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 Ga) shown in Figure 3.10. We also get the nearest edges \_edge(Check – in A, Ga), \_edge(Check – in B, Ga) and \_edge(check – in C, Ga), for each check-in. Those road segments are depicted in solid lines in Figure 3.10:

<table>
<thead>
<tr>
<th>User</th>
<th>Data point</th>
<th>POI name</th>
<th>(x,y,t)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15307763</td>
<td>Check-in A</td>
<td>Shop</td>
<td>(-75.5625555,6.249437,20140809134345)</td>
</tr>
<tr>
<td>15307763</td>
<td>Check-in B</td>
<td>Restaurant</td>
<td>(-75.576796,6.244406;20140809145517)</td>
</tr>
<tr>
<td>15307763</td>
<td>Check-in C</td>
<td>Shop</td>
<td>(-75.591672,6.257514,20140809173745)</td>
</tr>
</tbody>
</table>

Table 3.1. Check – in data of a particular user.
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 $\text{traj}([\text{Check} \rightarrow \text{in} \ A, \text{Check} \rightarrow \text{in} \ B], c)$ with $c = \text{Distance}$. The A * algorithm is used to find the imputed location data between $\text{get\_vertex\_target}(\text{get\_edge}(\text{Check} \rightarrow \text{in} \ A, Ga))$ and $\text{get\_vertex\_target}(\text{get\_edge}(\text{Check} \rightarrow \text{in} \ B, Ga))$ using the $\text{get\_x}$ and $\text{get\_y}$ functions. At the same time, the timestamps for those location data were set using the `set\_time` function and assigning proportionally the difference $\text{get\_time}(\text{Check} \rightarrow \text{in} \ B) - \text{get\_time}(\text{Check} \rightarrow \text{in} \ A)$. 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.
A Criteria based Function for Reconstructing Low-Sampling Trajectories as a Tool for Analytics

<table>
<thead>
<tr>
<th>User</th>
<th>(( \text{get}<em>x(\text{get}</em>\text{vertex}<em>\text{source}(e_k)), \text{get}<em>y(\text{get}</em>\text{vertex}</em>\text{source}(e_k)), \text{set}<em>\text{time}(\text{get}</em>\text{vertex}_\text{source}(e_k)) ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>((-75.5625553,6.2494373,20140809134345))</td>
</tr>
<tr>
<td>User 1</td>
<td>((-75.5620212,6.2491409,20140809134656))</td>
</tr>
<tr>
<td>User 1</td>
<td>((-75.5629924,6.2496343,20140809134717))</td>
</tr>
<tr>
<td>User 1</td>
<td>((-75.5635239,6.2488592,20140809135054))</td>
</tr>
<tr>
<td>User 1</td>
<td>((-75.5620212,6.2491409,20140809134717))</td>
</tr>
<tr>
<td>User 1</td>
<td>(...)</td>
</tr>
<tr>
<td>User 1</td>
<td>((-75.5754726,6.2450224,20140809144727))</td>
</tr>
<tr>
<td>User 1</td>
<td>((-75.5754846,6.2450904,20140809144748))</td>
</tr>
<tr>
<td>User 1</td>
<td>((-75.5760523,6.2451252,20140809144748))</td>
</tr>
<tr>
<td>User 1</td>
<td>((-75.5767275,6.2437119,20140809145314))</td>
</tr>
<tr>
<td>User 1</td>
<td>((-75.5767906,6.244046,20140809145517))</td>
</tr>
</tbody>
</table>

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(trajectory) from Check-in B to Check-in C is computed using \( \text{traj}(\{\text{Check-in B, Check-in C}\}, c) \) with \( c = \text{Distance} \). 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.
A Criteria based Function for Reconstructing Low-Sampling Trajectories as a Tool for Analytics

<table>
<thead>
<tr>
<th>User</th>
<th>(( \text{get}<em>x(\text{get}</em>\text{vertex}<em>\text{source}(e_k)), \text{get}<em>y(\text{get}</em>\text{vertex}</em>\text{source}(e_k)), \text{set}<em>\text{time}(\text{get}</em>\text{vertex}_\text{source}(e_k)) ) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>(-75.576790, 6.244406, 20140809145517)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5767275, 6.2437119, 20140809150356)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5776115, 6.244002, 20140809150903)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5777403, 6.2440447, 20140809150948)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5783803, 6.2442404, 20140809151329)</td>
</tr>
<tr>
<td>User 1</td>
<td>...</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5924922, 6.2564856, 20140809172702)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5925836, 6.2566938, 20140809172817)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5921396, 6.2577242, 20140809173431)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5916435, 6.2574905, 20140809173732)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.591672, 6.257514, 20140809173745)</td>
</tr>
</tbody>
</table>

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

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

Imputed trajectory using the Time criterion

Now, the criterion \( c = \text{Time} \) is set. The (sub)trajectory from Check-in A to Check-in B is computed using the \( \text{traj} \) function \( \text{traj}([\text{Check in A}, \text{Check in B}], c) \). The Dijkstra's algorithm is used to find the imputed location data between \( \text{get_vertex_target}(\text{get_edge}(\text{Check in A, Ga})) \) and \( \text{get_vertex_target}(\text{get_edge}(\text{Check in B, Ga})) \) using the \( \text{get}_x \) and \( \text{get}_y \) functions. The difference \( \text{get}_x(\text{Check in B}) - \text{get}_x(\text{Check in A}) \) was proportionally assigned using the \( \text{set}_x \) 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.

<table>
<thead>
<tr>
<th>User</th>
<th>( ((\text{get}_x(\text{get_vertex_source}(e_k)), \text{get}_y(\text{get_vertex_source}(e_k)), \text{set}_x(\text{get_vertex_source}(e_k)))) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>(-75.5625555,6.2494373,20140809134345)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5620212,6.2491409,20140809134656)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5623187,6.2483562,20140809134924)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5623576,6.2482814,20140809134939)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5635619,6.2487949,20140809135331)</td>
</tr>
<tr>
<td>User 1</td>
<td>...</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5754726,6.2450224,20140809144819)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5759484,6.2450904,20140809144838)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5760523,6.2451252,20140809144838)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5767275,6.2437119,20140809145314)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5767905,6.244064,20140809145517)</td>
</tr>
</tbody>
</table>

Table 3.4. Inferred observations using the Time criterion between Check-in A to Check-in B.
A Criteria based Function for Reconstructing Low-Sampling Trajectories as a Tool for Analytics

Next, from Check-in B to Check-in C the (sub)trajectory is computed using \( \text{traj}( \{ \text{Check — in B, Check — in C} \}, c) \) 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.

<table>
<thead>
<tr>
<th>User</th>
<th>((\text{get}<em>x(\text{get}</em>\text{vertex}<em>\text{source}(e_k)), \text{get}<em>y(\text{get}</em>\text{vertex}</em>\text{source}(e_k)), \text{set}<em>\text{time}(\text{get}</em>\text{vertex}_\text{source}(e_k))))</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>(-75.5767905, 6.244064, 20140809145517)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5767275, 6.2437119, 20140809150317)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5776115, 6.244002, 20140809150801)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5777403, 6.244047, 20140809150843)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5783803, 6.244204, 20140809151207)</td>
</tr>
<tr>
<td>User 1</td>
<td>...</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5924922, 6.2564856, 20140809172750)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5925836, 6.2566938, 20140809172900)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5921396, 6.2577242, 20140809173445)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5916435, 6.2574905, 20140809173733)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5916722, 6.2575147, 20140809173745)</td>
</tr>
</tbody>
</table>

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

Figure 3.14. Reconstructed Trajectory between Check-in A and Check-in B. using the Time criterion from the user 1.
A Criteria based Function for Reconstructing Low-Sampling Trajectories as a Tool for Analytics

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 \( c = \text{Touristic} \) is set. The (sub)trajectory from Check-in A to Check-in B is computed using \( \text{traj}((\text{Check in A}, \text{Check in B}), c) \). 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.

<table>
<thead>
<tr>
<th>User</th>
<th>( ((\text{get}<em>x(\text{get}</em>\text{vertex}<em>\text{source}(e_k)), \text{get}<em>y(\text{get}</em>\text{vertex}</em>\text{source}(e_k)), \text{set}<em>\text{time}(\text{get}</em>\text{vertex}_\text{source}(e_k))) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>(-75.5625555, 6.2494373, 20140809134345)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5620212, 6.2491409, 20140809134717)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5629924, 6.2496343, 20140809134717)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5635239, 6.2488592, 20140809135050)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5620212, 6.2491409, 20140809135050)</td>
</tr>
<tr>
<td>User 1</td>
<td>...</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5760523, 6.2451252, 20140809144731)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5759484, 6.2450904, 20140809144753)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5760523, 6.2451252, 20140809144753)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5767275, 6.2437119, 20140809145300)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5767905, 6.2444064, 20140809145517)</td>
</tr>
</tbody>
</table>

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.](image-url)
Next, the (sub)trajectory from Check-in B to Check-in C is computed using $\text{traj}([\text{Check-in B,Check-in C}], c)$ 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.

<table>
<thead>
<tr>
<th>User</th>
<th>((\text{get}<em>x(\text{get}</em>\text{vertex}<em>\text{source}(e_k)), \text{get}<em>y(\text{get}</em>\text{vertex}</em>\text{source}(e_k)), \text{set}<em>\text{time}(\text{get}</em>\text{vertex}_\text{source}(e_k))))</th>
</tr>
</thead>
<tbody>
<tr>
<td>User 1</td>
<td>(-75.5767905, 6.2444064, 20140809145517)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5767275, 6.2437119, 20140809150344)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5760523, 6.2451252, 20140809150344)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5767275, 6.2437119, 20140809151211)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5758606, 6.2456828, 20140809151211)</td>
</tr>
<tr>
<td>User 1</td>
<td>...</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5924922, 6.2564856, 20140809172717)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5925836, 6.2566938, 20140809172830)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5921396, 6.2577242, 20140809173435)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5916435, 6.2574905, 20140809173732)</td>
</tr>
<tr>
<td>User 1</td>
<td>(-75.5916722, 6.2575147, 20140809173745)</td>
</tr>
</tbody>
</table>

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.
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.

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](image)

*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.

![Average SIMPOI](image)

**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 met 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.

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 (Medellin)
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.
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.
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.

![Data warehouse architecture including traj function](image)

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.
Also, we need to load the graph Ga 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 Ga 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 Ga 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

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

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

![Dimensional Model of the Data warehouse proposal](image)

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$_j$ is reconstructed for each 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
A Criteria based Function for Reconstructing Low-Sampling Trajectories as a Tool for Analytics

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

<table>
<thead>
<tr>
<th>MO ID</th>
<th>Trajectory ID</th>
<th>Observation</th>
<th>RN ID</th>
<th>time</th>
<th>Criterion ID</th>
<th>Measure1</th>
<th>...</th>
<th>Measure n</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDi</td>
<td>Tj</td>
<td>[L^1, L^2]</td>
<td>RN_x</td>
<td>[t^1, t^2]</td>
<td>C_1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDi</td>
<td>Tj</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDi</td>
<td>Tj</td>
<td>[L^1, L^1+1]</td>
<td>RN_x</td>
<td>[t^1, t^1+1]</td>
<td>C_1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDi</td>
<td>Tj</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDi</td>
<td>Tj</td>
<td>[L^1, L^1-M+1, L^1-M]</td>
<td>RN_x</td>
<td>[t^1, t^1^M]</td>
<td>C_1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDi</td>
<td>Tj</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDi</td>
<td>Tj</td>
<td>[L^1, L^N]</td>
<td>RN_x</td>
<td>[t^1, t^N]</td>
<td>C_p</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDi</td>
<td>Tj</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDi</td>
<td>Tj</td>
<td>[L^1, L^1-M+1, L^1-M]</td>
<td>RN_x</td>
<td>[t^1, t^1^M]</td>
<td>C_p</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IDi</td>
<td>Tj</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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, observationlinitial, observationlfinal, distance, fuelconsumption,
FROM facttrajectory
```

![Figure 4.10. A factTrajectory fact table example](image-url)
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.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity of trajectories</td>
<td>Count all distinct trajectory ids that pass through a street segment</td>
</tr>
<tr>
<td>Quantity of users</td>
<td>Count all the MO IDs that pass through a street</td>
</tr>
<tr>
<td>Total Distance Traveled</td>
<td>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.</td>
</tr>
<tr>
<td>Total Travel Duration</td>
<td>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.</td>
</tr>
<tr>
<td>Fuel consumption</td>
<td>Adds up the fuel consumption according to the distance traveled</td>
</tr>
<tr>
<td>CO2 emissions</td>
<td>Adds up the co2 emission according to the distance traveled</td>
</tr>
</tbody>
</table>

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.

Also, measures such as CO2 emissions can be used for analysis in function of distance traveled [107]. Suppose that the set of MO analyzed here emit 300 grams of CO2 per km. In Figure 4.12, CO2 emissions (grams per km) are sliced by criteria and day between August 4, 2014 and August 10, 2014 are shown.

Figure 4.11. Fuel Consumption (litres / 100km) 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.
A Criteria based Function for Reconstructing Low-Sampling Trajectories as a Tool for Analytics

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.
SELECT criteria.CriterionDesc, \text{AVG}(\text{fact.distance})
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 >= 20140808
AND TimeFin.IdTime <= 20140808
GROUP BY 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 enhancing 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 has 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.

![Example of a Json File of the user response from the API Foursquare.](image.png)
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](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\VenuesList.js where 0<i<21) living in the Medellin, Colombia city were collected.

![Figure 5.2. Example of a Json File of the venue response from the API Foursquare](image)

5.3.1.3 Check-in

Get details of a check-in ([https://developer.foursquare.com/docs/checkins/checkins](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)
5.3.2 Point of Interest.

A list of touristic points of Medellín, Colombia city were defined. Those were extracted from OpenStreetMap were people can tagged 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.

![Example of a Json File of the check-in response from the API Foursquare.](image-url)
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 Medellin, 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 Medellin 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.

![Figure 5.5. An example of “colombiarn_2po_4pgr_medellin” table](image-url)
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.

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.
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 DBObjects\staging schema\tables\stg_check_in_data.sql. An example of the stg_check_in_data table is shown in Figure 5.8.

![Figure 5.8. An example of “stg_check_in_data” table](image)

5.4.1.6 stg_pois

The stg_pois table stores the content of the most representative POIs of the city of Medellin. See Section 5.3.2. The stg_pois table definition can be found in DBObjects\staging schema\tables\stg_pois.sql. An example of the stg_pois table is shown in Figure 5.9.
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.

Figure 5.9. An example of “stg_pois” table

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 into a view respectively. The Table 5.1 show the respective table, the view assigned and the source file to be executed.

<table>
<thead>
<tr>
<th>Table</th>
<th>view</th>
<th>source</th>
</tr>
</thead>
<tbody>
<tr>
<td>stg_users</td>
<td>vw_stg_users</td>
<td>\DBObjects\staging\schema\views\vw_stg_users.sql</td>
</tr>
<tr>
<td>stg_venues</td>
<td>vw_stg_venues</td>
<td>\DBObjects\staging\schema\views\vw_stg_venues.sql</td>
</tr>
<tr>
<td>stg_check_in_data</td>
<td>vw_stg_checkindata</td>
<td>\DBObjects\staging\schema\views\vw_stg_checkindata.sql</td>
</tr>
<tr>
<td>stg_pois</td>
<td>vw_stg_pois</td>
<td>\DBObjects\staging\schema\views\vw_stg_pois.sql</td>
</tr>
<tr>
<td>colombiarn_2po_4pgr_medellin</td>
<td>vw_roadnetwork</td>
<td>\DBObjects\staging\schema\views\vw_roadnetwork.sql</td>
</tr>
</tbody>
</table>

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 tesis. 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\Observation.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 `DBObjectsTDW\schema\tables\factTrajectory.sql`. An example of the factTrajectory table is shown in Figure 5.11.

![FactTrajectory Table Example](image.png)

**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 `DBObjectsTDW\schema\tables\dimroadnetwork.sql`. An example of the Dimroadnetwork dimension table is shown in Figure 5.12.
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 distinguishes 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.

![Figure 5.13. An example of the dimcriteria dimension table](image)

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.
### 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\tables\DimMovingObject.sql`. An example of the `DimMovingObject` dimension table is shown in Figure 5.15.

![An example of the DimMovingObject dimension table](image)

**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\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\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

<table>
<thead>
<tr>
<th>File</th>
<th>\ETL Solution\jobs\jobPrincipal.kbj</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>jobPrincipal</td>
</tr>
<tr>
<td>Description</td>
<td>Principal Job than orchestrates the Stage loading and TDW loading together.</td>
</tr>
</tbody>
</table>

Table 5.2. Job: jobPrincipal

5.6.1.2 Job: jobLoadStage

<table>
<thead>
<tr>
<th>File</th>
<th>\ETL Solution\jobs\jobLoadStage.kbj</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>jobLoadStage</td>
</tr>
<tr>
<td>Description</td>
<td>Job that orchestrates the load and transformation in the stage data area</td>
</tr>
</tbody>
</table>

Table 5.3. Job: jobLoadStage
5.6.1.3 Job: jobLoadTDW

<table>
<thead>
<tr>
<th>File</th>
<th>\ETL Solution\jobs\jobLoadTDW.kbj</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>jobLoadTDW</td>
</tr>
<tr>
<td>Description</td>
<td>Job that orchestrates the load of TDW</td>
</tr>
</tbody>
</table>

Table 5.4. Job: jobLoadTDW

5.6.1.4 Job: jobExtractData

<table>
<thead>
<tr>
<th>File</th>
<th>\ETL Solution\jobs\jobExtractData.kbj</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>jobExtractData</td>
</tr>
<tr>
<td>Description</td>
<td>Job that orchestrates the extract of user, venues, pois and check in data</td>
</tr>
</tbody>
</table>

Table 5.5. Job: jobExtractData
### 5.6.1.5 Job: `jobReconstructTrajectories`

<table>
<thead>
<tr>
<th>File</th>
<th><code>\ETL SOLUTION\jobs\jobReconstructTrajectories.kbj</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td><code>jobReconstructTrajectories</code></td>
</tr>
<tr>
<td>Description</td>
<td>Job that orchestrates the reconstruction of trajectories</td>
</tr>
</tbody>
</table>

*Table 5.6. Job: `jobReconstructTrajectories`*

### 5.6.1.6 Job: `jobLoadDimensions`

<table>
<thead>
<tr>
<th>File</th>
<th><code>\ETL SOLUTION\jobs\jobLoadDimensions.kbj</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td><code>jobLoadDimensions</code></td>
</tr>
<tr>
<td>Description</td>
<td>Job that orchestrates the load of Trajectory Datawarehouse Dimensions</td>
</tr>
</tbody>
</table>

*Table 5.7. Job: `jobLoadDimensions`*
5.6.1.7 Job: jobLoadFactTrajectory

<table>
<thead>
<tr>
<th>File</th>
<th>ETL Solution\jobs\jobLoadFactTrajectory.kbj</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>jobLoadFactTrajectory</td>
</tr>
<tr>
<td>Description</td>
<td>Job that orchestrates the load of TDW FactTrajectory</td>
</tr>
</tbody>
</table>

Table 5.8. Job: jobLoadFactTrajectory

5.6.2 Transformations

5.6.2.1 Transformation: traExtractUserData

<table>
<thead>
<tr>
<th>File</th>
<th>ETL Solution\transformations\traExtractUserData.ktr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>traExtractUserData</td>
</tr>
<tr>
<td>Description</td>
<td>Transformation that load the user data from foursquare</td>
</tr>
</tbody>
</table>

Table 5.9. Transformation: traExtractUserData
5.6.2.2 Transformation: traExtractVenuesData

<table>
<thead>
<tr>
<th>File</th>
<th>\ETL Solution\transformations\traExtractVenuesData.ktr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>traExtractVenuesData</td>
</tr>
<tr>
<td>Description</td>
<td>Transformation that load the venues data from foursquare</td>
</tr>
</tbody>
</table>

![Diagram](https://via.placeholder.com/150)

*Table 5.10. Transformation: traExtractVenuesData*

5.6.2.3 Transformation: traExtractCheckinData

<table>
<thead>
<tr>
<th>File</th>
<th>\ETL Solution\transformations\traExtractCheckinData.ktr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>traExtractCheckinData</td>
</tr>
<tr>
<td>Description</td>
<td>Transformation that load the check-in data from Foursquare</td>
</tr>
</tbody>
</table>

![Diagram](https://via.placeholder.com/150)

*Table 5.11. Transformation: traExtractCheckinData*
A Criteria based Function for Reconstructing Low-Sampling Trajectories as a Tool for Analytics

5.6.2.4 Transformation: traExtractPOIData

<table>
<thead>
<tr>
<th>File</th>
<th>\ETL Solution\transformations\traExtractPOIData.ktr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>traExtractPOIData</td>
</tr>
<tr>
<td>Description</td>
<td>Transformation that load the POI data from OpenstreetMap</td>
</tr>
</tbody>
</table>

Table 5.12. Transformation: traExtractPOIData

5.6.2.5 Transformation: traSetCostforCriteria

<table>
<thead>
<tr>
<th>File</th>
<th>\ETL Solution\transformations\traSetCostforCriteria.ktr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>traSetCostforCriteria</td>
</tr>
<tr>
<td>Description</td>
<td>Transformation set the Cost for Criteria in the RN</td>
</tr>
</tbody>
</table>

Table 5.13. Transformation: traSetCostforCriteria
5.6.2.6 Transformation: traSetPointProjection

<table>
<thead>
<tr>
<th>File</th>
<th>\ETL Solution\transformations\traSetPointProjection.ktr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>traSetPointProjection</td>
</tr>
<tr>
<td>Description</td>
<td>Transformation that finds the nearest edge</td>
</tr>
</tbody>
</table>

Table 5.14. Transformation: traSetPointProjection

5.6.2.7 Transformation: traSetPointProjection

<table>
<thead>
<tr>
<th>File</th>
<th>\ETL Solution\transformations\traSetPointProjection.ktr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>traSetPointProjection</td>
</tr>
<tr>
<td>Description</td>
<td>Transformation that finds the nearest edge</td>
</tr>
</tbody>
</table>

Table 5.15. Transformation: traSetPointProjection
5.6.2.8 Transformation: traReconstructTrajectory

<table>
<thead>
<tr>
<th>File</th>
<th>\ETL Solution\transformations\traReconstructTrajectory.ktr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>traReconstructTrajectory</td>
</tr>
<tr>
<td>Description</td>
<td>Transformation that implements the reconstruction of trajectories. It calls the traj function.</td>
</tr>
</tbody>
</table>

Table 5.16. Transformation: traReconstructTrajectory

5.6.2.9 Transformation: traLoadDimRoadNetwork

<table>
<thead>
<tr>
<th>File</th>
<th>\ETL Solution\transformations\traLoadDimRoadNetwork.ktr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>traLoadDimRoadNetwork</td>
</tr>
<tr>
<td>Description</td>
<td>Transformation that load the dimension of Road Network</td>
</tr>
</tbody>
</table>

Table 5.17. Transformation: traLoadDimRoadNetwork
5.6.2.10 Transformation: traLoadDimCriterion

<table>
<thead>
<tr>
<th>File</th>
<th>\ETL Solution\transformations\traLoadDimCriterion.ktr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>traLoadDimCriterion</td>
</tr>
<tr>
<td>Description</td>
<td>Transformation that load the dimension of Criteria</td>
</tr>
</tbody>
</table>

*Table 5.18. Transformation: traLoadDimCriterion*

5.6.2.11 Transformation: traLoadDimMovingObject

<table>
<thead>
<tr>
<th>File</th>
<th>\ETL Solution\transformations\traLoadDimMovingObject.ktr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>traLoadDimMovingObject</td>
</tr>
<tr>
<td>Description</td>
<td>Transformation that load the dimension of Moving Objects</td>
</tr>
</tbody>
</table>

*Table 5.19. Transformation: traLoadDimMovingObject*
5.6.2.12 Transformation: traLoadDimTrajectory

<table>
<thead>
<tr>
<th>File</th>
<th>\ETL Solution\transformations\traLoadDimTrajectory.ktr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>traLoadDimTrajectory</td>
</tr>
<tr>
<td>Description</td>
<td>Transformation that load the dimension of Trajectories</td>
</tr>
</tbody>
</table>

Table 5.20. Transformation: traLoadDimTrajectory

5.6.2.13 Transformation: traLoadFactTrajectory

<table>
<thead>
<tr>
<th>File</th>
<th>\ETL Solution\transformations\traLoadFactTrajectory.ktr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>traLoadFactTrajectory</td>
</tr>
<tr>
<td>Description</td>
<td>Transformation that load the reconstructed the low-sampling trajectories to the facttrajectory fact table</td>
</tr>
</tbody>
</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)](image)
5.7.1.2 Check-in data on August 5, 2014 (Medellín)

![Map of check-in points on August 5, 2014 (Medellín)](image)

*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)

![Map of check-in points on August 7, 2014 (Medellín)](image)

*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)

![Image of check-in points 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)
A Criteria based Function for Reconstructing Low-Sampling Trajectories as a Tool for Analytics

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

5.7.2.4 The reconstructed trajectories on August 7, 2014.

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

Figure 5.32. Reconstructed trajectories using Distance criterion on August 7, 2014 (Medellín)
A Criteria based Function for Reconstructing Low-Sampling Trajectories as a Tool for Analytics

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)
5.7.2.6 The reconstructed trajectories on August 9, 2014.

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

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)*
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

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


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


