Travel Time Estimation Using Sparsely Sampled Probe GPS Data in Urban Road Networks Context

Amnir Hadachi

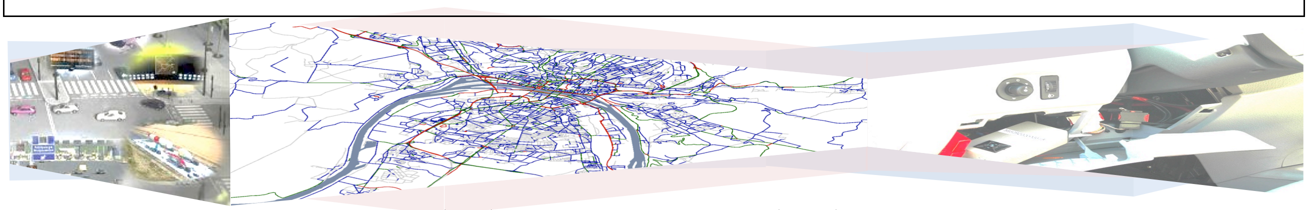
Normandie University
National Institute of Applied Sciences of Rouen
Computer Science, Information Processing and Systems Laboratory

Doctorate Thesis
Major: Computer Science

A Dissertation Presented
By
AMNIR HADACHI

Submitted
for the fulfillment of the requirements for the degree of
DOCTOR of Normandie University
INSA of Rouen

Travel Time Estimation Using Sparsely Sampled Probe GPS Data in Urban Road Networks Context



Fabrice Meriaudeau	Le2i Laboratory, Bourgogne University, France	**Rapporteur**
Fawzi Nashashibi	INRIA Research Center, Paris, France	**Rapporteur**
Alberto Broggi	VisLab, Parma University, Italy	**Examiner**
Jacques Jacot	Laboratory of Microengineering for Manufacturing EPFL, Switzerland	**Examiner**
Abdelaziz Bensrhair	LITIS Laboratory, INSA of Rouen, France	**Thesis Director**
Stephane Mousset	LITIS Laboratory, INSA of Rouen, France	**Supervisor**
Christele Lecomte	LITIS Laboratory, Rouen University, France	**Supervisor**
Bernard Matyjasik	Egis , France	**Guest**

Spring 2013

Travel Time Estimation Using Sparsely Sampled Probe GPS Data in Urban Road Networks Context

By
Amnir Hadachi

Abstract:

This dissertation is concerned with the problem of estimating travel time per road sections in urban context using sparsely sampled GPS data. One of the challenges in this thesis is the use of sparsely sampled data. The thesis work and the report were done in a period of two years and half.

The thesis research work was conducted within the project PUMAS, which is an advantage for our research regarding the collection process of our data from the real world field and also in making our tests. The project PUMAS (Plateforme Urbaine de Mobilite Avancee et Soutenable / Urban Platform for Sustainable and Advanced Mobility) is a preindustrial project that has the objective to inform about the traffic situation and also to develop and implement a platform for sustainable mobility in order to evaluate it in the region, specifically Rouen, France. The result is a framework for any traffic controller or manager and also estimation researcher to access vast stores of data about the traffic estimation, forecasting, and status.

Sparsely sampled probe GPS data refers to the case where vehicle send their current location at a fixed frequency. The frequency aspect makes it not enough to directly measure the speed or travel time on the road section.
In order to overcome these challenges the contributions of this thesis mainly around the following subjects:

1. **Digital map and GIS information:**

During our research work we noticed the importance of creating the digital map. For example, in transportation planning and logistic, it is advantageous to use digital map or numerical incorporated with information such as transportation facilities data, speed limits, roadway indicator, and type of roads. These information are very helpful to improve traffic algorithms and estimations. Thus, we extracted our digital map from OpenStreetMap (OSM) and we added our GIS information that contains all the information needed such as speed limit at each section, intersection, defining the type of roads and its directions, new features, etc.

2. **Map-matching problem**

In our approach we defined a spatial analysis criteria and we defined the area of interest by detecting the road sections that has the highest probability, where the GPS position should be matched on the digital map, without scanning the whole

map. Then, we added temporal analysis criteria. Finally, we applied an orthogonal projection on the GPS position into the concerned road section that have the highest score of the combination of spatial and temporal criteria. Moreover, we added the notion of the orientation in order to enhance the map matching. The orientation informs us about the direction that the car is following; making it easier to be located on the road that has the same direction as the car's heading. In some of the case we still have non-matching cases. Therefore, we added a correction method that makes a new prediction of the defective data and then apply the whole process of map matching to the corrected data.

3. Time dependent shortest path problem

The time-dependent shortest path problem given a departure and arrival time was one of our research interests. There are many solutions that has been developed for different types of graphs such as Dijkstra in the case of positive weights, Bellman in the general case, etc. However, these methods are time consuming regarding the computational process, which mean that they need enhancement and speed up. We adopted a new approach by applying the process in a database context. In this method we introduced a new step that we called a learning phase, where we process on offline mode using a recursive Dijkstra. The results of this learning process are saved in the database. Then, during the online mode the method uses the output of the learning step that is called by the map matching. Introducing this learning step at the level of the database enhanced the speed of finding the answers on online mode.

4. Travel time estimation per road sections

The use of sequential Monte Carlo approach to estimate travel time using sparsely sampled GPS data is certainly not new. However, to our knowledge the application of Monte Carlo methods on urban networks has not been found in the literature.

a. Travel time estimation using Monte Carlo method

Using this approach we adopted the Monte Carlo method to our case by defining a state equation. The creation of the filter was done in a way to process and use the sparsely sampled GPS data. The filter gave us the ability to estimate the moment when the probe vehicle enters a road section and also when it exits the same road section. By considering these information it was easy for us to know how long the probe vehicle was in that section.

b. Travel time estimation using Monte Carlo method enhanced with measurements and road sections characteristics

To enhance the approach, we injected the algorithm with additional information concerning the travel time distribution of each road sections. This information concerned the measurements and the features of the road sections on the road network.

Résumé :

Cette thèse porte sur le problème de l'estimation des temps de parcours, de véhicules, par section de route dans un contexte urbain, en utilisant les données GPS à faible densité d'échantillon. L'un des défis de cette thèse est d'utiliser ce genre de données. Par ailleurs, les travaux de cette thèse et le rapport ont été fait dans une période de deux ans et demi.

La thèse s'inscrit dans le cadre du projet PUMAS (Plateforme Urbaine de Mobilité Avancée et Soutenable), ce qui est un avantage pour nos recherches en ce qui concerne le processus de collecte de données réelles sur le terrain ainsi que pour faire nos tests. Le projet PUMAS est un projet préindustriel qui a pour objectif d'informer sur la situation du trafic mais également de développer et de mettre en œuvre une plate-forme de mobilité durable afin de l'évaluer dans la région, notamment à Rouen, France. Le résultat offre un cadre pour tout contrôleur de la circulation, gestionnaire ou chercheur pour accéder à de vastes réserves de données sur l'estimation du flux du trafic, sur les prévisions et sur l'état du trafic.

Les données GPS échantillonnées, réfèrent au cas où les véhicules envoient leurs positions courantes, à une fréquence fixe. L'aspect de la fréquence n'est pas suffisant pour mesurer directement la vitesse ou le temps de déplacement sur les sections de route.
Afin de surmonter ces défis les contributions de cette thèse s'articulent essentiellement autour des thèmes suivants:

1. **Carte numérique et information SIG:**

Au cours de notre travail de recherche, nous avons été convaincu de l'importance de la création de la carte numérique. Par exemple, dans la planification des transports et de la logistique, il est avantageux d'utiliser la carte numérique intégrante des informations telles que les données des installations de transport, les limites de vitesse, indicatrice de la chaussée et le type de routes. Ces informations ont été très utiles pour améliorer les algorithmes de la circulation et des estimations. Ainsi, nous avons extrait de OpenStreetMap (OSM) notre carte numérique et nous avons ajouté nos informations SIG qui contiennent toutes les informations nécessaires telles que la vitesse limite pour chaque section, les intersections, la définition du type de routes et de ses orientations, etc.

2. **Géo-référencement**

Dans notre approche, nous avons défini un critère d'analyse spatiale et nous avons défini la zone d'intérêt en détectant les tronçons de route qui ont la plus forte probabilité, où la position GPS doit être mappé sur la carte, sans scanner celle-ci entièrement. Ensuite, nous avons ajouté des critères d'analyse temporelle. Et enfin, nous avons appliqué une projection orthogonale des positions GPS, qui ont le score le plus élevé de la combinaison des critères spatiaux et temporels, sur le tronçon de route concerné. De plus, nous avons ajouté la notion du cap en vue d'améliorer l'appariement sur la carte. Le cap nous renseigne sur la direction que la voiture a

suivie; ce qui rend sa localisation sur la route facile vis-à-vis du sens de la circulation sur la section de route concernée. Suite à des rejets de données ou des mauvaises correspondances constatées, nous avons ajouté une méthode de correction qui fait une nouvelle prédiction des données défectueuses, avant d'appliquer l'ensemble du processus du géo-référencement aux données corrigées.

3. Problème du plus court chemin dépendant du temps

Le problème du plus court chemin dépendant du temps connaissant un départ et une arrivée a été l'un de nos sujets de recherche. Il existe de nombreuses solutions qui ont été développées pour les différents types de graphiques tels que Dijkstra dans le cas des poids positifs, Bellman dans le cas général, etc. Toutefois, ces méthodes prennent beaucoup de temps en ce qui concerne le processus de calcul, ce qui nous a amené à les améliorer. Nous avons, pour cela, adopté une nouvelle approche en appliquant le processus dans un contexte de base de données. Cette méthode, nous a permis d'introduire une nouvelle étape, que nous avons nommée phase d'apprentissage, où nous avons procédé, en mode hors ligne, en utilisant le Dijkstra récursive. Les résultats de la phase d'apprentissage sont enregistrés dans la base de données. Puis, pendant le mode en ligne, la méthode utilise la sortie de la phase d'apprentissage qui est appelée par le géo-référencement. L'introduction de cette phase d'apprentissage au niveau de la base de données améliore le temps de réponse en mode en ligne de façon significative.

4. Estimation du temps de parcours par section de route

L'utilisation de l'approche de Monte Carlo séquentielle pour estimer le temps de déplacement à l'aide des données GPS, à faible densité d'échantillon, n'est certes pas nouvelle. Toutefois, à notre connaissance, l'application de méthodes de Monte Carlo pour estimer les temps de parcours sur les réseaux urbains ne figure pas dans la littérature.

a. Estimation du temps de parcours en utilisant la méthode Monte Carlo

En utilisant cette approche, nous avons adopté la méthode de Monte Carlo pour notre cas en définissant une équation d'état. La création du filtre a été faite de manière à traiter et à utiliser les données GPS à faible densité d'échantillon. Le filtre nous a donné la capacité d'estimer le moment où le véhicule traceur pénètre dans une section de route et aussi quand il en sort. En tenant compte de ces informations, il était aisé de savoir combien de temps le véhicule traceur a mis pour traverser la section de route en question.

b. Estimation du temps de parcours en utilisant la méthode Monte Carlo améliorée avec les mesures et les caractéristiques des sections de route.

Pour améliorer l'approche, nous avons modifié l'algorithme en injectant des informations historiques relatives à la distribution des temps de parcours de chaque section. Ces informations concernent les mesures et les caractéristiques des sections de route sur le réseau routier.

This dissertation is dedicated to my awesome and amazing family, to whom I owe everything and my lovely fiancé, Pille.
I love you all

Acknowledgments

There are many people I would like to thank for all of their support in making this dissertation possible. First and foremost, I want to thank my parents, Ahmed and Rabha Hadachi, my brother Izgh, my sister Lacey and her husband Dennis for being supportive of absolutely everything I have done in my life and always encouraging me to pursue every opportunity. A special thanks to my second family, Roman and Tiia Ubakivi, and also Piret and Kalle Soovares for their support during this thesis period.

I wish to extend my heartfelt thanks to Professor Abdelaziz Bensrhair my thesis director for the kind assistance and guidance he has provided me throughout my academic life at LITIS and INSA of Rouen. He always pushed me to dig in more in the subject and understand the logic underneath the mathematical solution methods. Similarly, I would like to express my gratitude to my supervisors Dr Stéphane Mousset and Dr Christéle Lecomte for their kind support, patience, and guidance over the thesis period. I would like also to acknowledge the other members of the PUMAS project consortium, Mr David Chevin, Miss Anne-Charlotte Nicoud, Mr Pérot Cédric, Mr Jean-Marc Morin, Mr Jean-Marc Lasgouttes, and more others for their assistance and encouragement.

I warmly thank Mr Fawzi Nashashibi and Mr Fabrice Mériaudeau for accepting to be rapporteur in my thesis as well as Mr Jacques Jacot and Mr Alberto Broggi to be the examiners for this work and also a special thanks to Mr Bernard Matyjasik for accepting to be part of the jury for this thesis.

I thank all my interns that I worked with during this thesis period, Gabriel Wiart, Tahri Jouti, Ndèye Rokhaya Kane Diallo, and also all the teams of PIC ASI that I shared good moments with them working on the project PUMAS, Thibaut Yoquet, Charlie Boulo, Mathieu Fresquet, Thibaut Reiter, Ali Lazaar, Alexis Attimont, Youssef Hafi, Paul Jégouic, Sébastien NG Chun Hing, Benjamin, Hamza Errougani, Françoise Juvanon Du Vachat, Grégory Coutant, Benoit Boucher, Kok Choong Wong, Vincent Cassé, Mirela Rasinar, Jean Creusefond, Matthew, Perrote Jérémy Risso-Bourgés.

I would like to thank Mrs Samia Ainouz, Mrs Alexandrina Rogozan, Mrs Elsa Planterose, Mr. Stéphane Canu, Mr Sébastien Kramm, Mr Mhamed Itmi, Mr Pascal Vasseur, Mr Bruno Sadeg, Mr Claude Duvallet for their listening and advises when I am bordering them with some of my questions. I also want to thank all the teachers of the department STPI INSA of Rouen for their support and advice during the teaching and coaching period, Particularly Mr. Jean-Pierre Pecuchet and Laurent Vercounter.

Big thanks to the administrative and technical staff of LITIS, Especially Sandra Hagues, Brigitte Diarra, and Jean-Francois Brulard for their help, availability, and their good humor throughout this years.

A special thanks with no goodbye, just good memories for all my friends and mates at the lab, Yadu Prabhakar, Alina Dana Miron, Laura Giuliani, Nadeen Salameh, Carlo Abi Chahine, Zakariae Bram, Bassem Besbes, Yacine Sid Ahmed, Pierre Yver, Georges Challita, Iyadh Cabani, Florian Yger, Aurélie Boisbunon, Ovidiu Serban, Guillaume Dubuisson Duplessis, Zacharie Ales, Abbu Keita, Abir Zribi, Pierre Bourgeois, and Yacine sid ahmed.

Finally, I express my deepest gratitude to my parents for their sacrifice and their unconditional support despite the distance separating.

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 General view and Background

The urban road traffic today experiences increased rates of traffic jams that threaten the environment, the people's behavior and also the transport efficiency. In order to tackle these problems, knowledge about the traffic conditions is a must at many levels of urban traffic management and policy.

The involvement of traffic management can be seen in variations of factors, such as network structure, public transportation, city size, etc. Urban traffic management efficiency is relaying on the capability of individuals to plan their own trips more accurately. From this statement it is clear that travel time information plays a big role in urban traffic management. Moreover, it reflects the performance of urban road networks. Besides, travel time estimations or predictions, if accurate and reliable, can be beneficial to network users. The impact can be seen in the decreasing number and level of both the traffic jams phenomenon and the users stress.

Travel time information is essential for transportation planning, transportation operation, and of course, transportation management. In addition, it is a necessity to characterize urban traffic or traffic in general.

Travel time appears to be the most significant measure of road traffic information. Over the last decade, many researchers have conducted travel time estimation [14], [15], [16] and demonstrated its importance in practical applications of transportation and logistics [17], [18]. The use of travel time estimation enhances efforts in many fields to give road users the information needed for understanding the road traffic status [19].

In practice, the methodologies applied to estimate or predict travel time depends on the data available. Furthermore, the approaches that will be adopted differ in terms of the provided data from sensors available on the urban network. We have to take into consideration that the urban networks are not completely covered with sensors as much as the freeways. As a consequence, it will be a challenge to develop a model or a method that will work in all scenarios.

In order to make the real-time traffic monitoring a success, we need information about the networks state in the past, present and also the future. The information required is collected from a variety of sources, such as inductive loops, cameras and floating cars. Besides, this information should be as complete and accurate as possible.

Unfortunately, all data collected suffer some specific problems. For example, all local sensors detection techniques do not provide an actual traffic state in terms of queue lengths and delays due to traffic jam or traffic lights. Furthermore, the travel time collected from probes [10] showed high variances, which means that they need a preprocessing before use.

Another conceivable source of data is the Global Positioning System (GPS). Nowadays, the information we can get from GPS is so rich for exploitation. This latter can be used through floating car in the urban network. The data collected can give us information about the network status, which in turn implies traffic status.

Throughout time, the literature on travel time estimation using GPS sensors has grown thanks to the technology that become more available. Most of the research, such as [109] and [110] conducts high frequency data, which eliminate many of the challenges of interest due to the progress done. However, low sampling frequency continues to be a challenge because of the difficulties presented by this kind of data. Some of the difficulties faced when dealing with sparsely sampled data with low frequency is the inference of the probes path between two positions reports, which may involve in some case a considerable number of road section in the urban network, like it was stated in [111] and [112]. Another point regarding the difficulties is to estimate the travel time spent on each individual road section. A variety of local methods have been developed for this issue like in [113], [114], and [115].

By taking into consideration the idea proposed by Gustafsson [5] it is important to start by preprocessing the data before trying to estimate the travel time in order to reduce errors caused by the GPS [7] or the Map-matching [9]. These preprocessing steps are very important, especially when dealing with the sparsely sampled GPS data with low frequency.

In this chapter the background is described in order to give an idea how this research work took its path regarding the available data and tools. Next, the research objectives and restrictions are defined in order to make a clear statement of thesis context and boundaries of the research. Then, our strategy and contribution regarding the problematic are summarized. Finally, a clear outline of the dissertation of this thesis is given.

1.2 Research Objectives and Restrictions

This section is going to be elaborated in a manner to establish a clear description of the travel time estimation problematic. Travel time is the outcome of traffic flow processes, which brings up the question of the governance. In this case, travel time is ruled by interactions between traffic supply characteristics and traffic demands. Moreover, the non-linearity interactions between the heterogeneous groups (vehicle, drivers, road) bring to life a complex problem, due to their behavioral characteristics. Nevertheless, in real world field we have enough information to estimate travel time with a certitude that is close to the reality.

1.2.1 Objectives

The main objective of this dissertation is to develop a methodology capable of estimating travel time in a specific platform, using a database of sparsely sampled GPS data. To reach our main objective, we are obliged to deal with other issues related to the preprocessing of the sparsely sampled raw data. We named these in our dissertation the preprocessing tools. The preprocessing tools contain the process of building our digital map and its geographic information system (GIS), filtering the raw data, map matching and shortest path problem.

1.2.2 Scopes

Travel time estimation is a broad research topic. In literature, a wide variety of techniques were used in order to estimate travel time. Those methods can be classified in terms of special scope, road type, input traffic data, etc. In this dissertation the research will be limited in the following manner:

First, the work done applies to an urban road network. This means that the spatial scope is limited to urban road type in an urban network context. Of course the concept can be easily extended to other kind of networks.

Although there are a lot of factors influencing urban travel time, such as the weather, traffic composition and public transit. In this dissertation we will focus on the road network geometry that will be used in the process of the map matching and correcting the raw data errors.

The raw data that is used in this dissertation is the sparsely sampled GPS data with low frequency. This kind of data is massively available, that's why the researchers are using it a lot in their application and research. However, the sparsely sampled data present a big challenge when it comes to its use, especially when estimating travel time per road sections (details of the challenges chapter 2).

Finally, this research focuses on travel time estimation per road section in urban areas. The estimation will be run on a historical database of sparsely sampled GPS data on offline mode and also on online mode, by simulating the data. In general, travel time estimation is used for forecasting but also for real time route guidance. Apparently, longer the estimation accuracy of traffic either in the future or present is, the more the models rely on statistical and theoretical assumptions regarding the data size and future traffic conditions [116]. In fact, the estimation of travel time plays a big role in learning about the traffic aspects and its behaviors.

1.3 Project PUMAS

We had a chance to conduct this research within the project PUMAS, which helped us in the collection process of our data from the real world field and also in conducting our tests. The project PUMAS [117] (Plateforme Urbaine de Mobilite Avancee et Soutenable / Sustainable and Advanced Mobility and Urban Platform) is a preindustrial project that has the objective to inform us about the traffic situation, to evaluate gas emission and also to develop and implement a platform for sustainable mobility in order to evaluate it in the region, specifically in Rouen, France.

1.3.1 Description

The PUMAS project aims at developing a software platform for collection and analysis of road traffic information in real time. The software platform will benefit to local Public Authorities (towns, cities, regions), providing them with:

- Real time knowledge of current and future traffic conditions along urban and peripheral networks in terms of travel times and traffic patterns,
- A significant contribution to greenhouse and pollutant gas emission models, in real time and throughout the area, based on the knowledge of flow mean speed along all the city road sections
- An accurate tool for urban mobility decision-making through the permanent day-to-day monitoring of traffic conditions in the area.

The project is based on the floating car data concept, by equipping a significant sample of vehicles operating constantly in a given area (public buses, for example) with an on-board device with GSM and mapping capabilities, thus capturing their speed, journey time and location. This data is transmitted via GPRS or WiFi (Figure 1.1) to a platform for analysis. Real-time and predictive travel times are produced using enhanced algorithms.

A thousand of vehicles were fitted with a custom-made on-board unit in order to do travel time estimation, prediction and path reconstruction in the urban agglomeration of Rouen (Normandy). More details about the project will be discussed in chapter 2.

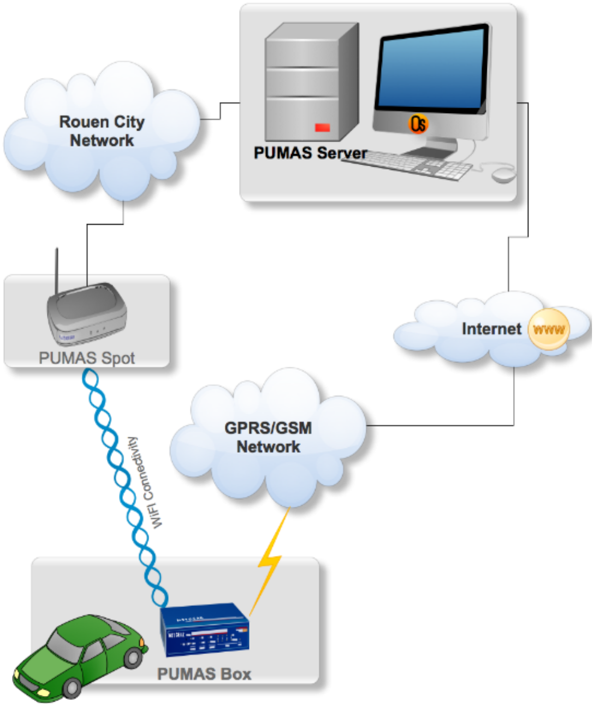


Figure 1.1: V2I Communication used in the project PUMAS

1.3.2 PUMAS Expectations

The results awaited from the PUMAS project are the following:

• Demonstrate the feasibility and added value of mixing conventional traffic data with probe data to generate travel times in a city context

• Compare the floating cars data source approaches of dedicated fleet versus general public floating cars data.

• Evaluate traffic based versus statistical algorithms and identify their best performance domains with a possibility of fusion

1.4 Contributions and Relevance

In this section we will show the main contributions of this dissertation and also the scientific relevance of the approach and the methods used.

1.4.1 Contributions

The main contributions of this thesis can be summarized as follows:

a. The estimation of travel time per road sections was done on urban road network, using sparsely sampled GPS data. Moreover, the estimation was based on the family of sequential Monte Carlo Method, more specifically on particle filter.

b. We made a study about the speed profile, where we conducted the characteristics of the speed distribution in the urban road network. The speed distribution study focused on the impact of the speed aspect on road sections.

c. We also implemented the same study for the location of sparse probes in order to know the evolution of the displacement of the vehicle in the urban road sections. The two studies concerning the location and the speed were done because we will use them in the estimation of travel time for improvement purposes.

d. Finally, this dissertation also dealt with the shortest path problem that we used in the process of map matching for raw data. Besides, we made an enhancement of the map matching process. In addition, the thesis work spawned software with friendly graphical user interface, where we can process and run the system on offline mode or online mode. Moreover, we can see the results of the process directly on the road network map.

1.4.2 Scientific Relevance

The aim of this work is to estimate travel time on urban road sections. The concerned roads are the ones with no dedicated sensing infrastructure and for which the only potentially available source of data is sparse GPS probe data. The challenge relies on the nature of the available data.

For this reason, first we proposed an enhanced map matching technique applied to sparse GPS data by using probe vehicle heading and road orientations, itinerary reconstruction without any external information, spatio-temporal analysis, and correction of the rejected data during the process.

Then we presented an original approach to solve this problem by using an applied Monte Carlo Method and historical probabilities distributions of travel time per road sections. The sequential Monte Carlo approach has been used in many research cases. However, the application of Monte Carlo method on urban networks to estimate travel time has not been found in the literature.

The output of this thesis can be very beneficial by using it in a data-driven method or model-based method in order to forecast or monitor in real time the traffic status. Due to the complexity of urban traffic, it is also promising to combine the two methods to get better results.

1.4.3 Practical Relevance

The research results of this dissertation are used in the PUMAS system. This shows that our approach meets the industrially required standards. To be more specific, the practical relevance of this research work is to estimate travel time per road sections, using the sparsely sampled GPS data as accurately as possible and to implement it in the PUMAS system to apply it for real world environment.

1.5 Road Map

Apart from the Chapter 1 where we present the dissertation in general, the following chapters will be as follows (Figure 1.2):

Chapter 2:
The chapter 2 gives a state-of-art of some techniques used to compute the travel time based on the technology and the nature of data collected for that purpose. Then, the chapter talks about the travel time estimation models with all the techniques used. It describes and analyses some similar project to ours. Finally, it gives a description and details about the project PUMAS in order to give a clear idea about the context of this thesis and the added value.

Chapter 3:
In this chapter we give a clear description of the tools used in the preprocessing step. In this step we make a raw data check for any incoherence. Then, we create the digital map embedded with our new characteristics of the geographic information system (GIS). Finally, we show the enhancement done to the map-matching process and the shortest path problem computation.

Chapter 4:
This chapter presents a study of the observed distributions from the empirical data in the historical database of the speed and location. This study is particularly important for the applied Monte Carlo Method used to estimate the travel time per road sections. Then, it presents a clear description of how the method was adapted to our case in order to estimate the travel time per road sections.

Chapter 5:
The chapter 5 presents the implementation's structures and the software processes of the system created in this thesis work. In addition, each section shows the results and analysis of the tests done in order to validate the methods and algorithms proposed in this thesis.

Chapter 6:
The final chapter summarizes the main conclusions of this research work and offers directions for future research.

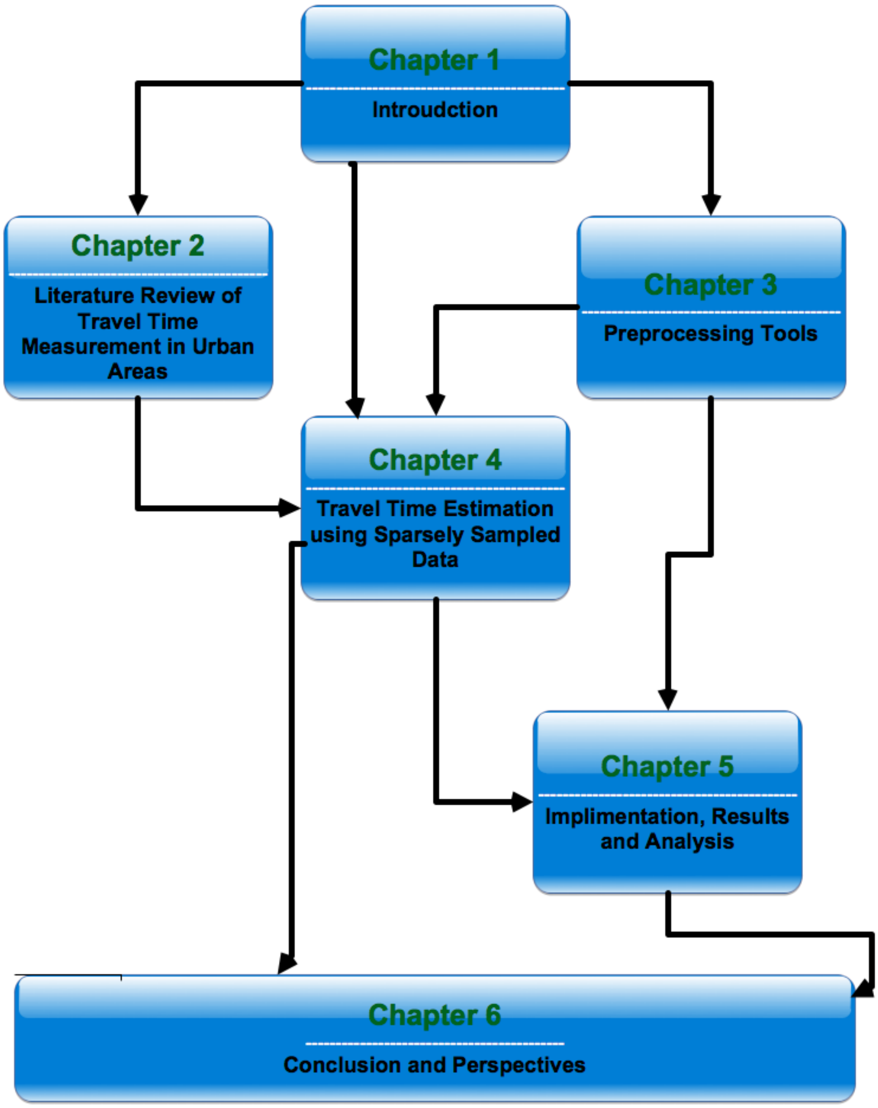


Figure 1.2: Schematic overview of the structure of the dissertation

Chapter 2: Literature Review of Travel Time Measurement in Urban Areas

2.1 Introduction

Since the earliest human settlement, cities and traffic have grown hand-in-hand. The same factors that draw the congregation of large urban areas leaded to intolerable levels of traffic jams on urban road networks.

The urban traffic management has been conducted by researcher through processing specific sources of information such as public transport management and priority, traffic management in urban areas, and real-time traffic light signal management. The traffic management measurement has increased throughout time that leaded the focus of research to change from architectural aspect like effective monitoring of traffic using Variable Message Signs (VMS) [30] to involving extensive data acquisition, enforcement, and control [31] such as pedestrian activity, traffic speed, heavy goods vehicles, cycling, public transport usage, vehicle occupancy and congestion. This change in the approach of research gives more accuracy and effectiveness in the management strategy and policy.

The more we know about the information available on the road network such as Variable Speed Limits (VSL) and Hard Shoulder Running (HSR), and its evolution through time, the more accurate and effective the traffic management will be.

The processing of these extensive data needs a careful selection of the appropriate speeds and operational lane strategies. However, the fluctuating aspect of their patterns makes it a real challenge to control and master the system effectiveness.

The urban context makes choosing the appropriate data an important step in the traffic management. The complexity of urban traffic relies on its actors (personal vehicles, public transportation, etc) that can constitute a source of conflict. Thus, priority system as Automatic Vehicle Identification and localization (AVI/AVL) is a must in order to enhance the traffic management strategies.

Of course there exists some systems like the Automatic Incident Detection (AID). This latter uses probe vehicle data, which are one of the interest of the Intelligent Transportation Systems (ITS) community research [32].

One important piece of metric information that can help in investigating traffic status is travel time. Many techniques have been developed in order to gather information about traffic flow, speed, and travel time including: probe vehicles [4], loop detectors, mobile sensors [6], and so on. Different methods based on prediction models and historical data have been explored by researchers to estimate travel time [8].

2.2 Travel Time Estimation Using Traffic Information Systems

Most of the studies related to travel time estimation were done based on the fact that the traffic flow is uninterrupted [1], [2]. However in the real world field these studies cannot be directly applicable because traffic flow is highly dynamic. In order to find a solution to this issue in the literature there are many approaches for estimating travel time in urban areas such as data fusion, fuzzy control theory, and microscopic traffic simulation, etc; but all these techniques need costly GPS data. Therefore, the choice of the approach to adopt in order to estimate travel time is related to the nature of data available.

During these last years various types of sensors have been developed to collect various type of data about the traffic. Overall, traffic data can be summed up to velocity data (distance per unit time), flow data (number of vehicles per time units), occupancy data (percentage of time a point on the road is occupied by vehicles), density data (number of vehicles per distance unit), and travel time (time needed to travel between two location). An additional kind of data is the vehicle trajectory data collected from vehicles equipped with GPS devices. The last kind of data gives the advantage to compute directly travel time and also distance velocities.

2.2.1 Loop Detectors

The traffic measurement is mainly based on counting the number of vehicles on the road. One of the technologies used for this purpose is the inductive loop detector that is buried under the road infrastructure to detect the passage of a vehicle. The information is collected locally in housing and conveyed to a central traffic management where they are aggregated.

Moreover, the loop detectors can provide accurate information regarding the velocity data that can be provided by the loops by checking consecutive crossing times. In addition to the flow and occupancy data [20], this leads to deduce the traffic density [21].

Computing technologies for travel time based on magnetic loops are reliable, but they correspond to old technology over fifty years and remain quite expensive because they require civil engineering and maintenance. Thus, they are only present in the economically advanced countries and in big cities.

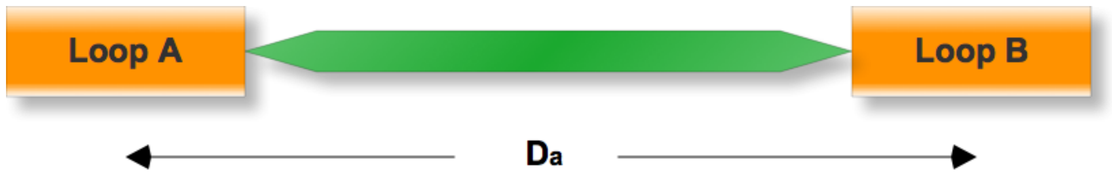


Figure 2.1: Schematic diagram for illustration

Regarding the information collected from the loop detectors many extrapolation methods have been developed to estimate travel time [22], [35].

2.2.1.1 Half-Distance Approach

In this approach the assumption is that the speed measured by a set of dual loop detectors (figure 2.1) is valid to the half-distance on both sides. Therefore, the travel time between the two loops is defined as follows (equation 2.1):

$$T_{a-b} = \frac{1}{2}\left(\frac{D_a}{V_a} + \frac{D_a}{V_b}\right) \quad (2.1)$$

Where,

- V_a and V_b: average speed measured at loop A and B respectively, for a specific time interval.
- T_{a-b}: travel time between loop A and B
- D_a: is the distance separating the two loops

2.2.1.2 Average Speed Approach

In this approach its name reflects the assumption. Which is: the average speed will be the average of the two speed measured by the two loops (equation 2.2). Then the equation will be:

$$T_{a-b} = \frac{D_a}{(V_a+V_b)/2} \quad (2.2)$$

Where,

- V_a and V_b: average speed measured at loop A and B respectively, for a specific time interval.
- T_{a-b}: travel time between loop A and B
- D_a: is the distance separating the two loops

2.2.1.3 Minimum Speed Approach

For this approach the minimum speed detected by the loops will be assumed to be the speed of the vehicle during his travel between the two loops. Hence the equation 2.3 will be:

$$T_{a-b} = \frac{D_a}{V_{min}} \quad (2.3)$$

Where,

- V_{min} : minimum speed measured by loop A and B.
- T_{a-b}: travel time between loop A and B
- D_a: is the distance separating the two loops

2.2.2 Video and License Plate Readers

Travel time can be measured also by using automatic plate recognition system (APR)[11]. The difference between loop detectors and ARP system is that loop detectors can only provide information about the flow and local speed; thus some errors are present in the estimation of travel time; however, the APR system gives more accurate travel time.

The process starts by having at least two fixed ARP systems on the road (figure 2.2). Then when the vehicles pass by the first APR system it will read its plate number. After, when he passes the second APR system it will do the same. Finally, the server will match the plate numbers and their time stamp tags in order to measure the travel time between the two APR systems [12].

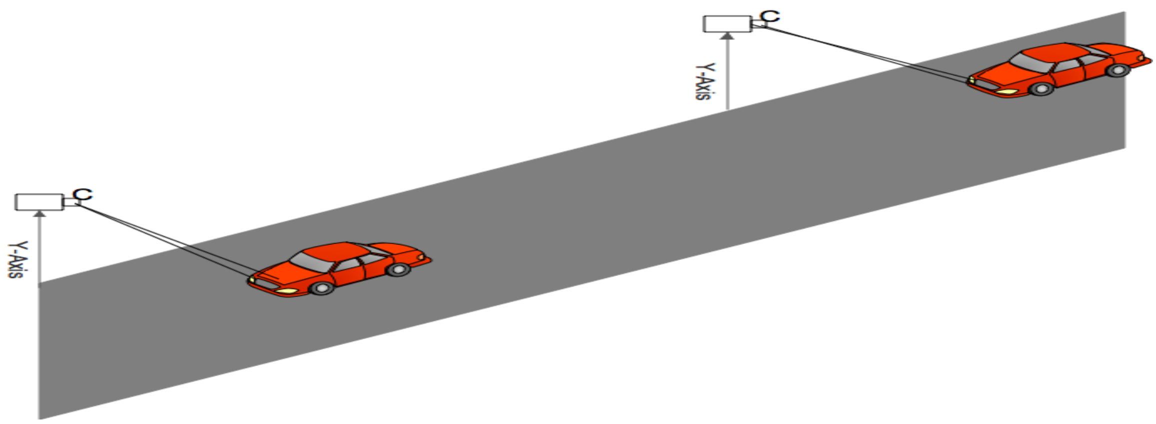


Figure 2.2: Schematic diagram for APR System Illustration

2.2.3 Radar

The radars have the ability to collect velocity, flows, and occupancy data when you place them along the side of roads. There are mainly implemented in highways rather than urban areas. Moreover, they are suitable with massive data collection; nevertheless, the collected data has low accuracy. Moreover, the accuracy data collected decreases in arterial environment.

The radar uses vehicle speed S computed using the time difference ΔT corresponding to the vehicle reaching at the leading edges of two range bins. The distance D separating the range bins is known. The vehicle speed is given by equation 2.4:

$$S = \frac{D}{\Delta T} \quad (2.4)$$

Where,

- D: distance between leading edges of the two range bins and
- ΔT: time difference corresponding to the vehicles arrival at the leading edge of each range bin.

2.2.4 Bluetooth

Bluetooth readers are based probes detection. They scan the area range and check if there is any Bluetooth enabled device; then they can determine travel times and speeds between points on a roadway network [133]. The Bluetooth data give a straight measurement of travel time between pairs of scanners. The data include the "duration" of the vehicle to pass the range detection of the Bluetooth scanner [3].

Thus the Bluetooth can give the following information (Figure 2.3): entry and exit timestamp of the Bluetooth (A) and (B) range. The information collected helps to conclude with the duration of the Bluetooth (A) and (B). Finally the travel time is given by the following equation 2.5:

$$\text{Travel Time} = ETb - ExTa + Db \quad (2.5)$$

Where,

- ETb : Entry Timestamp at Bluetooth range (B)
- ExTa: Exit Timestamp at Bluetooth range (A)
- Db: duration at Bluetooth range (B)

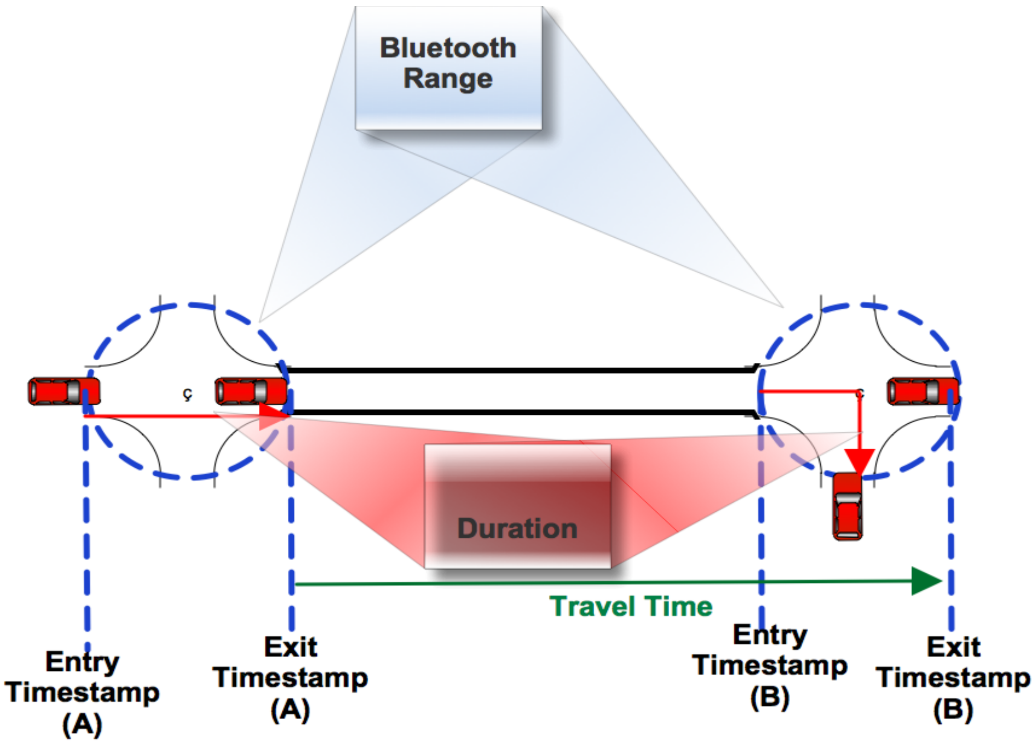


Figure 2.3: Road section travel time computation for Bluetooth approach

2.2.5 WiFi Technology

Getting travel time using WiFi localization alone is accurate enough for route planning but it's not the case for individual road section estimation.
The technique used to estimate travel time is by knowing the location of the vehicle and the distance to the next WiFi spot [134].

The difficulty appears when there is noise affecting the localization of the car making it difficult to attribute the travel time to the concerned road section. However this problem can be fixed partially by adding an algorithm to estimate the nearest neighbor as it was shown in [13]. The author compared two approaches a probabilistic classifiers method and k-nearest neighbor method. In [13] the results showed that the best classification accuracy is possible using a simple brute force approach (K-nearest neighbor).

Another issue with using WiFi localization alone is that sometimes it cannot detect hotspot accurately due to the outages present in the WiFi data. As consequence, they will be missing information of travel time in some road sections.

2.2.6 Sparsely sampled GPS Data

Sparsely probe GPS Data denotes the case when the probes send GPS information at a fixed frequency. Moreover, sometimes it is not frequent enough to measure the velocity evolution or the travel time in an accurate way (frequency can be more than 10 seconds).

The use of this kind of data to estimate travel time presents many challenges. The first one is related to mapping of the GPS location on the road network or the digital map. This implies that the correct position should be found and matched to the right road section as well as the right itinerary of the probe journey. Second challenge is related to the situation when the probe travels many road sections in the road network before sending the GPS data. Thus, the estimation of path is required in order to compute travel time estimation.

All these issues will be discussed in details in this dissertation because it is one of the topics conducted in this research work.

Sparsely sampled GPS data are the most challenging data to process in order to estimate travel time per road section. For example (Figure 2.4) illustrates the sparse data used in [23]. The data is covering the whole area and it is 2-weeks long multi-modal tracks (2 waves – to account for seasonal variation) of 81 users within 2010–2011 [23]. The ubiquity of this kind data is going to make it the most used and available among the traffic information systems next decades.

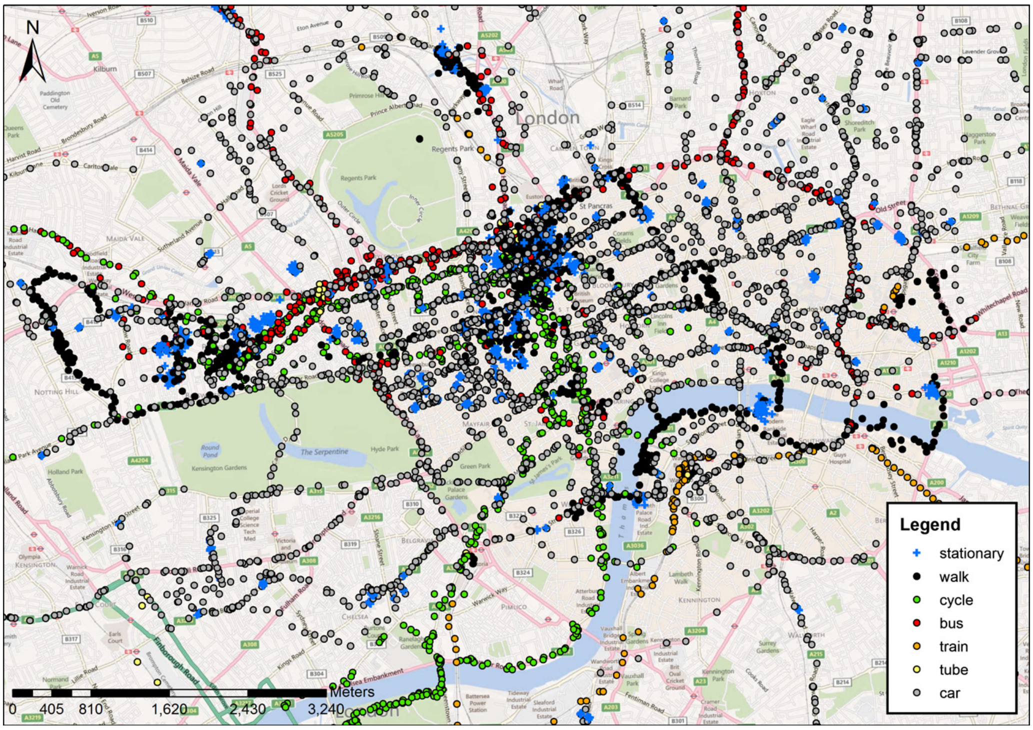


Figure 2.4: The study area and dataset in Greater London [23]

2.2.7 High frequency GPS Data

The high frequency GPS insinuates when the probe vehicle has the ability to send GPS information every few second or each second (no more than 10 seconds). This aspect makes the data the most accurate regarding the itinerary and also the travel time per road section.
It seems that using this kind of technology is the most accurate but still poses some problems with the map matching, especially close to the roundabouts or intersections. This will affect the localization on the map and also computing travel time [135].

Moreover, receiving this kind of data each second through the means of communication into the server is expensive. However, there some potential solution to this issue that Project PUMAS has treated.

2.3 Travel Time Estimation/Prediction Models

Available models can be categorized based on their way of computation to different kinds of statistical models, mathematical models, simulation models, and artificial intelligence models. Thanks to research, there are hybrid models too, where the hybrid models incorporate the combination of two or more the above cited models.
For example the statistical models include Particle Filtering [24], Monte Carlo Method [25], Kalman Filtering [26], Bayes Analysis [27], etc. The models that took advantage of research in the Artificial intelligence are mostly based on

Neuro-Fuzzy models [28] and Artificial Neural Networks [29]. All these cited models have shown good performance in the linear environments such as freeways and highways. Moreover some of them have shown also good results in complex and non-linear environments such as urban networks.
In this section the discussion will be about reviewing a number of these models and methods by type: "statistical, mathematical, and hybrid".

2.3.1 Statistical Approach

Proposed by Gordon et al [33], this kind of algorithm, which uses particle filtering, has gradually emerged as the best technique for processing nonlinear signals. The concept of resampling introduced by Gordon has allowed many issues related to estimations and predictions to be solved, and also opened new windows to be explored and developed. The particle filtering algorithms [34] can estimate probability through successive measurement by using a finite set of Dirac measurements centered in the corresponding points of "particles".

Applying the Monte Carlo Method relies on two things: making the Monte Carlo approximation and resampling size. This method permits us to represent the density of the filter p(:) by samples. In general sampling gives a probability that depends on sampling size. In addition, the method allows us to sample the particles using a law q(:) called the importance law, the size/the number being chosen by the user. The estimation made by this method is valid when the samples are normalized.

In [35] the approach presented was based on an unscented Kalman filter in order to estimate travel time in urban networks. The algorithm used stochastically the vehicles count data from loop detectors and also the travel time data. The process of this method allowed doing real time estimation of travel time and the estimation of upstream of vehicles number as well as the number of mid-road sections.

The Kalman filter is a very good tool to estimate the variables when we are dealing with a linear problem. In case the problem is non-linear then the approach will change either to particle filter or other derived methods from Kalman filter such as extended Kalman filter or particle Kalman.

Let's move now to the Bayesian Analysis. For example in [36] the article shows how they use Bayesian approach to estimate arterial road section travel speed and travel time in urban network. Besides, the input data was collected from loop detectors and probe vehicles equipped with Dedicated Short-range Communication device.

Moreover the Bayesian analysis gives this flexibility to combine with prior information and data. For example you can incorporate a prior distribution for future analysis with past information about the parameter. It provides also conditional inferences on the data without reliance on asymptotic approximation. In addition, it obeys to the likelihood principle and thanks to the probabilities representation in the method that gives a clear interpretation of the parameters.

However, the Bayesian analysis cannot tell you how to select a prior. Besides, the Bayesian inferences need skills to translate subjective prior beliefs into mathematically formulated prior. It can also produce posterior distributions that are heavily influenced by the prior distributions.

To sum up, during this presentation of the statistical methods we showed some advantages and also some disadvantages depending on the nature of the data and also on the objectives of the analysis behind using it. But still can be a good approach if combine between two or more methods and of course by choosing the right one that will fit the problematic or the situation treated.

2.3.2 Artificial Intelligence Approach

Artificial Intelligence is the discipline where we seek to understand the natural intelligence and create intelligent system capable of performing the same actions as the natural ones [41]. Today, applying this technique is kind of fashion among all fields of research, especially the intelligent transportation system sector [42].

One of the most popular Artificial Intelligence technique is the Artificial Neural networks (ANN) and the Neural Network (NN). These two latter techniques have been applied in many fields [43], [44].

Neural network has been explored a lot in research. For example in [29] a novel travel time prediction was developed using artificial neural network with cluster method. The logic of the algorithm is based on functional relation between real time traffic data as input variable and actual travel time data as the output. The use of clustering method is to reduce the data features with less input and preserve the original traffic physiognomies. Then the travel time forecasting is obtained by inserting the real time traffic data into the functional relation. The article results showed good forecasting performances.

Using neural network helped a lot dealing with nonlinear spatio-temporal relationship for example in [37] the author adopted recurrent neural networks in order to forecast freeway travel time. The test was done on synthetic data and the results were pretty good. The inconvenient issue with this approach is that in order to have good results you need detailed traffic information.

However the author tried to fix this problem, which is noticeable in his article in [38] where he suggests a freeway travel time forecasting that exhibits both qualities accuracy and robustness with respect the gap in input data.

To understand why this approach needs a lot of information about traffic in order to give good results; we have to go back to the structure of artificial neural networks. The ANN are constructed by using nodes, which are laid in different layers. The manner in which the layers and nodes are designed and connected to each other is called “topology or architecture”. The frequently used architecture is the “feed-forward” and “recurrent” architecture.

The most used and famous architecture is the feed forward [39] where the network is build with startup input layer and ends with an output layer. Between these two layers there a layer called the hidden layer, where most computations will be done. Besides, in this design unidirectional road sections or connections connect the nodes or neurons.

The second type is the recurrent architecture [40] and it is the same structure as the feed forward the only difference is the kind of connections between the neurons. In this case the road sections are bidirectional or even recurrent providing feedbacks to the neuron itself.

Using this kind of approach can be very helpful dealing with nonlinear complex problems. However, there are some disadvantages for example the ANN can be slow in converging to the final results [136] and this is due to the nonexistence of physical insight in the construction of the mapping approximation of the results parameters. Moreover, using the ANN can be less accurate when the future traffic patterns did not exist in the training samples and also the number of weight in ANN is pretty large and time-consuming regarding the training process (Meldrum, 1995)[139].

It is clear that if we have more information we will have a rich hidden layer, which means that we are improving the quality of the predictions or the results; however, we are making the situation more complicated for the mapping process, which means that the convergence process to the results will take a lot of time to do so.

To conclude, this technique is interesting but needs more adjustments to be done in order to improve it. And of course this opinion depends on our needs and also on the case that we are conducting.

2.3.3 Simulation Approach

Nowadays, traffic simulation models are becoming important tools for the assessment of the traffic infrastructure performance, traffic status, and driver’s behavior analysis. Simulation can help in the validation process and also to predict the behavior of a system regarding different scenarios.

Traffic is categorized as dynamic model. Thanks to this, we can represent the evolution of traffic over time. This representation will incorporate the notion of space and time variations. Besides, the characteristics of the road network are an important element in the creation of the simulation models.

The dynamic models are segmented into two groups: macroscopic and microscopic models. But we will consider also the mesoscopic models, which is a combination of the two other models, cited above. The mesoscopic approach is when a large number of vehicles can interact in a quantum-mechanically correlated fashion.

2.3.3.1 Macroscopic Simulation

The macroscopic model has this ability to generate dynamic characteristics of the traffic flow, where it is considered in its general shape and not by its constituted elements.

The macroscopic models are an analogy between the traffic flow and physics. Traffic is represented by aggregation to the characteristics of fluid mechanics such as density, speed, and flow rate. In this type of models the behaviors of the drivers is considered to be homogeneous. This kind of modeling assumes that the first movement of the vehicles depends on the overall state of traffic.

The macroscopic models in general describe the evolution of time and space. The variables used to illustrate the empirical dynamics of the traffic are listed and determine the change in the flow. In addition, the model normally puts the hypothesis that the traffic evolution can be described by successive steady states.

The most commonly used model to illustrate road traffic is LWF [Lighthill and Whitham, 1955] and [Richards, 1956] models. This model is build based on the analogy with fluids dynamics and it is a first order model. The LWF model is composed of three main equations:

- The velocity equation defined as the ratio of the flow on the concentration.
- The conservation equation. (Coming from the conservation of the number of vehicles on a road section and for a period of time).
- The fundamental diagram postulates that the speed of flow is obtained continuously for a steady state, it does depend only on the instantaneous concentration. This is an equation of state generally separating a fluid portion and a congested portion.

The equations 2.6 are as follows:

$$\begin{cases} Q(x,t) = K(x,t) \times V(x,t) \\ \frac{\partial Q(x,t)}{\partial x} + \frac{\partial K(x,t)}{\partial t} = 0 \\ V(x,t) = V_e(K(x,t)) \end{cases} \qquad (2.6)$$

Where,

- Q(x,t) is the flow corresponding to the number of vehicles flowing in an abscissa x and at a time t per time unit.
- K(x,t) is the concentration corresponding to the number of vehicles per unit of length lying on a section adjacent to the abscissa x, at time t.
- V(x,t) is the flow speed.
- $V_e(K(x,t))$ is the speed obtained from the fundamental diagram.

There is many research work done derived from LWF model. For example in [45] the author presents a dynamic macroscopic model which includes a system that presents a traffic light delay into a stream of vehicles entering a freeway. Moreover, the model is capable of predicting the average vehicle delays and also the queue length estimation. In addition, this latter estimation is compared to theoretical and empirical data.

After this brief introduction about the macroscopic model, we will discuss the advantages and disadvantages of this kind of simulation.

The main advantages of the macroscopic traffic simulation are the representation aspect. This means that the model gives the ability to simulate large road network. Besides, they are designed in order to size or resize infrastructure, obtain efficiency measures about: speed, density, travel time, traffic flow, and set back the travel time between pairs of origin-destinations.

Concerning the disadvantages we can resume them to the fact that the details of any localized traffic is flattened out. This means that it does not make any representation of:

- Different type of vehicles
- Driving style (playing with speed, acceleration, braking…)
- Drivers behavioral variations
- Intersection

To sum up, macroscopic models provide a global view of the traffic but not a focus analysis. The next section will be about the microscopic model.

2.3.3.2 Microscopic Simulation

In the microscopic models flow is not considered as overall homogenous when it is taken in its entirety because the models focus on the kinematics of each vehicle in the system. This means that the model takes into consideration the behavior of each driver in terms of its immediate environment. Moreover, the microscopic models have the ability to make assumptions about trigger actions from individual interactions between vehicles.

Furthermore this kind of model is useful to test algorithms and methods to estimate travel time. For example in [47] the author created an algorithm aimed at estimating travel time on road sections of a road network using a convex optimization framework. Then he used the approach presented in [46] to evaluate the algorithm. The microscopic approach used allowed to accurately reproduce the macroscopic properties of traffic as well as inconsistent driving patterns observed in real life.

Based on the article [46] it is clear that the microscopic approach can help to evaluate the effectiveness of conceivable intelligent transportation system (ITS) strategies under non-recurrent congestion. The ITS strategies conducted in [46] include incident management, adaptive ramp metering, traveler information systems, arterial management, and a combination of those strategies.

The hardest part of the microscopic models is to formalize the specific movement of the vehicles such as lane change, arrivals at intersections. These actions require the calibration of decisional parameters like the influence of the vehicle on other lines, or the reaction time.

The limitations of the microscopic models depend on the complexity of problem that the model is trying to solve. One of the main problems with microscopic traffic simulation is the heavy computation resource when we try to reproduce realistic patterns of traffic.

However this model has some advantages, one of them being the focus on the driver, which means that he is localized at anytime. In addition, the aggregation of individual behaviors leads to the fundamental variables of traffic. This model gives you the ability to introduce heterogeneity in the traffic classes to learn more about vehicles and drivers. Finally, they allow real consideration of the interactions in the development of specific actions.

The actual capacity of the transportation systems is not evaluated from the physical considerations of the infrastructure but from the interaction between the vehicles. This point pushes us to think about the general impact on the traffic flow and on the traffic status. To conclude, it seems that both microscopic and macroscopic approaches are motivating and by combining the two it will give interesting results regarding the flow of the traffic, density, and the travel time. Based on this view the next point will be the mesoscopic models, which are a combination between the two of them [49] (Figure 2.5).

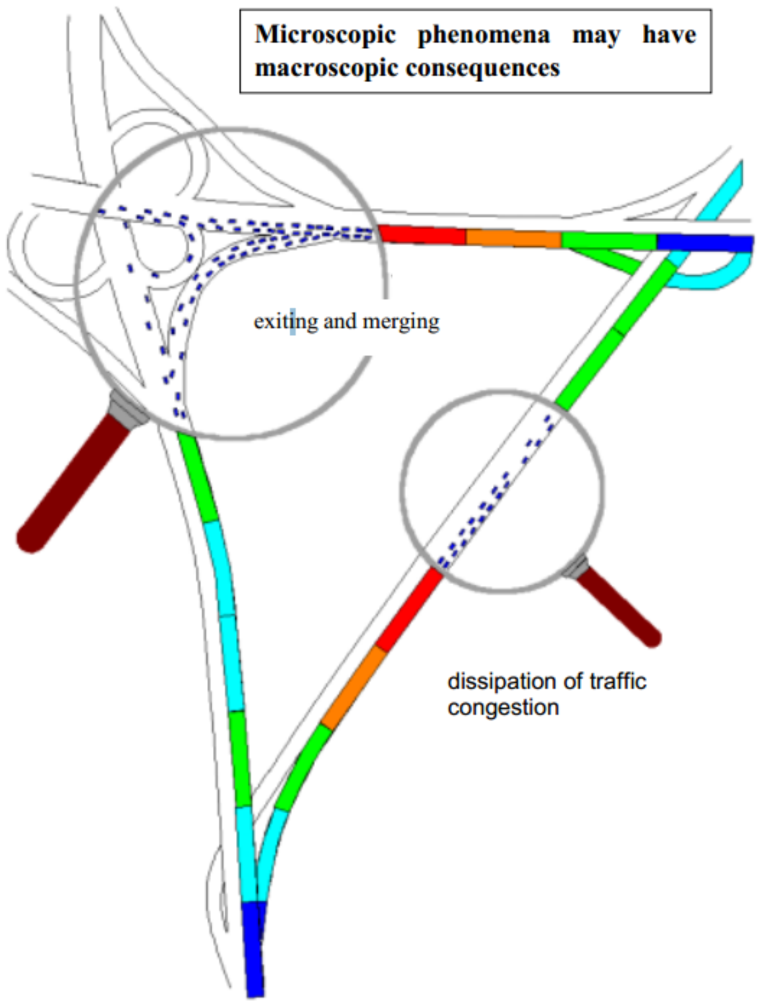


Figure 2.5: Example of scale interactions and their representation by model coupling [49]

The illustration in figure 2.5 shows the characteristics that the mesoscopic models can give us. We can observe the macroscopic aspect of the traffic and we have this ability to zoom in and see the microscopic aspect too.

2.3.3.3 Mesoscopic Simulation

The mesoscopic models comprehend individual agents that refer to real vehicles. Additionally, the simulation is driven by the road characteristics such as junctions. This means that the mesoscopic model fills the gap between the macroscopic and microscopic models. In many researches conducted using this kind of model treated many issues like queuing networks [50], or sustainability aspect of traffic like studying Carbone oxide emission from vehicles and gas kinetic models [48].

Generally underneath mesoscopic models traffic flow details are described at high level. However, at the same time the flow behavior and interactions are shown at a low level too. This aspect has been helping a lot in analyzing the traffic status. For example in [51] the author presents an application of multimodal mesoscopic dynamic traffic assignment model where he analyses the transportation system network under emergency conditions. To conduct this case an extension to a mesosscopic dynamic traffic assignment model was settled to determine quantitative indicators for estimating the exposure of the components regarding the traffic network.

Imagining the movement of vehicles as packages travelling together can represent the mesoscopic operations. Each road section is subject to the following computation (equation 2.7):

$$\tau_i(t) = f(\phi_i(t), C_i(t), P_i) \qquad (2.7)$$

Where,

- $\tau_i(t)$: travel time per road section i.
- $\phi_i(t)$: the flow.
- $C_i(t)$: the occupation.
- P_i: other parameters to conduct.

As it was mentioned before the mesoscopic models are a combination of microscopic and macroscopic models, which means that, they will combine the advantages and disadvantages of these two models. The mesoscopic models consume less time in computation regarding the microscopic models and they give more details than the macroscopic models.

Moreover, it is not easy to design the mesoscopic models without drifting off to one of the models cited before (Macroscopic, Microscopic).

2.4 Similar Projects

In this section we focus on different projects aimed to answer the same question with which we are confronted in our thesis. We chose to discuss in this section the Mobile Millennium project and INRIX Traffic project. The objective of this literature survey is to gather technical elements and research methods, which will give an overview of different technologies available and applied methods. This section contains also a presentation of the PUMAS project, which constitutes a platform for our thesis to test and evaluate our research approaches in real life application for pre-industrial purposes.

2.4.1 Mobile Millennium Project

The Mobile Millennium project is identified in the field of research aimed at developing a system for monitoring traffic in real time and it is based on the GPS integrated in mobile phones to gather traffic information. The project was launched after a public-private partnership agreement involving the University of Berkeley, NAVTEQ (digital map producer) Calstran (The California Department of Transportation), and finally Nokia Research Center. The project was launch on November 2008, although research and data collection have previously started at the University of Berkley. This research project consists of three concrete applications and they are as follows:

- A mobile phone application has been made available to the public for 12 months.
- An experiment size, in a controlled environment called "Millennium Century."
- Finally, Mobile Millennium ensures the confidentiality and security of data.

2.4.1.1 System Architecture

The Mobile Millennium project [52] aimed to gather data from probe vehicles preserving the anonymity of the sender of the information. This architecture as in shown figure 2.6 involves four different entities:

- Probe vehicle (carrying a phone with a GPS)
- Telephone operator (Network provider)
- A proxy server to ensure the anonymity of data
- A reconstruction system traffic

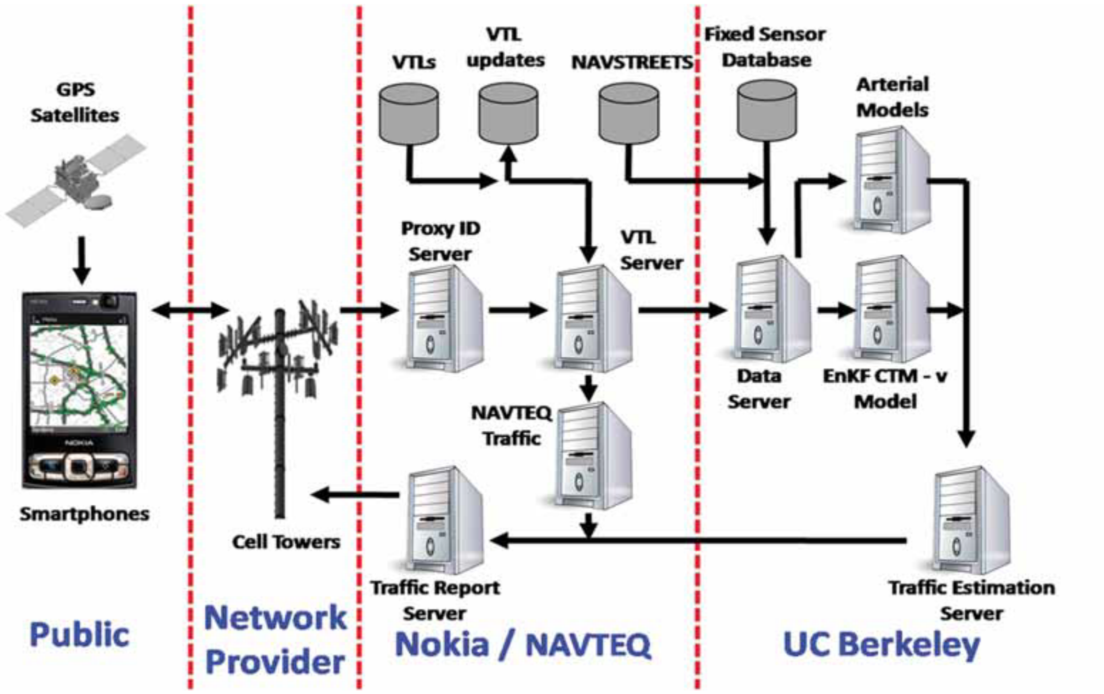


Figure 2.6: Architecture of the Mobile Millennium System [52]

The system architecture is based on the concept of "Virtual Trip Lines" (VTLs). These are a kind of geographical markings stored directly at the client level; in our case it is the mobile phone.

The passage of a vehicle by a VTL launches an update of the velocity and position where it is performed at the proxy server via an HTTP request. This applies to the anonymity of the information; it performs a suppression of the MAC address and the identifier of the array antenna. Then transmit it to the server traffic reconstruction. Not to mention that all data exchanged is encrypted for safety and reliability reasons.

The geographical aspect of VTLs facilitates the data anonymisation, because the only thing needed is the velocity at a point of several vehicles to estimate the traffic in the area; however, if the data is a timestamp kind, it is necessary to possess the identification of the vehicle in order to know its previous location; therefore, the estimation of the speed of the driven section will be possible.

Finally the data collected are used to reconstruct the state of traffic and speeds on different axes using a reconstruction algorithm of traffic flow. The algorithm is based on non-linear flow models capable of reproducing accurately the impact created by an accident or a bottleneck. An algorithm of inverse modeling estimates the integration of these models. This algorithm is based on filters; two filters were tested, the Kalman filter and Newton relaxation "Nudging" [52].

2.4.1.2 Mobile Phone Application

The operation of the application is based on a basic principle, when the application is launched on the free phone user; it sends information (position and velocity) of traffic each time they pass though VTLs, therefore, viewing real-time traffic conditions is visible.

However this application is beyond the original concept of our project and the thesis work since the traffic information obtained by the user from a data fusion of different sources:

- Mobile phones with the application, all taxis in San Francisco (GPS data)
- Fixed speed cameras
- Magnetic loops and historical data.

5000 users have downloaded the application over the period where the software was available. Today, the application is no longer available but the users who have previously downloaded it remains operational.

2.4.1.3 Millennium Century

Century Mobile is a full-scale experiment that took place on 8 February 2008 to study the feasibility of the System. The aim of this experiment is to estimate the traffic based only on data from the GPS integrated in mobile phones (Smartphone). The experiment involved 100 cars equipped with mobile phones Nokia N95 (with GPS) on a portion of highway about 16 km for 8 hours. This stretch of road was chosen because it has a large number of loops and also the presence of bottleneck. The Berkeley has mobilized 165 students to drive these cars between 10h and 18h. The objective was to determine if it is possible to estimate the statue of traffic with just the information sent by the "Smartphone", especially with a low penetration of vehicles.

The VTLs were placed at the same locations as the magnetic loops in order to establish a comparison. These comparisons were made on two portions each one is having a clean penetration rate (3 to 4% for the left side, rarely more than 2% for the right portion). Here is the result in (figure 2.7)[52]:

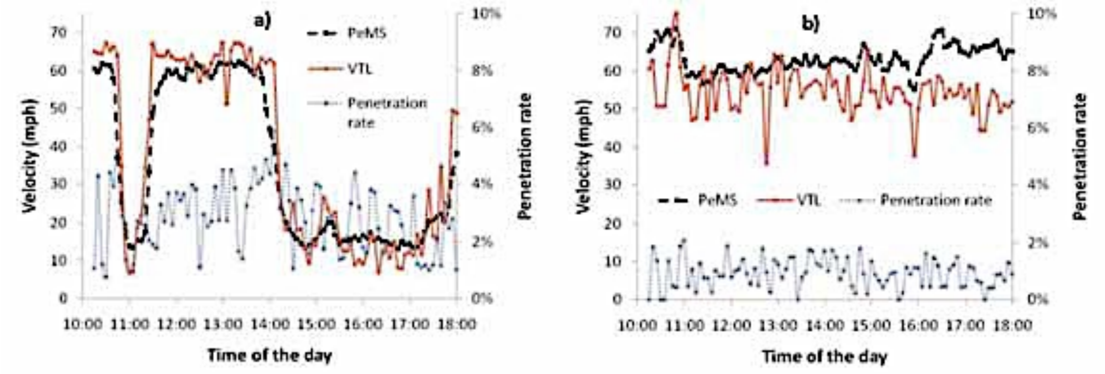


Figure 2.7: Penetration Rate (Magnetic Loops vs. VTLs)

As expected, the results are less accurate when the precision is high. However, it is interesting to note that even with a low penetration rate, information is of high quality and comparable to those obtained through the magnetic loops for a much lower cost. This experiment was therefore successful.

2.4.1.4 Conclusion

The project has achieved its goal of estimating traffic by collecting GPS data from mobile phones, even with low penetration rates while respecting the confidentiality and anonymity of data. However, the project was informing about only the main roads in the city not all the road sections.

2.4.2 INRIX Traffic Project

INRIX is a private company based in Seattle created in July 2004 by two former Microsoft executives. It is a leader in the field of traffic information in the U.S., either historical or real-time prediction. The principle is simple and relatively common to other projects. It develops in three stages [53]:

- The collection of information through various sources (trade name: Smart Dust Network).
- The analysis and treatment with the Fusion Engine.
- The redistribution results to various organizations and devices.

2.4.2.1 Smart Dust Network

A major benefit in INRIX system is the number and diversity of data collected. In fact it consists of more than 350 sources such as GPS traces provided by corporate fleets (taxis ...), existing sensors belonging to public bodies (up to 90% of the existing sensors in United States) such as magnetic loops, data from cell phones, information on accidents and construction cites but also information on the present and future climate or one-time events (concerts, games ...).

Thus, INRIX own announcement dated and past information on average speeds for more than 1600000 km 400000 km in the USA and in Europe through 6 countries (France, Germany, Netherlands, Belgium, Luxembourg and the United Kingdom) and that real-time data representing 126 urban and 260000 km of roads to the United States and 400,000 km of inland different to Europe [54].

2.4.2.2 Fusion Engine

Another strength of INRIX system is their analysis program, processing and data fusion system named Fusion Engine. The Fusion Engine is based on Bayesian network and it uses different proprietary algorithms error correction to deal with data acquired via the Smart Dust Network.

Another feature of the Fusion Engine is the use of error detection algorithms, and it allows also:

- The detection of dysfunctional traffic sensors as magnetic loops that can sometimes take longer to repair. The process is almost in real time, so that data provided by these sensors are not taken into account by the Fusion Engine in its calculations;
- Collaborative filtering: in fact, by combining different sources of information about a geographic area at a given time, INRIX is able to detect the data off sketches or outliers, comparing different information received;
- Optimization of spatial granularity: thanks to the diversity of sources and their number, INRIX is able to determine real-time traffic conditions with good accuracy. Regarding the quantity of data available at a specific time on a geographical area, the Fusion Engine is able to adjust in real time the accuracy or spatial granularity of data to maximize the accuracy of the traffic information;
- Although when the data are insufficient for a given stretch of road, which means when the minimum error threshold is not reached, no information will be given to that concerned road portion.

2.4.2.3 Conclusion

As we have seen, INRIX is positioned on the market at different levels: It provides both solutions directly to the end user (mobile application), data to integrators (manufacturers or manufacturers of GPS) or data for the public sector. Based on this last point, INRIX approaches PUMAS project objectives. However, INRIX and mobile millennium focuses mainly on major routes, roads and highways. Conversely, the project PUMAS is more interested in the entire urban traffic networks.

2.5 The PUMAS Project System

The project PUMAS is a pre-industrial project involving industrials and research laboratories. This variety creates an atmosphere full of different views and helps to create a flexible system design, where it was easy to conduct research and also industrial view. PUMAS stand for urban platform for advanced and sustainable mobility (Plateforme Urbaine de Mobilité Avancée et Soutenable). One of the main contributions of this dissertation has been design, implementation, and testing of preprocessing module of the PUMAS project System. This section presents an overview of the project and how our work is related to it. Moreover, the design decisions include using Historical database and real time data provided by our probe vehicles.

2.5.1 Introduction

PUMAS is a project labeled by competitiveness clusters Mov'eo (DAS Mobility Solutions) and Advancity (COS New uses and mobility). It was held at the call for proposals No. 8 Unified Inter-ministerial Fund (FUI). PUMAS is funded by the state (DGCIS - General Directorate for Competitiveness, Industry and Services), Ile de France and Haute-Normandie Region FEDER (European Union).
The project aims to develop mobility platform software, and to evaluate the territory of the Community of Agglomeration of Rouen Elbeuf Austreberthe (CREA).

The project offers urban areas one integrated solution, flexible and economical:

- Knowledge of traffic conditions and real-time decision support;
- The development of sustainable mobility;
- Measuring the emission of Greenhouse Gases.

Nowadays, collecting data about traffic is available using different kind of sensors. However, there is no tool to extrapolate the state of the circulation and the air quality in a dynamic and real-time manner over the whole territory. PUMAS try to give an answer to these needs by innovating ideas:

- Data collection method based on probe vehicle, less expensive and more flexible than traditional data collection by sensors embedded on the ground;
- Dynamic transmission architecture to use the best available data from probe vehicles in different places and times;

- Calculations performing real-time prediction and estimation of travel time;

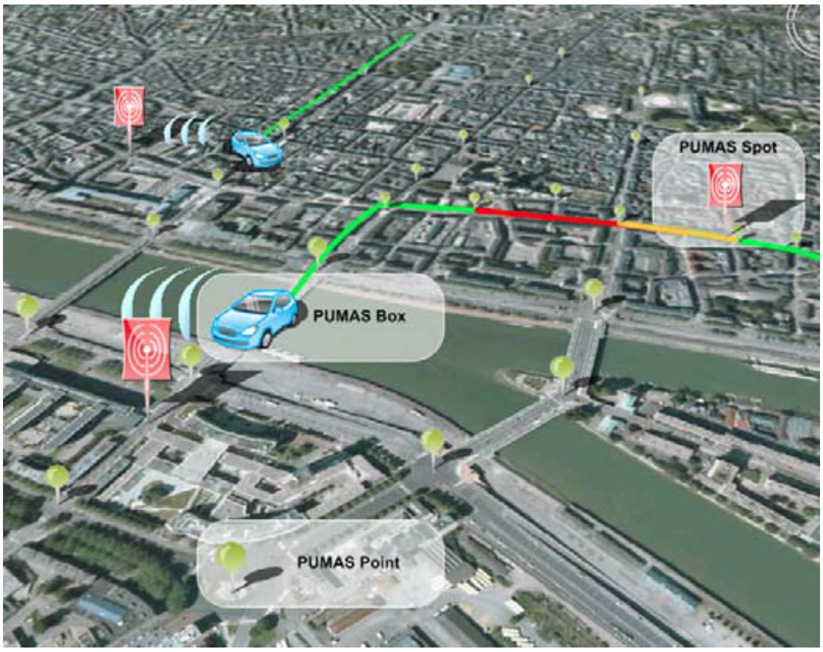


Figure 2.8: View of the PUMAS System Vision

The purpose of the PUMAS project is to create a platform of travel time information to towns and cities.
The main objectives are:

- Create a new generation of software producing journey times, matching the users expectations.
- Participate in the emerging sustainable economy where public transport systems are an alternative to road networks by providing better information on traffic conditions.

2.5.2 System Architecture

To view a system architecture there are many ways but we chose to show the functionality architecture of the system as it is illustrated in (Figure 2.9). There are two components, the server and the embedded box in the cars named PUMAS Box.

The PUMAS Box is capable of collecting information about probe vehicle journey. The Box performance can be resumed as follows:

- The ability to gather data, the GPS position, travel time, traffic jam.
- The ability to temporarily store and process data
- The ability to store reference data (eg. Mapping)
- The ability to communicate data to a server: GPRS or WiFi,
- Power supply (cigarette lighter or installation)

- The ability to read other sensors (plug OBD2, CAN)
- Functionality Automatic update of software

Concerning the server, it can specify tasks such as gathering information sent by PUMAS Box or historical data, filter the data, save the data into the database, process the data in the database.
Moreover, the server is capable of enriching the historical database and process all the information in it.

In order to improve server performance, it was necessary in addition to the treatments mentioned above, the following tools:

- Monitoring of data processing (reporting of treatment)
- Support for the production of maps (OpenStreetMap size)
- Help the placement of PUMAS Points (map + statistics)
- Production of updates for packages shipped (invisible to the user)
- PUMAS Administration Server.

After describing the PUMAS system architecture lets describe the nature of data stored in the databases. This latter is really important to understand it because the work done in this dissertation is related to the data collected from the real world field that constitute the historical database.

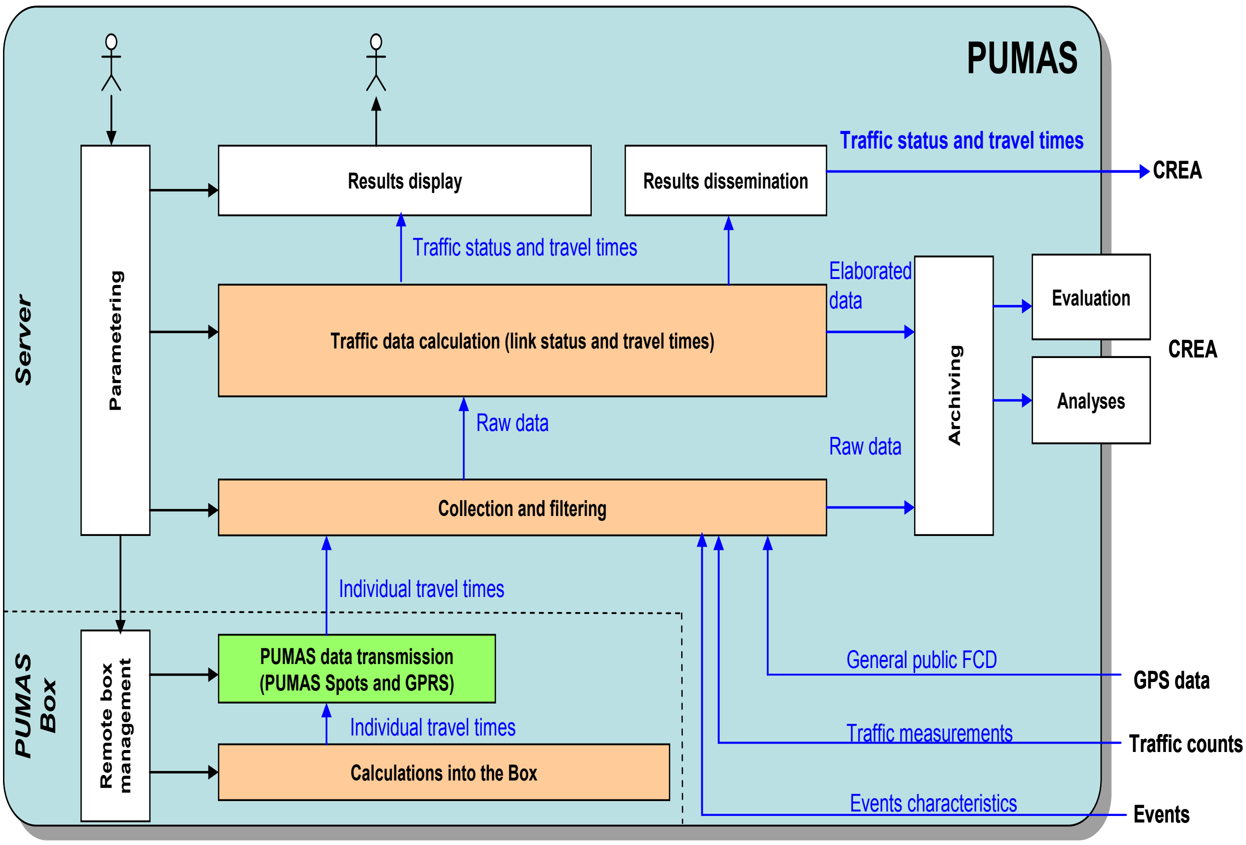


Figure 2.9: An Overview of PUMAS System

2.5.3 Nature of Data used in the Database

The data used in the database can be categorized by two kind PUMAS Box data, and Historical data. The PUMAS Box data contains information about the probe vehicle running in real time on the urban network. The information collected from them during a journey travel is containing: GPS position, Travel time per road section, traffic jam information, etc.

However, the historical data contains sparsely sample GPS information (discussed in section 2.3.6), where the GPS data has a frequency of one minute between each successive input. This means that we have a lack of information between two successive GPS positions.

Our focus in this thesis will be on "Historical data". The idea is to filter the data in order to correct all the errors related to the GPS and the positioning on the digital map. Then, enhance the data with estimated travel time per road sections.

Now, we described a little bit the nature of data in our database that it will play a big role in our work. Lets move to some details about the module in PUMAS system.

2.5.4 System Modules in the Server

The PUMAS server system modules can be divided into five sections: raw data input, preprocessing, processing, server manager, and output. All these components interact with each other in order to process the data and give information about the traffic. Each section in the structure has a specific functionality (see Figure 2.9). All of these sections will be described and discuss their role in the PUMAS server system.

Raw Data Input

The raw data used are received in real time either by WiFi or GPRS. The data is constituted of the PUMAS Box messages (discussed in section 2.5.2) that contain information about our probe vehicles. The second raw data is the sparsely sample GPS information (discussed in section 2.3.6 and 2.5.3), that we have in our historical database, which we receive also in real time and we added it to the old one existing in the database.

Preprocessing

This module does all the preprocessing of raw data. For example the PUMAS Box messages are conjugated and adjusted in such way to fit with the input interface of processing module. This way the processing module has the data in the right way to run on. Concerning the sparsely sample GPS information the module will prepare this data and filter it in order to be enrich with travel time estimation per road section. In order to do so, the system will create the digital map and filter the data from GPS errors and coherence errors. Then, it will reconstruct the itinerary of the probe vehicles. Finally it will run the algorithms to estimate the travel time per road section or what we call PUMAS section.

Processing

The processing module contains two main algorithms that will make the traffic information nicely viewed. The first algorithm is a simulation process of the evolution of travel time estimation based on the belief propagation approach. It was created and developed by our partner in the project INRIA [137]. The second algorithm is regroup all the travel time collected or estimated from the historical data during the preprocessing step to reconstruct a general view of traffic status. This latter was created and developed by our industrial partner in the project EGIS [138]. Then all the information coming from these algorithms will be regrouped in such a way to cover the whole urban network with travel time information per road section and also traffic jam detected in real time.

Server Manager

The server manager module is in charge of displaying the information given by the processing module and display it in a nicely way. Moreover, it process the data used in order to store it in the archive database. This module is in charge of doing any kind of update regarding the functionality of the whole system or regarding the digital map if needs an update. Besides, an external supervisor where he can supervise and check if there is any problem in the server system can control this section. The module does also an analysis of the information collected from the previous section in the chain. This latter process helps to detect the trouble if there is any and also locate the origin of the problem.

Output

The output of the PUMAS system will be a digital map representing the urban road network. In this map they will be information about the traffic, where travel time is specified per road section and also the traffic status (detection of any traffic jam). The output framework gives a real time visualization of the output of the PUMAS system. Moreover, inside the system there are two modules of analysis and evaluation that can be viewed too. This later and output can be used for optimization or research in the future.

After giving a general overview about the PUMAS server system, the next section will be about the data collection process and the experiment done in the real world field.

2.5.5 Data Collection and Field Experiments

The purpose of the experiment is to collect data from the real-world field. During the experiment we used three instrumented vehicles. The experiment allowed us to obtain data from the fields in sufficient quantities to calibrate and analyze the behavior of travel time estimator algorithms.

Firstly, we defined constraints such as jams in small urban road or in the bridge sections that we might face in order to test the performance of the system in tough situations. We chose the itineraries based on those constraints.

Our aim in the first scenario is to test our embedded system in the car's PUMAS Box (Figure 2.10). The PUMAS Box is a system that sends information to the server such as GPS position, traffic jam if detected (discussed in section 2.5.2), etc. Besides, it collects the data that we need for testing the travel time estimation algorithms.

Figure 2.10: The Embedded System PUMAS Box

We had the chance to choose the city of Rouen, France, which has a multitude of different kinds of urban area roadway including bridges in the downtown, expressways, and small roads between buildings. The city is an excellent test-field as it affords a variety of scenarios with different situations regarding the road infrastructure (Figure 2.11).

Moreover, the fact that the itineraries during the experiment were chosen based on the displacement of the traffic observed on the real-world field; will lead us to have a variety of situations. Besides, the data acquisition in the field was based on different modalities such as traffic flow, traffic density, rush hours, etc.

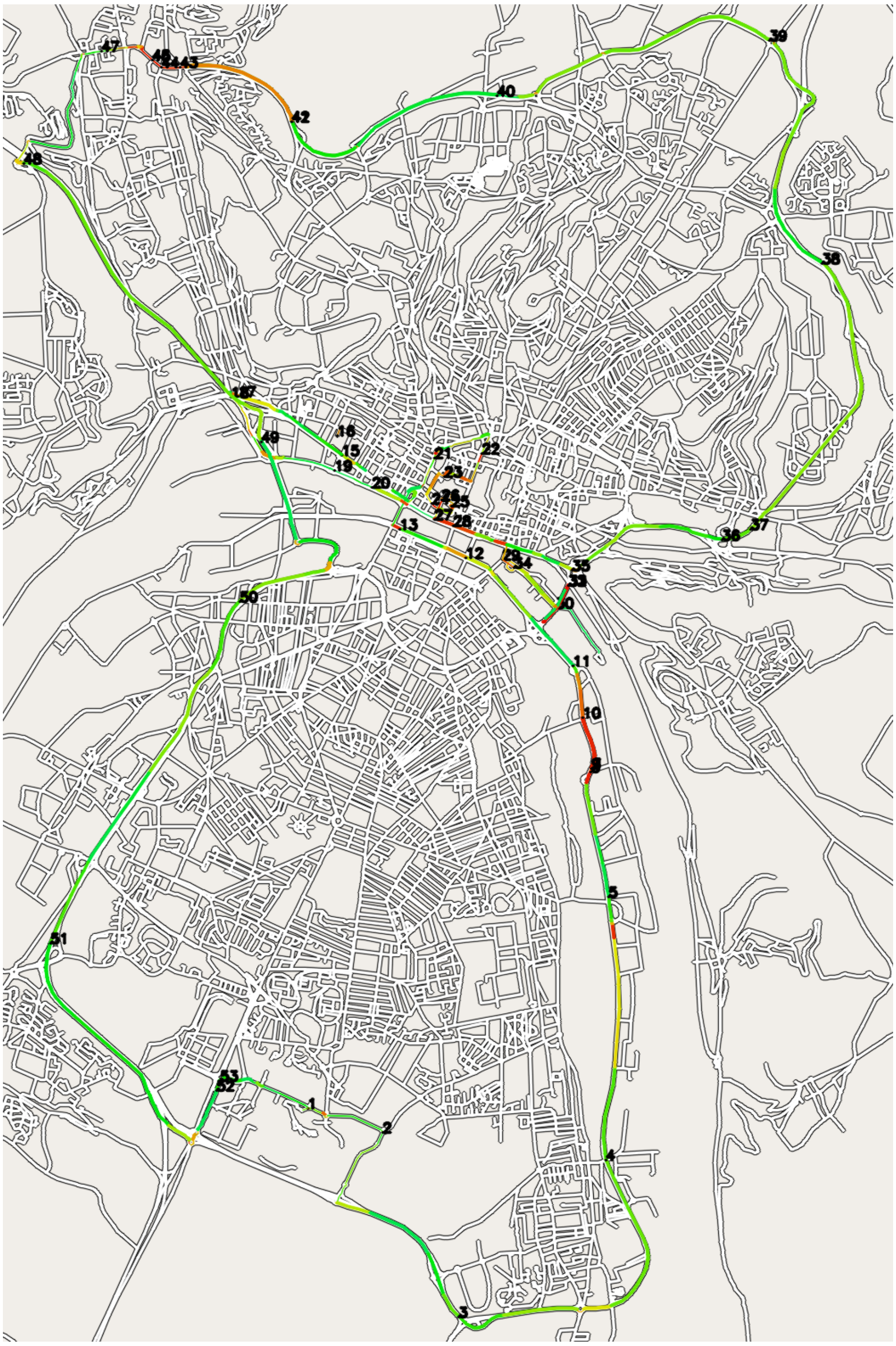


Figure 2.11: Itinerary Sample Reconstructed from The Collected Data

During this experiment the following equipment was used:

- Three Citroen C3 LaRA with their embedded systems consisting of: PC with RT maps (software with a component based framework for rapid development of multi-modal applications), a PUMAS-BOX connected to the PC and communicating with the server, a fixed GPS, a color camera positioned at the windshield, a sparsely sample GPS box transmitter; which we used in order to test our system and collect data in the real-world field. Moreover, C3 LARA is the first prototype of an intelligent vehicle that was developed by the robotics center of the Ecole des Mines de Paris and IMARA team from INRIA, France.

- Modes of transmission-type WiFi and GPRS.
- Server to retrieve data from PUMAS-BOX.

The experiment took place in the city of Rouen, France over three days. The outcome was more than 1400 km driven and almost 4GB of data was collected.
The data collected will be used to simulate our needed data and also to construct a reference data in order to run our tests and also compare our results.

2.6 Conclusion

This section has covered four main parts. First, we presented the methodologies adopted to estimate travel time based on the technologies used to gather traffic information. The second part was about the different kind of models that help to estimate travel time. Then, we presented a state of art about the similar projects that treats the same problematic. Finally, the last part was a description of the PUMAS project, this being important for the understanding of the context of this dissertation and thesis research.

Chapter 3: Preprocessing Tools

3.1 Introduction

There are two types of data used to estimate travel time. Data collected from fixed sensors on the roadside [59] and another one coming from floating cars [57], [58]. The one that interest us regarding our thesis will be the second type. The most common floating car data is collected from delivery vehicles or taxis and this type of vehicles are almost all the time on the freeway, highways, urban roads, or arterials roads. Based on the last idea it is clear that this kind of data coming from this kind of vehicles is convenient for the estimation of travel time in urban areas.

As we know there is nothing perfect, which means the second type has positive and negative attributes. The first thing that will come to mind especially nowadays is the cost of this data. It is really cheap to get them regarding other type of data collection. And this is due to the fact that all the companies collect those data for marketing purpose, business, or logistics optimization, which made those data available in a large scale. However, the data provided from these floating cars are sparsely sample GPS data where the frequencies vary between 1 to 3 minutes.

The reason why is that those companies running the system do not need to know precisely the path of their cars. They need only to know the areas location. Another challenge is that this kind of data presents the need of preprocessing before processing any computation to estimate the travel time.

Building a traffic system to estimate travel time per road section using millions of data coming from floating cars is an algorithmic and computational challenge. In order to make good estimations preprocessing the data is a must; therefore, the error coming from the data is reduced. The preprocessing steps have various subcomponents that need significant research work, development, and integration to be done.

This section presents the preprocessing that has been done in this thesis work to make our sparsely sampled GPS data ready to be used for travel time estimation.

The first step is to represent the urban network by building a road model using geographic information system (GIS). This step is crucial for the travel time estimation using floating cars information both for data collection and also for the results analysis [55], [56].

The second step is the data check or coherence check. Working with sparsely sampled GPS data implies a frequency in the data reception time. Thus the data frequency should be checked for any incoherence in the frequency or ambiguity. The most common and intuitive procedure, and also which is used in this work thesis is to track the number of consecutive GPS points that are less than a specified time distance (1 minute) from each other.

The third phase is the map-matching process. This step is very important to the travel time estimation process. Moreover, the map matching is widely conducted in research applied to transportation. It seems that this field of research on map-matching techniques is still challenging and potentially difficult. During the thesis work the map-matching technique has been used with enhancement in the algorithm to improve the performances.

Finally, the last step in the preprocessing part is the path reconstruction. This step is very important for the travel time estimation per road section. By using sparsely sampled GPS data, we need to define the road section taken by the probes. This process is a graphs problem well known in computer science, operation, and transportation research and it will be used in by the map matching process.

3.2 Urban Network Representation

As it was stated in the section (3.2) this part is about the building of the urban road networks using GIS information. This rubric contains an introduction about the importance of GIS information in constructing road networks. The following section is about road models that constitute information about the road and it is important for estimating traffic status. Besides, the context of the digital map is very important regarding the purpose of use. Finally, the extraction technique used in order to build the digital map using GIS with the information needed. All the work presented here in this dissertation section was implemented for the thesis purpose and also for the PUMAS project.

3.2.1 Introduction

Today, with the technological evolution many domains have been influenced, especially the geographic representation of maps has evolved from paper illustration to digital or numerical [60]. Besides, thanks to the evolution of databases all the geographical information are stored and integrated often by the geographical information system (GIS). The GIS is really useful because they make many tasks easy to do such as spatial planning for cities and land use.

Moreover, in transportation planning and logistic, it is advantageous to use digital map or numerical incorporated with information such as transportation facilities data, speed limits, roadway indicator, and type of roads. All this information is very helpful to improve traffic algorithms and estimations.

There are two kinds of GIS mapping data the vector model and the raster model. The vector approach is designed to store and encode as collection of coordinates. For example in our context the road will be encoded to a linear feature of a set of point coordinates. However, the raster model encodes the road map to a collection of multiple grid cells.

Each of these approaches has advantages and disadvantages. For example the advantages of Vector data can present as follows:

- Data can be represented at its original resolution.
- Graphic output is usually more visually pleasing;
- Accurate geographic location of data is maintained.
- Allows for efficient encoding of topology, and as a result more efficient operations that require topological information such as proximity, network analysis.

However, it has also some negative aspects like:

- The location of each vertex needs to be stored explicitly.
- For effective analysis, vector data must be converted into a topological structure. This is often processing intensive and usually requires extensive data cleaning. As well, topology is static, and any updating or editing of the vector data requires re-building of the topology.
- Algorithms for manipulative and analysis functions are complex and may be processing intensive.
- Continuous data such as elevation data is not effectively represented in vector form. Usually substantial data generalization or interpolation is required for these data layers.
- Spatial analysis and filtering within polygons is impossible

To sum up the analysis of Vector data approach: their coordinates define the map elements such as points, lines and polygons. The structure describing topological relations between different objects provides a faithful graphical representation. The advantages of this system characterized by better management of databases data, and better graphic measures are more accurate than they are. Its main disadvantages are to be able to present a more complex data structure.

After checking the Vector data we will move to the raster data. The advantage can be illustrated as follows:

- The geographic location of each cell is implied by its position in the cell matrix. Accordingly, other than the origin point such as bottom left corner, no geographic coordinates are stored.
- The programming is usually easy thanks to the nature of the data storage technique data analysis.
- The inherent nature of raster maps, for example one-attribute maps, is ideally suited for mathematical modeling and quantitative analysis.

- Discrete data, which facilitates the integration of the two data types.
- Grid-cell systems are very compatible with raster-based output devices like electrostatic plotters, graphic terminals.

Concerning the disadvantages of the raster data are as follows:
- The cell size establishes the data illustration resolution.
- It is difficult to represent linear features depending on the cell resolution. Consequently, network linkages are difficult to establish.
- Processing the data attributes may be heavy to do so if the existing data is large. Moreover, the Raster maps inherently reflect only one attribute or characteristic for an area.
- Since most existing input data is in vector form, thus a conversion vector-to-raster is needed which means extra processing (time consuming).
- Most output maps from grid-cell systems do not conform to high quality cartographic needs.

To summarize the raster model, the area is subdivided to represent a grid of cells. Each cell contains digital information relative to an identifier, a qualitative or quantitative parameter. This is comparable to the term "pixel" (picture element) which is the smallest information unit contained in an image divided into grid. The dimensions of the cell are the resolution space.

After introducing the GIS information, the next sections will discuss how we choose to represent our digital map. Based on the PUMAS project needs and also by conducting research approach we will show how we choose the appropriate approach. This part is a challenge in finding an compromise between industrial needs and research approaches that can adopted in order to be productive in the practical world (real word life situations).

3.2.2 Road Model and Digital Map

The road model in standard case involves information about geometric and topological characteristics of urban roads; all this information follows a hierarchical structure. The road system model describes the characteristics of the streets and roads in a large scale of transportation system.

The road model can be defined in different ways depending on the research case and it is a part of the GIS information. Both of road model and GIS constitute the final results, which is the digital map. Moreover, using a digital map in our case will assure us the following statements:
- Maps ensure consistency and facilitate traffic analysis operations.
- Maps support data collection and can help monitor vehicles and traffic activities.
- Maps make it easier to present, analyze and disseminate traffic estimation results.

In addition, technology evolution has given a considerable push to GIS information. For example nowadays, we have better database software that allows managing a vast amount of information, which is referenced to digital map.

Starting from this point the real world structure contains enough information or objects such as road network, road junctions, road sections, nature, etc. All this information is incubated in the digital map in order to approach a replica of the real world. Moreover, this information is organized in what is called GIS layer. This latter contains information that is not affected by any relation between the objects or layers. The real world represented by GIS information encloses many layers as the (Figure 3.1) shows.

Besides, in our case the need for modeling large areas (urban road network of Rouen) pushed us to choose the digital map, which is the best solution till now. Unluckily, most of these are used for routing purposes; consequently, it does not contain fine-grained information about the junctions and the connections lanes over junctions [62], [61].

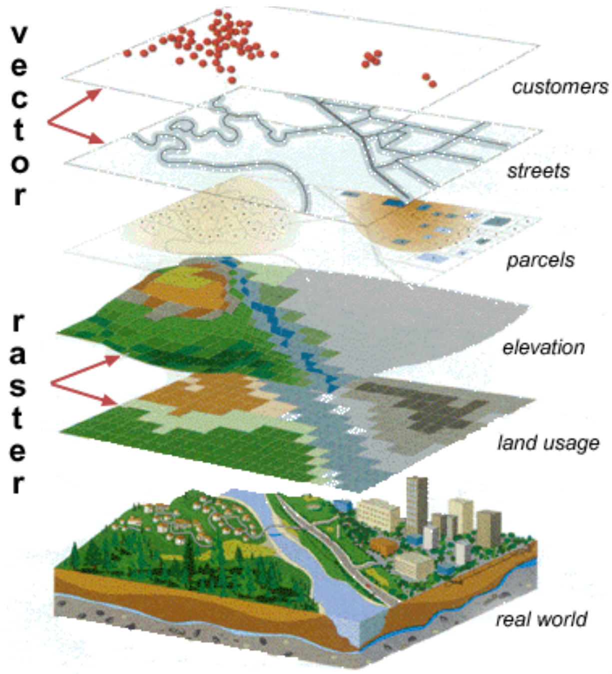


Figure 3.1: GIS Layers

Based on this analysis, it is important to add information to our digital map in order to improve the travel time estimation, map-matching, and path reconstruction. That's why we will add information about the junction and their lane road sections by detecting their locations. The section before introduced the vector data, which contains coordinates defining the road network map elements as points (Coordinates). Based on this, we will choose the vector data approach because it will give the information needed in order to construct the added information to the digital map in an easy and flexible way.

Thus, we will introduce a PUMAS point as the intersection between two different roads, the zone around this junction will be a PUMAS zone, and a PUMAS section is the set of segments and nodes between two PUMAS points (Figure 3.2).

In the previous paragraphs, we described our approach concerning the road model and its implication regarding the digital map. The next level, will be putting enlightenment on the context model. Since, the road model is extended by the knowledge about the context. The following section will analyze the context of our work in this thesis and define what we need as information to add to our road model that will be viewed on our digital map characteristics.

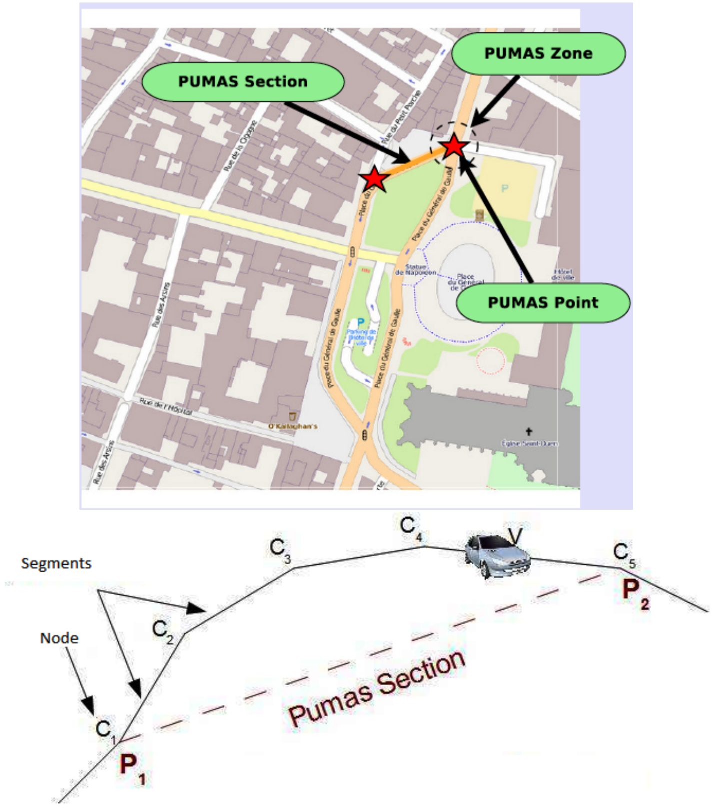


Figure 3.2: PUMAS Digital Map, Added Features

3.2.3 Context Model

As it was stated in the section 3.2.2, the context model will complete the information needed to the digital map though the road model characteristics. Let's start by describing the two existing context models. The first one is the global context and the second one is the local context.

The global context is related to the background objects such as buildings or vehicles. It is possible to find that the background information and the traffic roads have a relationship that give important information when it is combined and also there is the case where the roads show a typical obvious feature that is important. As a consequence, the road model and the context should be adapted to the region of interest where information is useful. However, in the urban road traffic context, the only thing that is very important is this relationship between the road and the vehicles, as Baumgartner et al stated it in his articles [63], [64].

Concerning the local context it is when we assign a relation between a small road number and the context objects. For example, in dense urban areas, the building footprints are almost parallel to the roads and vice-versa. The relation in this case is an occlusion or cast shadow on. Besides, all this relation must be implemented in the extraction process and internal evaluation.

However, for our case we will not focus on this aspect because we are not willing to create a new digital map from scratch but one by taking into consideration the relationship between road and vehicles. This decision is based on our objective regarding the thesis work and the project PUMAS.

The next section is about the map extraction strategy adopted in order to build our digital map by taking into consideration all the decisions made during the analysis presented in section 3.2.2 and 3.2.3.

3.2.4 Extraction and Digital Map Building Strategy

For our thesis work the best representation of the digital map is the direct graph, which means the digital map is built based on the set of nodes and road sections. This kind of representation is based on vector data as it was stated before this is the right representation for our case. Besides, for anything related to traffic estimation it is indeed the most used [65] by most map databases for navigation and display purpose. For these reasons in order to achieve our objective of this thesis, “Open Street Map” [148] has been chosen as a source to extract the digital map. OpenStreetMap (OSM) is an international project established in 2004 to create a free map of the world. The data collected contains many kind of information such as the roads, railways, rivers, forests, buildings and more. The data collected are re-usable under a free license ODbL (since 12 September 2012).

Moreover, this latter is an open source data built by consumers and it is an online access designed to build free editable map of the whole word from GPS traces uploaded by users around the world [66]. However, the physical representation of the road network topology is not accurate or unavailable. Thus, it is important to add the road layout such as intersections and junctions. For this reason, we will add these features that we described clearly in section 3.2.2 (PUMAS points, PUMAS sections, PUMAS zones).

The following graph (Figure 3.3) shows the extraction process and building of the new digital map with the added characteristics (illustration figure 3.4).

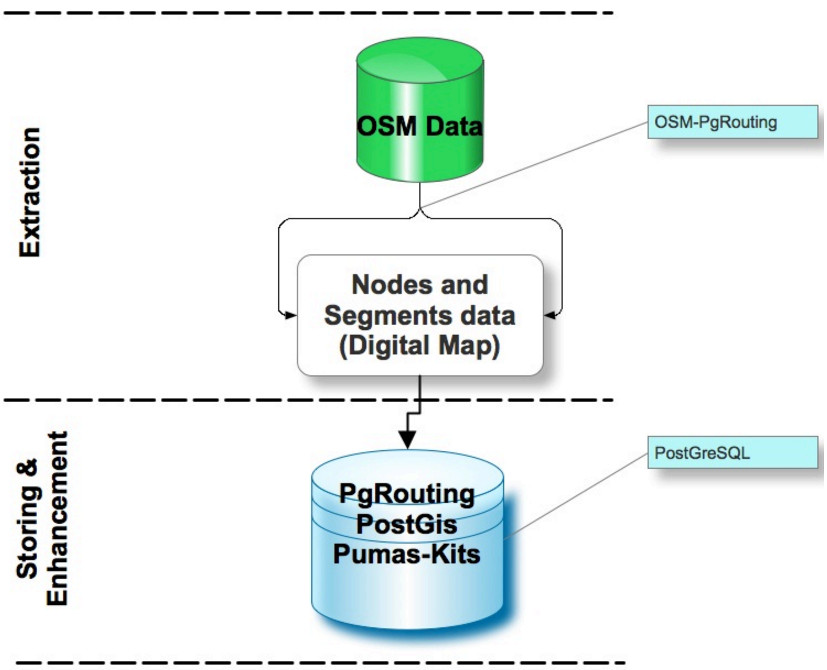


Figure 3.3: Extraction & Enhancement Process

First step is to import the Open Street Map data using "PgRouting" protocol, which allows as the extract the nodes and road sections. With these data we can constitute the digital map. Then at the level of our server using "PostGreSQL" and "PostGIS" queries we launch what we call PUMAS kits that allows adding the new characteristics (PUMAS points, PUMAS sections, PUMAS zones) to the extracted digital map.

PgRouyting and PostGreSql are packages containing all the functions that we can use to create a routing database; Load routing data, Import satellite images, or add information to the map. Moreover, it contains also the database server features in order to run and maintain the GIS information and the digital map data.

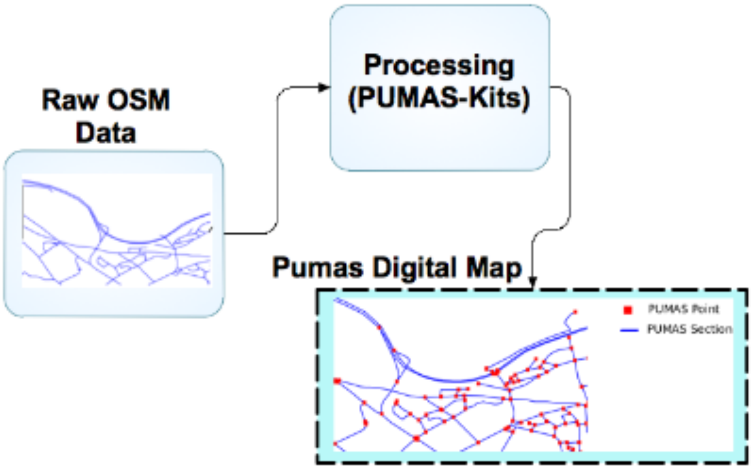


Figure 3.4: View of the Digital Map through the Process

All the data are stored in the database of raw data. Moreover, the system is capable of doing update to the map directly from the OSM data and processes the PUMAS Kits to add the characteristics.

Now after creating the digital map the next important part before doing any travel time estimation is preprocessing. The raw data used should be checked and cleaned and also mapped to the digital map or what is called the map matching process. The next section will debate these issues.

3.3 Raw Data Check and Map Matching

The purpose of this section is to put a light on the importance of validating the raw data before using it. The use of sparsely sampled data presents many challenges concerning the path reconstruction and the map-matching process. The Map matching process is the process of aligning a series of observed user positions with the road network on a digital map. It is a preprocessing requirement step for many applications, such as traffic flow analysis and travel time estimation, which is the case of this dissertation.

3.3.1 Introduction

Estimating travel times per road sections on a urban road network requires an efficient path reconstruction and accurate map-matching, which is the subject of the algorithms described in this section. As it was described in the section 2.3.6 the sparsely sampled GPS data has this characteristic of sending GPS information at a fixed frequency, which is unusual to use for road section travel time or measuring velocities. This type of data presents many challenges.

The first challenge is the frequency of the data that has consequences on the path reconstruction. The probe vehicle can often travel a multiple of road sections between two successive GPS measurements when the frequency is high. Thus, estimating the path travelled by the probe vehicle will have low probability of precision. As consequence it will affect the travel time estimation. For all these reasons the process of running a check or validation of the raw data is a must. The section 3.3.2 will introduce the approach adopted in our case to validate the data used in this dissertation work.

The second challenge is about the map matching. The GPS data measurement should be mapped to the digital map in order to know the exact location of the probe vehicle. Thus the exact location and the path reconstruction between two successive GPS data is a must for the travel time estimation.

The third challenge is the path reconstruction. The path reconstruction is related to look for the realistic path taken by the probe vehicle. This process is very important in order to know the concerned road sections that are involved in the vehicle itinerary. This latter is very important in the travel time estimation per road section.

Based on this introduction it is important to enlighten the fact that any estimation model needs precise information about the digital map (Section 3.2) and the data itself, which means path and location. For these reasons the following sections in this chapter will describe how we managed to reduce the errors coming from the raw data by introducing new approaches in dealing with these issues cited above.

3.3.2 Raw Data Check

The raw data that constitute our historical database is a sparsely sampled GPS data. Sparsely sampled probe GPS data is currently the most ubiquitous data source on the arterial network or urban network. Thus in order to make a check of data frequency or what we call time step let's start by illustrating the data.

The raw data contains information of probe vehicles let's put $Vi \in \{V_1, V_2, V_3, \dots, V_n\}$ the set of floating cars in the database. Each floating car V_i has a set of GPS locations and time information. We will note $t_j^i \in \{t_1^i, t_2^i, t_3^i, t_4^i, \dots t_m^i\}$ the time of reception of the GPS data coming from the floating cars. For the purpose of checking the time step between GPS location t_j^i of a specific floating V_i car we will define two kinds of time steps (equation 3.1):

$$\begin{cases} h_j^i = t_{j+1}^i - t_j^i \\ k_j^i = t_{j+2}^i - t_j^i \end{cases} \qquad (3.1)$$

Where, h_j^i check the time step between the first data and the second one. However, k_j^icheck the time step between the first data and the third one. Moreover our sparsely sample GPS data has a frequency of one minute; therefor, we will define the step $h_s = 60\ seconds$ and the second step $k_s = 120\ seconds$. The defined steps will be used in our equation in order to make a decision.

We define our two check functions (equation 3.2 and 3.3) as follows:

$$\mathcal{F}(h_j^i) = \begin{cases} 1 & if \quad h_j^i \leq h_s \\ 0 & if \quad h_j^i > h_s \end{cases} \qquad (3.2)$$

And,

$$\mathrm{T}(k_j^i) = \begin{cases} 1 & if\ k_j^i \leq k_s \\ 0 & if\ k_j^i > k_s \end{cases} \qquad (3.3)$$

The conditions defined on the two function above was based on the fact that there is a risk regarding the results of our work in this dissertation when the sparsely sampled GPS data with intervals beyond 60 seconds. For example, reconstructing the path will be affected and also the travel time estimation.

Finally, we define our check operator as follows (equation 3.4):

$$\sum_{\substack{1 \leq i \leq n \\ 1 \leq j \leq m}} \begin{cases} \mathcal{F}(h_j^i) + \mathrm{T}(k_j^i) \neq 0 & , then\ we\ validate\ the\ data\ of\ reception\ time\ t_j^i \\ \mathcal{F}(h_j^i) + \mathrm{T}(k_j^i) = 0 & , then\ we\ reject\ the\ data\ of\ reception\ time\ t_j^i \end{cases} \qquad (3.4)$$

The following chain graph shows the building tree structure in order to make the raw data validation using the equation 3.4 (Figure 3.5).

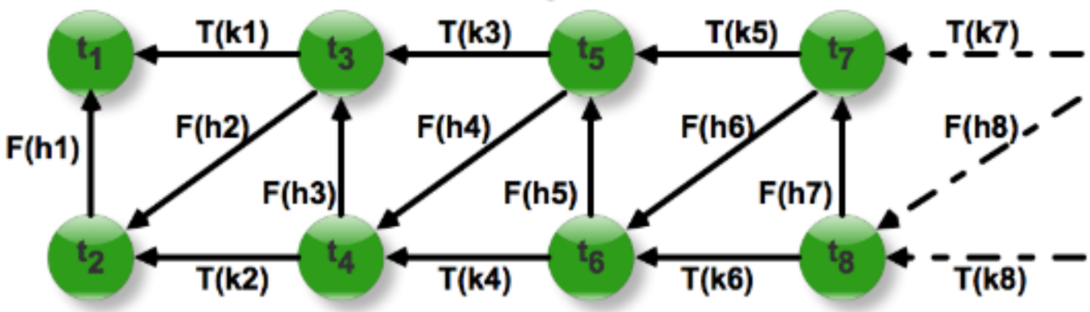


Figure 3.5: Chain structure of raw data Validation for a specific floating vehicle.

The graph shows how the computation structure is built and in each cell we will find the out put of this equation $\mathcal{F}(h_j^i) + \mathrm{T}(k_j^i)$ and then we apply the rule presented above in order to make a decision of keeping the data or not.

3.3.3 Map Matching Process

3.3.3.1 Overview

The map-matching problem is related to the correlation between the probe vehicle path and the vector map of the concerned road section. The problem aspect can vary depending on the input data and also on the desired output. In our case we are concerned by the current position of the probes in our historical database and the input that is the sparsely sampled GPS data.

The purpose behind using the map matching is to correct the GPS localization errors. The trajectory mapping has to process a stream of infrequent position samples and map them to an accurate itinerary. Using the sparsely sampled GPS data brings up a challenge of processing inaccurate data and matches them to an accurate trajectory.

In the literature we can classify the map-matching technique to three kinds or approaches: geometric approach, topological approach, and enhanced approach.

The geometric approach takes into consideration the shape of the road section and the GPS position. In other word, each GPS position is matched to the closed node detected on the road section extracted from the digital map. It is exactly what Bernstein and Kornhauted (1996) named point-to-point matching and the method does not take into consideration the way road sections are connected. Another type of geometric approach is point-to-curve method. In this method as it was clearly stated by Kim (2000) [67] the fixed GPS position is matched to the closed curve by using an orthogonal projection on the concerned road section. The other type is called curve-to-curve (White, 2000; Phuyal, 2002) and it refers to the probe trajectory is matched to the digital map road sections. Moreover there are many studies and research done by combining all these methods described above such as (Taylor, 2001; Bouju, 2002).

The next type is related to the topology of the roads, which means the relationships existing between the entities like points, lines, and polygons. The relationships can be categorized as follows: adjacency when it's a point, connectivity when it's a line, and containment when it's a polygon. We could also combine the geometry with the topology as it was done by Greenfeld (2002) where the map matching is using the geometrical road sections and also the connectivity and contiguity of the road sections.

The last category is the enhanced map matching and we mean by that when the algorithm uses such as Kalman filter, particle filter, fuzzy logic, etc. For example, Honey (1989) hosted in his article [68] the first kind of map matching technique. He used DR sensor positions and then used the map matching to match them into the map. This method has been discussed in many publications [67], [70], one being the [69] where GPS data was used instead of DR sensor positions.

Moreover, Zhao states that the error region can be derived from the GPS error variance. Thus, we can detect the road sections that have a high probability that the GPS position should be associated to them without scanning the whole map.

3.3.3.2 Problem Statement and Definitions

In order to understand the problem of map matching for sparsely sampled GPS data trajectories we will define it in a formal way:

Definition 1: (GPS Trace) A GPS trace is a set of GPS position of a specific probe vehicle $\Psi = \{c_1, c_2, c_3, \ldots, c_n\}$. Each position $c_i \varepsilon \Psi$ contains the following information: latitude “lat”, longitude “lng”, timestamp “t “ and the vehicle heading.

Definition 2: (Probe Trajectory) the probe trajectory T is a set of GPS position with a time interval between two successive GPS position. The GPS positions don’t exceed a certain threshold $\Delta T = 1 \min ute$. For example: $T : c_1 \to c_2 \to c_3 \to \ldots \to c_n$, where $c_i \varepsilon \Psi$ and $0 < c_{i+1} .\ t < \Delta T$ and $(1 \le i \le n)$.

Definition 3: (PUMAS section) is a road section “s” associated with an id. Each section contains the following information: length value s.l, a starting point s.start, ending point s.end, and a set of node and segments or polylines describing the road section in the digital map.

Definition 4: (Road Network) is direct orientated graph G(V,E), where V is a set of vertices representing the road aspects such as intersections, junction, and terminal points. Finally, E is a set of edges illustrating the PUMAS sections.

Definition 5: (Path) is a set of connected road sections which means in our case is a set of PUMAS sections thus $P : s_1 \to s_2 \to s_3 \to \ldots \to s_n$ where $s_k.start = V_i, s_k.end = V_j, s_{k+1}.end = s_{k+1}.start, s_k.cape, 1 \le k \le n$.

Now we can state the map matching problem as follows: Given a raw sparsely sampled GPS data trajectory T and a road network G(V,E), find the path P from G that matches T with its real path.

3.3.3.3 Map Matching System

To describe clearly the map matching that we adopted in our thesis work and also in order to show the modification added. The structure was built this way as it is shown in the (Figure 3.6).

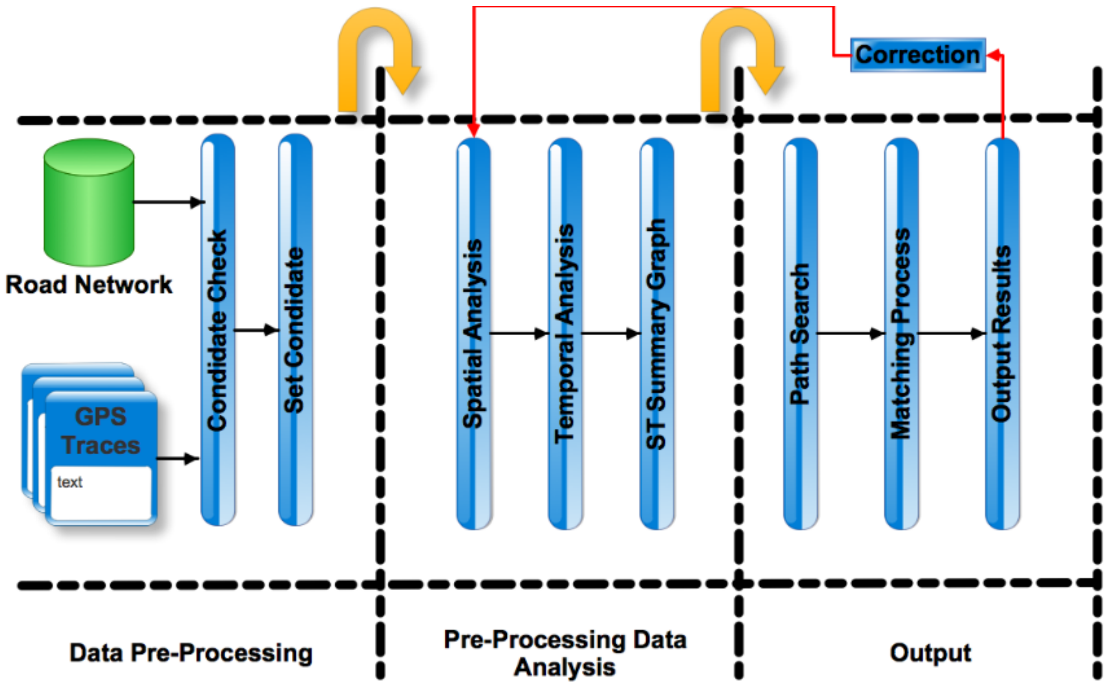


Figure3.6: Map Matching System

The system can be split into three main processes. The data preprocessing is the step where the data is prepared in order to be checked. In the preprocessing data analysis the data will be analyzed following some regulations regarding spatio-temporal rules or analysis. Finally, the results step is when the data is matched to the digital map with respect to the path reconstruction [132] that will be discussed in details in section 3.4.

3.3.3.3.1 Data Preprocessing

This component contains information about road network (edges, vertexes) and GPS traces from the probe vehicle (historical database). Each GPS trace is a GPS trajectory from the probe. This latter will be used to retrieve each possible situation for each GPS sample data. In other words, the system will build a set of all GPS samples and their candidate road sections they lie on.

Thus for a given trajectory $T: c_1 \rightarrow c_2 \rightarrow c_3 \rightarrow \ldots \rightarrow c_n$, we retrieve the entire PUMAS sections candidate for each GPS position c_i within a radius r defined by the GPS receiver error. The next step is to get the line segments candidate where the orthogonal projection of c_i will be on the PUMAS sections. The line segment orthogonal projection will be defined as follows:

Definition 6: (Line segment orthogonal projection, LSOP) the LOSP of a position c_i to a PUMAS section s_i is the point p on s_i such that $p = \operatorname{argmin}_{\forall p_i \in s} dist(p_i, c)$, where $dist(p_i, c)$ is the distance between c and any point p_i on s.

Definition 7. (Notation) we will note s_i^j the jth candidate edge and p_i^j the jth candidate of point p_i (illustration figure 3.7).

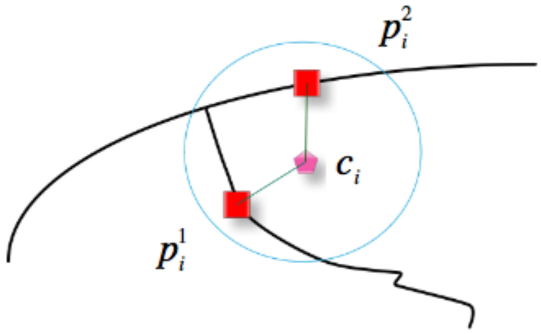


Figure 3.7: Candidate projection points for a sample pi

After selecting all the potential candidates $Pc: p_i^{j_1} \to p_i^{j2} \to p_i^{j3} \to \ldots \to p_i^{j_n}$ the issue now to find the best match trajectory $T: c_1 \to c_2 \to c_3 \to \ldots \to c_n$.

3.3.3.3.2 Preprocessing Data Analysis

In this step we were inspired by (Zhao, 97) [69] approach and we will conduct two analyses to match the trajectory. First the process will start with a spatial analysis and then a temporal analysis.

<u>Spatial Analysis</u>:

This analysis takes into account the distance between a single GPS position and its PUMAS sections candidate. Moreover, it will take into consideration the road network topology to avoid redundant information about the paths. To determine the path we will use the shortest path algorithm (Dijkstra) that we will discuss in details in section 3.4. In order to do this analysis we will define what we call an inspection probability function.

Definition 8: (Inspection probability function) it is defined as the likelihood that a GPS position c_i matches a candidate p_i^j using the distance between these two parameters $dist(p_i^j, c_i)$.

If we want to model the GPS error measurement it can be modeled as a normal distribution [71] of the distance between c_i and p_i^j. This probability indicates the weight of the fact that the point c_i can be matched to p_i^j. We define the Inspection probability function as follows (equation 3.5):

$$\Phi(p_i^j) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(\frac{(y_i^j - \mu)^2}{2\sigma^2}\right) \qquad (3.5)$$

Where,

$y_i^j = dist(p_i^j, c_i)$ is the distance between c_i and p_i^j. For the mean it will be a zero mean and the standard deviation equal to 20 meters based on the empirical evaluation.

The inspection probability function does not take into consideration the context where the GPS position it is. For that reason it can be some situation when matching will be wrong. For example we can have a situation when GPS location has been matched to the wrong PUMAS section, as shown in (figure 3.8) where the matched point was in another PUMAS section and there is no link connecting the two PUMAS sections. In order to solve this issue we will introduce the transfer probability function.

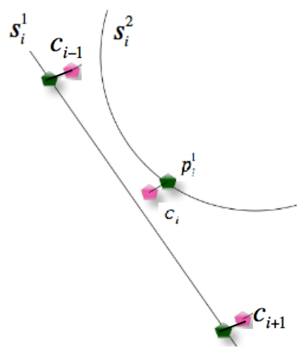


Figure 3.8: An example of wrong matching

Definition 9: (Transfer probability function) the transfer probability for a given projection candidate p_i^z and p_{i+1}^e for two successive sparsely sampled GPS data c_i and c_{i+1} is the likelihood that the true path from c_i to c_{i+1} follows the shortest path through p_i^z to p_{i+1}^e.

The transfer probability function will be expressed as follows (equation 3.6):

$$\Lambda(c_i \to c_{i+1}) = \frac{De_{i \to i+1}}{L_{(i,z) \to (i+1,e)}} \qquad (3.6)$$

Where,

- $De_{i \to i+1} = dist(c_{i+1}, c_i)$ is the Euclidian distance between c_{i+1} and c_i.
- $L_{(i,z) \to (i+1,e)}$ is the length of the shortest path given by the algorithm defined in section 3.4 between p_i^z and p_{i+1}^e.

Therefore the spatial analysis function will be the combination of the inspection probability and the transfer probability. It will be expressed as follows (equation 3.7):

$$\forall 1 \le i \le n, S(p_i^z \to p_{i+1}^e) = \Phi(p_{i+1}^e) * \Lambda(p_i^z \to p_{i+1}^e) \quad (3.7)$$

The spatial function explicitly computes the likelihood that a probe vehicle drove along p_i^z to p_{i+1}^e using the two probabilities defined before. This means that we are taking into consideration the geometric and topological aspect of the network in our computation. However, in reality it is not sure that the probe vehicle will take the shortest path that's why we keep $\Phi(p_i^j)$ in order to reflect that aspect. Each candidate generated will check and have a value of spatial function and the one with the highest scope is the closed map matching of sparsely sampled GPS data to reality.

Temporal Analysis:

The temporal analysis measures the average speed travel between two neighborhoods position. Then it will compare the average speed with the speed limit on each candidate paths. This information will be used to match the trajectory to the candidate's paths with the closest similar speed limit during that time interval.

Thus for a given two candidate points p_i^z and p_{i+1}^e for two successive GPS positions c_i and c_{i+1}. The shortest path from p_i^z to p_{i+1}^e is a set of PUMAS section $\{s_1^{'}, s_2^{'}, s_3^{'}, ..., s_n^{'}\}$. We will note average speed $V_{m,(i,z)\to(i+1,e)}$ of the shortest path as follows (equation 3.8):

$$V_{m,(i,z)\to(i+1,e)} = \frac{\sum_{m=1}^{k} l_m}{\Delta t_{i\to i+1}} \quad (3.8)$$

Where,

- $l_m = s_m^{'}.l$ is the length of concerned PUMAS section of $s_m^{'}$.
- $\Delta t_{i\to i+1} = c_{i+1}.t - c_i.t$ Is the time interval between two sparsely sampled data c_i and c_{i+1}.

Moreover, each PUMAS section $s_m^{'}$ is associated with $s_m^{'}.v$. In order to measure the similarity between the average speed from p_i^z to p_{i+1}^e and the section speed limit; we will use the cosine distance. By considering the vector with k elements of the same speed $V_{m,(i,z)\to(i+1,e)}$ and the vector $\left(s_1^{'}.v, s_2^{'}.v, \dots, s_k^{'}.v\right)^T$ the temporal analysis function will be as follows (equation 3.9):

$$T(p_i^z \to p_{i+1}^e) = \frac{\sum_{m=1}^{k}(s_m.v * V_{m,(i,z)\to(i+1,e)})}{\sqrt{\sum_{m=1}^{k}(s_m.v)^2} * \sqrt{\sum_{m=1}^{k}(V_{m,(i,z)\to(i+1,e)})^2}} \quad (3.9)$$

3.3.3.3.3 Output

In this component we will describe the matching process based on the spatial and temporal analysis. In order to make a clear description we will illustrate the process in the following (figure 3.8). Thus, we were able to generate candidate graph $G_T^{'}(V_T^{'}, E_T^{'})$ for a trajectory $T: c_1 \to c_2 \to c_3 \to \dots \to c_n$.
Where,

- $V_T^{'}$ is the set of candidates for each sparsely sampled GPS data.
- $E_T^{'}$ is the set of PUMAS sections representing the shortest path between any successive candidate positions.

Which means that each G' is associated with $\Phi(p_i^j)$. And each Section is associated with $\Lambda(p_i^z \to p_{i+1}^e)$ and $T(p_i^z \to p_{i+1}^e)$.
Therefore by combining the equations of spatial analysis function and the temporal analysis function we will have our final equation 3.10 as follows:

$$ST(p_i^z \to p_{i+1}^e) = S(p_i^z \to p_{i+1}^e) * T(p_i^z \to p_{i+1}^e), \forall 1 \le i \le n \qquad (3.10)$$

To make a decision about all candidates we will consider a candidate path sequence $P_p: p_1^{z_1} \to p_2^{z_2} \to p_3^{z_3} \to \dots \to p_n^{z_n}$ for the entire trajectory T that is a path in the candidate graph. The total score for candidate path P_p is $F^*(P_p) = \sum_{i=1}^{n} ST(p_i^{z_i} \to p_{i+1}^{z_{i+1}})$. Then the aim is to find the one with the highest score as the best matching path for the trajectory. Thus the selection expression will be (equation 3.11):

$$P = \arg\max_{P_p} F^*(P_p), \forall P_p \in G_T^{'}(V_T^{'}, E_T^{'}) \qquad (3.11)$$

After we find the path P that reflects the true path we run a check on the sparsely sampled GPS data orientation $c_i.cape$ if it is logical with the PUMAS section $s_i.cape$.

Definition 10: (Orientation check function) this function checks the orientation between the sparsely sampled GPS data $T: c_1.cape \rightarrow c_2.cape \rightarrow c_3.cape \rightarrow ... \rightarrow c_n.cape$ and the PUMAS section selected orientation $P: s_1.cape \rightarrow s_2.cape \rightarrow s_3.cape \rightarrow ... \rightarrow s_n.cape$. The function attributes a score of one if it fits otherwise a zero.

The orientation function check is defined as follows (equation 3.12):

$$CF_c(c_i.cape, s_i.cape) = \begin{cases} 1 \quad if \quad c^\circ \cap s^\circ = true \\ 0 \quad if \quad c^\circ \cap s^\circ = false \end{cases} \qquad (3.12)$$

Where,
$c^\circ \cap s^\circ$ is an operator with condition rules defined as follows:

Table 3. 1: $c^\circ \cap s^\circ$ **Operator Table**

$c^\circ = \frac{c_i.cape}{360^\circ}$	$s^\circ = \frac{s_i.cape}{360^\circ}$	$c^\circ \cap s^\circ$
$c^\circ \in]0,0.25]$	$s^\circ \in]0,0.25]$	**True**
$c^\circ \in]0.25,0.5]$	$s^\circ \in]0.25,0.5]$	**True**
$c^\circ \in]0.5,0.75]$	$s^\circ \in]0.5,0.75]$	**True**
$c^\circ \in]0.75,1]$	$s^\circ \in]0.75,1]$	**True**

The table above shows the condition when the intersection rule should be true. Which means that the operator is equal to true when c° *and* s° are from the same interval.

However we don't run this check on a specific sparsely sampled GPS data if the speed on the probe at that position is lower than 4km/h. Because based on the empirical data the orientation is not precise or sometimes wrong when the speed is lower than 4km/h.

In case the check was activated and finds a false case where the orientation is not following the same orientation of the PUMAS section. We will call the correction process to correct the sparsely sampled GPS data concerned.

<u>Correction Step</u>

This process of correction aims to change the concerned position c_{k+1} by a predicted position c'_{k+1} knowing c_k and c_{k-1}. Many researches have been done regarding this issue of prediction. In our case we will be inspired by [72] where the author used a method to complete the gap of lost data and we will adopt it to our case.

In our case we have each sparsely sampled GPS data can be characterized by the position coordinated x-axis and y-axis. Thus we can write $c_k(x_k, y_k)$ and $c_{k-1}(x_{k-1}, y_{k-1})$. The future position or predicted position will be $c'_{k+1}(x_{k+1}, y_{k+1})$ and will be predicted as follows (equation 3.13):

$$\begin{cases} x_{k+1} = x_k + u * \cos\theta \\ y_{k+1} = y_k + u * \cos\theta \\ and, \quad \theta = \arctan\left[\dfrac{y_k - y_{k-1}}{x_k - x_{k-1}}\right] \\ and, \quad u = \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2} \end{cases} \tag{3.13}$$

Where, u represents the step length between each sparsely sampled data. θ is the estimated direction angle of the probe.

After predicting the new position, the correction process is done and we run again the system with new correction added to the raw data as (figure 3.6) shows. Moreover, this process of correction is done only once for each set of trajectories T from the raw data.

To sum up in this section 3.3 we showed how the raw data is processed in order to make it ready to be used for the travel time estimation. In the following section we will describe in details the path reconstruction that we used in the map matching process.

3.4 Path Reconstruction on Database Context

Path reconstruction can be seen as the shortest path problem between two vertices in weighted graphs is very well known as a graph problem. There are many solutions that have been developed for different types of graphs such as Dijkstra in the case of positive weights, Bellman in the general case, etc. However, these methods are time consuming regarding the computational process, which means that they need enhancement and speed up. In this section we will discuss these techniques and how we dealt with it in this thesis work.

3.4.1 Introduction

The choice of a way to move goods can be done by taking into account the travel time, cost, parking, etc. In general, the tools used to guide this choice are based on solving technical "problem of shorter path". Given a graph $G_T^{'}(V_T^{'}, E_T^{'})$ for given starting candidates position $p_i^z \in V_T^{'}$ to the final destination candidate position $p_{i+1}^e \in V_T^{'}$ the issue is to find the optimal shortest path regarding the constraints to travel from p_i^z to p_{i+1}^e.

The shortest path problem is one of optimization problems over the network studied since the fifties (Bellman, 1958 [74]). All these studies are based on finding a path in a graph based on a single objective. Indeed, the shortest path problem for a traditional objective of minimizing the time, cost, or both.

One aspect of shortest path problem that interests us in this dissertation is the time-dependent shortest path. This issue has been studied for the case of determining the shortest path. In other words we looking for the shortest path between two nodes or positions and it is called in the literature the point-to-point shortest path problem with time dependency.

Concerning the point-to-point algorithm the most famous one is the Dijkstra 1959. The process starts by the initial position and it scans all the neighbors around it. All the paths found will be ranked based in the minimal sum of cost of the visited edges or roads in our case. In each subsequence step, the algorithm will apply a relaxation to the minimal cost, which means it has the highest rank then visits again all its neighbors and ranks them again. The process will stop when the end node or position is reached.

The number of nodes visited during the search usually helps to evaluate the routing algorithm performance. For example, Dijkstra makes a scan of the neighbors in a circular area aspect enclosing the start position and the ending position. This explains why the algorithm needs a lot of computation time when the road network is dense. For this reason, many methods have been embedded to the algorithm to make it faster.

To sum up these enhancement methods we will put them into categories as it was cited in [75] as follows:

Shortest-path with Bi-directed search: In this approach as it was illustrated in [76], [77] the search starts from both positions start and end, at the same time. Thus, the scanning time will be reduced because there is two processes running at the same time and the visit for a specific node will happen only once. By using this approach the running time improves.

Shortest-path with Goal-directed search: This approach affects the nodes ranks that are unlikely to be on the shortest path [78]. Moreover, it applies the same principal to the road section regarding the shortest path.

Shortest-path with Multi-level approach: This method in [80] adds shortcuts indexes to the road network when it is possible in order to bypass many edges at a time during the routing process. The sub-class follows a hierarchical routing. First layer contains the full resolution of the network and the next layer contains less nodes and road sections. This aspect makes the searchlight but we lose in the accuracy.

Shortest-path with Bounding boxes: Checks if there is any possibility that the concerned node can be in the shortest path if not it is rejected and not considered [79].

There are many researches done in the shortest path problem but we have to be more specific regarding our case to narrow our focus on the real issue in this dissertation. The fact that the data used is sparsely sampled GPS data and we have to find the path between each pairs of successive data put as in the situation of point-to-point shortest path. Moreover, we have to take into consideration that between each two GPS position there is a time gap, which means that our approach to solve this issue is time dependent. To sum up we are dealing with point-to-point shortest paths on time dependent road networks.

3.4.2 Related work and Analysis

The shortest path problem is one of the subjects that have been conducted for many decades. Besides, many innovative ideas have been proposed to compute the point-to-point shortest path. Starting from the 1950th a lot of work related to our subject was developed but the purpose behind was different. The studies done in solving the shortest path were more for transportation planning or road network analysis not for individual travel time.

The first work in this period related to time dependent shortest path with a recursive formula that gave the needed shortest path with the minimum travel time between a starting point and ending position was citied in [81]. In this article by taking the positive travel time value the method gives the shortest path from all nodes to a given final destination.

In Dijkstra's algorithm [73] the shortest path problem in a static directed graph is solved by using non-negative weights in a polynomial time. The algorithm can easily be extended to dynamic case with time dependence [82]. However, the FIFO in the tree structure is not mentioned which is really important to get the right path in time dependence case.

There were many similar algorithms developed to solve this issue such as by Whiting & Hillier. But the most used and enhanced algorithm is Dijkstra and there are many derived algorithm from it. One of the famous one is the A* algorithm which adds the concept of closeness [87] to push the search to the target node [88]. Besides, based on the nodes visited the Dijkstra visits almost all of them however in terms of norm he is actually the smaller.

One of the aspect that shortest path have it is this time dependence issue which is interesting for our dissertation. The time-dependent shortest path problem has many applications in transportation. One of the articles that attracted us is [86] where the other adopted an algorithm where he compute all-to-one shortest path in a discrete dynamic network. Moreover, the author emphasizes that his method is the most appropriate to the dynamic transportation systems like the intelligent transportation systems. He put also in his approach specific characteristics that the shortest path problem depends on. We summarized them as follows in the table below.

Table 3.2: Time-dependent Shortest Path Classification after [86]

Characteristic	Variants
Network properties	• FIFO network • Periodic network: cost and delay follow a periodic patterns • Zero delay arcs: no delay taken into consideration
Waiting cost	• With memory: waiting cost depends on waiting period • Without memory: cost is independent from waiting time
Waiting constraints	• Forbidden waiting • Allowed waiting • Bounded waiting: waiting allowed but bounded.
Source and destination	• One-to-all: finding the path from the source to all possible destinations. • All-to one: finding the path from all possible sources to one destination. • One-to-one: finding the path from one source to one destination
Objective	• Fastest path: minimum cost path with cost = delay • Bi-criteria: cost and time • Multi-criteria path: the use more than one criteria.

Another aspect of shortest path problem is in the database context, which is similar to our work in this dissertation. The fact it is on a database it involves the development of intelligent transportation systems (ITS). Our thesis deals with this issue of time-dependence shortest path on a database context. Besides, there is much reason by storing this processed data that can help for ITS purposes. For example in [84] the author suggested storing data in order to perform shortest path process. In [85] the authors explicitly refer to this issue where they deal with moving objects and they use a classical shortest path approach based on distance.

3.4.3 Problem statement

In the domain of traffic the road network are dynamic and time dependent. In fact between two positions of sparsely sampled GPS positions c_k and c_{k+1}, we can have many possible paths. Moreover, the time received at the first position and the second position can vary depending on the probe vehicle. In our work we will focus on finding the right path based on the travel time per road section and also the travel time between the two positions c_k and c_{k+1} in order to reconstruct the path that was used by our probe vehicle.

The problem in our case is to find the optimal path that respects the travel time between c_k and c_{k+1}. By stating our problematic, it will be easy to find the appropriate algorithm, adapt it, and enhance it to our case.

As a start we will define clearly the model:

Definition 11: G(V,E) is an oriented graph which means that each edge (i, j) has a defined orientation regarding its connections with other edges. This aspect makes the graph dynamic. Moreover, each position c_k has a time of reception associated to it $c_k.t$.

Definition 12: let $\prod : E \rightarrow R$ be a cost function that assigns a cost Π_{ij} to each $(i, j) \in E$. Given a root $r_p \in V$ the shortest path problem (SPP) consists in finding the directed trajectory T such as the path r_p from i to j exist and it is the shortest path with respect to the weight function.

Definition 13: (First in First out property) for each pair of time t and t' with t>t': $\forall (i, j) \in E, \Pi(i, j, t) + t \leq \Pi(i, j, t') + t'$. This means in our case, that when the probe vehicle left from a node at time t he cannot arrive a final destination node before t'. That's why this condition is really important in our case to rebuild the path of our sparsely sampled GPS data in our historical database especially in the case of a death node (the reached node does not lead to the final destination).

First, before modeling the time-dependent shortest problem (TDSPP), which is the interest in our case, let's start by formulating the shortest path problem. The classical formulation of shortest path problem is as follows:
Let $M \in \{-1,0,1\}^{|V|*|E|}$ be the incidence matrix of the graph G(V,E) and its elements are m_{ij}^{v} and (equation 3.14):

$$m_{ij}^{v} = \begin{cases} 1 \quad if \quad v = i \\ 0 \qquad otherwise \\ -1 \quad if \quad v = j \end{cases} \tag{3.14}$$

Moreover we will consider a network flow problem [83] illustrated by f_v such as (equation 3.15):

$$f_v = \begin{cases} 1 \quad for \quad v = i \\ -1 \quad for \quad v = j \\ 0 \quad \forall v \in V \setminus \{i,j\} \end{cases} \tag{3.15}$$

Thus the shortest path problem will be as follows (equation 3.16):

$$(SPP) \begin{cases} \min \quad \sum_{(i,j)\in E} \Pi_{ij} x_{ij} \\ \forall v \in V \quad \sum_{(i,j)\in E} m_{ij}^{v} x_{ij} = f_v \\ \forall (i,j) \in E \quad x_{ij} \in \{0,1\} \end{cases} \tag{3.16}$$

The representation above illustrates one unit of data with the requirement that the unit reaches the destination passing the arcs (i,j) at minimum total cost. The fact that SSP is a linear program, then all the solution is solvable. Thus it is easy to add the time dependence aspect to our formula (equation 3.16) by introducing extra variable t_v, which is the time of arrival at node v.

$$
(TDSPP)\begin{cases}
\min \quad t_v \\
\forall (i,j) \in E \quad x_{ij}(t_i + \Pi(i,j,t_i)) \le t_j \\
\forall v \in V \quad \sum_{(i,j)\in E} m_{ij}^v x_{ij} = f_v \\
\forall (i,j) \in E \quad x_{ij} \in \{0,1\} \\
\forall v \in V \quad t_i \ge 0
\end{cases} \tag{3.17}
$$

Where $\Pi(i,j,t_i)$ represents the cost of the arc (i,j) at time t_i. The representation above of the TDSPP contains the flow conservation constraint of the road network and also the arrival time t_j at the destination node j starting from departure node i at time t_i. Since, in our case we are trying to find the shortest path to reach node j with respect to the travel time between node i and j that we know from our sparsely sampled data; the arrival time constraint is satisfied. Which means $t_i + \Pi(i,j,t_i) = t_j$.

However this is a general representation of our case with a cost function and its constrain regarding the data used. Besides, we will state clearly how to define our cost function and the process in the following sections of this dissertation. But the problem can be summarized to a *Time-dependent Shortest Path Problem for a Given Departure and Arrival Time (TDSPP-GDAT).*

3.4.4 Strategy adopted To Solve TDSPP-GDAT

In this section we will start by reminding the case problem and define some concepts and definitions that play an important role in the solving process.

First, in our case we have sparsely sampled GPS data that constitute our historical database. This data has gap of information between each successive data. The problem is to find out the real path that has been taken by the probe vehicle between two known positions [129].

Let's now define some function that we will use in showing how we solved this graph problem.

Definition 14: (Travel time constraint function) is the time needed to go from the source position $c_k.t$ to the destination or target $c_{k+1}.t$. The function will be defined as follows: $h(c_{i\to i+1}.T) = c_{k+1}.t - c_k.t$. In other words this function represents the travel time that the probe vehicle did to go through the two positions.

Definition 15: (Travel time per road section function) this function computed the travel time needed to cross it. The formula is a basic one based on the length of the road section and the speed limit on the concerned road section. This latter information we get from the digital map. $f(e_i.t) = \dfrac{e_i.lenght}{e_i.V_l}$. (3.18)

Definition 16: (Cost function) clearly the cost function will be related to two other functions defined early that are related to the departure time and arrival time. The cost function will give a sense to the constraints by an accumulated cost associated to the estimated travel path or formally (equation 3.19):

$$\Pi(k,k+1,c_{k\to k+1}.T) = \sum_{k\le i\le k+1} \frac{f(e_i.t)}{h(c_{k\to k+1}.T)}. \qquad (3.19)$$

Definition 17: (Total path weight) the total weight path is the weight that we attribute to a specific path starting from source position to the final destination (equation 3.20).

$$\{W(P_i) = c_k.t + \Pi(k,k+1,c_{k\to k+1}.T) \qquad (3.20)$$

These formulations constitute the core of the process of making a decision regarding the time dependency and constraints in our case.

The system process was done at the database level in order to run the process of learning routing process. This means that we will find the entire time dependent shortest path possible between all the nodes. The process will be done only once on offline status. Then on online status we will call the module, he will give the solution that he has already in his archives saved on the database.

In order to solve the time- depended shortest problem in our case *TDSPP-GDAT* we decided to combine two approaches the classical algorithm of Dijkstra [73] and the time-dependent shortest path problem with least travel time in a given interval presented by Ding [89]. However, this latter approach has been chosen because it can be easily generalized and modified for our case *TDSPP-GDAT*, by simplifying the arrival time function to our special case. This means that the path is determined after the travel time per road section function is processed. In other words we defined the time dimension on each node of the graph. Then we have to apply the travel time constraint function in order to have the weight on the edges for all the paths. Based on this decoupling we are close to what is called in literature Two-step- least travel time [89].

Definition 18: (Two-step- least travel time) the algorithm determines the least arrival time to a specific node by performing a Dijkstra. The main advantage of this algorithm of decoupling the time refinement and path selection is the ability to figure out the right arrival time when we already know the arrival time.

Based on this definition, the situation is applicable to our case we will add some modification to the strategy in order to make useable to solve our case. Because in Tow-step least travel time they try to estimate the right arrival time function; however, in our case we already know the arrival time. The issue then is to find the path with arrival time function that is close to reality.

For this reason we adopt an approach where we will gradually explain the process in the database context step by step. We distinguish two kinds of steps; the first one is an offline step where we make a learning process of our digital map or graph and the second one is online step where we find the appropriate path for our case.

Before starting the algorithm we generate our graph by having all the vertices set and all edges sets and also attribute the time distance to them by using the travel time constraint function.
First step (offline)- Dijkstra based on travel time. At this level the purpose to scan the whole graph that is already built with all the edges and vertices enhanced with travel time per road section function information. This means that the Dijkstra is involved in his search aspect. The algorithm starts by initialization process where he gets from the list of vertices the start point v_k and the arrival point v_{k+1}. Then he will search for all possible connections that can lead him to reach the arrival point and computing at the same time the total travel time per road section function as it was defined in definition 15 for each path found. The final results will be a graph that contains information about all paths possible from a starting vertex to an arrival vertex knowing the travel time needed to do so. In order to make it clear we break it down into phases as follows:

- Phase1: all the vertices in the graph are assigned to a travel time distance equal to infinity except the initial vertex, which is equal to zero.
- Phase2: set the initial node to status of current and all the other vertices to unvisited. Create a set of unvisited vertices excluding the initial vertex.
- Phase3: take into consideration all unvisited neighbors of the current vertex and compute their travel time distance.
- Phase4: after considering all the potential neighbors we mark the current vertex as visited thus it will be removed from unvisited vertices set. This means that the visited vertex will be never visited again and its time distance to all neighbors is recorded.
- Phase5: the next current vertex is marked with the lowest time distance in the unvisited vertices set.
- Phase6: in case the unvisited set is empty then we stop. Otherwise, the vertex in the unvisited set vertices with the lowest time distance is set to current and we go back to phase3.

This offline process will provide us a database with all possible tree paths from a start point to an arrival point including the Travel time constraint function or time distance information for each connection. Now during the online step the information collected from the offline step will be used to compute the Cost function and find the logical path for each pairs of the sparsely sampled GPS data in the historical database.

Second step (online)- shortest pathfinder. In this step we will have as input the two position from our sparsely sampled GPS data, starting position c_k and the arrival position c_{k+1} with their time respectively $c_k.t$ and $c_{k+1}.t$. Moreover, we will have also the information concerning to which edge they are assigned to. This latter information will help us find the concerned vertices v_i and v_{i+k}. By consulting our database of possible paths done in first step we will have the set of all possible paths. Therefore, it is easy to compute the weight of each path using the cost function defined in definition 16 and as a consequence we will have also the total weight for each path by conducting definition 17. Now we will have the set of total weight per path $W(P_i)$.

The next step is to get the real weight because we did not take into the consideration the time distance between the GPS positions and the concerned vertices.

Definition 19: (Data time distance function) is the function that computes the time distance between the GPS position on the edge and the end or start point of the edge. We will define as follows: $T_{DS}(c_k) = \frac{f(e_k.t) * L(c_k)}{e_k.lenght}$ (3.21).

Where, $L(c_k) = dist(v_k, c_k)$ is the Euclidian distance between GPS position and the concerned vertex on the graph.

Therefore the weight for a specific path will be as follows(equation 3.22):

$$Total\_Weight(P_k) = T_{DS}(c_k) + \Pi(k, k+1, c_{k \to k+1}.T) + T_{DS}(c_{k+1}) \qquad (3.22)$$

Finally, the optimal path solving our TDSPP-GDAT will be the one with the highest total weight. Thus the method proposed eliminates the data outliers when compared to an estimated route

3.5 Conclusion

In this chapter we went through what we call in our dissertation the preprocessing tools. This latter makes reference to all processes needed to make the raw data in an urban context ready to be treated.
The first process was to extract the digital map from the “openstreetmap” and also add new characteristics to it for enhancement purposes. Then we processed

a raw data check in order to clean our Historical database from any ambiguity information.

The second main tool in this chapter was the map matching process that we used to match our raw data to the digital map. Our map matching was explained and also the entire enhancement added to it in order to reduce all the error factors and get a good representation of the data reflecting almost the reality.

The third part was related to the time-dependent shortest path problem given a departure and arrival time. This is used by the map matching process at a certain level of the process. In this part we explained our new approach in a database context. We have adopted a new step called a learning phase that we processed on offline status and then we described how the algorithm reacts when the map matching on online status calls it.

Finally, after making all our raw data ready and filtered we will discuss in the following chapter the core of our dissertation which is the travel time estimation in a historical database context.

Chapter 4: Travel Time Estimation using Sparsely Sampled Data in an urban network

4.1 Introduction

Estimating travel time in an urban road network using sparsely sampled data from probes has proven to be a substantial challenge. The sparsely sampled GPS data represents the vast majority of the data available in urban road networks environment.

Nowadays the feature of probe vehicle data includes the variety of data type, the lack of ubiquity and reliability, and the random aspect of its spatio-temporal coverage. All these features make it insufficient to make a clear idea about the macroscopic traffic model parameters with clear statement about the state estimation for the large transportation network.

Moreover, the penetration rate of this kind of data is still typically low which mean the information regarding the urban network can not be homogeneous thus it is not representing the full traffic state of the system [90].

However, this data can be processed and enriched to make it usable for the macroscopic level. In order to so we can for example enrich it with travel time estimation per road section.

Regarding the literature on travel time estimation using this kind of data presents two challenges the first is finding the path that was taken by the probe, which tried to solve in chapter 3 of this dissertation. The second one is to find the travel time spent in each road section and there are many methods that have been developed for this purpose such as:

In [90] where the author used a probabilistic approach to model the travel time in an arterial network using low frequency taxi GPS information. The model developed assumes the following hypothesis: between two successive data there is at least a road section and it does not take into account the case when the two GPS data are located in the same road section. The problem with this approach, it needs to observe the route travel time on the road section, which is not the case with sparsely sampled data. That's why he used a simulation based on the maximum likelihood. This solution is good after all but it is not reflecting the reality and it complexity.

Another approach was presented in [92], where the author used a statistical approach to increase the reliability and the estimation of travel time by using all the observed data from the probes. The model learns from the data and predicts the travel time for any trajectory in the network.

Whereas, there is another context of estimating travel time, which concerns us in our dissertation, is estimating travel time in a historical database. Some of the work done in this area was based on the analysis of traffic information from the real world data. By applying many techniques such as fuzzy logic, artificial intelligence, mathematical models or statistical models.

For example in [93] the authors established a mathematical model to estimate travel time using GPS data from probe vehicles. The historical travel time data was used to adjust his statistical travel time estimation model based on historical travel time data. The result of this approach was good in a fluid traffic but the model was unable to perform good results when there is congestion in the network.

A probabilistic model to estimate travel time in arterial network using sparsely sampled GPS data is presented in [149]. The author used an Expectation-Maximization algorithm to estimate the probability distribution of travel time through the road section. As the roads sections travel time are not directly observed, a simulation based on EM algorithm is proposed. The author assumes in his analysis that the travel time per road sections are independent, which not the case in reality.

In some article like in [150] and [151], they took into consideration the dependency of travel time per road sections using Spatio-temporal auto regression moving average approach. These models, however, were applied to stationary sensor data and not probe vehicle data. Thus, they used the traffic flow information in order to estimate travel time.

Another approach, using sparsely sampled GPS data from ambulance, is presented in [152]. The author used a Bayesian approach to estimate travel time. The travel time was assumed to be independent and follow a log-normal distribution. Besides, the parameters are estimated using a Markov chain. However, the approach presented does not take into consideration the variance of travel time and speed.

All the methods presented do not take into account very important fact such as the dependency and the variance of travel time per road sections. That’s why we choose to use particle filter in our approach because it gives us this ability to take into consideration all those features of travel time in reality (dependency, variance).

Our approach in this thesis work will be by presenting a new method to estimate travel time per road section at a microscopic level using Monte Carlo Method and historical probability distributions of travel time on each road section. The estimation will be done in the Historical database in order to enrich it with travel time information using sparsely sampled GPS data.

4.2 Problem Statement

In this section we will try to formalize the problem of travel time estimation in our case. We will start by illustrating the situation that we have to solve. First, we will state some definitions in order to make it easy to understand the problem.

Definition 19: (PUMAS Points) is the virtual node on the digital map where the road section intersects (green squares on Figure 4.1). This means that any road section or PUMAS section is defined by a starting PUMAS point n_{pp}^{i} and an ending PUMAS point n_{pp}^{j}.

Definition 20: (Moment of Passage, Mp_i^j) is the time when the probe vehicle passes through the PUMAS points. Thus each PUMAS point will have the information related to its position on the map and also the estimated time when the probe vehicle went through it Mp_i^j.

Definition 21: (Travel time per section) is the total time needed to cross the whole road section. In other words it is the time needed to go through the starting PUMAS point of the road section till reaching the ending PUMAS point of the same road section.

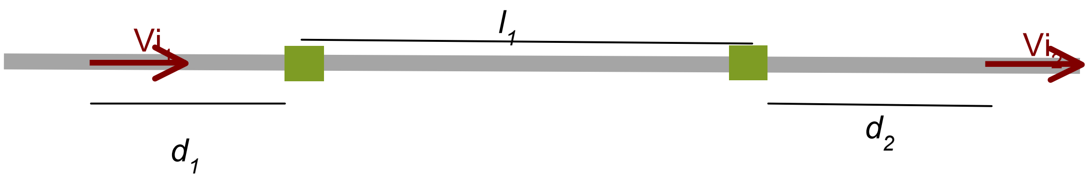


Figure 4.1: Illustration of Sparsely Sampled GPS Data on the Digital Map.

The issue is to estimate the travel time between two successive sparsely sampled GPS data. The process start by detecting all the road sections which means find out all the PUMAS points between the two successive sparsely GPS data. Moreover this process can be done successfully when we apply all the processes described in chapter 3. After determining the PUMAS points the remaining step is to estimate the moment of passage through those virtual position (PUMAS points) then it is easy to find out the travel time estimation per road sections.

Our approach to solve the problem will be based on the family of sequential Monte Carlo Methods (MCM). These methods are interesting for us due to their

aspect. They are a comprehensive approach based on an exploration of the space state of the problem using particles with a randomly changing dynamics. All these particles are distributed according to the probability of the process to estimate conditional observations issued by the sensors.
Moreover, it does not require explicit resolution of the problem equations; this method is applicable regardless of the complexity of these equations, especially in terms of non-linearity and non-Gaussian.

4.3 Particle Filter Model

The particle filter (PF) proposed originally by Gordon et al [95], this kind of algorithm which uses particle filtering has gradually emerged as the best technique for processing nonlinear signals. The concept of resampling introduced by Gordon has allowed many issues related to estimations and predictions to be solved, and also opened new windows to be explored and developed. The particle filtering algorithms can estimate probability through successive measurement by using a finite set of Dirac measurements centered in the corresponding points of "particles".

This approach has a lot of success in the scientific community and its application has covered all kind of fields. It was used in the 1950s to simulate polymer chains of great length [99] and also to solve problems of physics during World War II [100]. Since the 1990s, these simulation methods have become widespread in various fields such as radar signal processing [101], genetics [102], robotics [103], and the classical Bayesian estimation [104], and also it has been applied to localization, navigation, tracking [94] or multiple target tracking [96], vision [97], and communication [98].

In many applications the goal is to estimate the posterior probability density for the states by using some observations. Thus we have to define the states equation parameters that we will denote by the vector $X_t = \{x_0, x_1, x_2, \ldots, x_t\}$ while the vector $Y_t = \{y_0, y_1, y_2, \ldots, y_t\}$ denotes all the observations up to time t. Let's consider the following model representing the state equation and the observation equation 4.1:

$$\begin{cases} x_{t+1} = f(x_t) + u_t \\ y_t = h(x_t) + b_t \end{cases} \qquad (4.1)$$

Where $u_t \sim p_{u_t}(.)$ and $b_t \sim p_{b_t}(.)$ are the process and measurement noises respectively. Besides, f and h are two arbitrary nonlinear functions. Moreover they are assumed to follow an independent distribution with known densities. And the system state process can be defined as follows (equation 4.2):

$$p(X_t) = p(x_0)\prod_{k=1}^{t} p(x_k \setminus x_{k-1}) \qquad (4.2)$$

The prior distribution of the state at initial state t=t₀ is given by $p(x_0)$. The particle filters are usually used when the posterior density $p(X_t \setminus Y_t)$ and the observation density $p(Y_t \setminus X_t)$ are non-Gaussian. The observations are conditionally independent given the states (equation 4.3):

$$p(Y_t \setminus X_t) = \prod_{k=0}^{t} p(y_k \setminus x_k) \qquad (4.3)$$

We should forget that the purpose of the method is to estimate the posterior density $p(x_t \setminus Y_t)$. Since the case is neither linear nor Gaussian the posterior representation should be the total probability density function. Thus the estimation will be as follows (equation 4.4):

$$\begin{aligned} p(X_t \setminus Y_t) &= \frac{p(y_t \setminus X_t, Y_{t-1}) p(x_t \setminus X_{t-1}, Y_{t-1})}{p(y_t \setminus Y_{t-1})} p(X_{t-1} \setminus Y_{t-1}) \\ &= \frac{p(y_t \setminus x_t) p(x_t \setminus x_{t-1})}{p(y_t \setminus Y_{t-1})} p(X_{t-1} \setminus Y_{t-1}) \end{aligned} \qquad (4.4)$$

The expression became as it is shown above due the definition of the system's state process (equation 4.2).
The key idea behind the PF is to approximate the probability distribution by applying a weighted sample set. Let's $\left\{X_t^{(i)}\right\}_{i=1}^{N}$ are the samples set drawn from the posterior. Then the estimation expression will be:

$$\hat{P}(X_t \setminus Y_t) = \frac{1}{N}\sum_{i=1}^{N} \delta(X_t - X_t^{(i)}) \qquad (4.5)$$

Where $\delta(X_t)$ is the Dirac delta function. The sampled in this situation are equally correct because they are drawn from the posterior itself. As a consequence, their weight sum will be equal to one. The value 1/N added is in order to respect the law of total probability. This estimation can be used for example to have an idea about different moments of the posterior by computing the expectation and the covariance.

The samples that we have cannot be drawn from the posterior simply because it is unknown. To mitigate this problem, it is possible to choose an alternative known probability distribution density $q(X_t \setminus Y_t)$ instead of using $p(X_t \setminus Y_t)$. Therefore, they are drawn using $q(X_t \setminus Y_t)$ which is depending on Y_t (equation 4.6).

$$q(X_t \setminus Y_t) = q(x_t \setminus X_{t-1}, Y_t)q(X_{t-1} \setminus Y_t)$$
$$= q(x_t \setminus X_{t-1}, Y_t)q(X_{t-1} \setminus Y_{t-1}) \quad (4.6)$$

The last equality (equation 4.6) that is shown is due to the restriction that the states at time t-1 and older are independent of the measurement at time t. This implies that we have the power to draw $\left\{x_t^{(i)}\right\}_{i=1}^{N}$ from $q(x_t \setminus X_{t-1}, Y_t)$ and $\left\{X_t^{(i)} = \left\{X_{t-1}^{(i)}, x_t^{(i)}\right\}\right\}_{i=1}^{N}$ form $\left\{X_{t-1}^{(i)}\right\}_{i=1}^{N}$ the set without adjusting.

To make the estimation of the posterior by exploiting these samples we should associate them with what is called the importance weight w_t (equation 4.7).

$$w_t^{(i)} = \frac{p(X_t^{(i)} \setminus Y_t)}{q(X_t^{(i)} \setminus Y_t)} = (\frac{p(Y_{t-1})}{p(Y_t)})\frac{p(y_t \setminus x_t^{(i)})p(x_t^{(i)} \setminus x_{t-1}^{(i)})}{q(x_t^{(i)} \setminus X_{t-1}^{(i)}, Y_t)}w_{t-1}^{(i)} \quad (4.7)$$

The only relationship that is important for us is the weights regarding the particles thus we can neglect $\frac{p(Y_{t-1})}{p(Y_t)}$. Thus the expression of equation 4.7 will be:

$$w_t^{(i)} = \frac{p(y_t \setminus x_t^{(i)})p(x_t^{(i)} \setminus x_{t-1}^{(i)})}{q(x_t^{(i)} \setminus X_{t-1}^{(i)}, Y_t)}w_{t-1}^{(i)} \quad (4.8)$$

We can make it simpler by using the state of propagation density (equation 4.9),

$$q(x_t \setminus X_{t-1}, Y_t) = p(x_t \setminus x_{t-1}) \quad (4.9)$$

Therefore, the weight equation 4.8 will finally as follows:

$$w_t^{(i)} = p(y_t \setminus x_t^{(i)})w_{t-1}^{(i)} \quad (4.10)$$

As a consequence, the estimation of the posterior equation 4.5 will be at the end as follow:

$$\hat{P}(X_t \setminus Y_t) = \sum_{i=1}^{N} \bar{w}_t^{(i)} \delta(X_t - X_t^{(i)})$$
$$\bar{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^{N} w_t^{(j)}} \quad (4.11)$$

During the PF algorithm process, time evolves; as a consequence, it pushes the samples tend to spread and the weights for almost all the samples tends to zero. This means they are not helping to converge to the estimation. However, there is a solution to this issue. We should just check the covariance of the samples and the covariance obtained from the weighted sampled. Besides, by comparing the two covariance, we will get the efficiency measurement of our sampling. It is shown in many application of the filter like in [108] that we can estimate the efficiency by using the following expression (equation 4.12):

$$N_{eff} = \frac{1}{\sum_i (\bar{w}_t^{(i)})^2} \qquad (4.12)$$

And in order to know if the samples have spread far enough the use of threshold can give this information normally it is defined as follows: $N_{th} = \frac{2N}{3}$.

To sum up, the particle filter algorithm can be illustrated as follows:

1. Initialization at t=t₀	Generate samples $x_0 \sim p(x_0)$ for $i = 1,........,N$ Calculate $w_0^{(i)} = p(y_0 \setminus x_0^{(i)})$ for $i = 1,........,N$ and Normalize $\bar{w}_0^{(i)} = \frac{w_0^{(i)}}{\sum_{j=1}^{N} w_0^{(j)}}$
2. Measurement update t> t₀	Calculate $w_t^{(i)} = p(y_t \setminus x_t^{(i)}) w_{t-1}^{(i)}$ for $i = 1,........,N$ and Normalize $\bar{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^{N} w_t^{(j)}}$
3. Re-sampling for each t> t₀	(a) Take N samples with replacement from $\{x_t^{(i)}\}_{i=1}^{N}$ where the probability to take sample i is $w_t^{(i)} = \frac{1}{N}$. This step is called Sampling Importance Re-sampling (SIR) or Bayesian bootstrap. (b) Only re-sample when $N_{eff} = \frac{1}{\sum_i (\bar{w}_t^{(i)})^2} < N_{th}$ and set $w_t^{(i)} = \frac{1}{N}$

4. Prediction or estimation	$\hat{P}(X_t \setminus Y_t) = \sum_{i=1}^{N} \bar{w}_t^{(i)} \ \delta(X_t - X_t^{(i)})$ $\bar{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^{N} w_t^{(j)}}$
5. Iteration	Let t=t+1 and iterate to process #2

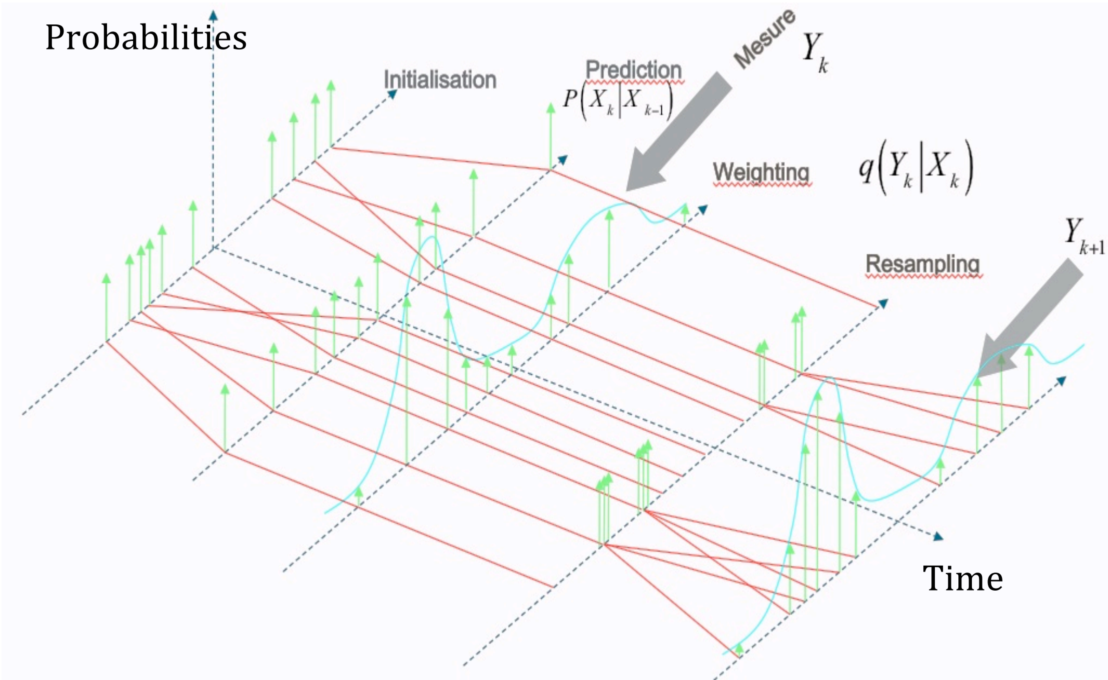


Figure4.2: Particle Filter Process

After this remainder (figure 4.2) of some generalities about the particle filter based on the Monte Carlo methods. This approach allows us to construct recursively a cloud of particle weighted approaching the filter law. However, the recursive nature of the filter leads to a degeneration of the cloud induced by an augmentation through time of the unconditional variance of its weights. In order to control this degeneration of the particle clouds the resampling process allows to limit this aspect by giving more importance to the significant weights. Our objective in this dissertation is not to make the basic algorithm of particle filter better, but to find out the best adaptive one for our case of study in this thesis work. More details about the changes done will be shown in the next sections of this chapter.

After this discussion we will describe some of the known distribution used in the field.

a. Normal distribution

The most used probability distribution to describe a physical situation is the normal distribution. The reason behind this statement is that most of the outputs from many processes are normally distributed.

All normal distributions have the same general shape. However, they can differ in their mean value and their variation, regarding the situation studied.
The normal distribution can be described mathematically as follows (equation 4.13):

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \qquad (4.13)$$

Where,

- x is any value of the random variable,
- σ is the population standard deviation,
- μ is the population mean.

And by influencing the mean and the standard deviation we can have different representation of the probability distribution, however, the total area (probability) under each normal curve equals 1.

b. Lognormal distribution

The log-normal distribution is a skewed distribution, which starts zero, rises to a maximum before falling more slowly to infinity (figure 4.3). It is connected to normal distribution: X is a log-normal distribution if ln(X) is a normal distribution.

The normal distribution can be described mathematically as follows (equation 4.14 and 4.15):

$$f(x) = \frac{1}{x\sigma_l\sqrt{2\pi}} e^{\frac{-(\ln x-\mu_l)^2}{2\sigma_l^2}} \qquad (4.14)$$

And the parameters required to specify the function are: μ_l the average natural log transformation of the input data, and the variance σ_l^2 of the natural-log transformation of the input data. They are defined as follows:

$$\mu_l = \ln \frac{\mu^2}{\sqrt{(\sigma^2 + \mu^2)}}$$
$$\sigma_l = \sqrt{\ln(\frac{\sigma^2}{\mu^2} + 1)} \qquad (4.15)$$

Where,

- x is any value of the random variable,
- σ is the population standard deviation,
- μ is the population mean.

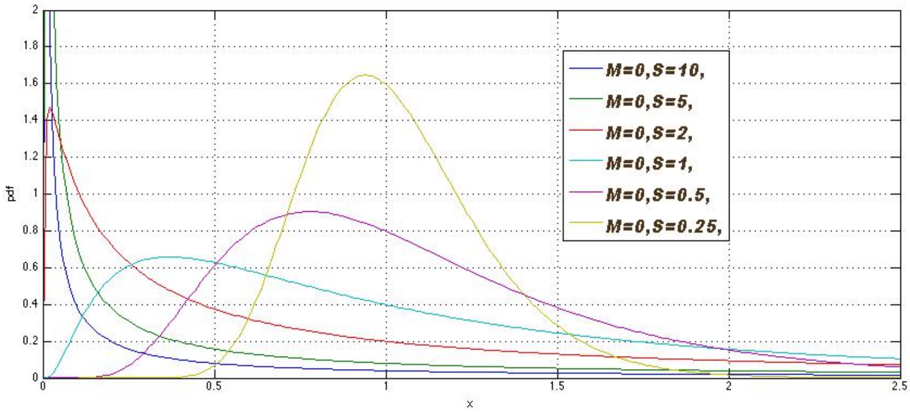


Figure 4.3: Several lognormal distributions with different mean and variance

Galton and McAlister 1897 did the first article in the literature where they initiated the study of the distribution in order to use the geometric mean as an estimation of location. It seams that this distribution give interesting results for example in [107] the author used this distribution to estimate the path/location and travel time using sparsely sampled GPS data.

4.4 Travel time parameters Distributions

In the previous section we skimmed through the particle sampling process where we choose a distribution for our particles. Which is an important process in the Monte Carlo approaches because it will affect the output of the particle filter. Regarding our work in this dissertation to estimate the travel time per road section is touching three aspects location or distance, time, and speed. Based on this three aspect we will discuss and try to define their distributions.

4.4.1 Speed distribution on a road section

Making a good estimation of travel time need a good understanding of traffic dynamics. Especially in our case in this dissertation we will use an adaptive particle filter where vehicle's speed is one of the parameters that we will affect with particles approach. Thus defining the closed probability density distribution of speed is needed in order to give a good estimation of travel time. Our interest in this section is to define the speed distribution model per road section in an urban context.

We have to take into consideration that the driver selects the speed. Each driver has his/her own way of driving. Some of them respect the speed limits others not. Besides, there are other factors involved that can push the driver to not respect the regulations such as vehicle's capacity and power, road condition, driver ability and so on.

The speed distribution can provide answer to the issue discussed above. Where the distribution will show all the arrangement of speed values showing their observed or theoretical frequency of occurrence. The research done in this subject has shown that the speed distribution is normally distributed; however, the properties or the characteristics of the probability distribution function (pdf) on each urban road section can vary depending on the driver behavior (Figure 4.4).

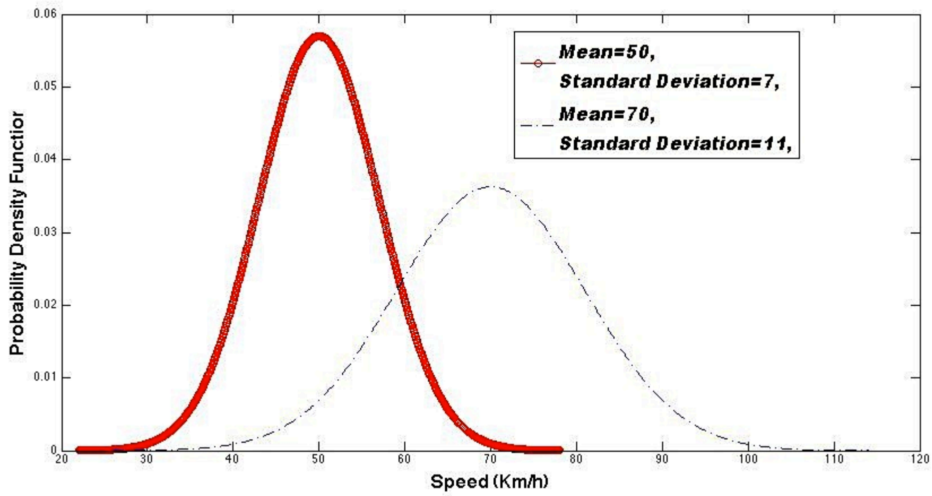


Figure 4.4: Speed Normal Probability Distribution

This kind of distribution allows us to characterize the speed using two features the mean and the standard deviation. Another aspect of the distribution is when we have the same mean but different standard deviations (Figure 4.5). This issues discussed will be check in chapter 5.

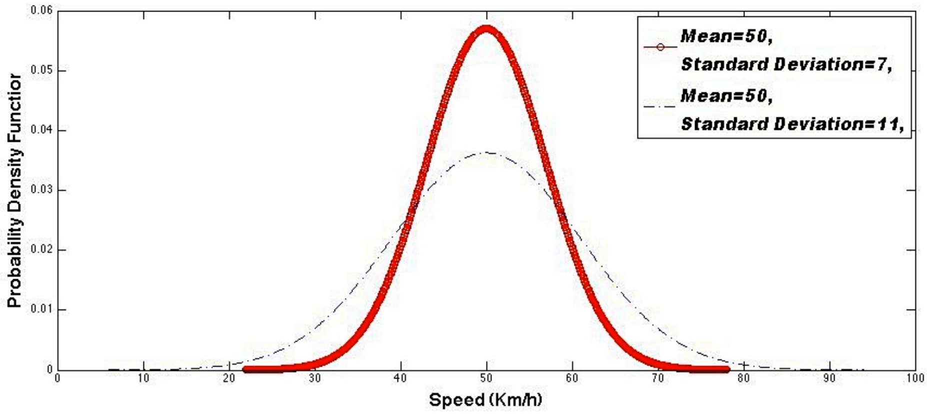


Figure 4.5: Example of two pdf with same mean and different standard deviations

These kinds of speed can be clearly noticed when we try to find out the speed profile in the road network (more details in chapter 5). The speed profiles are a graphical representation of speed features plotted by location. Besides, the use of speed profiles can help to analyze and evaluate the speed attitudes. The purpose behind this enlightenment is to show: how hard to estimate the right speed distribution that reflects the reality. As the Figure 4.6 shows that the driving behavior can be really complicated because of the people's interaction when using their vehicles between themselves and the use of speed on the urban road. Moreover, based on the figure 4.6 we can notice that the speed displacement of the male's speed graph cover in uniform way a large interval.

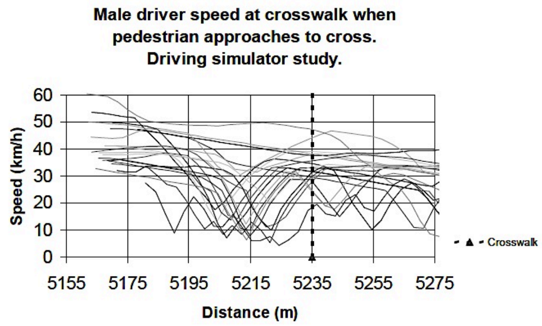


Figure 4.6: Speed Profile [Aronson 06]

After this brief discussion about the speed distribution it is clear that it is not easy to come out with a perfect distribution reflecting the reality but we will try to approach this reality. In order to determine the shape of the speed distribution, a common method to use is the histogram. Although the histogram method is widely used for data interpretation in many areas, it should be commented that the histogram method has a potential limitations because this method strongly depends on data origin, bin interval, and bin width [118]. This issue point to the fact that the histogram should be well constructed. Besides, the histogram is very good to use to get brief information about the speed distribution and there are some cases when we don't have this issue of discontinuity of data.

In the case when the characteristics of speed data are homogeneous which is the case for expressway in the urban areas then the speed probability density distribution can be conventionally described as normal distribution. However, the speed in urban networks can follow a bimodal distribution or polymodal distribution in this case there is no specific distribution function available [119]. However, it seems from figure 4.6 that the appropriate distribution of the speed is a bimodal distribution, will conduct an analysis in chapter 5 based on the observed speed data from historical data

Thus the appropriate way to estimate density function of the bimodal or polymodal distributions is the mixture model. The mixture model tries to cluster data into specific groups so that the data in the group have more similarity between each other [120]. We can represent the mixture model by using any kind of probability density distribution. In general the most used distribution is the Gaussian mixture model due to its simplicity of the estimation process; moreover, it is used for computational, mathematics, and optimization operations [121].

The Gaussian mixture model can be defined in general as follows [121]:

$$p(x \setminus \mu_1,\ldots,\mu_P,s_1,\ldots,s_P,\tau_1,\ldots,\tau_P)=\sum_{i=1}^{P}\tau_i N(\mu_i,s_i^{-1}) \qquad (4.16)$$

Where τ_i ($\tau_i \geq 0$) is a mixing proportion and each Gaussian distribution $N(\mu_i,s_i^{-1})$ has its own mean μ_i and s_i^{-1} is the inverse variance which refers to the precision. For example in the case of bimodal model (figure 4.7) the value of P will be equal to 2. Besides, the bimodal distribution has two mixture components (τ_1 and τ_2); the condition on these proportions is the sum is equal to one $\sum_{i=1}^{P}\tau_i = 1$ [119], [121].

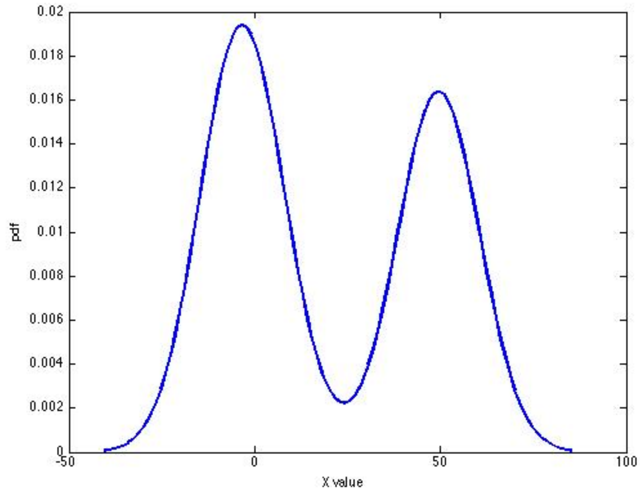


Figure 4.7: Example of Gaussian Bimodal Distribution

In the urban network due to the traffic jam phenomena a single distribution cannot be appropriate to the situation. Thus the speed distribution needs two separate regimes as a mixture of two different Gaussian or normal density distribution. From this point, the first mixture component representing the low-speed regime can be formulated as $N(\mu_1, s_1^{-1})$ and the second mixture component showing the high-speed regime can be represented by $N(\mu_2, s_2^{-1})$. As consequence, the speed Gaussian mixture model in this case will be as follows:

$$p_v(v \setminus \mu_1, \mu_2, s_1, s_2, \tau_1, \tau_2) = \tau_1 N(\mu_1, s_1^{-1}) + \tau_2 N(\mu_2, s_2^{-1}) \quad (4.17)$$

By using $\sum_{i=1}^{P} \tau_i = 1$ the equation 4.17 become:

$$p_v(v \setminus \mu_1, \mu_2, s_1, s_2, \tau) = (1-\tau) N(\mu_1, s_1^{-1}) + \tau N(\mu_2, s_2^{-1}) \quad (4.18)$$

Where $\tau = \tau_2$

Moreover we will put the hypothesis that the speed distribution parameters will be define by two speeds of low-speed and high-speed regime as follows:

1. The low-speed regime is when we have a heavy traffic which mean either we have traffic jam situation or stop and go situation due to light traffic or road intersection or maybe just a dense traffic situation. The value of this kind of speed will be defined based on the study done on each road section where we find the min speed used on the concerned road section. We will note it as $v_{\min}$.

2. The high-speed regime v_{max} is when we have fluid traffic no traffic jam. In order to define the value of that speed we will use our data to define the 85th percentile speed.

The hypothesis will be checked during the test phase in chapter 5.

Definition 22: (85th percentile speed) is the speed at or below which 85 percent of vehicle travel in a road section.

4.4.2 Locations distribution on a road section

The probability density function of a vehicle's displacement on a road sections is a linear combination of normal distribution based on [122]. This aspect is due to the uncertainty of the traffic on the road section or to the different behaviors of the drivers (figure 4.8).

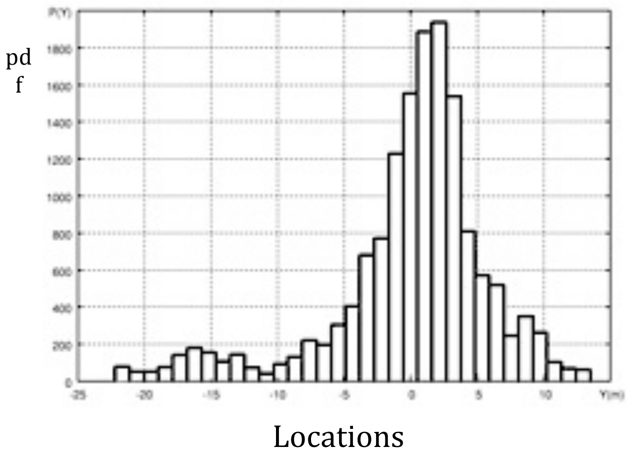


Figure 4.8: the pdf of GPS locations [122]

Based on this analysis, it is clear that most of the cases the vehicle locations x follows a normal distribution on a road section. Moreover, the vehicle locations distribution is related to the speed evolution during the period Δt when the vehicle passes by the road section. Thus the locations distribution is a normal distribution $p(x)$ (as defined in 4.3).

4.4.3 Travel Time distribution on a road section

For the travel time probability distribution we will use the fact that travel time is related to the speed on the road section. Which means that the probability distribution of travel time is proportional to the speed probability distribution on the road section (equation 4.19).

$$p_x(t) \propto p_v(v) \quad and \quad v = \frac{x}{t}$$

$$Thus \quad p_x(t) \propto p_v(\frac{x}{v}) \tag{4.19}$$

by using the fact that the travel time distribution is proportional to the speed distribution (defined in 4.4.2, equation 4.18) we have the following results:

$$p_x(t) = (1-\tau)N(\frac{x}{v_{\min}}, \mu_1, s_1^{-1}) + N(\frac{x}{v_{\max}}, \mu_2, s_2^{-1}) \tag{4.20}$$

Where the $N(\frac{x}{v}, \mu_1, s_1^{-1})$ will be defined as follows (equation 4.21):

$$N(\frac{x}{v}, \mu_1, s_1^{-1}) = \frac{1}{(s^{-1})^2 \sqrt{2\pi}} \exp(\frac{-\mu(v - \frac{x}{\mu})}{2v^2 (s^{-1})^2}) \tag{4.21}$$

and, $v_{\min}$ and $v_{\max}$ are respectively low and high-speed regime when the traffic status on the road section is congested or fluid.

4.5 Travel time estimation using adaptive Monte Carlo approach

In this section we will expose the process adopted in order to create our adaptive method to estimate travel time [129]. As a remainder, our historical data has special features. The data used is a sparsely sampled GPS data where an interval of one minute between each successive GPS data point [130]. Therefore, the data has a frequency of one minute between each GPS point which is not the case for normal GPS data received from a normal GPS receiver where data is received each second (illustration figure 4.9).

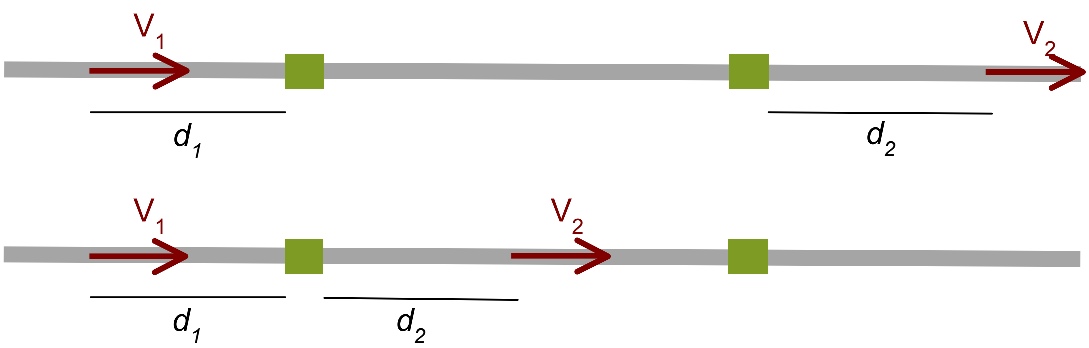


Figure 4.9:Illustration of Sparsely Sampled Data on the Map (green squares are the PUMAS points)

The fact that our estimation is only related to the travel time we can assume that our estimation process can be associated with the state equation as follows:

$$S_p = S_k + V_k * (t_p - t_k) + U \qquad (4.22)$$

Where Sp and Sk refer respectively to GPS coordinate of the PUMAS point and the first GPS data received, Vk is the speed of the car at the moment of receiving the GPS coordinates, tp and tk are respectively the time when the car was at the PUMAS point position and when we receive the GPS data, and finally U is a white state noise following a normal distribution that we add to the equation in order to represent to the imprecision of our parameters. Then our objective is to estimate tp :

$$t_p = \frac{S_p - U - S_k}{V_k} + t_k \qquad (4.23)$$

In our case we will process the unknown parameters of the equation by applying a defined model are known. Thus, we will inject our data directly by applying the particle process for each unknown parameters in order to simulate the uncertainty of our data.

As a start, we will apply particles based on a normal distribution (Gaussian distribution) for which the mean is the value of the parameter and the total area under the normal curve is equal to 1. In our case, we choose a probability density with a fixed standard deviation σ that will make the particles evolve during the resampling stage following the equation below:

$$P(Y_t / X_t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(Y_p - Y^*_{X_t/X_{t-1}})^2}{2\sigma^2}\right) \qquad (4.24)$$

Where Y_p and $Y^*_{X_t/X_{t-1}}$ are respectively the observed positions of the PUMAS points and the observed positions of the data on the map.

Thus our global particles equation will be as follows:

$$S_p = S_k^{(i)} + V_k^{(i)} * (t_p^{(i)} - t_k) + U^{(i)} \qquad (4.25)$$

Therefore equation (4.25) becomes:

$$t_p^{(i)} = \frac{S_p - U^{(i)} - S_k^{(i)}}{V_k^{(i)}} + t_k \qquad (4.26)$$

Now we apply this formula to the first GPS data received and we call the estimated time of passage tp forward (tpf). Then we do the same for the second GPS data received after one minute and we call estimated time of passage tp backward (tpb) (figure 4.10).

$$t_{pf}^{(i)} = \frac{S_p - U^{(i)} - S_k^{(i)}}{V_k^{(i)}} + t_k \quad (4.27)$$

$$t_{pb}^{(i)} = \frac{S_p - U^{(i)} - S_{k+1}^{(i)}}{V_{k+1}^{(i)}} + t_{k+1} \quad (4.28)$$

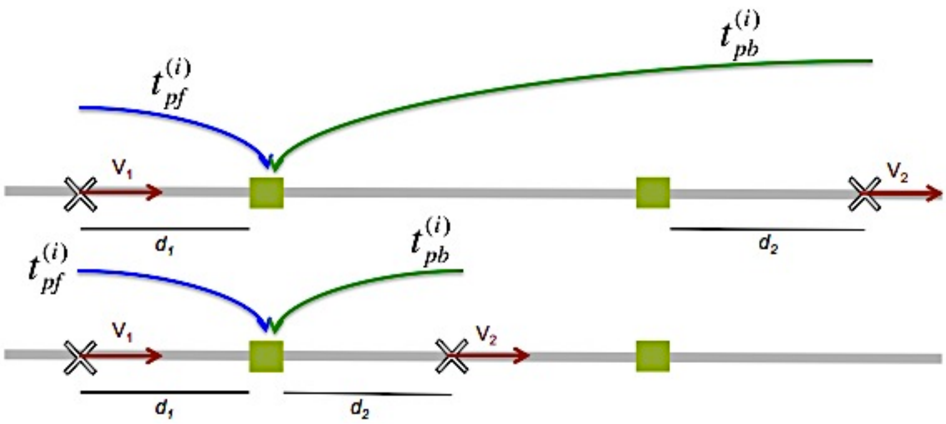


Figure 4.10:tp-forward and tp-backward illustration

Moreover we attribute a weight to each estimated tpf and tpb. The weight function was defined in such a way to take into account the accuracy of the estimation by using a time distance notion. These equations are:

$$W_1^{(i)} = \frac{t_{pf}^{(i)} - t_k}{t_{k+1} - t_k} \quad (4.29)$$

$$W_2^{(i)} = \frac{t_{k+1} - t_{pb}^{(i)}}{t_{k+1} - t_k} \quad (4.30)$$

Where $W_1^{(i)}$ and $W_2^{(i)}$ are the weight of tpf and tpb respectively. Finally the estimation is computed as follows:

$$t_p = \tfrac{1}{2} \sum_{i=1}^{N} (W_1^{(i)} * t_{pf}^{(i)} + W_2^{(i)} * t_{pb}^{(i)}) \quad (4.31)$$

The algorithm will help us to estimate at what time the vehicle passed by the PUMAS points tpi (algorithm 1, figure 4.11) and then we can compute the travel time per PUMAS sections Tpj where j and i refer to the sections and PUMAS points respectively.

$$T_{P_j} = t_{p_{i+1}} - t_{p_i} \qquad (4.32)$$

After obtaining our estimation we add the estimated delay time on the PUMAS zone due to traffic light duration as mentioned in [105] when we detect a crossroad or the presence of traffic light in the itinerary of the vehicle; we can compute it using the following equation:

$$D = \frac{L}{V} * S_{out} \qquad (4.33)$$

Where L is length of the PUMAS section, V is the limit speed in the PUMAS section concerned, and Sout is the number of vehicles that leave the PUMAS section during a given period of time [t, t+ Δ t [.

The filter estimator was created in such a way as to be adapted to our special historical database and in the next section we will explain our experimental process and then we will show our results.

At time to.
Initialization.
Define an initial state.
Generate N particles for ($S_k, S_{k+1}, V_k, V_{k+1}, U$) Following p(S_k),p(S_{k+1}),p(V_k),p(V_{k+1}), p(U).

Put $W^{(i)} = \frac{1}{N}$ Sampling Importance Resampling (SIR).

For j=1 to M do

Update the particles ($S_k^{(i)}, S_{k+1}^{(i)}, V_k^{(i)}, V_{k+1}^{(i)}, U^{(i)}$)

For i=1 to N do

Compute $t_{pf}^{(i)}$ and $t_{pb}^{(i)}$.

Compute the weight $W_1^{(i)}$ and $W_2^{(i)}$.

Normalize the weights $W_1^{(i)} = \frac{W_1^{(i)}}{\sum_{i=1}^{N} W_1^{(i)}}$ and $W_2^{(i)} = \frac{W_2^{(i)}}{\sum_{i=1}^{N} W_2^{(i)}}$.

End.

Compute $N_{eff1} = \frac{1}{\sum_{i=1}^{N} (W_1^{(i)})^2}$ and $N_{eff2} = \frac{1}{\sum_{i=1}^{N} (W_2^{(i)})^2}$.

If $N_{eff1} \geq N_{th}$ (threshold) then

Resampling $\left\{ t_{pf}^{(i)}, W_1^{(i)} \right\}_{i=1}^{N}$.

Update the weights $W_1^{(i)}$.

Else If $N_{eff2} \geq N_{th}$ (threshold) then

Resampling $\left\{ t_{pb}^{(i)}, W_2^{(i)} \right\}_{i=1}^{N}$.

Update the weights $W_2^{(i)}$.

End If.

End If

Compute the estimation: $t_p = \frac{1}{2} \sum_{i=1}^{N} (W_1^{(i)} * t_{pf}^{(i)} + W_2^{(i)} * t_{pb}^{(i)})$.

End.

Figure 4.11: Estimation filter algorithm1 (N: Number of particles, M: Number of iterations) [129]

4.6 Travel time estimation using Monte Carlo method enhanced with measurements and road sections characteristics

In this part we will show how we made an enhancement to the previous algorithm. Some of the steps are similar but with some changes in the process and the equations used.

We keep the same state equation 4.22 as it was stated before with the same notations and meaning:

$$S_p = S_k + V_k * (t_p - t_k) + U \qquad (4.22)$$

then after applying the particle with the same characteritics as it was shown before. Which means the particles follow a normal distribution (Gaussian distribution) for which the mean is the value of the parameter and the total area under the normal curve is equal to 1. In our case, we choose a probability density with a fixed standard deviation σ as before equation 4.25

$$P(Y_t / X_t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(Y_p - Y^*_{X_t/X_{t-1}})^2}{2\sigma^2}\right) \qquad (4.24)$$

Therefore, the expression with particles (i) will be as before equation 4.27:

$$t_p^{(i)} = \frac{S_p - U^{(i)} - S_k^{(i)}}{V_k^{(i)}} + t_k \qquad (4.26)$$

after this step we apply as before this formula to the first GPS data received and we call the estimated time of passage tp forward (tpf). Then we do the same for the second GPS data received after one minute and we call estimated time of passage tp backward (tpb).

$$t_{pf}^{(i)} = \frac{S_p - U^{(i)} - S_k^{(i)}}{V_k^{(i)}} + t_k \qquad (4.27)$$

$$t_{pb}^{(i)} = \frac{S_p - U^{(i)} - S_{k+1}^{(i)}}{V_{k+1}^{(i)}} + t_{k+1} \qquad (4.28)$$

however this time we will detect all the PUMAS points between the two GPS data. Let's note j the number of PUMAS points detected. Thus the formula (equations 4.27 and 4.28) will become as follows:

$$\begin{aligned} t_{p_j f}^{(i)} &= \frac{S_{p_j} - U^{(i)} - S_k^{(i)}}{V_k^{(i)}} + t_k \\ t_{p_j b}^{(i)} &= \frac{S_{p_j} - U^{(i)} - S_{k+1}^{(i)}}{V_{k+1}^{(i)}} + t_{k+1} \end{aligned} \quad for \quad j = \{1,2,3,....,P\} \qquad (4.34)$$

As a concequence, the weight expression based on time distance will become (equation 4.35):

$$W_{1,j}^{(i)} = \frac{t_{p_j f}^{(i)} - t_k}{t_{k+1} - t_k} \quad for \quad j = \{1,2,3,...,P\}$$
$$W_{2,j}^{(i)} = \frac{t_{k+1} - t_{p_j b}^{(i)}}{t_{k+1} - t_k} \tag{4.35}$$

the next step will be defining the new weight where the road sections characteristics based on measurements introduced in the weight of the particles equations.

Based on the study done before on this chapter, we know the probability distribution of travel time of each road section on our network (section 4.4.4). Thus the expression will be as follows (equation 4.36):

$$\omega_1^{(i)} = \sum_{j=1}^{P} W_{1,j}^{(i)} P_x(t_{p_j f}^{(i)})$$
$$\omega_2^{(i)} = \sum_{j=1}^{P} W_{2,j}^{(i)} P_x(t_{p_j b}^{(i)}) \tag{4.36}$$

Where,

$$p_x(t) = (1-\tau)N(\frac{x}{v_{\min}}, \mu_1, s_1^{-1}) + \tau N(\frac{x}{v_{\max}}, \mu_2, s_2^{-1}) \tag{4.37}$$

Finally the estimation is computed as before using the formula 4.38:

$$t_{p_j} = \tfrac{1}{2} \sum_{i=1}^{N} (\omega_1^{(i)} t_{p_j f}^{(i)} + \omega_2^{(i)} t_{p_j b}^{(i)}) \quad for \quad j = \{1,2,3,....,P\} \tag{4.38}$$

now we have the estimation of the moment of passage on each PUMAS point detected on the path and we can conclude with the travel time estimation per road section using the equation stated before (equation 4.32) and we add also the notion delay at intersections using equation 4.33.

At time to.
Initialization.
Define an initial state.
Generate N particles for ($S_k, S_{k+1}, V_k, V_{k+1}, U$) Following p(S_k),p(S_{k+1}),p(V_k),p(V_{k+1}), p(U).

Put $W^{(i)} = \frac{1}{N}$ Sampling Importance Resampling (SIR).

For l=1 to M do

Update the particles ($S_k^{(i)}, S_{k+1}^{(i)}, V_k^{(i)}, V_{k+1}^{(i)}, U^{(i)}$)

For j=1 to P do

For i=1 to N do

Compute $t_{p_j f}^{(i)}$ and $t_{p_j b}^{(i)}$.

Compute the weight $W_{1,j}^{(i)}$ and $W_{2,j}^{(i)}$.

Normalize the weights $W_{1,j}^{(i)} = \frac{W_{1,j}^{(i)}}{\sum_{i=1}^{N} W_{1,j}^{(i)}}$ and $W_{2,j}^{(i)} = \frac{W_{2,j}^{(i)}}{\sum_{i=1}^{N} W_{2,j}^{(i)}}$.

End.

End.

Generate $P_x(t_{p_j f}^{(i)})$ and $P_x(t_{p_j b}^{(i)})$.

Compute $\omega_1^{(i)}$ and $\omega_2^{(i)}$.

Normalize the weights $\omega_1^{(i)} = \frac{\omega_1^{(i)}}{\sum_{i=1}^{N} \omega_1^{(i)}}$ and $\omega_2^{(i)} = \frac{\omega_2^{(i)}}{\sum_{i=1}^{N} \omega_2^{(i)}}$.

Compute $N_{eff\,1} = \frac{1}{\sum_{i=1}^{N} (\omega_1^{(i)})^2}$ and $N_{eff\,2} = \frac{1}{\sum_{i=1}^{N} (\omega_2^{(i)})^2}$.

If $N_{eff\,1} \geq N_{th}$ (threshold) then

Resampling $\left\{ t_{p_j f}^{(i)}, \omega_1^{(i)} \right\}_{i=1}^{N}$.

Update the weights $\omega_1^{(i)}$.

Else If $N_{eff\,2} \geq N_{th}$ (threshold) then

Resampling $\left\{ t_{p_j b}^{(i)}, \omega_2^{(i)} \right\}_{i=1}^{N}$.

Update the weights $\omega_2^{(i)}$.

End If.

End If

Compute the estimation: $t_p = \frac{1}{2} \sum_{i=1}^{N} (\omega_1^{(i)} * t_{p_j f}^{(i)} + \omega_2^{(i)} * t_{p_j b}^{(i)})$.

End.

Figure 4.12: Estimation filter algorithm2 (M number of iterations, P number of PUMAS points, N number of particles) [132]

4.7 Conclusion

To sum up, first we stated our problem statement where we defined exactly the problematic that we are dealing with. Then we gave an overview of the study on the parameters characterizing the traffic and their distributions regarding a road section approach. The study of those parameters was used later on the enhancement of the method proposed to estimate the travel time per road section or road section. Next, we presented our approach to estimate the travel time using a Monte Carlo method. Moreover, the approach was adapted to our case of study. Finally, we showed how we made and enhancement to the proposed method using the measurements and characteristics of the road sections in urban road network [132]. The next chapter will show all the details regarding the implementation of the whole system (software) plus the results of all what was done in this research work.

Chapter 5: Implementation, Results and Analysis

5.1 Introduction

The following chapter will present the implementation of the system. Furthermore, it will show and discuss the results of the experiments done using real world data. We should remind that the data used in our historical database is sampled with a rate of 60 seconds between each successive GPS data.

The results presented will be about each proposed algorithm. First, we will discuss the results of the map matching which include:

- A spatial map matching,
- Updated spatial map matching,
- Spatio-temporal map matching
- A corrected spatio-temporal map matching.

The second point will be the verification of the stated hypothesis about the speed distribution. Finally, we will show the results of travel time estimation using basic Monte Carlo Method and enhanced one.

5.2 Building the Digital Map

In this section we will describe the implementation done in order to extract the digital map and add the GIS information. This latter refer to the PUMAS points, PUMAS sections, road type, road orientation, road section's speed limit, etc.

5.2.1 Description and Implementation

As we said before in chapter 3 that we used OpenStreetMap (OSM) in order to extract the XML file (Raw OSM Data) containing the information about the map. The figure 5.1 shows the way the information extracted are written. For example the node contains its id and to each way or section is connected and its coordinates. Besides, it will have also the information about the ways or section, their id, connection id, etc.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6" generator="CGImap 0.0.2">

 <bounds minlat="54.0889580" minlon="12.2487570" maxlat="54.0913900"
                          maxlon="12.2524800"/>


 <node id="298884269" lat="54.0901746" lon="12.2482632" user="SvenHRO"
                  uid="46882" visible="true" version="1" changeset="676636"
                  timestamp="2008-09-21T21:37:45Z"/>
 <node id="261728686" lat="54.0906309" lon="12.2441924" user="PikoWinter"
                  uid="36744" visible="true" version="1" changeset="323878"
                  timestamp="2008-05-03T13:39:23Z"/>
 <node id="298884272" lat="54.0901447" lon="12.2516513" user="SvenHRO"
                  uid="46882" visible="true" version="1" changeset="676636"
                  timestamp="2008-09-21T21:37:45Z"/>

<way id="11" user="Masch" uid="55988" visible="true" version="5"
                  changeset="4142606" timestamp="2010-03-16T11:47:08Z">
  <nd ref="22"/>
  <nd ref="33"/>
  <nd ref="44"/>
  <tag k="highway" v="primary"/>
  <tag k="name" v="bouhbouh"/>
  <tag k="oneway" v="-1"/>
  <tag k="maxspeed" v="90"/>
</way>
```

Figure 5.1: Example of OpenStreeetMap's XML file

The tool created using OSM-PgRouting allow us to import from the XML file all the tags that we need such as the node points and tag way, corresponding to the sections, the number of lanes on a section, the section is that a way or not (road section orientation) and the speed limit. These attributes are essential because later we used them in the map-matching process and the calculation of the shortest paths and calculation of travel time estimation per road section. Moreover, the calculation of the shortest path at the level of database context is done during the process of creating the map with its GIS information. The following figure 5.2 shows the classes implemented in order to do this task.

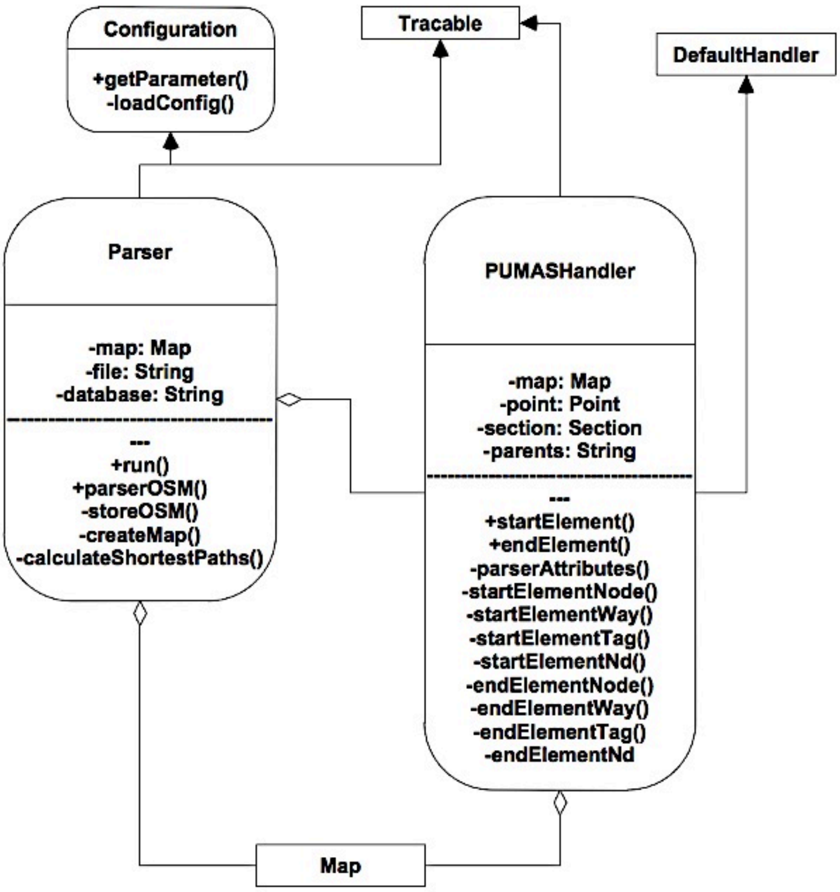


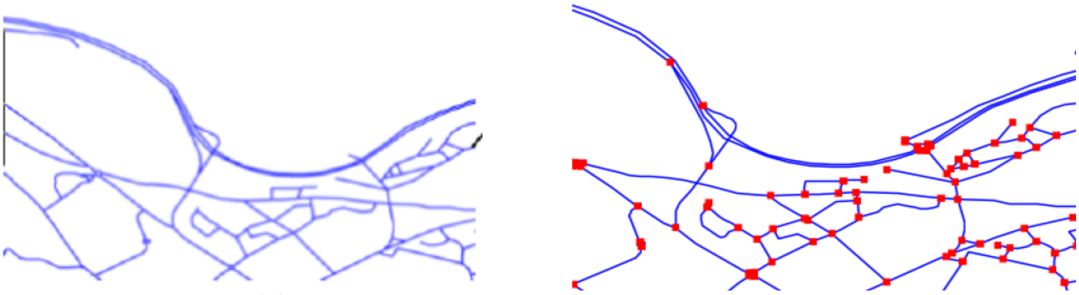
Figure 5.2: Class Diagram of the Building Digital Map Process

The parser class has a lot of functionalities. It reads the OSM file (XML) and recovers the data using the function “parseOSM”, stores the data in the database using the function “storeOSM” and transforms them into PUMAS Sections and Points PUMAS using the function “createMap”. Finally, it starts the calculation of shortest paths through the method “calculateShortestPaths”. Moreover, almost all the function inside the parser class uses some instances of the PUMASHandler class to handle some processes or events.

The PUMASHandler class it contains all the information and dictionary of the new GIS information. Additionally, it helps the building of the new map with all the information needed.

5.2.2 Results

The results are shown as map illustration (figure 5.3). By building, the digital map we have all the information stored in the database. We have the information regarding the exact position of PUMAS point, which refer to the start and end of each road section. In addition to the PUMAS sections where know the length of each section, the starting and the ending PUMAS points defining the section, and also the speed limit on each section. Finally, we have the type of the PUMAS section if it is expressway or an urban road, plus the orientation of the road (one way or two ways). The road network contains 7739 PUMAS points and 18874 PUMAS sections.



(a) (b)

Figure 5.3: (a) Extraction and reconstruction of the digital map, (b) is the new map with the PUMAS points in red and the PUMAS section linking the PUMAS sections

5.3 Map-Matching

As discussed in chapter 3, the navigations system cannot have good performances sometimes in positioning the vehicles on a digital map of road network due to the urban canyons, streets with dense tree cover, and tunnels. In this section we will test the algorithm presented in chapter 3 and analyze the results.

5.3.1 Description and Implementation

In chapter 3 we described in details our method. Figure 5.4 shows the top-down analysis diagram where the map-matching function uses or calls other function in its process.

For example, the function cap_threshold is the function that checks the condition on the speed of the sparsely sampled GPS data (speed condition details chapter 3 section 3.3.3.3.3). The functions are coded using SQL; therefore, they contain SQL queries. By receiving the raw data, the map_matching function sends an SQL query to start the mapping process with the information about the activation of the orientation_check process or not. Thus, the algorithm described in chapter 3 is running and processing the input data.

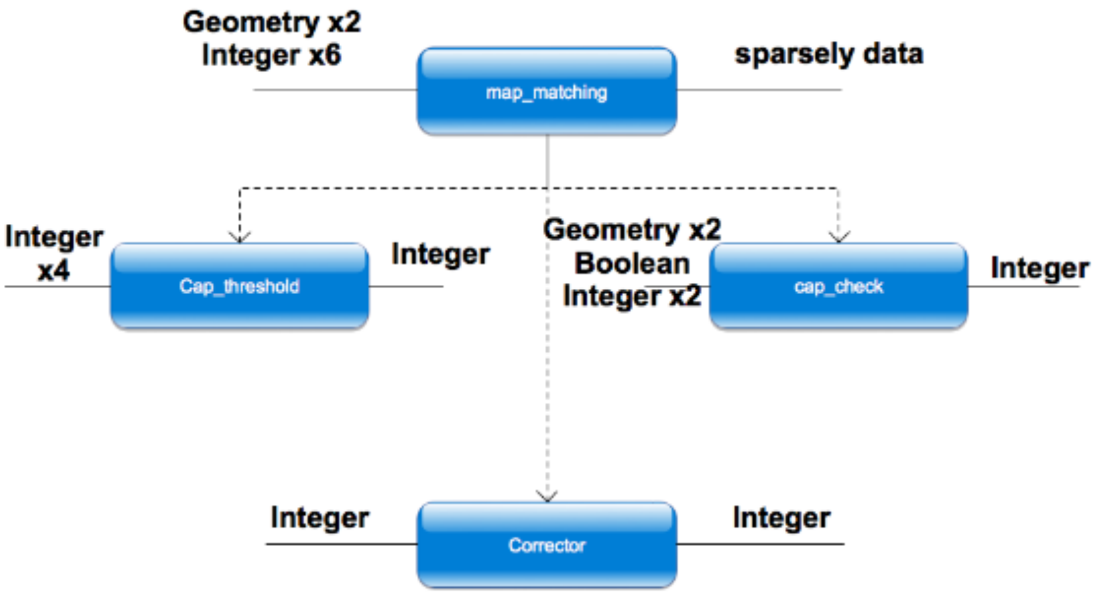


Figure5.4: Top-Down Analysis of the SQL map matching function

As we presented in chapter 3 during the process of map_matching we call the function that find out the shortest path between two sparsely sampled GPS data. In the previous section of this chapter we showed that the learning process of the finding shortest path is done when we are creating the digital map.

The function that the map_matching uses to call the shortest pathfinder is the TDSPP_Calculator. This latter uses the class graph, class vertex, and class edge (Figure 5.5). The class Edge represents the arcs of our graph. Then the class vertex is the node of our graph. Finally the class graph is set of vertices and edges constructing the whole graph.

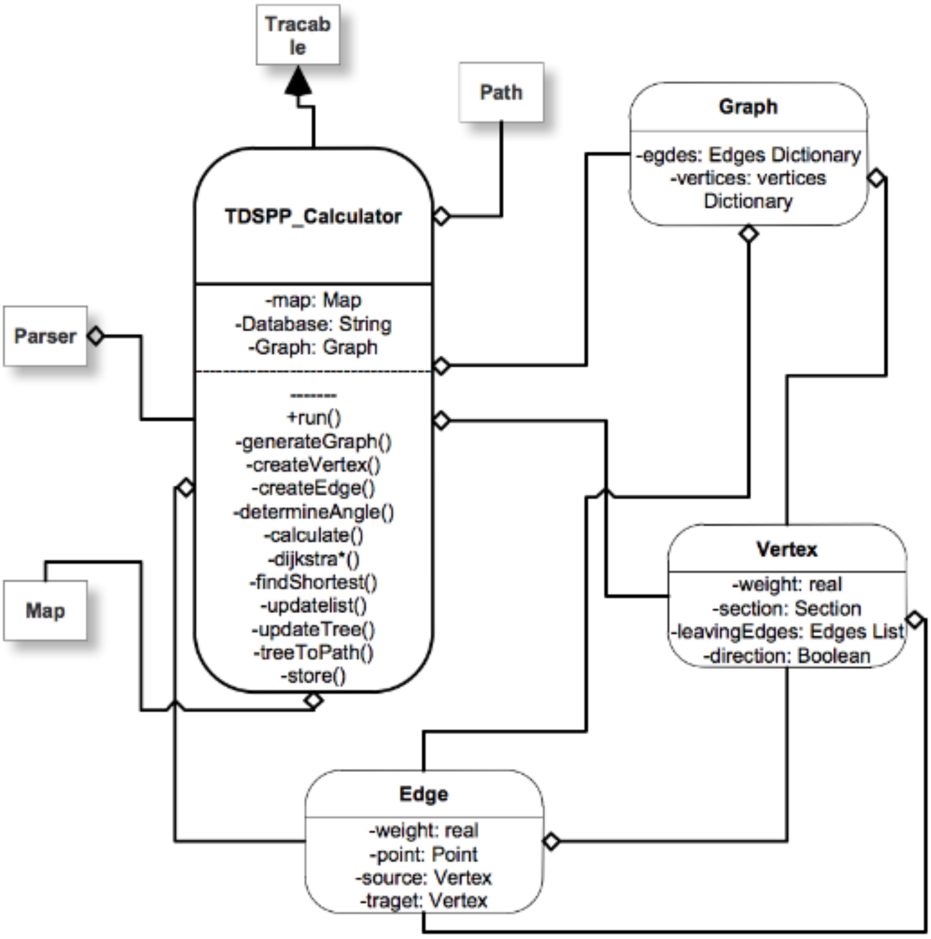


Figure 5.5: Class Diagram of the TDSPP_Calculator

5.3.2 Test Strategy

The test strategy describes the nature of data used and the manner how the test will be done. In addition, we will state the criteria chosen in order to make the evaluation of the results.

5.3.2.1 Dataset Description

In this part we will define the data used in order to run the test.

Road Network: In our experiment, we used the road network of Rouen city that we created in the previous section (figure 5.6). The network contains 7739 vertices and 18874 road sections. The figure 5.6 shows the created digital map that we use in our system. The gray segments are the road section and the one in color show an example of trajectory with colored quality information.

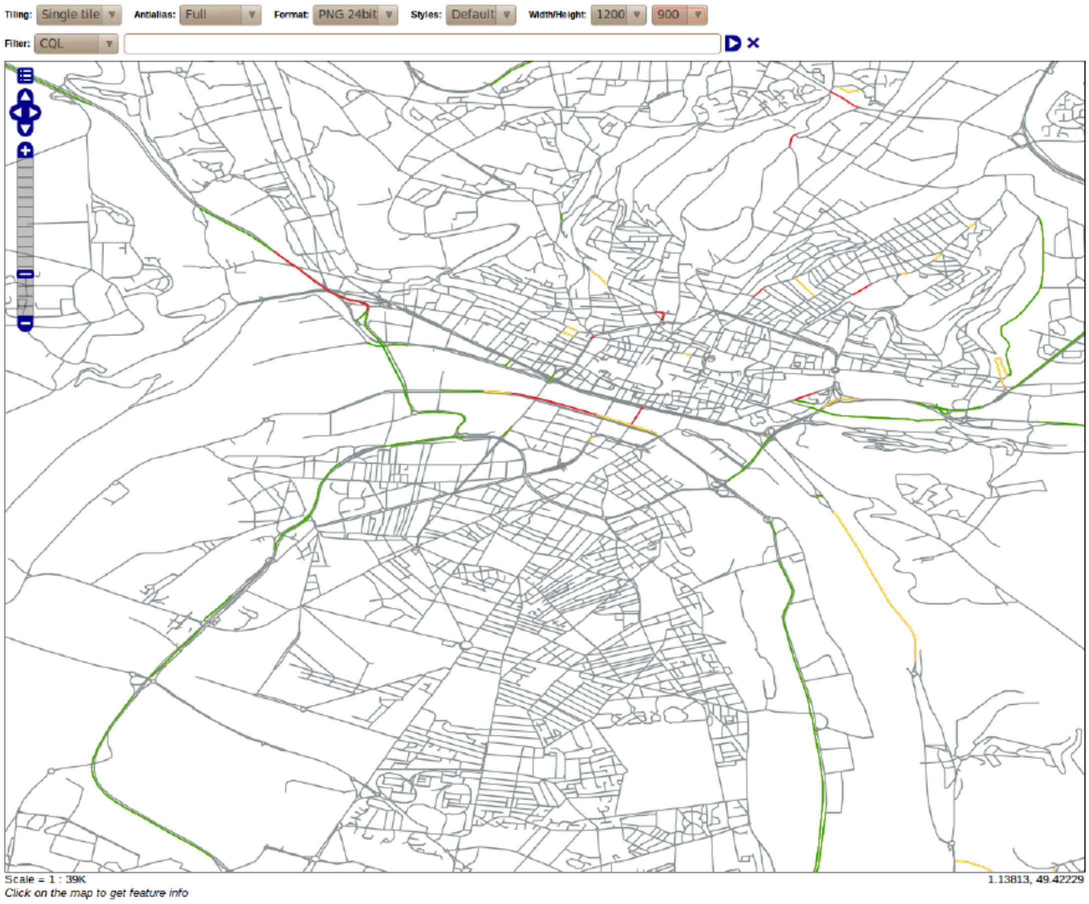


Figure 5.6: Road Network of Rouen

Test Data: the data was collected from the real world as it was described in chapter 2 section 2.5.5. We used those data that have GPS information with a frequency of one seconds. In order to make our tests we defined an experiments plan. In this plan we defined trajectories that will pass by the big axes and the small road in the city center.

This will allow us to compare the output of the algorithms and the ground truth because we already know the exact trajectories.

5.3.2.2 Evaluation Approach

Ground Truth: the data collected from the real world field constitute the ground truth. Moreover, the trajectories are known and defined before doing the test. Thus all the data collected is from known trajectories and road sections. The data that we will use in this test is the GPS data with a frequency of one second collected from the field. Then we will use it to simulate our database sparsely sampled GPS data with a frequency of 60 seconds. These simulated data will be the input of our map-matching system.

Evaluation Criteria: in order to evaluate our map-matching approach (STC-Matching). We will check the running time and the matching quality. The running time is measured by the actual program execution time. The matching quality is measured using the accuracy metric defined as follows:

$$A_{num} = \frac{Number \quad data \quad correctly \quad matched}{Total \quad number \quad of \quad data} \qquad (5.1)$$

Where, *number data correctly matched* is the number of data matched to the right road section. The *Total number of data (n)* is the total number of data used in the test, which are all the available GPS positions.

Baselines: during the presentation of the results and the analysis, we chose to show the evolution of the idea that we had during working on this issue of map matching, till reaching the final method presented in the chapter 3. First approach that we adopted to deal with the problem was the spatial analysis approach (S-Matching). Then we added the temporal analysis to the method (ST-Matching). Next step was to take in to account the heading of the vehicle and the road orientation, then make correction of the raw data that was rejected during the process.

5.3.3 Experimental Results

5.3.3.1 S-Matching

5.3.3.1.1 Results

First test result that we are going to show is the accuracy results of the S-Matching approach. Figure 5.7 shows that we have 60% of the data were matched correctly to their concerned road sections. And 14% of the data was matched but to the wrong road sections. Finally 26% of the data was not matched at all.

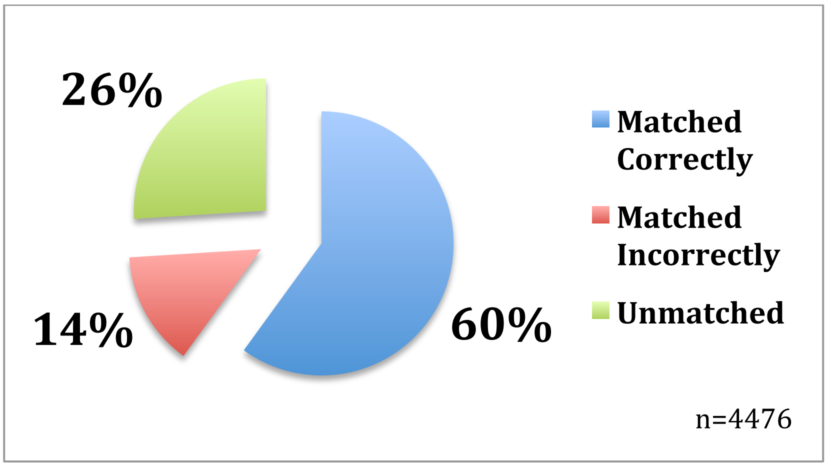


Figure 5.7: S-Matching Results

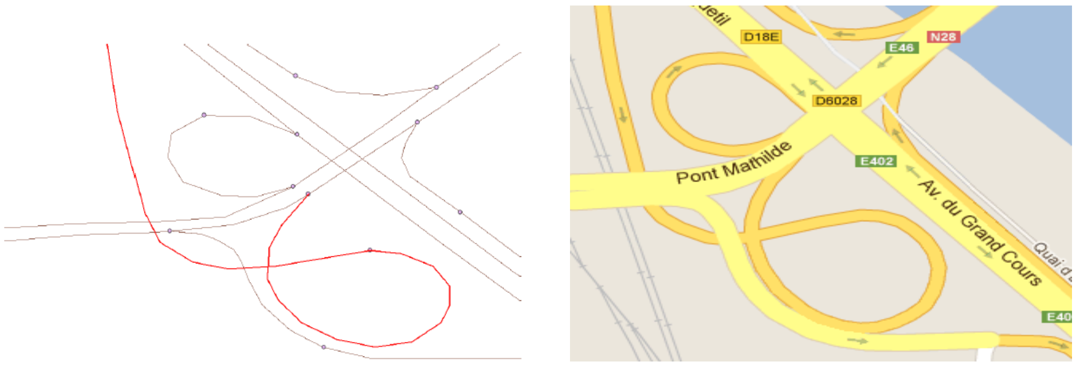
5.3.3.1.2 Analysis

When we check the results closely we find out that there is some problem with the digital map. Which explains the unmatched results. The problems that we found is some missing nodes and sections on the digital map that they don't exit in the extracted XML file (figure 5.8).

(a) (b)

Figure 5.8: Case of missing node in the OSM map, (white node exist, black node does not exist) ((a) digital map of the system, (b) Google map 2012)

Another problem detected with the digital map in some case we have a bad fusion of OpenStreetMap road sections, which affect the creation of the PUMAS Sections in the digital map used (figure 5.9). For example in figure 5.9 we can see the creation of two road section in the circled road; however, in reality it is only one road section.



(a) (b)

Figure 5.9: Example of bad fusion between OSM data and creation of PUMAS sections ((a) digital map of the system, (b) Google map 2012)

Thus we made an update of the digital map in order to fix these issues by adding the missing nodes and fixing this fusion problem in the creation of the PUMAS sections.

5.3.3.1.3 S-Matching after The Map Update (Sup-Matching)

Figure 5.10 show that we made an enhancement of the results. We have 65% of the data were correctly matched to the right sections and 19% of the data was unmatched. Then the matched incorrectly has increased, now it is 16%. Which mean that we saved some of the rejected data.
In the next section we will show the results when we add the temporal analysis to the algorithm.

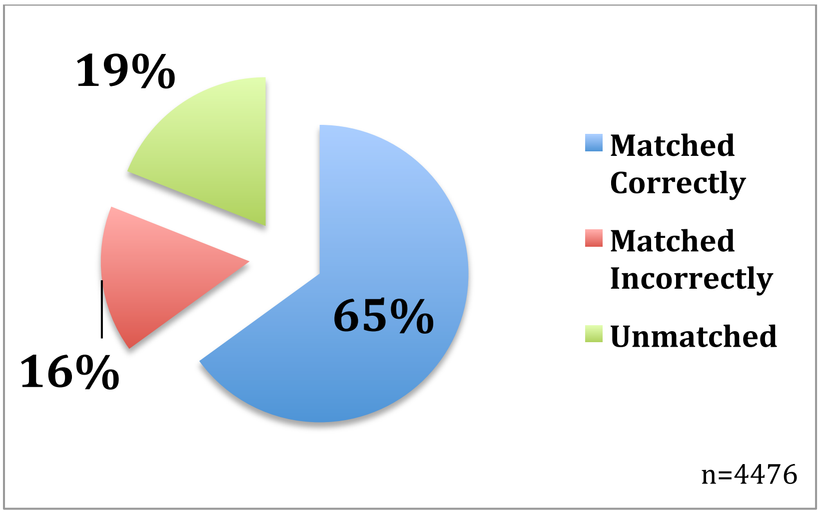


Figure 5.10: S-Matching results after updating the map

5.3.3.2 ST-Matching

5.3.3.2.1 Results

The next result is the results of the spatio-temporal approach (ST-Matching approach). Figure 5.11 show that we have 89% of the data were matched correctly to the concerned road sections. And 3% of the data was matched but to the wrong road sections. Finally 9% of the data was unmatched at all.

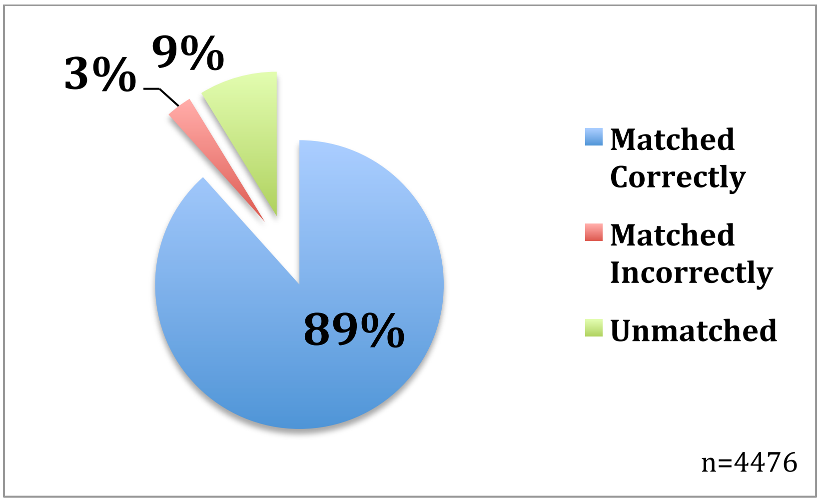


Figure 5.11: ST-Matching results

5.3.3.2.2 Analysis

The temporal aspect add to the algorithm allow us to take into consideration the real path logic in the matching process. Thus we take into consideration trajectory logic between the sparsely sampled GPS data.

However, we noticed that some of the data is matched to the wrong section. This problem is due to the fact the road orientation is in the opposite direction compared to the vehicle's heading received from the probes. The figure 5.12 illustrates clearly this problem related to the road orientation and the vehicle's heading.


Figure 5.12: Example of the orientation problem

Another problem that we detected due to the road section orientation is illustrated in figure 5.13. The figure shows in red the road that was chosen by the mapping as the trajectory of the probe. But, it is not the right path and the right one is in green, which the system should have chosen.

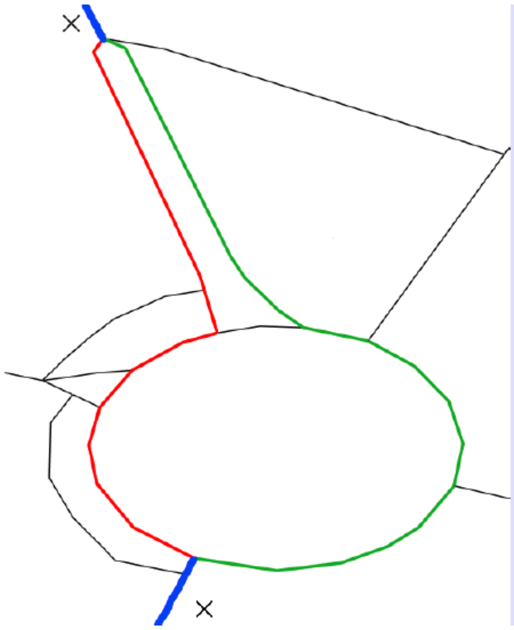


Figure 5.13: Orientation problem in the matching process

In order to reduce the rejected data due to the problem discussed above and also the wrong matching we will add the orientation aspect and a correction step to the algorithm (STC-Matching) as it was described in chapter 3.

5.3.3.3 STC-Matching

5.3.3.3.1 Results

The result illustrated in the figure 5.14 show the final result of the algorithm proposed in this section. We succeed to make the matching process better. We have to take into consideration that some of the problem are due to the digital map missing data.

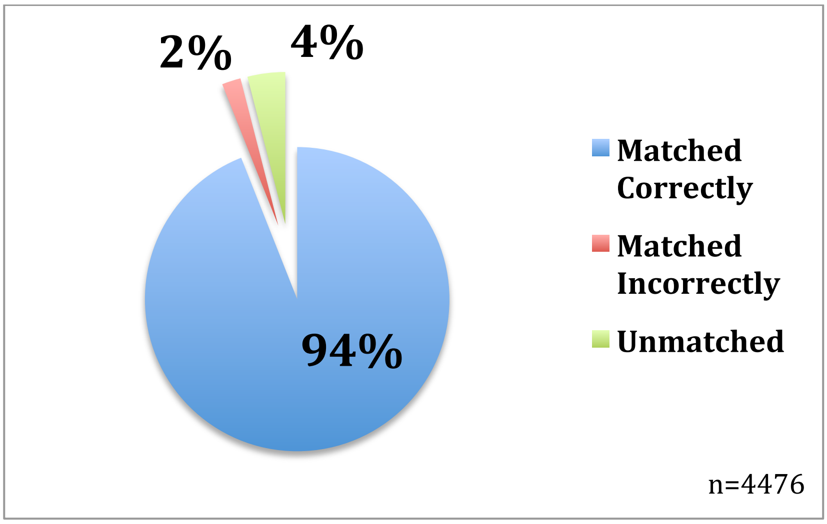


Figure 5.14: STC-Matching results

5.3.3.3.2 Analysis

The results obtained are satisfying to continue in our work for the next step of estimating the travel time. Of course there is always more enhancement that we can do. Besides, the 2% of wrong matching is not that significant compared to the amount of data used, which will not affect our results.

5.3.4 Conclusion

After we tested the Map-matching algorithm proposed in this thesis. The outputs show good results and encouraging. The table 5.1 resumes all the results regarding all the algorithms approaches regarding the map matching process. During the testing process we found out that the good functioning of the map-matching algorithm depends not only on the robustness of the algorithm but also on other factor such as a good GIS digital map and a good extraction algorithm.

Table 5.1: Results Summary of the Map-Matching Algorithm

	S-Matching	**Sup-matching**	**ST-Matching**	**STC-Matching**
Unmatched	26%	19%	9%	4%
Matched Incorrectly	14%	16%	3%	2%
Matched Correctly	60%	65%	89%	94%

The performance of the STC-Matching developed in this research can be compared to other techniques of existing map matching algorithms. We will summarize them in the following table:

Table 5.2: The performance of some existing map matching algorithms and ours

Map Matching algorithms	Navigation Sensors	Test Environments	Matched Correctly
Pyo et al, 2001	GPS	Urban and suburban	88.8%
White et al, 2000	GPS	Suburban	85.8%
Bouju et al 2002	GPS	Suburban	91.7%
Yu et al, 2002	-	-	86.3%
Srinivasan , 2003	-	-	80.2%
Fu et al, 2004			80.5%
Syed and Cannon, 2004	GPS/DR (Dead Reckoning)	Urban and suburban	92.5%
Yang et al, 2003	GPS	Suburban	96%
STC-Matching	**GPS**	**Urban and suburban**	**94%**

The results in table 5.2 shows the STC-Matching algorithm give good results compared to the others. Especially, compared to the ones that they used the same navigation systems.

Moreover the running time of the process figure 5.15 is acceptable. The "Inf" notation in the graph means that we did not put a limit to the number of data processed. The number of candidates represents the number of GPS position injected into the system to be matched. Overall, the running time of the system is acceptable.

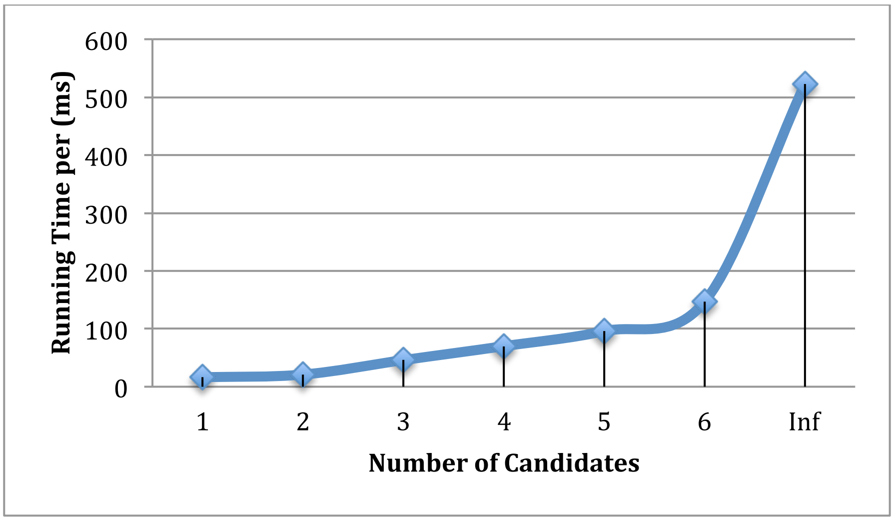


Figure 5.15: STC-Matching Running Time vs Number of candidates

Now that we showed the results concerning the map-matching process, we will check our hypothesis stated about the speed distribution per road section.

5.4 Speed Distribution

In this section we will check the hypothesis stated in chapter 4 about the speed distribution.

5.4.1 Description

The test that we will conduct in this part is to try to found out if the speed follows a bimodal distribution on each road section.

5.4.2 Test Strategy

5.4.2.1 Dataset Description

Road Network: In our experiment, we will use the same network defined before in the map-matching test. The network contains 7739 vertices and 18874 road sections.

Test Data: the data used in this test is the sparsely sampled GPS data available in our historical database. The data contain about ten millions GPS positions that was collected over one year.

5.4.2.2 Evaluation Approach

Evaluation Criteria: in this test analysis we will start by creating a histogram of the speeds detected in each road section in our road network. The histogram will help us to get an idea about the speed distribution.

5.4.3 Experimental Results

5.4.3.1 Results

The test ran on the historical database showed that almost all the speed histograms of each section follow a bimodal distribution. The figure 5.16 gives an overview of the results that we found after running the test.

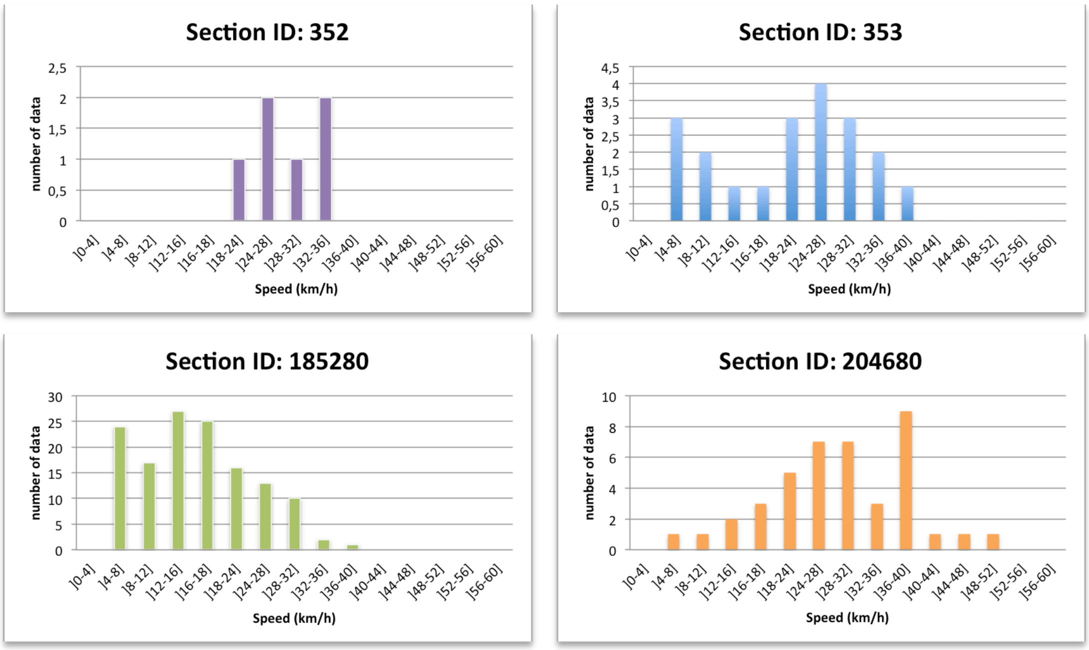


Figure 5.16: Overview of the speed histogram per road sections

The next figure 5.17 gives a view of the probability distribution of the speed on each road section.

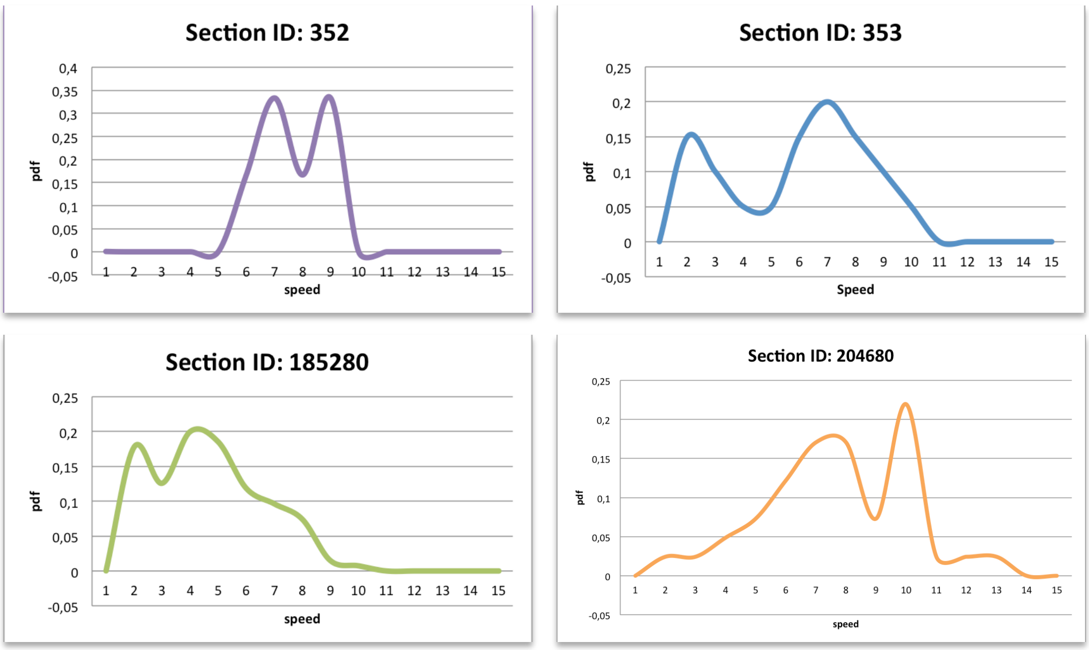


Figure 5.17: overview of the pdf of the speed per road sections

5.4.3.2 Analysis

Based on the results the bimodal distribution is the closed probability distribution that can describe the speed distribution in reality. Thus, our hypothesis is confirmed.

5.5 Travel Time Estimation

In this section we will conduct our test regarding the two approaches that we presented in chapter 4. First we will present the results of the travel time estimation using Monte Carlo Method (TT-MCM). Next we will show the results of the travel time estimation based on the Monte Carlo Method enhanced with measurements and road section's characteristics (TT-MCM-E).

5.5.1 Description and Implementation

In order to run the process and get the travel time estimation per road section. The function that estimates the travel time has to call another function that compute the estimation of the moment of passage through the PUMAS points.

The moment of passage is computed using two different methods (TT-MCM and TT-MCM-E). The TT-MCM approach uses the distance function and the shortest path function (TDSPP_Calculator). Concerning the TT-MCM-E approach needs the same functions as TT-MCM plus the road section's distribution function that is used in the process. The function road section characteristics uses the speed data saved in the historical database in order to create the travel time distributions for each road section.

The figure 5.18 shows the top-down analysis of the implementation.

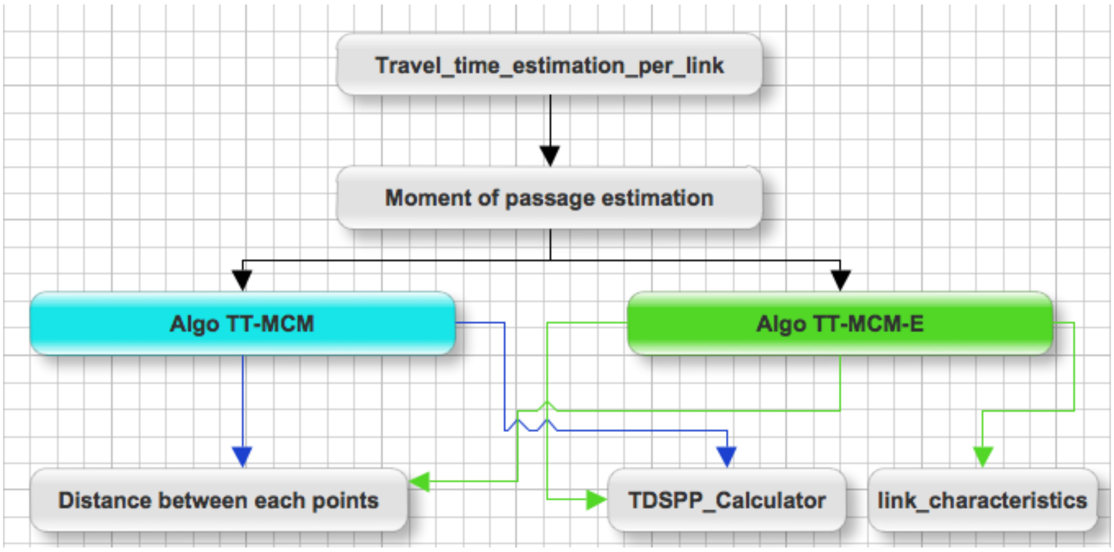


Figure 5.18: Top-Down analysis of travel time estimation process

5.5.2 Test Strategy

5.5.2.1 Dataset Description

Road Network: In our experiment, we will use the same network defined before in the map-matching test. The network contains 7739 vertices and 18874 road sections.

Test Data: In order to make our test we used the GPS data (one second frequency) that was described and used for the map-matching test. The same data collected from the experiment described in chapter 2.

5.5.2.2 Evaluation Approach

Ground Truth: we simulated our historical data using the same data used for the map-matching test. As it was explained before, we changed the receiving frequency of GPS information from one second to one minute. This yielded data similar to our historical database, but with the advantage of the travel time reference collected from the real-world field. The real world reference data will allow us to validate our results coming from the two estimation methods.

Evaluation Criteria: to evaluate the two methods, we will check the running time and the estimation errors. The running time is measured by the actual program execution time. The estimation errors is measured using the percentage difference error metric defined as follows:

$$\%DifferenceError = \left| \frac{Tp_{est} - Tp_{ref}}{\left(\frac{Tp_{est} + Tp_{ref}}{2} \right)} \right| * 100 \qquad (5.2)$$

Where,

- Tp_{est} is the estimated travel time
- Tp_{ref} is the travel time of reference.

Baseline: to present the results we will process each method alone. In addition we will discuss the results of each method.

5.5.3 Experimental Results

5.4.3.1 Travel time estimation based on Monte Carlo method (TT-MCM)

5.5.3.1.1 Results

After running program on our simulated historical database we get the travel time estimation. Then we compute the percentage difference error with the travel time reference that we computed from the real-world field experiment (Table 5.3).

Table 5.3: Sample of the Percentage Difference Error of TT-MCM Table

Section ID	%Difference Error TT-MCM
18	20
19	0
20	0
23	8.10
26	24.11
82	5.88
83	33.33
84	5

In order to have a clear view of the results we will outline the percentage difference error in a histogram (Figure 5.19).

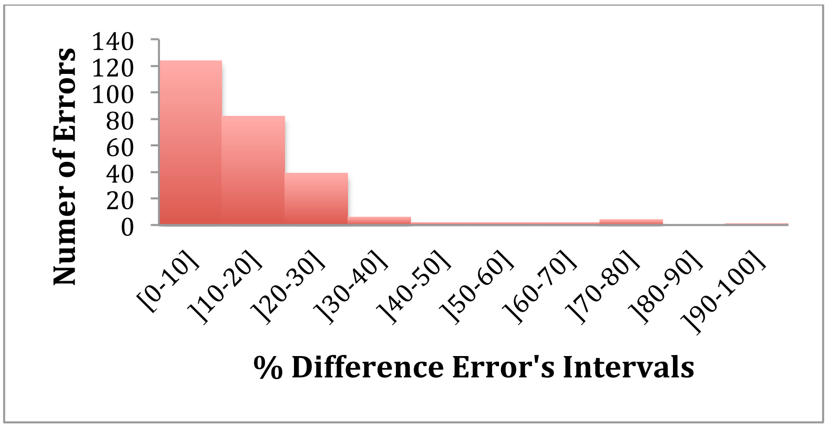


Figure 5.19: Histogram of Percentage Difference Error Frequency of The TT-MCM

The histogram has a shape of half Gaussian distribution with an average percentage difference error per road section equal to 7% and the standard deviation equal to 13.57%.

5.5.3.1.2 Analysis

Based on the results obtained it seems that the TT-MCM algorithm give interesting results by an average percentage difference error equal to 7% per road section.

5.5.3.2 Travel time estimation based on Monte Carlo method enhanced with measurements and road sections characteristics (TT-MCM-E)

5.5.3.2.1 Results

We will run the same process for the TT-MCM-E on our simulated historical database we get the travel time estimation $Tp_{est-MCM-E}$. Next we compute the percentage difference error (Table 5.4).

Table 5.4: Sample of The Percentage Difference Error of The TT-MCM-E Table

Section ID	%Difference Error TT-MCM
18	11,11
19	0
20	0
23	0
26	0
82	5.88
83	0
84	12,5

The same we will represent the results of the percentage difference error in a histogram representation (Figure 5.20).

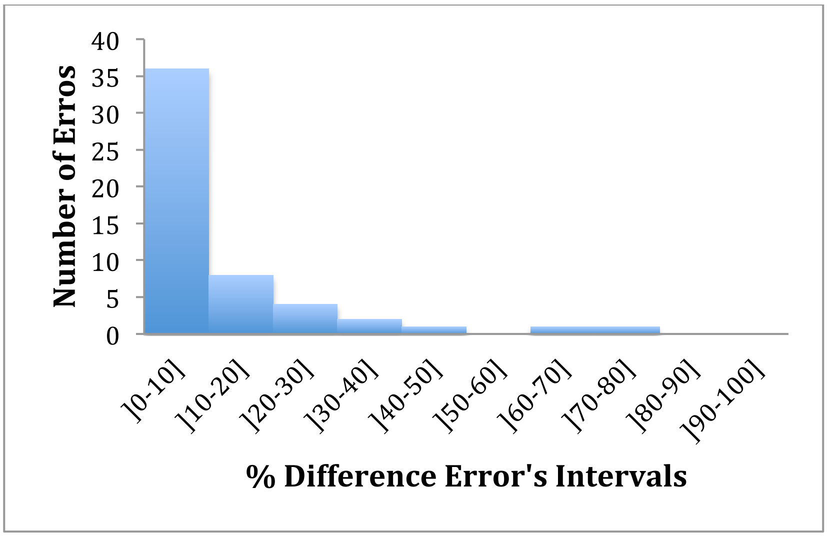


Figure 5.20: Histogram of the Percentage Difference Error Frequency of the TT-MCM-E

The histogram has a shape of half Gaussian distribution with an average percentage difference error per road section equal to 2% and its standard deviation is 8%. Based on the description of the histogram there is a big chance that the errors will be in the area of]0-10].

5.5.3.2.2 Analysis

The result seems to prove that the TT-MCM-E approach gives better results than the TT-MCM. However, in order to have a clear idea about the results we will conduct a different criterion in comparing the two methods.

5.5.3.3 Comparison of the Two Methods

5.5.3.3.1 Evaluation Criteria

In order to have more information about the performance of the two methods we will conduct a mean square error (MSE) evaluation. The MSE analysis is used to evaluate the error between a series of numbers found in a pair of same dimension vector arrays. For two distinct arrays X and Y, the general formula for the MSE equation is defined as follows:

$$MSE = \frac{[\sum_{i=1}^{n}(x_i - y_i)^2]^{1/2}}{n} \qquad (5.3)$$

Where,

- x_i is the ith element in array X
- y_i is the ith element in array Y
- n is the number of elements in arrays X and Y

Thus in our case the equation will be as follows:

$$MSE = \frac{[(Tp_{ref} - Tp_{est})^2]^{1/2}}{1} \qquad (5.4)$$

We will define also the percentage MSE as follows:

$$PCTMSE = 100\% * [\frac{MSE}{Tp_{est}}] \qquad (5.5)$$

5.5.3.3.2 Results

The results will be represented in figure 5.21. The histogram gives an idea about the percentage of mean square errors of the two methods. On one hand, It is clear that the errors produced by the MCM approach are higher than the MCM-E.

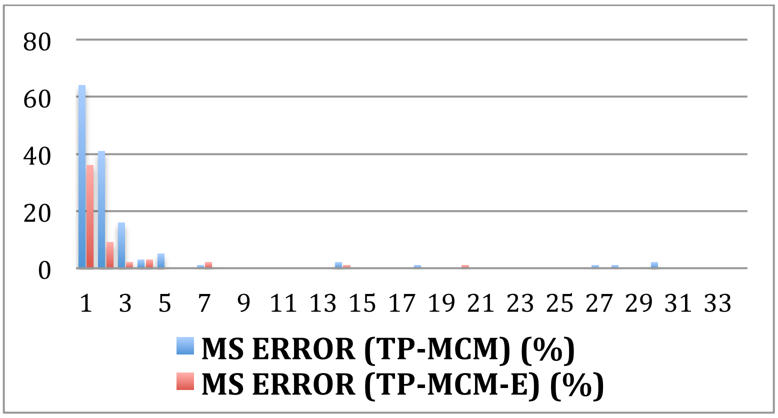


Figure 5.21: Histogram of PCTMSE (%) of the two methods

On the other hand, the MCM method produced almost a mean error of 32% and the MCM-E approach a mean of 8% (Table 5.5). Therefore, the MCM-E has a tendency to make less error per road section than the MCM method.

Table 5.5: Sample MSE Analysis of TT-MCE and TT-MCM-E

Section - ID	TP-REF	TP-MCM	TP-MCM-E	MSE (TP,MCM) (s)	MSE (TP,MCM-E) (s)	PCTMSE (TP,MCM) (%)	PCTMSE (TP,MCM-E) (%)
18	2	3	2,5	1	0.5	33,33%	20%
19	3	3	3	0	0	0%	0%
23	40	34	40	6	0	17,64%	0%
83	9	8	7	1	2	12,5%	28,57%
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
Average				**11,58**	**0,40**	**31,66%**	**8,39%**

The mean error does not give a clear idea about the performance of the two methods. For this reason will give the errors interval in order to know the MS error percentage per road sections. The TP-MCM approach has an interval error between 0% and 97% and the TP-MCM-E has an interval error between 0% and 34% (Table 5.6). Thus, the TP-MCM-E gives less error per road sections compared to the TP-MCM method.

Table 5.6: minimum and maximum MS Error per road sections

Methods	**MS Error (TP-MCM)**	**MS Error (TP-MCM-E)**
min	0%	0%
max	97%	34%

5.5.3.3.3 Discussion of MSE Analysis

By checking the average MSE and PCTMSE in table 5.5 and figure 5.21 it is clear that the TT-MCM-E method has the lowest percentage MS error per road section. Based on this it seems that the TT-MCM-E approach gives better results than the TT-MCM method. However, by checking the graph figure 5.21 and table 5.5 carefully, we found that in some cases the TT-MCM-E method misses sometimes the real results or produces more error than the MCM method. Which means that the problem is coming from the probability density function of travel time extracted from the historical database of the sparsely sampled data that we used.

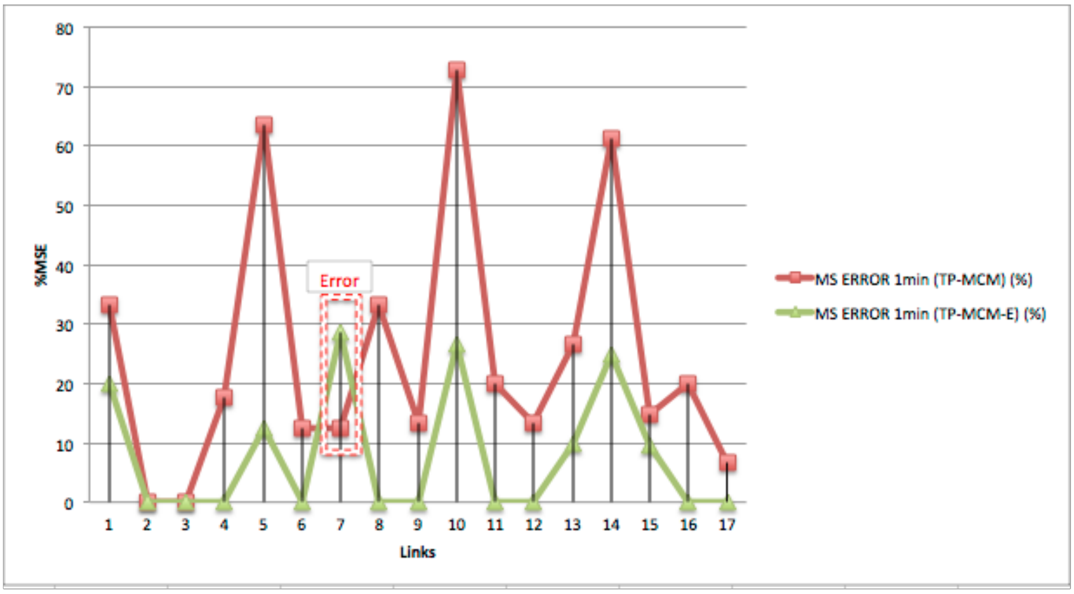


Figure 5.22: Example of %MSE vs Road sections with error problem

To understand the source of this issue illustrated in figure 5.22. We will show the speed data that we have in our historical database and it's importance per road section. This action will give us the impact on the probability density function of travel time per road section because we use the speed information to extract it.

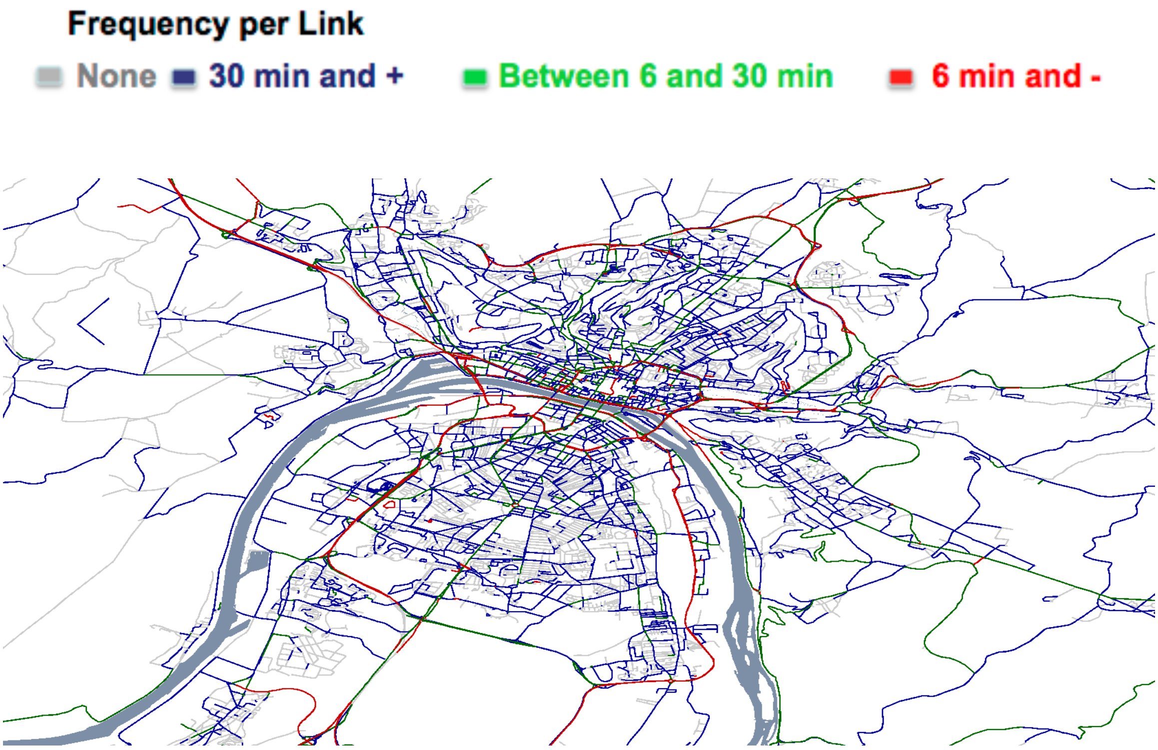


Figure 5.23: Data Frequency Presence per Road section

From the map (figure 5.23) it is clear that some section does not have enough data to learn about the travel time distribution per road section. Which explain the reaction of the TT-MCM-E approach and its results. Because, in some road sections either we don't have enough data or no data in order to give a realistic probability distribution of the travel time per road section that we use in MCM-E approach. As consequence, the weighting process in the particle filter affects bad quality weights to the particle. Thus, the travel time estimation is not good.

Concerning the running time of the system is shown in figure 5.24. We can see that the system is capable of running a lot of data without any problem.

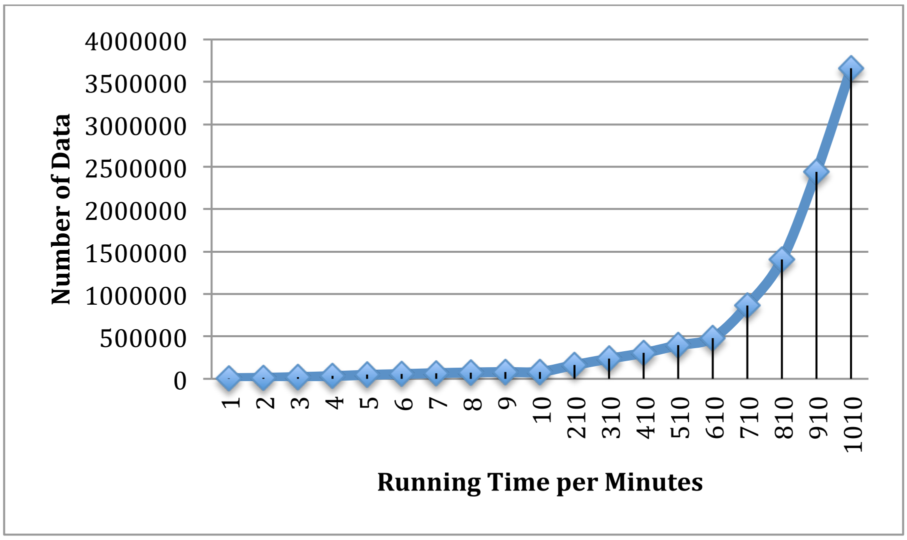


Figure 5.24: Number of processed Data vs running time

5.5.4 Conclusion

The results collected in this thesis work regarding the travel time estimation using the Monte Carlo Methods with an enhancement gave interesting and encouraging results. In addition, the approach to estimate the travel time is new and of Couse needs more test and enhancement. It is hard to do so, because of the use of sparsely sampled GPS data, which is already challenging. Moreover, filtering the data and the map-matching make the issue more complicated; because the good performance depends also on the good performance of the other components (filtering, map-matching, etc). Plus, to make good estimation we need important amount of data with good quality.

5.6 System Platform for project PUMAS

The fact that this thesis works was done within a project; the Creation of a friendly graphical interface where we can view all the information of the system was a need. The interface will allow viewing the results in a statistic and cartographic manner.

The parameters interface is set by default; however, we are allowed to change the setting of the parameters as it is shown figure 5.25. The parameter window contains all the parameters that are used in the system and we can control and change them.

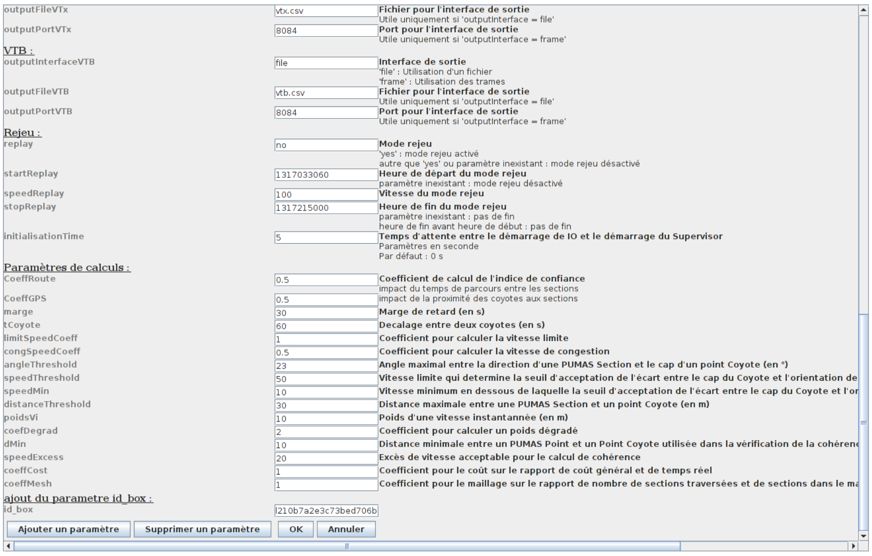


Figure 5.25: Software Parameter window view

The graph representation is using layer in order to have a view of the map. For example figure 5.26 show the PUMAS sections and the PUMAS points on the map.

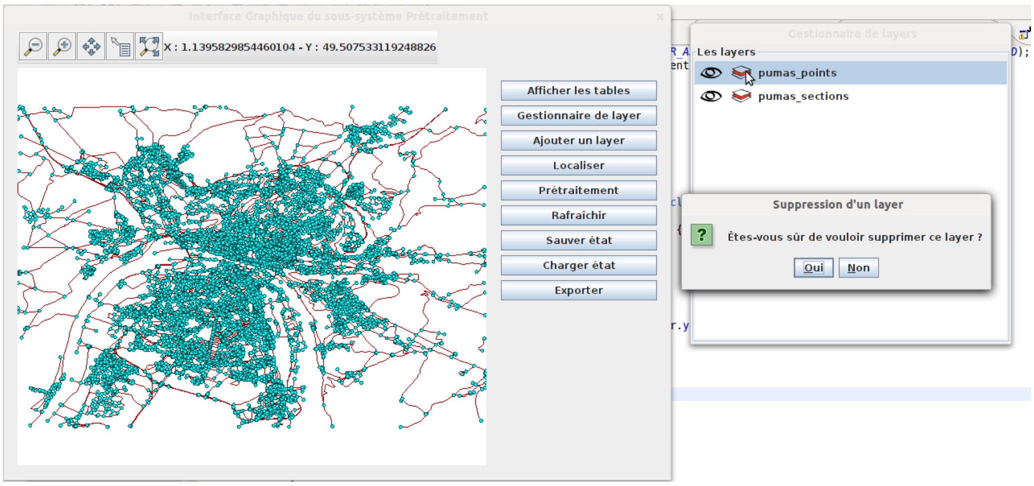


Figure 5.26: View of the software with PUMAS sections and PUMAS point's layers

Through the interface we can view a specific trajectory and we can get all the information regarding the travel time estimation, section ID, speed, etc by clicking on the concerned road section. The figure 5.27 shows an example of a trajectory.

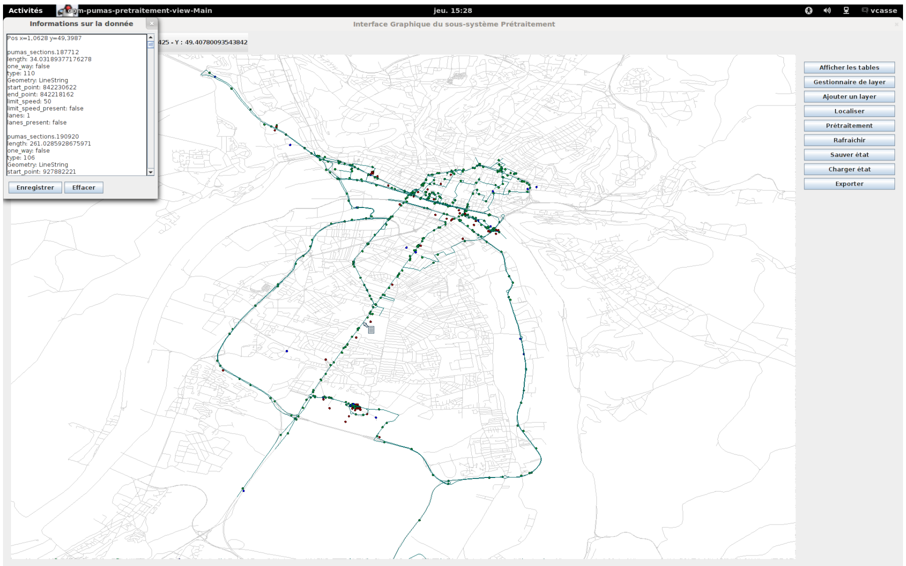


Figure 5.27: Software View with information in the small window on the left about a specific trajectory

5.7 Conclusion

To sum up, in this chapter we presented the results of this thesis work. The chapter dealt with the implementation description, test strategy for each section, evaluation criteria for each proposed method. Each presented results was analyzed and discussed.

In general the results obtained concerning the map matching are encouraging. However, there some issue where there is more work to be done; especially regarding the digital map and the GIS information, some of the problems was discussed in that section. For example, the map is not really representing the real world, some road sections are missing or some intersections. We did as we could an update of the digital map in order to reduce the errors.

Besides, we tested the travel time estimation results and analyze them. The new approach presented is very interesting and promising. However, the approach depends on other information that we collect from the historical database, which mean we need a good data to get better performance.

Finally at the end we presented the GUI developed during this thesis period for viewing the results in friendly way and easy to understand. The global results of this thesis work are a good start for the new approach and as we know there is nothing perfect from the first try. Thus, there some work to be done in the future to make the approach more robust.

Chapter 6: Conclusion and Perspectives

6.1 Conclusion

This thesis research work was conduct within the project PUMAS, which was an advantage for our research regarding the collection process of our data from the real world field and also in making our tests. The research done can be listed under the field of intelligent transportation systems. Besides, the output of this thesis was software that was used in the project. The software was designed in such manner to be integrated in any other design and also it is easy to add other modules to it or make changes. Our objective was to estimate travel time using sparsely sampled data in an urban context. The context and the nature of data used presented many challenges, which pushed us to conduct many other issues before treating our main goal.

After a bibliography study about the subject in chapter 2 we come out that we need to investigate other sub subjects before estimating travel time per road section. Our research work in this thesis investigated four fields and they are as follows: building digital map with its specific geographic information system (GIS), map-matching problem, shortest path problem, and estimating travel time per road sections in a urban context.

6.1.1 Digital map and GIS information

During our research we faced the importance of creating the digital map for example in transportation planning and logistic, it is advantageous to use digital map incorporated with information such as transportation facilities data, speed limits, roadway indicator, and type of roads. All this information is very helpful to improve traffic algorithms and estimations. Thus, we extracted our digital map from OpenStreetMap (OSM) and we added our GIS information that contain all the information needed such as speed limit on each section, intersection, defining the roads directions and type, new features, etc. the digital map created in this thesis work can be used in any application for traffic purpose.

6.1.2 Map-matching problem

In our approach we were inspired by (Zhao, 97) approach to define the area of interest and we defined a spatial analysis criteria by detecting the road section that has the highest probability that the GPS position should be matched to it in the digital map, without having to scan the whole map. Then we added temporal analysis criteria that we defined. Finally, we used (Kim et al, 2000) method by applying an orthogonal projection on the GPS position into the road section concerned that have the highest score of the combination of spatial and temporal criteria. Moreover, we added the idea of road orientation and the vehicle's heading in order to enhance the map matching. The orientation informs us about the direction that the car is following making it easier to locate the road that has the same direction as car's heading. In some of the case we still have a non-matching cases. Therefore, we added a correction method (Balasundaram, 2009) that make a new prediction of the defective data and then apply the whole process of map matching to the corrected data.

6.1.3 Time-dependent shortest path problem

The time-dependent shortest path problem given a departure and arrival time was one of our research interests. There are many solutions that has been developed for different types of graphs such as Dijkstra in the case of positive weights, Bellman in the general case, etc. However, these methods are time consuming regarding the computational process, which mean that they need enhancement and speed up. We adopted a new approach in a database context. In this new method we introduced a new step that we called a learning phase that we process on offline mode using a recursive Dijkstra and then when the method is called by the map matching on online status. Introducing this learning step at the level of the database enhanced the speed of finding the answer on online mode.

6.1.4 Travel time estimation per road sections

The use of sequential Monte Carlo approach to estimate travel time using sparsely sampled GPS data is certainly not new. However, the application of Monte Carlo methods on urban networks has not been found in the literature.

Accurate travel time estimation is an important element for advanced traveler information systems and advanced traffic management systems as well as for all transport users. In urban networks, travel time estimation is challenging due to many reasons such as the fluctuations in traffic flow due to traffic signals and drivers behavior.

6.1.4.1 Travel time estimation using Monte Carlo method

Using this approach we adopted the Monte Carlo method to our case by defining the state equation of the case studied. The creation of the filter was done in such way to process and use the sparsely sampled GPS data. The filter gives us the ability to estimate the moment when the probe vehicle enters a road section and also when he exits the same road section. By considering these information it was easy for us to know how long the probe vehicle was in that section. Which means that we have the travel time of the probe vehicle in that specific road section.

6.1.4.2 Travel time estimation using Monte Carlo method enhanced with measurements and road sections characteristics

The enhancement of this approach was by changing the algorithm by injecting it with information concerning the measurements and characteristics of the concerned road section on the road network. This information was injected at the level of weighting process of the particles generated by the particle filter or Monte Carlo method. The weight injected is a probability density distribution of the time distribution on the road section learned from the historical database. This maneuver makes this approach original and new.

Moreover, in order to have a results closed to reality we added in both methods (TT-MCM and TT-MCM-E) the notion of delay time at the level of road intersections. This aspect was added in order to illustrate the lost time at a traffic light or when we are changing road sections during our journey trajectory.

In addition the results obtain are very encouraging to continue on the evolution and enhancing the approach presented in this thesis work. Besides, it can be an open window to solve other problematic in the domain of ITS and traffic management.

6.2 Perspectives

The travel time estimation is a small puzzle of the big image of traffic status. However, each small pawn constitutes the traffic chessboards that are very important. Travel time estimation is one of those pawns. The future work that can be done is to use the other pawns that will be discussed in the next sections.

6.2.1 Short-term agenda

6.2.1.1 Travel time in urban arterials

As a continuation of our research work it will be beneficial to use different sensors to make the estimation of travel time more precise. The traffic data from probe vehicles, loop detectors and camera detector sources have different levels of precision, which may result in inconsistency and sometimes even contradictory estimates. Data fusion is the processing tool and solution. This latter takes into account the quality of the data provided by each source with an aim of increasing the precision, reliability and robustness of the estimation.

Another aspect that can be very interesting is the combination of the mathematical models and statistical models based on real world observation. This approach will help to increase the precision of the estimation and also build an understanding of the real world and its behaviors. This approach needs a large amount of data to be processed; however, nowadays with an advance in technology and research it is possible to afford the data needed.

By building and making available a historical data of estimated travel time per road sections. The next step will be to understand the traffic behavior. In order to do so, for example we can introduce an artificial intelligence (AI) method. This latter technique seeks to understand natural intelligence and to build intelligent systems. There are many techniques that have been applied in the fields (Lee, 2009) [123]. The most common method in the AI is Neural Network (ANN). The ANN have been applied in many fields like engineering management, science and also psychology and it helps to improve the solving process of the transportation problems (Dougherty, 1995) [124]. Moreover, the ANN capacity includes classification, detection, pattern recognition, adaptive filtering, data inversion, target tracking, estimation, modeling, etc. Besides, there is a statistical approach that is applied to estimation, prediction and modeling. As an illustration, we can take the work of (Zhang, 2003) [125] where he used data from loop detectors and probe vehicles to create a linear model for travel time prediction.

6.2.1.2 Traffic jam detection in urban network

One of the aspects that characterize traffic is the appearance of congestion. Thus it is an important step after getting information about travel time to detect congestions in order to complete the picture about traffic status. Most of the approaches used in the literature are based on the speed as a metric to describe traffic situation. However, the speed in urban areas is influenced significantly by road condition and traffic lights. Thus, applying this approach in real world field is not a good idea. Which means using speed alone as a metric is not a good choice. From this point of view, it is important to define a new metric in order to detect the congestions. For example, as it was shown by (K. Zhang, 2010) [126] the use of traffic rate as a metric and applying a spatio-temporal OD matrix gave good results regarding traffic jam detection. Our idea will be to use what was developed regarding the travel time estimation and use it as an input for the OD matrix. By using an accurate representation of traffic status regarding spatio-temporal aspect (ex. Travel time) the results of the OD matrix should be better and closer to the reality.

Another approach to detect traffic congestions is by using cooperative vehicles. Cooperative vehicle have been used to enhance traffic safety and management. Though by this approach we can get more accurate information about the traffic status using V2V communications. For example (Bauza, 2010) [127] proposed a method to detect congestions by using a cooperative probe based on V2V communications and a fuzzy logic without the need of any infrastructure sensors deployment. The use of cooperative vehicle can help to detect the birth of congestions. Moreover, we can add the sharing information aspect through the V2V communication to avoid the phenomenon of traffic jams. Therefore, the approach presented may be promising for good work that can be done in this issue, which will help a lot to understand the urban traffic.

6.2.2 Long-term agenda

For a long-term perspective we have to take into consideration the implications and the impact of our research. All what is developed in transportation is a small pixel of the whole image. This image is our cities and metropolitan regions. Nowadays, managing our cities is an issue of concern of everybody such as regional officials, road administrators, urban planners and citizens; in order to make good decision about the future of our cities, they need correct information. For example, making decisions about building new freeways, transit service expansion, land use regulations are often expensive and controversial with long-term consequences.

From this prospective we must attempt to understand how different alternatives might affect land use, transportation, and the environment over the next decades. Therefore as a project related to what was done in our research regarding understanding the traffic via travel time estimation and prediction, traffic jam detection, and traffic simulation or forecasting; we can add the idea of land use in the system. Then we have to study the relationship between the road traffic and the land use. As we know, when we create a new housing area for example, we are obliged to create new roads. As a consequence, a new flow of traffic will be injected in the road network of the city. Afterward by merging the land use model and the dynamic traffic simulation model, we can help to make decisions regarding the choice of the location and size of the land used.

Another interesting research subject for our society is evacuation under emergency situation. Dynamic traffic simulation models are used to support decisions when planning an evacuation. Till now, understanding the traveller's behavior under emergency evacuation condition is still conducted by researchers. There are some models trying to model the situation like (Pel, 2011) [128], but there is a lot of effort to be done regarding this issue in research.

Publications

- Amnir Hadachi, Christele Lecomte, Stephane Mousset and Abdelaziz Bensrhair,"An Application of the Sequential Monte Carlo to Increase the Accuracy of Travel Time Estimation in Urban Areas", 14th International IEEE Conference on Intelligent Transportation Systems, Washington, DC, USA. October 5-7, 2011.

- Amnir HADACHI, Stéphane MOUSSET, Abdelaziz BENSRHAIR, "Practical Testing Application of Travel Time Estimation Using Applied Monte Carlo Method and Adaptive Estimation from Probes", IEEE INTELLIGENT VEHICLES SYMPOSIUM, Alcalá de Henares, Spain, June 3-7, 2012.

- Amnir Hadachi, Stephane Mousset, Christele Lecomte and Abdelaziz Bensrhair," Travel Time Estimation Using Cooperative Probe Vehicles", The 6th International Symposium on signal, Image, Video and Communications, Valenciennes, France on July 4-6, 2012.

- Amnir Hadachi, Stephane Mousset, Abdelaziz Bensrhair, "Increasing the Travel Time Accuracy in Urban Arterial Networks using Particle Filter and Enhanced Map Matching." Submitted to the IEEE ITS Transactions Journal 2012.

Bibliography

[1] Coifman, B., and Krishnamurthy, S. "Vehicle Reidentification and Travel Time Measurement across Freeway Junctions using the Existing Detector Infrastructure." Transportation Research Part C: Emerging Technologies, 15(3), 135-153, 2007.

[2] Wei, C.-H., and Lee, Y. "Development of Freeway Travel Time Forecasting Models by Integrating Different Sources of Traffic Data." Vehicular Technology, IEEE Transactions on, 56(6), 3682-3694, 2007.

[3] Takahiro Tsubota, Ashish Bhaskar, Edward Chung, Romain Billot, ¨Arterial traffic congestion analysis using Bluetooth Duration data,¨ Australasian Transport Research Forum 2011 Proceedings, Adelaide, Australia, 28 - 30 September 2011.

[4] M. LI, Z. Zou, and F. Bu, "Arterial Travel Time Estimation Using VII Probe Data and Point-based Detection Data", 13th International IEEE Annual Conference on Intelligent Transportation Sytems, Madeira Island, Protugal, September 19-22, 2010.

[5] F.Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. J. Nordlund. “Particle filters for positioning, navigation, and tracking,” Signal precessing, IEEE transaction on Signal processing, Acoustics , and speech, 50(2):425-437,2002.

[6] J.-C. Herrera and A. Bayen,"Traffic flow reconstruction using mobile sensors and loop detector data", 87th Transportation Research Board Annual Meeting, Washington D.C., January 13-17 2008

[7] Carolina Pinana-Diaz, ËIJ Rafael Toledo-Moreo, David Betaille, Antonio F. Gomez-Skarmeta, “GPS Multipath Detection and Exclusion with Elevation-Enhanced Maps”, 2011 14th International IEEE Conference on Intelligent Transportation Systems, Washington, DC, USA. October 5-7, 2011

[8] Steven I-Jy Chien and Chandra Mouly Kuchipudi,"Dynamic Travel Time Prediction with Real-time and Historical Data," Presentation and Publication to the Transportation Research Board, the 81st Annual Meeting.J, Transp. Engrg. Volume 129, Issue 6, pp. 608-616 (November/ December 2003)

[9] A. Hadachi, C. Lecomte, S. Mousset and A. Bensrhair,"An Application of the Sequential Monte Carlo to Increase the Accuracy of Travel Time Estimation in Urban Areas",2011 14th International IEEE Conference on Intelligent Transportation Systems, Washington, DC, USA. October 5-7, 2011

[10] Torday, A.; Dumont, A.-G, "Probe Vehicles Based Travel Time Estimation in Urban Networks". Proceedings of the Triennial Symposium on Transportation Analysis TRISTAN V, Le Gosier – French West Indies, June 13 - 18, 2004.

[11] Takaba, S., T. Morita, T. Hada, T. Usami, and M. Yamaguchi. "Estimation and Measurement of Travel Time by Vehicle Detectors and License Plate Readers," Proceedings, Vehicle Navigation And Information Systems Conference, Vol. 1, Society of Automotive Engineers, pp. 257–267, 1991.

[12] Amri M. Y, M. R. Karim, A.Saifzul Abdullah, "Travel Time Measurement in Real Time using Automatic Number Plate Recognition for Malaysian Envirement," Journal of the Estern Asia Society for Transportation Studies, Vol 8, 2009.

[13] Damian Kelly, Ross Behan, Rudi Villing and Sean McLoone, "Computationally Tractable Location Estimation on WiFi Enabled Mobile Phones," ISSC, UCD, June 10–11th, 2009.

[14] A. Hofleitner and A. Bayen," Optimal decomposition of travel times measured by probe vehicles using a statistical traffic flow model", 2011 14th International IEEE Conference on Intelligent Transportation Systems, Washington, DC, USA. October 5-7, 2011.

[15] Fang Clara Fang and Neftali Torres," Analysis of Hydrogen Station Network Using Geographic Information Systems," 14th International IEEE Conference on Intelligent Transportation Systems, Washington, DC, USA. October 5-7, 2011.

[16] S.Vasantha Kumar, Lelitha Vanajakshi, and Shankar C. Subramanian," A Model Based Approach to Predict Stream Travel Time using Public Transit as Probes", 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, June 5-9, 2011.

[17] Meng Li, Zhijun Zou, and Fanping Bu, "Arterial Travel Time Estimation Using VII Probe Data and Point-based Detection Data", 2010 13th International IEEE Annual Conference on Intelligent Transportation Systems, Madeira Island, Portugal, September 19-22, 2010.

[18] Gao Lianxiong, Liang Hong, Liao Zerong, "Mining Travel Time from Intelligent Card Toll Data", 2010 13th International IEEE Annual Conference on Intelligent Transportation Systems, Madeira Island, Portugal, September 19-22, 2010.

[19] Nanthawichit, C., Nakatsuji, T., and Suzuki, H, "Application of Probe Vehicle Data for Realtime Traffic State Estimation and Shortterm Travel Time Prediction

on a Freeway" 82nd Annual Meeting Preprint CD-ROM, Transportation Research Board, January 12-16, Washington, D.C. 2003.

[20] C. Chen, K. Petty, A. Skabardonis, and P. Varaiya. Freeway performance measurement system: mining loop detector data. In 80th Annual Meeting of the Transportation Research Board, Washington, D.C., January 2001.

[21] Z. Jia, C. Chen, B. Coifman, and P. Varaiya. The PeMS algorithms for accurate, real-time estimates of g-factors and speeds from single-loop detectors. In 4th IEEE Conference on Intelligent Transportation Systems, Oakland, CA, August 2001.

[22] Lianyu Chu, "Adaptive Kalman Filter Based Freeway Travel time Estimatio",California Center for Innovative Transportation University of California, Berkeley, TRB Annual Meeting, 2005.

[23] Bolbol, A., et al. "Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification." Computers, Environment and Urban Systems Journal, 2012.

[24] Sanjeev Arulampalam, Simon Maskell, and Neil Gordon. "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," IEEE Transactions on Signal Processing, 50:174-188, 2002.

[25] Ma Hongwei, Zhou Xizhao, Fu Liyou, "Study on the Reliability of Travel Time in Transportation Network ," Intelligent System Design and Engineering Application (ISDEA), pp 545- 548, 2012.

[26] Jiann-Shiou Yang, "Travel time prediction using the GPS test vehicle and Kalman filtering techniques," American Control Conference, pp 2128- 2133 vol. 3, 2005:

[27] Casper G. Chorus, Theo A. Arentze, Harry J.P. Timmermans, " A General Model of Bayesian Travel Time Learning,"Transportation research Board 87th Annual Meeting Washington, D.C, 2008.

[28] S. A. Oke , L. Ogunwolu , O. Adedokun , O. Orimoloye, "A neuro-fuzzy approach to vehicular traffic flow prediction for a metropolis in a developing country," Journal of Industrial Engineering International Volume 7, Issue 13, 30 Mar 2011.

[29] Ying Lee, "Freeway Travel Time Forecast Using Artificial Neural Networks With Cluster Method," 12th International Conference on Information Fusion Seattle, WA, USA, July 6-9, 2009.

[30] Tai-Jin Song, Cheol Oh, “Estimation of legibility Distance for Portable Variable Message Signs,” Journal of the Eastern Asia Society for Transportation Studies, Vol8, 2010.

[31] Antonio Fernandez-Caballero, Francisco J. Gomez, Juan Lopez-Lopez, ”Road-traffic monitoring by knowledge-driven static and dynamic image analysis”, Expert Systems with Applications 35, 701–719, 2008.

[32] Mahmoud Abuelela and Stephan Olariu, "Automatic Incident Detection In VANETs: A Bayesian Approach," In Proceedings of the IEEE Vehicular Technology Conference - Spring. Barcelona, April 2009.

[33] N. J. Gordon, D. J. Salomond, and A. F. M. Smith. “Novel approach to nonlinear/non Gaussian Bayesian state estimation.” Radar and signal Processing, IEE Processing F, 140(2): 107-113, 1993.

[34] Challita, G., Mousset, S, Nashashibi, F, Bensrhair, A., “Particle filters for accurate localization of communicant vehicles using GPS and vision systems ,” Computer Systems and Applications,, AICCSA , pp 238-242, 2009.

[35] Van Lint, J. W. C, and Van der Zijpp, N. J, “Improving A travel Time Estimation Algorithm by Using Dual Loop Detectors,” Transportation Research Record, 1855, Transportation Research Board, Washington, D.C, pp 41-48, 2003.

[36] Ré-Mi Hage, David Betaille, François Peyret, Dominique Meizel, “Unscented Kalman filter for urban network travel time estimation,” 15th meeting of the EURO Working Group on Transportation, 2012.

[36] Taehyung Park and Sangkeon Lee, “A Bayesian Approach for Estimating Link Travel Time on Urban Arterial Road Network,” COMPUTATIONAL SCIENCE AND ITS APPLICATIONS – ICCSA 2004.

[37] Lint, J.W.C., van, Hoogendoorn, S.P., Zuijlen, H.J., van (2002). “Freeway travel time prediction with State-Space Neural Networks,” the Transportation Research Board Annual Meeting, 14-17, Washington D.C., USA, January 2002.

[38] Van Lint, J. W. C, Hoogendoorn S. P, VAN ZUYLEN H. J, “Accurate freeway travel time prediction with state-space neural networks under missing data ,” Journal of Transportation research. Part C, Emerging technologies-ISSN:0968-090X, vol. 13, pp. 347-369, 2005.

[39] Lee, T.-C, Peterson, A.M, Tsai, J.-C, “A multi-layer feed-forward neural network with dynamically adjustable structures,” IEEE International Conference on Systems, Man and Cybernetics, 1990.

[40] Ben Guo; Liying Hu; Yang Bai, "PMSM servo system based on dynamic Recurrent Neural Networks PID controller," 7th International Conference on Power Electronics and Motion Control Conference (IPEMC), 2012.

[41] Eduardo Lalla-Ruiz, Belén Melián-Batista, J. Marcos Moreno-Vega, "Artificial intelligence hybrid heuristic based on tabu search for the dynamic berth allocation problem," Journal of Engineering Applications of Artificial Intelligence, Volume 25 ,Pages 1132-1141, 2012.

[42] Maryam Keyarsalan, Gholam Ali Montazer, "Designing an intelligent ontological system for traffic light control in isolated intersections," Journal of Engineering Applications of Artificial Intelligence, Volume 24, Issue 8, Pages 1328-1339, 2011.

[43] Raúl Pino, José Parreno, Alberto Gomez, Paolo Priore, "Forecasting next-day price of electricity in the Spanish energy market using artificial neural networks," Journal of Engineering Applications of Artificial Intelligence, Volume 21, Issue 1, Pages 53-62, 2008.

[44] Ching-Huei Huang, Chun-Liang Lin, "Evolutionary neural networks and DNA computing algorithms for dual-axis motion control," Engineering Applications of Artificial Intelligence, Volume 24, Issue 7, Pages 1263-1273, 2011.

[45] E. Chevallier and L. Leclercq, "A macroscopic theory for unsignalized intersections," Transportation Research Part B: Methodological, vol. 41, pp. 1139–1150, 2007.

[46] L. Chu, H. Liu, and W. RECKER. "Using microscopic simulation to evaluate potential intelligent transportation system strategies under nonrecurrent congestion." Transportation Research Record, 1886(- 1):76–84, 2004.

[47] Sébastien Blandin , Laurent El Ghaoui and Alexandre Bayen, "Kernel regression for travel time estimation via convex optimization," 48th IEEE Conference on Decision and Control Shanghai, China, Dec. 2009.

[48] Nagatani, T, "Gas kinetic approach to two-dimensional traffic flow," Journal of the Physical Society of Japan,, 60(10), 3150–3152, 1996.

[49] Lerner, G., A. Hochstaedter, R. Kates, C. Demir, J. Meier, A. Poschinger. "The interplay of multiple scales in traffic flow: coupling of microscopic, mesoscopic, and macroscopic simulation". 7th ITS World Congress on Intelligent Transport Systems, Turin, Italy, 2000.

[50] Jury Tolujew, Mihail Savrasov, " Transport System's Mescopic Model Validation Using", the 8th International Conference "RELIABILITY and Statistics in Transportation and Communication, 2008.

[51] Di Gangi, M. “Modeling Evacuation of a Transport System: Application of a Multimodal Mesoscopic Dynamic Traffic Assignment Model,” IEEE Transactions on Intelligent Transportation Systems, Vol 12, pages 1157-1166, 2011.

[52] Daniel B. and Alexandre M. Bayen, “The Mobile Millennium Project” , IQ Information Quarterly Magazine, Volume 8, Number 2, 2009.

[53] Tim Lomax, Shawn Turner, Bill Eisele, David Schrank, Lauren Geng, and Brian Shollar, “Refining the Real-Timed Urban Mobility Report”, UTCM University Transportation Center for Mobility, DOT Grant No. DTRT06-G-0044, March 2012.

[54] INRIX, Inc. website. Retrieved May 2012, from http://www.inrix.com/, last accessed October 1, 2011.

[55] Zito, R. and Taylor, M.A.P. “The use of GPS in travel-time surveys, Traffic Engineering and Control,” Vol. 35, No. 12,pages 685-690, 1994.

[56] Taylor, M.A.P., Woolley, J.E. and Zito, R. “Integration of the global positioning system and geographical information systems for traffic congestion studies,” Transport Research-Part C, Vol. 8, pages 257-285, 2000.

[57] Teemu Leppänen, Mikko Perttunen, Pekka Kaipio, Jukka Riekki, “Sensor Data Fusion Middleware for Cooperative Traffic Applications” International Journal on Advances in Networks and Services, vol 4 no 1 & 2, 2011.

[58] Corrado de Fabritiis, Roberto Ragona, Gaetano Valenti, “Traffic Estimation And Prediction Based On Real Time Floating Car Data”, Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems Beijing, China, October 12-15, 2008.

[59] Del Arco, E.; Morgado, E.; Ramiro-Bargueno, J.; Mora-Jimenez, I.; Caamano, A.J. “Vehicular Sensor Networks in congested traffic: Linking STV field reconstruction and communications channel”, 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2011.

[60] Manchun Li; Feixue Li; Chongya Jiang; Xue Li; Qiuhao Huang, “Construction of GIS scientific experience zone based on experience learning theory and constructivism”, 18th International Conference on Geoinformatics, 2010.

[61] Krajzewicz, Daniel und Hertkorn, Georg und Ringel, Julia und Wagner, Peter “Preparation of Digital Maps for Traffic Simulation; Part 1: Approach and Algorithms.“ the 3rd Industrial Simulation Conference 2005.

[62] Y.Hormann, H. P Grossmann, W. H. Khalifa, M. Salah, O. H. Karam, “Simulation for Inter-vehicle Communication Based on Traffic Modeling,” Intelligent Transportation Symposium IV, 2004.

[63] Baumgartner, A., Steger, C., Mayer, H., Eckstein, W., Ebner, H., “Automatic road extraction in rural areas.” Internat. Arch. Photogrammet. Remote Sensing, 107–112, 1999.

[64] Baumgartner, A., Steger, C., Mayer, H., Eckstein, W., Ebner, H. “Automatic road extraction based on multi-scale, grouping and context.” Photogrammet. Eng. Remote Sensing (65), 777–786, 1999.

[65] Census. “Topologically Integrated Geographic Encoding and Referencing system.” Retrieved September 20th from http://www.census.gov/geo/www/tiger/, 2012.

[66] M. Haklay and P. Weber. “OpenStreetMap: user-generated street maps.” IEEE Pervasive Computing, 7(4):12–18, 2008.

[67] Kim, W., Jee, G, Lee, J. " Efficient use of the digital road map in various positionning for ITS ". In: IEEE Symposium on Position Location and Navigation, San Diego, CA.2000.

[68] Honey, S. K., Zavoli, W. B., Milnes, K. A., Phillips, A. C., White, M. S., Loughmiller, G. E.,. " Vehilce navigation system and method ", United states Patent No., 4796191, 1989.

[69] Zhao, Y," vehicle location and navigation system ". Artech house,Inc,MA. 1997.

[70] Jae-seok Yang, Seung-pil KANG, Kyung-soo CHON, “THE MAP MATCHING ALGORITHM OF GPS DATA WITH RELATIVELY LONG POLLING TIME INTERVALS”, Journal of the Eastern Asia Society for Transportation Studies, Vol. 6, pp. 2561 - 2573, 2005.

[71] N. Bormann, S. Saarinen, G. Kelly and J.-N. Thepault, The spatial structure of observation error in the atmospheric motion vectors from geostationary satellite data. Geophysical Research Letters, 27 ,2661-2664, 2000.

[72] S.R. Balasundaram and A. Saravanan,”Location Prediction Mechanisms for Tracking Lost Unmanned Vehicles”, IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.12, December 2009.

[73] Dijkstra E.W. “A note on two problems in connection with graphs”, Numerische Mathematik, Vol. 1, pp. 269–271. 1959.

[74] Bellman R.E. “On a routing problem”, Quarterly of Applied Mathematics, Vol. 16, pp. 87–90.1958.

[75] Martin Holzer, Frank Schulz, and Thomas Willhalm, “Combining Speed-Up Techniques for Shortest-Path Computations”, ACM Journal of Experimental Algorithmics JEA, pages 1-18, 2005.

[76] Pohl, I.: Bi-directional and heuristic search in path problems. Technical Report
104, Stanford Linear Accelerator Center, Stanford, California,1969.

[77] Kaindl, H., Kainz, G.: Bidirectional heuristic search reconsidered. Journal of Artificial Intelligence Research 7, 283–317, 1997.

[78] Schulz, F., Wagner, D., Weihe, K.: Dijkstra’s algorithm on-line: An empirical case study from public railroad transport. ACM Journal of Exp. Algorithmics 5, 2000.

[79] Wagner, D., Willhalm, T.: Geometric speed-up techniques for finding shortest paths in large sparse graphs. In: Proc. 11th European Symposium on Algorithms (ESA). Volume 2832 of LNCS., Springer, 776–787, 2003.

[80] Holzer, M.: Hierarchical speed-up techniques for shortest-path algorithms. Technical report, Dept. of Informatics, University of Konstanz, Germany, 2003.
[81] K.Cooke & E.Hasley. “The Shortest route through a network with time dependence intermodal transit times,” Journal of Mathematical Analysis and Applications, 14:493-498, 1966.

[82] S. Dreyfus. An appraisal of some shortest-path algorithms. Operations Research, 17(3):395–412, 1969.

[83] R. Ahuja, T. Magnanti, and J. Orlin. Network flows:theory, algorithms, and application. Prentice-Hall, Inc, Upper Saddle River, NJ, USA,1993.

[84] L. Everett, L. –S Wang, and S. Hannenhalli. “Dense subgraph computation via stochastic search: application to detect transcriptional modules”, In ISBM (Supplement of Bioinformatics), pages 117-123, 2006.

[85] F.B Zhan and C. E. Noon, “Shortest Path algorithms: an evaluation using real road network”, Transportation science, 32(1):65-73, 1998.

[86] I. Chabini. “Discrete dynamics shortest path problems in transportation applications: Complexity and algorithms with optimal run time,” Transportation Research Records, 1645:170-175,1998.

[87] P. Ji, Y. Z. Wu, H. Z. Liu, and H.T. Wu, "The Vehicle routing problem with time-varying travel tiles and a solution method," International Journal of Innovation Computing Information and Control, 5(4):1001-1011, 2009.

[88] A. Osvald and L. Z. Stirn. "Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length," Journal of the Association for Computing Machinery, 37(3):607-625, 1990.

[89] B. Ding et all. "Finding time-dependent shortest path in a large graphs", In EDBT: Proceedings of the 11th International conference on Extending database technology, pages 205-216, NY,USA, 2008.

[90] van Aerde, M., Hellinga, B., Yu, L. and Rahka, H. (2003) Vehicle probes as realtime ATMS sources of dynamic O-D and travel time data. In Large Urban Systems: Proceedings of the Advanced Traffic Management Conference, Pages: 207–230, 2003.

[91] Hunter, T., R. Herring, P. Abbeel, and A. Bayen. "Path and travel time inference from GPS probe vehicle data." Neural Information Processing Systems Foundation (NIPS) Conference, Vancouver, Canada, 2009.

[92] Jenelius, Erik, Rahmani, Mahmood, Koutsopoulos, Haris N, "Travel Time Estimation for Urban Road Networks Using Low-Frequency GPS Probes",Transportation Research Board 91st Annual Meeting, Washington, DC, 2012.

[93] Li, Y. and McDonald, M. "Link Travel Time Estimation Using Single GPS Equipped Probe Vehicle." Proceedings of The IEEE 5th International Conference on Intelligent Transportation Systems, IEEE, Singapore, 932-937, September 2002.

[94] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. J. Nordlund. "Partilce Filters for Positioning, Navigation, and Tracking." IEEE Transactions on Signal Processing, pages 425-437, Feb. 2002.

[95] N. J. Gordon, D. J. Salomond, and A. F. M. Smith. "Novel approach to nonlinear/non Gaussian Bayesian state estimation." Radar and signal Processing, IEE Processing F, 140(2): 107-113, 1993.

[96] C. Hue, J.-P. Le Cadre, and P. Perez. "Sequential Monte Carlo methods for multiple target tracking and data fusion." IEEE Transactions on Signal Processing, pages 309–325. 2002.

[97] M. Isard and A. Blake, "CONDENSATION - Conditional Density Propagation for Visual Tracking," International Journal on Computer Vision 1(29) 5-28,1998.

[98] X. Wang, R. Chen, and J. S. Liu. Monte Carlo Bayesian signal processing for wireless communications. Journal of VLSI Signal Processing, 30(1–3 (Special issue on Signal Processing for Wireless Communication Systems)) :89–105, Jan.–Feb.–Mar. 2002.

[99] M. Rosenbluth and A. Rosenbluth. “Monte Carlo calculation of the average extention of molecular chains,” Journal of Chemical Physics, pages 356-359, 1955.

[100] J. M. Hammersley and K. W. Morton. Poor man’s monte carlo. Journal of the Roayal Statistics Society, pages 23-38, 1954.

[101] N. J. Gordon, D. J. Salomond, and A. F. M. Smith. “Novel approach to nonlinear/non Gaussian Bayesian state estimation,” Radar and signal Processing, IEE Processing F, 140(2): 107-113, 1993.

[102] M. Irwing, N. Cox, and A. Kong. “Sequential imputation for multilocus linkage analysis.” Proc. Natl. Acad. Sci. USA, 91:11684-11688, 1994.

[103] Dieter Fox and Wolfram Burgard. “Markov localization for mobile robots in dynamic environment.” Jounal of artificial intelligence research,11:391-427, 1999.

[104] A. Kong, J. S. Liu, and W. Wong. “Sequetial imputation method and Bayesian missing data problems,” Journal of the American Statistical Association, 89:278-288, 1994.

[105] Ke Zhang, Guangtao Xue," A Real- time urban traffic detection algorithm based on spatio-temporal OD matrix in vehicular sensor network ", Wireless Sensor Network, Scientific Research Journal, 668-674, 2004.

[106] Karin F. M. Aronsson, “Speed characteristics of urban streets based on driver behaviour studies and simulation”, KTH School of Architecture and the Built Environment, Doctoral Thesis in Infrastructure Royal Institute of Technology Stockholm, Sweden 2006.

[107] T. Hunter, R. Herring, P. Abbeel, A. Bayen, “Path and Travel Time Inference from GPS Probe Vehicle Data”, Neural Information Processing Systems Foundation, Vancouver, Canada, 2009.

[108] Jun S. Liu and Rong Chen. “Sequential Monte Carlo Methods for Dynamics Systems”, Journal of the American Statistical Association, 93:1032-1044, 1999.

[109] Zou, L., Xu, J.-M. and Zhu, L.-X. “Arterial speed studies with taxi equipped with global positioning receivers as probe vehicle.” Proceedings of the

International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1343-1347, 2005.

[110] Work, D. B., Tossavainen, O.-P., Blandin, S., Bayen, A. M., Iwuchukwu, T. and Tracton, K. "An ensemble Kalman filtering approach to highway traffic estimation using GPS enabled mobile devices." Proceedings of the 47th IEEE Conference on Decision and Control, pp. 5062-5068, 2008.

[111] Rahmani, M., and Koutsopoulos, H. N. "Path inference of low-frequency GPS probes for urban networks." Technical report, KTH Royal Institute of Technology, Stockholm, Sweden, 2012.

[112] Miwa, T., Sakai, T., and Morikawa, T. "Route identification and travel time prediction using probe-car data." International Journal of ITS Research 2, 21-28, 2008.

[113] Miller, J., Kim, S., Ali, M. and Menard, T. "Determining time to traverse road sections based on mapping discrete GPS vehicle data to continuous flows." IEEE Intelligent Vehicles Symposium, pp. 615–620, 2010.

[114] Hellinga, B., Izadpanah, P., Takada, H. and Fu, L. "Decomposing travel times measured by probe-based traffic monitoring systems to individual road links." Transportation Research Part C 16, 768–782, 2008.

[115] Zheng, F. and van Zuylen, H. "Urban link travel time estimation based on sparse probe data." Transportation Research Part C, in press, 2012.

[116] Van Lint, J. W. C and H. J Van Zuylen. "Monitoing and predicting freeway travel time reliability- using width and skew of day-to-day travel time distribution," Transaction Research Record 1917:54-62, 2005.

[117] The official website of the project Pumas (Plateforme Urbaine de Mobilite Avancee et Soutenable / Sustainable and Advanced Mobility and Urban Platform), http://www.projet-pumas.fr

[118] Martinez, W.L., Martinez, A.R.. "Computational Statistics Handbook with MATLAB." Chapman & Hall/CRC, 2002.

[119] Dey, P.P., Chandra, S., Gangopadhaya, S. "Speed distribution curves under mixed traffic conditions." Journal of Transportation Engineering, ASCE, 475–481, 2006.

[120] Zhu, W., Boriboonsomsin, K., Barth, M. "Microscopic traffic flow quality of service from the drivers' point of view." IEEE Intelligent Transportation Systems Conference, Seattle, WA, 2007.

[121] Hastie, T., Tibshirani, R., Friedman, J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer 2001.

[122] Jean Laneurit, Roland Chapuis, and Frédéric Chausse "Accurate Vehicle Positioning on a Numerical Map," International Journal of Control, Automation, and Systems, vol. 3, no. 1, pp. 15-31, March 2005.

[123] Ying Lee, "Freeway Travel Time Forcast Using Artifical Neural Networks with Cluster Method", 12th Internation Conference on Information Fusion, Seattle, WA, USA, 2009.

[124]Dougherty, M, "A Review of Neural Networks Applied to Transport", Transport Research-Part C, Vol.3, No.4, pages 247-260, 1995.

[125] Zhang, X and Rice, J. A, "Short-term travel time prediction", Transport Research-Part C, Vol.3, No.11, pages 187-210, 2003.

[126] Ke Zhang, Guangtao Xue, "A Real time urban traffic detection algorithm based on spatio-temporal OD matrix in vehicular sensor network", Scientific Research Journal: Wireless Sensor Network, 2010.

[127]Bauza R, Gozalvez J, Sanchez-Soriano J, "Road Traffic Congestion Detection through Cooperative Vehicle-to-Vehicle Communication", IEEE 35th Conference on Local Computer Networks, 2010.

[128] Adam Pel, Michiel Bliemer, Serge Hoogendoorn, "Modelling Traveller Behavior under Emergency Evacuation Conditions", EJTIR, Issue 11(2), pages 166-193, ISSN:1567-7141, 2011.

[129] Amnir Hadachi, Christele Lecomte, Stephane Mousset and Abdelaziz Bensrhair,"An Application of the Sequential Monte Carlo to Increase the Accuracy of Travel Time Estimation in Urban Areas", 14th International IEEE Conference on Intelligent Transportation Systems, Washington, DC, USA. October 5-7, 2011.

[130] Amnir Hadachi, Stéphane Mousset, Abdelaziz Bensrhair, "Practical Testing Application of Travel Time Estimation Using Applied Monte Carlo Method and Adaptive Estimation from Probes", IEEE Intelligent Vehicles Symposium, Alcalá de Henares, Spain, June 3-7, 2012.

[131] Amnir Hadachi, Stephane Mousset, Christele Lecomte and Abdelaziz Bensrhair," Travel Time Estimation Using Cooperative Probe Vehicles", The 6th International Symposium on signal, Image, Video and Communications, Valenciennes, France on July 4-6, 2012.

[132] Amnir Hadachi, Stephane Mousset, Abdelaziz Bensrhair, "Increasing the Travel Time Accuracy in Arterial Networks using Particle Filter and Enhanced Map Matching." Submitted to the ITS Transaction Journal 2012.

[133] Malinovskiy Y., Wu Y., Wang Y. and Lee U., "Field Experiments on Bluetooth-based Travel 473 Time Data Collection," In Transportation Research Board 89th Annual Meeting. Transportation Research Board, Washington D.C., 2010.

[134] Kwong, K., & Kavaler, R. Arterial Travel Time Estimation Based on Vehicle Re-Identification Using Wireless Magnetic Sensors. Transportation Research Part C: Emerging Technologies. Vol 17, No. 6, pages 586-606, 2009.

[135] Sen, A., Thakuriah, P., Zhu, X.-Q., and Karr, A.: Frequency of probe reports and variance of travel time estimates, Journal of Transportation Engineering,Vol.123, No.4, Pp.290-297, 1997.

[136] Sudarshan Nandy , Partha Pratim Sarkar and Achintya Das, "TRAINING A FEED-FORWARD NEURAL NETWORK WITH ARTIFICIAL BEE COLONY BASED BACKPROPAGATION METHOD", International Journal of Computer Science & Information Technology (IJCSIT) Vol 4, No 4, 2012.

[137] C. Furtlehner, J.M. Lasgouttes, and A.D.L. Fortelle, "A Belief Propagation Approach to Traffic Prediction Using Probe Vehicles," Proc. IEEE Intelligent Transportation Systems (ITS) Conf., pp. 1022-1027, 2007.

[138] Egis is a consulting and engineering group working in the fields of transport, urban development, construction, industry, water, environment and energy. In the road and airport fields, the group is also involved in project financing, turnkey-contract projects and facility operation. http://www.egis-group.com/

[139] Meldrum R. Dierdre y Cynthia E. Taylor, "WA-RD 365.1 FREEWAY TRAFFIC DATA PREDICTION USING ARTIFICIAL NEURAL NETWORKS AND DEVELOPMENT OF A FUZZY LOGIC RAMP METERING SYSTEM", http://www.wsdot.wa.gov/ppsc/research/onepages/WA-RD3651.htm, 1995.

[140] Pyo, J., Shin, D., Sung, T. "Development of a map-matching method using the multiple hypothesis technique". IEEE Proceedings on Intelligent Transportation Systems, 23–27, 2001.

[141] White, C.E., Bernstein, D., Kornhauser, A.L. "Some map-matching algorithms for personal navigation assistants". Transportation Research Part C 8, 91–108, 2000.

[142] Bouju, A., Stockus, A., Bertrand, F., Boursier, P. "Location-based spatial data management in navigation systems". IEEE Symposium on Intelligent Vehicle 1, 172–177, 2002.

[143] Yu, M., Chen, W., Li, Z., Chen, Y. "Improvement on integrity and reliability of vehicle positioning by a new map-matching method". In: Proceedings of the 17th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS 2004, pp. 2086–2094, 2004.

[144] Fu, M., Li, Jie, Wang, M. "A hybrid map-matching algorithm based on fuzzy comprehensive Judgment". IEEE Proceedings on Intelligent Transportation Systems, 613–617, 2004.

[145] Syed, S., Cannon, M.E. "Fuzzy logic-based map-matching algorithm for vehicle navigation system in urban canyons". In proceedings of the Institute of Navigation (ION) national technical meeting, California, USA, 2004.

[146] Srinivasan, D., Cheu, R.L. "Development of an improved ERP system using GPS and AI techniques". IEEE Proceedings on Intelligent Transportation Systems, 554-559, 2003.

[147] Yang, D., Cai, B., Yuan, Y. "An improved map-matching algorithm used in vehicle navigation system". IEEE Proceedings on Intelligent Transportation Systems 2, 1246–1250, 2003.

[148] OpenStreetMap, "www.openstreetmap.org", retrieved on October 2010.

[149] Hunter, T., Herring, R., Abbeel, P. and Bayen, A. "Path and travel time inference from GPS probe vehicle data". Neural Information Processing Systems Foundation (NIPS) Conference, Vancouver, Canada, 2009.

[150] Min, X., Hu, J., Chen, Q., Zhang, T. and Zhang, Y. "Short-term traffic flow forecasting of urban network based on dynamic STARIMA model". 12th International IEEE Conference on Intelligent Transportation Systems, pp. 461-466, 2009.

[151] Herring, R., Hofleitner, A., Amin, S., Abou Nasr, T., Abdel Khalek, A., Abbeel, P. and Bayen, A. "Using mobile phones to forecast arterial traffic through statistical learning". TRB Annual Meeting, 2010.

[152] Westgate, B. S., Woodard, D. B., Matteson, D. S., and Henderson, S. G. "Travel time estimation for ambulances using Bayesian data augmentation". ORIE Technical Reports, 2011.